



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA
ÁREA DE COMPUTACIÓN Y ELECTRÓNICA



T E S I S

SISTEMA DE PROCESO DE EVALUACIÓN DE PERFIL LINGÜÍSTICO, CASO DE
ESTUDIO: ALUMNOS DEL IDIOMA ALEMÁN APLICANDO EL PROGRAMA
INSTITUCIONAL DE LENGUAS DEL CENTRO DE LENGUAS DE LA UAEH

PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS COMPUTACIONALES

PRESENTA:
P.L.C.C CAMPOS GUZMAN LESLIE

ASESOR:
M.I.D ALBERTO SUÁREZ NAVARRETE

CO-ASESOR:
M.A.E DAVID HERNÁNDEZ SÁNCHEZ

MINERAL DE LA REFORMA, HIDALGO, MÉXICO 2023



Mineral de la Reforma, Hgo., a 27 de noviembre de 2023

Número de control: ICBI-D/1819/2023
 Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio le comunico que el Jurado asignado a la Pasante de la Licenciatura en Ciencias Computacionales **Leslie Campos Guzmán**, quien presenta el trabajo de titulación “**Sistema de Proceso de evaluación del perfil lingüístico; caso de estudio alumnos del idioma alemán aplicando el Programa Institucional de lenguas del Centro de Lenguas de la UAEH**”, después de revisar el trabajo en reunión de Sinodales ha decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente M.C.C. Gonzalo Alberto Torres Samperio

Secretario: Dr. Edgar Olguín Guzmán

Vocal: MID. Alberto Suárez Navarrete

Suplente: MID. Ma. de Jesús Gutiérrez Sánchez

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente
 “Amor, Orden y Progreso”

Dr. Otilio Arturo Acevedo Sandoval
 Director del ICBI



OAASYCC

Ciudad del Conocimiento
 Carretera Pachuca-Tulancingo km 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
 Teléfono: 771 71 720 00 ext. 2231 Fax 2109
 direccion_icbi@uaeh.edu.mx





UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences
Área Académica de Computación y Electrónica
Department of Electronics and Computer Science

Mineral de la Reforma. Hgo., a 22 de noviembre del 2023

Número de control: ICBI-AACyE/1446/2023
Asunto: integración en el repositorio institucional.

MTRO. JORGE E. PEÑA ZEPEDA
DIRECTOR DE BIBLIOTECAS Y CENTROS DE INFORMACIÓN.

Por medio del presente hago constar que la tesis en formato digital titulado: "Sistema de proceso de evaluación de perfil lingüístico, caso de estudio: alumnos del idioma alemán, aplicando el programa institucional de lenguas del centro de lenguas de la UAEH", que presenta Leslie Campos Guzmán, con número de cuenta 319672, es la versión final validada por el Comité Tutorial y cumple con el oficio de autorización de impresión, por lo que solicito su integración en el repositorio institucional de tesis.

Sin otro particular, me despido de usted.

Atentamente
"Amor Orden y Progreso"

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Dr. Edgar Olgún Guzmán
Coordinador de la Licenciatura en
Ciencias Computacionales

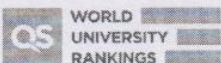


Leslie Campos Guzmán

Instituto de Ciencias Básicas e Ingeniería
Área Académica de Computación y Electrónica

EOG/PMZ

Ciudad del Conocimiento
Carretera Pachuca-Tulancingo km 4.5 Colonia
Carboneras, Mineral de la Reforma, Hidalgo,
México. C.P. 42184
Teléfono: +52 (771) 71 720 00 ext. 2250, 2251
Fax 2109
aacye_icbi@uaeh.edu.mx



www.uaeh.edu.mx



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Centro de Lenguas

Language Center

Mineral de la Reforma, Hidalgo, a 3 de octubre de 2023.

CEL/738/2023

ASUNTO: Liberación de Proyecto para Titulación

A quien corresponda,

Por este medio, informo que el proyecto titulado *"Sistema de proceso de evaluación de perfil lingüístico. Caso de estudio: alumnos del idioma alemán, aplicando el Programa Institucional de Lenguas del Centro de Lenguas de la UAEH"*, el cual fue desarrollado desde el 3 de enero de 2022 hasta el 14 de septiembre de 2023, ha sido liberado para fines de Titulación de la alumna Leslie Campos Guzman.

De acuerdo con los resultados obtenidos en su utilización, se concluye que esta herramienta tecnológica permite la generación dinámica y aplicación de evaluaciones basadas en los procesos de evaluación lingüística de la Universidad Autónoma del Estado de Hidalgo establecidos en el *Programa Institucional de Lenguas del Centro de Lenguas*, establecidos en el Programa Institucional de Lenguas de esta misma institución, y que tiene la capacidad de efectuar sugerencias de estudio con las que se seguimiento al rendimiento académico de los alumnos.

Este proyecto, por su capacidad de ser una nueva forma de efectuar evaluaciones de lenguas extranjeras, se encuentra en calidad de factible. Asimismo, se valida la originalidad del trabajo desarrollado con la aplicación de la metodología SCRUM, la cual implica las actividades de preparación, planeación, diseño y resultados, elaborado con la participación de Leslie Campos Guzman, estudiante de la Licenciatura en Ciencias Computacionales de esta institución, con número de cuenta 319672, además de contar con la supervisión de Fernando Daniel Tehozol y O. Omar S. Juárez, por parte del Centro de Lenguas de la UAEH, y la asesoría del MAE David Hernández Sánchez y del MID Alberto Suárez Navarrete.

Atentamente
"Amor, Orden y Progreso"

Edward Amador Pliego
Director



Ciudad del Conocimiento
Carretera Pachuca-Tulancingo, km 4.5
Col. Carboneras, Mineral de la Reforma,
Hidalgo, México; C.P. 42184
Teléfono: 52 (771) 71 720 00 Ext. 4581
cel@uaeh.edu.mx



www.uaeh.edu.mx

Agradecimientos

A Dios, por brindarme la protección y fuerza necesarias en cada etapa de mi vida, en los tiempos más difíciles, y guiar mi camino.

A la Universidad Autónoma del Estado de Hidalgo por permitirme estudiar y finalizar mi programa educativo de licenciatura en esta honorable institución y por otorgarme las herramientas necesarias para mi formación académica. También quiero expresar mi agradecimiento al personal docente de la Licenciatura en Ciencias Computacionales por transmitir su conocimiento que me enriqueció profesionalmente y con el cual fue posible la realización del presente proyecto.

Al MAE David Hernández Sánchez por su apoyo y guía durante el inicio de este trabajo de tesis, y al MID Alberto Suárez Navarrete por su tiempo y asesoría para su conclusión. Un especial agradecimiento a ambos por su orientación, consejos y el trato amable y cordial que recibí.

Al personal del Centro de Lenguas conformado por Fernando Daniel Tehozol y O. Omar S. Juárez por su compromiso y la oportunidad que me brindaron para la creación de este proyecto y llevarlo a mi proceso de titulación, así como también por todo el aprendizaje que obtuve de ambos en el uso de varias tecnologías durante todo el tiempo que comprendió el desarrollo de este sistema; y la Lic. Sandra Hornung por ayudarme a ampliar mi dominio de la lengua alemana y por su servicial y amable asesoría en el desarrollo de este sistema en lo referente al lenguaje.

A mi madre Erika Massiel Guzman Cruz y mis abuelos Jaime Guzman Zamora y Leticia Cruz Aguilar por los valores que me enseñaron, su confianza, paciencia y su continuo apoyo en general en todas las etapas de mi vida, y con los cuales, alcancé este que es uno de los logros más importantes dentro de mi formación universitaria y de mi vida. A mis tíos Jorge Alonso Guzman C., Narda Margarita Guzman C. y Carlos Alberto Gonzales M., y mi primo Carlos Daen Gonzales G. por estar a mi lado en los altos y bajos y auxiliarme en las situaciones difíciles. A Martín Garnica de Dios por su apoyo incondicional y con quien siempre estaré agradecida. A Sara G. Cruz Y Cruz por su ayuda que me brindó para ampliar mi conocimiento.

Índice general

Agradecimientos	I
Índice general	II
Índice de figuras	VI
Índice de ilustraciones	VIII
Índice de tablas	IX
Índice de gráficas	X
Resumen	XI
Abstract	XII
Introducción	XIII
Planteamiento del problema	XIV
Justificación	XV
Propuesta de solución	XVI
Objetivos	XVIII
Objetivo general	XVIII
Objetivos específicos	XVIII
Alcances	XIX
Limitaciones	XX
Aportaciones	XXI
1. Capítulo I. Marco Teórico	1
1.1 Fundamento teórico	1
1.1.1 Aprendizaje de lenguas extranjeras	1
1.1.2 Evaluaciones	2
1.1.3 Aprendizaje de la lengua alemana en la UAEH	4
1.2 Marco Conceptual	5
1.2.1 Base de datos	5
1.2.2 Framework	6
1.2.3 Librería	6
1.2.4 Modelo-Vista-Controlador - MVC	7

1.2.5 Programación Orientada a Objetos - POO.....	8
1.2.6 Objeto	8
1.3 Marco tecnológico.....	9
1.3.1 Herramientas tecnológicas	9
1.3.2 Lenguajes de programación, modelado y consulta	19
1.3.3 Frameworks y librerías.....	40
1.4 Estado del arte.....	43
1.4.1 Plataforma digital	43
1.4.2 Google Forms	44
1.4.3 Moodle.....	45
1.4.4 GexCat	46
1.4.5 Testmoz.....	47
2. Capítulo II. Metodología.....	49
2.1 SCRUM	49
2.2 Extreme Programming - XP	52
2.3 Dynamic Systems Development Method - DSDM	54
2.4 Adaptive Software Development - ASD.....	55
2.5 Metodología a emplear	56
3. Capítulo III. Diseño y desarrollo	60
3.1 Análisis de requerimientos	60
3.1.1 Requerimientos funcionales.....	61
3.1.2 Requerimientos no funcionales.....	62
3.2 Diagramas de Casos de Uso	64
3.2.1 Caso de uso para alumnos	64
3.2.2 Caso de uso para docente y administrador.....	64
3.2.3 Caso de uso para crear una evaluación.....	66
3.2.4 Caso de uso para visualizar preguntas asignadas a la evaluación	67
3.2.5 Caso de uso para eliminar una evaluación	67
3.2.6 Caso de Uso para contestar una evaluación.....	68
3.3 Diagramas de Actividades	70
3.3.1 Diagrama de actividades para crear una evaluación.....	70

3.3.2 Diagrama de actividades para visualizar preguntas asignadas a la evaluación	72
3.3.3 Diagrama de actividades para eliminar una evaluación	72
3.3.4 Diagrama de actividades para contestar una evaluación	74
3.4 Diagrama de clases	75
3.5 Diseño arquitectónico	77
3.6 Diseño de la base de datos.....	79
3.6.1 Modelo Entidad-Relación.....	79
3.6.2 Modelo relacional.....	83
3.6.3 Descripción de la base de datos	85
3.7 Metodología empleada: SCRUM.....	102
3.7.1 Designación de roles	103
3.7.2 Product Backlog.....	103
3.7.3 Sprint Planning	104
3.7.4 Sprint Backlog	104
3.7.5 Sprint	105
3.7.6 Daily Scrum	105
3.7.7 Trabajo Terminado	106
3.7.8 Retroalimentación del Sprint.....	106
3.7.9 Retrospectiva del Sprint.....	107
3.8 Herramientas empleadas	108
3.8.1 Herramientas tecnológicas	109
3.8.2 Lenguajes de programación, modelado, y estilos	113
3.8.3 Frameworks y librerías.....	117
4. Capítulo IV. Resultados	119
4.1 Desarrollo de la interfaz principal de la plataforma	119
4.2 Desarrollo de la interfaz para visualizar la lista de alumnos	120
4.3 Desarrollo de la interfaz para crear un nuevo examen	121
4.4 Desarrollo de la interfaz para visualizar el listado de exámenes	126
4.5 Desarrollo de la interfaz para visualizar las preguntas asignadas a un examen	128

4.6 Desarrollo de la interfaz para visualizar detalles acerca de una pregunta en particular	129
4.7 Desarrollo de la interfaz para visualizar un examen	131
4.8 Desarrollo de la interfaz para visualizar los resultados obtenidos	135
4.9 Desarrollo de la interfaz para visualizar los reportes de un examen.....	139
4.10 Validación	141
4.10.1 Validación	141
4.10.2 Conclusiones de las pruebas de validación	155
5. Conclusiones	158
6. Trabajo futuro.....	161
7. Referencias	162
8. Apéndices.....	171
8.1 A. Manual de Usuario: Plataforma para la creación de exámenes aplicados a los alumnos en el Centro de Lenguas.	172
8.1.1 Escritorio de la plataforma	172
8.1.2 Ver alumnos	173
8.1.3 Crear un nuevo examen	173
8.1.4 Abrir listado de exámenes creados.....	176
8.1.5 Ver resultados	183
8.2 B. Cuestionario de validación para la Plataforma para la creación de exámenes aplicados a alumnos en el Centro de Lenguas	185
8.2.1 Funcionalidad	185
8.2.2 Usabilidad.....	186
8.2.3 Confiabilidad.....	188
8.2.4 Eficiencia	188
8.3 C. Modelo Entidad-Relación del Sistema de proceso de evaluación de perfil lingüístico.....	190
8.4 D. Modelo Relacional del Sistema de proceso de evaluación de perfil lingüístico.....	192
9. Anexos	194
9.1 Anexo 1	195

Índice de figuras

Figura 1. Interfaz de Xampp.....	13
Figura 2. Sintaxis de JSON.....	21
Figura 3. Estructura básica de un documento HTML.....	23
Figura 4. Sintaxis de CSS.....	26
Figura 5. Proceso de Scrum.....	50
Figura 6. Proceso de la metodología XP	53
Figura 7. Fases de ASD.....	56
Figura 8. Diagrama de casos de uso para alumnos	65
Figura 9. Diagrama de casos de uso para docente o administrador	65
Figura 10. Diagrama de casos de uso para crear un examen	66
Figura 11. Diagrama de casos de uso para visualizar preguntas asignadas a un examen	67
Figura 12. Diagramas de casos de uso para eliminar un examen	68
Figura 13. Diagrama de casos de uso para contestar un examen.....	69
Figura 14. Diagrama de actividades para crear un examen	71
Figura 15. Diagrama de actividades para la visualización de preguntas asignadas a un examen	73
Figura 16. Diagrama de actividades para eliminar un examen	73
Figura 17. Diagrama de actividades para contestar un examen	74
Figura 18. Diagrama de clases	76
Figura 19. Patrón arquitectónico MVC	78
Figura 20. Diagrama Entidad-Relación (A).....	81
Figura 21. Diagrama Entidad-Relación (B).....	82
Figura 22. Modelo relacional.....	84
Figura 23. Interfaz principal del Sistema de proceso de evaluación de perfil lingüístico.....	120
Figura 24. Interfaz para visualizar el listado de alumnos	121
Figura 25. Primera sección del formulario para crear un nuevo examen.....	122
Figura 26. Segunda sección del formulario para crear un nuevo examen	122
Figura 27. Tercera sección del formulario para crear un nuevo examen	123
Figura 28. Cuarta sección del formulario para crear un nuevo examen.....	123
Figura 29. Final de la interfaz para crear un nuevo examen.....	124
Figura 30. Ejemplo 1 de mensajes de error.....	125
Figura 31. Ejemplo 2 de mensajes de error.....	125
Figura 32. Diseño de la interfaz para visualizar listado de exámenes creados ...	126

Figura 33. Mensaje del sistema que indica que no existen registros de exámenes	128
Figura 34. Diseño de la interfaz para ver las preguntas asignadas a un examen	128
Figura 35. Diseño de la interfaz para visualizar detalles acerca de una pregunta en específico.....	130
Figura 36. Diseño de la interfaz para contestar un examen.....	131
Figura 37. Pregunta de tipo abierta	132
Figura 38. Pregunta de tipo lista desplegable	132
Figura 39. Pregunta de tipo selección múltiple	133
Figura 40. Función AJAX para el guardado de respuestas	134
Figura 41. Resumen de los resultados obtenidos en un examen	136
Figura 42. Ejemplo de retroalimentación en tipos de pregunta opción múltiple y lista desplegable	137
Figura 43. Ejemplo de retroalimentación en tipo de pregunta selección múltiple	137
Figura 44. Retroalimentación de pregunta sin responder	138
Figura 45. Vista del guardado de resultados	139
Figura 46. Diseño de la interfaz de reportes.....	140
Figura 47. Escritorio del sistema	172
Figura 48. Lista de alumnos	173
Figura 49. Ejemplo de llenado de formulario para crear un nuevo examen	175
Figura 50. Campos disponibles y no disponibles para la creación de un examen con fines de egreso	175
Figura 51. Listado de exámenes creados.....	177
Figura 52. Vista de la ventana donde el alumno podrá realizar su examen.....	178
Figura 53. Mensaje de confirmación para eliminar un examen.....	180
Figura 54. Reporte de resultados ordenado por semana	180
Figura 55. Ejemplo de Listado de las preguntas de un examen	181
Figura 56. Información de una pregunta en específico.....	182
Figura 57. Mensaje que aparece cuando se intenta acceder a un examen que ya fue respondido	183
Figura 58. Vista general de los resultados obtenidos en un examen.....	184
Figura 59. Diagrama Entidad-Relación	191
Figura 60. Modelo relacional.....	193

Índice de ilustraciones

Ilustración 1. Logotipo de Visual Code Studio	9
Ilustración 2. Logotipo de Sublime Merge	10
Ilustración 3. Logotipo de XAMPP	12
Ilustración 4. Logotipo de StarUML	14
Ilustración 5. Logotipo de GitLab.....	15
Ilustración 6. Diagrama de clases	29
Ilustración 7. Diagrama de objetos	30
Ilustración 8. Diagrama de componentes	30
Ilustración 9. Diagrama de despliegue	32
Ilustración 10. Diagrama de actividades.....	34
Ilustración 11. Diagrama de casos de uso.....	35
Ilustración 12. Diagrama de tiempo	36
Ilustración 13. Diagrama de secuencia	37
Ilustración 14. Diagrama de comunicación	38
Ilustración 15. Logo de Laravel	40

Índice de tablas

Tabla 1. Cuadro comparativo de metodologías ágiles.....	57
Tabla 2. Análisis de la tabla para el alumno	85
Tabla 3. Análisis de la tabla para la relación entre el alumno y el programa educativo	86
Tabla 4. Análisis de la tabla para los datos de aplicación de los exámenes	87
Tabla 5. Análisis de la tabla para el idioma	87
Tabla 6. Análisis de la tabla para instituto	88
Tabla 7. Análisis de la tabla para la materia o asignatura	88
Tabla 8. Análisis de la tabla para la modalidad	89
Tabla 9. Análisis de la tabla para los tipos de datos de elementos multimedia.....	89
Tabla 10. Análisis de la tabla para el nivel	90
Tabla 11. Análisis de la tabla para el programa educativo	90
Tabla 12. Análisis de la tabla para el semestre	91
Tabla 13. Análisis de la tabla para el subnivel.....	91
Tabla 14. Análisis de la tabla para el tipo o propósito del examen	92
Tabla 15. Análisis de la tabla para el tipo de pregunta	92
Tabla 16. Análisis de la tabla para la unidad del programa temático	93
Tabla 17. Análisis de la tabla para el examen	94
Tabla 18. Análisis de tabla para las habilidades del examen.....	94
Tabla 19. Análisis de la tabla para la relación entre el examen y su nivel	95
Tabla 20. Análisis de la tabla para la relación entre el examen y las preguntas ...	95
Tabla 21. Análisis de la tabla para la relación entre instituto y el programa educativo	96
Tabla 22. Análisis de la tabla para los elementos multimedia.....	96
Tabla 23. Análisis de la tabla para la relación entre los elementos multimedia y la pregunta.....	97
Tabla 24. Análisis de la tabla para las preguntas	98
Tabla 25. Análisis de la tabla para las respuestas.....	98
Tabla 26. Análisis de la tabla para los resultados.....	99
Tabla 27. Análisis de la tabla para la temática	100
Tabla 28. Análisis de la tabla para la relación entre la temática y la pregunta	100
Tabla 29. Análisis de la tabla para la aplicación de un examen a un estudiante.	101
Tabla 30. Factores del proceso de calidad de la norma ISO/IEC 9126.	141

Índice de gráficas

Gráfica 1. Resultados de la primera pregunta de Funcionalidad	144
Gráfica 2. Resultados de la segunda pregunta de Funcionalidad	144
Gráfica 3. Resultados de la tercera pregunta de Funcionalidad	145
Gráfica 4. Resultados de la cuarta pregunta de Funcionalidad	146
Gráfica 5. Resultados de la quinta pregunta de Funcionalidad.....	146
Gráfica 6. Resultados de la primera pregunta de Usabilidad.....	147
Gráfica 7. Resultados de la segunda pregunta de Usabilidad	148
Gráfica 8. Resultados de la tercera pregunta de Usabilidad.....	149
Gráfica 9. Resultados de la cuarta pregunta de Usabilidad	150
Gráfica 10. Resultados de la quinta pregunta de Usabilidad	150
Gráfica 11. Resultados de la sexta pregunta de Usabilidad	151
Gráfica 12. Resultados de la primera pregunta de Confiabilidad	152
Gráfica 13. Resultados de la segunda pregunta de Confiabilidad	153
Gráfica 14. Resultados de la primera pregunta de Eficiencia	154
Gráfica 15. Resultados de la segunda pregunta de Eficiencia.....	154
Gráfica 16. Resultados de la tercera pregunta de Eficiencia	155
Gráfica 17. Porcentaje de aceptación en cada factor evaluado.....	156
Gráfica 18. Nivel de aceptación de la plataforma	157

Resumen

Con el propósito de mejorar el proceso de elaboración de exámenes utilizado por el Centro de Lenguas, dependencia de la Universidad Autónoma del Estado de Hidalgo (UAEH), se planteó y llevó a cabo el desarrollo de una plataforma web que permitiera a los usuarios realizar esa tarea de forma más rápida y simple. Además, se implementó la función de sugerir a los alumnos, según las preguntas respondidas erróneamente, los temas que deberían estudiar con la finalidad de elevar su dominio del idioma.

En el presente documento se describe la metodología utilizada durante la elaboración de la plataforma, así como sus pasos. Seguido, se enlistan las herramientas tecnológicas, los diferentes lenguajes de programación y frameworks empleados. Sobre el análisis del proyecto, se enlistan los requerimientos funcionales y no funcionales descritos por el Centro de Lenguas de la UAEH. Posteriormente, se muestra la fase de diseño: a través de diagramas UML se ilustran el comportamiento y las actividades realizadas por la plataforma, se incluyen los diagramas del diseño de la base de datos, así como el desarrollo de las interfaces pertenecientes a este módulo de la plataforma. Después, se exponen los resultados obtenidos en las diferentes pruebas de validación, las conclusiones, trabajo futuro, las referencias empleadas a lo largo del documento, en los apéndices se encuentran el manual de usuario, el cuestionario utilizado para la validación del sistema, y los modelos entidad-relación y relacional de la base de datos. Por último, se encuentran los anexos.

Palabras clave.

Plataforma, framework, base de datos, metodología ágil, MVC.

Abstract

In order to improve the exam elaboration process used by the Language Center, a department of the Autonomous University of the State of Hidalgo (UAEH), the development of a web platform that would allow users to perform this task in a faster and simpler way was proposed and carried out. In addition, it was implemented the function of suggesting to the students, according to the questions answered wrongly, the topics they should study in order to increase their proficiency in the language.

This document describes the methodology used during the development of the platform, as well as its steps. Next, the technological tools, the different programming languages and frameworks used are listed. About the analysis of the project, the functional and non-functional requirements described by the Language Center of the UAEH are listed. Subsequently, the design phase is shown: through UML diagrams, the behavior and activities performed by the platform are illustrated, the diagrams of the database design are included, as well as the development of the interfaces belonging to this module of the platform. Then, the results obtained in the different validation tests, conclusions, future work, references used throughout the document, the user's manual, the questionnaire used for the validation of the system, and the entity-relationship and relational models of the database are presented in the appendices. Finally, there are the annexes.

Key words.

Platform, framework, database, agile methodology, MVC.

Introducción

Actualmente, el Centro de Lenguas ha elaborado los exámenes que aplican a los estudiantes en muchos de los institutos y dependencias de la UAEH mediante formularios en línea que, si bien han cumplido su objetivo hasta ahora, su proceso de creación resultó tardado y con mayor tendencia a errores.

La solución propuesta en este trabajo es el desarrollo de una plataforma web a medida que consiga sustituir el método de elaboración de exámenes actual por uno más simple pero robusto a su vez, el cual cubra los requisitos expresados por el Centro, haciendo uso de diversas herramientas tecnológicas y lenguajes de programación como JavaScript, PHP, hojas de estilo en cascada (CSS), lenguaje de marcado de hipertexto para páginas web (HTML), entre otros. Asimismo, esta plataforma debe:

- Incorporar la capacidad de efectuar sugerencias a los estudiantes, basándose en su desempeño en el examen aplicado.
- Mostrar la retroalimentación con los temas que deben reforzar con el propósito de elevar su dominio del idioma y que logren los objetivos y/o competencias marcadas por sus respectivos programas educativos.

Planteamiento del problema

El Centro de Lenguas (CeL), encargado de la enseñanza de diversos idiomas en la UAEH, lleva a cabo evaluaciones a sus estudiantes con diferentes propósitos, desde exámenes parciales de múltiples programas educativos de la institución, hasta medición de conocimientos de determinada lengua. En los días que corren, el CeL realiza sus evaluaciones haciendo uso de formularios en línea y software especializado en la creación de cursos digitales.

Aunque estos recursos cumplen su cometido, es cierto que en ellos se han encontrado errores en la navegación a través de las preguntas, pues en lugar de utilizar botones especializados para ello, se contaba con botones propios de la reproducción de elementos multimedia como videos o audios para ir a la siguiente pregunta o la anterior; en ocasiones el diseño empleado dificulta la lectura de los textos, como la utilización de colores de fondo que asemejan al de la letra, o el diseño era demasiado elaborado que propiciaba distracciones; no es claro el modo de interactuar con el examen, ya que los botones se bloqueaban sin razón aparente o no reaccionaban de la manera esperada; la calificación, producto de los resultados, no era claro cómo fue que se obtuvo ni qué criterios empleó para otorgarla, entre otros.

Asimismo, la seguridad que estos exámenes poseían era lo suficientemente baja como para que un estudiante, con los conocimientos de computación necesarios, lograra obtener las respuestas para usarlas a su favor, poniendo en duda su dominio real de las habilidades del lenguaje que se le estuviera evaluando.

Por otra parte, el proceso actual de elaboración de un examen toma su tiempo, ya que esos software requieren de la escritura manual de las preguntas, añadir elementos multimedia, opciones, y señalar la respuesta correcta para cada uno para que pueda calificar la evaluación automáticamente; además, mientras menor sea el tiempo requerido para crear un examen, y mayor sea la cantidad de estos a generar, la tendencia a errores ortográficos y gramaticales incrementa. Por ello, si se considera la necesidad de crear mayor número de ellos con diferentes características o propósitos, se volvería una tarea ardua y laboriosa.

Igualmente, estas herramientas no garantizaban que el progreso se guardara y volviera a ser visible en cuanto se reanudara el examen, en caso de algún accidente en que la computadora se desconecte de la fuente de luz, se pierda la conexión a Internet, se cierre el navegador, o cualquier situación similar.

Justificación

La aplicación de evaluaciones que miden el nivel de dominio del estudiante de una lengua extranjera es, independientemente de su propósito, una actividad con la cual es posible determinar en qué medida el alumno puede aplicar los conocimientos en este tema y, con ello, realizar diferentes actividades o trámites como la asignación de una calificación a un estudiante de algún programa educativo, obtención de un documento o constancia que acredite el nivel de dominio del idioma, etc.

Es por ello que, al ser muchas de estas evaluaciones aplicadas de forma digital en múltiples institutos, preparatorias y demás dependencias de la universidad, es importante contar con un sistema que garantice la seguridad del examen en términos de protección a las respuestas, característica que algunos de los software empleados por el Centro de Lenguas no toman en cuenta, lo que pone en vulnerabilidad a la prueba y la obtención de resultados cuestionables por parte de los estudiantes, como ya se mencionó. Así, con este módulo, se busca implementar un sistema que permita realizar exámenes de calidad de forma simple y rápida, concediendo la obtención de resultados veraces en los perfiles lingüísticos de los alumnos y que, con la automatización de este proceso, la universidad se vea favorecida en el ahorro de recursos materiales y de tiempo.

Además, al ser un proyecto elaborado a medida, desde cero y que pasaría a ser propiedad de la UAEH, la institución se vería beneficiada directamente en la eliminación de la posibilidad de un pago de licencias en el caso de que se hiciera la búsqueda de algún software en el mercado cuyas funcionalidades asemejen a las que el Centro requiere. Igualmente, la plataforma está planeada para que la curva de aprendizaje de ésta se reduzca respecto a la que podría generarse con la implementación de una aplicación de terceros.

Propuesta de solución

La solución propuesta ante la problemática descrita, es el desarrollo del Módulo Sistema de proceso de evaluación de perfil lingüístico, en este caso para el estudio de alumnos del idioma alemán aplicando el Programa Institucional de Lenguas del Centro de Lenguas de la UAEH, cuyo propósito será el de facilitar y acelerar el proceso de elaboración de las diferentes clases de evaluaciones que aplica este departamento, para obtener exámenes de calidad que cumplan su propósito sin algún tipo de inconveniente, y que el manejo de esta herramienta digital sea óptimo. Además, que éste incorpore y se encuentren siempre a disposición de los usuarios, las funciones que otras aplicaciones (software) utilizadas en el presente no comprenden o que no actúan de la manera esperada.

Para el desarrollo de este sistema de plataforma, será necesario que el Centro de Lenguas establezca los requerimientos con los que debe contar, las funcionalidades que deberá incluir, y que toda la información se muestre en interfaces que sean intuitivas para todos los usuarios sin importar el rol que éstos posean (por ejemplo, alumnos, catedráticos, administradores). Lo anterior, con el fin de que el producto final sea de calidad y que garantice su operatividad en todo momento.

Como primer paso, se elaborarán los diagramas UML con los que se puedan expresar las funcionalidades e interacciones que el sistema tendrá con su entorno con ayuda de una herramienta de modelado basado en este lenguaje (StarUML).

Después, se construirá el diagrama entidad-relación para establecer las tablas que conformarán la base de datos de este sistema y, posteriormente, se realizará el diagrama relacional para representar las tablas de una forma más estándar, mediante una plataforma que provea de los elementos necesarios para la elaboración de diagramas como éstos (Miro). Además, deberá contarse con un medio del cual puedan obtenerse elementos multimedia que auxilien en el desarrollo de esta plataforma (Pixabay).

Este sistema será construido empleando lenguajes de programación para la comunicación con el servidor (PHP), para añadir dinamicidad a cada una de las interfaces (JavaScript) acompañado de una librería que pueda simplificar la manipulación del DOM (jQuery), el lenguaje de marcado de hipertextos para estructurar cada interfaz (HTML), un lenguaje para el intercambio de información dentro del sistema (JSON), y se requerirá de un framework para que el proceso de desarrollo sea uno más rápido y eficiente (Laravel). Igualmente, se hará uso de un

motor de plantillas que ayude en la separación de la lógica del sistema de las vistas (Blade).

También es necesario contemplar un lenguaje de consulta (MySQL) que pueda ser utilizado para manipular la información almacenada en una base de datos, así como también se requerirá de un gestor de bases de datos en donde ésta pueda ser creada (Workbench). Para la escritura del código del sistema, será necesario emplear un editor de código (VS Code), y un servidor local con el que se le puedan realizar múltiples pruebas al sistema a fin de garantizar su funcionamiento (XAMPP). Por último, para gestionar cada uno de los cambios realizados al sistema, se necesitará hacer uso de un repositorio donde puedan almacenarse sus distintas versiones (GitLab), así como de un cliente que permita su guardado (Sublime Merge).

Objetivos

Objetivo general

Desarrollar una plataforma que permita la implementación de exámenes digitales que solucione los problemas generados por las actuales herramientas empleadas, mediante metodologías y herramientas para su construcción e implementación.

Objetivos específicos

- Diseño del Front-End, para construir las interfaces del sistema; y del Back-End, que contendrá toda su lógica, utilizando diversas tecnologías como lenguajes de programación, frameworks, librerías, etc.
- Crear una base de datos que facilite el análisis y consulta de la información mediante el diseño de un modelo entidad-relación (E-R) que sirva de referencia para la elaboración de su script empleando el lenguaje SQL.
- Estructurar la evaluación con preguntas de varios tipos, mediante la implementación de plantillas para hasta 4 tipos de preguntas con las que pueda delimitarse la amplitud de la respuesta esperada por el alumno.
- Generar la lista de sugerencias de estudio, y reportes para fines estadísticos y de seguimiento que puedan ser ordenados por día, por mes o por año mediante la obtención de resultados determinados por las respuestas correctas e incorrectas de los alumnos.
- Para evitar pérdida de información, guardar periódicamente en la base de datos, las respuestas ingresadas hasta el momento utilizando AJAX, JSON y otras herramientas tecnológicas.
- Validar la plataforma mediante la norma ISO 9126 y la escala de Likert para su ponderación, desarrollando los ítems con la taxonomía de Bloom.

Alcances

El Sistema de proceso de evaluación de perfil lingüístico tiene como propósito proponer una alternativa al proceso de creación de evaluaciones del idioma alemán realizado por docentes y administradores, y ofreciendo la posibilidad de su resolución mediante este módulo que hace énfasis en estas actividades.

Con estas actividades como punto de partida, se pretende desarrollar un módulo de la plataforma que sea capaz de generar dinámicamente evaluaciones, mostrar datos de ellas y de los alumnos, permitir la resolución de evaluaciones, obtener resultados conformados por la retroalimentación, sugerencias de estudio y la calificación, y generar reportes a partir de esta última. Las sugerencias de estudio, y los reportes que podrán ser ordenados por día, semana o mes, podrán ser impresos cuantas veces los usuarios lo requieran.

Para ello, se utilizarán varios recursos tecnológicos como son un framework que sirva de base para el desarrollo del sistema, lenguajes de programación para crear tanto funciones con tareas específicas, como el diseño del aspecto visual, creación de cada una de las vistas, un lenguaje de consulta para la generación de la base de datos, lenguajes de modelado para la diagramación de la estructura del sistema, y algunas herramientas como editores de código, un servidor local para realizar pruebas, un gestor de versiones, etc.

Limitaciones

El Sistema de proceso de evaluación de perfil lingüístico incluye, únicamente, las funciones propias del módulo que fueron solicitadas por el Centro de Lenguas. Por ello, la plataforma se encontrará en calidad de finalizado una vez que los demás módulos sean desarrollados.

El sistema contará con la capacidad de generar evaluaciones para algunos tipos o propósitos, pues para generar exámenes de todos ellos, será necesario contar con una muy vasta colección de preguntas para crear apenas uno de ellos; y, de hecho, las preguntas con las que se hicieron pruebas durante su desarrollo, fueron proporcionadas por el Centro de Lenguas, así como varios de los elementos multimedia. Asimismo, se contará sólo con un método para calificar las evaluaciones en este sistema, el cual es el utilizado para las evaluaciones ordinarias.

Se presenta la limitación de que no se cuenta con ningún otro módulo de la plataforma, por lo que no será posible la separación del sistema con las funciones propias para el alumnado y las propias para los docentes y administradores.

Aportaciones

El desarrollo de este proyecto tecnológico implicó la realización de un estudio diagnóstico de la situación en la que se encontraba el Centro de Lenguas (CeL) con el empleo de herramientas de terceros, concluyendo que estas no cumplían con las exigencias de este departamento. Por ello, la aportación más relevante es la creación de este sistema que fue diseñado y programado atendiendo los requerimientos expresados por el CeL, ya que no sólo tiene la capacidad de resolver la problemática planteada, sino que, por su naturaleza enfocada a la evaluación de asignaturas, su base de datos y sus componentes puede ser utilizados como referencia para la realización de proyectos similares dentro de la institución.

Derivado de la característica de los exámenes de ser aplicados de manera digital y ser reutilizados, el ahorro de recursos de tiempo y materiales de la institución será notorio, favoreciendo la reducción del gasto en la inversión de elementos con los que tradicionalmente se realizan las evaluaciones.

Por la capacidad del Sistema de evaluación de perfil lingüístico de mostrar los resultados y retroalimentación de la evaluación, los alumnos pueden realizar un análisis y autorreflexión de su desempeño en cada tópico o habilidad evaluada, y serán capaces de mejorar su rendimiento gracias a la guía proporcionada por el sistema en forma de sugerencias de estudio, de modo que puedan enfocarse en fortalecer sus conocimientos en las áreas donde presenten mayores debilidades.

La función de reportes de las evaluaciones permite conocer el desempeño de los alumnos en el dominio de la lengua alemana a lo largo del periodo escolar; situación que orientará mejor al personal docente y administrativo de la Universidad, a una mejor toma de decisiones que apoyen a los alumnos a alcanzar los objetivos esperados.

Este proyecto informático enfatiza la evaluación de las habilidades de los alumnos en el dominio de la lengua alemana. Por ello, además de los reportes generados, se sugiere la integración de Inteligencia Artificial al sistema para ayudar a determinar las áreas o temas estudiados donde los alumnos suelen tener mayores dificultades, de modo que sea posible decidir un plan de acción que minimice esa situación. Asimismo, que su integración pueda también determinar mejor las sugerencias de estudio proporcionadas mediante un análisis más profundo de las respuestas, y no determinado únicamente porque la respuesta ingresada fue errónea.

1. Capítulo I. Marco Teórico

1.1 Fundamento teórico

El aprendizaje de una lengua extranjera es una práctica que trae consigo beneficios para los estudiantes para su desarrollo personal y profesional. Es por ello que la Universidad Autónoma del Estado de Hidalgo, con la misión de formar estudiantes con habilidades competentes, que sean capaces de integrarse al ámbito laboral para brindar contribuciones y aportaciones a la sociedad, incluye en muchos de sus programas educativos, y dependiendo de estos, asignaturas en diferentes lenguas tales como inglés, francés, italiano y alemán. En cada periodo escolar, como forma para determinar en qué medida los alumnos han alcanzado los objetivos esperados, se aplican evaluaciones con distintos propósitos.

1.1.1 Aprendizaje de lenguas extranjeras

La Universidad Autónoma del Estado de Hidalgo (UAEH), para alcanzar su internacionalización, optó por la incorporación de enseñanza y aprendizaje de lenguas extranjeras como parte de los diferentes programas educativos que oferta mediante asignaturas especializadas para ello y, de acuerdo a estos, de idiomas como son el inglés, francés, italiano y alemán. Para el alumnado, la adquisición de las habilidades y conocimientos que implica aprender un nuevo idioma significaría la oportunidad para establecer relaciones de manera nacional e internacional, enriquecer su cultura, e incluso, mejorar su perfil de egreso.

Las asignaturas de idiomas impartidas por la Universidad, se encuentran coordinadas por la Dirección Universitaria de Idiomas a fin de que el proceso de enseñanza y aprendizaje sea óptimo y congruente entre todos los institutos donde son instruidas. Por otra parte, con el propósito de que los alumnos logren los objetivos esperados, se creó el Programa Institucional de Lenguas (PIL) (DUI-UAEH, 2013), que contiene las actividades, estrategias y metas que deben aplicarse como parte del proceso de enseñanza (Castañón García et al., 2013).

Este documento señala la segmentación de cada nivel de enseñanza tomando como referencia el Marco Común Europeo de Referencia (MCER), el cual establece una escala de las habilidades que el alumno adquiere desde el nivel más básico,

A1, hasta el nivel más avanzado, C2; además, cada uno de los niveles del MCER también se encuentran divididos en subniveles (Castañón García et al., 2013). Utilizando como guía los alcances establecidos en cada uno de los niveles del MCER, se favorece la escritura de evaluaciones de calidad que midan con exactitud la capacidad lograda en cada una de las cuatro habilidades que integran el aprendizaje de una lengua extranjera.

Estas evaluaciones son realizadas con diferentes propósitos que pueden ir desde la asignación de una calificación de una materia, hasta la realización de trámites como pueden ser ingreso, egreso o medición de conocimientos.

1.1.2 Evaluaciones

Una evaluación es un proceso sistemático cuyo objetivo principal es la recopilación y análisis de información sobre el progreso logrado por los estudiantes en una materia en particular. Esta información es habitualmente utilizada para análisis respecto al desempeño de los alumnos y que sirva de punto de partida para la toma de decisiones que encaminen a la mejora de su rendimiento académico (Escuela de Profesores del Perú, 2023).

Estas evaluaciones sirven de instrumento para la UAEH para conocer qué tanto los alumnos han adquirido los conocimientos y habilidades esperados dentro de un periodo de tiempo. Para ello, la institución realiza diferentes tipos de pruebas con diferentes propósitos. De entre estas evaluaciones que la institución implementa, se ejemplifican algunas a continuación (Castañón García et al., 2013):

- Evaluaciones iniciales o diagnósticas, para determinar en qué medida un alumno es capaz de satisfacer los objetivos marcados en un programa educativo. Estas evaluaciones son aplicadas a alumnos en el inicio de los periodos escolares, particularmente a los alumnos de nuevo ingreso o en el primer nivel de la lengua extranjera que cursarán durante su programa educativo. Esta evaluación se distingue por centrarse en conocer los alcances de las fortalezas y debilidades de los alumnos y, dependiendo del resultado, los alumnos serán desplazados al nivel que se ajusta mejor a sus conocimientos.
- Evaluaciones procesuales o formativas, que son realizadas a lo largo de cada nivel de enseñanza del idioma y su propósito es medir progresivamente el rendimiento de los alumnos. Además, el CeL integra dentro de las evaluaciones procesuales, algunas otras, que son:

- La autoevaluación, que posee un valor de 5% de la calificación final (o dependiendo del programa de lenguas y de la Academia Institucional de Lenguas), que tiene como fin que el alumno realice una autorreflexión de su desempeño.
- La coevaluación, con un valor de 10% de la calificación final, y que cuyo propósito es evaluar el desempeño de sus compañeros.
- Y la heteroevaluación, que posee un valor de 85% de la calificación final, y que contempla todos los componentes que integran la medición de las habilidades en su forma más básica (comprensión y expresión oral y escrita), aunque también se le pueden integrar otros instrumentos de evaluación como exposiciones, portafolio de evidencias, proyectos, listas de cotejo, tareas, entre otros. Además, el CeL propone que estas evaluaciones tengan como peso sobre la calificación global de los alumnos, el 30% para el primer y segundo periodo parcial, y 40% en el último, obteniendo así el 100% de la calificación.
 - Exámenes departamentales, que miden todo el conocimiento adquirido durante el nivel de enseñanza del idioma. Por ello, estos exámenes son aplicados durante el último periodo parcial por su característica de ser acumulativos, cuestión que los distingue de los exámenes procesuales que solo evalúan el aprendizaje adquirido en un periodo de tiempo.
 - Evaluación sumativa, que es aplicada a los alumnos cuando culminan el último nivel de aprendizaje de una lengua extranjera. Estos exámenes sirven para medir los logros alcanzados respecto al nivel A2 en bachillerato, y B1 en licenciatura.
 - Acreditación, que libera al alumno de la obligación de cursar la asignatura y otorga la calificación obtenida en todos los niveles de enseñanza del idioma en su programa educativo, y solo puede ser logrado presentando un documento oficial que acredite las habilidades del alumno.

Aunque solo se presentaron los ejemplos más comunes de evaluaciones en este documento, dentro del Programa Institucional de Lenguas (PIL), se encuentran algunos otros con especificaciones de los instrumentos con los que los alumnos son evaluados, los porcentajes que poseen cuando son integrados para obtención de una calificación general, las condiciones bajo las cuales son aplicados y los programas educativos y semestres para los que van dirigidos.

1.1.3 Aprendizaje de la lengua alemana en la UAEH

En el año 2010, la Universidad Autónoma del Estado de Hidalgo integró la enseñanza de la lengua alemana para el Nivel Medio Superior. Esta enseñanza consta de cinco asignaturas pertenecientes a Bachillerato, donde cada una tiene destinadas 400 horas de estudio divididas en 60 horas de clases y 20 horas de trabajo independiente por cada uno de los semestres, resultando en un total de 23.5 créditos con una cantidad de 4.65 por cada uno de los cinco semestres. Éstas son *Alemán I*, *Alemán II*, *Alemán III*, *Alemán IV* y *Alemán V*.

Dentro de esas asignaturas, se considera el aprendizaje para comprensión de frases aisladas y expresiones utilizadas con mayor frecuencia, intercambio de información simple, la capacidad de describir de forma sencilla su información y la del ambiente en que se encuentra. Así, el alumno concluirá su aprendizaje habiendo alcanzado el nivel A2+ de acuerdo al MCER, además de adquirir la capacidad de presentar exámenes de certificación con validez internacional.

Además de la enseñanza de este idioma, la UAEH también ofrece evaluaciones de certificación a través de *Fit in Deutsch 1 y 2*, el cual está dirigido a jóvenes de entre 15 hasta 17 años y que evalúa el nivel A1 de esta lengua; *Start Deutsch 1 y 2*, que va dirigido a adultos y que evalúa los niveles A1 y A2; y *Deutsch Zertifikat für Jugendlichen*, que también va dirigido a jóvenes ante instancias evaluadoras con el Instituto Goethe en México y Alemania, y que evalúa el nivel B1 (Castañón García et al., 2013).

1.2 Marco Conceptual

El marco conceptual está conformado por una serie de conceptos que son mencionados frecuentemente en varias secciones de este documento, y para cada uno se incluye más de una definición. Esto con el propósito de que el lector tenga una idea clara de todos ellos, ya que en los siguientes apartados serán útiles para comprender varios de los componentes del sistema y algunos recursos tecnológicos que serán utilizados en su desarrollo. Los conceptos que fueron seleccionados para esta sección son: base de datos, framework, librería, MVC, orientación a objetos y objeto.

1.2.1 Base de datos

Mencionar este concepto es indispensable, pues buena parte del funcionamiento del sistema construido depende de la información almacenada en la base de datos. La definición de base de datos proporcionada por el glosario de Mozilla, es que una base de datos es un sistema de almacenamiento cuyo objetivo es la recolección de datos organizados para facilitar actividades tales como búsqueda de información, estructuras, etc. (Mozilla, 2023).

(Benitez & Arias, 2017) por su parte, argumentan que una base de datos es un conjunto de datos dispuestos destinados a proporcionar información a los usuarios, y permitirles realizar diferentes clases de transacciones como inserciones, eliminación y actualización de los datos. También, exponen que hay diferentes modelos para construir una base de datos:

- **Flat file.** Archivos planos contenedores de información.
- **Relacional.** Su nombre proviene del modo en que organiza los datos en tablas y establece relaciones entre sí.
- **Orientado a objetos.** Organiza los datos en clases y objetos.
- **Jerárquico.** Organiza los datos en forma de árbol.
- **Red.** Organiza los datos en tablas conectados por referencias, dando como resultado una estructura muy parecida a una red.
- **Dimensional.** Organiza los datos en múltiples dimensiones, simplificando así las consultas de los datos.

El tipo de base de datos a crear en el presente proyecto corresponde al tipo relacional.

1.2.2 Framework

Conocer el concepto de framework permitirá comprender la forma en la que el sistema fue construido. Un framework es un esquema de software que promueve la reducción de tiempo de construcción e implementación de un software, y facilita su escalabilidad y mantenibilidad. Los frameworks también proveen funcionalidades extendidas a un determinado lenguaje de programación, favorece la automatización de procesos, y estructura el código de forma que este sea más entendible y se encuentre organizado. Habitualmente, los frameworks se dividen en tres capas, como se explica a continuación (Martínez Villalobos et al., 2010):

- Lógica de presentación, que se encarga de administrar las interacciones efectuadas entre el usuario y el sistema.
- Lógica de datos, que se encarga de administrar el acceso a un agente de almacenamiento persistente.
- Lógica de dominio, que se encarga de manipular los modelos de datos en función de las órdenes recibidas en la presentación.

Otra definición proporcionada por (IONOS, 2023) dicta que un framework, traducido del inglés como *infraestructura* o *armazón*, no es un programa autónomo, sino que se trata de una especie de biblioteca de clases y que cuentan con modelos de desarrollo que sirven para el proceso de construcción de una aplicación totalmente nueva e independiente. Así, un framework vendría siendo la representación arquitectónica de la aplicación.

1.2.3 Librería

El motivo por el que se hace mención de la librería es porque es necesario aclarar la diferencia que existe entre este concepto y el de framework. Una librería se puede definir como un *conjunto de archivos de código* que sirve para el desarrollo de software a fin de facilitar la tarea de programación, gracias a que provee funciones que ya han sido creadas por otros programadores. Así, las librerías son una herramienta especial que ayuda a los programadores a evitar la posible duplicación de código y favorece la minimización de errores (Equipo de datos.gob.es, 2022).

Una segunda definición de este concepto indica que una librería es una *librería de programas*. Una librería está conformada por subprogramas que, en el proceso de desarrollo de software, facilitan su programación con las funciones que provee.

Contrario a un framework, el uso de una librería se hace de manera determinada y posee funciones que se ajustan entre sí. Estas librerías se vinculan a un software y se accede a ellas solamente cuando es necesario, por tanto, estas solo se vuelven funcionales dentro del entorno de la aplicación, y pueden ser ejecutadas independientemente (IONOS, 2023).

1.2.4 Modelo-Vista-Controlador - MVC

Correspondiente a *Modelo-Vista-Controlador*, el término MVC se añade a esa sección debido a la importancia que cobra al ser la arquitectura bajo la cual el presente sistema fue construido. MVC es un patrón de diseño de software utilizado para la implementación de interfaces de usuario, datos y la lógica de control, y su característica más distintiva es la separación entre la lógica de negocios y la visualización (MDN contributors, 2022).

Una definición más amplia proporcionada por (Caballé Llobet & Xhafa, 2007) indica que MVC es un patrón de diseño de aplicaciones que proporciona un alto nivel de modularidad en las aplicaciones, sobre todo en la interfaz de usuario. El propósito de este patrón de diseño es que los cambios de una parte signifiquen un número bajo de cambios en otras partes del programa que se esté desarrollando. Como se mencionó anteriormente, MVC separa la capa del modelo, la capa de presentación y la capa de eventos producidos de una aplicación, siendo este el producto de las acciones efectuadas por el usuario. También describen cada una de las partes que integran a este modelo de la siguiente forma:

- **Modelo.** Es el modelo del dominio de la aplicación.
- **Vista.** Es la interfaz gráfica que presenta la información al usuario.
- **Controlador.** Gestiona los eventos resultantes de las acciones del usuario, haciendo saber al modelo y/o a la vista de los cambios que eso implica para ambos.

1.2.5 Programación Orientada a Objetos - POO

La programación orientada a objetos puede entenderse como una forma de programación donde el código es organizado en clases, y cuyo propósito es crear objetos que se interrelacionan para cumplir los objetivos para los que fue construida la aplicación informática. Esta forma de programar se fundamenta en la idea de describir los objetos del mundo real lo más parecido posible y, distinto de la programación estructural, un software elaborado con programación orientada a objetos se constituye de clases, objetos, propiedades, métodos, etc., y no con subrutinas o funciones (Alvarez, 2021).

Otra definición indica que la programación orientada a objetos hace más énfasis en los objetos que son manipulados por los usuarios, en lugar de la lógica necesaria para esa manipulación. Además, la programación orientada a objetos es utilizada más frecuentemente en la construcción de aplicaciones grandes y complejas que requieren más actualizaciones (Universidad Europea, 2022).

1.2.6 Objeto

(Guagliano, 2019) explica que, en un ambiente cotidiano, los objetos son todas aquellas cosas que cumplen una función y poseen características propias. Estos objetos pueden ser tanto cosas tangibles como abstractas. De este modo, dentro del paradigma orientado a objetos, el modelado de objetos permite la resolución de problemas del mismo modo como se haría en la realidad. Sin embargo, los objetos resultantes no serán modelos totalmente fieles al objeto del que toman inspiración, sino que también se les asignan una serie de características necesarias que sirvan para resolver el problema.

Otra definición de lo que es un objeto viene dada por (Cervantes O. et al., 2016), donde definen que un objeto es una entidad virtual con datos y funciones, y que tiene como propósito simular las propiedades de un objeto del mundo real. También explican que los objetos, vistos desde el punto de vista en un ámbito real, poseen dos atributos: un estado, que indica la situación interna del objeto como su velocidad, calificación, color, capacidad, etc.; y el comportamiento, que es la respuesta a estímulos del ambiente. Por otro lado, en el punto de vista computacional, un objeto tiene dos características distintas: los atributos, que corresponden a los datos de un objeto (características); y los métodos, que refieren al comportamiento del objeto.

1.3 Marco tecnológico

1.3.1 Herramientas tecnológicas

Para el desarrollo de software, se emplean diversos recursos tecnológicos con distintos propósitos, y que acomoden mejor a sus objetivos. El desarrollo de esta plataforma, desde su diseño, requirió el empleo de software que facilitara los elementos con los que se elaborarían los diagramas que expresaran su estructura. Para la fase de programación, era recomendable contar con otros recursos tecnológicos que facilitaran la escritura del código y realización de pruebas como editores, repositorios para gestionar el proyecto, etc.

1.3.1.1 Visual Studio Code

Visual Studio Code (o VS Code), cuyo logo se muestra en la ilustración 1, es un editor de código fuente gratuito y multiplataforma disponible para Windows, Linux y OS X, capaz de soportar múltiples lenguajes de programación, de marcado y de bases de datos tales como JavaScript, C#, C++, PHP, Java, HTML, TypeScript, JSON, XML, Python, etc. VS Code también puede integrarse con administradores de paquetes, repositorios, entre otros, a fin de optimizar el flujo de trabajo diario. Este editor es totalmente compatible con Git y proporciona buenos flujos de trabajo con este sistema de control de versiones distribuido (Kahlert & Giza, 2016).



ILUSTRACIÓN 1. LOGOTIPO DE VISUAL CODE STUDIO

VS Code proporciona todas las herramientas o funciones de forma simple, que los desarrolladores necesitan durante las fases del proceso, o ciclo de creación, y depuración de código, pero de modo que estas puedan ser aprovechadas al máximo. Por ejemplo, algunas de estas funciones incluyen el resaltado de la

sintaxis, la coincidencia de paréntesis, sangría automática, compatibilidad con una amplia lista de idiomas, entre otras (Kahlert & Giza, 2016).

También provee soporte para la edición de código, navegación y comprensión, acompañado de la función de depuración ligera y un modelo de extensibilidad vasto; además, es capaz de integrar otras herramientas a las ya existentes. Derivado de la función de actualización integrada, mes a mes, VS Code es actualizado con el propósito de proporcionar nuevas funciones, así como la resolución de errores, asegurando así que Visual Studio Code se mantenga actualizado en todo momento (GitHub, 2023).

Además de las funciones y lenguajes de programación ya mencionados al principio, los desarrolladores pueden sumar muchos más gracias a la integración de extensiones que se mantienen independientes del sistema operativo (Kahlert & Giza, 2016).

1.3.1.2 Sublime Merge

Sublime Merge es un editor de texto multiplataforma compatible con Windows, Mac y Linux, que puede ser descargado de forma gratuita; no obstante, para seguir utilizándolo requiere pago de licencia (Sublime Merge, s.f.). También es un cliente gráfico de Git, pues envuelve la funcionalidad central de ese sistema de control de versiones (Sublime Merge, s.f.).



ILUSTRACIÓN 2. LOGOTIPO DE SUBLIME MERGE

(Dinder, 2022) define a este software como *“una forma simple de combinar código y realizar operaciones Git”*. Por otro lado, (Skinner, 2018) lo describe como *“la combinación del motor de la interfaz de usuario de Sublime Text y una implementación desde cero de Git”*. En ese sentido, Sublime Merge, cuyo logo se muestra en la ilustración 2, ofrece una herramienta de combinación capaz de

resolver los posibles conflictos de combinación suscitados en las ramas de forma directa sin necesidad de abrir el editor (Ubunlog, 2022).

Sublime Merge ofrece las siguientes características (Ramesh, 2018):

- **Herramienta de combinación integrada.** Esta herramienta le permite al desarrollador resolver dificultades dentro del entorno de Sublime Merge, mostrando los cambios realizados por uno mismo del lado izquierdo, y los cambios hechos por otros se mostrarán del lado derecho.
- **Diferencias avanzadas.** Cada que sea necesario, Sublime Merge indicará con exactitud los cambios realizados en los caracteres. Estos cambios refieren a modificaciones de nombre, movimientos, problemas resueltos y el historial de *commits*.
- **Atajos de teclado.** Sublime Merge ofrece opciones de usabilidad en el teclado. Por ejemplo, se puede navegar a través de la aplicación con la tecla Tab, se puede alternar la expansión por la barra espaciadora, entre otros. Por tanto, la paleta de comandos otorgará mayor velocidad a los usuarios en el acceso a los comandos Git.
- **Integración de línea de comandos.** Los cambios efectuados en el repositorio son visibles en tiempo real, y las cosas funcionan en la línea de comandos como lo harían en la interfaz de usuario.
- **Búsqueda avanzada.** Sublime Merge integra un buscador el cual, conforme se escribe, busca las coincidencias exactas en mensajes de *commit*, autores de los *commit*, nombres de archivos, etc.
- **Uso de Git real.** Al trabajar con Sublime Merge, se trabaja a la vez con Git real, y no como una versión simplificada. Utiliza una implementación personalizada capaz de leer repositorios favoreciendo funcionalidades de alto rendimiento. Aun así, Git se utiliza directamente en Sublime Merge para ejecutar operaciones como clonación de repositorios, consulta de ramas, etc.

1.3.1.3 XAMPP

XAMPP es una distribución de Apache desarrollada por Apache Friends de uso totalmente gratuito y que contiene las tecnologías de desarrollo web más comunes. Fue creada con el propósito de ofrecer una distribución cuya instalación resulte un proceso simple para los desarrolladores principiantes en el mundo de Apache; una vez XAMPP es instalado, todas sus configuraciones y opciones ya se encuentran activadas de manera predeterminada. Igualmente, su contenido, su tamaño reducido y portabilidad hacen de XAMPP una herramienta ideal para la creación y

testeo de aplicaciones en PHP y MySQL. Además, como es el caso de las distribuciones Linux, XAMPP es una compilación de software; por ello, su uso no requiere pago de licencias y puede ser copiado de manera libre bajo la licencia GNU GPL (Apache Friends, 2022; Dvorski, 2007). El logo de este software puede apreciarse en la ilustración 3.

XAMPP está disponible en dos paquetes específicos: el paquete completo y el paquete ligero. El primero provee una amplia matriz de herramientas de desarrollo, mientras que el segundo contiene Apache HTTP Server, PHP, MySQL, phpMyAdmin, OpenSSL, y SQLite (Dvorski, 2007).



ILUSTRACIÓN 3. LOGOTIPO DE XAMPP

Además de estos programas, XAMPP también contiene, dependiendo del sistema operativo, otras herramientas como son el servidor de correo Mercury, el programa de analítica web Webalizer, Apache Tomcat y los servidores de FTP FileZilla o ProFTPd (IONOS, 2023).

El nombre de XAMPP deriva de un acrónimo formado por las letras iniciales de los programas que lo conforman: Apache, un servidor web de código abierto; MySQL/MaríaDB, ambos gestores de bases de datos; PHP, que es un lenguaje de programación empleado para el lado del servidor; y Perl, un lenguaje de programación empleado para la administración del sistema. En cuanto a la letra X restante, esta es una representación de los sistemas operativos Windows, Mac OS X y Linux (Carrión Bou, 2019).

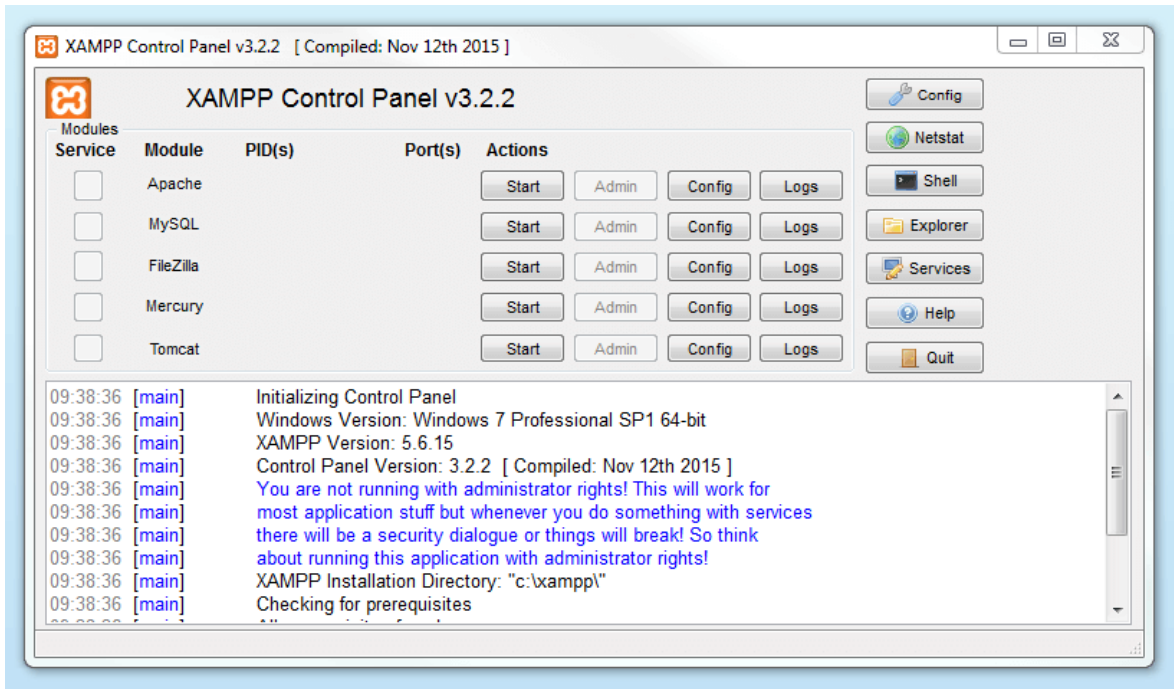


FIGURA 1. INTERFAZ DE XAMPP.

Fuente: (IONOS, 2023)

En la figura 1 se muestra la interfaz de usuario del panel de control de XAMPP, esta se conforma de los diferentes módulos que contiene este software, los cuales se activan y desactivan de forma separada. También incluye otras utilidades como son (IONOS, 2023):

- **Config.** Su función es la de configurar XAMPP y otros componentes aislados.
- **Netstat.** Muestra los procesos funcionando en el momento en el dispositivo local.
- **Shell.** Su función es la de lanzar una ventana de comandos UNIX.
- **Explorer.** Cuando se da clic, abre la carpeta de XAMPP en el explorador de Windows.
- **Services.** Muestra los servicios funcionando en el momento.
- **Help.** Expone enlaces que redirigen a foros de usuarios.
- **Quit.** Botón cuya función es la de salir del panel de control de XAMPP.

1.3.1.4 StarUML

StarUML es una herramienta tecnológica multiplataforma disponible para Windows, MacOS y Linux, cuya utilidad es el modelado de software basándose en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture) para admitir un modelado ágil y conciso. Fue creado con los objetivos de servir a usuarios pertenecientes a equipos de desarrollo ágiles pero pequeños, para profesionales y para institutos educativos (Zamenfeld, 2011; StarUML.io, 2022). El logo de esta herramienta es presentado en la ilustración 4.



ILUSTRACIÓN 4. LOGOTIPO DE STARUML

(López Ortega & Santa Villa, 2012) explican que StarUML es un proyecto rápido y flexible de software libre que pretende resaltar de entre la competencia de herramientas UML como una opción convincente. Este software se actualiza constantemente a fin de ser compatible con las nuevas versiones de UML. Igualmente, exponen que StarUML posee una herramienta introducida por OMG, llamada MDA, herramienta para la cual StarUML sirve de apoyo, además de ofrecer variables como lo son la personalización del perfil UML, enfoque, marco del modelo, NX (extensión de la notación), código MDA, etc.

Uno de los principios fundamentales de StarUML es ofrecer variedad de diagramas UML en su versión 2.0, entre los que se encuentran diagramas de casos de uso, de clases, de secuencia, de colaboración, de estados, de actividades, entre otros. Otra característica que posee es la capacidad de generar código y de ingeniería inversa con los lenguajes de programación Java, C++ y C#; así como también generar documentos con extensiones propias de Microsoft Office, códigos personalizables para el usuario y archivos de comandos activados (JScript). La razón por la que StarUML fue implementado con la idea central de proveer características “amigables” para los usuarios es porque, de acuerdo a los desarrolladores de este software, la usabilidad es la característica más importante con la que debe contar un sistema tecnológico (López Ortega & Santa Villa, 2012).

1.3.1.5 GitLab

Antes de entender qué es GitLab, es necesario conocer qué es Git. Git es un sistema de control de versiones no centralizado dirigido al mantenimiento de versiones de aplicaciones con ficheros de código. Hoy en día, es probable que este sea el sistema de control de versiones más utilizado a nivel global derivado de las características útiles que provee, tales como su eficiencia, eficacia y sencillez; puede ser utilizado en su forma más básica por los principiantes en el mundo de Git y a su vez puede ser utilizado de forma muy compleja por expertos (inLab FIB, 2022).



ILUSTRACIÓN 5. LOGOTIPO DE GITLAB

Ahora bien, GitLab, cuyo logo se muestra en la ilustración 5, es un sistema de control de versiones utilizado, por lo general, en el área de desarrollo de software. GitLab fue escrito y publicado por Dimitri Saporoschcz en el año 2011, está basado en la web y fue escrito con el lenguaje de programación Ruby on Rails. Como su nombre sugiere, está basado en Git, del cual está disponible una versión gratuita como software de código abierto. Además, GitLab también es considerado como una alternativa ante GitHub, otro sistema de control de versiones que goza de buena popularidad (IONOS, 2022).

Este sistema ofrece tres formas diferentes de utilizarlo (Torrado, 2017):

1. Usarlo para una empresa, ámbito profesional o para una organización dentro de sus propios servidores sin generar costes adicionales. Con esta opción, es necesario mencionar que es de carácter obligatorio mantener el servidor, realizar sus configuraciones, actualizaciones, etc.
2. Usar GitLab desde GitLab.com con pago por el servicio, lo que otorga la colección completa de funcionalidades y herramientas de GitLab.
3. Usar GitLab sin coste para la publicación de repositorios libres de la misma forma que lo hace GitHub.

GitLab cuenta con su propia interfaz gráfica de usuario que puede también ser instalada en un servidor propio. GitLab se constituye de proyectos que contienen el código que se va a editar en archivos digitales, o sea, en los repositorios. Dentro de estos directorios se almacenan los contenidos y archivos que conforman un proyecto tecnológico, por ejemplo, archivos con extensiones propias de JavaScript, HTML, CSS, PHP, etc. (IONOS, 2022).

Una de las funciones más notables de GitLab es la ramificación. La creación de “ramas” permite a los usuarios bifurcar la parte principal del código de modo que puedan realizar cambios o modificaciones sin que la rama principal se vea afectada. Para comenzar a utilizarlo, primero es necesario que todos los integrantes involucrados en el proyecto descarguen una copia del proyecto central en su dispositivo, los cambios posteriores se hacen mediante *commits* (captura de los cambios en el proyecto en ese momento) y, una vez finalizado el proceso de edición de código, dichos cambios se agregan al repositorio (IONOS, 2022). Así, GitLab puede usarse de diferentes maneras: ya sea con las herramientas que se usan regularmente, mediante una terminal o un programa de interfaz, o desde el sitio de GitLab en la web, siempre que el dispositivo desde donde se accede al sistema esté conectado a Internet (Torrado, 2017).

Dentro de las funcionalidades de GitLab, se tiene que este sistema puede utilizarse para la gestión de proyectos (que refiere a, básicamente, los repositorios gestionados por GitLab), grupos (empresas, usuarios) y *snippets* (refiere a fragmentos de código que pueden ser utilizados para realizar cualquier cosa). Otras funcionalidades que ofrece este sistema son el listado de *commits*, donde se pueden conocer los últimos cambios hechos en el repositorio; las etiquetas (*Tags*) definen puntos del estado del código y restringir su acceso para que solo usuarios específicos puedan hacer modificaciones; los problemas (*issues*), permiten la detección de problemas en el software y monitoreo, también permite generar discusiones sobre los problemas, tomar control sobre el flujo de trabajo como medida para resolverlo, a través del establecimiento de qué personas tendrán la tarea de solucionarlo, el tiempo estimado para ello junto con el tiempo utilizado, la fecha límite, entre otros (Torrado, 2017).

Otras funcionalidades de GitLab que (inLab FIB, 2022) describe son las siguientes:

- Ofrece la opción de determinar distintos tipos de acceso y permisos basados en roles y grupos.
- Seguimiento de incidencias y comentarios dentro del proyecto.
- Facilidad de acceso al código de manera remota.
- Capacidad de importación de repositorios ya existentes.
- Posibilidad de generar copias de seguridad.

- Visualización del historial de modificaciones, entre otras.

1.3.1.6 Miro

Miro es un espacio de trabajo en la web, preocupado por proveer de una herramienta digital que facilite el trabajo colaborativo y su organización empleando recursos como videollamadas, presentaciones en línea, uso compartido, entre otros. Miro puede ser integrado sin dificultades con otras plataformas de trabajo o en la nube, entre algunos de estos se encuentran Dropbox, Google Suite, Microsoft Teams, Trello, Sketch, y más. Miro puede ser utilizado de forma gratuita, pero con limitaciones en el uso de algunos elementos o funciones. Sin embargo, también provee de planes de pago mensuales que amplían las herramientas, almacenamiento e incluso varias de las integraciones (Sean, 2023; Allende, 2021; AppMaster, 2023).

Esta plataforma consta de un tablero o pizarra digital, en el que se pueden crear variedad de esquemas, plantillas y diagramas en conjunto con otras personas. Para ello, Miro ofrece una amplia variedad de formas, íconos y plantillas además de la posibilidad de integración de elementos multimedia como imágenes o videos. También permite la adición de notas adhesivas dentro del tablero y otros elementos que facilitan la comunicación de ideas y su organización para el equipo (AppMaster, 2023).

Una de las mayores ventajas de esta plataforma es que el trabajo colaborativo dentro de una pizarra en Miro se lleva a cabo en tiempo real, de modo que todos los integrantes pueden compartir y editar su contenido eficientemente desde cualquier lugar. Por ello, Miro es una plataforma que favorece el trabajo remoto para equipos que no se encuentran laborando en un mismo lugar, pero que requieren de su colaboración (AppMaster, 2023).

El uso de Miro es en sí fácil e intuitivo, incluso para los usuarios que anteriormente no han trabajado con alguna herramienta digital como esta. Para comenzar a trabajar con Miro, es necesario crear una cuenta y registrarse ya sea para utilizarlo de forma gratuita o para comenzar a pagar un plan para acceder a más funciones. Una vez realizado este paso, ya es posible comenzar con la creación de tableros y, en conjunto con el equipo de trabajo, incluir todos los textos, imágenes, diagramas, y otros elementos visuales que sirvan en la planificación y gestión de proyectos o ideas (AppMaster, 2023).

1.3.1.7 Pixabay

Pixabay es una plataforma en la web que se preocupa por proveer imágenes de calidad, a personas que requieran integrar estos elementos multimedia a sitios web, para fines publicitarios o, sencillamente, para uso personal. En esta plataforma, los usuarios tienen acceso a una colección de más de 1,7 millones de fotos, imágenes, vectores e ilustraciones e, inclusive, videos de los que es posible elegir el contenido que mejor se acomode a sus necesidades (Liderlogo, s.f.).

Esta plataforma se caracteriza por su contenido que se encuentra bajo el Creative Commons CCO. Lo anterior significa que ese contenido puede ser utilizado para diferentes propósitos sin la necesidad de solicitar permisos por su uso, ni proporcionar atribuciones (Liderlogo, s.f.).

Aún con lo anterior, existen algunas excepciones que los usuarios deben tener en cuenta para su uso (Liderlogo, s.f.):

- Las imágenes en donde se encuentren personas identificables no deben utilizarse de modo que la persona se vea mal, o que su uso sea ofensivo sin contar con consentimiento.
- También, no se concede autorización de utilizar imágenes de personas o de organizaciones que endosen productos o servicios si no lo hacen.

1.3.2 Lenguajes de programación, modelado y consulta

Además de las herramientas tecnológicas, una parte significativa del proyecto consta del código escrito en diferentes lenguajes de programación, y con los que se pueden crear funciones que están destinadas a satisfacer los estrictos requerimientos que debe tener un sistema.

El Sistema de proceso de evaluación de perfil lingüístico contempla el uso de varios de ellos como JavaScript, para la generación de contenido dinámico; JSON, para el intercambio de información entre el sistema y el cliente; HTML, para la estructura básica de cada página web; PHP, para la programación del lado del servidor; y CSS para hacer del aspecto visual del sistema, uno más atractivo.

Además, se considera el empleo del lenguaje UML para el diseño del sistema a través de diagramas que llevan el mismo nombre; y SQL, con el que es posible generar la base de datos y administrarla.

1.3.2.1 JavaScript - JS

JavaScript (también conocido como JS) es un lenguaje de programación interpretado creado en el año 1995 inicialmente por NetScape Communications (hoy en día conocido como Mozilla Foundation), basado en el estándar ECMAScript y utilizado principalmente para la creación de páginas web interactivas. Algunos ejemplos comunes de ello son los carruseles de imágenes, menús desplegados, elementos que cambian de color, etc. Además, puede ser utilizado para el paradigma orientado a objetos, para programación procesual o funcional. En los días que corren, JavaScript está integrado en los motores de los navegadores web modernos (Luna, 2019; AWS Amazon, s.f.; IONOS, 2023).

Un lenguaje de programación interpretado, como JavaScript, es aquel en el que cada línea del programa es leída de manera descendente, traducida a lenguaje máquina y ejecutada, cuestión que lo diferencia de los lenguajes compilados en los que el código es traducido en su totalidad a lenguaje máquina, y genera un archivo que sea ejecutable por el sistema operativo (Fernández, 2015).

Una parte significativa de la sintaxis de este lenguaje de programación está basada en el lenguaje C, y también algunos nombres propios y convenciones del lenguaje de Java. Por añadidura, es necesario hacer mención que JavaScript y Java no tienen relación alguna (Luna, 2019).

Cuando se visita un sitio web, se tienen tres capas en las que, en cada una, se ejecuta una tecnología web que forma parte de este sitio. En esta jerarquía de capas, JavaScript ocupa el tercer puesto, siendo sus predecesores HTML y CSS en ese orden. El primer nivel, HTML es el encargado de dar estructura y propósito; luego, CSS se encarga de agregar estilos para que la vista sea agradable; y, finalmente, JavaScript incorpora el comportamiento dinámico de la página web (MDN contributors, 2023).

Cabe mencionar que JavaScript surgió como respuesta ante las páginas estáticas o de diseño fijo para hacer de ellas sitios dinámicos para los visitantes. Conforme este lenguaje evolucionaba, se fueron creando a su vez bibliotecas, marcos y prácticas de programación. Hoy en día, JavaScript ya no solo es utilizado para la programación del lado del cliente, sino que también del servidor (AWS Amazon, s.f.):

- **JavaScript del lado del servidor.** Con JavaScript del lado del servidor es posible acceder a una base de datos para realizar diferentes acciones sobre ella como ejecutar operaciones lógicas y utilizarlo como respuesta ante eventos desencadenados por el sistema operativo del servidor.
- **JavaScript del lado del cliente.** Por otro lado, del lado del cliente, JavaScript genera dinámicamente contenido nuevo a través de la lógica en la interfaz del usuario y modificando el contenido de la página web.

1.3.2.2 JavaScript Object Notation - JSON

El nombre de JSON corresponde a las siglas de *JavaScript Object Notation*. Fue creado por Douglas Crockford como un estándar para el envío y recepción de datos entre el navegador y el servidor que puede ser utilizado para el intercambio de información entre una aplicación, dos aplicaciones, o bien, dentro de una aplicación del lado del cliente, etc. (Aguirre, JSON - Vol.1: Primeros pasos - Sintaxis - Tipos de datos, 2020). JSON está basado en el estándar JavaScript ECMA-262 de la 3ra edición establecido en diciembre del año 1999 (Jackson, 2016).

JSON en sí es un lenguaje para definir estructuras de objetos diseñado para el intercambio de datos orientados a objetos JavaScript de manera ligera. Particularmente, para ser tan sencillo como pueda ser a fin de facilitar la lectura de estructuras de datos, y para escribirlas más rápido y con mayor eficiencia (Jackson, 2016). Un ejemplo de la sintaxis de JSON para escribir una estructura de datos se puede ver en la figura 2, en donde se describe a una persona con su nombre y su apellido. Para crear el objeto JSON, primero se define el nombre de la variable, en

este ejemplo será *persona*, y contendrá el arreglo donde se almacenarán las propiedades del objeto *persona*. Para escribir una propiedad, se escribe el nombre de la propiedad entre comillas (“), seguido por dos puntos (:) y el valor que tomará dicha propiedad escrito entre comillas (“). Para separar una propiedad de otra, se utiliza una coma (.). En el ejemplo, la propiedad “primerNombre” toma el valor de “Santiago”, y la propiedad “apellido” toma el valor de “Perez”.

```
Var persona =  
{“primerNombre”: “Santiago”, “apellido”: “Perez”}
```

FIGURA 2. SINTAXIS DE JSON.

Fuente: (Aguirre, JSON - Vol.1: Primeros pasos - Sintaxis - Tipos de datos, 2020)

Un objeto JSON puede almacenarse dentro de un archivo propio que posea la extensión característica de esta notación (*.json*) y una MIME type (*application/json*). Dentro de la estructura de JSON, los tipos de datos que es capaz de aceptar en sus objetos son cadenas de texto, valores numéricos, arreglos, valores booleanos y otros literales de objetos (MDN contributors, 2023).

JSON puede utilizarse de dos formas: la primera consiste en escribir las estructuras de datos con JSON desde el inicio, y la segunda donde se escriben los objetos con JavaScript para después efectuar el *parseo*, y transformar los objetos a sintaxis de JSON. Esta notación puede trabajar con una larga lista de lenguajes, entre los que se encuentran C, C++, Cobol, Fortran, Java, JavaScript, MatLab, Perl, PHP, Python, Ruby, entre muchos otros más y, afortunadamente, todos los lenguajes mencionados proporcionan ya sea soporte nativo o librerías externas para la manipulación, *parseo* y envío de datos con JSON (Aguirre, JSON - Vol.1: Primeros pasos - Sintaxis - Tipos de datos, 2020).

Adicionalmente, (Aguirre, JSON - Vol.1: Primeros pasos - Sintaxis - Tipos de datos, 2020) sugiere, como requerimiento previo antes del uso de JSON, comprender JavaScript y la forma de creación de objetos en ese lenguaje como mínimo.

1.3.2.3 HyperText Markup Language - HTML

Por sus siglas en inglés, *HyperText Markup Language*, HTML es un lenguaje de marcado de hipertexto creado por Tim Berners-Lee, que permite estructurar la información de páginas web mediante el uso de etiquetas. Las páginas creadas con este lenguaje no sólo son capaces de mostrar texto, sino también elementos

multimedia como videos, audio e imágenes. Es importante mencionar que existe distinción entre un lenguaje de programación y un lenguaje de marcado, pues este último no posee instrucciones de control de flujo, no utiliza variables, operadores y funciones, etc. De hecho, la palabra *hipertexto* refiere a los enlaces que conectan páginas web unas con otras ya sea dentro del mismo sitio web o en otros (Rubiales Gómez, 2021; MDN contributors, 2023).

HTML fue creado con el objetivo de compartir información de manera simple entre diferentes universidades. La primera versión de este lenguaje se publicó en el año 1991 con el nombre “HTML Tags”. HTML pasó por varias versiones hasta llegar a la actual versión 5 (Tokio School, 2020). Las novedades que esta versión ofrece son (EBAC, 2023):

- Es gratuito, lo que indica que no es necesario un software de pago para programar páginas web.
- El lenguaje se presenta de forma clara, ordenada y simple de aprender.
- Compatibilidad con los navegadores web más recorridos por los usuarios.
- Capacidad de soportar elementos multimedia y funcionalidades interactivas.
- El uso de APIs permiten la inclusión de dibujos 3D o mapas de forma directa en el documento.
- Permite la formulación de consultas SQL, entre otras.

Un documento HTML, en su forma más básica está estructurado en dos partes: la cabecera, que se representa con la etiqueta `<head>` y que se ocupa de recoger la información referente al documento HTML, sin embargo, esta información no es visible para los visitantes a la página web; y el cuerpo, que se representa con la etiqueta `<body>` y es el contenedor de todo el contenido que será presentado en la página web (Rubiales Gómez, 2021). Un ejemplo de ello puede apreciarse en la figura 3. En esta figura, se agrega la etiqueta `<html>` para representar la raíz del documento HTML y tendrá en su interior todos los elementos que conforman la página web; la etiqueta `<head>`, para añadir un encabezado a la página web, y posee en su interior la etiqueta `<title>` para añadir un título con el texto “Ejemplo”; la etiqueta `<body>`, que representará todo el contenido de la página web, y posee una etiqueta `<h2>` en su interior con el texto “Hola a todos”. Igualmente, cada etiqueta posee su etiqueta de cierre.

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    <h2>Hola a todos</h2>
  </body>
</html>
```

FIGURA 3. ESTRUCTURA BÁSICA DE UN DOCUMENTO HTML.

Fuente: (Rubiales Gómez, 2021)

Además de la cabecera y el cuerpo, un documento HTML está conformado por otros elementos más, como son:

- **Etiqueta de apertura.** Se trata del nombre del elemento, y se encuentra encerrado por paréntesis angulares (<>) que indica en dónde iniciará el elemento o desde dónde comenzará a tener efecto.
- **Etiqueta de cierre.** Similar a la etiqueta de apertura, señala el nombre del elemento, pero con la diferencia de que la etiqueta de cierre hace la acción contraria, pues señala el final del elemento o dónde finaliza el efecto. Además, utiliza también los paréntesis angulares para encerrar el nombre del elemento, pero incluye también una barra de cierre (/).
- **Contenido.** Indica el contenido del elemento, que bien puede ser solo texto.
- **Elemento.** Es la integración de la etiqueta de apertura, la etiqueta de cierre y el contenido. Los elementos pueden tener atributos, los cuales contienen información adicional sobre ese elemento

De acuerdo a (Celaya Luna, 2019), HTML en su versión 5, la cual es la más reciente, no se limita únicamente a ser un lenguaje de etiquetas HTML, sino que también combina las nuevas etiquetas de HTML sumando propiedades CSS3, JavaScript y más tecnologías relacionadas al desarrollo de sitios web. La primera versión de este lenguaje surgió en 1991 y en ella se describían 18 elementos básicos que conformarían la estructura de un documento HTML. Hoy en día, solo existen 13 de esos 18 elementos iniciales (Rubiales Gómez, 2021).

1.3.2.4 PHP: Hypertext Preprocessor - PHP

El nombre de este lenguaje es un acrónimo recursivo para *PHP: Hypertext Preprocessor*, de hecho, anteriormente significaba *Personal Home Page*. PHP es un lenguaje del tipo interpretado utilizado principalmente para el desarrollo de aplicaciones, pero obrando del lado del servidor (Arias, 2013).

Este lenguaje puede ser utilizado en multitud de sistemas operativos, entre los que se encuentran Linux, una abundante cantidad de variantes de Unix, Microsoft Windows, mac OS, RISC OS, etc. PHP también soporta una amplia variedad de servidores web modernos, como Apache e IIS por mencionar algunos. Sumado a ello, PHP tiene también el soporte necesario para la comunicación con otros servicios, tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, entre otros. Otra característica de PHP es que puede utilizar programación orientada a objetos, programación por procedimientos o, inclusive, una combinación de ambos (ThePHPGroup, s.f.).

PHP es un lenguaje de alto nivel que es muy común encontrarlo incrustado en documentos HTML. Este lenguaje tiene características muy buenas que lo han posicionado como uno de los más utilizados hoy en día. Algunas de estas características son (Peña Millahual, 2019):

- **Sintaxis sencilla.** Para quienes ya poseen conocimientos de algún lenguaje de programación, PHP no les resultará complicado. Por otra parte, los principiantes pueden comprender PHP fácilmente, pues la curva de aprendizaje de este lenguaje no es muy pronunciada.
- **Soporte para bases de datos.** PHP ofrece un amplio soporte para trabajar con una buena cantidad de bases de datos, entre las que se pueden encontrar las más utilizadas por los desarrolladores como son MySQL, Oracle, PostgreSQL, entre otras.
- **Uso de bibliotecas externas.** Esta característica le proporciona ventaja a este lenguaje. Por su buena versatilidad, permite realizar diversas tareas como generación de documentos en formato PDF, envío de mensajes o correos electrónicos, análisis de código XML, etc.
- **Soporte.** Es posible encontrar ayuda sin complicaciones relacionado a este lenguaje derivado de la extensa cantidad de desarrolladores que lo utilizan globalmente. Además, por la misma razón, es soportado por la mayoría de navegadores web.
- **Código abierto.** Por el hecho de ser un lenguaje de código abierto, PHP puede ser modificado en su código fuente para así obtener beneficios de los avances aportados por la comunidad para el desarrollo del proyecto, y al

mismo tiempo se hacen aportaciones, correcciones y mejoras para que otros puedan hacer uso de ellas.

- **Multiplataforma.** Por esta característica, trabajar con PHP en varios sistemas operativos sin depender de uno en particular, no representa ningún problema.

Como ya se dijo, PHP es utilizado en su mayoría de veces para programar del lado del servidor. Lo que lo diferencia ante un lenguaje como JavaScript, que trabaja del lado del cliente, es que el código será ejecutado en el servidor y el cliente recibirá el resultado de esa ejecución sin conocer cuál fue el código que generó esa respuesta (ThePHPGroup, s.f.).

A diferencia de otros lenguajes donde se utilizan muchos comandos para mostrar HTML, las páginas creadas con PHP contienen HTML directo con código embebido. El código de PHP se puede reconocer a simple vista, este se caracteriza por iniciar con la etiqueta `<?php` y concluir con la etiqueta de finalización `?>` (ThePHPGroup, s.f.).

1.3.2.5 Cascading Style Sheets - CSS

Traducido de sus siglas en inglés *Cascading Style Sheets*, Hojas de Estilo en Cascada (CSS), es un lenguaje para aplicar estilo a la presentación de documentos HTML o XML. CSS describe cómo se va a renderizar un elemento en la pantalla, papel, u otros medios. CSS fue desarrollado por el World Wide Web Consortium (W3C) en el año 1996 como solución a la problemática que HTML planteaba por no ser diseñado para dar formato a las páginas, ya que solo cumple con la función de escribir el marcado para el sitio web (MDN contributors, 2023; B., Hostinger Tutoriales, 2022).

En un principio se añadieron etiquetas en la versión 3.2 para el formato como `` en HTML, pero lejos de solucionarlo, causó más problemas a los desarrolladores, pues el diseño de un sitio web contempla fuentes de letra, colores, etc. y la reescritura o modificación de todo ese código se convirtió en un proceso largo y costoso (B., Hostinger Tutoriales, 2022).

La sintaxis de CSS se basa del inglés con un conjunto de reglas que la conducen. Su estructura es muy simple y está conformada por (B., Hostinger Tutoriales, 2022):

- El selector, que apunta al elemento o elementos HTML a los que les serán aplicados el estilo.

- El bloque de declaración, que contiene todas las propiedades y sus valores que se van a aplicar sobre el elemento o elementos. El bloque de declaración está encerrado con llaves, y las propiedades van separadas una de otra por un punto y coma.

Esas propiedades indican el atributo específico a aplicar, y su valor va separado de la propiedad por dos puntos. En la figura 4 se muestra un ejemplo de aplicación de CSS que corresponde al tipo interno o externo (que se explicará más adelante). En este ejemplo, la etiqueta `<style>` es añadida al documento HTML, y en su interior se especifican los elementos a los que se les aplicarán estilos escribiendo el nombre del elemento sin los paréntesis angulares (`<>`), seguido de llaves (`{}`). Dentro de estas llaves se detallan cada una de las propiedades iniciando por el selector, seguido de dos puntos (`:`) y el bloque de declaración, y son separadas unas de otras por un punto y coma (`;`). En el ejemplo se utiliza el selector *color* para cambiar el color de la fuente, y el bloque de declaración *blue* (azul); y el selector *text-weight* para especificar el grosor de la letra, con el bloque de declaración *bold* (negrita). Por lo tanto, el ejemplo expresado en la figura 4 muestra el código necesario para cambiar el color de fuente del párrafo a azul y lo pone en negrita (B., Hostinger Tutoriales, 2022).

```
<style>
p {
  color: blue;
  text-weight: bold;
}
</style>
```

FIGURA 4. SINTAXIS DE CSS.

Fuente: (B., Hostinger Tutoriales, 2022)

Sin embargo, esta estructura solo aplica para cuando se utiliza CSS de manera interna o externa. En total, existen tres formas de aplicar estilos a un documento HTML con CSS (V., 2023):

1. **CSS Interno.** Este tipo de aplicación de CSS requiere que se agregue al documento la etiqueta `<style>`, dentro de la cual se encontrará el código propio de CSS. El CSS interno sirve para dar estilo a una sola página HTML, para que los estilos sean visibles en todas las páginas que conforman al sitio, es necesario que ese fragmento de código CSS esté presente en todos los documentos de todas las páginas.
2. **CSS Externo.** Para aplicar estilos CSS de forma externa se requiere un archivo con extensión `.css` que sirva de contenedor para los selectores y sus

propiedades. Para que los estilos sean visibles en la página web, es necesario enlazar el archivo .css con el documento HTML con la etiqueta `<link>` que se encuentra habitualmente en la sección del encabezado marcado con la etiqueta `<head>`.

3. **CSS Inline.** El tipo Inline se utiliza cuando se le quiere aplicar estilo a un elemento del documento HTML en particular. Para ello, dentro de la etiqueta del elemento se añade el atributo `style` seguido por un signo “=” y por el valor de la propiedad a asignar. El CSS tipo Inline no utiliza la estructura de los selectores con los bloques de declaración.

1.3.2.6 Unified Modeling Language - UML

Por sus siglas en inglés, *Unified Modeling Language*, el Lenguaje de Modelado Unificado es un estándar para representar visualmente los objetos, los estados y los procesos de un sistema mediante diferentes tipos de diagramas. UML sirve tanto para modelar un proyecto y establecer la arquitectura de información estructurada, como también sirve de herramienta para mostrar la descripción del sistema de forma comprensible. UML es muy utilizado dentro del paradigma orientado a objetos para el desarrollo de software. En este paradigma, las piezas centrales para el desarrollo de sistemas son las clases y los objetos: un objeto es la representación de un objeto existente o abstracto del espacio del problema, una clase es la descripción de una colección de objetos similares (IONOS, 2018; Lizcano Bueno, s.f.).

El paradigma orientado a objetos, dentro del desarrollo de software, provee la base fundamental que servirá para la construcción del sistema partiendo de sus componentes y con el uso de tecnologías como plataformas de programación. Así, la visualización, especificación, construcción y documentación de sistemas orientados a objetos es, prácticamente, el propósito de UML. UML fue creado en los 90's por los ingenieros de software Grady Booch, Ivar Jacobson y James Rumbaugh. Su objetivo era desarrollar una forma más simple de representar los complejos desarrollos de software, separando al mismo tiempo la metodología del proceso (Lizcano Bueno, s.f.; Microsoft 365 Team, 2019).

Los diagramas de UML utilizan los siguientes elementos para representar todos y cada uno de los componentes del sistema (IONOS, 2018):

- Objetos individuales, que son elementos básicos.
- Clases, que combinan los elementos con propiedades.
- Relaciones entre objetos, a través de jerarquías, sus comportamientos y su comunicación con otros objetos.

- Actividad, o sea, la combinación compleja de acciones o módulos de comportamiento.
- Interacciones entre los objetos y las interfaces.

UML posee muchas variaciones de diagramas para representar múltiples escenarios utilizados por diferentes tipos de personas. Todos estos diagramas se encuentran dentro de una de las dos clasificaciones de diagramas que UML ofrece: los de estructura y los de comportamiento. Los diagramas de estructura tienen la tarea de representar la estructura estática de un software o sistema y mostrar los niveles de abstracción y de implementación. Se muestran como una jerarquía todos los componentes o módulos que conforman al sistema o software y cómo estos se interrelacionan e interactúan entre sí. Por otro lado, los diagramas de comportamiento muestran la funcionalidad del sistema, centrándose en qué debe ocurrir en el sistema que se está modelando. Dentro de la clasificación de diagramas estructurales se encuentran los siguientes (Microsoft 365 Team, 2019):

Diagrama de clases. Este diagrama es probablemente el más común dentro de todos los diagramas UML. El diagrama de clases se utiliza para la representación del diseño lógico y físico del sistema, mostrando sus diferentes clases y cómo estas se relacionan entre sí. Su aspecto es similar al de un diagrama de flujo debido a que las clases son representadas con cuadros. Cada clase posee tres secciones: la sección superior, que contiene el nombre de la clase; la sección central, que contiene los atributos de la clase; y la sección inferior, que contiene todos los métodos u operaciones de clase (Microsoft 365 Team, 2019). En la ilustración 6 se muestra un ejemplo de diagrama de clases.

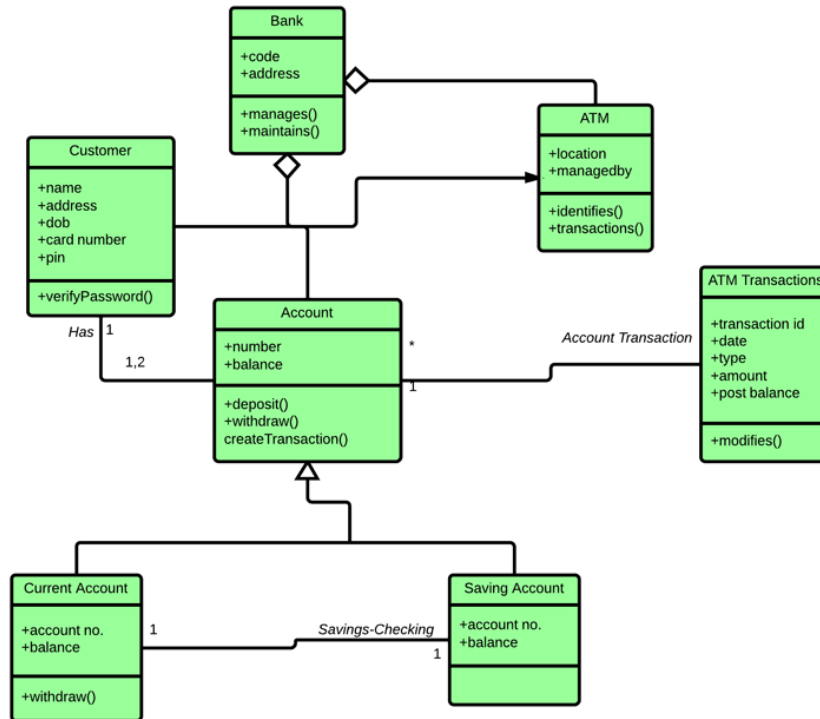


ILUSTRACIÓN 6. DIAGRAMA DE CLASES

Fuente: (Lucidchart, 2023a)

Todas las clases dentro de este diagrama tienen cierto nivel de acceso, estos niveles pueden verse con ayuda de los modificadores de acceso, los cuales son: público (+), privado (-), protegido (#), paquete (~), derivado (/) y estático (subrayado). Entre los componentes adicionales del diagrama de clases están (Lucidchart, 2023a):

- **Señales.** Son símbolos que refieren a las comunicaciones unidireccionales asincrónicas entre los objetos activos.
- **Tipos de datos.** Son clasificadores que establecen los valores de los datos.
- **Paquetes.** Son figuras diseñadas para organizar clasificadores relacionados en un diagrama y son representados por un rectángulo con pestañas.
- **Interfaces.** Las interfaces son recopilaciones de firmas de operaciones o definiciones de atributos que describen un conjunto uniforme de comportamientos.
- **Enumeraciones.** Son representaciones de tipos de datos definidos por el usuario.
- **Objetos.** Instancias de una clase o varias clases.

- **Artefactos.** Son elementos modelo que representan a las entidades de un sistema en concreto como son documentos, bases de datos, archivos ejecutables, entre otros.

Diagrama de objetos. Este diagrama es utilizado muy a menudo como forma de comprobar la revisión de un diagrama de clases a fin de obtener precisión. El diagrama de objetos muestra los objetos del sistema y sus relaciones, y visualiza los posibles defectos en el diseño que necesitarán ser reparados. En la ilustración 7 se muestra un ejemplo de diagrama de objetos (Microsoft 365 Team, 2019).

Diagrama de componentes. El diagrama de componentes, también llamado *Diagrama de flujo de componentes*, muestra las agrupaciones lógicas de elementos y sus relaciones. O sea, permite visualizar de forma más simple un sistema complejo al separarlo en componentes más pequeños. En la ilustración 8 se muestra un ejemplo de diagrama de componentes (Microsoft 365 Team, 2019).

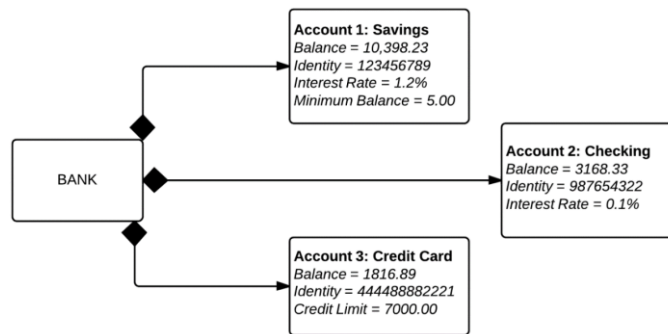


ILUSTRACIÓN 7. DIAGRAMA DE OBJETOS

Fuente: (Lucidchart, 2023b)

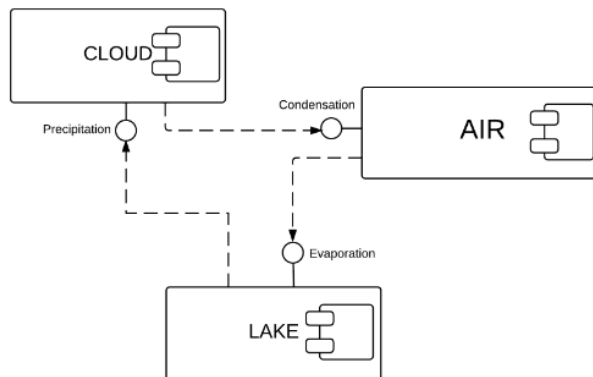


ILUSTRACIÓN 8. DIAGRAMA DE COMPONENTES

Fuente: (Lucidchart, 2023c)

Diagrama de estructura compuesta. Este diagrama es rara vez utilizado por las personas externas al campo de desarrollo del software. El diagrama de estructura compuesta es parecido al diagrama de clases, pero con la diferencia de que el de estructura compuesta utiliza un enfoque más profundo, pues describe la estructura interna de varias clases junto con sus relaciones entre sí (Microsoft 365 Team, 2019).

Diagrama de despliegue. El diagrama de despliegue muestra como nodos a los componentes de hardware y de software que conforman al sistema, y describe también sus relaciones. Además, provee una representación visual de dónde se implementa cada uno de estos componentes. En la ilustración 9 se muestra un ejemplo de un diagrama de despliegue. Los elementos que conforman este diagrama son los siguientes (Microsoft 365 Team, 2019; Lucidchart, 2023d):

- **Artefactos.** Refiere a un producto de software ya desarrollado, se representa con un rectángulo con su nombre y la palabra “artefacto” encerrado por flechas dobles.
- **Asociación.** Es una línea que indica un mensaje o algún otro método de comunicación entre los nodos.
- **Componente.** Es un rectángulo con dos pestañas que representan a un elemento del software.
- **Dependencia.** Es una línea discontinua que termina en flecha que expresa que un nodo es dependiente de otro.
- **Interfaz.** Se representa con un círculo que expresa una relación contractual.
- **Nodo.** Los nodos son objetos de software o hardware, y son representados por un cuadro tridimensional.
- **Nodo como contenedor.** Refiere a un nodo que tiene a otro nodo en su interior.
- **Estereotipo.** Refiere un dispositivo que se encuentra dentro del nodo, y es representado en la parte superior del nodo con su nombre entre flechas dobles en forma de corchetes.

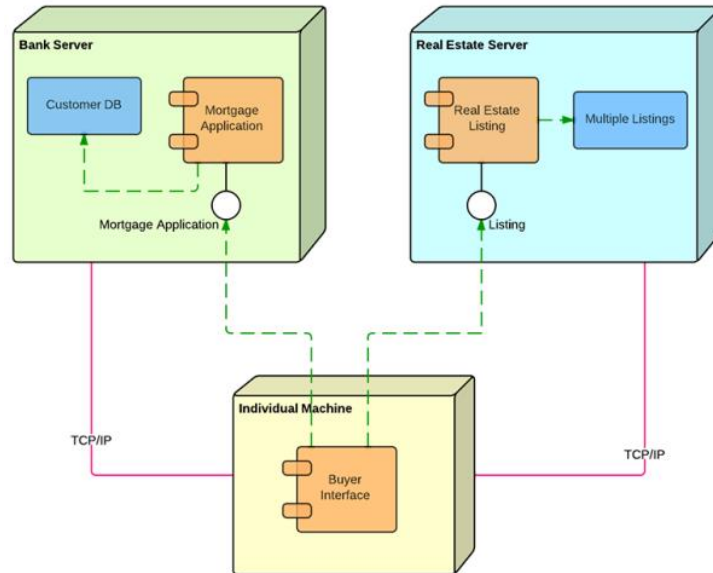


ILUSTRACIÓN 9. DIAGRAMA DE DESPLIEGUE

Fuente: (Lucidchart, 2023d)

Diagrama de paquetes. Estos diagramas son utilizados para la representación de dependencias entre los paquetes que conforman un modelo, y su objetivo es mostrar la relación existente entre los componentes grandes de un sistema complejo (Microsoft 365 Team, 2019).

Diagrama de perfiles. Este diagrama tiene mayor parecido con un lenguaje que con un diagrama. El diagrama de perfiles sirve para crear nuevas propiedades y semántica para los diagramas UML, ya que establece estereotipos personalizados, marcados y restricciones (Microsoft 365 Team, 2019).

Por otra parte, los diagramas UML de comportamiento son:

Diagrama de actividades. Los diagramas de actividades, como el que se presenta en la ilustración 10, se ocupan de representar un proceso con todos sus pasos. Más detalladamente, este diagrama describe los pasos que deben ser efectuados antes de pasar al siguiente para lograr un objetivo. Los elementos que este diagrama utiliza para su elaboración son (Microsoft 365 Team, 2019; Lucidchart, 2023e):

- **Símbolo de inicio.** Identifica el inicio de un proceso o flujo de trabajo.
- **Símbolo de actividad.** Apunta a las actividades que deben ser realizadas dentro del proceso de modelado.
- **Símbolo conector.** Indican la dirección a la que se dirige el flujo de control de actividades.

- **Símbolo de unión o barra de sincronización.** Realiza la combinación de dos actividades simultáneas, para después reincorporarlas en un solo flujo de actividades donde cada actividad se ejecuta una a la vez.
- **Símbolo de bifurcación.** El flujo es dividido en dos actividades que ocurren de manera simultánea.
- **Símbolo de decisión.** Indica una decisión y, habitualmente, suelen presentarse sólo dos caminos opcionales que el usuario puede tomar.
- **Símbolo de nota.** Con este símbolo, todos aquellos colaboradores del diagrama pueden comunicarse con mensajes fuera del diagrama.
- **Símbolo de enviar señal.** Símbolo que apunta a que una señal está siendo enviada a una actividad que la recibirá.
- **Símbolo de recibir señal.** Símbolo que denota que un evento ha sido aceptado.
- **Símbolo de pseudoestado de historia superficial.** Símbolo que describe una transición que invoca el último estado activo.
- **Símbolo de bucle de opción.** Con este símbolo, los creadores pueden modelar una secuencia cíclica dentro del símbolo de bucle de opción.
- **Símbolo de final de flujo.** Indica el final de un proceso en particular.
- **Texto de condición.** Ayuda a identificar las condiciones bajo las cuales una actividad es bifurcada en esa dirección.
- **Símbolo de finalización.** Símbolo utilizado para indicar el estado finalizado de una actividad, así como también para representar el fin de todos los flujos de procesos.

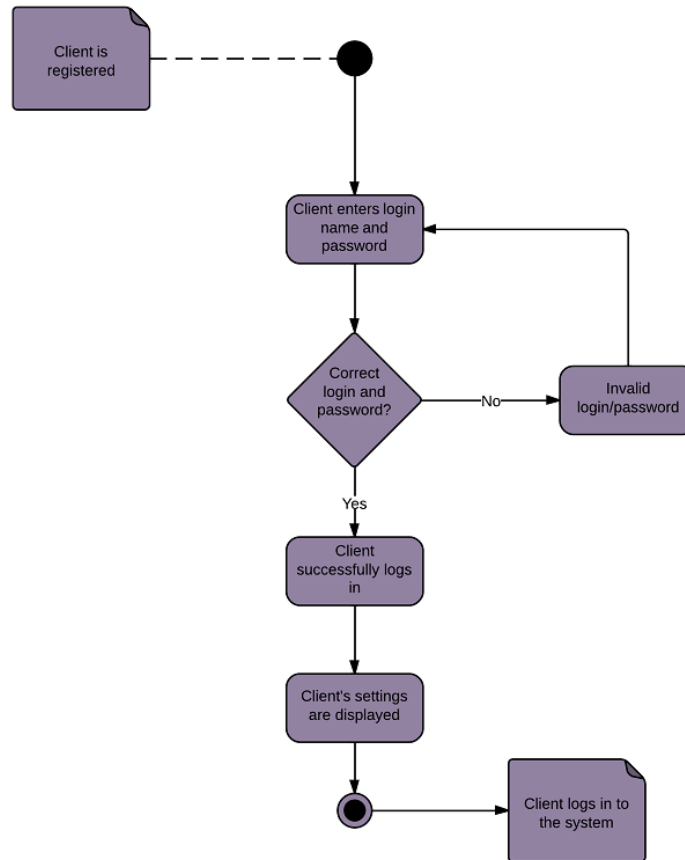


ILUSTRACIÓN 10. DIAGRAMA DE ACTIVIDADES

Fuente: (Lucidchart, 2023e)

Diagrama de casos de uso. Los diagramas de caso de uso describen las acciones que hace un sistema, no obstante, no describe la forma en que las hace. En este tipo de diagrama, un caso de uso se define como el conjunto de eventos que suceden cuando un “actor” utiliza el sistema para efectuar una acción y así completar un proceso. Los actores refieren a cualquier persona u objeto que interactúa con el sistema desde el exterior del sistema. Básicamente, este diagrama sirve para la representación de los requerimientos funcionales del sistema. Los elementos utilizados para la elaboración de un diagrama de caso de uso son (Microsoft 365 Team, 2019; Lucidchart, 2023f):

- **Casos de uso.** Los casos de uso son representados con óvalos en posición horizontal y representan los diferentes usos que podría tener.
- **Actores.** Son representados con figuras de palo que representan a personas u objetos que utilizan los casos de uso.
- **Asociaciones.** Se representan con una línea que une a los actores con los casos de uso.

- **Cajas delimitadoras del sistema.** Este símbolo es una caja que delimita al sistema. Si existiesen casos de uso fuera de esta caja, entonces se consideran como inalcanzables para el sistema.
- **Paquetes.** Es un elemento propio de UML que permite agrupar diferentes elementos, y son representados con el símbolo de un folder.

En la ilustración 11 puede observarse un ejemplo de diagrama de casos de uso.

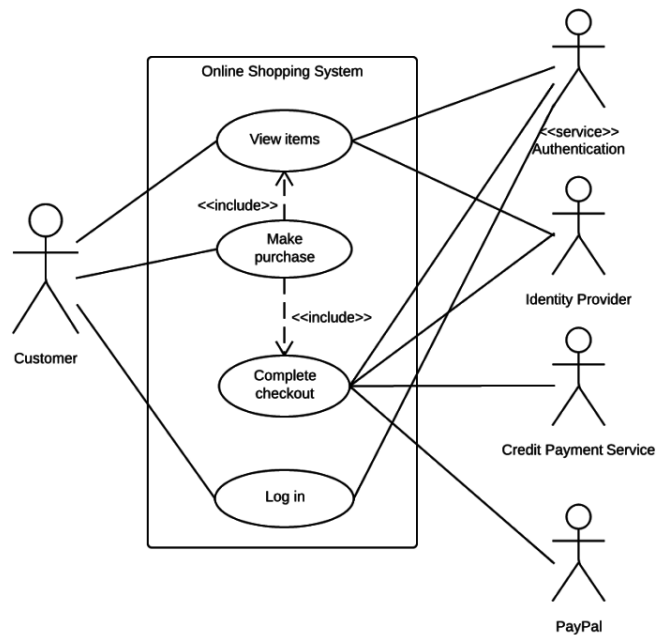


ILUSTRACIÓN 11. DIAGRAMA DE CASOS DE USO

Fuente: (Lucidchart, 2023f)

Diagrama de descripción general de interacción. Es parecido al diagrama de actividades ya que ambos muestran la secuencia de las actividades paso a paso, pero con las distinciones de que el diagrama de descripción general de interacción es más complicado y este se compone de diferentes diagramas de interacción. Utilizan los mismos componentes de un diagrama de actividad como son el nodo inicial, final, decisión, etc., pero añade algunos más como la interacción, el uso de la interacción, restricciones de tiempo y restricción de la duración (Microsoft 365 Team, 2019).

Diagrama de tiempo. El diagrama de tiempo se utiliza en los casos donde el tiempo es una pieza central. Este diagrama, también llamado *diagrama de secuencia o eventos*, no muestra las interacciones entre los objetos, sino que permite visualizar cómo los objetos y los actores se desempeñan dentro de una línea de tiempo. El enfoque central en este diagrama es la duración de los eventos y los cambios producidos de acuerdo a las restricciones de duración establecidas. El diagrama de

tiempo se constituye por los siguientes elementos (Microsoft 365 Team, 2019; Lucidchart, 2023g):

- Línea de vida, que representa un participante de manera individual.
- Línea de tiempo de estado, que refiere a los estados diferentes en los que puede estar la línea de vida dentro de una canalización.
- Restricción de duración, que refiere al tiempo necesario para concluir una restricción.
- Restricción de tiempo, refiere al periodo de tiempo que el participante tiene para concluir una acción.
- Destrucción, que sucede cuando finaliza la línea de vida de un objeto y, una vez destruida, no se produce otra ocurrencia.

En la ilustración 12 puede observarse un ejemplo de diagrama de tiempo.

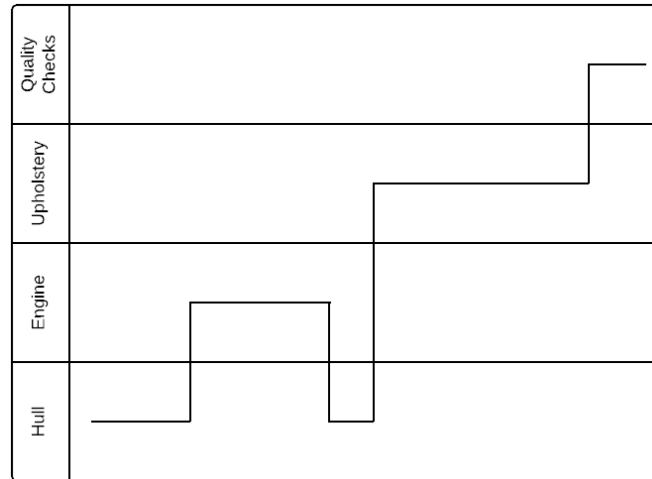


ILUSTRACIÓN 12. DIAGRAMA DE TIEMPO

Fuente: (Lucidchart, 2023g)

Diagrama de máquina de estados. Este diagrama es aplicado cuando el comportamiento de un objeto es complejo y el detalle es necesario. El diagrama de máquina de estados, también conocido como *gráfico de estados*, describe el comportamiento de un objeto y sus cambios de acuerdo a los eventos tanto internos como externos (Microsoft 365 Team, 2019).

Diagrama de secuencia. Su tarea es la de mostrar la estructura del sistema a través de la secuencia de mensajes e interacciones entre los actores y los objetos de forma cronológica, mostrando a su vez interacciones y ramificaciones simples. Los diagramas de secuencia poseen dos dimensiones: una vertical y una horizontal. En el caso de las líneas verticales, se muestran las secuencias de mensajes y llamadas ordenadas cronológicamente. Por el lado de las líneas horizontales, se

describen las instancias de objetos en las que los mensajes son transmitidos. Los pasos para la creación de un diagrama de secuencia son los siguientes (Microsoft 365 Team, 2019; Lucidchart, 2023h):

1. El primer paso consiste en escribir el nombre de la instancia de la clase dentro de una figura rectangular.
2. Después se dibujan las líneas entre las instancias de las clases a fin de expresar quién será el emisor del mensaje y quién será el receptor de ese mensaje.
3. Finalmente, se hace uso de flechas oscuras para simbolizar mensajes sincrónicos, para los mensajes asíncrónicos se ocupan puntas de flecha abiertas, y para los mensajes de respuesta se utilizan líneas discontinuas.

En la ilustración 13 se puede apreciar un ejemplo de un diagrama de secuencia.

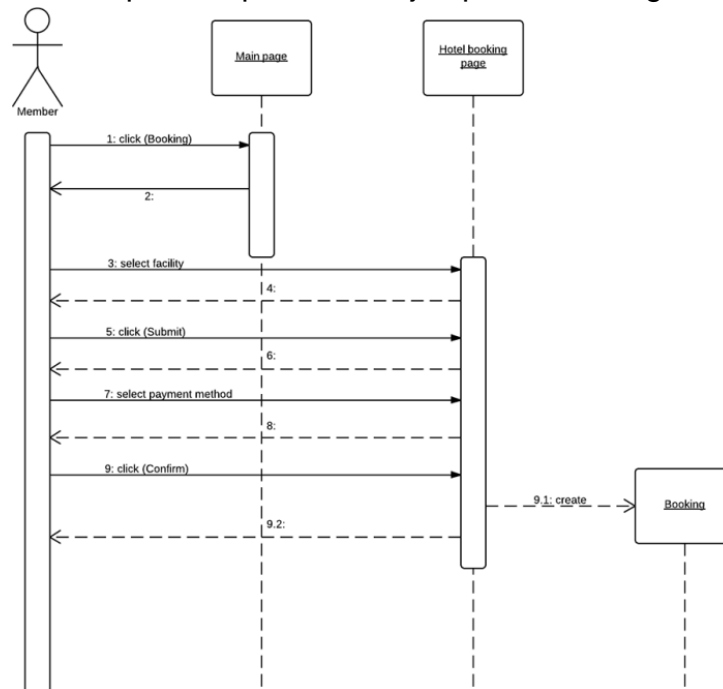


ILUSTRACIÓN 13. DIAGRAMA DE SECUENCIA

Fuente: (Lucidchart, 2023h)

Diagrama de comunicación. Es parecido al diagrama de secuencia, pero con la diferencia de que el diagrama de comunicación centra su atención en la comunicación entre objetos, por lo que muestra la organización de aquellos objetos que están presentes en una interacción y muestra, además, sus iteraciones y ramificaciones más complejas. Los elementos que se utilizan para crear diagramas de comunicación son (Microsoft 365 Team, 2019; Lucidchart, 2023i):

- **Rectángulos.** Estas figuras se utilizan para representar a los objetos que conforman al software.
- **Líneas.** Las líneas dibujadas entre las instancias apuntan a las relaciones entre diferentes partes del software.
- **Flechas.** Estos símbolos se ocupan para representar los mensajes que son enviados entre los objetos.
- **Numeración.** La numeración permite conocer el orden en que los mensajes son enviados, y cuantos de éstos son necesarios para concluir un proceso.

En la ilustración 14 se muestra un ejemplo de un diagrama de comunicación.

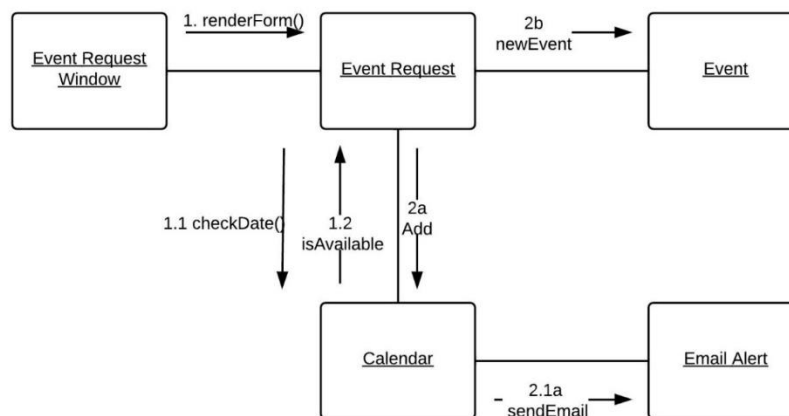


ILUSTRACIÓN 14. DIAGRAMA DE COMUNICACIÓN

Fuente: (Lucidchart, 2023i)

Además de las actividades ya mencionadas que este lenguaje puede hacer, UML también ha servido como indicación para modelar bases de datos. Aunque es cierto que UML no tiene especificaciones para modelarlas, si puede ser de ayuda para la creación de diagramas, más aún por que las bases de datos también son utilizadas en la programación orientada a objetos (Microsoft 365 Team, 2019).

1.3.2.7 Structured Query Language - SQL

Structured Query Language, por sus siglas, *SQL*, y del inglés, Lenguaje de Consulta Estructurada, es un lenguaje declarativo de alto nivel destinado al trabajo con agrupaciones de datos y sus relaciones entre sí. Comparado con otros lenguajes, la sintaxis de SQL no es complicada, aún para una persona sin mucha experiencia y, al igual que otros lenguajes computacionales, SQL es un estándar internacional reconocido por organismos de estandarización como ISO y ANSI (Microsoft, 2022).

Este lenguaje de consulta permite utilizar sentencias SQL para manipular la información almacenada en la base de datos, como actualización, eliminación y búsqueda de datos, y recuperación de información. De hecho, SQL solo trabaja con bases de datos relacionales. Esta clase de base de datos almacena la información en tablas con columnas que representan los atributos de los datos, y filas que representan datos con esos atributos. No obstante, SQL no es solo una herramienta exclusiva de consultas, sino que también puede controlar las funciones de un sistema gestor de bases de datos, o sea, el sistema que permite gestionar y extraer la información almacenada en la base de datos. Estas funciones incluyen (AWS Amazon, s.f.; Marqués et al., 2001):

- **Definición de datos.** Con SQL es posible definir la estructura y la organización de los datos almacenados y sus relaciones.
- **Obtención de datos.** Una consulta SQL devuelve datos de la base de datos para que el usuario pueda utilizarlos.
- **Control de acceso.** SQL puede distinguir a qué usuarios se les puede restringir el acceso a determinada información para evitar alteraciones en los datos, ya sea por añadir información, eliminarla o modificarla.

EL lenguaje de SQL, en términos de sus sentencias, puede ser dividido en tres tipos (Bisson, 2018):

1. **DDL (Data Definition Language).** Los comandos correspondientes al Lenguaje de Definición de Datos son utilizados para crear, modificar o eliminar estructuras de la base de datos. Estos comandos son CREATE, ALTER y DROP.
2. **DML (Data Manipulation Language).** Los comandos correspondientes al Lenguaje de Manipulación de Datos son utilizados para manipular la información almacenada en la base de datos. Estos comandos son SELECT, INSERT, DELETE y UPDATE.
3. **DCL (Data Control Language).** Los comandos correspondientes al Lenguaje de Control de Datos se utilizan para administrar la seguridad en términos del acceso a la información. Estos comandos son GRANT y REVOKE.

1.3.3 Frameworks y librerías

Además de las herramientas tecnológicas ya mencionadas, el empleo de frameworks en el desarrollo de software proporciona ahorro de tiempo derivado de que estos recursos contienen ya estructuras iniciales en su forma más básica, y que pueden ser utilizados para la resolución de diferentes problemáticas. Es por ello que se plantea la utilización de Laravel, un framework para el desarrollo en PHP de aplicaciones web, que facilite y acelere la tarea de programación a través de las funcionalidades que ya proporciona.

Igualmente, las librerías o bibliotecas, en programación, favorecen la reutilización de código que soluciona problemas previamente solucionados. La librería jQuery de JavaScript, que permite a los programadores manipular el DOM, controlar eventos, uso de AJAX, entre otras, le proporcionará al sistema gestionar esos eventos de una forma más simplificada.

1.3.3.1 Laravel

Laravel es definido como un *framework*, o marco de trabajo, de PHP que permite el desarrollo de aplicaciones web escalables de calidad con una menor cantidad de código, y utilizando una sintaxis que puede ser descrita como elegante. Laravel posee un repositorio en GitHub que permite el acceso a su código, por ello, este *framework* entra en la categoría de tecnología de código abierto (Laravel, 2023; Aguirre, FRAMEWORK TOTAL - Vol.1: Crea APPs desde Cero con Laravel + Bootstrap + MySQL, 2021; desarrolloweb, 2023). El logotipo de esta aplicación puede visualizarse en la ilustración 15.

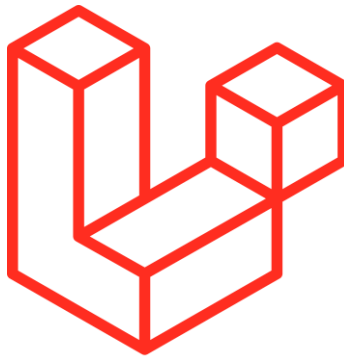


ILUSTRACIÓN 15. LOGO DE LARAVEL

La arquitectura de Laravel, la cual es MVC (Modelo-Vista-Controlador), consiste de la estructuración avanzada de carpetas que favorece a la separación de archivos para que estos se encuentren ordenados y definidos. También, su arquitectura de clases permite la separación del código de acuerdo a responsabilidades. Laravel proporciona a los desarrolladores multitud de características utilizadas habitualmente en la construcción de aplicaciones web. Algunas de las más distintivas son (desarrolloweb, 2023):

- Utiliza un sistema de rutas con el que es más sencillo crear, mantener y administrar todo tipo de URLs que resulten amigables para los usuarios, rutas de APIs, etc.
- Posee un sistema de abstracción de base de datos con un ORM con el que es posible tratar la información extraída de la base de datos como un objeto.
- Permite acceder a los datos en tiempo real y conocer cuándo estos son alterados en la base de datos.
- Provee de un sistema para creación de colas de trabajo para así, enviar tareas a ejecutarse en el background y elevar el rendimiento de las aplicaciones.

Como se ha mencionado, Laravel se preocupa por proveer a los desarrolladores todas las herramientas necesarias para la construcción de aplicaciones web, incluyendo su interfaz. Laravel proporciona un par de opciones para el desarrollo de interfaces: utilizando PHP o usando marcos de JavaScript como Vue o React. Para el desarrollo del presente proyecto computacional, compete hablar sobre la creación de interfaces haciendo uso de PHP (Laravel, 2023).

En este framework, las páginas web se renderizan con HTML utilizando vistas elaboradas con Blade. Blade es un motor de plantillas incluido en Laravel que, a diferencia de otros motores de este estilo, Blade si permite el uso de PHP dentro de las plantillas. El modo de operación de Blade es compilar las plantillas en código PHP y almacenarlas dentro del caché hasta el momento en que estas sean modificadas. Para identificar los archivos pertenecientes a Blade, basta con identificar que contengan la extensión *.blade.php*; además, es habitual encontrar estos archivos en la carpeta *resources/views* (Laravel, 2023).

Laravel también permite la interacción con las bases de datos compatibles, y lo hace de manera simplificada mediante el uso de SQL sin procesar, un generador de consultas fluido y el ORM Eloquent. ORM Eloquent es un mapeador relacional de objetos que permite interactuar con la base de datos de una forma más amigable y más fácil de entender. Cuando se utiliza Eloquent, a cada tabla de la base de datos se le asigna un *modelo* que utiliza para interactuar con su tabla correspondiente.

Entre las acciones que se pueden realizar con ORM Eloquent se encuentran la inserción, actualización y eliminación de información (Laravel, 2023).

1.3.3.2 jQuery

jQuery es una librería basada en JavaScript que simplifica un documento de HTML en términos del manejo de eventos, animaciones e interacciones AJAX. Esta librería contiene funcionalidades que favorecen la reducción de código ofreciendo buenos resultados, y empleando menor tiempo y espacio. jQuery tiene buena compatibilidad con otras librerías de JavaScript, pero no se adapta bien con otros lenguajes de programación. También, es compatible con navegadores como Mozilla Firefox, Safari, Opera, Google Chrome e Internet Explorer (Villarreal Fuentes, 2013; B., Hostinger Tutoriales, 2023).

jQuery se distingue por su capacidad de cambiar el contenido de una página web sin necesidad de volver a cargarla, ya que solo manipula el DOM y peticiones AJAX para llevarlo a cabo. Las funciones propias de esta librería se distinguen porque llevan las estructuras $\$()$ y $jQuery()$, denominadas como *selectores*. Con ello, es posible seleccionar elementos específicos, manipular eventos e incluso hojas de estilo en cascada CSS, crear animaciones y efectos, realizar peticiones AJAX y otras utilidades como obtener información contenida en el navegador, o realizar operaciones con objetos (Villarreal Fuentes, 2013).

Para hacer uso de esta librería es necesario contar con un script de JavaScript dónde escribir las líneas de código con jQuery. Para interactuar con la página, será necesario escribir el selector $\$()$ y sus parámetros, que bien pueden ser expresiones CSS o etiquetas de elementos HTML. Esto obtendrá todos los nodos o elementos que concuerden con los parámetros especificados en el selector y, una vez obtenidos, ya es posible aplicarle cualquiera de las funciones que provee esta librería (Villarreal Fuentes, 2013).

1.4 Estado del arte

Anteriormente, se han presentado aportaciones tecnológicas que buscaron solucionar una problemática similar a la que se trata en el presente documento. Estas aportaciones contribuyen con resultados que señalan qué es lo que se hizo respecto a esa problemática para solucionarla, cuánto del problema pudo ser resuelto, y qué se sigue investigando o trabajando. Lo anterior sirve como guía de información relativo al tema o problemática que se desee abordar.

En las próximas páginas se describirán múltiples herramientas digitales dedicadas a la creación y aplicación virtual, e incluso tradicional, de evaluaciones utilizando diferentes software, no sin antes definir qué es una plataforma digital, que es el objeto que se espera producir al término del proyecto con las características señaladas por el Centro de Lenguas.

1.4.1 Plataforma digital

(De La Garza, 2021) define a una plataforma digital como un espacio en Internet donde es posible la ejecución de variadas aplicaciones dentro del mismo lugar para satisfacer diversas necesidades.

El sitio web (Cognizant, 2022) expresa que una plataforma digital consiste de un software y de tecnología utilizada para la unificación y optimización de operaciones de negocio y los sistemas de tecnologías de la información.

Una última definición, perteneciente al sitio web (Espacios Media, 2022), describe a este concepto como una solución en línea que permite efectuar numerosas tareas dentro del mismo lugar a través de Internet, y que cuyo propósito es lograr que múltiples servicios puedan interoperar unos con otros, para poder crear y gestionar aplicaciones, usuarios y, en ocasiones, también flujos de trabajo.

Como conclusión, una plataforma digital es un espacio en Internet capaz de gestionar diversas actividades y usuarios, y que permite a su vez la realización de diversas tareas dentro de la misma. Esto es precisamente lo que se busca desarrollar: una plataforma en la web que permita a los docentes o administradores la creación y gestión de exámenes digitales, y a su vez, que les permita a los alumnos realizar estos exámenes dentro de ese mismo espacio y que les otorgue el puntaje o calificación obtenida al término de su evaluación.

1.4.2 Google Forms

Google Forms es una herramienta digital que permite gestionar registros de eventos, encuestas de opinión, cuestionarios, etc. Su visualización no requiere el uso o instalación de un software adicional o especial, solo basta con contar con un navegador web o móvil, y el enlace hacia el formulario, el cual puede ser compartido de forma ilimitada. Sin embargo, es posible limitar el número de respuestas recibidas (Centro de Aprendizaje de Google Workspace, 2023).

El proceso que Google Forms ocupa para la creación de un formulario en línea no es para nada complejo, de hecho, su interfaz intuitiva y la abundante ayuda en línea proporcionada tanto por la empresa, como por otros usuarios, facilitan aún más su uso. Google Forms añade a las preguntas del formulario diferentes modos de interacción, dígame preguntas que requieren simplemente respuestas textuales, de selección múltiple, listas desplegables, etc., permitiendo hacer una versión digital muy fiel de lo que sería un cuestionario físico. Además, otra función valiosa en términos de creación de evaluaciones en el ámbito educativo, es la función de autocalificar la evaluación y añadir comentarios en caso de ser necesario. También, Google Forms permite añadir elementos multimedia a las preguntas, cuestión que resulta importante en exámenes de idiomas cuando se trata de evaluar habilidades como las auditivas o comprensión de lectura.

Dentro del ámbito educativo, los resultados que se pueden obtener con esta aplicación son formularios, sea cual sea su propósito, que actúen como evaluaciones y que cuya estructura asemeje a la de una evaluación de idiomas como las que aplica el Centro de Lenguas de la UAEH. Asimismo, Google Forms facilitará el trabajo de calificar numerosas respuestas obtenidas, ya que esta actividad puede ser ejecutada de forma automática por la aplicación siempre que se le indique. Además, los docentes o administradores podrían obtener reportes de cada registro de respuestas recibido, ya sea mediante gráficos o tablas, lo cual facilita la obtención de estadísticas.

Como conclusión, Google Forms es una herramienta muy completa que permite agregar preguntas de diferentes clases y que puede ser accedida desde casi cualquier dispositivo que tenga conexión a Internet. Además, esta aplicación posee funciones que pueden reducir en buena medida la carga de trabajo de los docentes y administradores. Las aportaciones a considerar que esta aplicación proporciona son la integración de diferentes formas de interacción del examen mediante múltiples tipos de preguntas, y la función de autocalificar las evaluaciones registradas.

1.4.3 Moodle

Moodle es una plataforma de aprendizaje preocupada por proveer a docentes, administradores y estudiantes un sistema único, robusto y seguro donde se puedan crear ambientes de aprendizaje totalmente personalizables (Moodle, 2022).

Esta plataforma de aprendizaje posee características atractivas como la facilidad de su uso, es un programa gratuito sin cargos por licenciamiento, recibe constantes actualizaciones, y es robusto, seguro y privado. El proceso de construir un examen en Moodle es muy sencillo y tiene una función llamativa: la de añadir preguntas de un banco de preguntas ya existente ya sea seleccionándolas manualmente, o dejar que sea Moodle quien las elija de manera aleatoria. Esta característica no solo permitirá ahorrar más tiempo en el proceso de creación de la evaluación, sino que también minimizará la tendencia a errores en la escritura de las preguntas y respuestas que puedan confundir a los alumnos en cada examen. Sin embargo, si no se desea hacer uso de esta función, Moodle ofrece la opción de elaborar el examen escribiendo cada interrogante de forma manual desde cero (Moodle, 2022). Para realizar las evaluaciones, será necesario que los usuarios accedan con sus credenciales a la plataforma y, una vez sean validadas y sea visible la página inicial del curso, se deberá hacer clic en el enlace del examen.

El resultado que esta plataforma provee, contrario a otros sistemas informáticos descritos en esta sección, es que los exámenes creados con Moodle utilizan una cantidad menor de tiempo en su proceso de creación derivado de la función de selección de preguntas ya existentes. A su vez, estos exámenes cuentan con la seguridad necesaria para evitar pérdida de información y evitar el mal uso de los datos por parte de otros usuarios, cuestión que no muchas aplicaciones de este estilo toman en cuenta. De hecho, ya que Moodle es una plataforma que requiere de credenciales para acceder a las evaluaciones, y que permite dividir al alumnado en cursos, es posible gestionar el acceso a determinadas evaluaciones.

Como conclusión, Moodle proporciona diferentes opciones para construir una evaluación, ya sea desde cero o tomando preguntas creadas anteriormente. También, esta plataforma se preocupa por la seguridad de la información, por lo que incluye los mecanismos necesarios para asegurar la integridad del examen y de los resultados que los alumnos generan. Finalmente, la aportación más considerable de esta plataforma es la característica de selección de preguntas aleatorias partiendo de un banco ya creado con anterioridad, pues suele suceder que los docentes aplican varias versiones diferentes de un examen a fin de evitar que los estudiantes compartan sus respuestas con otros alumnos; esta característica

ahorraría todavía más tiempo, pues se suprimiría la tarea de elegir manualmente qué preguntas conformarán cada evaluación.

1.4.4 GexCat

GexCat es un programa destinado, en un inicio, para la gestión, preparación y corrección de exámenes, y su público objetivo eran catedráticos de enseñanza de nivel medio superior y superior. Actualmente, GexCat es un programa que sirve para gestionar el proceso de preparación y corrección automática de exámenes estilo test, mejora su calidad y permite ahorrar tiempo hasta en un 99% (INNOCAN, 2022).

Una característica de este programa es el uso de una base de datos para almacenar las preguntas, problemas, prácticas, etc., y pueden o no incluir elementos multimedia o notación LaTeX. Un modo de gestión de las preguntas ofrecido por este programa es agruparlas en formatos o criterios como, por ejemplo, Asignatura/Tema o, también, utilizar la función que permite asignarles etiquetas que facilitan su selección y agrupación (INNOCAN, 2022).

GexCat no solo puede crear exámenes digitales, sino que también proporciona la opción de crear exámenes en papel. Independientemente de si el examen será aplicado en papel o en formato electrónico, el proceso de su creación es el mismo.

Los resultados obtenidos de elaborar exámenes con esta aplicación se resumen en que estas evaluaciones pueden ser reutilizadas y modificadas cuantas veces el usuario lo desee. Esto derivado de que GexCat provee modelos con los que se puede alterar el orden en que se presentan las preguntas y sus posibles respuestas, haciendo que cada examen sea, de cierto modo, diferente. De hecho, si cada alumno tiene un examen diferente, la probabilidad de que intenten copiar respuestas de otros alumnos disminuirá.

En conclusión, GexCat es un programa que permite a docentes elaborar evaluaciones que pueden ser aplicadas en formato digital e incluso en formato tradicional. GexCat puede no ser tan amigable en sus primeros usos ya que es necesario por lo menos un poco de capacitación para sacar su máximo provecho, pero una vez entendidos lo necesario de este programa, resulta ser una buena herramienta digital que facilita la creación de exámenes y que disminuye el tiempo empleado en ese proceso. Igualmente, se considera su característica de hacer uso de una base de datos para almacenar las preguntas, y hacer su selección en función de criterios o etiquetas para construir los exámenes.

1.4.5 Testmoz

Testmoz es una aplicación dedicada a generar exámenes y cuestionarios y, aunque su uso es sencillo, incluye numerosas herramientas empleadas por los docentes para la creación de sus exámenes como la posibilidad de ver promedios, puntajes y respuestas de los alumnos. Esta aplicación es de uso gratuito y no contiene anuncios publicitarios, no obstante, se puede pagar una cuota para tener acceso a herramientas adicionales, tener una cantidad ilimitada de preguntas por cada examen, entre otras opciones (Rodríguez Antonio, 2020).

Testmoz puede añadir preguntas de diferentes tipos a los exámenes, ya sean de opción múltiple, llenar espacios en blanco, etc., y pueden ser agrupadas dentro de bancos de preguntas. Para contestar sus exámenes, los alumnos no requerirán la creación de cuentas para acceder a esta aplicación, sino que solo se les solicitará introducir el código de acceso del examen, un identificador y su correo electrónico. Esta aplicación admite hasta 100 registros de respuestas recibidas y pueden ser visualizados en histogramas, y también, puede exportar estos datos en un archivo con extensión CSV para que se les puedan aplicar análisis de datos (Testmoz, 2022).

Los exámenes realizados con esta aplicación cumplen su propósito, permite agregar hasta un máximo de 50 preguntas por examen, y pueden ser autocalificados por esta aplicación. De hecho, estas evaluaciones pueden ser reutilizadas para que varios alumnos puedan responder un mismo examen, sin embargo, el límite de máximo 100 resultados podría presentar problemas para los docentes que requieran aplicar el examen a una cantidad superior de alumnos. A su vez, el límite de 50 preguntas por examen impide que se puedan crear exámenes para finalidades o propósitos que requieren de un número mayor de preguntas. Para distribuir la prueba, Testmoz ofrece una forma simple de hacerlo que consiste en distribuir el enlace al examen entre los alumnos que lo aplicarán, no obstante, es importante señalar que debido a que no es necesaria una cuenta para utilizar este software, no existe ninguna clase de control de acceso a las evaluaciones.

Testmoz es una aplicación destinada a los docentes que ofrece las herramientas necesarias gratuitamente y extiende el uso de algunas otras mediante el pago de licencia. También, este software no posee la suficiente seguridad para mantener protegidas las evaluaciones de cualquier tipo de riesgo o vulnerabilidad. Las limitantes en cuanto número de registros y número de preguntas máximo por examen representan un obstáculo para la creación de algunos tipos de exámenes. Por todo lo anterior, solo se consideran como aportaciones significativas los aspectos de seguridad que un sistema de este tipo debería incluir, así como reducir

1. CAPÍTULO I. MARCO TEÓRICO

o eliminar las limitaciones que pueden presentar a fin de que las evaluaciones que los alumnos realicen sean de calidad y sean apropiadas independientemente del tipo o propósito de su examen.

2. Capítulo II. Metodología

Cuando se desarrolla un proyecto, es recomendable utilizar una metodología que guíe su desarrollo con mayor eficiencia, además de facilitar la distribución y estimación de tiempos y de costes, y que el producto final sea de calidad. En el desarrollo de software, se contempla el uso de metodologías tradicionales, cuyos procesos carecen de flexibilidad y baja tolerancia a cambios; y ágiles que, contrario a las tradicionales, son flexibles y los cambios que surgen durante el proceso de desarrollo son administrados de forma que no afecten en gran medida al flujo de la metodología. Por este motivo, se decidió hacer uso de una metodología ágil para la construcción del Sistema de proceso de evaluación de perfil lingüístico.

En las siguientes páginas se hace mención de varias metodologías ágiles que bien pueden ser utilizadas para este propósito, y el capítulo es concluido con la elaboración de un cuadro comparativo de todas ellas en las que se analizan características de cada una como sus fases, cualidades principales, ventajas y desventajas; y una conclusión en donde se determina la metodología más viable para guiar el desarrollo de este proyecto computacional.

2.1 SCRUM

La metodología Scrum fue propuesta por Ken Schwaber, Jeff Sutherland y Mike Biddle a principios de la década de 1990. Scrum tiene como propósito proveer una solución simple y adaptable como respuesta ante problemas complejos (Schwaber & Sutherland, 2020).

Scrum utiliza un enfoque iterativo a fin de prever y controlar posibles riesgos; igualmente, involucra la creación de grupos de personas que cuenten con las habilidades y experiencias necesarias para llevar a cabo las tareas asignadas, y compartirlas con los demás si fuese el caso (Schwaber & Sutherland, 2020).

La metodología Scrum consta de múltiples fases, mostradas en la figura 5. (Sommerville, 2011) las describe de la siguiente manera:

1. **Planeación del bosquejo.** Se definen los objetivos generales del proyecto y se establece el diseño que tendrá la arquitectura del software.

2. **Ciclos *sprint*.** Durante estos ciclos, donde cada uno es un incremento del sistema, se puntualiza el trabajo que se va llevar a cabo, se seleccionan las características a desarrollar y se implementa el software.
3. **Fase de cierre.** El proyecto se concluye, se finaliza la documentación necesaria, y se valoran las lecciones aprendidas durante el desarrollo del proyecto.

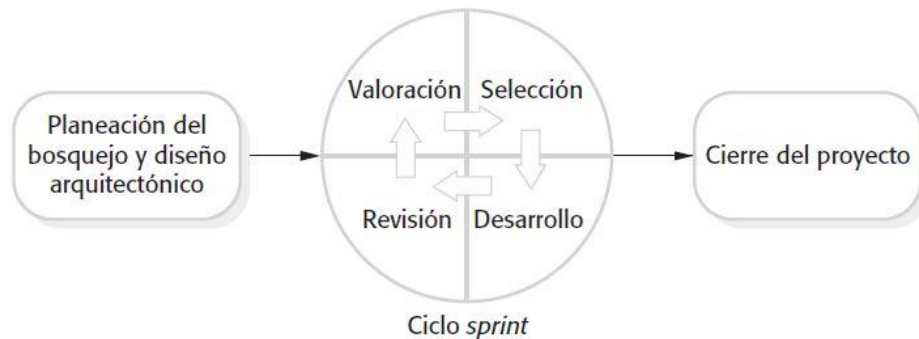


FIGURA 5. PROCESO DE SCRUM

Fuente: (Sommerville, 2011)

En Scrum, los equipos están conformados por un Scrum Master, un dueño del producto y los desarrolladores. Estos equipos tienen las características de ser multifuncionales, son autogestionados por los miembros del equipo, y no existen sub-equipos dentro de ellos ni se establecen jerarquías, sino que estos equipos actúan como una sola unidad enfocada en un objetivo a la vez (Schwaber & Sutherland, 2020).

Los roles ya mencionados consisten en lo siguiente:

- **Scrum Master.** Es el responsable de establecer Scrum y seguir su proceso, ayudando a los demás miembros a comprender tanto la teoría como la práctica de este método. Igualmente, es responsable de la efectividad del equipo Scrum. El Scrum Master es básicamente el líder que sirve al equipo y a la organización.
- **Dueño del producto.** Es la persona responsable de maximizar el valor del producto obtenido por el trabajo del equipo Scrum, y de gestionar la pila del producto (*Product Backlog*).
- **Desarrolladores.** Los desarrolladores son los miembros del equipo Scrum a los que se les encarga crear cualquier aspecto de un incremento usable en cada Sprint (Schwaber & Sutherland, 2020).

El proceso de Scrum incluye eventos, los cuales tienen el propósito de proveer la transparencia necesaria, y con los que es posible inspeccionar y adaptar los artefactos Scrum. Por lo tanto, de no ser realizados de acuerdo a lo establecido por la metodología, se omitirían las oportunidades de inspección y adaptación (Schwaber & Sutherland, 2020).

Dichos eventos se encuentran contenidos dentro de cada ciclo *sprint*. Estos son:

- 1) **El Sprint.** Tiene una duración fija comprendida usualmente igual o menor a un mes. Es necesario que antes de iniciar el siguiente *sprint*, el anterior debe ser concluido. Las actividades que se encuentran dentro de este evento incluyen la Planificación (*Sprint Planning*), los Scrums Diarios, la Revisión del Sprint (*Sprint Review*), y la Retrospectiva (*Sprint Retrospective*).
- 2) **Planificación del Sprint.** Este evento inicia el *Sprint* en el momento que se define el trabajo a realizar. La *Planificación del Sprint* aborda un total de 3 temas:
 - a) ¿Por qué el Sprint es valioso?
 - b) ¿Qué se puede hacer este Sprint?
 - c) ¿Cómo se realizará el trabajo elegido?
- 3) **Scrum diario.** Estas reuniones tienen una duración no mayor a 15 minutos para los desarrolladores del equipo Scrum, y suelen ser programadas en el mismo lugar a la misma hora. Los Scrums diarios ayudan a mejorar la comunicación, identificar impedimentos y permiten una toma de decisiones más rápida, lo cual suprime la necesidad de llamar a más reuniones.
- 4) **Revisión del Sprint.** En este evento se inspeccionan los resultados obtenidos en el *Sprint* para así poder determinar futuras adaptaciones. Aquí, el equipo Scrum y las personas interesadas revisan qué se logró y qué cambios surgieron durante el Sprint. Con esta información se procede a tomar una decisión sobre cómo se procederá a continuación.
- 5) **Retrospectiva de Sprint.** Su objetivo es planificar formas de elevar los niveles de calidad y eficacia, y es este evento el que concluye al Sprint, dando lugar a uno nuevo. En este evento se inspecciona el último *Sprint* revisando aspectos como individuos, interacciones, procesos, herramientas y su definición (Schwaber & Sutherland, 2020).

Finalmente, los productos Scrum están diseñados para aumentar la transparencia de la información. Por ello, cada uno de los artefactos que son inspeccionados tienen la misma base para ser fácilmente adaptados (Schwaber & Sutherland, 2020).

Estos artefactos son, de acuerdo a (Schwaber & Sutherland, 2020):

- **Product Backlog.** Es una lista ordenada que contiene la información necesaria para la mejora del producto. Los elementos pendientes del Product Backlog se consideran listos una vez se realiza un evento de planificación de *Sprint*.
 - a) **Product Goal.** Dentro del Product Backlog se encuentra el Product Goal, cuyo propósito es el de señalar al equipo el objetivo a largo plazo.
- **Sprint Backlog.** Se compone del Sprint Goal (*por qué*), de los elementos pendientes que contiene el Product Backlog (*qué*), y de un plan para entregar la iteración (*cómo*). Este es un plan hecho por y para los desarrolladores. El *Sprint Backlog* es una imagen visible en tiempo real del trabajo que planean realizar los desarrolladores durante el Sprint para lograr el objetivo de ese *Sprint*.
 - a) **Sprint Goal.** Este es el único objetivo para el *Sprint*, crea coherencia y enfoque, y proporciona flexibilidad en cuanto al trabajo necesario para alcanzar ese objetivo.
- **Iteración.** Una iteración es un escalón específico hacia el Product Goal. Estos incrementos se van sumando a los anteriores y se verifican de manera exhaustiva, de modo que, en conjunto, todos ellos funcionen correctamente. Es posible crear varios incrementos durante un Sprint, aunque se puede entregar una iteración a las partes interesadas previo a la finalización del Sprint. Es importante mencionar que al trabajo no se le puede considerar como parte de un incremento salvo que cumpla con la Definición de Realizado (*Definition of Done*).
 - a) **Definition of Done.** Consta de una descripción del estado del incremento cuando este cumple con los criterios de calidad señalados en el producto.

2.2 Extreme Programming - XP

La programación extrema, o simplemente *XP*, es una metodología creada por Kent Beck. *XP* es una metodología ligera, eficiente, flexible y de bajo riesgo para desarrollar software. También, es una de las más utilizadas y está diseñada para trabajar con pequeños equipos de desarrollo con un número de entre dos hasta diez programadores en el equipo (Beck, 2000).

De acuerdo a (Pressman, 2010), el proceso de la programación extrema ocupa a la orientación a objetos como paradigma predilecto de desarrollo, y abarca reglas y prácticas realizadas dentro de cuatro actividades: la planeación, el diseño, la codificación y las pruebas. En la figura 6 se puede observar este proceso.

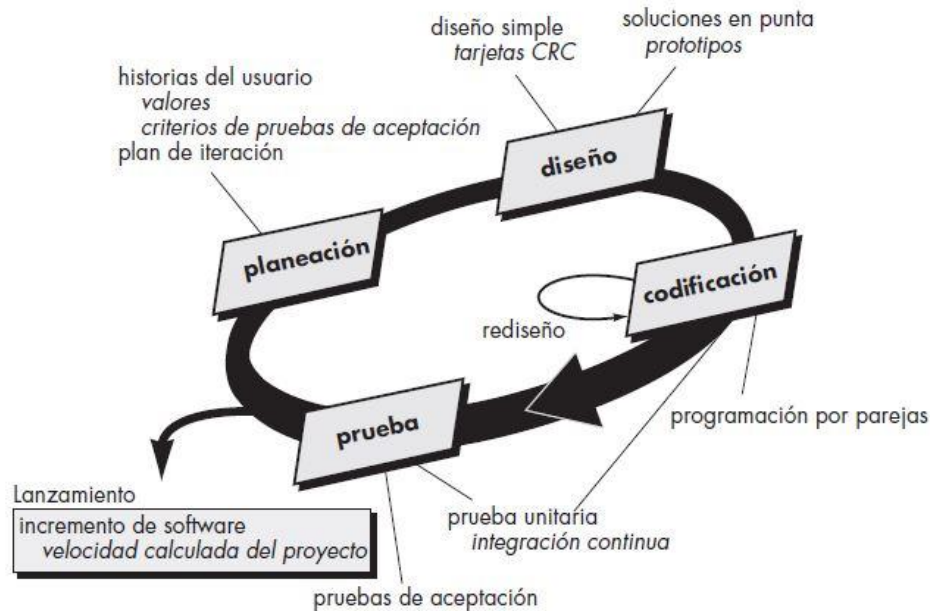


FIGURA 6. PROCESO DE LA METODOLOGÍA XP

Fuente: (Pressman, 2010)

A continuación, se resumen cada una de las actividades que conforman el proceso de XP de acuerdo a (Pressman, 2010):

1. **Planeación.** Esta fase consiste en *escuchar*, o sea, recabar los requerimientos que sirvan a los miembros del equipo a tener una idea del software. Esto permite la creación de historias de usuario, las cuales describen las características del software a desarrollar y cómo debe ser su funcionalidad.
2. **Diseño.** XP trabaja con el principio MS (mantenlo sencillo), esto quiere decir que se le da más preferencia a un diseño simple ante uno complejo. El diseño también sirve de guía para la implementación de una historia. La metodología XP también favorece el rediseño, pues éste se realiza tanto antes como durante su programación.
3. **Codificación.** Previo a la codificación, una vez que las historias están desarrolladas y se tenga un diseño preliminar, se desarrollan pruebas unitarias para cada una de las historias que se incluirán en la entrega. Una vez terminado el código, se le aplica una prueba unitaria y el resultado servirá de retroalimentación para los desarrolladores.
4. **Pruebas.** Las pruebas unitarias creadas se implementan utilizando alguna estructura que permita su automatización para estimular estrategias de pruebas de regresión cada vez que el código sea modificado. Conforme las pruebas unitarias se organizan, las pruebas de integración y validación pueden ejecutarse diariamente.

2.3 Dynamic Systems Development Method - DSDM

Por sus siglas en inglés, Método de Desarrollo de Sistemas Dinámicos, es una metodología propuesta con el fin de proveer una estructura para la construcción y mantenimiento de sistemas con limitaciones de tiempo reducido, a través de la realización de prototipos incrementales (Pressman, 2010).

Una versión modificada de la regla de Pareto dicta que el 80% de la aplicación puede ser entregado en un 20% del tiempo total que requiere para su desarrollo por completo. Dicho esto, DSDM utiliza un proceso iterativo en el que cada uno de sus incrementos se apega a la regla del 80 por ciento, sumando así el 100 por ciento. Esto significa que se requiere una cantidad tan solo suficiente del trabajo para cada incremento antes de ir al siguiente paso (Pressman, 2010).

(Pressman, 2010) expresa que el proceso de la metodología DSDM consta de tres ciclos iterativos más un par de actividades que les siguen. Estos son:

1. **Estudio de factibilidad.** Para evaluar si la aplicación es candidata a ser viable, primero se definen los requerimientos y las restricciones del negocio relacionados a la aplicación a desarrollar.
2. **Estudio de negocio.** En esta fase se establecen los requisitos funcionales e información que le dan valor al negocio, y se establece la arquitectura que tendrá el sistema.
3. **Iteración del modelo funcional.** Se crean prototipos iterados a fin de que el cliente conozca la funcionalidad con una demostración. La razón de existir de esta fase, es recolectar posibles requerimientos adicionales con la retroalimentación dada por los usuarios que prueban y practican con el prototipo.
4. **Diseño e iteración de la construcción.** Se revisan los prototipos ya creados en la fase anterior, para asegurar que en cada iteración se ha hecho ingeniería que pueda dar valor operativo del negocio a los usuarios finales.
5. **Implementación.** Coloca el software obtenido en la iteración más reciente dentro del ambiente de operación. Es importante apuntar que el incremento puede no ser el final, y que cabe la posibilidad de realizar los cambios solicitados una vez el incremento esté en su lugar.

2.4 Adaptive Software Development - ASD

La metodología ASD, por sus siglas en inglés, Desarrollo Adaptativo de Software, fue creada por Jim Highsmith, y está destinada al desarrollo de software y sistemas complejos tomando como base la filosofía de la colaboración humana y la organización del equipo. Su autor afirma que una metodología centrada en la adaptación y fundamentada en la colaboración, da como resultado una “*fuentes de orden en las interacciones, tales como disciplina e ingeniería*” (Pressman, 2010).

El proceso para el desarrollo de software con esta metodología se sustenta en un ciclo constituido por tres fases: la especulación, la colaboración y el aprendizaje (Pressman, 2010). La figura 7 presenta un diagrama de este ciclo de vida, precedido de una breve explicación de cada etapa:

1. **Especulación.** Se inicia el proyecto y se hace la planeación adaptativa del ciclo. La especulación hace uso de la información de inicio del proyecto, como son la misión de los clientes, restricciones del proyecto y los requerimientos básicos, y con ello establecer el conjunto de ciclos o incrementos de software de entrega necesarios para el proyecto. Inevitablemente, el plan del ciclo cambiará sin importar cuán complejo sea. Es por esto que el plan se revisa y se ajusta una vez terminado el primer incremento, adecuándose así a la realidad en la que el equipo DAS se encuentra trabajando.
2. **Colaboración.** Los niveles elevados de talento y producción creativa con los que trabajan las personas, derivan de su motivación y el empleo de la colaboración. Este es un tema bastante común en lo que refiere a las metodologías ágiles. Y aunque la colaboración no siempre es sencilla, se destaca el individualismo gracias a la creatividad de cada persona, pues favorece el pensamiento colaborativo.
3. **Aprendizaje.** A medida que se desarrollan los componentes que conforman el ciclo adaptativo, el “aprendizaje” de lo que se tiene en el avance hacia la terminación del ciclo se vuelve el punto de interés. Así, los equipos tienen tres formas de llegar a este aprendizaje: mediante grupos de enfoque, con revisiones técnicas y con el análisis post mórtem del proyecto.
4. El alto grado de probabilidad de éxito de los proyectos tecnológicos elaborados siguiendo el proceso de ASD se debe a la atención central que se pone sobre la dinámica de los equipos con organización propia, el aprendizaje individual y colaborativo, y el trabajo en equipo (Pressman, 2010).

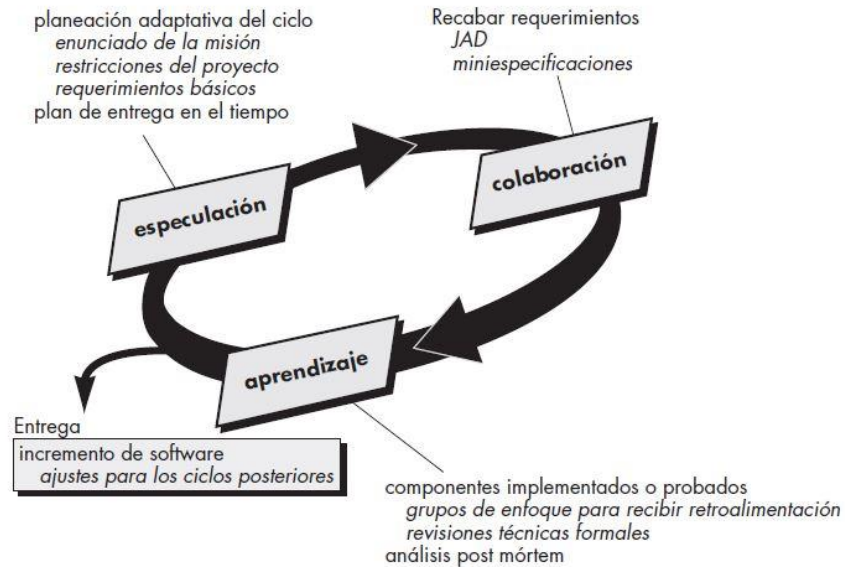


FIGURA 7. FASES DE ASD.

Fuente: (Pressman, 2010)

2.5 Metodología a emplear

El Sistema de proceso de evaluación de perfil lingüístico puede ser desarrollado siguiendo los pasos de cualquiera de las metodologías presentadas en este capítulo. Para elegir la metodología que mejor acomode a los participantes involucrados en el desarrollo de este sistema, se realizó un cuadro comparativo, mostrado en la tabla 1, donde se resaltan las características más relevantes de cada una de las metodologías mencionadas en este capítulo, sus pasos o etapas, y sus ventajas y desventajas.

Después de analizar las posibles metodologías a emplear, se determinó que Scrum sería la más ideal. Aunque otras metodologías poseen características similares como lo son el alto nivel de adaptación ante los cambios, empleo de ciclos incrementales para gestionar tiempos y actividades, y la posibilidad de ser aplicadas en equipos de personas reducidos en número, se eligió Scrum por lo sencillo que resulta y porque, en comparación con las otras metodologías:

- La desventaja de Scrum de ser aplicado a equipos pequeños, como se menciona en la tabla 1, no presenta ningún inconveniente en el desarrollo de este proyecto, pues no son demasiadas personas las que están involucradas en el desarrollo.

2. CAPÍTULO II. METODOLOGÍA

- Otras metodologías expresan que es indispensable que el cliente se encuentre presente en todo momento, cuestión que no sería posible debido a que la mayor parte del tiempo, los clientes y usuarios finales de este sistema requieren realizar otras actividades en sus lugares de trabajo.
- El proceso de desarrollo aplicando otras metodologías sería mucho más extenso e incluso comprender la metodología elegida en sí misma requeriría mayor esfuerzo y tiempo por parte del equipo.
- Los miembros del equipo no poseen ninguna clase de experiencia aplicando las metodologías, a excepción de Scrum. Scrum no es complicado y su aprendizaje estaría guiado por aquellos con experiencia en su uso para quienes no la tienen.
- Las reuniones diarias que ofrece Scrum facilitarían la retroalimentación por parte del equipo sobre el trabajo realizado, el trabajo por hacer y el trabajo que será realizado próximamente, permitiendo mejor gestión de las actividades y los tiempos empleados para cada una de ellas.

TABLA 1. CUADRO COMPARATIVO DE METODOLOGÍAS ÁGILES

Metodología	Qué es	Características notables	Fases/Actividades /Eventos	Ventajas	Desventajas
Scrum	Scrum es un marco de enfoque iterativo, cuyo propósito es ofrecer soluciones rápidas y adaptables ante requerimientos cambiantes y tiempos de entrega cortos (Equipo de redacción de Drew, 2019).	<ul style="list-style-type: none"> – Su aprendizaje no es complejo, pero dominarlo sí. – Cada miembro del equipo Scrum juega un rol con tareas específicas. – Utiliza equipos de trabajo con pocos integrantes. – Utiliza tres <i>artefactos Scrum</i> que sirven como punto de partida para la ejecución de tareas. – Entrega de avances periódicos al cliente. 	<ul style="list-style-type: none"> – Planificación del Sprint. – Ciclo Sprint. – Scrum diario. – Revisión del Sprint. – Retrospectiva del Sprint. 	<ul style="list-style-type: none"> – Entrega de resultados anticipados. – Flexibilidad y adaptación ante los problemas. – Gestión de actividades donde se da prioridad al trabajo más urgente. – El trabajo segmentado en iteraciones facilita la gestión de actividades y de tiempos. – Retroalimentación en el equipo gracias a las reuniones diarias. 	<ul style="list-style-type: none"> – Se aplica a grupos reducidos. – Es estrictamente necesario definir las tareas y los tiempos para efectuarlas, para que Scrum funcione correctamente. – Requiere experiencia por parte de los usuarios en su uso.
XP	Se trata de una metodología iterativa, ligera, flexible y de bajo riesgo utilizada para desarrollar	<ul style="list-style-type: none"> – Retroalimentación inmediata. – Tendencia a lo sencillo. 	<ul style="list-style-type: none"> – Planeación. – Diseño. – Codificación. – Pruebas. 	<ul style="list-style-type: none"> – No es necesario programar trabajos innecesarios. 	<ul style="list-style-type: none"> – El cliente está implicado en el proceso. – El tiempo requerido es extenso.

2. CAPÍTULO II. METODOLOGÍA

	software y que puede ser adaptada para trabajar hasta con un máximo de 10 programadores dentro del equipo (IONOS, 2020).	<ul style="list-style-type: none"> – Está orientado a las necesidades del cliente. – Los cambios se realizan por pasos pequeños y por incrementos. – Los cambios son aceptados como algo positivo. 		<ul style="list-style-type: none"> – Menos tendencia a errores gracias al trabajo colaborativo en parejas. – Los cambios pueden ser aplicados rápidamente. – El código debe ser de comprensión sencilla todo el tiempo. – El software resultante es estable gracias a las constantes pruebas que se le aplican. 	<ul style="list-style-type: none"> – Requiere mayor esfuerzo de trabajo. – Relativamente caro.
<i>DSDM</i>	Es una metodología de enfoque ágil e iterativo que proporciona la estructura para construcción y mantenimiento, a sistemas que posean restricciones apretadas de tiempo a través de la construcción de prototipos (Pressman, 2010; Anónimo, Ingeniería de software, 2018).	<ul style="list-style-type: none"> – Sigue la regla del 80 por ciento: se requiere solo suficiente trabajo para cada incremento y los detalles se terminan después. – Involucra al cliente en gran medida. – Ideal para proyectos con presupuesto y tiempos apretados. – Se preocupa por entregar avances del producto final durante el ciclo. 	<ul style="list-style-type: none"> – Estudio de factibilidad. – Estudio de negocio. – Iteración del modelo funcional. – Diseño e interacción de la construcción. – Implementación. 	<ul style="list-style-type: none"> – Los cambios se efectúan sin mayor problema. – El tiempo de desarrollo y costo del proyecto se reducen. – La calidad del producto final incrementa gracias a la constante participación del cliente. – Facilita la gestión de tiempo y costo del proyecto. 	<ul style="list-style-type: none"> – Es necesario que cada entrega del software sea a tiempo, de modo que se respete el presupuesto y su calidad. – El sistema, naturalmente, no será realizado a la perfección en el primer intento. – La participación del cliente es indispensable. – Su comprensión puede resultar compleja.
<i>ASD</i>	Se trata de una metodología de enfoque iterativo, destinado al desarrollo de software que sigue el principio de adaptación continua, y toma como base la colaboración humana y la organización del	<ul style="list-style-type: none"> – Se orienta más a los componentes del software que a las tareas. – Tolerancia los cambios y no lucha contra ellos. – La revisión de los componentes 	<ul style="list-style-type: none"> – Especulación. – Colaboración. – Aprendizaje. 	<ul style="list-style-type: none"> – Es utilizada para aprender de los errores y así iniciar nuevamente el ciclo de desarrollo. – Utiliza la información disponible de los cambios para mejorar 	<ul style="list-style-type: none"> – Los errores que no fueron detectados anteriormente afectarán la calidad del producto y su costo. – Como sucede con otras metodologías ágiles, no es posible

2. CAPÍTULO II. METODOLOGÍA

equipo (Pressman, 2010; Anónimo, Metodologías Ágiles de Desarrollo, 2012).

favorece el aprendizaje de errores y la colaboración entre desarrolladores y clientes.
– Guiado por riesgos.

el comportamiento del software.
– Favorece la colaboración e interacción entre las personas.

realizar procesos propios de las metodologías tradicionales.

3. Capítulo III. Diseño y desarrollo

El punto de partida para el desarrollo del proyecto es el análisis de los requerimientos expresados por el cliente, o sea, comprender lo que ellos necesitan que el sistema haga y así construir un proyecto lo más cercano a ello. Estos requerimientos se obtuvieron a través de múltiples reuniones, utilizando como forma de recolección de información reuniones con los usuarios finales y con el director del Centro de Lenguas, quienes compartían su opinión y expresaban sus inquietudes y sugerencias de lo que el sistema debería tener o hacer. En estas reuniones se encontraban presentes los miembros del equipo de desarrollo, acompañados de todos aquellos que serían los usuarios finales, y el dueño del producto.

Ya con los requerimientos definidos, se decidió seleccionar la metodología Scrum para el proceso de desarrollo del sistema. Dentro de cada uno de los *Sprints* del ciclo Scrum, se realizaron actividades que incluyen la recolección de los requerimientos funcionales y no funcionales que deberá incluir el sistema, diagramación de la estructura del sistema empleando el lenguaje de modelado UML, mediante casos de uso, diagramas de actividades y un diagrama de clases, y la estructura de la base de datos a través de los modelos entidad relación (E-R) y relacional. Igualmente, dentro de los incrementos se realizaron las tareas de programación y generación de la base de datos, desde el script hasta su puesta en marcha en un entorno gráfico para su administración, todo ello de acuerdo al *Sprint Backlog*.

3.1 Análisis de requerimientos

El sistema tiene como propósito servir como un proceso automatizado de la tarea de elaboración de exámenes que, hasta el momento, se realiza de manera tradicional y de manera digital utilizando un software que, eventualmente, no cumplió con su propósito como se esperaba. Por ello, los requerimientos recolectados de los usuarios contemplan las tareas que este proceso conlleva, a fin de ser una herramienta que pueda ser de utilidad tanto para alumnos, docentes y administradores en todos los institutos y dependencias de la UAEH.

Los requerimientos recolectados expresan que el sistema está destinado a la generación de exámenes partiendo de criterios o especificaciones dados

previamente, y su visualización debe ser dinámica y ser capaz de soportar diferentes tipos de preguntas para que el estudiante pueda interactuar con su evaluación de diferentes maneras, así como también que el sistema pueda proveer de retroalimentación y una calificación asignada, junto con una lista de sugerencias sobre los temas que deberían prestar especial atención para estudiar. Además, este sistema debería tener compatibilidad en términos de integración con otros sistemas que el Centro de Lenguas posee y con otros que se encuentra desarrollando actualmente. Así, las características que este sistema debe poseer son las siguientes.

3.1.1 Requerimientos funcionales

- El sistema deberá ser capaz de generar exámenes de manera automática y con preguntas seleccionadas de manera aleatoria, pero atendiendo a las características dadas.
- La información ingresada por los alumnos debe ser guardada dentro del sistema de modo que, en caso de algún accidente, pérdida de conexión a Internet o alguna situación similar, y en cuanto el examen sea abierto de nuevo, las respuestas previamente ingresadas continúen ahí.
- El sistema deberá indicar si aún existen preguntas del examen sin responder antes de que este sea enviado, así como también contará con un tiempo límite para ser respondido, y el tiempo restante será mostrado en la misma pantalla donde se muestre el examen. Además, los alumnos no podrán enviar salvo un intento de un mismo examen.
- Al término de su examen, el sistema deberá: calificarlo y mostrar el tiempo empleado, número de aciertos y puntaje o calificación obtenidos; la retroalimentación del examen con las preguntas, la respuesta correcta y la respuesta dada por el alumno; y enlistar los temas a los que el alumno debe prestar especial atención para estudiar a fin de alcanzar mayor comprensión del tema, teniendo como referencia las preguntas contestadas incorrectamente.
- Los administradores o docentes podrán conocer información del examen como las preguntas que le fueron asignadas, idioma, unidad, materia, tipo de examen y las habilidades que evalúa.
- El examen deberá mostrar elementos multimedia siempre que la pregunta los requiera.
- Los exámenes construidos deben contar con diferentes tipos de preguntas para que el alumno pueda interactuar de diversos modos con su evaluación,

siendo estos: opción múltiple, selección múltiple, lista desplegable y preguntas abiertas.

- El sistema deberá mostrar la información de los estudiantes registrados por el Centro de Lenguas.

3.1.2 Requerimientos no funcionales

- Los criterios con los que cada examen debe ser construido son:
 - Hora de inicio y hora de finalización especificados con hora y minutos.
 - Fecha de inicio y fecha de finalización especificados con día y mes.
 - Duración especificada con horas, minutos y segundos.
 - El número máximo de alumnos por examen.
 - Número de preguntas que conformarán la evaluación.
 - Tipo o propósito del examen.
 - Habilidad a evaluar.
 - Modalidad en la que será aplicada la evaluación.
 - Semestre al que está destinada la evaluación.
 - El idioma que evaluará (para el caso de este sistema, solo contará con el idioma alemán).
 - La materia para la cual fue creado.
 - La unidad a la que corresponde del programa temático.
- El examen deberá mostrar una confirmación cuando esté por enviarse, o una alerta cuando trate de ser enviado con preguntas no respondidas aún.
- Los colores utilizados en el sistema serán los incluidos en la paleta de colores otorgada por el Centro de Lenguas.
- Además de los elementos multimedia proporcionados, las imágenes faltantes a utilizar durante el desarrollo del proyecto, deberán ser extraídas de la plataforma web Pixabay, y cada una deberá proporcionar los derechos de autor haciendo uso del símbolo “©” y el nombre de esta plataforma.
- Cuando el tiempo disponible para contestar la evaluación se agote, el sistema deberá mostrar una alerta indicando que no existe más tiempo disponible restante, y la evaluación será enviada de manera automática.
- Los exámenes y su evaluación deberán estar apegados al Programa Institucional de Lenguas del Centro de Lenguas de la UAEH.
- En la interfaz correspondiente a la vista del examen se deberán mostrar los elementos:
 - Barra de progreso para que el alumno conozca el avance que ha realizado en su evaluación conforme navega a través de su examen.

- Un elemento que indique el número de pregunta actual contra el número de preguntas totales (por ejemplo: 7/20).
- Información adicional del examen como es el tipo de evaluación, el idioma y el semestre.
- Igualmente, deberá mostrar el tiempo total destinado para contestar el examen y el tiempo restante en todo momento.
- Se mostrará la pregunta actual con sus posibles respuestas (en caso de que aplique) de acuerdo al tipo de pregunta al que pertenezca.
- El modo de navegación del examen será navegar a través de las preguntas en dirección hacia adelante y hacia atrás con un par de botones que indiquen “Siguiente” y “Atrás”, respectivamente.
- La información mostrada en la retroalimentación se deberá mostrar, para mejor comprensión, en una tabla:
 - La fecha y hora de inicio del examen.
 - El estado en el que se encuentra el examen.
 - La fecha y hora de finalización.
 - El tiempo empleado expresado en horas.
 - El número de aciertos del examen del total de incisos del examen.
 - La calificación o puntaje obtenido.
- La retroalimentación también deberá presentar las posibles respuestas (si el tipo de pregunta aplicara), e indicar las respuestas del alumno de la siguiente manera: con color verde la(s) respuesta(s) correcta(s), con color rojo la(s) respuesta(s) incorrecta(s) o, si fuese el caso, las preguntas no respondidas deberán indicarlo. También, cada pregunta deberá estar acompañada, independientemente de si la pregunta fue contestada correcta o incorrectamente, de la respuesta correcta y del elemento multimedia si tuviera uno relacionado.
- Para el caso de las preguntas abiertas, se mostrará la leyenda “Texto pendiente por revisar” en lugar de una calificación o puntaje asignado.
- Derivado del punto anterior, el puntaje obtenido será el tentativo, pues en caso de que el examen tuviera preguntas abiertas que deban ser revisadas por un docente o un administrador, no serán consideradas y solo se tomarán en cuenta las preguntas a las que el sistema pueda asignar una calificación automáticamente sin ayuda de un experto.
- Las sugerencias de estudio deberán mostrarse al finalizar la retroalimentación y no deberán repetirse, de modo que, por ejemplo, si dos preguntas fueron contestadas erróneamente y corresponden al mismo tema, bastará con que la sugerencia se haga una sola vez.

3.2 Diagramas de Casos de Uso

Ahora que se conocen los requerimientos del sistema, el siguiente paso es diagramar su estructura. Para ello, se emplearon diagramas UML a fin de que estos puedan ser entendidos por todas las personas involucradas en el desarrollo del sistema, así como el cliente y los usuarios finales. Los diagramas UML empleados para el sistema incluyen los diagramas de caso de uso, los diagramas de actividades y el diagrama de clases.

3.2.1 Caso de uso para alumnos

El primer paso es definir las actividades que los diferentes tipos de usuario pueden hacer en el sistema. Para el caso de los alumnos, independientemente de si son internos o externos a la institución, el servidor les proporcionará a los alumnos la capacidad de contestar multitud de tipos de evaluaciones de acuerdo al programa educativo y asignatura que se encuentren cursando, conocer sus resultados y las sugerencias dadas por el sistema para mejorar sus estudios y dominio del idioma alemán. Estas actividades son expresadas en el diagrama de la figura 8.

3.2.2 Caso de uso para docente y administrador

Similar a la situación de los alumnos, y teniendo en cuenta los alcances del presente sistema, el servidor le proporcionará a los docentes o administradores la capacidad de generar exámenes, ver todos los exámenes generados hasta el momento, conocer las preguntas que les fueron asignadas de manera aleatoria, eliminar los exámenes, y visualizar el examen como lo vería el alumno.

Igualmente, podrán acceder al listado de alumnos registrados. El diagrama de caso de uso de esta situación se observa en la figura 9.

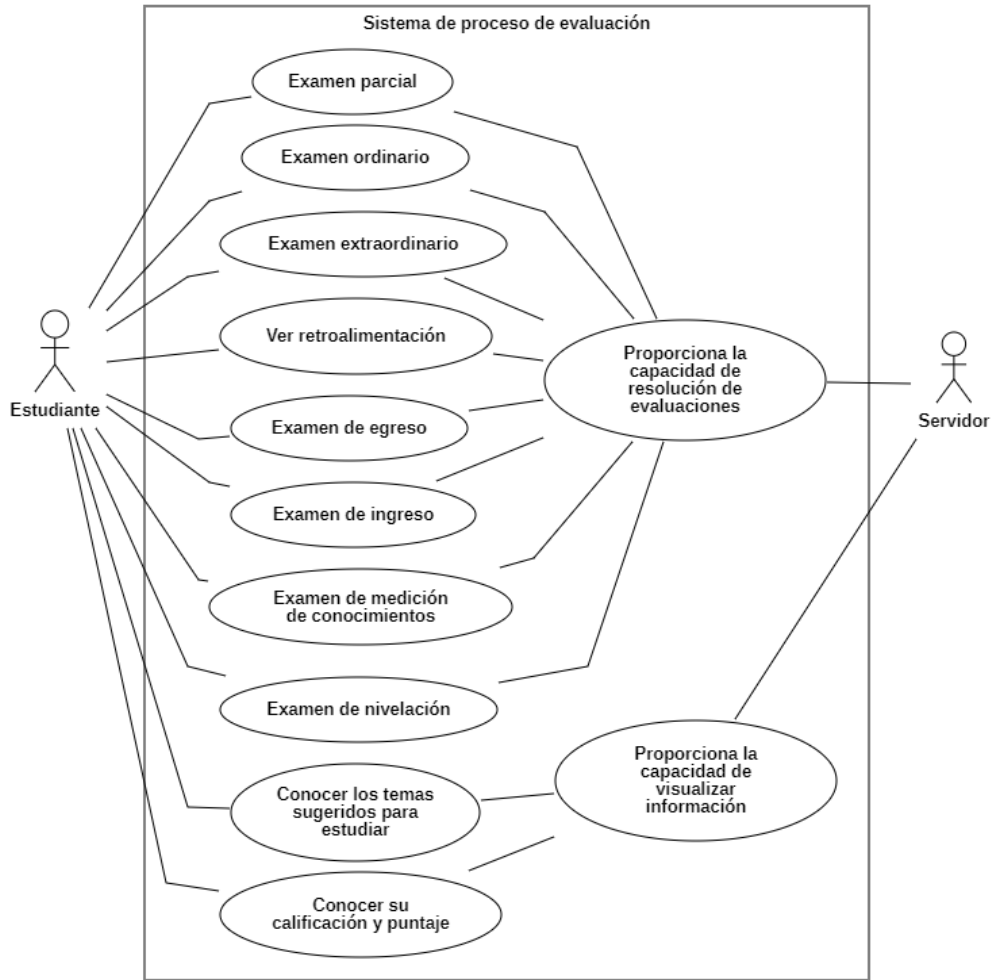


FIGURA 8. DIAGRAMA DE CASOS DE USO PARA ALUMNOS

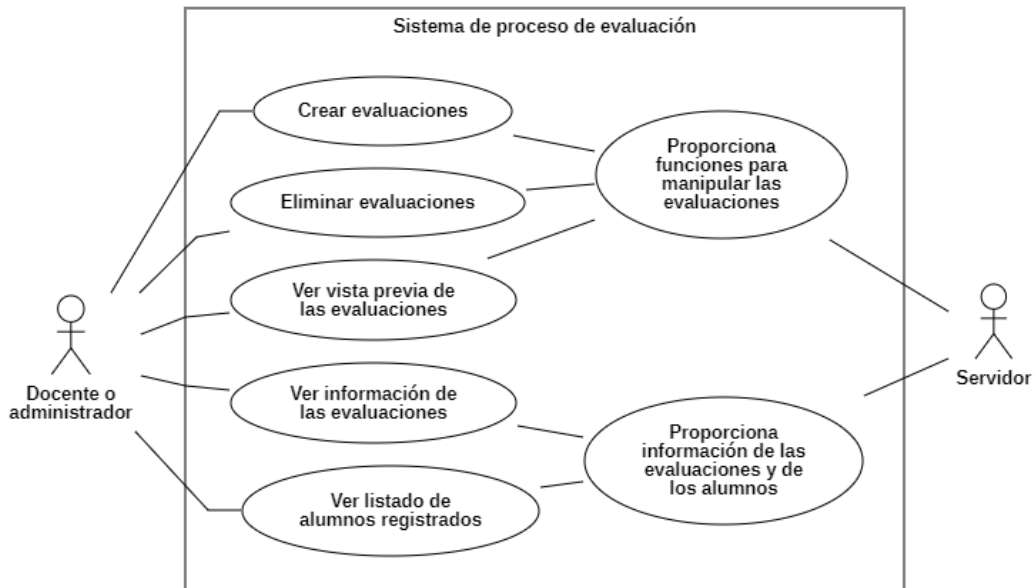


FIGURA 9. DIAGRAMA DE CASOS DE USO PARA DOCENTE O ADMINISTRADOR

3.2.3 Caso de uso para crear una evaluación

Para crear un examen, el usuario deberá interactuar con el sistema accediendo al formulario destinado para la generación de evaluaciones, completar los campos que contiene y solicitar su creación al servidor.

Por otra parte, la interacción que el servidor tendrá en este caso con el sistema, será a través de los mensajes de errores encontrados por los datos introducidos anteriormente por el usuario, o notificándole que el número de preguntas solicitado es insuficiente. También, será tarea del servidor crear las evaluaciones de acuerdo a los criterios especificados por el docente o por el administrador. Estas interacciones pueden apreciarse en la figura 10.

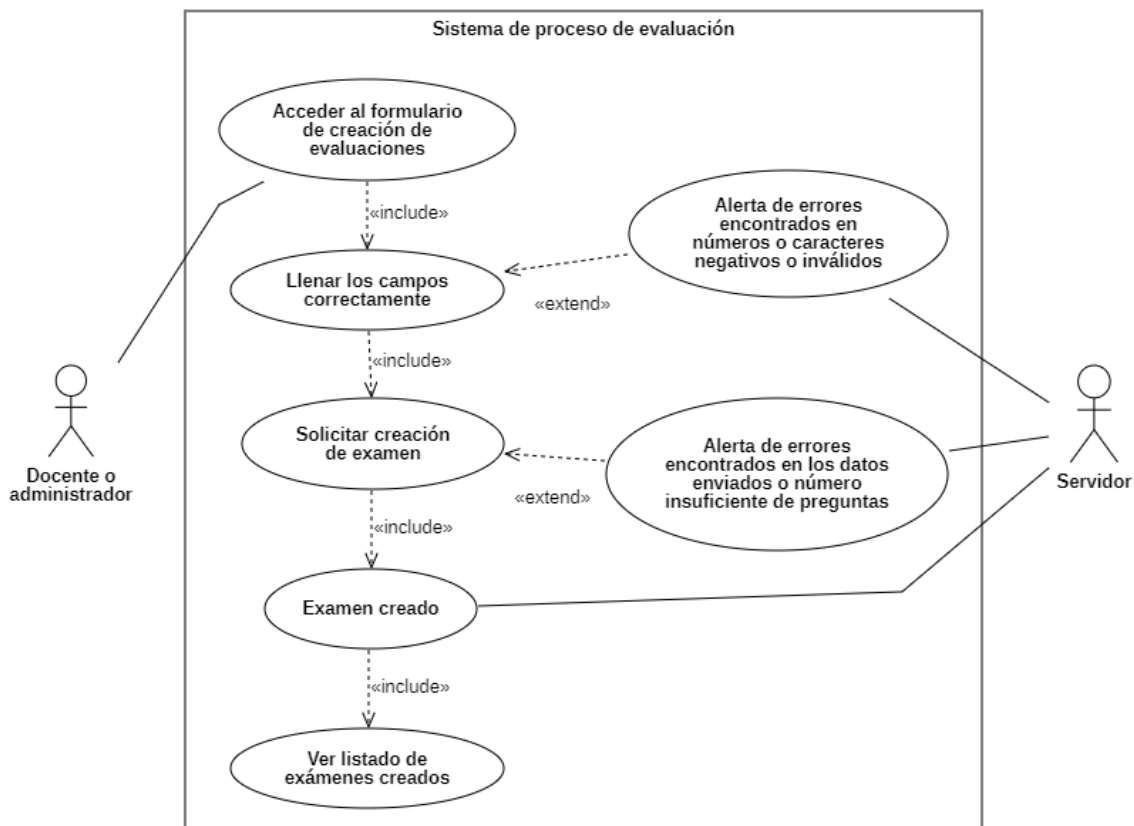


FIGURA 10. DIAGRAMA DE CASOS DE USO PARA CREAR UN EXAMEN

3.2.4 Caso de uso para visualizar preguntas asignadas a la evaluación

Otra función que este sistema posee es la de mostrarle al docente o al administrador las preguntas con las que cuenta la evaluación generada. En la figura 11 se expresa que las únicas interacciones que el usuario debe hacer con el sistema, es acceder al listado de exámenes y solicitar dicha información. Esta solicitud será procesada por el servidor y devolverá los datos de las preguntas a la vista para que puedan ser visualizados. Es totalmente opcional si el usuario desea conocer más información de una pregunta en especial, lo cual requerirá que haga una solicitud más para ver esta información.

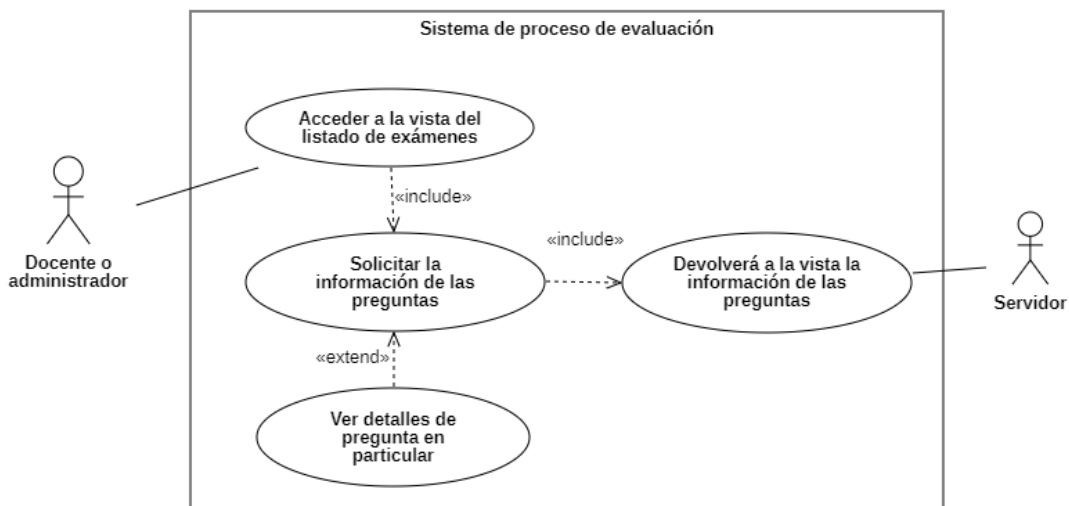


FIGURA 11. DIAGRAMA DE CASOS DE USO PARA VISUALIZAR PREGUNTAS ASIGNADAS A UN EXAMEN

3.2.5 Caso de uso para eliminar una evaluación

La figura 12 presenta las interacciones necesarias por parte del docente o del administrador, y del servidor para eliminar un examen correctamente. Este proceso consiste en que el usuario deberá acceder al listado de exámenes y localizar aquel que debe ser eliminado, después deberá hacer la petición de eliminación, el servidor enviará una solicitud de confirmación. Si se confirma, el servidor se encargará de realizar todos los procedimientos necesarios para que el examen sea eliminado en su totalidad. En caso contrario, el examen no será eliminado.

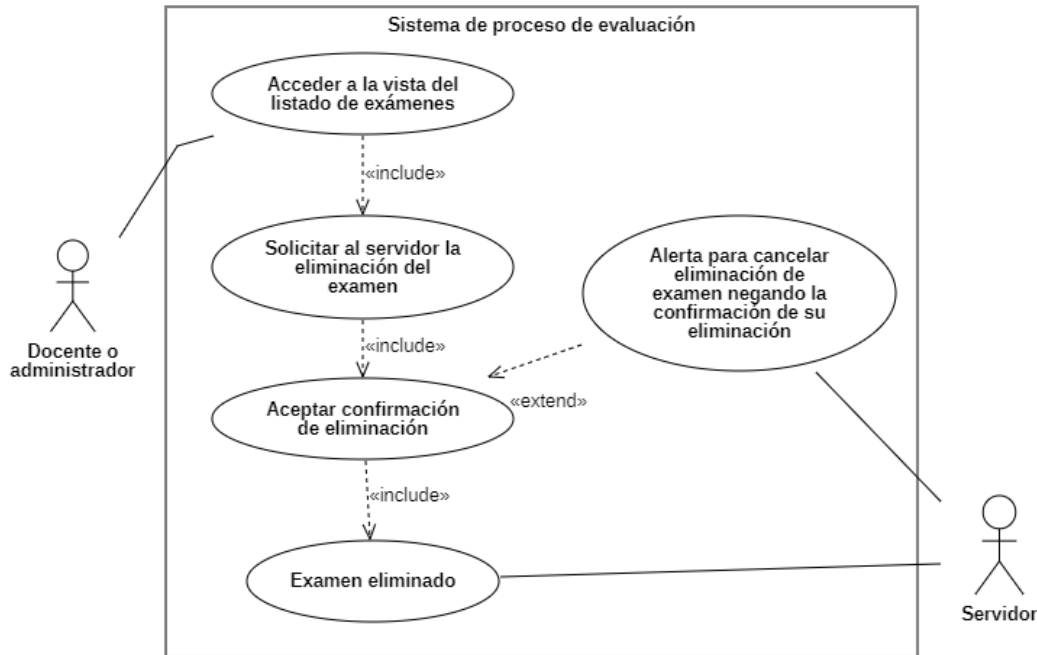


FIGURA 12. DIAGRAMAS DE CASOS DE USO PARA ELIMINAR UN EXAMEN

3.2.6 Caso de Uso para contestar una evaluación

Aunque esta tarea es propia de los estudiantes, dados los alcances de este proyecto donde no se tiene acceso al sistema o módulo encargado de la gestión de roles y asignación específica de tareas, los exámenes generados en el presente sistema podrán ser vistos de manera general desde el listado de exámenes creados. Dicho esto, el usuario deberá interactuar con su examen primero seleccionando el examen que va a contestar, y después dar respuesta a cada una de las preguntas. Una vez haya terminado de responder todos los cuestionamientos, deberá enviarlo y será tarea del servidor notificar errores en el envío, calificarlo, y devolver la retroalimentación y los temas sugeridos de estudio.

El diagrama de caso de uso correspondiente a esta situación se muestra en la figura 13.

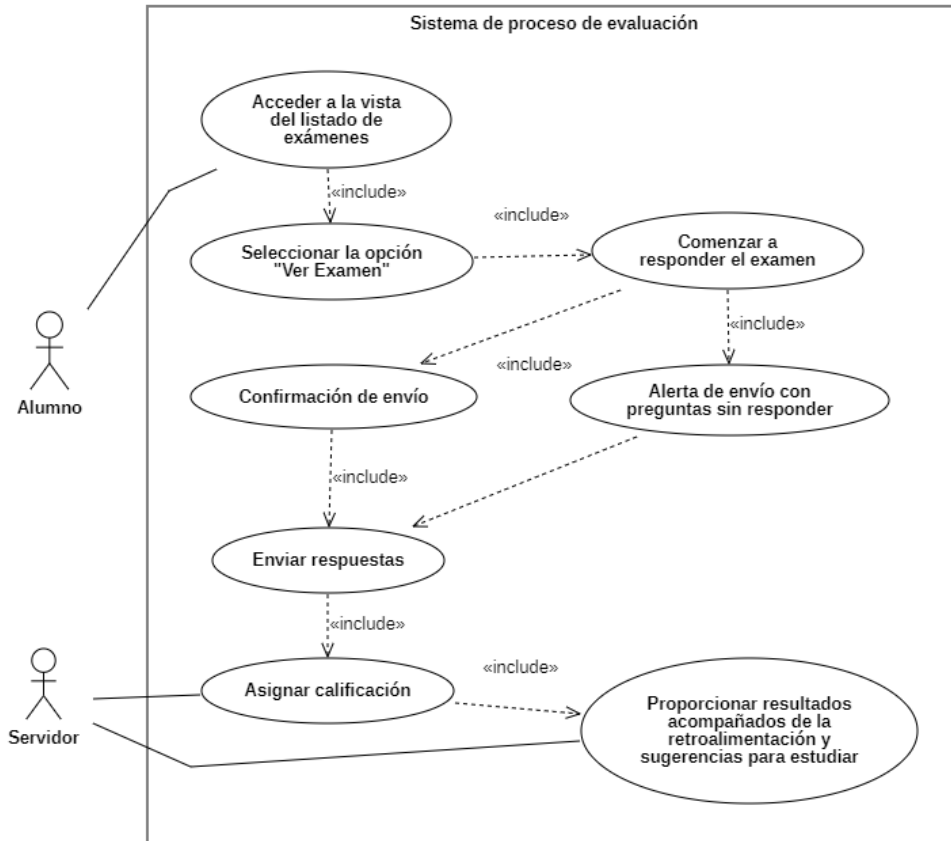


FIGURA 13. DIAGRAMA DE CASOS DE USO PARA CONTESTAR UN EXAMEN

3.3 Diagramas de Actividades

Además de los diagramas de casos de uso, se diseñaron diagramas de actividades que se caracterizan por mostrar el flujo de las actividades que realiza el sistema, y que presenta las interacciones de este último con de los usuarios. Los diagramas de actividades presentados a continuación muestran las actividades que los alumnos y los docentes y administradores pueden realizar en el sistema, y las acciones que deben ejercer para completarlas. Estas actividades incluyen la creación de una evaluación, su resolución, visualización y eliminación.

3.3.1 Diagrama de actividades para crear una evaluación

Las actividades que conforman el proceso de creación de un examen son expresadas en la figura 14, las cuales son acceder al sistema en el formulario destinado para esta función, ingresar los valores correspondientes en cada campo y, en caso de que existiera algún error en estos datos introducidos, el servidor lo deberá notificar inmediatamente.

Si no se encuentra ningún problema, el examen podrá ser creado seleccionando la opción "Crear examen". El servidor se encargará de revisar que no existan datos inválidos, decimales o negativos, y que el número de preguntas con los criterios solicitados sean suficientes. Si se llegase a encontrar alguna irregularidad relacionada a estas situaciones, el servidor retornará el mensaje de error correspondiente, y solicitando de nuevo al usuario que ingrese nuevos valores. Finalmente, si ya no se encontraron irregularidades en los datos, el examen procederá a ser creado por el servidor y redireccionará al usuario al listado de los exámenes generados previamente mostrándolos en forma de tablas y acompañados de información como el propósito para el que están destinados, idioma, número de preguntas, etc., y con un grupo de botones que permiten ejercer acciones sobre cada evaluación, entre los que se encuentran visualizar la prueba, eliminarla o ver las preguntas que el servidor le asignó.

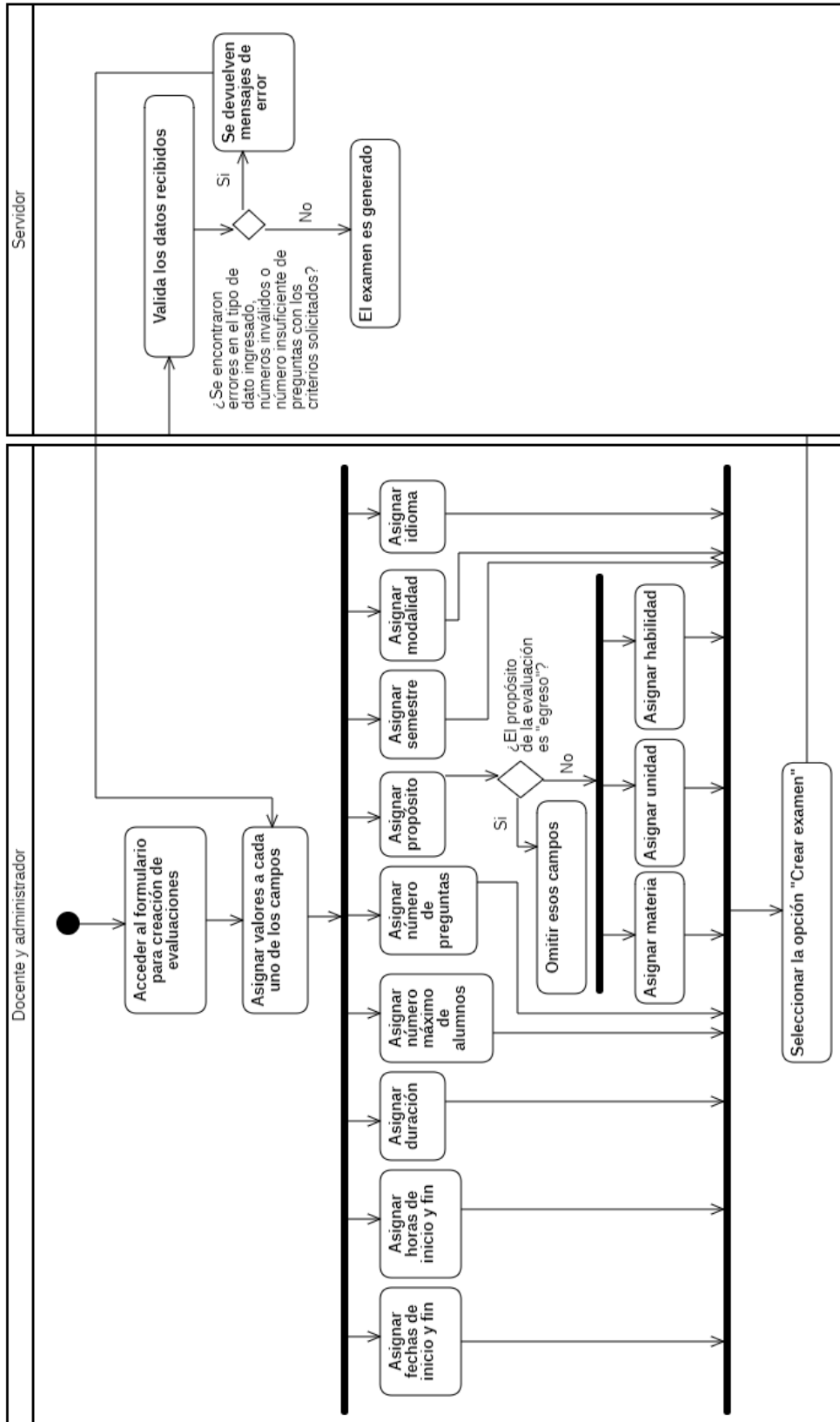


FIGURA 14. DIAGRAMA DE ACTIVIDADES PARA CREAR UN EXAMEN

3.3.2 Diagrama de actividades para visualizar preguntas asignadas a la evaluación

Las actividades involucradas en la visualización de las preguntas asignadas a un examen son bastante simples, estas son expresadas en la figura 15. Este proceso consiste en, sencillamente, acceder al listado de exámenes, encontrar el examen del cual se desea conocer las preguntas asociadas a este, y seleccionar la opción “Ver Preguntas”, y se mostrará al usuario el listado de preguntas. También, contendrá la opción *Ver detalles* al lado de cada pregunta para que así, el usuario pueda conocer información detallada de una pregunta en específico.

3.3.3 Diagrama de actividades para eliminar una evaluación

Al igual que otras de las actividades ya descritas, las que corresponden a la eliminación de un examen también son sencillas. El usuario deberá entrar al listado de exámenes y encontrar el examen que desea eliminar, posteriormente deberá seleccionar la opción de “Eliminar examen” y confirmar dicha eliminación. Así, el servidor tendrá conocimiento del examen que debe eliminar y le retornará al usuario el mismo listado de los exámenes generados actualizado, de modo que podrá corroborarse que el examen efectivamente ya no es visible en esta lista. Este proceso se puede ver en la figura 16. En caso de que el usuario niegue la confirmación de eliminación, dicho proceso será cancelado y el examen seguirá formando parte del conjunto de evaluaciones registradas en el sistema.

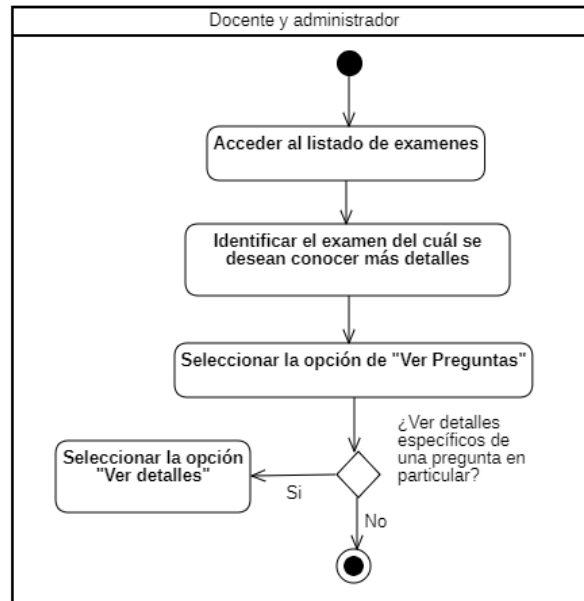


FIGURA 15. DIAGRAMA DE ACTIVIDADES PARA LA VISUALIZACIÓN DE PREGUNTAS ASIGNADAS A UN EXAMEN

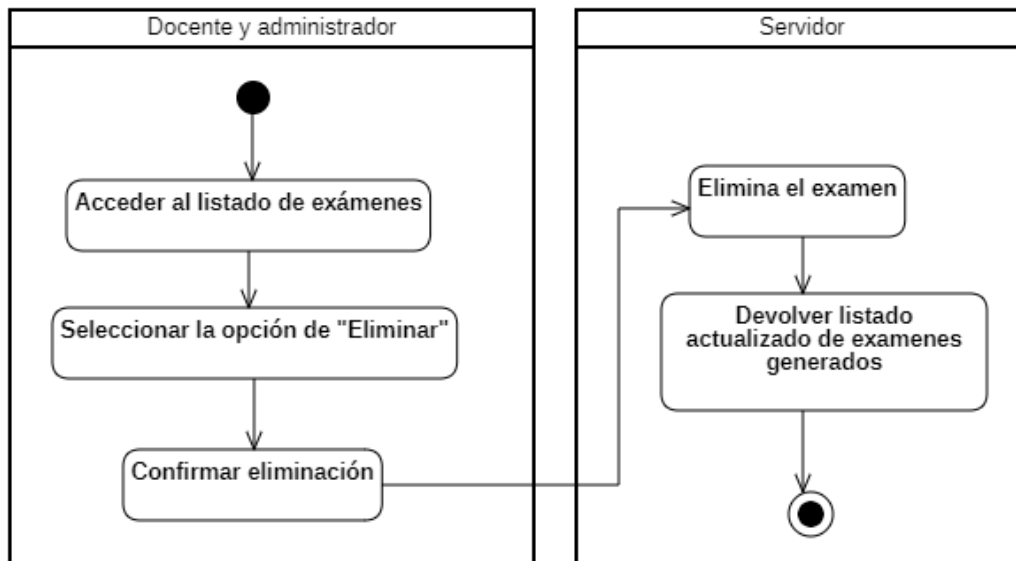


FIGURA 16. DIAGRAMA DE ACTIVIDADES PARA ELIMINAR UN EXAMEN

3.3.4 Diagrama de actividades para contestar una evaluación

El último diagrama de actividades realizado, mostrado en la figura 17, es el correspondiente al del proceso para contestar un examen. Como se dijo previamente, lo ideal es que sea el alumno quien realice esta actividad, sin embargo, por la naturaleza del proyecto, en este caso será realizado por cualquier usuario que esté interactuando con el sistema.

Este proceso consiste en que el usuario debe acceder al listado de exámenes y seleccionar el que será realizado. Posteriormente, el usuario deberá contestar cada uno de los cuestionamientos y, al término, enviarlo. Si el sistema identifica que aún hay preguntas sin responder, entonces el usuario deberá volver y responder aquellas preguntas. En caso contrario, el examen será enviado y calificado por el servidor, y este último también designará los temas sugeridos de estudio para el usuario. Finalmente, toda esta información, acompañada de otra relativa al examen y la retroalimentación, será devuelta por el servidor al usuario para que pueda leerla.

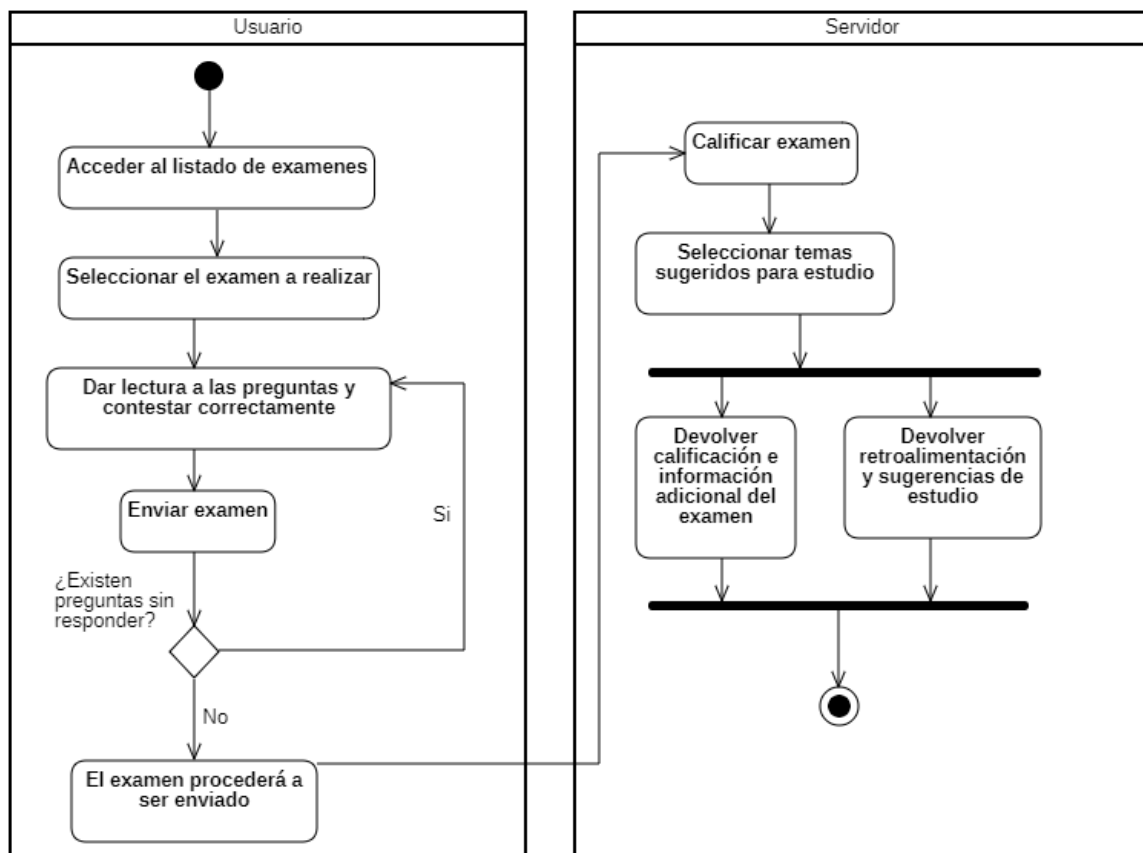


FIGURA 17. DIAGRAMA DE ACTIVIDADES PARA CONTESTAR UN EXAMEN

3.4 Diagrama de clases

El último tipo de diagrama UML elaborado es el de clases, cuyo propósito es mostrar las clases que utiliza el sistema y las relaciones entre ellas. El Sistema de proceso de evaluación de perfil lingüístico está conformado por un total de siete clases, tal como se describe en la figura 18.

Este tipo de diagrama utiliza diferentes símbolos, flechas y líneas que representan las relaciones existentes entre las clases. En la figura 18 pueden observarse varias de estas como (Miro, s.f.):

- Asociación unilateral, que expresa que una clase tiene conocimiento de la existencia de otra clase, y es representada con una línea recta y una flecha plana.
- Agregación, que es representada con una línea recta y un símbolo en forma de diamante hueco junto a la clase principal, y expresa que una clase secundaria es capaz de existir por sí sola sin depender de la clase principal.
- Composición, que expresa que una clase secundaria deja de existir en cuanto la clase principal es eliminada, y se representa con una línea recta y un símbolo de diamante lleno.
- Herencia, que se representa con una línea recta y un símbolo triangular que apunta a la clase independiente, y expresa que una clase hereda propiedades de otra.

Este diagrama presenta la clase *Student* (alumno), que describe a los alumnos a través de su nombre completo, número de cuenta y formas de contacto, y cuya única relación se da con la clase *Exam* (Examen), ya que los alumnos dan resolución a sus evaluaciones.

La clase *Evaluation* (Evaluación), que posee atributos que sirven para describir características específicas de las evaluaciones como asignatura a la que pertenece, su propósito, duración, habilidades que evalúe, etc.; es la superclase de la que otras clases heredan propiedades, y que serán denominadas como subclases. Estas subclases son *Exam*, que describe a la evaluación en términos de su aplicación; y *Report* (Reporte), que describe a los resultados generados a partir de las evaluaciones, y que utilizan varias propiedades de esa superclase para describir estos objetos.

A su vez, la clase *Exam* es una superclase de la subclase *Result* (Resultado), que describe los resultados obtenidos por el alumno al término de su examen. Por otra parte, la clase *Report* depende de la clase *Result*, ya que no es posible la existencia

3. CAPÍTULO III. DISEÑO Y DESARROLLO

de reportes sin resultados de los exámenes. De igual forma, la clase *UI* (*User Interface*, o Interfaz de Usuario), que describe a la interfaz que se genera cuando el examen será resuelto, no puede ser creada si no existen exámenes registrados en la base de datos que pueda estructurar a través de textos, botones de navegación, tiempo restante para contestar, etc.

Otra relación presentada en el diagrama de clases, es la de *Question* (Pregunta), que describe características de las preguntas como el texto de la pregunta, sus opciones, su respuesta, tipo, tipo de examen al que va dirigida, etc. Esa relación se da con la clase *Exam* que expresa que, a diferencia de otras clases que dependen estrictamente de otras para existir, la clase *Question* puede existir como parte de los exámenes, como también fuera de ellos mediante la consulta de la información propia de cada pregunta.

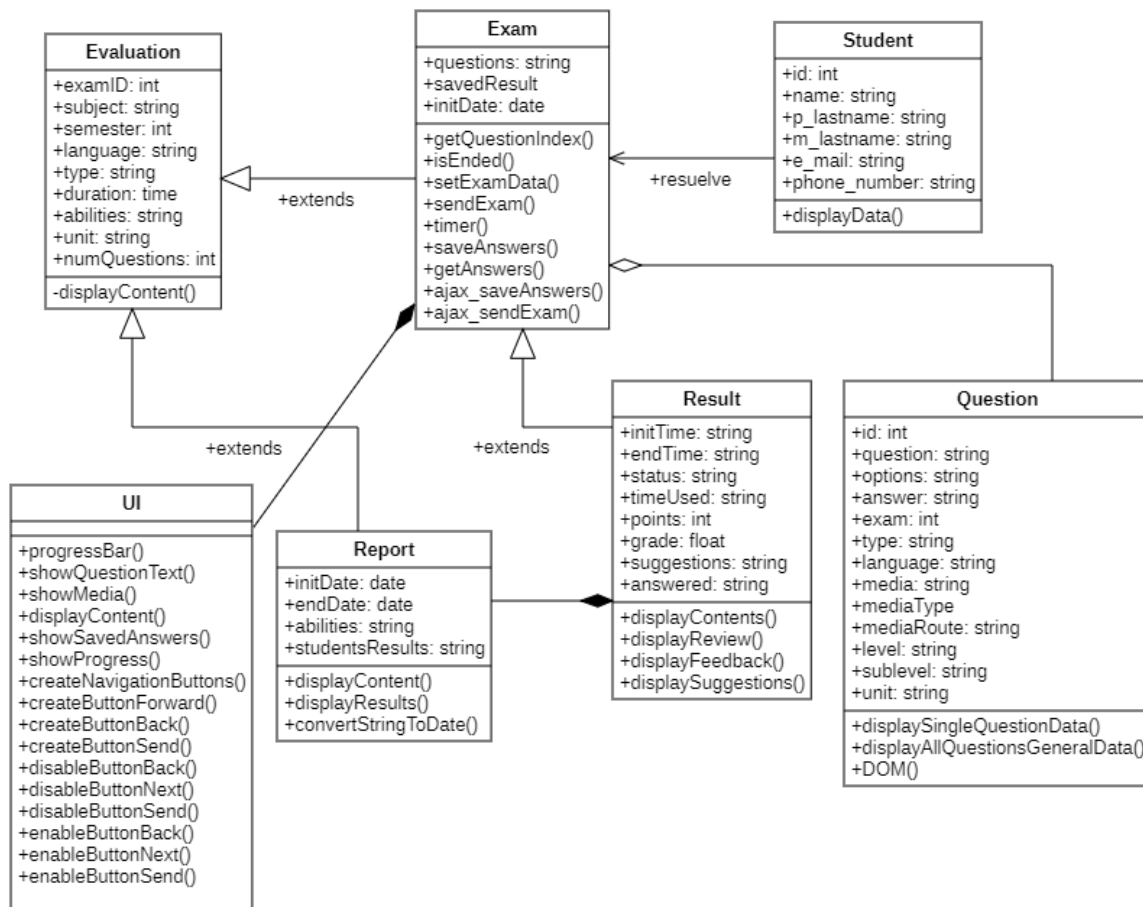


FIGURA 18. DIAGRAMA DE CLASES

3.5 Diseño arquitectónico

El patrón de diseño MVC (Modelo-Vista-Controlador) consiste en un patrón de diseño de software con el que es posible implementar interfaces de usuario, datos y lógica de control. Una característica distintiva de este patrón es la separación de la lógica de negocios de la visualización. Esto permite que el trabajo pueda ser dividido de forma óptima, y mejora su mantenimiento. MVC está conformado por tres partes: el modelo, encargado de controlar los datos y la lógica de negocios; la vista, que se enfoca en el diseño y presentación de la información; y el controlador, que controla a los modelos y a las vistas (MDN contributors, 2022). El diagrama de la figura 19 muestra como es el proceso de interacción entre estas tres partes.

El primer componente de MVC, el modelo, tiene la tarea de definir los datos que tendrá la aplicación. Para este sistema, se crearon los modelos para cada una de las tablas de la base de datos y en ellos se especificaron los campos que posee, a fin de poder realizar operaciones con los datos como, por ejemplo, consultas o inserciones de registros.

Las vistas, como ya se dijo, refieren a la forma en cómo los datos son presentados. En este sistema se construyeron múltiples vistas donde, cada una, muestra datos específicos a fin de mantener la información organizada.

Por último, el controlador, que es la parte encargada de controlar las interacciones del usuario con el sistema de modo que, cuando el usuario realiza una petición, el controlador deberá dirigirla al modelo correcto, realizar las operaciones solicitadas por el usuario y devolverá una respuesta a la vista para que pueda ser revisada. En este proyecto se acordó que cada miembro del equipo tendría su propio controlador donde escribiría todo el código relativo a las actividades asignadas, por lo tanto, en el caso de este módulo, solo se requirió de un controlador que llevaría una parte de la lógica de negocios exclusiva de este.

MVC se distingue por la separación de cada uno de los componentes que conforman la aplicación, cuestión que favorece una mejor organización de las actividades que realiza cada persona, pues cada controlador posee responsabilidades específicas, cada vista presenta determinada información, y cada modelo gestiona ciertos datos. Por ello, el trabajo realizado de forma colaborativa utilizando este patrón no se vio afectado de ninguna manera, ya que cada integrante escribía su código y le realizaba modificaciones sin que el resto se viera afectado.

Igualmente, se puede sacar el máximo provecho al código derivado de la capacidad de MVC para reutilizarlo, y su mantenimiento es mucho más sencillo debido a que

es menos complejo detectar problemas y los cambios no comprometen a otros componentes. También, MVC puede ser utilizado para el desarrollo web dando como resultado aplicaciones robustas y de calidad, que puedan ser escalables conforme crecen y cambian los requerimientos por parte del cliente. Por todo lo anterior, se seleccionó el patrón arquitectónico MVC para el desarrollo del Sistema de proceso de evaluación de perfil lingüístico.

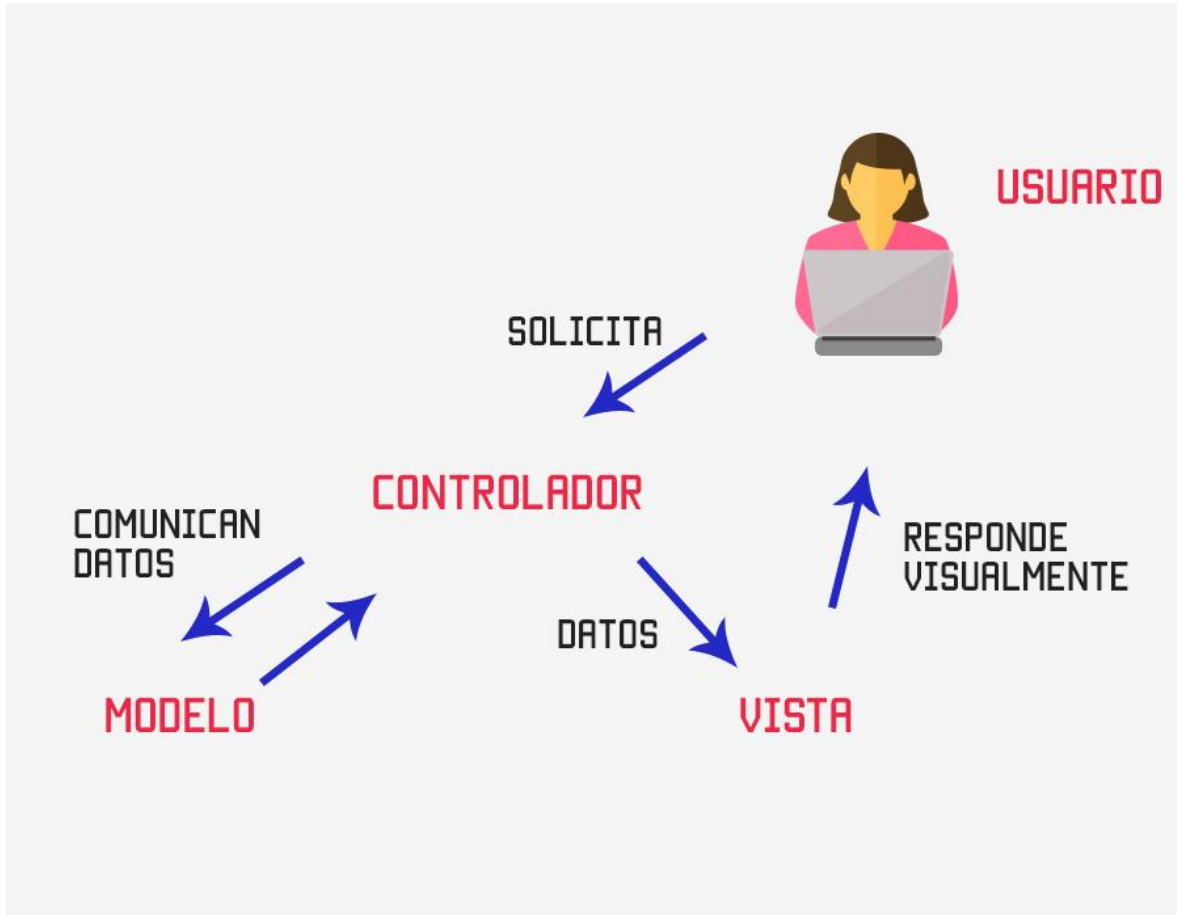


FIGURA 19. PATRÓN ARQUITECTÓNICO MVC

Fuente: (Hernández, 2015)

3.6 Diseño de la base de datos

Con el propósito de almacenar y preservar la información de este sistema, se creó una base de datos relacional que, por el patrón arquitectónico elegido, estará en constante comunicación con los modelos. El proceso de creación de la base de datos inició con su diseño mediante un modelo entidad-relación, seguido de un modelo relacional, los cuales serán presentados a continuación acompañados de una breve explicación de cada una de las tablas resultantes.

3.6.1 Modelo Entidad-Relación

El modelo entidad-relación es un diagrama con una estructura bastante parecida al de un diagrama de flujo, y que puede ser utilizado para el diseño de una base de datos (Lucidchart, 2023). Este modelo utiliza elementos muy básicos para representar los datos de una base de datos y sus relaciones (Juárez, 2020):

- El rectángulo, que representa a las entidades.
- El rombo, que representa las relaciones entre las entidades.
- La elipse, que representa los atributos de las entidades.

El modelo entidad-relación que será presentado a continuación incluye las tablas utilizadas en este módulo, junto con algunas que ya han sido consideradas para otros módulos. Por lo tanto, los módulos desarrollados en el futuro podrían requerir que la estructura de la base de datos actual sea modificada a fin de que pueda ser adaptada a ellos.

En las figuras 20 y 21 se presentan, en dos partes sintetizadas, el diagrama entidad-relación de la base de datos, esto con el propósito de preservar su seguridad y también, por el aspecto visual. Sin embargo, puede verse una vista más general de este modelo en el Apéndice C dentro de este documento. Con respecto a las entidades correspondientes a los símbolos con forma rectangular, se encuentran elementos que influyen dentro de las evaluaciones como son la estructura de la evaluación en sí, las preguntas que poseerá y sus respuestas, las recomendaciones que les serán dadas a los estudiantes, los elementos multimedia que las preguntas pueden poseer, etc.; también, se presentan catálogos que describen las características que pueden tener las evaluaciones como el caso del idioma, instituto de procedencia de los alumnos, materia, modalidad, etc. Para identificar estos catálogos, a las tablas correspondientes se les añadió el prefijo *cat_*

previo al nombre de la característica que describe, por ejemplo, *cat_idioma*, *cat_instituto*, *cat_materia*, etc. Con los rombos se pretende señalar la clase de relación existente entre estas entidades, y con las elipses describir las características o atributos propios de cada entidad. De estas relaciones entre entidades, puede concluirse lo siguiente:

- Varios exámenes pueden estar conformados por muchas preguntas.
- Para crear un examen, es necesario contar con los campos de las habilidades del examen a evaluar, su duración, su tipo o propósito, la materia a la que está destinado, el número de preguntas que tendrá, el idioma, el semestre y la unidad del programa temático a evaluar, además de contar con un número de identificación único. Y ese examen será aplicado en determinadas fecha y hora de inicio, y contará con fecha y hora de finalización, un número máximo de alumnos que pueden aplicar el examen y definirá una modalidad, además de que la aplicación contará con un número de identificación único.
- Un examen podrá producir múltiples resultados y contarán con un puntaje, un nivel MCER, se identificará la respuesta del alumno con su número de cuenta, el programa educativo al que pertenece, el número identificador del examen que realizó, se indicará el estado en el que se encuentra el examen, y cada resultado tendrá un número de identificación único.
- Cada elemento multimedia puede ser utilizado para una o más preguntas, pero una pregunta puede solo tener un solo elemento multimedia asociado.
- Cada una de las preguntas posee solo una respuesta.
- Un elemento multimedia puede ser únicamente de un solo tipo.
- Los alumnos pueden pertenecer a uno o más de un programa educativo dentro de la institución.
- Los diferentes programas educativos ofrecidos por la institución pueden ser impartidos en más de un instituto.
- Un examen puede generar muchos resultados, sin embargo, un alumno solo podrá generar un resultado de un examen en particular.
- Un examen puede pertenecer sólo a una materia, un tipo o propósito, poseer una sola habilidad e ir dirigido a un solo semestre.

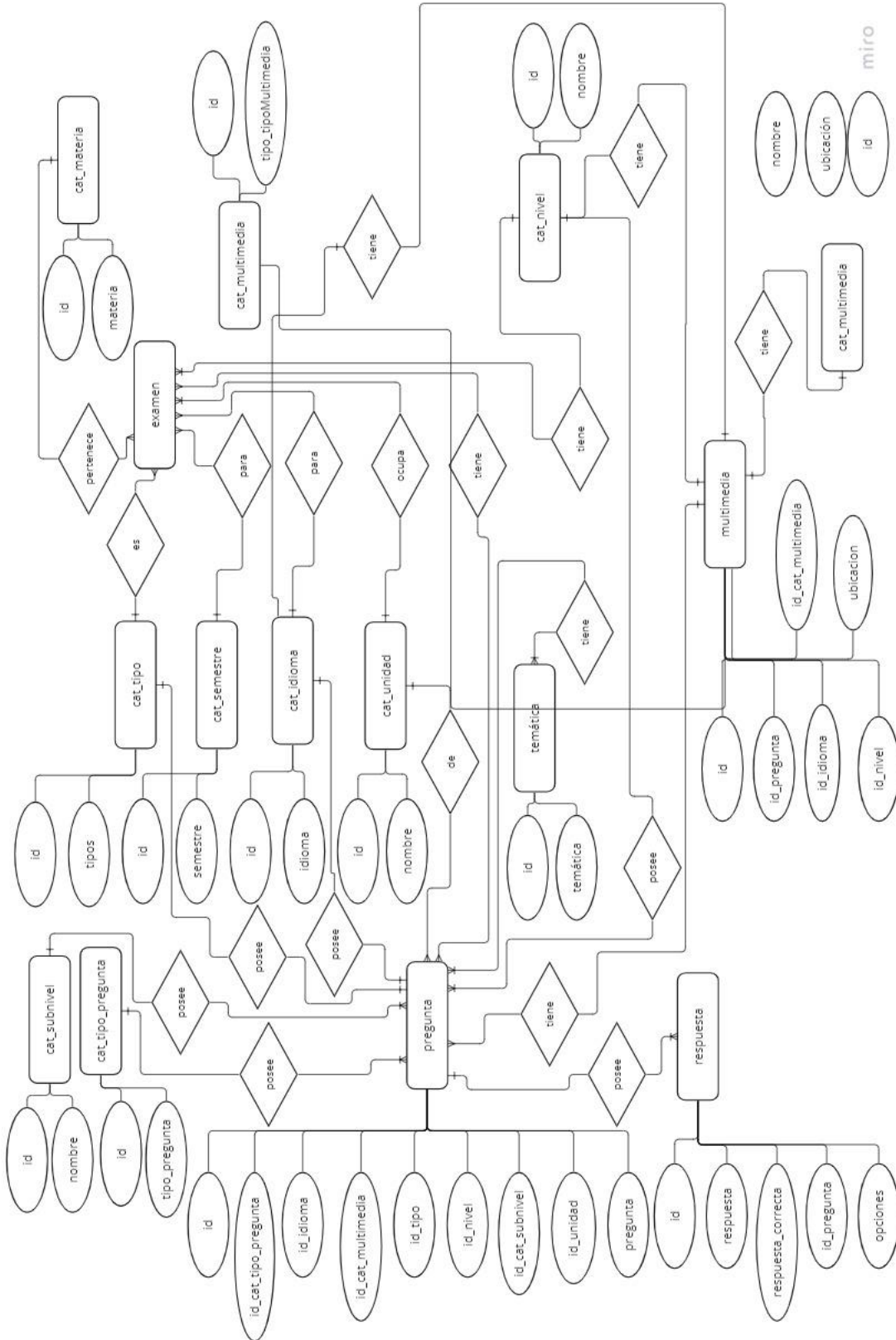


FIGURA 20. DIAGRAMA ENTIDAD-RELACION (A)

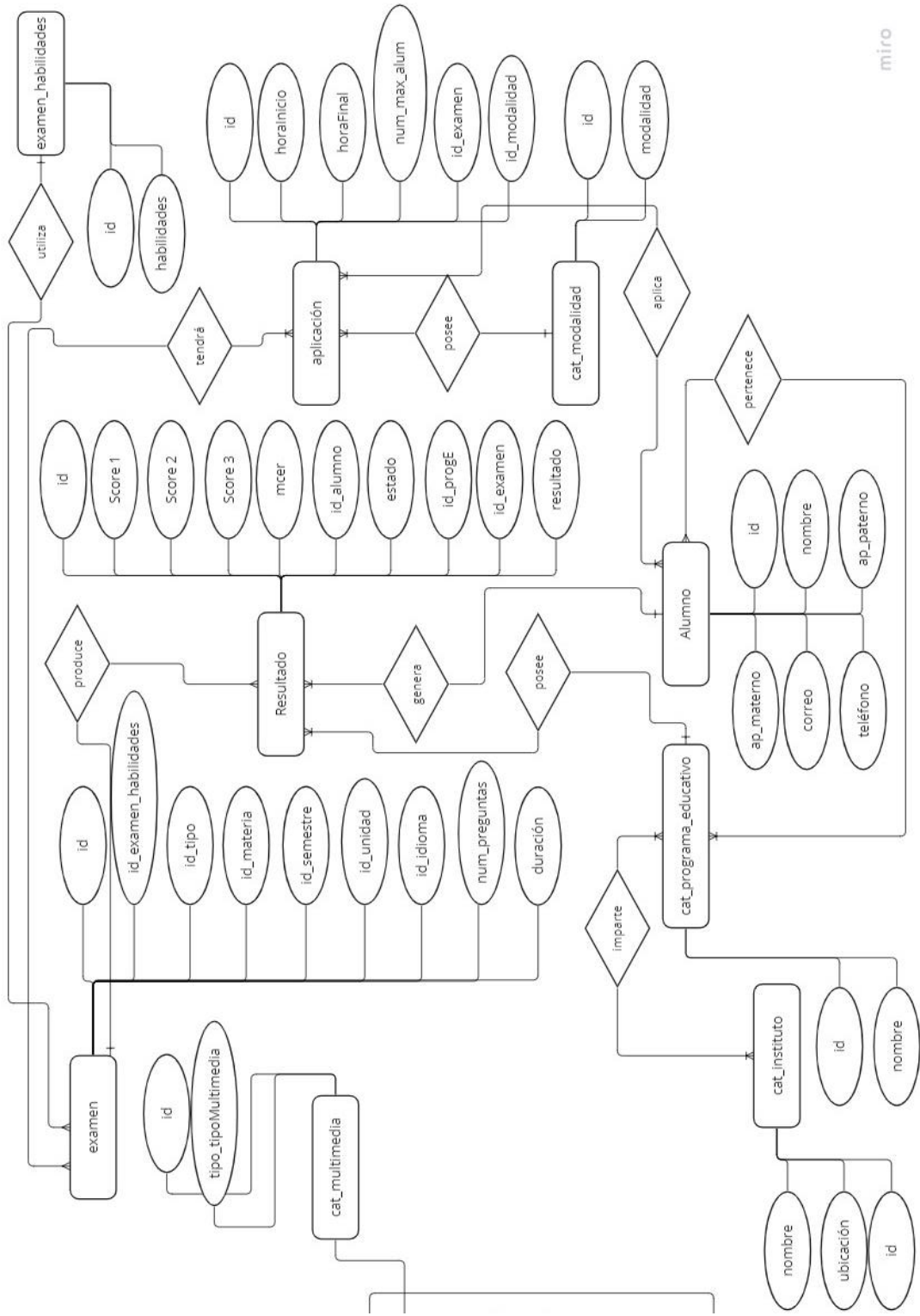


FIGURA 21. DIAGRAMA ENTIDAD-RELACIÓN (B)

3.6.2 Modelo relacional

Una vez se ha definido el modelo entidad-relación (E-R), el siguiente paso fue elaborar el diagrama relacional donde se ilustran las tablas que contendría la base de datos, y que proporciona una vista más cercana a su estructura. Este fue creado partiendo del resultado obtenido en el modelo E-R, tomando en cuenta cada una de las entidades, sus atributos y sus relaciones. Por lo tanto, respecto a las relaciones muchos a muchos (N:N) surgidas entre varias entidades, en este modelo se añaden sus respectivas tablas de unión. Igualmente, ya que este modelo derivó del modelo E-R, las conclusiones que pueden obtenerse de su análisis no son diferentes de las realizadas anteriormente.

Una versión segmentada de este modelo se muestra en la figura 22 debido a aspectos visuales y de seguridad, no obstante, una vista más general del modelo puede encontrarse en el Apéndice D. En cada una de las tablas se pueden identificar sus nombres, las columnas que poseen y algunas características de cada columna como son el tipo de dato que almacenará, las claves primarias y las claves foráneas.

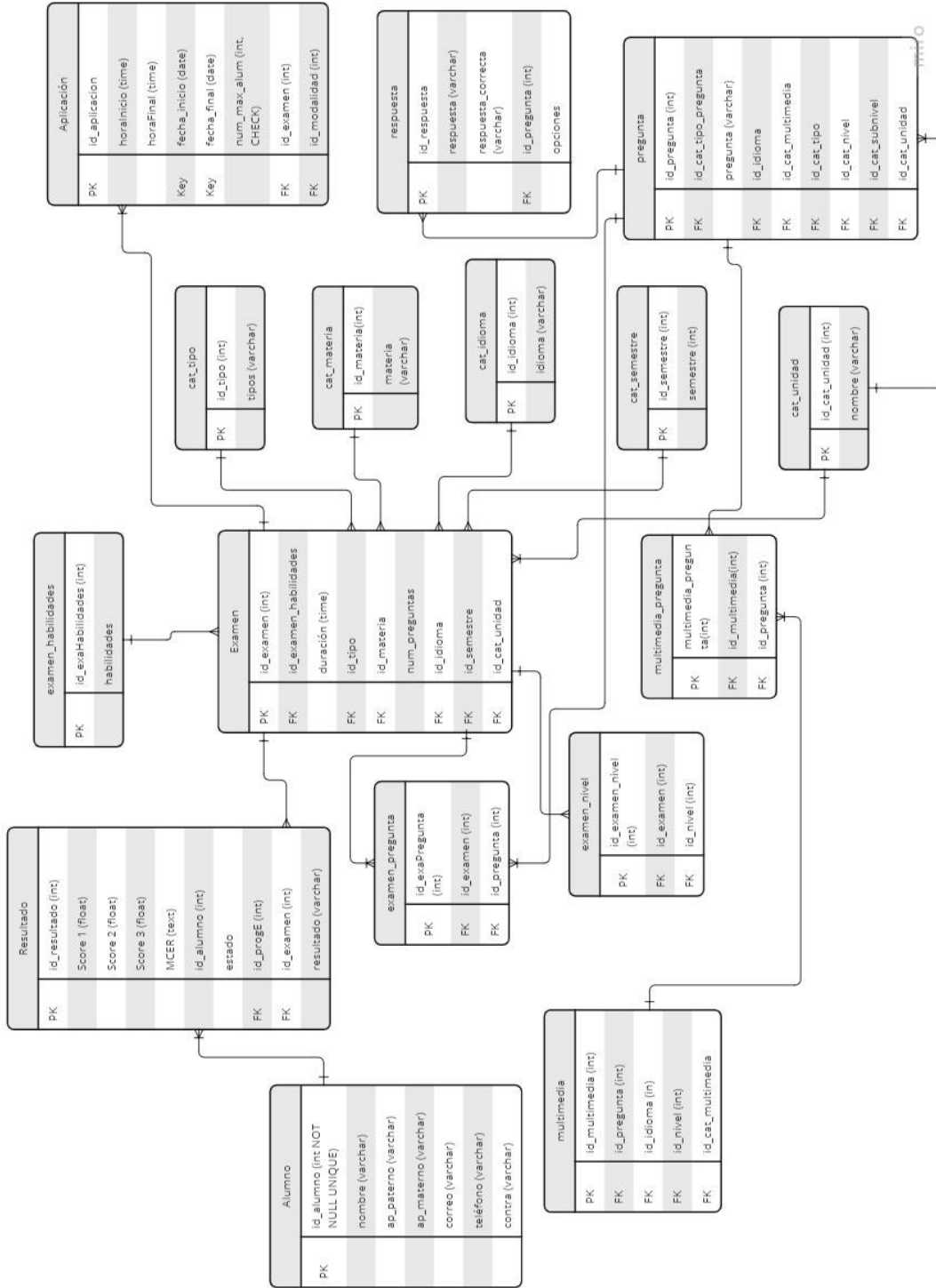


FIGURA 22. MODELO RELACIONAL

3.6.3 Descripción de la base de datos

En el presente proyecto, la base de datos tendrá varios propósitos, entre los que se encuentran el almacenar los resultados obtenidos por los estudiantes en sus evaluaciones, servir de contenedor de las preguntas, sus opciones y respuestas, almacenar la información de las evaluaciones creadas, guardar registro de los estudiantes, etc.

Tomando todo lo anterior como punto de partida, se realizó el análisis y diseño que dieron como resultado los dos diagramas obtenidos ya mostrados. Ese análisis llevó a crear diversas tablas y catálogos donde cada uno tendría una tarea específica a fin de que las actividades que realizará el sistema puedan ser efectuadas correctamente. Cada una de estas tablas se describen a continuación mediante breves descripciones de sus propósitos, sus atributos y una vista general de ellas mediante tablas.

3.6.3.1 alumno

Para contar con un registro de todos los estudiantes que aplicarán un examen se requerirá la implementación de una tabla que guarde la información de cada alumno, como son sus datos personales y de contacto. Estos últimos constan únicamente de su número de teléfono y su correo electrónico institucional. Es importante mencionar que esta tabla registrará exclusivamente a los estudiantes internos de la universidad. A continuación, se describen sus atributos:

- *id_alumno*. Número de cuenta del estudiante.
- *nombre*. Nombre(s) del alumno.
- *ap_paterno*. Apellido paterno del estudiante.
- *ap_materno*. Apellido materno del estudiante.
- *correo*. Correo electrónico institucional del estudiante.
- *teléfono*. Número telefónico del estudiante.
- *contra*. Contraseña de la cuenta del estudiante.

TABLA 2. ANÁLISIS DE LA TABLA PARA EL ALUMNO

alumno		
Índice	Campo	Tipo
PK	id_alumno	int
	nombre	varchar

	ap_paterno	varchar
	ap_materno	varchar
	correo	varchar
	teléfono	varchar
	contra	varchar

3.6.3.2 alumno_progE

Esta tabla surge de la relación que existe entre los estudiantes y los programas educativos de la universidad. Se considera que muchos estudiantes pueden cursar más de un programa educativo, y que un programa educativo puede ser estudiado por muchos estudiantes. Para establecer esta relación, se implementa la tabla *alumno_progE*. Esta tabla tendrá el objetivo de asignar un identificador único por cada programa educativo que curse un estudiante. A continuación, se describen sus atributos:

- *id_alumno_progE*. Identificador único de la unión entre un alumno y un programa educativo.
- *id_alumno*. Número de cuenta del estudiante.
- *id_progE*. Identificador del programa educativo.

TABLA 3. ANÁLISIS DE LA TABLA PARA LA RELACIÓN ENTRE EL ALUMNO Y EL PROGRAMA EDUCATIVO

alumno_progE		
Índice	Campo	Tipo
PK	id_alumno_progE	int
FK	id_alumno	int
FK	id_progE	Int

3.6.3.3 aplicación

Es importante llevar un control de cada aspecto que implica la aplicación de un examen, para ello se utilizará la tabla *aplicación* que se encargará de registrar los días y horas de inicio y término de la aplicación del examen, la modalidad en la que será aplicado, así como también un número máximo de estudiantes que pueden tomar ese examen. A continuación, se describen sus atributos:

- *id_aplicación*. Identificador de la aplicación.

- *horalInicio*. Hora en la que inicia la aplicación del examen.
- *horaFinal*. Hora en la que finaliza la aplicación del examen.
- *num_max_alum*. Número máximo de alumnos que pueden tomar el examen.
- *fecha_inicio*. Fecha en la que iniciará la aplicación del examen.
- *fecha_final*. Fecha en la que finalizará la aplicación del examen.
- *id_examen*. Identificador del examen.
- *id_modalidad*. Identificador de la modalidad en la que se aplicará el examen.

TABLA 4. ANÁLISIS DE LA TABLA PARA LOS DATOS DE APLICACIÓN DE LOS EXÁMENES

aplicación		
Índice	Campo	Tipo
PK	id_aplicación	int
	horalInicio	time
	horaFinal	time
	num_max_alum	int
	fecha_inicio	date
	fecha_final	date
FK	id_examen	int
FK	id_modalidad	int

3.6.3.4 cat_idioma

La UAEH imparte clases para el aprendizaje de varios idiomas, por lo tanto, esta tabla tendrá la tarea de almacenar todos los idiomas que la institución ofrece. A continuación, se describen sus atributos:

- *id_idioma*. Identificador del idioma.
- *idioma*. Nombre del idioma.

TABLA 5. ANÁLISIS DE LA TABLA PARA EL IDIOMA

cat_idioma		
Índice	Campo	Tipo
PK	id_idioma	int
	idioma	varchar

3.6.3.5 cat_instituto

La UAEH cuenta con muchos institutos, escuelas superiores, etc. en la que los alumnos cursan diversos programas educativos y que, al mismo tiempo, sirven de sede para la aplicación de los exámenes de idiomas. Para llevar registro de estas dependencias, se implementará la tabla *instituto* que contendrá la información referente a ellos como su nombre y su dirección. A continuación, se describen sus atributos:

- *id_instituto*. Identificador del instituto o plantel de la UAEH.
- *nombre*. Nombre del instituto o plantel de la UAEH.
- *ubicación*. Ubicación o dirección del instituto o plantel de la UAEH.

TABLA 6. ANÁLISIS DE LA TABLA PARA INSTITUTO

cat_instituto		
Índice	Campo	Tipo
PK	id_instituto	int
	nombre	varchar
	ubicación	varchar

3.6.3.6 cat_materia

La universidad, en muchos de sus programas educativos, ofrece las materias de lengua extranjera. La tabla *cat_materia* contendrá las materias o asignaturas de todos los programas educativos impartidos en todos los semestres que apliquen exámenes de lengua extranjera. A continuación, se describen sus atributos:

- *id_materia*. Identificador de la materia o asignatura.
- *materia*. Nombre de la materia o asignatura.

TABLA 7. ANÁLISIS DE LA TABLA PARA LA MATERIA O ASIGNATURA

cat_materia		
Índice	Campo	Tipo
PK	id_materia	int
	materia	varchar

3.6.3.7 cat_modalidad

Actualmente, los exámenes pueden impartirse de dos formas: de forma presencial en un instituto o dependencia de la universidad, o desde casa de manera virtual contando únicamente con una computadora y conexión a internet. Es por ello que los exámenes contarán con un parámetro que indique la modalidad en que será aplicado y, por lo tanto, será necesario la implementación de una tabla que incluya las posibles modalidades en las que se puede aplicar un examen, las cuales se resumen en modalidad presencial y modalidad virtual. A continuación, se describen sus atributos:

- *id_modalidad*. Identificador de la modalidad.
- *modalidad*. Nombre de la modalidad en la que se aplicará el examen.

TABLA 8. ANÁLISIS DE LA TABLA PARA LA MODALIDAD

cat_modalidad		
Índice	Campo	Tipo
PK	id_modalidad	int
	modalidad	varchar

3.6.3.8 cat_multimedia

Es usual que los exámenes de idiomas incluyan elementos multimedia que complementen las preguntas, ya sea mediante, por ejemplo, la visualización de dibujos o escuchar una conversación en un audio. De hecho, estos tipos de elementos pueden ser videos, audios e imágenes. Para identificar el tipo de elemento multimedia asociado a una pregunta, se utilizará la tabla *cat_multimedia*. A continuación, se describen sus atributos:

- *id_cat_multimedia*. Identificador del tipo de elemento multimedia.
- *dato_tipoMultimedia*. Nombre del tipo de elemento multimedia.

TABLA 9. ANÁLISIS DE LA TABLA PARA LOS TIPOS DE DATOS DE ELEMENTOS MULTIMEDIA

cat_multimedia		
Índice	Campo	Tipo
PK	id_cat_multimedia	int
	dato_tipoMultimedia	varchar

3.6.3.9 cat_nivel

La forma de clasificar el grado de conocimiento de un idioma se mide en niveles. Estos niveles indican en qué medida el estudiante posee conocimientos propios de un principiante, de un intermedio o bien, de un nivel avanzado. Estos niveles se guardarán en esta tabla. A continuación, se describen sus atributos:

- *id_nivel*. Identificador del nivel.
- *nombre*. Nombre del nivel.

TABLA 10. ANÁLISIS DE LA TABLA PARA EL NIVEL

cat_nivel		
Índice	Campo	Tipo
PK	id_nivel	int
	nombre	varchar

3.6.3.10 cat_programa_educativo

Como se ha mencionado, la UAEH, en varias de sus dependencias, imparte programas educativos que incluyen asignaturas de lengua extranjera. Para identificar el programa educativo al que pertenece el estudiante, se implementará una tabla encargada de guardar los datos referentes a los programas educativos de la universidad en los están inscritos. A continuación, se describen sus atributos:

- *id_progE*. Identificador del programa educativo.
- *nombre*. Nombre del programa educativo.

TABLA 11. ANÁLISIS DE LA TABLA PARA EL PROGRAMA EDUCATIVO

cat_programa_educativo		
Índice	Campo	Tipo
PK	id_progE	int
	nombre	varchar

3.6.3.11 cat_semestre

La función de este catálogo es sencillamente indicar el semestre al que irá dirigido el examen que se generó. A continuación, se describen sus atributos:

- *id_semestre*. Identificador del semestre.
- *semestre*. Número del semestre del programa educativo en el que está inscrito el estudiante.

TABLA 12. ANÁLISIS DE LA TABLA PARA EL SEMESTRE

cat_semestre		
Índice	Campo	Tipo
PK	id_semestre	int
	semestre	int

3.6.3.12 cat_subnivel

Además de los niveles, la universidad maneja subniveles en las asignaturas que imparte en sus programas educativos que indican con mayor precisión el nivel en el que los alumnos se encuentran en su aprendizaje de una lengua extranjera. Estos subniveles, al igual que los niveles, serán guardados en una tabla con el fin de que detalle aún más el nivel al que pertenece una pregunta. A continuación, se describen sus atributos:

- *id_cat_subnivel*. Identificador del subnivel.
- *nombre*. Nombre del subnivel.

TABLA 13. ANÁLISIS DE LA TABLA PARA EL SUBNIVEL

cat_subnivel		
Índice	Campo	Tipo
PK	id_cat_subnivel	int
	nombre	varchar

3.6.3.13 cat_tipo

Esta tabla se encargará de identificar el tipo o propósito del examen que presentarán los estudiantes. Hay una gran variedad de ellos entre los que se pueden citar el examen parcial, examen ordinario, examen extraordinario, examen para egreso, etc. A continuación, se describen sus atributos:

- *id_tipo*. Identificador del tipo de examen.
- *tipos*. Nombre del tipo o propósito del examen.

TABLA 14. ANÁLISIS DE LA TABLA PARA EL TIPO O PROPÓSITO DEL EXAMEN

cat_tipo		
Índice	Campo	Tipo
PK	id_tipo	int
	tipos	varchar

3.6.3.14 cat_tipo_pregunta

Hoy en día se encuentran exámenes en los que los estudiantes interactúan con ellos de diferentes formas a través de sus preguntas, ya sea mediante elegir una opción de entre varias posibles respuestas, selección de más de una opción, entradas de texto o listas desplegables.

Esta tabla se implementará para identificar de qué tipo es la pregunta que se añadirá al examen. A continuación, se describen sus atributos:

- *id_cat_tipo_pregunta*. Identificador del tipo de pregunta.
- *tipo_pregunta*. Nombre del tipo de la pregunta.

TABLA 15. ANÁLISIS DE LA TABLA PARA EL TIPO DE PREGUNTA

cat_tipo_pregunta		
Índice	Campo	Tipo
PK	id_cat_tipo_pregunta	int
	tipo_pregunta	varchar

3.6.3.15 cat_unidad

Los programas temáticos de los programas educativos ofertados por la UAEH suelen estar divididos en unidades y, para identificar la unidad exacta a la que pertenece una pregunta y a la que evaluará un examen, se utilizará esta tabla. A continuación, se describen sus atributos:

- *id_unidad*. Identificador de la unidad.
- *nombre*. Nombre de la unidad.

TABLA 16. ANÁLISIS DE LA TABLA PARA LA UNIDAD DEL PROGRAMA TEMÁTICO

cat_unidad		
Índice	Campo	Tipo
PK	id_unidad	int
	nombre	varchar

3.6.3.16 examen

Al ser una aplicación enfocada en la creación de exámenes, es natural que exista una tabla que se encargue de guardar la información relacionada a los exámenes, como la materia y semestre a los que irá dirigido, las habilidades que va a evaluar, las unidades del programa temático que incluirá, el número de preguntas que tendrá, etc. A continuación, se describen sus atributos:

- *id_examen*. Identificador del examen.
- *id_examen_habilidades*. Habilidades que evalúa el examen.
- *id_tipo*. Tipo o propósito del examen.
- *id_materia*. Materia para la cual está dirigido el examen.
- *id_semestre*. Semestre para el que va dirigido el examen.
- *id_unidad*. Unidad del programa temático que se evaluará en el examen.
- *id_idioma*. Idioma a evaluar.
- *num_preguntas*. Número de preguntas que contendrá el examen.
- *duración*. Tiempo que durará el examen.

TABLA 17. ANÁLISIS DE LA TABLA PARA EL EXAMEN

examen		
Índice	Campo	Tipo
PK	id_examen	int
FK	id_examen_habilidades	int
FK	id_tipo	int
FK	id_materia	int
FK	id_semestre	int
FK	id_unidad	int
FK	id_idioma	int
	num_preguntas	int
	duración	time

3.6.3.17 examen_habilidades

Los exámenes pueden evaluar diferentes habilidades, por ejemplo, puede existir un examen que evalúe la producción escrita, la comprensión escrita, la comprensión auditiva, uso del idioma, etc. La tabla *examen_habilidades* se utilizará para almacenar las posibles habilidades que incluye una evaluación. A continuación, se describen sus atributos:

- *id_examen_habilidades*. Identificador de las habilidades del examen.
- *habilidades*. Identificador de la habilidad a evaluar.

TABLA 18. ANÁLISIS DE TABLA PARA LAS HABILIDADES DEL EXAMEN

examen_habilidades		
Índice	Campo	Tipo
PK	id_examen_habilidades	int
FK	habilidades	varchar

3.6.3.18 examen_nivel

Esta tabla surge de la idea de que varios exámenes pueden evaluar más de un nivel, y que varios niveles pueden ser evaluados en varios exámenes. Para hacer esta relación se utilizará la tabla *examen_nivel* que especificará qué nivel o niveles estarán considerados en cada examen. A continuación, se describen sus atributos:

- *id_examen_nivel*. Identificador único de la unión entre un examen y el nivel al que corresponde.
- *id_examen*. Identificador del examen.
- *id_nivel*. Identificador del nivel.

TABLA 19. ANÁLISIS DE LA TABLA PARA LA RELACIÓN ENTRE EL EXAMEN Y SU NIVEL

examen_nivel		
Índice	Campo	Tipo
PK	id_examen_nivel	int
FK	id_examen	int
FK	id_nivel	int

3.6.3.19 examen_pregunta

Las preguntas constituyen la mayor parte de un examen. Para su construcción se considera que muchos exámenes pueden contener varias preguntas, y que varias preguntas pueden estar presentes en varios exámenes. Por ello, se utilizará esta tabla para identificar qué preguntas pertenecen a qué exámenes. A continuación, se describen sus atributos:

- *id_exaPregunta*. Identificador único de la unión del examen con la pregunta.
- *id_examen*. Identificador del examen.
- *id_pregunta*. Identificador de la pregunta que contendrá el examen.

TABLA 20. ANÁLISIS DE LA TABLA PARA LA RELACIÓN ENTRE EL EXAMEN Y LAS PREGUNTAS

examen_pregunta		
Índice	Campo	Tipo
PK	id_exaPregunta	int
FK	id_examen	int
FK	id_pregunta	int

3.6.3.20 instituto_prog

La universidad ofrece una muy amplia lista de programas educativos ya sea de preparatoria, licenciatura, posgrado, etc. que se imparten en diferentes planteles de la UAEH. Bajo esta premisa, se considera que varios programas educativos se estudian en varios institutos, y que varios institutos imparten más de un programa

educativo. Así, surge esta tabla que tendrá como objetivo indicar qué programas educativos se estudian en los distintos planteles que posee la universidad. A continuación, se describen sus atributos:

- *id_institutoProgE*. Identificador único de la unión entre un instituto y un programa educativo.
- *id_instituto*. Identificador del instituto o plantel.
- *id_progE*. Identificador del programa educativo.

TABLA 21. ANÁLISIS DE LA TABLA PARA LA RELACIÓN ENTRE INSTITUTO Y EL PROGRAMA EDUCATIVO

instituto_prog		
Índice	Campo	Tipo
PK	id_institutoProgE	int
FK	id_instituto	int
FK	id_progE	int

3.6.3.21 multimedia

Las preguntas, en muchas ocasiones, llevan consigo un elemento multimedia asociado que complementa la pregunta ya sea con un audio, una imagen e incluso un video. Esta tabla contendrá la lista de todos los elementos multimedia disponibles para estructurar los exámenes. A continuación, se describen sus atributos:

- *id_multimedia*. Identificador del elemento multimedia.
- *id_pregunta*. Identificador de la pregunta que utilizará el elemento multimedia.
- *id_idioma*. Identificador del idioma al que pertenece el elemento multimedia.
- *id_nivel*. Identificador del nivel al que pertenece el elemento multimedia.
- *id_cat_multimedia*. Identificador del tipo de elemento multimedia asociado.
- *ubicación*. Ruta de la ubicación del elemento multimedia.

TABLA 22. ANÁLISIS DE LA TABLA PARA LOS ELEMENTOS MULTIMEDIA

multimedia		
Índice	Campo	Tipo
PK	id_multimedia	int
FK	id_pregunta	int
FK	id_idioma	int
FK	id_nivel	int

FK	id_cat_multimedia	int
	ubicación	varchar

3.6.3.22 multimedia_pregunta

En las preguntas que utilizarán un elemento multimedia como complemento, sucede que muchas preguntas utilizan un mismo multimedia. Por ello, se utilizará la tabla *multimedia_pregunta* para establecer e indicar las relaciones entre cada elemento multimedia y cada pregunta a la que esté asociado. A continuación, se describen sus atributos:

- *id_multimedia_pregunta*. Identificador de la unión entre el elemento multimedia y la pregunta.
- *id_multimedia*. Identificador del elemento multimedia.
- *id_pregunta*. Identificador de la pregunta.

TABLA 23. ANÁLISIS DE LA TABLA PARA LA RELACIÓN ENTRE LOS ELEMENTOS MULTIMEDIA Y LA PREGUNTA

multimedia_pregunta		
Índice	Campo	Tipo
PK	id_multimedia_pregunta	int
FK	id_multimedia	int
FK	id_pregunta	int

3.6.3.23 pregunta

Las preguntas son la parte fundamental de un examen, pues con ellas se estructurará la mayor parte de él. Para llevar más orden de todas las posibles preguntas que se pueden agregar a un examen, se utilizará esta tabla como listado de dichas preguntas junto con otros atributos que permiten, por ejemplo, señalar el elemento multimedia que tiene asociado, su tipo o propósito, el idioma al que pertenece, el texto de la pregunta, entre otros. A continuación, se describen sus atributos:

- *id_pregunta*. Identificador de la pregunta.
- *id_cat_tipo_pregunta*. Identificador del tipo de la pregunta.
- *id_idioma*. Identificador del idioma al que pertenece la pregunta.
- *id_cat_multimedia*. Identificador del tipo de elemento multimedia asociado a la pregunta.

- *id_tipo*. Identificador del tipo o propósito del examen.
- *id_nivel*. Identificador del nivel al que pertenece la pregunta.
- *id_cat_subnivel*. Identificador del subnivel al que pertenece la pregunta.
- *id_unidad*. Identificador de la unidad del programa temático al que pertenece la pregunta.
- *pregunta*. Texto de la pregunta.

TABLA 24. ANÁLISIS DE LA TABLA PARA LAS PREGUNTAS

pregunta		
Índice	Campo	Tipo
PK	id_pregunta	int
FK	id_cat_tipo_pregunta	int
FK	id_idioma	int
FK	id_cat_multimedia	int
FK	id_tipo	int
FK	id_nivel	int
FK	id_cat_subnivel	int
FK	id_unidad	int
	pregunta	varchar

3.6.3.24 respuesta

Toda pregunta de un examen posee una respuesta. Esta tabla tendrá la tarea de almacenar las posibles respuestas a una determinada pregunta y su respectiva respuesta correcta (si aplica de acuerdo al tipo de pregunta). A continuación, se describen sus atributos:

- *id_respuesta*. Identificador de la respuesta.
- *respuesta*. Respuesta a la pregunta.
- *respuesta_correcta*. Indica el valor numérico de la respuesta correcta para cuando el examen sea calificado.
- *id_pregunta*. Identificador de la pregunta a la que corresponde la respuesta.
- *opciones*. Posibles opciones de respuesta que la pregunta tiene.

TABLA 25. ANÁLISIS DE LA TABLA PARA LAS RESPUESTAS

respuesta		
Índice	Campo	Tipo
PK	id_respuesta	int

	respuesta	varchar
	respuesta_correcta	int
FK	id_pregunta	int
	opciones	varchar

3.6.3.25 resultado

La salida que produce un examen es un resultado, por lo que será necesario crear una tabla que se encargue de recoger los resultados obtenidos por los estudiantes cuando finalicen su evaluación, para que los docentes y administradores puedan conocerlos. A continuación, se describen sus atributos:

- *id_resultado*. Identificador del resultado.
- *Score1*. Puntaje del examen.
- *Score2*. Puntaje del examen.
- *Score3*. Puntaje del examen.
- *mcer*. Marco de referencia europea.
- *id_alumno*. Número de cuenta del alumno.
- *estado*. Estado actual en el que se encuentra el examen.
- *id_progE*. Identificador del programa educativo que cursa el estudiante.
- *id_examen*. Identificador del examen.
- *resultado*. Resultado obtenido por el estudiante.

TABLA 26. ANÁLISIS DE LA TABLA PARA LOS RESULTADOS

resultado		
Índice	Campo	Tipo
PK	id_resultado	int
	score1	float
	score2	float
	score3	float
	mcer	varchar
FK	id_alumno	int
	estado	varchar
FK	id_progE	int
FK	id_examen	int
	resultado	varchar

3.6.3.26 temática

La tabla temática tiene como propósito almacenar los temas que le serán sugeridos al alumno al término de su evaluación por cada pregunta contestada de forma incorrecta. A continuación, se describen sus atributos:

- *id_temática*. Identificador de la temática.
- *temática*. Nombre de la temática.

TABLA 27. ANÁLISIS DE LA TABLA PARA LA TEMÁTICA

temática		
Índice	Campo	Tipo
PK	id_temática	int
	temática	varchar

3.6.3.27 temática_pregunta

Esta tabla surge de la idea de que una pregunta puede tener asociada a más de una temática, y que una temática puede estar asociada a más de una pregunta. Por ende, esta tabla llevará el registro de qué preguntas se relacionan con qué temáticas. A continuación, se describen sus atributos:

- *id_temática_pregunta*. Identificador de la unión entre una pregunta con su temática.
- *id_temática*. Identificador de la temática.
- *id_pregunta*. Identificador de la pregunta.

TABLA 28. ANÁLISIS DE LA TABLA PARA LA RELACIÓN ENTRE LA TEMÁTICA Y LA PREGUNTA

temática_pregunta		
Índice	Campo	Tipo
PK	id_temática_pregunta	int
FK	id_temática	int
FK	id_pregunta	int

3.6.3.28 usuario_aplicación

Para saber cuándo los alumnos aplicaron un examen, y saber qué alumnos estuvieron en una determinada aplicación de un examen, se utilizará esta tabla que mostrará ese aspecto. A continuación, se describen sus atributos:

- *id_usuario_aplicación*. Identificador de la unión entre un alumno y la aplicación de su evaluación.
- *id_alumno*. Identificador del alumno asociado a una aplicación de un examen.
- *id_aplicación*. Identificador de la aplicación de un examen para los estudiantes.

TABLA 29. ANÁLISIS DE LA TABLA PARA LA APLICACIÓN DE UN EXAMEN A UN ESTUDIANTE

usuario_aplicación		
Índice	Campo	Tipo
PK	id_usuario_aplicación	int
FK	id_alumno	int
FK	id_aplicación	int

3.7 Metodología empleada: SCRUM

Después del análisis, descrito en el capítulo anterior, de las posibles metodologías ágiles que pueden ser empleadas en el desarrollo del Sistema de proceso de evaluación de perfil lingüístico, entre los miembros que conforman el equipo de desarrollo involucrado en la creación del sistema, se optó por el uso de la metodología Scrum para su construcción. Las razones:

1. **Integrantes y trabajo en equipo.** Una característica de la metodología Scrum es su capacidad de adaptación a grupos de trabajo pequeños y, de hecho, el equipo de desarrollo para este sistema es bastante reducido, pero contiene a los miembros suficientes para cubrir los roles necesarios en esta metodología y así, evitar que el flujo de trabajo con Scrum se vea entorpecido. También, Scrum no maneja una jerarquía en sus miembros, cuestión que facilita la organización y asignación de actividades como se considere más conveniente.
2. **Sencillez.** Comparado con otras metodologías como las mencionadas en el capítulo II, el proceso de Scrum es muy sencillo de comprender y ejecutar aún para personas sin mucha experiencia en el uso de esta metodología. Así, a fin de evitar complicaciones y agilizar las actividades, la metodología seleccionada para el desarrollo del sistema es Scrum.
3. **Adaptación a los cambios.** Scrum, como muchas otras metodologías ágiles, tiene la capacidad de adaptación y tolerancia ante cambios siempre que estos no afecten en gran medida al producto final. Sin embargo, los ciclos de corta duración y las reuniones diarias rápidas de Scrum, facilitarán la identificación de posibles problemas en el sistema y solucionarlos lo más pronto posible y, también, permiten gestionar mejor los cambios que puedan surgir durante un Sprint.
4. **Entregas periódicas.** Una ventaja que provee Scrum es que el cliente tiene conocimiento de los avances del proyecto desde el primer incremento, de modo que puede conocer las funcionalidades o cambios añadidos en cada incremento, y poder así, entregar un producto que sea más fiel a los requerimientos expresados por el cliente.
5. **Agilidad.** Scrum permite al equipo de desarrollo gestionar las actividades por hacer y priorizar cada una de ellas. De ese modo y gracias a la tarea de planificación de dichas actividades, empezando por aquellas que sean más indispensables para la organización, se favorecerá el ahorro de tiempo y de recursos, pues como se ha dicho antes, las reuniones diarias y la entrega constante de incrementos al cliente ayudarán a detectar más rápido posibles fallos, administrar cambios o requerimientos nuevos.

3.7.1 Designación de roles

Scrum establece que, a cada uno de los miembros involucrados en el desarrollo del producto, le sea asignado un rol para coordinar las actividades que efectuarán. En este proyecto, la designación de roles se hizo de la siguiente manera: la persona que recibiría el rol de Scrum Master, o sea, la persona que serviría de guía durante el desarrollo del sistema, fue asignado a un miembro del equipo que estaría involucrado en el desarrollo del sistema y el mismo que desarrolló este módulo, mientras que el resto de integrantes tomarían el rol de miembros del equipo de desarrollo. El siguiente rol a designar fue el del cliente; el cual es el Centro de Lenguas de la UAEH, quien solicitó la creación de este proyecto. Por último, era necesario definir el rol de dueño del producto, que fue otorgado al director de este departamento de la UAEH.

3.7.2 Product Backlog

Para dar inicio al desarrollo de software implementando la metodología Scrum, el primer paso es la designación del *Product Backlog*, que consiste en la lista del trabajo por hacer, donde cada elemento listado debe estar ordenado por prioridad. La obtención de los requerimientos del sistema que conformaron este artefacto propio de Scrum, se dio mediante reuniones entre el personal del Centro de Lenguas (CeL) y el dueño del producto, donde daban a conocer las cualidades con las que el sistema debía contar, y se encontraban acompañados por el Scrum Master y el equipo de desarrollo quienes orientaban al personal sobre la viabilidad de cada característica, ya que este sistema estará funcionando en conjunto con otros sistemas del CeL, por lo que considerar su compatibilidad era indispensable para determinar el plan de trabajo y los recursos tecnológicos que serían utilizados para su desarrollo. Una vez que los requerimientos fueron consolidados, fueron priorizados para dar inicio a la siguiente etapa del ciclo Scrum.

3.7.3 Sprint Planning

Dentro de la etapa del *Sprint Planning*, las reuniones realizadas tenían el objetivo de determinar el trabajo que se haría a continuación. Para ello, era necesario que al inicio de cada ciclo *Sprint*, el dueño del producto entregara el *Product Backlog* con actualizaciones en la prioridad de cada elemento, para que el equipo pudiese seleccionar las características que serían trabajadas en la iteración actual. Independientemente de la cualidad del sistema que se fuera a trabajar, se abordaba también el cómo sería realizado. Por citar un ejemplo, del *Product Backlog* se determinó que era prioritaria la creación de la base de datos del sistema para que, conforme a ella, pudiera programarse la lógica del sistema; y sería realizado comenzando por su diseño a través del diagrama entidad-relación, posteriormente se diseñaría el modelo relacional, y finalmente se escribiría el script para su creación.

3.7.4 Sprint Backlog

Para que el equipo tuviese claro el alcance del trabajo a realizar y el producto esperado al final del *Sprint*, se realizaron breves descripciones similares a historias de usuario de cada uno de los elementos a trabajar seleccionados en el *Sprint Planning*. Estas descripciones también apoyaron al equipo a determinar los recursos tecnológicos necesarios para completar cada una de ellas. Retomando el ejemplo citado anteriormente, ya teniendo claro que el primer paso sería la creación del diagrama E-R, el equipo optó por emplear el tablero digital de Miro para su elaboración, de modo que cada integrante pudiera realizar aportaciones en el diseño en tiempo real; y la realización de esas breves descripciones de los elementos del *Product Backlog*, permitieron al equipo tener una idea más clara de las entidades que estarían involucradas en el modelo, así como sus atributos y su cardinalidad.

3.7.5 Sprint

Dentro de los Sprint, cuya duración fija de 2 semanas fue acordada por el equipo, se realizaban actividades para la elaboración de cada uno de los elementos del *Product Backlog* que incluiría el sistema. Estas actividades consistían principalmente en tareas de diseño y programación de las características del sistema, que eran decididas durante el *Sprint Planning*. También, en los *Sprints* se llevaban a cabo las reuniones diarias de Scrum, donde el equipo daba a conocer el estado del trabajo hecho, en proceso y por hacer.

Cuando el *Sprint* llegaba a su fin, las características del sistema elaboradas eran verificadas y sometidas a pruebas para confirmar que el trabajo pudiera ser considerado como Terminado, y recibirse retroalimentación por parte del dueño del producto una vez que lo evalúe, y poder así, conocer en qué grado de aceptación se encontraba el sistema, y recibir comentarios o sugerencias que serían considerados en el próximo *Sprint*.

Igualmente, al término del *Sprint*, los miembros del equipo de desarrollo realizaban una autorreflexión de la iteración concluida con el fin de considerar cómo sería la planeación de las futuras acciones en el próximo ciclo.

3.7.6 Daily Scrum

Como parte de cada *Sprint*, las reuniones Scrum, diarias dentro de la iteración, ayudaron al equipo a mantenerse al día de la situación en que se encontraba el trabajo que se estaba haciendo, y en ellas se encontraban siempre el equipo de desarrollo en compañía del Scrum Master y, siempre que estuviese disponible, se encontraba presente el dueño del producto. En estas reuniones, cuya duración estaba comprendida entre los 10 y 15 minutos, se abordaba qué trabajo ya se había realizado, qué trabajo estaba en proceso y cuál iba a ser realizado próximamente, y cómo se iba a llevar a cabo. Igualmente, los miembros del equipo discutían los problemas hallados para encontrar una solución en conjunto a ellos o, si era necesario, reasignación de actividades, cuestión que era muy común cuando el sistema se encontraba en la fase de programación de los controladores y la generación del script de la base de datos.

3.7.7 Trabajo Terminado

Cuando el ciclo *Sprint* llegaba a su fin, el equipo se encargaba de verificar que el producto obtenido haya sido el esperado. Para ello, se evaluaba si el producto, efectivamente, incluía todas las características que fueron seleccionadas durante la reunión del *Sprint Planning*, mediante la realización de múltiples pruebas que demostraran su funcionalidad en cada una de esas características. Una vez que el equipo decidía que, en efecto, el producto cumplía el objetivo esperado, se consideraba como Terminado y se encontraba en espera de que el dueño del producto pudiera evaluarlo también.

Como medida para que las características del sistema a trabajar se terminaran a tiempo, el equipo procuraba concluir las lo más pronto posible, y antes del fin del *Sprint* para tener tiempo de sobra para realizar correcciones en caso de que se encontraran en la fase de pruebas. Igualmente, se evitaba seleccionar más elementos del *Product Backlog* de los que el equipo pudiera hacerse cargo en el periodo de dos semanas que comprendía cada iteración. En las ocasiones que algunas características no estaban completadas a tiempo, el equipo las regresaba al *Product Backlog* y se retiraban del sistema que evaluaría el dueño del producto.

3.7.8 Retroalimentación del Sprint

En las reuniones donde se recibía la retroalimentación del producto entregado, se contaba con la presencia del Scrum Master, el equipo de desarrollo, algunos de los coordinadores de idiomas del Centro de Lenguas (usuarios finales) y el director del CeL o, en su caso, alguien que lo representase, quienes probaban el sistema y proporcionaban comentarios, sugerencias, y expresaban en qué grado consideraban que este producto satisfacía los requerimientos expresados. Además, solía suceder que también solicitaban modificaciones en características que ya habían sido elaboradas e incluso aceptadas. Estos cambios eran considerados para los siguientes ciclos *Sprint*, de modo que pudieran ser gestionados en conjunto con los siguientes elementos del *Product Backlog* a elaborar, y se evitara el entorpecimiento de la implementación de Scrum en general. Además, también se les orientaba sobre qué tan viable era añadir esos cambios o nuevas características al sistema.

3.7.9 Retrospectiva del Sprint

Cuando el ciclo Sprint llegaba a su fin y el dueño del producto ya había proporcionado su evaluación del producto entregado, el equipo realizaba una autorreflexión sobre las actividades hechas durante las dos semanas que llevaron a la conclusión del incremento del producto entregado. Aquí se analizaba el éxito de algunas actividades, y el por qué de aquellas que no lo fueron; con ello, se determinaba qué cambios debían efectuarse en el plan de trabajo para el siguiente *Sprint*, para que el flujo de actividades fuese más eficaz y eficiente, y el producto a entregar no experimentara retrasos significativos.

3.8 Herramientas empleadas

La elaboración de un sistema computacional hace uso de diversas herramientas tecnológicas según las características que este tendrá. Para la construcción de este proyecto de tesis fue esencial utilizar, en términos de software, múltiples lenguajes de programación, de consulta, de marcado, lenguajes de modelado para expresar las funcionalidades o estructura del sistema, etc.; y en términos de hardware, los dispositivos electrónicos donde este sistema fue desarrollado, como se explica a continuación.

Como punto de partida del desarrollo del sistema, la descripción de la estructura del sistema, sus relaciones con su entorno, funciones y demás, se expresaron haciendo uso de múltiples diagramas propios del lenguaje UML, los cuales fueron diseñados dentro del software StarUML.

Ya teniendo claro qué era lo que iba a elaborarse, se procedió con el diseño de los diagramas entidad-relación y relacional que servirían de guía para la creación de la base de datos. Estos diagramas fueron diseñados empleando la plataforma web Miro.

Una vez definidos los puntos anteriores, el siguiente paso fue diseñar y construir la base de datos que serviría para almacenar toda aquella información relativa a los exámenes. El lenguaje utilizado para realizar operaciones sobre esta información fue SQL dentro del entorno de MySQL Workbench.

Posteriormente, se seleccionó el framework Laravel para el desarrollo del sistema y, dentro de él, se escribió con lenguaje HTML el código destinado para las vistas, con código en JavaScript se dio dinamicidad al sistema, PHP para todas las operaciones relativas del servidor, con CSS se dio estilo a todas las páginas, y el formato JSON para realizar intercambio de datos dentro del sistema.

Para la escritura del código era imprescindible contar con un editor de código que fuera capaz de soportar los lenguajes ya mencionados. Visual Studio Code cuenta con soporte para multitud de lenguajes como los utilizados en este proyecto, y permite el uso de extensiones que facilitaron el trabajo de codificación.

El uso de XAMPP permitió el alojamiento del sistema en el servidor web de Apache y poder así, cargar componentes que conforman el sistema en general. Finalmente, para gestionar los cambios en el código, revisarlos y aceptarlos, se utilizó el sistema de control de versiones GitLab, lugar donde fueron almacenadas varias versiones del proyecto conforme se realizaban los incrementos. Para ejecutar las acciones requeridas para subir una versión al repositorio, se utilizó el entorno de Sublime

Merge. Asimismo, Sublime Merge permitió visualizar fácilmente los cambios realizados en cada uno de los archivos.

3.8.1 Herramientas tecnológicas

Las herramientas tecnológicas que auxiliaron en el diseño, la escritura y la gestión del código, así como aquellos programas que sirvieron para la realización de pruebas que confirmen la funcionalidad de cada característica desarrollada del sistema, que se decidieron emplear, se encuentran a continuación, acompañados de los motivos que llevaron a esa selección.

3.8.1.1 Visual Studio Code

Previo al inicio de la programación, era imprescindible contar con una herramienta digital donde escribir el código del sistema. El editor de código seleccionado para esta tarea fue Visual Studio Code. Una de las razones por las cuales este editor fue elegido, es porque era el más utilizado entre de los miembros del equipo involucrado en el desarrollo del sistema y, por lo tanto, ya se contaba con experiencia en su uso. También, este potente editor es capaz de soportar múltiples lenguajes de programación, entre los cuales se encuentran los que fueron necesarios para el desarrollo del sistema: HTML, CSS, JavaScript y PHP.

Otra característica de Visual Studio Code es el uso de extensiones, de las cuales se utilizaron algunas que facilitaron la escritura del código y que ayudaron a reducir la cantidad de errores en la sintaxis.

3.8.1.2 Sublime Merge

GitLab fue el sistema de control de versiones seleccionado para gestionar los cambios efectuados en la programación del sistema. Sin embargo, era necesaria una herramienta que permitiera visualizar estos cambios en el código al momento, así como también, que facilitara la ejecución de las acciones propias de los comandos Git para agregar el nuevo código al repositorio. Para cumplir esas tareas, se utilizó Sublime Merge.

El uso de Sublime Merge facilitó llevar a cabo el control de las versiones realizadas por los miembros del equipo. Este software muestra los cambios hechos por cada miembro exactamente en los archivos que fueron modificados, señalando el código eliminado (si fuera el caso) y el código nuevo añadido o modificado. También, Sublime Merge puede ejecutar comandos Git de una forma más simple, los cuales fueron utilizados para añadir las nuevas versiones al repositorio y descargar los nuevos cambios aceptados por todos los miembros del equipo. Igualmente, la gestión de las versiones resultó más simple gracias al sistema de ramificación que provee, pues en estas era más sencillo conocer los cambios como una versión separada de otras existentes, y poder así descartar o aceptar el código contenido evitando que el código incluido en la rama principal se alterara.

3.8.1.3 XAMPP

Con el propósito de verificar continuamente que el código del sistema funcionara de la manera esperada, así como para visualizarlo en el navegador, fue necesario contar con un servidor donde se pudiera implementar el sistema en el mismo computador. XAMPP es el servidor seleccionado para estas tareas.

XAMPP es un servidor de uso gratuito que incluye el servidor de Apache e intérpretes para los lenguajes de programación PHP y Perl. Aunque XAMPP incluye más herramientas útiles en el desarrollo de aplicaciones, su uso se enfatizó en montar un servidor en la computadora para que, una vez la carpeta contenedora del código del sistema estuviese cargada, pudiera ser ejecutada correctamente por cualquier navegador. Las razones por las que XAMPP fue seleccionado incluyen que no requiere el pago de licencia de ningún tipo, XAMPP contiene todas las herramientas necesarias para montar un servidor web, y también su instalación y uso general no son para nada complicados.

3.8.1.4 Star UML

Como se mencionó al principio, antes de comenzar a desarrollar un sistema, es importante iniciar por su diseño. El uso de diagramas propios de UML facilitó la descripción de su diseño y, gracias a que es un lenguaje estandarizado, pudo ser entendido por otros fácilmente.

Para la realización de diagramas UML, lo más recomendable es utilizar un software que provea de los elementos utilizados por este lenguaje para construir estos diagramas. Existen multitud de aplicaciones, programas y sitios en la web en los que se pueden desarrollar diagramas UML. Entre todos ellos, se encuentra StarUML. Este es un software especializado en la creación de diagramas y diseños basados en el estándar de UML. StarUML fue elegido porque, aunque es un software que tiene una versión de pago, también posee una versión gratuita que contiene todas las herramientas indispensables para la creación de varios tipos de diagramas UML. Además, StarUML también fue elegido debido a que ya se cuenta con experiencia en su uso, y los resultados obtenidos anteriormente con este software fueron satisfactorios.

3.8.1.5 GitLab

Durante el proceso de codificación del sistema, fueron surgiendo diversas versiones de este con correcciones o modificaciones significativas, y llevar control de estas versiones se hubiese complicado de no haberse hecho adecuadamente. Igualmente, cuando en un proyecto están trabajando varias personas o equipos simultáneamente, previo a añadir el nuevo código, se decide qué versión es la más idónea para ser añadida al proyecto y así evitar que las versiones de todos los demás miembros se combinen. A fin de prever estas situaciones y efectuar la gestión del código de forma más organizada, se utilizó el controlador de versiones GitLab.

Aunque existen diversos sistemas controladores de versiones, como GitHub, GitLab fue elegido por ofrecer ventajas que se acomodan mejor a las necesidades del proyecto como las siguientes:

- Las limitaciones que la versión gratuita de GitLab no son muchas y provee de las más indispensables para almacenar correctamente la información en el repositorio, por lo tanto, no será necesario utilizar la versión de pago.
- El proyecto, por el hecho de pertenecer a una institución que salvaguarda su información como lo es la UAEH, es importante que el proyecto posea cierto grado de privacidad. Para mantener protegido el código generado del sistema, GitLab permite la creación de repositorios privados.
- GitLab permite almacenar numerosas versiones del código del sistema sin restricciones de espacio.

3.8.1.6 Miro

Este módulo, aunque fue programado por una sola persona, en su planeación estuvieron involucrados algunos miembros del área de sistemas del Centro de Lenguas de la UAEH con quienes se determinó la estructura de la base de datos.

Como herramienta para trabajar de manera colaborativa y de forma remota para esta actividad, se empleó la plataforma Miro. Miro provee de elementos que pueden utilizarse para la creación de diagramas entidad-relación y relacionales. Gracias a ello, dentro de un tablero en esta plataforma, se crearon diagramas prototipo que expresan una idea general de qué tablas estarían involucradas junto sus relaciones y cardinalidad. Estos diagramas pudieron ser tomados como referencia en el momento de la escritura del script que serviría para la creación de la base de datos, lo cual redujo en buena medida la tendencia a errores o modificaciones, ya que ya se contaba con un diseño revisado y aceptado por los integrantes del equipo.

3.8.1.7 Pixabay

En las evaluaciones de idiomas es común que las preguntas se encuentren acompañadas por elementos multimedia como imágenes, videos y audios, que las complementan. Y aunque ya se contaba desde el inicio del proyecto con varios de esos elementos, fue necesario realizar la búsqueda de otros con los que no se contaban y que serían útiles para que la interrogante se entendiera mejor o que, de hecho, formaría parte de la misma.

El Centro de Lenguas solicitó que Pixabay, al ser una plataforma web de la que puede utilizarse su contenido multimedia sin la necesidad de proporcionar atribuciones, fuese de esa herramienta digital de dónde se obtendrían las imágenes que servirían para la realización de pruebas del presente sistema, y que acompañarían a varias de las preguntas que se integrarían con el fin de que los alumnos puedan comprender qué clase de respuesta es la esperada. Además, el uso de esta plataforma también se fundamenta en la muy amplia variedad de imágenes, fotos y vectores, de la cual se puede encontrar con precisión el contenido que se requiere.

3.8.2 Lenguajes de programación, modelado, y estilos

Los lenguajes de programación fueron seleccionados una vez que se tenían claros los objetivos y requerimientos de este sistema. Se emplearon varios de ellos para la programación del lado del cliente, algunos otros para obrar en el lado del servidor, uno más que serviría para el intercambio de información entre estos dos, y otro que auxiliaría en las acciones que serían ejercidas sobre la base de datos.

3.8.2.1 JavaScript

El lenguaje JavaScript fue uno de los más utilizados durante el proyecto y uno de los más importantes debido a que permitió efectuar muchas actividades fundamentales para el funcionamiento general del sistema. Con este lenguaje, se logró añadir más dinamicidad a todas y cada una de las páginas que conforman la plataforma web, característica que HTML no puede cubrir por sí solo. Gracias a esta dinamicidad, el sistema se convirtió en uno más agradable y amigable para el usuario, y permite que estos interactúen de diversas maneras con el sistema. También, el funcionamiento del examen fue desarrollado utilizando orientación a objetos, y JavaScript permitió establecer los modelos y métodos necesarios para ello. Igualmente, con este lenguaje se programaron las peticiones AJAX que servirían para intercambiar datos en formato JSON entre el cliente y el servidor.

3.8.2.2 JSON

Uno de los principales usos de la notación JSON es el intercambio de información entre aplicaciones o en una sola aplicación. Este último, es como JSON será utilizado en el caso de este sistema. Gracias a la estructura de su notación, JSON puede ser entendido fácilmente por multitud de lenguajes, entre los que competen aquí, PHP y JavaScript.

En el código del sistema, existen funciones que requieren el constante intercambio de datos entre el cliente y el servidor y viceversa. La notación JSON hizo esto posible gracias a la alta compatibilidad que este posee. Estas funciones consisten, con ayuda de AJAX, en el envío de datos como son las respuestas de los alumnos de sus exámenes, y la recepción de datos según la solicite el usuario como son la

información referente a los alumnos, a los exámenes, a las preguntas de los exámenes, etc.

3.8.2.3 HTML

El lenguaje de marcado de hipertexto HTML sirve para estructurar los elementos que contendrá una página web. Por ello, se hizo uso de este lenguaje para crear y organizar todos los elementos que conformarán cada página del sistema como, por ejemplo, títulos, párrafos, tablas, imágenes, etc. Aunque el sistema no posee ningún archivo con extensión .html, si existen archivos con extensión .blade.php que corresponden al motor de plantillas Blade integrado en Laravel, y estos incluyen código HTML, además de código PHP y sintaxis propia de Blade; por lo tanto, el uso de este lenguaje en el proyecto consistió en añadir algunos de los elementos más esenciales como los ya mencionados a varias de las páginas que conforman el sistema con HTML puro, pues también se hizo uso del DOM (Document Object Model) con JavaScript para añadir más elementos, eliminarlos o modificarlos.

Aunque existe software con características atractivas como facilitar la creación de sitios web generando las páginas de manera automática, proveer “mejor” gestión de ellos, entre otras herramientas que ahorran una buena cantidad tanto de trabajo manual como de tiempo, la razón de utilizar HTML frente a estos software se sustenta en que estas en muchas ocasiones no son compatibles en el momento de integrarlas con otros sistemas o, al ser plantillas, no se adaptan del modo esperado.

3.8.2.4 PHP

El uso del lenguaje PHP en este proyecto computacional permitió efectuar múltiples operaciones principalmente del lado del servidor. Con PHP y con ayuda del ORM Eloquent, se estableció la conexión con la base de datos para extraer información de ella y mostrarla al usuario en las vistas, así como también actualizar, eliminar o añadir registros. El envío y recepción de datos entre el cliente y el servidor fue programado utilizando este lenguaje y con ayuda de la notación JSON y AJAX. Igualmente, al ser este proyecto programado siguiendo los principios del patrón arquitectónico MVC (Modelo-Vista-Controlador), los modelos y las funciones contenidas en el controlador fueron escritos con PHP.

También, con PHP se realizó la validación de los datos que el usuario intenta enviar y que serían almacenados dentro de la base de datos, si la función encargada de esta validación encontrase algún problema con algún dato, lo haría saber al cliente mediante mensajes que serán visibles en varias de las vistas Blade.

3.8.2.5 CSS

La vista de los sitios web utilizando únicamente HTML sería totalmente básica, en blanco y negro, e incluso, probablemente poco entendible derivado de la organización simple que tendrían los elementos. Mientras que JavaScript se encarga de que la página web sea dinámica y enfatiza en la interacción con el usuario, CSS se encarga de dar estilo a los elementos HTML, por ejemplo, cambiando su disposición, márgenes, espaciado, alineación, color, etc. de modo que el sitio web sea más atractivo a la vista.

En el desarrollo de este sistema, CSS fue utilizado para organizar los elementos HTML a fin de que su lectura sea entendible y agradable a la vista, y también para dar color a varios otros elementos como son barras de progreso, botones, íconos, etc. El código CSS generado fue completamente escrito desde cero, utilizando los tres modos de empleo de CSS explicados en el marco tecnológico y, es importante mencionar, que durante su escritura no se hizo uso de ninguna librería de terceros, pues el código escrito en este lenguaje, por sí solo, fue más que suficiente para conseguir el diseño especificado por el Centro de Lenguas.

3.8.2.6 UML

Previo a la construcción del sistema, es sustancial expresar y definir los procesos del sistema, su estructura, interacciones, etc., mediante modelos que puedan ser entendidos por todos los miembros involucrados, desde los clientes hasta los desarrolladores. Con UML, mediante varios de sus diagramas fue posible dar un panorama de cómo fue elaborado el sistema, y sirvió de guía en el momento que el sistema fue programado. Por ejemplo, los diagramas de caso de uso permitieron representar funciones específicas del sistema, así como indicar quienes se ven implicados en ellas; los diagramas de actividades especifican los flujos de trabajo de una determinada tarea o de un componente del sistema; y el diagrama de clases expone cómo las clases del sistema interactúan entre sí, y muestra el flujo de la información. Cabe destacar que UML es principalmente utilizado en el desarrollo de

software orientado a objetos y, ya que una parte de la plataforma fue desarrollada mediante ese paradigma, UML resultó ser la mejor elección.

3.8.2.7 SQL

Una vez han sido definidos la estructura y diseño del sistema, será necesario contar con una base de datos donde pueda almacenarse información relativa a los exámenes como son las interrogantes que contendrá, elementos multimedia asociados, opciones de respuestas, respuestas, resultados obtenidos por los estudiantes, etc. Para su creación, administración, y acceso a la información, así como para su modificación en caso de ser necesario, debe contarse con un lenguaje que permita realizar estas actividades, y el lenguaje SQL tiene todo el potencial para efectuar estas tareas.

Para la creación de la base de datos, se escribió un script que contendría todas las sentencias necesarias para crear cada una de las tablas que la conformarían, especificando sus longitudes, tipos e incluso, en algunos casos, restricciones. Este script sería ejecutado en el entorno de MySQL Workbench. Igualmente, para realizar modificaciones a las tablas de la base de datos, esto se haría también desde el mismo entorno en un editor de texto proporcionado por el mismo. El resto de operaciones como lo son la inserción, eliminación y modificación de datos, no se efectuaron utilizando sentencias escritas en lenguaje SQL puro, pero si se utilizó Eloquent en Laravel, herramienta que permite manipular o consultar la información de una base de datos utilizando un lenguaje más entendible y parecido al nuestro.

3.8.3 Frameworks y librerías

Como apoyo para la programación del sistema, se consideró viable el desarrollo del sistema empleando un framework, como Laravel, que otorgara funciones que agilizaran esta actividad; y el empleo de la librería jQuery, que proporcionara código reutilizable para la creación de las funciones que obrarían del lado del cliente, manipulando el DOM de acuerdo a las acciones ejercidas por el usuario en el sistema.

3.8.3.1 Laravel

Laravel, que es un framework dedicado al desarrollo de aplicaciones web, permitió estructurar el sistema y organizarlo como MVC, patrón arquitectónico implementado por Laravel. Derivado de ello, con este framework se hizo la separación de datos en estructuras de datos (modelos), la interfaz de usuario (vista) y la lógica de la aplicación (controlador) favoreciendo el ahorro de tiempo en cargar cada página del sistema.

Laravel integra el motor de plantillas Blade, que utiliza archivos con extensión .blade.php capaces de contener el lenguaje HTML con el que se construyó la estructura de cada página web. En adición, gracias al lenguaje HTML empleado, se pudo enlazar con la etiqueta `<link>` a la hoja de estilos CSS. Igualmente, Blade permite el uso de comandos especiales que permiten codificar los datos enviados por el usuario en tokens a fin de dar mayor seguridad a esta información, mostrar mensajes de error por parte del validador si los existiera, y manipular los datos con el uso de estructuras y ciclos.

El ORM de Laravel, nombrado Eloquent, favoreció el ahorro de complejas sentencias SQL, pues Eloquent permite realizar diversas acciones sobre una base de datos con una sintaxis más sencilla. Con Eloquent, se realizó la extracción de información de la base de datos para que esta pueda ser enviada a las vistas y ser mostrada al usuario, y ayudó a la inserción de registros proporcionados por el usuario en formularios y en las evaluaciones de los estudiantes, así como actualizarlos o eliminarlos cada ocasión que fuera necesario.

La razón por la que Laravel fue seleccionado es debido a las características ya mencionadas, pues facilitaron en gran medida el proceso de programación del sistema, además de proporcionar algunas otras que le favorecen como la seguridad que provee, los comandos Artisan que permiten automatizar actividades que, para

el caso del proyecto, favoreció el ahorro de tiempo en la construcción de cada modelo y cada controlador, pues Artisan provee de una especie de plantilla de cada uno donde solo resta escribir el código correspondiente.

Finalmente, otra razón por la que fue elegido es debido a que varios de los miembros del equipo de desarrollo ya contaban con experiencia manejando este framework, lo cual agilizó el proceso de desarrollo del sistema.

3.8.3.2 jQuery

Para que la navegación a través de los exámenes, su presentación, el guardado de respuestas, su eliminación, etc. no se vieran afectados por múltiples recargas de las páginas web que conforman este módulo, se empleó jQuery para tener un mejor manejo de los elementos que le son presentados a los usuarios y controlar su apariencia y aparición en todo momento. También, fue posible obtener la información relativa al contenido de las evaluaciones y alumnos, y el guardado de las respuestas ingresadas empleando AJAX con jQuery, los cuales realizan las peticiones necesarias para que el controlador manipulara los controladores y así, obtener información o almacenarla dentro de la base de datos.

Igualmente, con jQuery y en conjunto con JavaScript puro, fue posible el manejo de eventos y la adición de dinamicidad a las páginas HTML.

4. Capítulo IV. Resultados

Una vez que se concluyó el proceso de diseño, comenzó el proceso de desarrollo del producto final. Para ello, se aplicó la metodología ágil para desarrollo de software Scrum. Como esta metodología lo exige, al término de cada *Sprint* se entregaba al cliente las características desarrolladas para que fueran inspeccionadas y se recibiera la retroalimentación pertinente a fin de considerar, si existiesen, los cambios para el siguiente ciclo *Sprint* en conjunto con las siguientes características a desarrollar.

En las siguientes páginas se describe el proceso llevado a cabo que encaminó la culminación de este módulo. Se muestra el conjunto de interfaces que lo conforman, así como sus respectivos elementos y cómo estos fueron desarrollados.

4.1 Desarrollo de la interfaz principal de la plataforma

Debido a la ausencia del módulo encargado de la gestión de los usuarios que controle la información que es presentada en función del tipo de usuario que ingresa a la plataforma, no existe una interfaz de inicio de sesión. Por lo tanto, la primera interfaz que se le mostrará al usuario será la del menú principal de opciones, que puede ser apreciada en la figura 23.

Este menú posee un diseño básico en donde se presentan tres opciones: *Ver alumnos*, *Crear nuevo examen* y *Exámenes creados*.

La opción *Ver alumnos* mostrará al usuario una nueva interfaz donde se muestra, en forma de tabla, información de los alumnos registrados en la base de datos de este sistema.

La opción de *Crear nuevo examen*, tal como su nombre sugiere, direcciona al usuario a un formulario en una nueva interfaz donde se le solicitan las características que tendrá el examen que será creado.

Finalmente, la opción de *Exámenes creados*, lleva al usuario una nueva interfaz donde se presenta al usuario el listado completo de los exámenes que han sido creados hasta el momento.



FIGURA 23. INTERFAZ PRINCIPAL DEL SISTEMA DE PROCESO DE EVALUACIÓN DE PERFIL LINGÜÍSTICO

4.2 Desarrollo de la interfaz para visualizar la lista de alumnos

Esta interfaz, que puede apreciarse en la figura 24, se implementó para que los docentes puedan visualizar el listado de alumnos que pueden presentar las evaluaciones creadas con este sistema. Por ahora, y como se ha mencionado, dado que no se cuenta con el módulo encargado de la gestión de usuarios, se mostrará el listado completo, y no las listas que le corresponderían a un docente o administrador en particular.

En esta interfaz se presenta una tabla con las columnas: ID alumno, que corresponde al número de cuenta del estudiante; Apellido paterno, Apellido materno y Nombre, que corresponden al nombre de los estudiantes; Correo, que muestra el correo institucional del estudiante; y Teléfono, que muestra el número telefónico del estudiante.

Por último, se añadió el botón llamado *Volver al menú* cuya función es la de proporcionar al usuario una forma de volver al menú principal de opciones.



FIGURA 24. INTERFAZ PARA VISUALIZAR EL LISTADO DE ALUMNOS

4.3 Desarrollo de la interfaz para crear un nuevo examen

Esta interfaz tiene como propósito recopilar la información necesaria por parte del usuario para que sea utilizada como referencia de las características que tendrá el nuevo examen. Esta recopilación se realiza a través de múltiples campos dentro de un formulario. Estos campos se dividen en:

FECHA Y HORA.

Esta sección, mostrada en la figura 25, recoge los datos relativos a la fecha y hora en que será aplicado el examen. Los campos que contiene son *Hora de inicio*, que indica la hora en la que el examen dará inicio; *Hora de finalización*, que indica la hora en que el examen concluirá; *Fecha de inicio*, que corresponde al día en que el examen dará inicio; *Fecha de finalización*, que corresponde al último día en que el examen estará disponible para ser resultado; y *Duración*, que indica la duración total que tendrá el examen.

Los campos utilizados para las horas de inicio y término utilizan un *input* de tipo *time*, el cual permite a los usuarios seleccionar con precisión y de modo intuitivo una determinada hora y minutos. Por otro lado, para los campos destinados para las fechas de inicio y término, se utilizó el tipo de campo *date*, el cual permite

4. CAPÍTULO IV. RESULTADOS

seleccionar una fecha dentro de lo que parece una miniatura de un calendario. Finalmente, para los campos que componen la característica de duración, se utilizaron tres *inputs* de tipo *number*. El motivo por el cual no se utilizó el *input* tipo *time* es porque este incluye la posibilidad de seleccionar el periodo del día (a.m. y p.m.), lo cual carecería de lógica ya que el dato que se solicita es una cantidad y no un horario.

FECHA Y HORA

Hora de inicio: --:--:--	Hora de finalización: --:--:--
Fecha de inicio: dd/mm/aaaa	Fecha de finalización: dd/mm/aaaa
Duración: : : Horas Minutos Segundos	

FIGURA 25. PRIMERA SECCIÓN DEL FORMULARIO PARA CREAR UN NUEVO EXAMEN

ALUMNO.

Esta sección, mostrada en la figura 26, tiene el único propósito de recabar el número máximo de alumnos que pueden aplicar este examen. Este campo se creó utilizando un *input* de tipo *number* ya que el valor esperado es numérico.

ALUMNO

Número máximo de alumnos:
Ej: 35

FIGURA 26. SEGUNDA SECCIÓN DEL FORMULARIO PARA CREAR UN NUEVO EXAMEN

ESPECIFICACIONES.

Esta sección presentada en la figura 27, posee los campos relativos a características generales del examen. Estos son: *Número de preguntas*, que indica el número total de preguntas que contendrá el examen; *Tipo/Propósito*, que indica para qué fines será el examen; *Habilidad a evaluar*, que indica aquellas habilidades del lenguaje que evaluará el examen; *Modalidad*, que especifica si el examen será aplicado de forma presencial o virtual; y *Semestre*, que indica el semestre para el que el examen está dirigido.

4. CAPÍTULO IV. RESULTADOS

El valor esperado para el número de preguntas es numérico, por lo tanto, se utilizó un *input* tipo *number* para obtenerlo. En cuanto al resto de campos pertenecientes a esta sección, se utilizaron listas desplegables que contienen los posibles valores que pueden tener.

ESPECIFICACIONES	Número de preguntas: Ej: 20	Tipo / Propósito: Elegir Propósito
	Habilidad a evaluar: Elegir Habilidad	Modalidad: Elegir Modalidad
	Semestre: Elegir Semestre	

FIGURA 27. TERCERA SECCIÓN DEL FORMULARIO PARA CREAR UN NUEVO EXAMEN

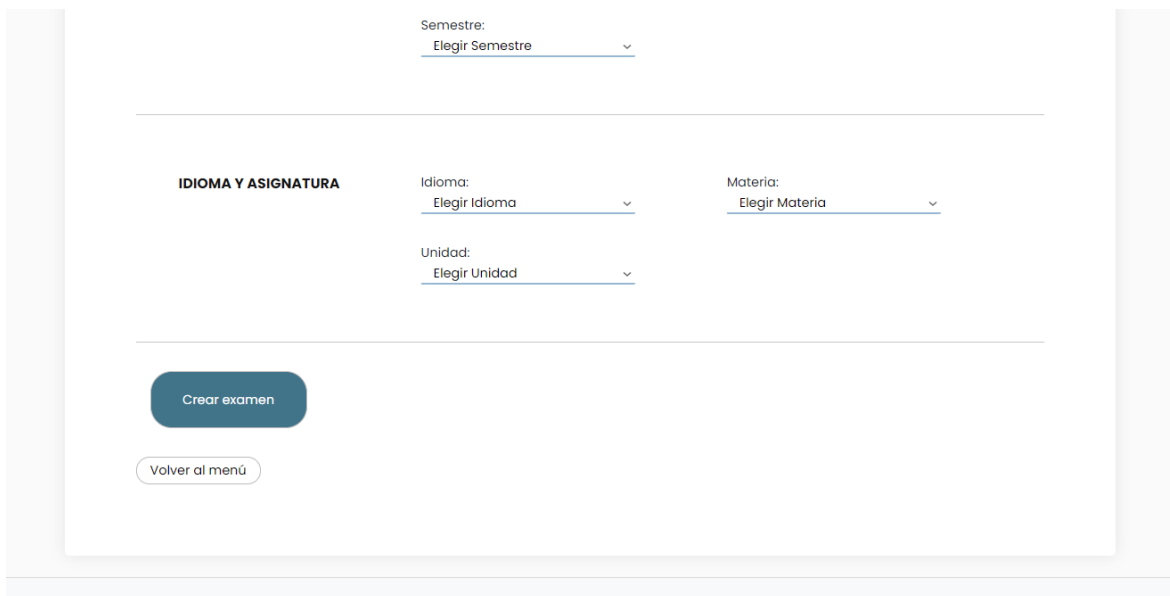
IDIOMA Y ASIGNATURA.

Este campo, mostrado en la figura 28, se encarga de recopilar características del examen en términos del idioma y aspectos del programa temático. Posee únicamente tres campos que son: *Idioma*, el cual contiene únicamente la opción de “alemán” debido a que son los alumnos de este idioma el objeto de estudio; *Materia*, que especifica la materia en particular que será evaluada; y *Unidad*, que especifica la unidad del programa temático de la materia que será evaluada. Para todos estos campos, se utilizaron listas desplegables donde las opciones corresponden a los posibles valores que pueden tener.

IDIOMA Y ASIGNATURA	Idioma: Elegir Idioma	Materia: Elegir Materia
	Unidad: Elegir Unidad	

FIGURA 28. CUARTA SECCIÓN DEL FORMULARIO PARA CREAR UN NUEVO EXAMEN

Al final del formulario, que puede observarse en la figura 29, se encuentra el botón llamado *Crear examen*, el cual enviará los datos al controlador para que éste pueda manipular al modelo y almacenar la información en la base de datos. Con el objetivo de prever que se introduzcan datos inválidos, se añadió la validación de datos tanto del lado del cliente como del lado del servidor.



Semestre:
Elegir Semestre

IDIOMA Y ASIGNATURA

Idioma:
Elegir Idioma

Materia:
Elegir Materia

Unidad:
Elegir Unidad

Crear examen

Volver al menú

FIGURA 29. FINAL DE LA INTERFAZ PARA CREAR UN NUEVO EXAMEN

El validador del lado del servidor puede mostrar mensajes de error por dos motivos: ya sea porque se escribieron valores inválidos o se dejaron campos vacíos, o porque el número de preguntas solicitado es insuficiente. Para el primer caso, si el validador encuentra campos vacíos o con valores inválidos, devolverá una alerta al usuario indicando lo anterior y solicitará al usuario que los complete correctamente. Esta alerta va acompañada de mensajes de error que se encontrarán debajo de todos aquellos campos vacíos o que contengan valores inválidos, solicitando al usuario que ingrese valores correctos.

Para el segundo caso, es necesario explicar que cada una de las preguntas almacenadas en la base de datos pueden ser utilizadas únicamente para ciertos tipos de exámenes. Esto significa que existen preguntas exclusivas para exámenes parciales, para exámenes extraordinarios, para exámenes de egreso, etc. Por lo tanto, si el usuario solicita la creación de un examen para cierto propósito con una determinada cantidad de preguntas, y si la cantidad de preguntas contenidas en la base de datos es menor a la solicitada con esos criterios, esa situación le será notificada al usuario acompañada del número de preguntas encontradas con esas características a fin de que pueda hacer una mejor selección de esa cantidad, en lugar de intentar a ciegas con diferentes números de preguntas.

En la figura 30 puede observarse un ejemplo de cómo se visualizan los mensajes de error cuando se identifica que hay campos vacíos, acompañados de una alerta que lo indica.



FECHA Y HORA

Hora de inicio:

--:-- --:--



Ingrese una hora válida

Hora de finalización:

--:-- --:--



Ingrese una hora válida

FIGURA 30. EJEMPLO 1 DE MENSAJES DE ERROR

El lado del cliente muestra mensajes de error que aparecen en el momento en que el usuario escribe un valor inválido, y desaparece cuando el campo se vuelve a encontrar vacío o cuando se ingresa un valor válido. Estos mensajes son mostrados cuando el usuario ingresa valores decimales, negativos o fuera de los límites establecidos. En la figura 31 se muestra un ejemplo de ello cuando se introducen valores inválidos en los campos de Horas, Minutos y Segundos correspondientes a la duración del examen.

Duración:

-2 : 60 : 0.7

Horas

Minutos

Segundos



No es posible introducir un valor negativo, decimal o mayor a 59 en el campo Horas



No es posible introducir un valor negativo, decimal o mayor a 59 en el campo Minutos



No es posible introducir un valor negativo, decimal o mayor a 59 en el campo Segundos

FIGURA 31. EJEMPLO 2 DE MENSAJES DE ERROR

En caso de que los valores introducidos sean correctos y el número de preguntas solicitadas sea suficiente, se solicitará la confirmación de creación del examen al usuario. Si la confirmación es afirmativa, el examen será creado y se redireccionará al usuario a la interfaz del listado de exámenes creados.

Por último, se muestra el botón llamado *Volver al menú*, que dirige al usuario al menú principal.

4.4 Desarrollo de la interfaz para visualizar el listado de exámenes

Esta interfaz, presentada en la figura 32, tiene el propósito de mostrar en forma de lista todos los exámenes que el usuario ha creado hasta el momento, junto con información propia de estos para su fácil y pronta identificación.

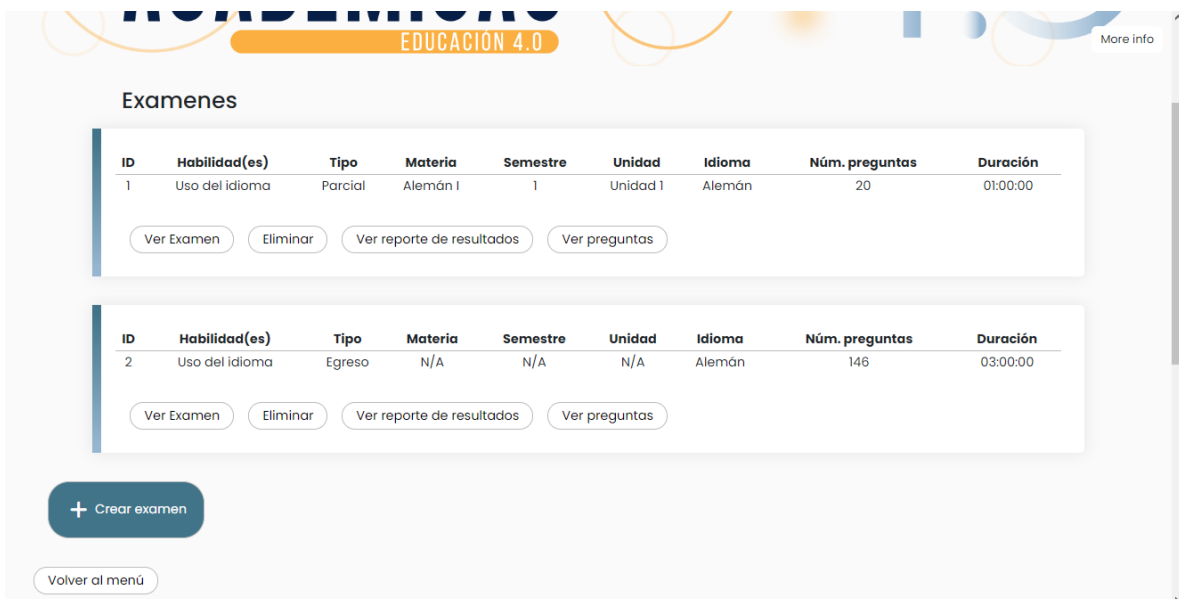


FIGURA 32. DISEÑO DE LA INTERFAZ PARA VISUALIZAR LISTADO DE EXÁMENES CREADOS

Cada elemento de la lista se muestra con sus propios datos y botones. Estos datos se presentan en forma de tabla y son:

- *ID*, que representa el número de identificador único del examen.
- *Habilidades*, que especifica las habilidades que serán evaluadas.
- *Tipo*, que indica el tipo o propósito del examen.
- *Materia*, que indica la materia para la que fue creado el examen.

- *Semestre*, que indica el semestre al que va dirigido el examen.
- *Unidad*, que indica la unidad de programa temático que evalúa.
- *Idioma*, que indica el idioma que evalúa.
- *Núm. de preguntas*, que expresa en forma numérica la cantidad de preguntas que incluye el examen.
- *Duración*, que marca el tiempo total disponible para resolver el examen.

Algunos de estos datos pueden aparecer con el valor *N/A* (no aplica) para algunas características de algunos tipos de exámenes. En el ejemplo de la figura 32, los valores de materia, semestre y unidad no aplican debido a que, para un examen de egreso, para ese momento, los alumnos ya no pertenecerán a un semestre en particular, por lo que no se encontrarán cursando una asignatura, y ya no se evalúa una sola unidad, sino que se toma en cuenta todo el conocimiento que se espera haya adquirido durante su programa educativo.

Los botones que incluye cada examen son: *Ver Examen*, para visualizar el examen de la misma forma que lo haría el alumno; *Eliminar*, para eliminar ese examen; *Ver reporte de resultados*, para ver los resultados obtenidos por los estudiantes en ese examen; y *Ver preguntas*, que muestra el listado de preguntas que fueron asignadas a ese examen de forma aleatoria en el momento de su creación.

De todos los botones, el de *Eliminar* es el que realiza acciones visibles en esta interfaz, pues el resto tienen la función de llevar al usuario a otras interfaces. Cuando el botón de *Eliminar* es presionado, se muestra un mensaje de confirmación donde se le pregunta al usuario si realmente desea eliminar ese examen, esto con el propósito de evitar que el examen sea eliminado por accidente. En caso de que el usuario confirme dicha eliminación, el sistema procederá a suprimir toda información relativa a ese examen dentro de la base de datos, incluyendo los resultados registrados por los alumnos de esa evaluación. Una vez ha sido eliminado, el examen será retirado del listado sin necesidad de actualizar la página. En caso de que no exista ningún examen registrado, la interfaz mostrará un mensaje que indique lo anterior acompañado de una invitación a crear uno nuevo, como se muestra en la figura 33.

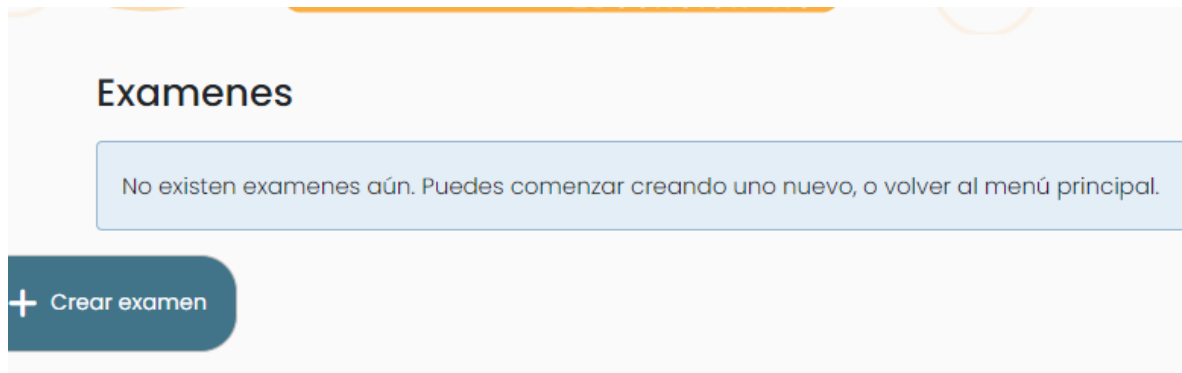


FIGURA 33. MENSAJE DEL SISTEMA QUE INDICA QUE NO EXISTEN REGISTROS DE EXÁMENES

También se añadió el botón de *Crear examen* para que el usuario pueda ir a la interfaz de creación de exámenes sin necesidad de retroceder hasta el menú principal y, al igual que en otras interfaces, también se añadió el botón de *Volver al menú* que llevará al usuario al menú principal de esta plataforma.

4.5 Desarrollo de la interfaz para visualizar las preguntas asignadas a un examen

Esta interfaz fue implementada para tener la tarea de mostrar toda la información referente a qué es lo que se va a preguntar en un examen. Para acceder a este listado es necesario ir al listado de exámenes, ubicar el examen del cual se desean conocer las preguntas asignadas y presionar el botón *Ver preguntas*.

ID pregunta	Pregunta	Opciones	Respuesta	Acciones
21	Was siehst auf dem Foto? Ich sehe...	A) das Papier, B) den Brief, C) die E-Mail	C) die E-Mail	Ver detalles
23	Was siehst auf dem Foto? Ich sehe...	A) eine Adresse, B) Briefe, C) eine Bank	B) Briefe	Ver detalles
24	Was siehst auf dem Foto? Ich sehe...	A) ein Geburtsjahr, B) einen Vornamen, C) eine Unterschrift	C) eine Unterschrift	Ver detalles
25	Was siehst auf dem Foto? Ich sehe...	A) die Post, B) die Polizei, C) die Bank	B) die Polizei	Ver detalles
32	Ergänzen Sie das passende Wort. die Adresse - die Straße - der Absender - das Land - die Biefmarke - die Stadt - die Hausnummer - der Empfänger - die Postleitzahl.			Ver detalles
33	Ergänzen Sie das passende Wort. die Adresse - die Straße - der Absender - das Land - die Biefmarke - die Stadt - die Hausnummer - der Empfänger - die Postleitzahl.			Ver detalles
37	In der Adresse fehlt die	A) ausfüllen, B) überweisen, C) Schalter, D)	n) Postleitzahl	Ver

FIGURA 34. DISEÑO DE LA INTERFAZ PARA VER LAS PREGUNTAS ASIGNADAS A UN EXAMEN

La figura 34 muestra un ejemplo de listado de todas las preguntas que posee un examen, y puede notarse que la información dada por cada una es la siguiente:

- *ID pregunta*, que indica el número de identificador único de la pregunta.
- *Pregunta*, que expone el texto completo de la pregunta.
- *Opciones*, que muestra las posibles respuestas a la pregunta (exceptuando a las preguntas de tipo abierta).
- *Respuesta*, que indica la respuesta correcta (exceptuando a las preguntas de tipo abierta).
- *Acciones*, donde se coloca un botón llamado *Ver detalles*.

El motivo del por qué no se muestran las opciones ni respuesta correcta de las preguntas de tipo abierta, es porque se considera que sea el docente o evaluador quien tome la decisión de si la respuesta recibida es correcta o no, ya que los alumnos pueden responder de forma correcta a una interrogante de diferentes formas de acuerdo a su criterio y conocimientos personales. Además, es importante mencionar que las preguntas que se visualizan a lo largo de todo el desarrollo de la plataforma, fueron proporcionadas por el Centro de Lenguas, así como otros elementos multimedia como videos, imágenes y audios con los que se realizaron pruebas durante su programación.

La acción que ejercen los enlaces de *Ver detalles* es la de dirigir al usuario a otra interfaz donde se le presentará información adicional de la pregunta. Esto será descrito a mayor detalle en la sección correspondiente a esa interfaz.

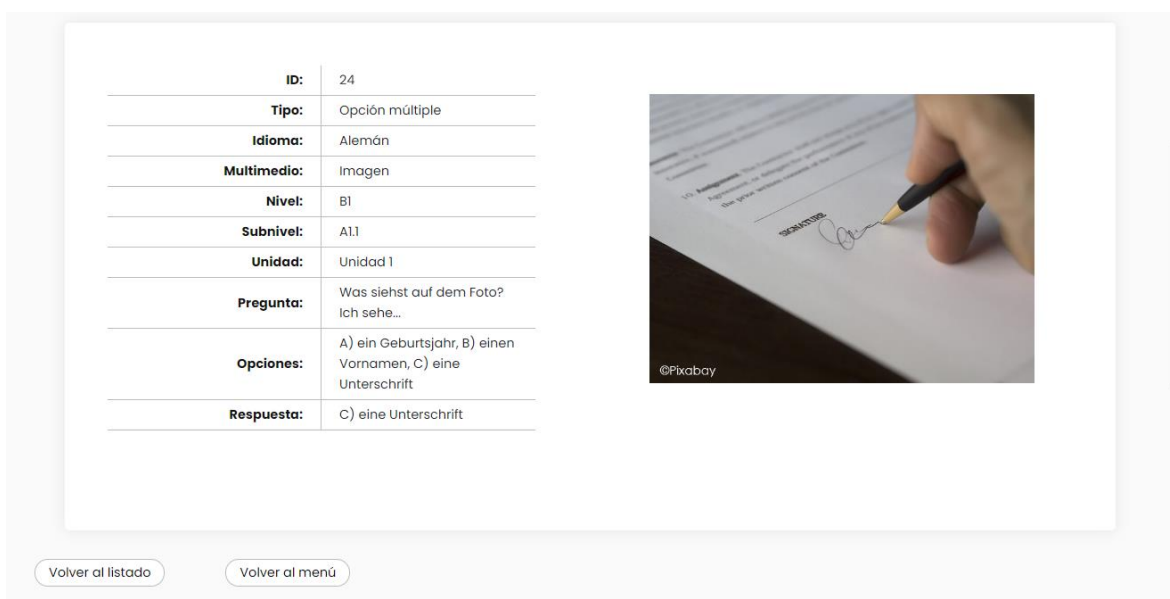
Como en otras interfaces, también se añade el botón de *Volver al menú* para volver al menú principal de la plataforma, pero esta vez acompañado del botón *Volver a Exámenes*, que direcciona al usuario al listado de exámenes a fin de facilitar al usuario una forma de continuar viendo la información de cada evaluación con un menor número de clics.

4.6 Desarrollo de la interfaz para visualizar detalles acerca de una pregunta en particular

En caso de que se desee conocer más detalles de una pregunta en específico, la plataforma podrá facilitar al usuario esta información presentada en esta interfaz. En la figura 35 se observa que los datos son presentados en forma de tabla. Muchos de estos datos que aparecen en el listado general de preguntas asignadas a un examen también aparecen en esta interfaz, pero con otros adicionales:

4. CAPÍTULO IV. RESULTADOS

- *Tipo*, que indica el tipo o propósito del examen.
- *Idioma*, que indica el idioma de la pregunta.
- *Multimedia*, que indica si la pregunta posee algún elemento multimedia asociado y de qué tipo.
- *Nivel y Subnivel*, que indican los niveles y subniveles de la pregunta de acuerdo al marco MCER.
- *Unidad*, que indica la unidad del programa temático al que pertenece.



ID:	24
Tipo:	Opción múltiple
Idioma:	Alemán
Multimedia:	Imagen
Nivel:	B1
Subnivel:	A1.1
Unidad:	Unidad 1
Pregunta:	Was siehst auf dem Foto? Ich sehe...
Opciones:	A) ein Geburtsjahr, B) einen Vornamen, C) eine Unterschrift
Respuesta:	C) eine Unterschrift

©Fixabay

Volver al listado Volver al menú

FIGURA 35. DISEÑO DE LA INTERFAZ PARA VISUALIZAR DETALLES ACERCA DE UNA PREGUNTA EN ESPECÍFICO

Solo cuando una pregunta posea un elemento multimedia asociado, será presentado del lado derecho de la interfaz. Si se trata de un video o de un audio, el usuario podrá repetirlo una cantidad de veces indefinida. En la figura 35 puede observarse que la pregunta está asociada a un multimedia de tipo imagen, y se encuentra colocada a un costado de la tabla. Además, al igual que las preguntas, varios de los elementos multimedia utilizados en el desarrollo de este proyecto, fueron también proporcionados por el Centro de Lenguas desde el inicio.

Al final, se muestra el botón de *Volver a listado*, que facilita al usuario el volver al listado de las preguntas de ese mismo examen; y también se añade el botón de *Volver al menú* con el que puede dirigirse al menú principal de la plataforma.

4.7 Desarrollo de la interfaz para visualizar un examen

Para visualizar un examen, es necesario dirigirse al listado de exámenes, ubicar el examen que se desea visualizar, y hacer clic en el botón *Ver Examen*. El sistema dirigirá al usuario hasta la interfaz donde se muestra el examen del modo en cómo lo verán los estudiantes. Ver Anexo 1 en el documento.

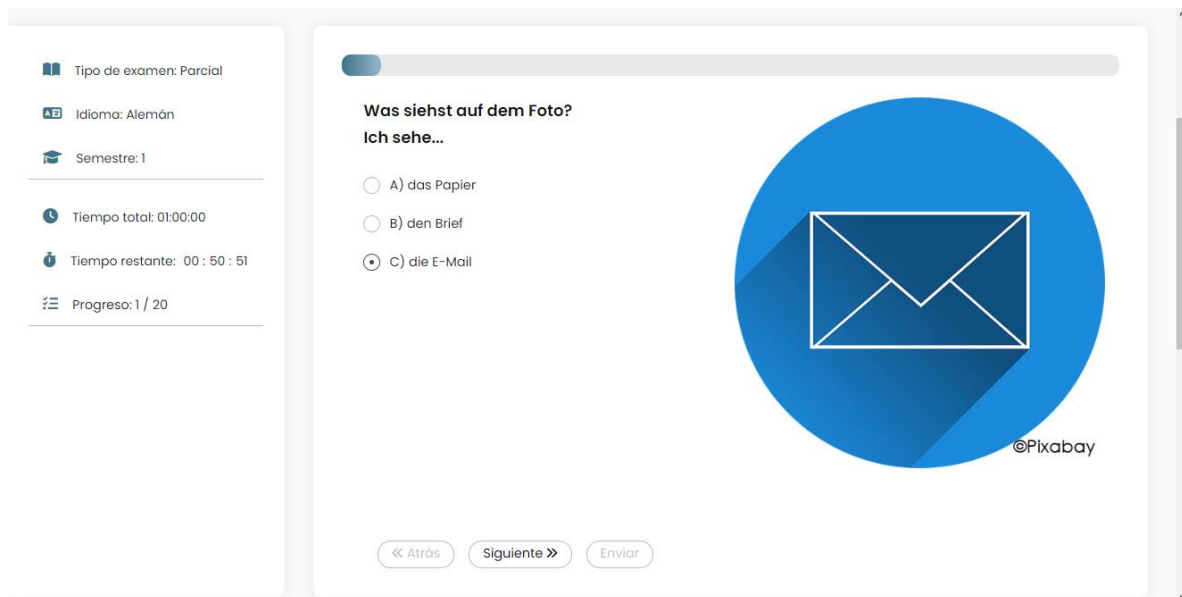


FIGURA 36. DISEÑO DE LA INTERFAZ PARA CONTESTAR UN EXAMEN

En la figura 36 se puede observar que la interfaz está conformada por dos paneles. El primer panel, ubicado del lado izquierdo, presenta información general del examen como son el tipo de examen que está presentando el alumno, el idioma que evalúa y el semestre en que se encuentre. Debajo se encuentra el tiempo total que tiene para contestar su evaluación, el cronómetro que le indicará el tiempo restante segundo a segundo, y el indicador de progreso, que muestra la pregunta en la que se encuentra.

En el panel del lado derecho se observa la barra de progreso que muestra la misma idea que el indicador de progreso del lado izquierdo, pero esta vez mediante una barra que incrementa o decrementa su longitud de acuerdo a la pregunta en que se encuentra posicionado el alumno. Debajo de esta barra se encuentran el texto de la pregunta y, dependiendo del tipo de pregunta, aparecerán las posibles opciones o un cuadro de texto. Para el caso de las preguntas abiertas, como puede verse en la

4. CAPÍTULO IV. RESULTADOS

figura 37, el usuario contará con un cuadro de texto donde puede escribir su respuesta; para las preguntas de opción múltiple y selección múltiple, se enlistarán todas las opciones para que el alumno seleccione su respuesta o respuestas, como puede observarse en las figuras 36 y 39 respectivamente; y para las preguntas de tipo de lista desplegable, como se aprecia en la figura 38, simplemente se muestra el elemento de lista.

Tipo de examen: Parcial
Idioma: Alemán
Semestre: 1
Tiempo total: 01:00:00
Tiempo restante: 00 : 58 : 54
Progreso: 5 / 20

Ergänzen Sie das passende Wort.
die Adresse - die Straße - der Absender -
das Land - die Biefmarke - die Stadt - die
Hausnummer - der Empfänger - die
Postleitzahl.

Susi Reifer (2) _____
89 rue Manfeld (1) _____
95005 Bam (3) _____
FRANKREICH

Tom
Kornelia Franke (4) _____
Clemensstraße 25 (5) _____ (9) _____
80498 München (6) _____ (7) _____
DEUTSCHLAND (8) _____

Your answer ...

<< Atrás Siguiente >> Enviar

FIGURA 37. PREGUNTA DE TIPO ABIERTA

Tipo de examen: Parcial
Idioma: Alemán
Semestre: 1
Tiempo total: 01:00:00
Tiempo restante: 00 : 58 : 08
Progreso: 7 / 20

In der Adresse fehlt die

Selecciona una opción

<< Atrás Siguiente >> Enviar

FIGURA 38. PREGUNTA DE TIPO LISTA DESPLEGABLE

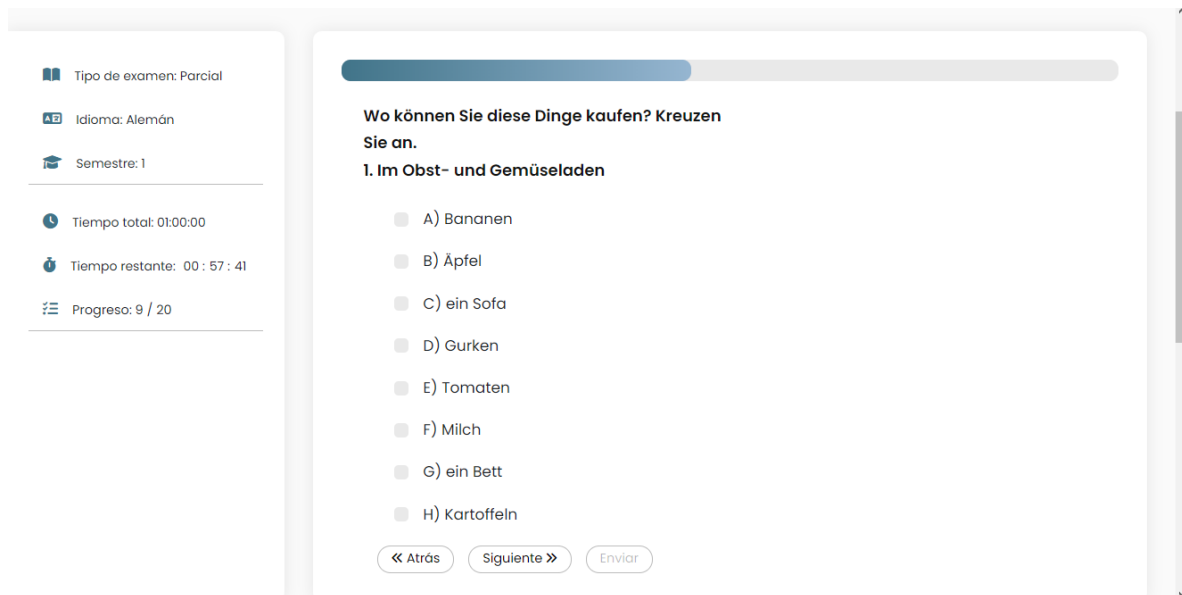


FIGURA 39. PREGUNTA DE TIPO SELECCIÓN MÚLTIPLE

Si la pregunta cuenta con un elemento multimedia, este será mostrado del lado derecho. Para los elementos multimedia de tipo audio o video, estos podrán ser reproducidos únicamente un par de veces. Un video o audio se considera como ya reproducido en cuanto el audio o video finaliza. Por lo tanto, para evitar que los alumnos puedan reproducirlo indefinidamente, alternando entre las preguntas antes de que se considere la reproducción como finalizada, se bloquean los botones de navegación y se desbloquean en cuanto termina. Este registro de veces reproducidas se lleva en el almacenamiento local (local storage) del navegador por cada video y audio, indicando el número de veces que ya ha sido reproducido. Una vez que el almacenamiento local ya ha registrado dos reproducciones de un audio o video, el elemento multimedia deja de ser visible para el alumno. Igualmente, estos botones se bloquean para evitar que los alumnos que deseen escuchar y/o visualizar su audio o video cambien de pregunta por accidente y pierdan una de las dos oportunidades para reproducirlo.

Por último, se encuentran los botones de navegación, que corresponden a los botones de retroceso (Atrás) y avance (Siguiente), y el botón de envío del examen (Enviar). El botón de retroceso se encontrará bloqueado siempre que el alumno se encuentre en la pregunta con posición número 1 para evitar retroceder a “una pregunta con posición 0”. Similar a ello, se bloquea el botón de avance cuando el alumno se encuentre en la última pregunta para evitar llegar a “una pregunta adicional que no existe”. En cuanto al botón de envío, se encontrará bloqueado en

todo momento a fin de evitar que el alumno envíe su examen por accidente, por lo tanto, solo podrá enviarlo cuando se encuentre en la última pregunta.

Cuando se resuelve un examen de forma digital pueden suceder situaciones en las que exista pérdida de datos como, por ejemplo, falla o pérdida en la conexión a Internet, el navegador se cierre o incluso que el dispositivo se apague. Como medida de recuperación del progreso ya realizado por los alumnos, las respuestas serán almacenadas en cuanto se les dé clic o sean escritas, en el almacenamiento local del navegador. Entonces, cada 5 minutos, todo el registro del progreso realizado por el alumno guardado en el almacenamiento local será enviado a través de peticiones AJAX al controlador, para que este pueda manipular al modelo y poder así almacenar en la base de datos las respuestas ya ingresadas por el alumno, acompañadas de otros datos como son el tiempo restante. La función desarrollada para llevar a cabo este guardado de respuestas se muestra en la figura 40.

```
ajax_saveAnswers(data){
  //AJAX setup config
  $.ajaxSetup({headers: {'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')}})
  var route = $('#saveAnswersURL').attr('url')

  // AJAX
  $.ajax({
    url: route,
    type: 'POST',
    data: JSON.stringify(data),
    cache: false,
    contentType: 'application/json',
    processData: false,
    success: function(data){
    }
  })
}
```

FIGURA 40. FUNCIÓN AJAX PARA EL GUARDADO DE RESPUESTAS

El propósito de almacenar este valor es, por ejemplo, si el alumno decide continuar desde otro navegador, que no tenga ventaja de reanudar su examen con el tiempo corriendo desde el principio. Por ello, se almacena el tiempo restante para que este pueda ser tomado en cuenta desde que el alumno vuelve a ingresar a su evaluación. También, mientras el examen no sea enviado bajo ninguna circunstancia, el estado registrado del examen en la base de datos será el de “En progreso”.

El envío del examen se dará por dos razones: porque el alumno envió su evaluación, o porque el tiempo restante para contestar su evaluación se ha agotado. Si el alumno desea enviar su examen, pero aún hay preguntas sin respuestas, la plataforma mostrará una alerta donde se le notifique esta situación al alumno y preguntando si desea enviarlo de todos modos. En caso de que aún tenga tiempo

restante y desee dar respuesta a esas preguntas, podrá hacerlo; en caso contrario, el examen será enviado aun cuando esas preguntas no tengan respuesta. Por otro lado, independientemente de si aún restan preguntas sin contestar o no, si el tiempo restante se ha agotado, se le notificará al alumno mediante una alerta y se enviará el examen.

Una vez que el examen sea enviado, se almacenarán de nuevo las respuestas del alumno, esto debido a que, si aún no terminaba el intervalo de 5 minutos para almacenar las respuestas en el momento que se envía el examen, se tenga garantía que las respuestas serán almacenadas. Posteriormente, el controlador se encargará de verificar las respuestas recibidas acorde al tipo de pregunta. Así, todas las preguntas abiertas no serán consideradas para la calificación y puntaje obtenido por el estudiante.

El resto de tipo de preguntas serán comparadas con la respuesta correcta registrada en la base de datos y, si la respuesta es incorrecta, dentro de un arreglo se guardarán las sugerencias de estudio para el alumno relacionadas a esa pregunta. Si otra pregunta fue contestada incorrectamente y la sugerencia de estudio de esa pregunta ya se encuentra en el arreglo, no será agregada, eso para evitar que al alumno le sea presentado un listado con temas repetidos que, lejos de ayudarlo, le será más confuso. Por cada pregunta correcta se incrementará en 1 el puntaje total obtenido y, al término, el puntaje obtenido y el número total de preguntas serán los valores fundamentales para obtener la calificación final del alumno.

Una vez se ha concluido el proceso de calificar el examen, esta información será almacenada en la base de datos con ayuda del modelo, que se encargará de actualizar los campos del puntaje y del estado del examen que cambiará de “En progreso” a “Finalizado”. El objetivo de registrar el estado del examen es el de prevenir que el alumno envíe una segunda respuesta de su evaluación y así obtener una calificación a su favor, pues una vez que el examen finaliza se muestran los resultados obtenidos con sus respuestas, en conjunto con las respuestas correctas a cada interrogante.

4.8 Desarrollo de la interfaz para visualizar los resultados obtenidos

Al término de su evaluación, ya que el examen se encuentre en estado “Finalizado”, el alumno podrá visualizar los resultados alcanzados. También, esta interfaz también puede ser accedida desde el listado de exámenes presionando el botón de

4. CAPÍTULO IV. RESULTADOS

Ver examen, solo que esta vez no se visualizará el examen, sino que se mostrará un mensaje que indica que la evaluación ya ha sido respondida, pero que se pueden volver a ver los resultados obtenidos.

En la figura 41 puede observarse que al principio se le muestra al estudiante una tabla con un resumen de dichos resultados: la fecha y hora en que dio inicio a su examen, el estado del examen (que para ese momento es de carácter obligatorio que sea “Finalizado”), el tiempo que utilizó para resolver su examen, la fecha y hora en que el examen fue enviado, los puntos obtenidos, y la calificación obtenida.



Resumen de resultados	
Fecha de inicio:	06/10/2023, 20:41 hrs.
Estado:	Finalizado
Finalizado en:	06/10/2023, 20:52 hrs.
Tiempo empleado:	00:11:22 hrs
Puntos:	7/20
Calificación:	3.5

FIGURA 41. RESUMEN DE LOS RESULTADOS OBTENIDOS EN UN EXAMEN

Debajo de este resumen se encuentra la retroalimentación donde se muestran las preguntas junto con la respuesta recibida, y la respuesta correcta en cada una. Para las preguntas abiertas solo se muestra la leyenda “Estado: Texto pendiente por revisar”, pues como se dijo antes, será tarea de un docente o evaluador calificar estas preguntas.

Independientemente del tipo de pregunta y de si fue respondida correcta o incorrectamente, siempre se le mostrará al alumno la respuesta correcta de forma textual, así como también dentro de las opciones marcadas con colores dependiendo de cómo fue contestada. Para los tipos de preguntas opción múltiple y lista desplegable, si la respuesta del alumno fue correcta, se mostrará la opción en color verde, por el contrario, si la respuesta fue incorrecta se mostrará la opción marcada en color rojo y la respuesta correcta que no seleccionó se mostrará en color verde, como puede observarse en la figura 42.

4. CAPÍTULO IV. RESULTADOS

Por último, para las preguntas de tipo selección múltiple que poseen más de una sola respuesta, se mostrará el texto de la pregunta seguido de sus opciones. Si las opciones marcadas por el alumno fueron correctas, se mostrarán en color verde; si las opciones marcadas por el alumno fueron incorrectas, se mostrarán en color rojo; y si al alumno le faltaron seleccionar opciones que están incluidas en la respuesta correcta, serán marcadas con azul. Un ejemplo de esto puede apreciarse en la figura 43.

8) Was siehst auf dem Foto?
Ich sehe...

- A) eine Verkäuferin
- B) eine Kundin
- C) ein Angebot

Respuesta correcta: A) eine Verkäuferin




FIGURA 42. EJEMPLO DE RETROALIMENTACIÓN EN TIPOS DE PREGUNTA OPCIÓN MÚLTIPLE Y LISTA DESPLEGABLE

9) Wo können Sie diese Dinge kaufen? Kreuzen Sie an.
1. Im Obst- und Gemüseladen

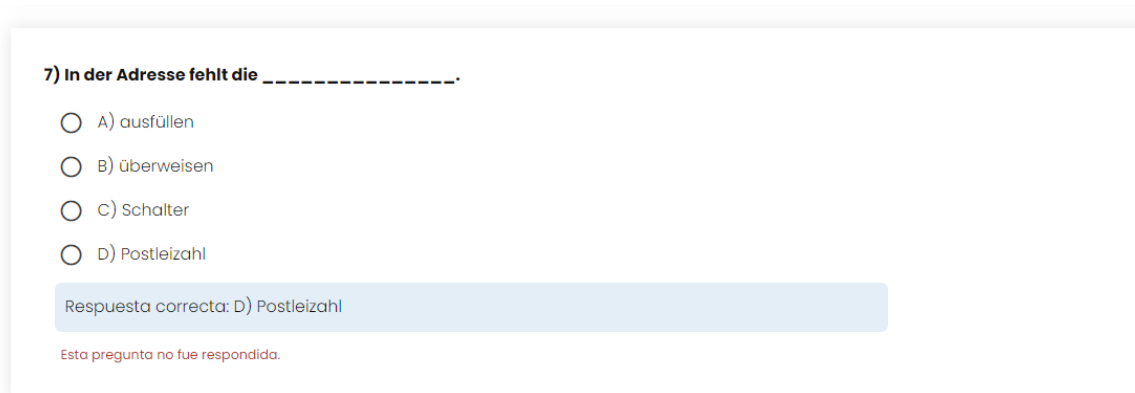
- A) Bananen
- B) Äpfel
- C) ein Sofa
- D) Gurken
- E) Tomaten
- F) Milch
- G) ein Bett
- H) Kartoffeln

Respuesta correcta: A) Bananen, B) Äpfel, D) Gurken, E) Tomaten, H) Kartoffeln

FIGURA 43. EJEMPLO DE RETROALIMENTACIÓN EN TIPO DE PREGUNTA SELECCIÓN MÚLTIPLE

Si una pregunta no fue respondida, debajo de la pregunta se podrá leer la leyenda en color rojo “*Esta pregunta no fue respondida*” y, a diferencia de cuando una

pregunta si es respondida, no se mostrará la respuesta correcta marcada en color verde. El uso de colores en únicamente las preguntas que fueron respondidas tiene como propósito proporcionar al alumno una forma amigable de comparar su respuesta con las demás opciones. Un ejemplo de pregunta sin responder se muestra en la figura 44.



7) In der Adresse fehlt die _____

- A) ausfüllen
- B) überweisen
- C) Schalter
- D) Postleitzahl

Respuesta correcta: D) Postleitzahl

Esta pregunta no fue respondida.

FIGURA 44. RETROALIMENTACIÓN DE PREGUNTA SIN RESPONDER

Al finalizar la retroalimentación, se integra el listado de sugerencias de estudio para el alumno de acuerdo a las preguntas respondidas incorrectamente. Debajo de este listado se encuentra el botón de *Guardar resultados en PDF* que, como su nombre lo sugiere y con la finalidad de que el alumno pueda conservar los temas sugeridos de estudio, este botón permite guardar únicamente ese listado para que el alumno pueda consultarlo cuando lo desee y cuantas veces lo requiera. En la figura 45 puede apreciarse la ventana que permite el guardado de las sugerencias resultantes del examen.

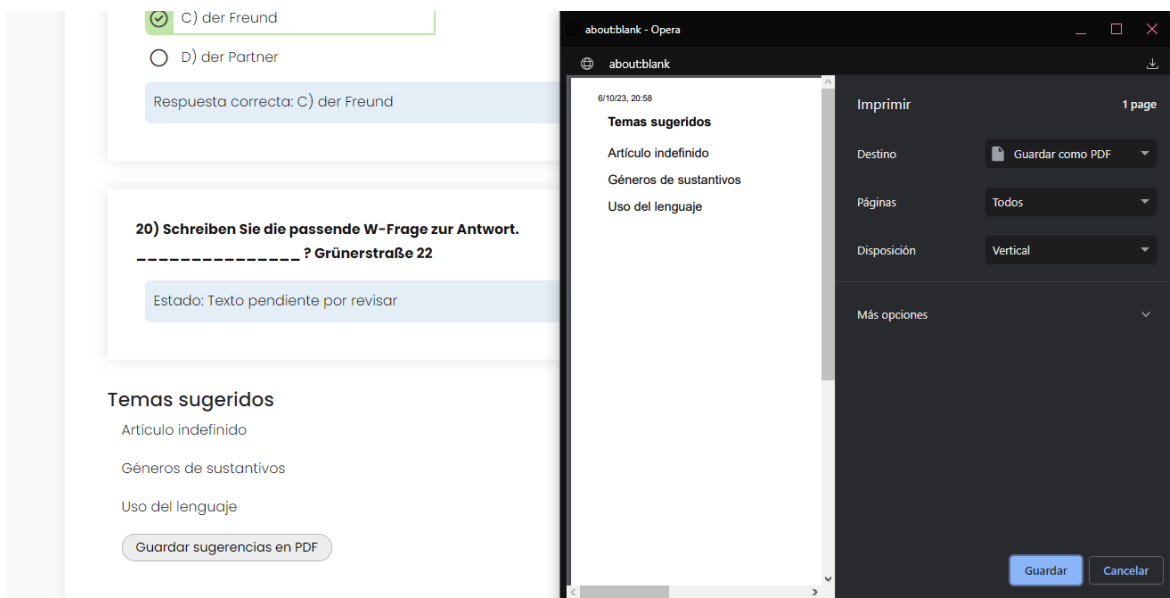


FIGURA 45. VISTA DEL GUARDADO DE RESULTADOS

Al final de la interfaz, se muestra el botón de *Volver al menú*, donde se direcciona al menú principal de la plataforma.

4.9 Desarrollo de la interfaz para visualizar los reportes de un examen

Esta interfaz puede ser accedida desde el listado de exámenes, presionando el botón de *Ver reporte de resultados*. Ese botón direccionará al usuario a esta interfaz donde se presenta un resumen en forma de tabla que contiene información general del examen como es su ID, sus fechas de inicio y término, las habilidades evaluadas, su tipo o propósito, la asignatura y semestre a los que está destinado, y el idioma que evalúa.

La razón por la que esta interfaz fue desarrollada surge de la idea de que, dadas las sugerencias de estudio a los alumnos, que sea mediante estos reportes, que los docentes o administradores puedan llevar un registro de las calificaciones obtenidas por los alumnos a fin de que puedan realizar estadísticas con esa información o, simplemente, realizar estudios comparativos sobre las calificaciones de los alumnos y comprobar si su rendimiento se ha elevado gracias a las sugerencias dadas. Estos reportes pueden ser visualizados por día, semana o mes.

4. CAPÍTULO IV. RESULTADOS

Posteriormente, se presenta otra tabla con información general de los alumnos como son su número de cuenta (ID), apellidos paterno y materno, nombre, la calificación obtenida, y el día en que el alumno envió su examen. En la figura 46 se observa un ejemplo de un registro de resultados ordenado por semana.

Ordenar los datos por: **Semana**

Resumen

ID examen:	2
Fecha de inicio:	2023-10-11
Fecha de término:	2023-10-11
Habilidades evaluadas:	Uso del idioma
Tipo/propósito:	Parcial
Asignatura:	Alemán I
Semestre:	1
Idioma:	Alemán

Resultados

Fecha: 8/10/2023 hasta 14/10/2023

ID alumno	Apellido paterno	Apellido materno	Nombre	Calificación	Fecha de entrega
328174	Smith	Smith	John	3.5	11/10/2023, 20:06

Imprimir Volver al listado

FIGURA 46. DISEÑO DE LA INTERFAZ DE REPORTES

Debajo de esa tabla se añadió el botón de *Imprimir*, el cual le permitirá al usuario imprimir toda la información ya descrita. A su lado, también se encuentra el botón de *Volver al listado*, el cual permite volver al usuario al listado de exámenes.

4.10 Validación

Siempre que se desarrolla un software, es necesario que le sean aplicadas pruebas de validación, a fin de que sus resultados permitan comprobar si el producto final cumple con los requerimientos expresados por el cliente. Durante el proceso de desarrollo de cada una de las funciones que integran el Sistema de proceso de evaluación de perfil lingüístico, se realizaron diversas pruebas para verificar que su comportamiento fuera el esperado. Estas pruebas se hicieron con propósito de demostrar su funcionalidad, usabilidad, confiabilidad y eficiencia.

4.10.1 Validación

Para garantizar que los productos de software sean de calidad, se han creado muchos estándares internacionales que miden este aspecto, entre los que se encuentra la norma ISO/IEC 9126 (Chua & Dyson, 2004), la cual provee de métricas de calidad aplicables a sistemas informáticos. Esta norma define una serie de factores básicos divididos, a su vez, en subfactores, que permiten hacer más precisa la evaluación del producto. Estos factores y subfactores se enlistan en la tabla 30.

TABLA 30. FACTORES DEL PROCESO DE CALIDAD DE LA NORMA ISO/IEC 9126.

FACTORES	SUBFACTORES
Funcionalidad	Idoneidad Exactitud Interoperabilidad Seguridad Conformidad
Confiabilidad	Madurez Recuperación Tolerancia a fallos
Usabilidad	Comprensión Facilidad de aprender Operatividad
Eficiencia	Comportamiento en el tiempo Comportamiento de recursos
Facilidad de mantenimiento	Estabilidad Facilidad de análisis Facilidad de cambio Facilidad de pruebas
Portabilidad	Adaptabilidad Facilidad de instalación

Para que los docentes y administradores de la Universidad Autónoma del Estado de Hidalgo (UAEH) obtengan resultados de calidad de las evaluaciones, es necesario que la construcción de los exámenes, la asignación de calificaciones, visualización de información de sus alumnos, etc. sean precisos. Por ello, con el propósito de confirmar que las funciones del sistema actúan conforme a los requerimientos expresados por el Centro de Lenguas, se aplicarán pruebas de funcionalidad que permitan conocer si estas funciones tienen la capacidad de proporcionar los resultados que esperan y que necesitan los usuarios.

Es común encontrar software que, para sacar su máximo provecho, es necesario contar con experiencia en el uso de sistemas similares, tomar múltiples cursos para aprender a utilizarlo, sus funciones no son claras o son poco intuitivas, entre otras. Tomando esta situación en consideración, el Sistema de proceso de evaluación de perfil lingüístico fue diseñado para servir a sus usuarios de la manera más intuitiva y sencilla posible. Aun así, se considera la aplicación de pruebas de usabilidad para confirmar que, efectivamente, el sistema cuenta con estas características.

Durante el desarrollo de la plataforma, se realizaban pruebas antes de que este fuese entregado al dueño del producto en cada *Sprint* de la metodología Scrum, a fin de que este tuviese la menor cantidad de errores durante su ejecución, y aunque la aceptación del sistema solía ser favorable, se realizarán pruebas de confiabilidad que prueben que el sistema es capaz de continuar su ejecución durante un periodo de tiempo establecido.

También, es de interés que la experiencia del usuario en el momento que se encuentre utilizando este sistema, sea ágil y atienda sus solicitudes en el menor tiempo posible. Por lo tanto, se efectuarán pruebas de eficiencia que demuestren que tan capaz es el sistema de responder rápidamente con los recursos disponibles.

La forma en cómo se realizará la validación de la plataforma será mediante un cuestionario que atienda estos factores junto con la escala de Likert. Esta escala, de acuerdo a (Bedoya Laguna, 2017), es un instrumento de medición en la investigación cuantitativa. Dicho instrumento se compone de ítems escritos de forma afirmativa, los cuales se solicita a determinado sujeto responder para conocer su opinión sobre algo.

Previo a la elaboración de los ítems, fue necesario determinar sobre qué se desea conocer la opinión de los usuarios. Anteriormente, ya se expusieron los factores de la norma ISO/IEC 9126 que serán evaluados mediante pruebas al sistema, y por

qué son necesarios. El siguiente paso fue escribir cada uno de los ítems de forma afirmativa, y que cuyas posibles respuestas sean estrictamente cerradas y que serán aplicadas para todos ellos. Se eligieron un total de 5 posibles respuestas de las cuales dos opciones son positivas, una neutra y dos negativas: *Totalmente de acuerdo*, *De acuerdo*, *Regular*, *En desacuerdo* y *Totalmente en desacuerdo*.

Una vez definidos los ítems, presentados en el Apéndice B, se elaboró un formulario digital que se solicitó a los usuarios dar respuesta con el objetivo de conocer su opinión sobre los diferentes factores del sistema. Los resultados obtenidos se exponen a continuación.

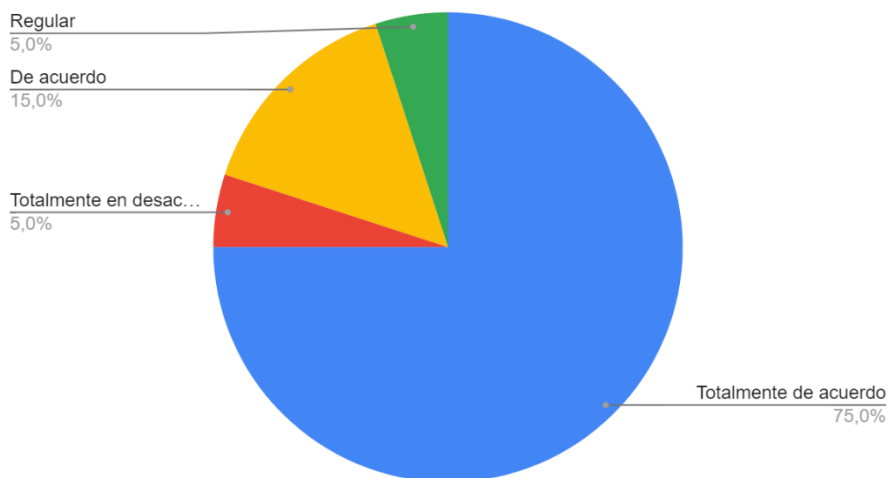
4.10.1.1 Pruebas de Funcionalidad

La funcionalidad de un software, de acuerdo a la norma ISO/IEC 9126, evalúa el grado en que el sistema satisface correctamente las tareas para las cuales fue diseñado. Para conocer el grado de conformidad alcanzado, se plantearon las siguientes afirmaciones.

1. El examen creado fue acorde a los parámetros establecidos.

Con esta afirmación, se pretende conocer si el examen generado por el sistema obedece a las características señaladas por el docente o administrador. Los resultados obtenidos se observan en la Gráfica 1 que expone que el 75% de usuarios indicaron estar *Totalmente de acuerdo*, el 15% estar *De acuerdo*, el 5% indicaron *Regular* y otro 5% estar *Totalmente en desacuerdo*.

El examen creado fue acorde a los parámetros establecidos.

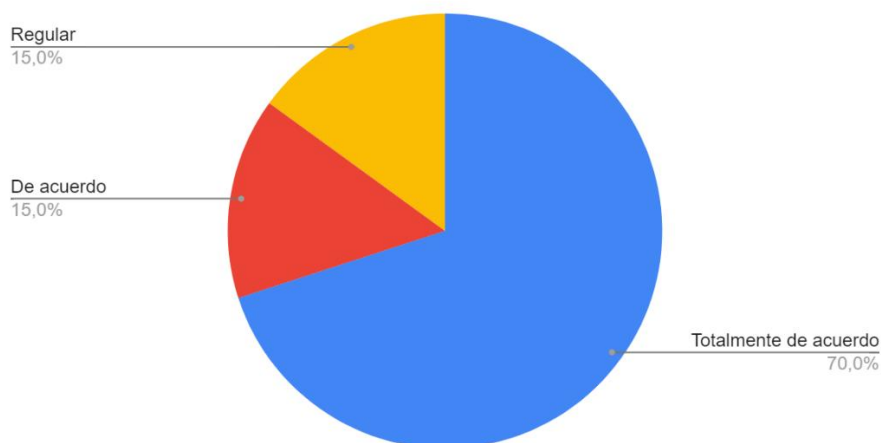


GRÁFICA 1. RESULTADOS DE LA PRIMERA PREGUNTA DE FUNCIONALIDAD

2. Los parámetros requeridos son los apropiados para la creación de un examen.

Esta afirmación fue elaborada para conocer el nivel de conformidad con los parámetros solicitados a los docentes y administradores, para construir evaluaciones que atiendan a las características requeridas por estos usuarios. La Gráfica 2 indica que el 70% estuvo *Totalmente de acuerdo*, el 15% estuvo *De acuerdo*, y el restante 15% estuvo *Regular*.

Los parámetros requeridos son los apropiados para la creación de un examen.

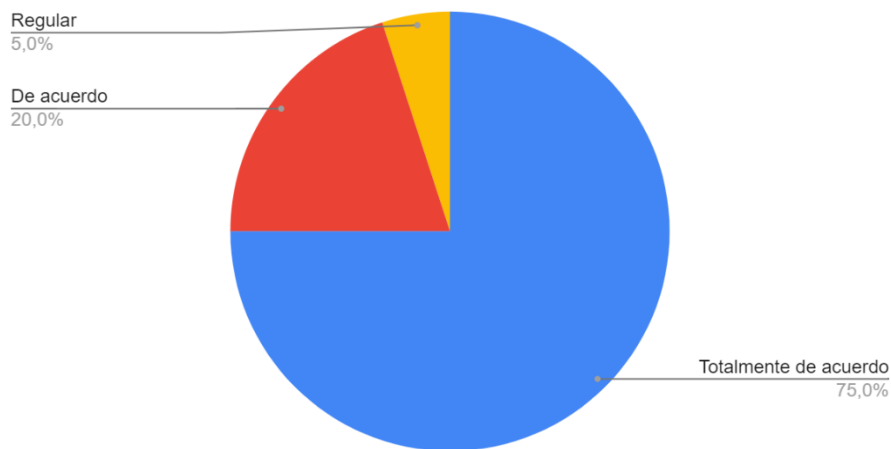


GRÁFICA 2. RESULTADOS DE LA SEGUNDA PREGUNTA DE FUNCIONALIDAD

3. El examen digital creado se adhiere a las características que tendría un examen físico de idiomas.

Los exámenes de idiomas evalúan diferentes aspectos para determinar el nivel de dominio de un alumno respecto a una lengua. Estas evaluaciones están estructuradas por preguntas que evalúan una habilidad en particular. Mediante la opinión proporcionada por los usuarios en esta afirmación, se desea conocer qué tanto el examen virtual se adhiere a las características que poseen los exámenes físicos de idiomas que son aplicados actualmente. La Gráfica 3 expone que el 75% indicó estar *Totalmente de acuerdo*, el 20% estar *De acuerdo* y el 5% *Regular*.

El examen digital creado se adhiere a las características que tendría un examen físico de idiomas.

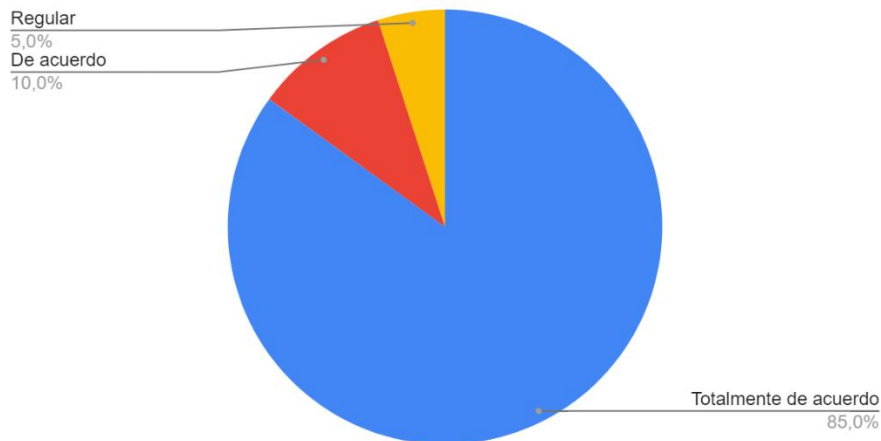


GRÁFICA 3. RESULTADOS DE LA TERCERA PREGUNTA DE FUNCIONALIDAD

4. Las ventanas y botones que se presentan son adecuadas.

Es importante para el usuario que los botones empleados, así como las ventanas necesarias para visualizar determinada información estén presentes en todo momento, en una posición apropiada y, además, que sean los adecuados para navegar a través del sistema. Los resultados obtenidos se observan en la Gráfica 4, en la que el 85% se encuentra *Totalmente de acuerdo*, 10% *De acuerdo* y 5% *Regular*.

Las ventanas y botones que se presentan para la visualización de la información y su navegación son adecuadas.

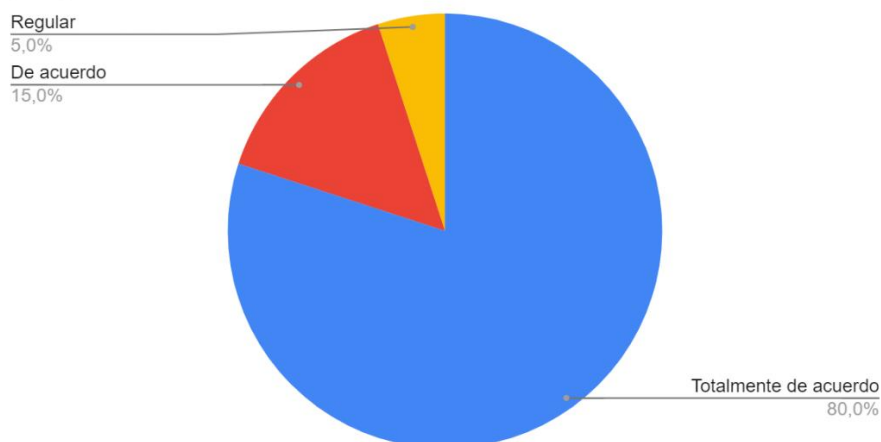


GRÁFICA 4. RESULTADOS DE LA CUARTA PREGUNTA DE FUNCIONALIDAD

5. La información solicitada sobre un estudiante, examen o pregunta son veraces.

Si el administrador o el docente desean consultar la información relacionada a los alumnos o a los exámenes que serán aplicados, la información mostrada por el sistema debe ser veraz. Por esta razón, esta afirmación tiene el propósito de medir que tan fiable es la información presentada a los usuarios. En la Gráfica 5 se observan los resultados obtenidos: el 80% estuvo *Totalmente de acuerdo*, el 15% *De acuerdo* y el 5% indicó estar *Regular*.

La información solicitada sobre un estudiante, examen o pregunta son veraces.



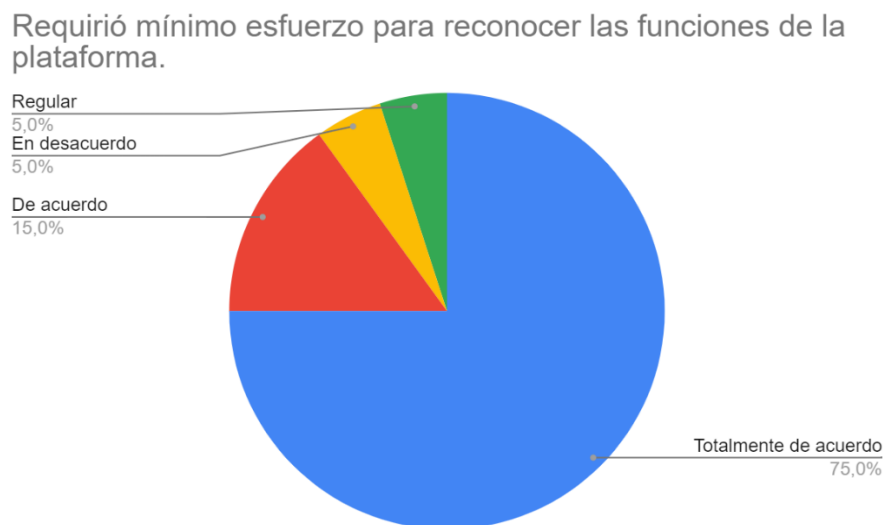
GRÁFICA 5. RESULTADOS DE LA QUINTA PREGUNTA DE FUNCIONALIDAD

4.10.1.2 Pruebas de Usabilidad

La usabilidad, según la norma ISO/IEC 9126, determina el nivel de esfuerzo que un usuario requiere para hacer uso correcto del sistema. Para evaluar este factor, se elaboraron afirmaciones que hacen énfasis en la comprensibilidad del sistema. Estas afirmaciones y sus resultados se presentan a continuación.

6. Requirió mínimo esfuerzo para reconocer las funciones de la plataforma.

El sistema se desarrolló teniendo en mente que este fuese lo más intuitivo y sencillo posible, de modo que el usuario requiriera de poco esfuerzo para identificar cada una de sus funciones (crear examen, visualizar datos de alumnos, visualizar datos de un examen, etc.). Por ello, se consultó acerca de esto mediante esta afirmación. La Gráfica 6 expone los resultados obtenidos: el 75% estuvo *Totalmente de acuerdo*, el 15% indicó estar *De acuerdo*, el 5% estar *Regular* y el 5% estuvieron *En desacuerdo*.

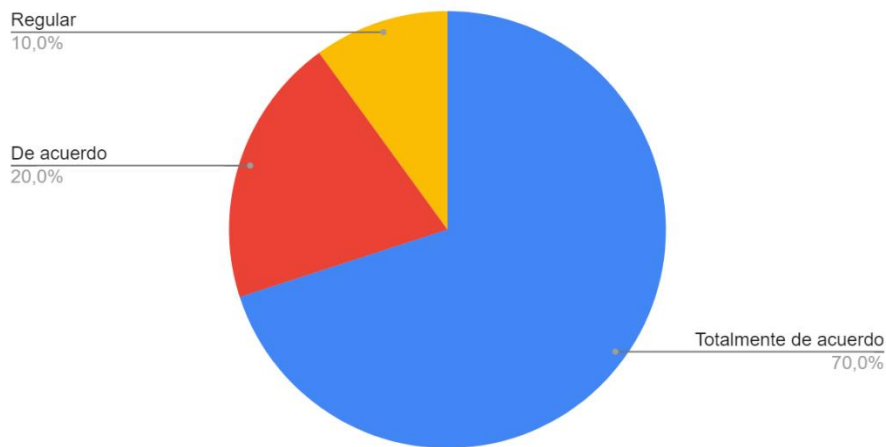


GRÁFICA 6. RESULTADOS DE LA PRIMERA PREGUNTA DE USABILIDAD

7. Requirió mínimo esfuerzo para aprender a utilizar la plataforma para la resolución de un examen.

Esta afirmación se elaboró con el objetivo de conocer que tanto esfuerzo requirió el usuario final para resolver su examen dentro de la plataforma, y conocer que tan complejo o sencillo le fue navegar a través de su examen, seleccionar o escribir su respuesta en función del tipo de pregunta, identificar y utilizar la función de envío de su examen, entre otros. La Gráfica 7 describe que el 70% de los usuarios estuvieron *Totalmente de acuerdo*, el 20% indicó estar *De acuerdo* y el 10% estar *Regular*.

Requirió mínimo esfuerzo para aprender a utilizar la plataforma para la resolución de un examen.



GRÁFICA 7. RESULTADOS DE LA SEGUNDA PREGUNTA DE USABILIDAD

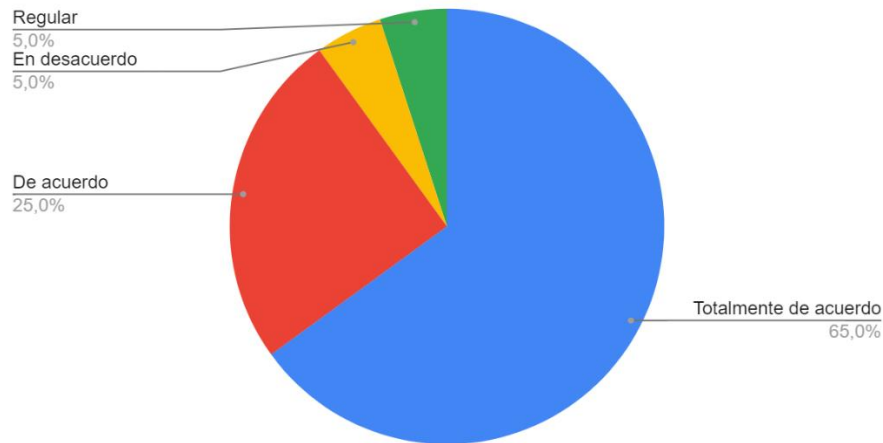
8. Requirió mínimo esfuerzo para aprender a utilizar la plataforma para la creación de un examen.

De manera similar a la anterior, se elaboró esta afirmación para conocer el nivel de esfuerzo que el usuario requirió para identificar las funciones que contribuyen a la generación de un examen, además de comprender lo que cada parámetro solicita para llevar a cabo esta tarea (indicar fecha y hora de inicio y finalización, idioma, habilidades a evaluar, etc.). En la Gráfica 8 se puede apreciar que el 65% indicó estar *Totalmente de acuerdo*, el 25% indicó estar *De acuerdo*, el 5% *Regular* y el 5% restante estar *En desacuerdo*.

9. La interfaz utiliza una paleta de colores, la distribución de botones y textos, y el tamaño y tipo de fuente son adecuados.

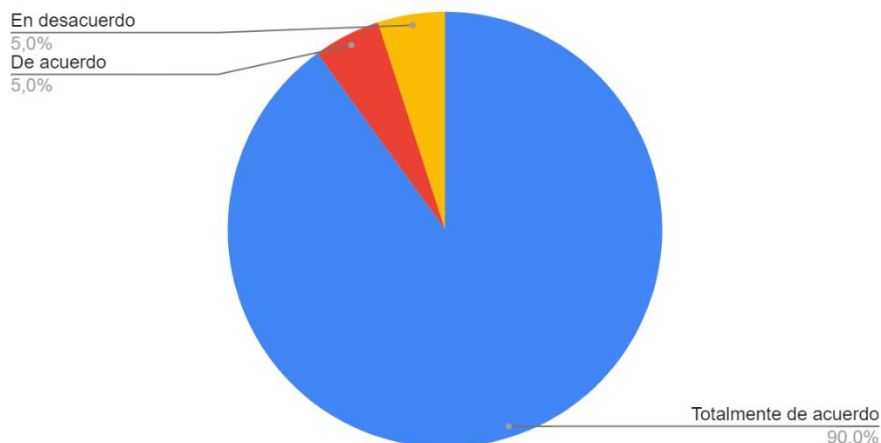
La paleta de colores utilizada es la misma que la del banner presentado en cada una de las páginas del sistema y que fue añadido por el propio Centro de Lenguas de la UAEH. La distribución de los botones y la presentación de la información deberían mostrarse de forma que el usuario no tenga dificultad para leer y hacer uso de ellos; asimismo, el tipo y tamaño de la fuente debe ser uno que no complique su lectura. Esta afirmación se elaboró para determinar si la presentación de estos aspectos son los ideales, y se obtuvieron los resultados presentados en la Gráfica 9 que indican que el 90% de usuarios estuvieron *Totalmente de acuerdo*, el 5% estuvieron *De acuerdo* y otro 5% indicaron estar *En desacuerdo*.

Requirió mínimo esfuerzo para aprender a utilizar la plataforma para la creación de un examen.



GRÁFICA 8. RESULTADOS DE LA TERCERA PREGUNTA DE USABILIDAD

La interfaz utiliza una paleta de colores, la distribución de botones y textos, y el tamaño y tipo de fuente son adecuados.

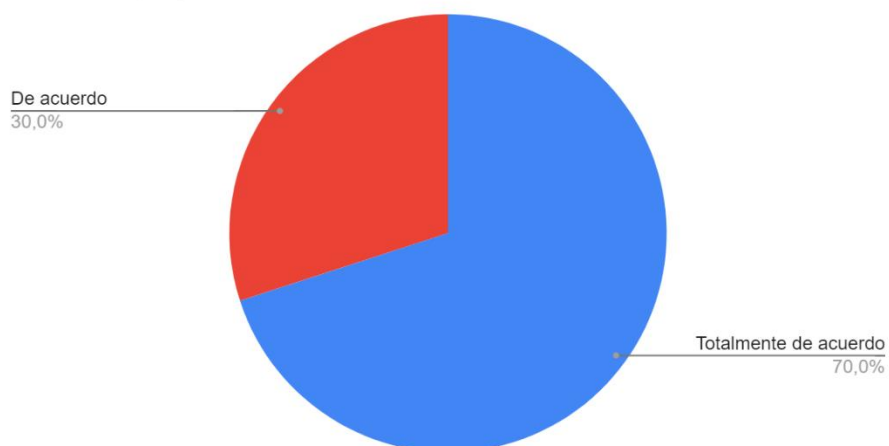


GRÁFICA 9. RESULTADOS DE LA CUARTA PREGUNTA DE USABILIDAD

10. La plataforma creada y sus funciones son las ideales para la creación y aplicación de un examen virtual.

Para conocer si las funciones que la plataforma utiliza para crear un examen y aplicarlo a un alumno son las necesarias, se escribió esta afirmación. La Gráfica 10 expone que el 70% indicó estar *Totalmente de acuerdo* y el restante 30% indicó estar *De acuerdo*.

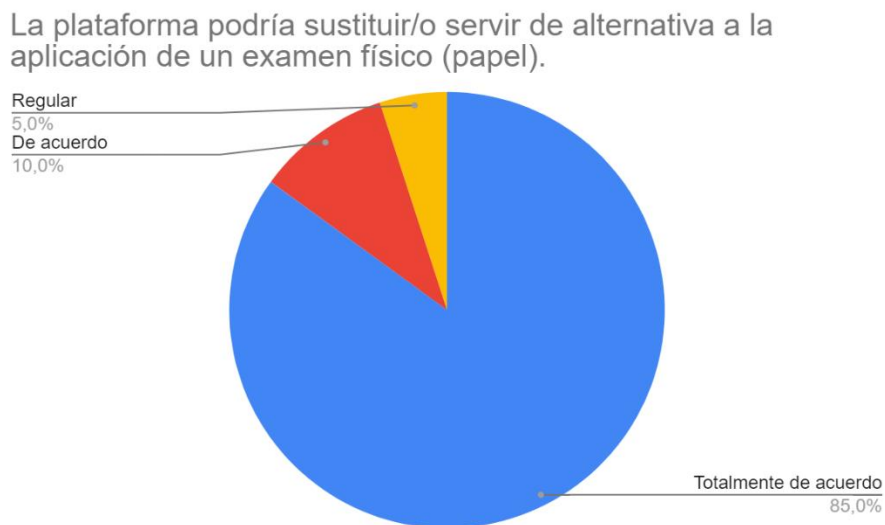
La plataforma creada y sus funciones son las ideales para la creación y aplicación de un examen virtual.



GRÁFICA 10. RESULTADOS DE LA QUINTA PREGUNTA DE USABILIDAD

11. La plataforma podría sustituir/o servir de alternativa a la aplicación de un examen físico (papel).

El sistema fue desarrollado pensando como una alternativa para aplicar, de manera digital, los exámenes que, actualmente, se aplican de forma tradicional. Para determinar si la plataforma podría sustituir la creación y resolución de exámenes utilizando recursos como papel y tinta, se elaboró esta afirmación cuyos resultados pueden ser observados en la Gráfica 11 que indica que el 85% de usuarios señaló estar *Totalmente de acuerdo*, el 10% señaló estar *De acuerdo* y un 5% estar *Regular*.



GRÁFICA 11. RESULTADOS DE LA SEXTA PREGUNTA DE USABILIDAD

4.10.1.3 Pruebas de Confiabilidad

El siguiente factor para medir la calidad de este sistema es la confiabilidad, que evalúa la capacidad del sistema de ser ejecutado en condiciones normales durante un determinado lapso de tiempo. Las afirmaciones correspondientes a esta prueba se presentan a continuación.

12. Es posible recuperar un examen con respuestas ya ingresadas por un estudiante aún después de una falla.

Cuando se realiza alguna actividad de manera digital, es posible que ocurriese alguna falla o situación donde pueda existir pérdida de información. Con esta

afirmación se pretende obtener la capacidad del sistema de recuperar la información ya introducida por el alumno en su examen. En la Gráfica 12 se observa que el 45% indicó estar *Totalmente de acuerdo*, el 40% indicó estar *De acuerdo*, un 5% indicó una respuesta *Regular*, y el 10% *En desacuerdo*.

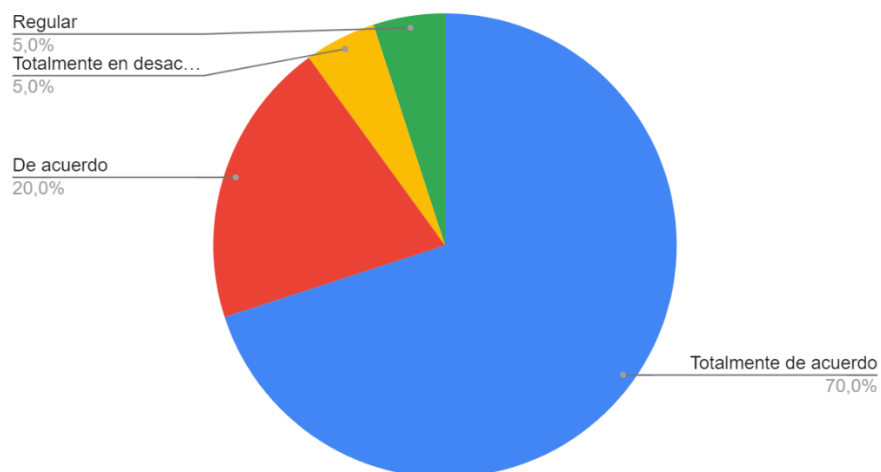


GRÁFICA 12. RESULTADOS DE LA PRIMERA PREGUNTA DE CONFIABILIDAD

13. La tasa de errores encontrados en la plataforma es baja.

Para conocer con qué frecuencia y cantidad, los usuarios encuentran errores, ya sea solicitando información a la base de datos, en el proceso de creación de un examen o incluso en la navegación en el momento de utilizar la plataforma, se elaboró esta afirmación. Los resultados se pueden visualizar en la Gráfica 13 que describe que el 70% indicó estar *Totalmente de acuerdo*, el 20% indicó *De acuerdo*, el 5% *Regular* y el 5% estar *Totalmente en desacuerdo*.

La tasa de errores encontrados en la plataforma es baja.



GRÁFICA 13. RESULTADOS DE LA SEGUNDA PREGUNTA DE CONFIABILIDAD

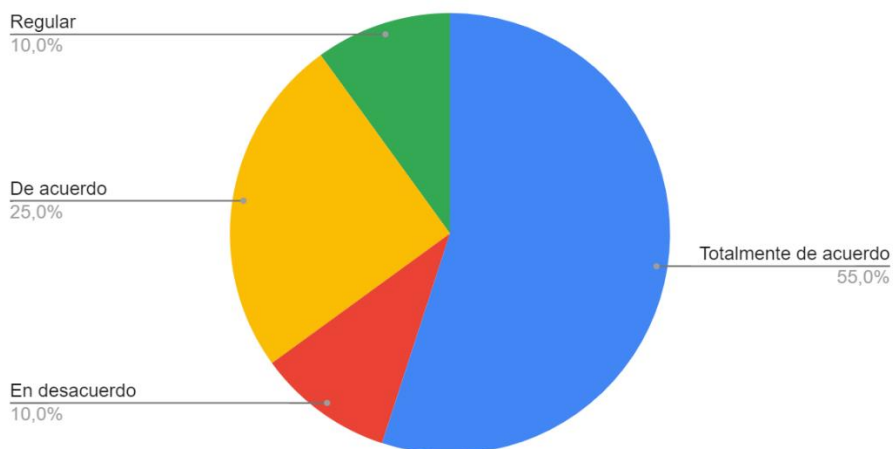
4.10.1.4 Pruebas de Eficiencia

El último factor para evaluar la calidad de la plataforma es el de eficiencia. Este factor mide que tan capaz es el sistema de optimizar los recursos que utiliza, en función del tiempo y recursos de los que dispone. Las afirmaciones correspondientes a esta prueba se presentan a continuación.

14. El tiempo de respuesta para la creación de un examen es igual o menor al esperado.

Cuando se va a acceder a una vista o se ejecuta una acción, lo ideal es que el tiempo de espera sea el menor posible. Esta afirmación se elaboró para determinar si dicho tiempo de respuesta cumple con la condición de ser igual o menor al esperado cuando se solicita la creación de un examen. En la Gráfica 14 se observa que el 55% de usuarios está *Totalmente de acuerdo*, un 25% indicó estar *De acuerdo*, otro 10% estar *Regular* y un último 10% *En desacuerdo*.

Recuento de 14. El tiempo de respuesta para la creación de un examen es igual o menor al esperado.

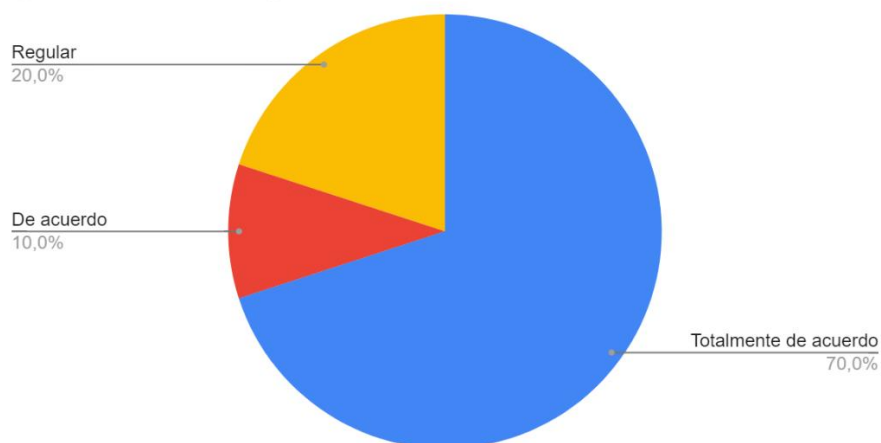


GRÁFICA 14. RESULTADOS DE LA PRIMERA PREGUNTA DE EFICIENCIA

15. El tiempo de respuesta para la visualización de un examen es igual o menor al esperado.

Esta afirmación se encargará de indicar en qué medida el tiempo de respuesta es el ideal para visualizar la información correspondiente de los alumnos y exámenes. En la Gráfica 15 se presentan los resultados obtenidos que expresan que el 70% estaba *Totalmente de acuerdo*, un 10% estaba *De acuerdo* y un 20% indicó estar *Regular*.

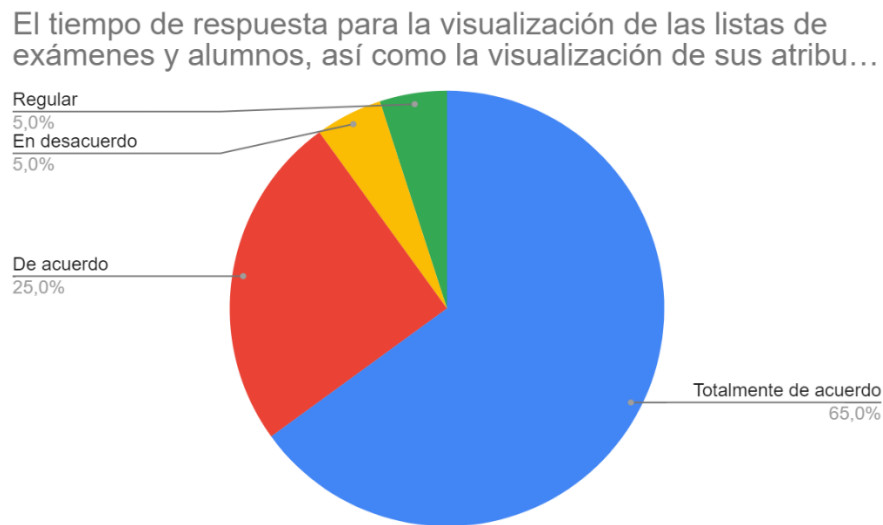
El tiempo de respuesta para la visualización de un examen es igual o menor al esperado.



GRÁFICA 15. RESULTADOS DE LA SEGUNDA PREGUNTA DE EFICIENCIA

16. El tiempo de respuesta para la visualización de las listas de exámenes y alumnos, así como la visualización de sus atributos de cada uno, es igual o menor al esperado.

Por último, esta afirmación indicará si el tiempo de respuesta de la plataforma para visualizar información, ya sea de los exámenes o de los alumnos, es el ideal. En la Gráfica 16 se presentan los resultados obtenidos que expresan que el 65% estaba *Totalmente de acuerdo*, un 25% estaba *De acuerdo*, un 5% indicó estar *Regular* y otro 5% *En desacuerdo*.



GRÁFICA 16. RESULTADOS DE LA TERCERA PREGUNTA DE EFICIENCIA

4.10.2 Conclusiones de las pruebas de validación

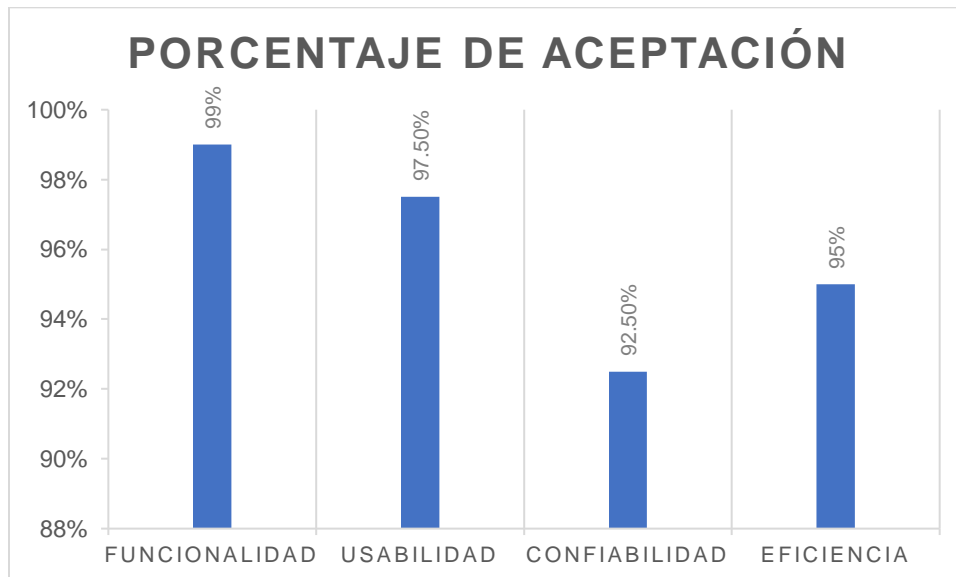
De acuerdo a la información recibida por parte de los usuarios, quienes dieron respuesta al cuestionario para evaluar la calidad del sistema, se realizaron operaciones para conocer los niveles de aceptación del sistema en cada factor evaluado. Estas operaciones consistieron en la simplificación de las respuestas de todas las afirmaciones, considerando como satisfacción con el sistema las marcadas con *Totalmente de acuerdo*, *De acuerdo* y *Regular*; y como insatisfacción con el sistema, las respuestas de *En desacuerdo* y *Totalmente en desacuerdo*.

Con esta simplificación se obtuvieron dos valores por cada afirmación: aceptación y rechazo del sistema. Para obtener los promedios, se realizó la sumatoria de los valores de aceptación en cada factor por separado, y el resultado fue dividido entre la cantidad de afirmaciones evaluadas, obteniendo así, 4 resultados que se

4. CAPÍTULO IV. RESULTADOS

encuentran en la gráfica 17, de la que se puede concluir que el nivel de aceptación en las pruebas de funcionalidad es de un 99%, en las pruebas de usabilidad los usuarios expresan un nivel de conformidad del 97.50%, en términos de confiabilidad los usuarios muestran un 92.50% de satisfacción y, por último, las pruebas de eficiencia arrojan un resultado de conformidad de 95%.

Con estos valores, se procedió a realizar una nueva operación para obtener el promedio final del grado de aceptación de la plataforma, mediante la sumatoria de los resultados de conformidad de cada factor ya mencionados, dividido entre 4, que es la cantidad de factores evaluados. El resultado de esta operación se ve reflejado en la gráfica 18 que expone que el porcentaje de aceptación del sistema es igual a 96%. Con esto, se puede afirmar que, para los usuarios, la plataforma presenta un nivel de calidad alto, que satisface las tareas para las que fue diseñado, que es usable para los usuarios que estarán utilizándolo en el futuro, que es capaz de mantener su ejecución en condiciones normales durante su uso, y que es capaz de optimizar los recursos que utiliza en función del tiempo y otros recursos de los que dispone.



GRÁFICA 17. PORCENTAJE DE ACEPTACIÓN EN CADA FACTOR EVALUADO



GRÁFICA 18. NIVEL DE ACEPTACIÓN DE LA PLATAFORMA

5. Conclusiones

El Centro de Lenguas (CeL), actualmente, emplea varios software de terceros que cubrían parcialmente la necesidad de generar y aplicar exámenes para diferentes propósitos. Estos software son carentes de seguridad y su usabilidad no era la mejor. Previo al inicio de la construcción de este módulo, se analizó software del mercado que pudiera servir como alternativa al que ha sido empleado hasta ahora, lo cual llevó a concluir que ninguno de estos podría solucionar la problemática, ya sea porque que requerían pagos de licencia elevados para utilizar el software en su totalidad, no contaban con las funciones requeridas o incluían algunas otras que no recibirían ningún uso, etc. Situación que encaminó a la planeación y desarrollo de un sistema que pudiera satisfacer eficientemente los requerimientos expresados por el Centro, que consistiera en el desarrollo tanto del Front-End como del Back-End.

Como resultado del empleo de una metodología que guiara el proceso de desarrollo del sistema, la cual fue Scrum, fue posible obtener como producto final un sistema de calidad que pudiera servir al Centro de Lenguas en la creación y aplicación de evaluaciones de la lengua alemana, y que puede ser reconocido como una pieza importante en el camino a la automatización de actividades de la institución. Su construcción implicó el uso de las tecnologías disponibles, constante comunicación con el personal del Centro, la capacidad de resolución de problemas y adaptación a los cambios, y múltiples pruebas que garantizaran su funcionamiento. Utilizar Scrum fue una tarea compleja al inicio ya que no se contaba con ninguna clase de experiencia en su manejo, sin embargo, gracias a la instrucción y continua ayuda del personal de sistemas del Centro de Lenguas de la UAEH, la metodología pudo ser llevada a cabo eficientemente.

Con la implementación de la metodología ágil Scrum, las actividades involucradas en el desarrollo del sistema como diseño, programación y pruebas, pudieron ser mejor gestionadas gracias a la elaboración del *Product Backlog* y su constante actualización de prioridades; las reuniones diarias facilitaron la pronta detección de los problemas encontrados en el sistema, y el empleo de esta metodología permitió mostrar diferentes versiones parciales del producto final con anticipación, de modo que el dueño del producto y los usuarios finales pudiesen probarlo, realizar sugerencias de cambios o integraciones de más características, y expresar su conformidad con el avance del sistema. Estas sugerencias y cambios hicieron que el sistema se aproximara aún más a las expectativas del Centro de Lenguas, y benefició a los niveles de aceptación del sistema durante las pruebas de validación.

5. CONCLUSIONES

Dado que el sistema provee de sugerencias para mejorar las habilidades de los alumnos, se consideró integrar la capacidad del sistema de realizar reportes, para que los docentes o administradores puedan tener un mejor control y seguimiento de las calificaciones obtenidas en los resultados de los alumnos antes y después de estudiar, conforme las recomendaciones dadas por la plataforma.

Este sistema demostró ser capaz de sustituir a las herramientas empleadas actualmente por el CeL, por su capacidad de elaborar exámenes en menos tiempo, y con la misma precisión de los temas que se les evalúa a los alumnos. Asimismo, se redujeron cantidades significativas de tiempo y de recursos de papelería con los que tradicionalmente se aplican exámenes a los alumnos del idioma alemán, ya que las evaluaciones generadas con este sistema pueden ser resueltas dentro del mismo, y operaciones básicas como su revisión en general y asignación de una calificación son mostradas inmediatamente al término de su resolución.

También, la implementación de botones especializados para la navegación a través de las preguntas del examen, y la creación de plantillas para diferentes preguntas que funcionaran adecuadamente, solucionaron los problemas presentados por los software empleados por el CeL para crear exámenes, por utilizar los elementos apropiados para estas acciones, y su uso es intuitivo y simple.

Además de lo anterior, con este sistema se logró la obtención de resultados más precisos de las habilidades de los alumnos en el dominio de la lengua alemana por su capacidad de no exponer características de las preguntas que faciliten la obtención de sus respectivas respuestas, durante la resolución de las evaluaciones.

Sin embargo, además del exitoso empleo de la metodología y el producto final, se destaca el compromiso de todas las personas involucradas en este proyecto, en especial la participación del Centro de Lenguas y su constante comunicación con el equipo, además de su asesoría en relación a la lengua alemana, y que fueron la clave para que este sistema lograra alcanzar los objetivos planteados al inicio. Sumado a ello, el trabajo en equipo propició un mejor desarrollo del sistema ya que, la búsqueda de soluciones en las problemáticas que surgían durante su creación, y el diseño de cada uno de sus componentes pudieron ser ejecutadas mejor de lo que una sola persona pudo haber hecho.

A su vez, también se reconoce que el Sistema de proceso de evaluación de perfil lingüístico juega un papel importante en lo que refiere a la evaluación de las habilidades y dominio de los estudiantes en una lengua extranjera, debido a que, aunque el objeto de estudio son los alumnos del idioma alemán en este documento, puede ser reutilizado o adaptado para la medición de conocimiento de otros

5. CONCLUSIONES

lenguajes, automatizando este proceso por completo para que la UAEH se vea beneficiada por las aportaciones que este proyecto informático puede proveer.

6. Trabajo futuro

Este sistema consta de un módulo que formará parte de un sistema más grande para la aplicación de exámenes digitales en la UAEH, que incluya todas las funciones requeridas para lograr ese cometido.

Dicho lo anterior, en conjunto con otros módulos se planea trabajar en la separación de los sistemas acorde al rol del usuario que ingrese, de modo que cada uno sea capaz de visualizar la información que le corresponde referente a la aplicación, resolución y creación de exámenes, visualización de reportes, etc. De la mano de lo anterior, se contempla el trabajo de mostrar a cada docente o administrador las evaluaciones creadas por ellos mismos a fin de evitar que sean eliminados por otros usuarios, y que cada usuario se encargue de su gestión. También, se considera trabajar en funciones que permitan la edición de exámenes ya creados, ya sea eliminando preguntas o añadiendo otras que atiendan a las características del examen.

Una vez esto haya sido realizado, y el sistema esté conformado por todos sus módulos, podrá iniciar el proceso de instalación e implementación dentro de un servidor propio del Centro de Lenguas para que pueda comenzar a operar.

También, se realizarán tareas de mejora continua para asegurar que el sistema trabaje de manera óptima, y mantenerlo en buen estado para que su uso pueda prolongarse por mucho tiempo.

Cuando un producto de software es entregado y ya se encuentra en marcha, el mantenimiento de este favorece la prolongación de su vida útil. Por ello, el trabajo futuro también incluye tareas de mantenimiento preventivo y correctivo del Sistema de proceso de evaluación de perfil lingüístico a fin de garantizar su correcta operatividad en todo momento. En caso de requerirlo, se llevarán a cabo tareas de mantenimiento perfectivo, para agregar o eliminar características o funciones según sea necesario; y adaptativo, para adaptar al software al ambiente de operación en el que se encuentre.

7. Referencias

Aguirre, S. (2020). *JSON - Vol.1: Primeros pasos - Sintaxis - Tipos de datos*. RedUsers.

Aguirre, S. (2021). *FRAMEWORK TOTAL - Vol.1: Crea APPs desde Cero con Laravel + Bootstrap + MySQL*. RedUsers.

Allende. (10 de marzo de 2021). *Creatividad Cloud*. Obtenido de Miro, plataforma para crear esquemas y diagramas: <https://www.creatividad.cloud/miro-plataforma-para-crear-esquemas-y-diagramas-visuales-de-forma-colaborativa-y-en-tiempo-real/>

Alvarez, M. A. (18 de septiembre de 2021). *desarrolloweb*. Obtenido de Qué es la programación orientada a objetos: <https://desarrolloweb.com/articulos/499.php>

Anónimo. (2 de diciembre de 2012). *Metodologías Ágiles de Desarrollo*. Obtenido de Adaptive Software Development (ASD): <https://agiliddeldesarrollo.wordpress.com/2012/12/02/adaptive-software-development-asd/>

Anónimo. (9 de abril de 2018). *Ingeniería de software*. Obtenido de Dynamic Systems Development Method (DSDM): <https://todosobreingenieriadesoftware.blogspot.com/2018/04/dynamic-systems-development-method-dsdm.html>

Apache Friends. (2022). *ApacheFriends.org*. Obtenido de Acerca de: <https://www.apachefriends.org/es/about.html>

AppMaster. (23 de febrero de 2023). *AppMaster*. Obtenido de Qué es Miro y cómo utilizarlo en su empresa: <https://appmaster.io/es/blog/que-es-miro-y-como-utilizarlo-en-su-empresa>

Arias, M. Á. (2013). *Introducción a PHP*. CreateSpace Independent Publishing Platform.

AWS Amazon. (s.f.). *AWS Amazon*. Obtenido de ¿Qué es JavaScript?: <https://aws.amazon.com/es/what-is/javascript/>

AWS Amazon. (s.f.). *AWS Amazon*. Obtenido de ¿Qué es SQL?: <https://aws.amazon.com/es/what-is/sql/>

7. REFERENCIAS

- B., G. (12 de julio de 2022). *Hostinger Tutoriales*. Obtenido de ¿Qué es CSS?: <https://www.hostinger.mx/tutoriales/que-es-css>
- B., G. (5 de abril de 2023). *Hostinger Tutoriales*. Obtenido de ¿Qué es jQuery? Introducción a la biblioteca jQuery para principiantes: <https://www.hostinger.mx/tutoriales/que-es-jquery>
- Beck, K. (2000). *Extreme Programming Explained*. Addison-Wesley Professional, 2000.
- Bedoya Laguna, C. A. (2017). *Diseño de un instrumento tipo escala Likert para la descripción de las actitudes hacia la tecnología por parte de los profesores de un colegio público de Bogotá*. Obtenido de <https://repository.udistrital.edu.co/bitstream/handle/11349/6881/BedoyaLagunaCarihstianAlberto2017.pdf?sequence=1>
- Benitez, M. Á., & Arias, Á. (2017). *Curso de Introducción a la Administración de Bases de Datos: 2ª Edición*. IT Campus Academy.
- Bisson, A.-C. (2018). *SQL. Los fundamentos del lenguaje : con ejercicios corregidos*. Ediciones ENI.
- Caballé Llobet, S., & Xhafa, F. (2007). *Aplicaciones distribuidas en Java con tecnología RMI*. Delta Publicaciones.
- Carrión Bou, R. (2019). *Usando XAMPP con Bootstrap y WordPress*. Mercedes Gómez Alcalá.
- Castañón García, G. Y., Hernández Cortés, D. M., Hernández Hernández, C. L., Amador Pliego, E., Alamilla Martínez, F., Hernández Téllez, J. A., . . . Espino Barranco, V. (noviembre de 2013). *Centro de Lenguas*. Obtenido de Programa Institucional de Lenguas: <https://www.uaeh.edu.mx/cel/pil.html>
- Celaya Luna, A. (2019). *Creación de páginas Web: HTML 5*. ICB, S.L. (Interconsulting Bureau S.L.).
- Centro de Aprendizaje de Google Workspace. (2023). *Centro de Aprendizaje de Google Workspace*. Obtenido de Empezar a usar Formularios en Google Workspace. Qué puedes hacer con Formularios: <https://support.google.com/a/users/answer/9302965?hl=es>
- Cervantes O., J., Gómez F., M. d., Gonzáles P., P. P., & García N., A. (2016). *UAM*. Universidad Autónoma Metropolitana, Unidad Cuajimalpa. Obtenido de Introducción: https://www.cua.uam.mx/pdfs/revistas_electronicas/libros-electronicos/2016/2intro-poo/programacion_web.pdf

7. REFERENCIAS

- Chua, B. B., & Dyson, L. E. (2004). Applying the ISO 9126 model to the evaluation of an e-learning system. In Proc. of ASCILITE (Vol. 5, No. 8, pp. 184-190).
- Cognizant. (2022). *Cognizant*. Obtenido de Plataforma digital: <https://www.cognizant.com/es/es/glossary/digital-platform>
- De La Garza, S. (13 de abril de 2021). *Aleph*. Obtenido de ¿Qué es una plataforma en la web?: <https://aleph.org.mx/que-es-una-plataforma-en-la-web>
- desarrolloweb. (2023). *desarrolloweb*. Obtenido de Laravel: <https://desarrolloweb.com/home/laravel>
- Dinder, M. (2022). *Becoming an Enterprise Django Developer: Discover best practices, tooling, and solutions for writing and organizing Django applications in production*. Packt Publishing Ltd.
- DUI-UAEH. (2013). Universidad Autónoma del Estado de Hidalgo. Obtenido de Centro de Lenguas: <https://www.uaeh.edu.mx/cel/pil.html>
- Dvorski, D. D. (marzo de 2007). *Installing, configuring, and developing with XAMPP*. Obtenido de <http://dalibor.dvorski.net/downloads/docs/InstallingConfiguringDevelopingWithXAMPP.pdf>
- EBAC. (1 de junio de 2023). *EBAC*. Obtenido de ¿Qué es HTML5 y para qué sirve?: <https://ebac.mx/blog/que-es-html5>
- Equipo de datos.gob.es. (3 de mayo de 2022). *datos.gob.es*. Obtenido de 11 librerías para crear visualizaciones de datos: <https://datos.gob.es/es/blog/11-librerias-para-crear-visualizaciones-de-datos>
- Equipo de redacción de Drew. (3 de diciembre de 2019). *Drew*. Obtenido de Ventajas y desventajas de la metodología Scrum: <https://blog.wearedrew.co/productividad/-ventajas-y-desventajas-de-la-metodologia-scrum>
- Escuela de Profesores del Perú. (2023). *Escuela de Profesores del Perú*. Obtenido de Evaluación educativa, características, funciones, etapas y más.: <https://epperu.org/evaluacion-educativa-caracteristicas-funciones-etapas/>
- Espacios Media. (2022). *Espacios Media*. Obtenido de ¿Qué es una Plataforma Digital?: <https://www.espacios.media/que-es-una-plataforma-digital/>
- Fazt Web. (4 de octubre de 2021). *GitHub*. Obtenido de javascript-quiz-oop: <https://github.com/FaztWeb/javascript-quiz-oop>

7. REFERENCIAS

Fernández, L. (2015). *JavaScript*. Lulu.com.

GitHub. (2023). Obtenido de Visual Studio Code - Open Source ("Code - OSS"):
<https://github.com/microsoft/vscode>

Guagliano, C. (2019). *Programación en Python II: Programación orientada a objetos*. RedUsers.

Hernández, U. (22 de febrero de 2015). *códigofacilito*. Obtenido de MVC (Model, View, Controller) explicado: <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>

inLab FIB. (2022). *inLab FIB*. Obtenido de Descubriendo GitLab:
<https://inlab.fib.upc.edu/es/blog/descubriendo-gitlab>

INNOCAN. (2022). *INNOCAN*. Obtenido de Conceptos básicos:
<https://innocan.com/help/gexcat/index.html>

IONOS. (26 de octubre de 2018). *Digital Guide Ionos*. Obtenido de UML, lenguaje de modelado gráfico: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>

IONOS. (21 de enero de 2020). *IONOS*. Obtenido de Extreme Programming: desarrollo ágil llevado al extremo: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/extreme-programming/>

IONOS. (29 de septiembre de 2022). *IONOS*. Obtenido de Tutorial de GitLab: instalación y primeros pasos: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/tutorial-de-gitlab/>

IONOS. (1 de marzo de 2023). *Digital Guide IONOS*. Obtenido de Instala tu servidor local XAMPP en unos pocos pasos: <https://www.ionos.mx/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/>

IONOS. (1 de marzo de 2023). *IONOS Digital Guide*. Obtenido de Las librerías y los frameworks JavaScript más populares: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/frameworks-javascript-y-librerias-populares/#:~:text=JavaScript%2C%20también%20conocido%20por%20su,la%20programación%20procesual%20o%20funcional.>

Jackson, W. (2016). *JSON Quick Syntax Reference*. Apress.

7. REFERENCIAS

- Juárez, I. (2020). *Platzi*. Obtenido de Diagramas Entidad Relación (ER): <https://platzi.com/tutoriales/1566-bd/9341-diagramas-entidad-relacion-er/>
- Kahlert, T., & Giza, K. (marzo de 2016). *Microsoft*. Obtenido de Visual Studio Code - Tips & Tricks Vol. 1: <https://download.microsoft.com/download/8/A/4/8A48E46A-C355-4E5C-8417-E6ACD8A207D4/VisualStudioCode-TipsAndTricks-Vol.1.pdf>
- Laravel. (2023). *Laravel*. Obtenido de Installation: <https://laravel.com/docs/9.x>
- Liderlogo. (s.f.). *Liderlogo*. Obtenido de Pixabay, qué es y para qué sirve en el diseño web: <https://www.liderlogo.es/disenio-web/pixabay-que-es/>
- Lizcano Bueno, L. I. (s.f.). *UML: Un Lenguaje de Modelo de Objetos*. Obtenido de <https://dialnet.unirioja.es/descarga/articulo/5555261.pdf>
- López Ortega, D., & Santa Villa, J. A. (2012). *Recursos Biblioteca UTP*. Obtenido de Estudio Comparativo de las Herramientas CASE: StarUML, Poseidon for UML y Enterprise Architect, para el modelamiento de diagramas UML: <http://recursosbiblioteca.utp.edu.co/tesis/textoyanexos/0053L864e.pdf>
- Lucidchart. (2023). *Lucidchart*. Obtenido de ¿Qué es un modelo entidad relación?: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion>
- Lucidchart. (2023a). *Lucidchart*. Obtenido de Tutorial de diagrama de clases UML: <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>
- Lucidchart. (2023b). *Lucidchart*. Obtenido de Tutorial de diagrama de objetos: <https://www.lucidchart.com/pages/es/diagrama-de-objetos-uml>
- Lucidchart. (2023c). *Lucidchart*. Obtenido de Component Diagram Tutorial: <https://www.lucidchart.com/pages/uml-component-diagram>
- Lucidchart. (2023d). *Lucidchart*. Obtenido de Tutorial de diagramas de despliegue: <https://www.lucidchart.com/pages/es/tutorial-de-diagramas-de-despliegue>
- Lucidchart. (2023e). *Lucidchart*. Obtenido de Tutorial de diagrama de actividades UML: <https://www.lucidchart.com/pages/es/tutorial-diagrama-de-actividades-uml>
- Lucidchart. (2023f). *Lucidchart*. Obtenido de UML Use Case Diagram Tutorial: <https://www.lucidchart.com/pages/uml-use-case-diagram>
- Lucidchart. (2023g). *Lucidchart*. Obtenido de Timing Diagram Tutorial: <https://www.lucidchart.com/pages/uml-timing-diagram>

7. REFERENCIAS

- Lucidchart. (2023h). *Lucidchart*. Obtenido de Qué es el lenguaje unificado de modelado (UML): https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#section_10
- Lucidchart. (2023i). *Lucidchart*. Obtenido de Communication Diagram Tutorial: <https://www.lucidchart.com/pages/uml-communication-diagram>
- Luna, F. (2019). *JavaScript - Aprende a programar en el lenguaje de la web*. RedUsers.
- Marqués, A. M., Aliaga, E. J., García, G. S., & Quintana, O. G. (2001). *SQL y desarrollo de aplicaciones en Oracle 8*. Publicacions de la Universitat Jaume I, 2001.
- Martínez Villalobos, G., Camacho Sánchez, G. D., & Biancha Gutiérrez, D. A. (abril de 2010). *Redalyc*. Obtenido de Diseño de Framework web para el desarrollo dinámico de aplicaciones: <https://www.redalyc.org/pdf/849/84917316032.pdf>
- MDN contributors. (5 de diciembre de 2022). *MDN web docs*. Obtenido de MVC: <https://developer.mozilla.org/es/docs/Glossary/MVC>
- MDN contributors. (21 de julio de 2023). *MDN web docs*. Obtenido de ¿Qué es JavaScript?: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- MDN contributors. (21 de julio de 2023). *MDN web docs*. Obtenido de Trabajando con JSON: <https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>
- MDN contributors. (24 de julio de 2023). *MDN web docs*. Obtenido de HTML: Lenguaje de etiquetas de hipertexto: <https://developer.mozilla.org/es/docs/Web/HTML>
- MDN contributors. (13 de marzo de 2023). *MDN web docs*. Obtenido de CSS: <https://developer.mozilla.org/es/docs/Web/CSS>
- Microsoft. (2022). *Support Microsoft*. Obtenido de Access SQL: conceptos básicos, vocabulario y sintaxis: <https://support.microsoft.com/es-es/office/access-sql-conceptos-básicos-vocabulario-y-sintaxis-444d0303-cde1-424e-9a74-e8dc3e460671#bm1>
- Microsoft 365 Team. (24 de septiembre de 2019). *Microsoft*. Obtenido de La guía sencilla para la diagramación de UML y el modelado de la base de datos: <https://www.microsoft.com/es-mx/microsoft-365/business-insights->

7. REFERENCIAS

- ideas/resources/guide-to-uml-diagramming-and-database-modeling#:~:text=El%20Lenguaje%20Unificado%20de%20Modelado,de%20un%20sistema%20o%20proceso.
- Miro. (s.f.). *Miro*. Obtenido de Diagrama de clases UML: <https://miro.com/es/diagrama/que-es-diagrama-clases-uml/>
- Moodle. (26 de diciembre de 2022). *Moodle*. Obtenido de Acerca de Moodle: https://docs.moodle.org/all/es/Acerca_de_Moodle
- Mozilla. (8 de junio de 2023). *MDN*. Obtenido de Database: <https://developer.mozilla.org/en-US/docs/Glossary/Database>
- Peña Millahual, C. (2019). *PHP 7 - Sitios Dinámicos: Aprenda a programar sin conocimientos previos*. RedUsers, 2019.
- Pressman, R. S. (2010). *Ingeniería de Software. Un enfoque práctico*. McGraw Hill.
- Ramesh, P. (21 de septiembre de 2018). *Packt Hub*. Obtenido de Meet Sublime Merge, a new Git client from the makers of Sublime Text: <https://hub.packtpub.com/meet-sublime-merge-a-new-git-client-from-the-makers-of-sublime-text/>
- Rodríguez Antonio, R. (07 de septiembre de 2020). *umVirtual*. Obtenido de Cinco aplicaciones educativas para profesores innovadores: <https://umvirtual.org/cinco-aplicaciones-educativas-para-profesores-innovadores/#:~:text=Testmoz%20es%20una%20aplicación%20educativa,r espuestas%20y%20preguntas%20de%20desarrollo>).
- Rubiales Gómez, M. (2021). *Curso de desarrollo Web. HTML, CSS y JavaScript. Edición 2021*. Comercial Grupo ANAYA, S.A., 2021.
- Schwaber, K., & Sutherland, J. (noviembre de 2020). *Scrum Guides*. Obtenido de La Guía Scrum: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- Sean. (20 de septiembre de 2023). *Miro Help Center*. Obtenido de ¿Qué es Miro?: <https://help.miro.com/hc/es/articles/360017730533--Qué-es-Miro->
- Skinner, J. (20 de septiembre de 2018). *Sublime HQ*. Obtenido de Sublime Merge - Git, Done Sublime: <https://www.sublimetext.com/blog/articles/sublime-merge>
- Sommerville, I. (2011). *Ingeniería de Software*. México: Pearson.

7. REFERENCIAS

- StarUML.io. (2022). *StarUML Documentation*. Obtenido de Introduction: <https://docs.staruml.io/user-guide/readme>
- Sublime Merge. (s.f.). *Sublime Merge*. Obtenido de Documentation FAQ: <https://www.sublimemerge.com/docs/faq>
- Sublime Merge. (s.f.). *Sublime Merge*. Obtenido de Buy Sublime Merge: <https://www.sublimehq.com/store/merge>
- Testmoz. (2022). *Testmoz*. Obtenido de Features: <https://testmoz.com/features>
- ThePHPGroup. (s.f.). *PHP*. Obtenido de ¿Qué es PHP?: <https://www.php.net/manual/es/intro-whatis.php>
- ThePHPGroup. (s.f.). *PHP*. Obtenido de ¿Qué puede hacer PHP?: <https://www.php.net/manual/es/intro-whatcando.php>
- Tokio School. (29 de junio de 2020). *Tokio School*. Obtenido de ¿Qué es HTML y para qué sirve?: <https://www.tokioschool.com/noticias/historia-lenguaje-html/>
- Torrado, J. (12 de octubre de 2017). *Desarrollo Web*. Obtenido de Qué es GitLab, qué le diferencia de sus competidores como GitHub o Bitbucket y qué herramientas engloba, además de los repositorios remotos Git, para gestionar los proyectos de desarrollo.: <https://desarrolloweb.com/articulos/introduccion-gitlab.html>
- Ubunlog. (2022). *Ubunlog*. Obtenido de Llega la nueva versión de Sublime Text 3.2 y estas son sus novedades: <https://ubunlog.com/llega-la-nueva-version-de-sublime-text-3-2-y-estas-son-sus-novedades/#:~:text=Sublime%20Merge%20combina%20el%20motor,la%20funcionalidad%20de%20búsqueda%20avanzada.>
- Universidad Europea. (24 de agosto de 2022). *Universidad Europea*. Obtenido de Programación orientada a objetos: <https://universidadeuropea.com/blog/programacion-orientada-objetos/>
- V., B. (31 de enero de 2023). *Hostinger*. Obtenido de Diferencias entre los 3 tipos de estilos CSS: Interno, Externo e Inline: <https://www.hostinger.mx/tutoriales/tipos-de-estilos-css>
- Villarreal Fuentes, C. A. (noviembre de 2013). *Northware*. Obtenido de ¿Qué es jQuery?: <https://www.northware.mx/wp-content/uploads/2021/04/que-es-jquery.pdf>

7. REFERENCIAS

Zamenfeld, S. (1 de julio de 2011). *Brainlabs*. Obtenido de StarUML una herramienta para modelado: <https://www.brainlabs.com/novedad/staruml-una-herramienta-para-modelado/>

8. Apéndices

8.1 A. Manual de Usuario: Plataforma para la creación de exámenes aplicados a los alumnos en el Centro de Lenguas.

En esta sección se muestran las diferentes funciones que el Sistema de proceso de evaluación de perfil lingüístico ofrece. En caso de no encontrar alguna, se sugiere contactar con el administrador.

8.1.1 Escritorio de la plataforma

Esta interfaz de la plataforma, mostrada en la figura 47, corresponde al menú principal del sistema. En ella pueden observarse 3 opciones:

- **Ver alumnos.** Al hacer clic, direcciona al usuario al listado de alumnos donde se presentan datos personales de ellos y de contacto.
- **Crear nuevo examen.** Al hacer clic, direcciona al usuario al formulario de creación de exámenes.
- **Exámenes creados.** Al hacer clic, direcciona al usuario al listado de exámenes creados previamente.

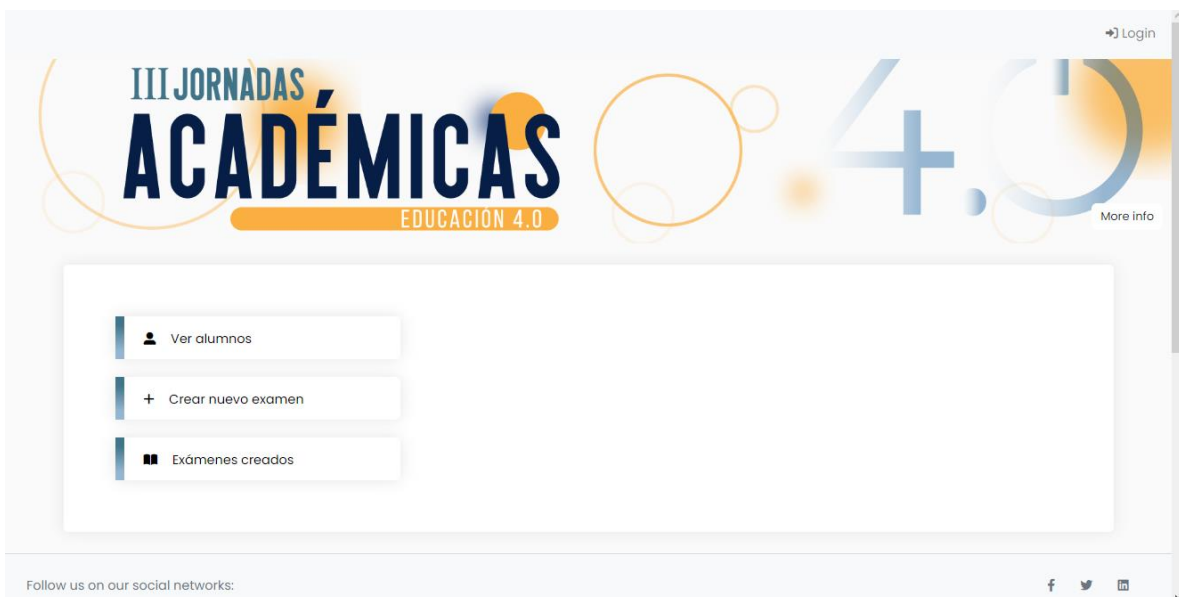


FIGURA 47. ESCRITORIO DEL SISTEMA

8.1.2 Ver alumnos

Esta vista tiene como propósito mostrar el listado de los alumnos registrados por el Centro de Lenguas de la UAEH. Este listado se presenta en forma de tabla para que su lectura sea más simple y fluida y, por cada alumno, se detalla su nombre completo con apellidos, su número de cuenta, su correo institucional y un número telefónico de contacto. Esta ventana, mostrada en la figura 48, también incluye un botón ubicado en la parte inferior derecha que le permite regresar al escritorio del sistema o menú principal.



ID alumno	Apellido paterno	Apellido materno	Nombre	Correo	Teléfono
313548	Pérez	Pérez	Juan	pe313548@uach.edu.mx	7719465
328174	Smith	Smith	John	sm328174@uach.edu.mx	7725764

FIGURA 48. LISTA DE ALUMNOS

8.1.3 Crear un nuevo examen

Para acceder a esta ventana, basta con ir al escritorio del sistema y seleccionar la opción de *Crear nuevo examen*.

La ventana que permite la creación de exámenes presenta un formulario que deberá ser llenado en su totalidad. En caso de que se deje por accidente algún campo sin llenar o con caracteres o valores erróneos, se notificará mediante mensajes de alerta indicando el problema.

Los campos que incluye el formulario para la creación de evaluaciones en línea son los siguientes:

8. APÉNDICES

1. *Hora Inicio.* Al hacer clic en el ícono del reloj, se abrirá un panel en el que se deberá marcar la hora en la que dará inicio el examen.
2. *Hora de finalización.* Al hacer clic en el ícono del reloj, se abrirá un panel en el que se deberá marcar la hora en la que concluirá el examen.
3. *Fecha de inicio.* Al hacer clic en el ícono del calendario, se abrirá un panel que permitirá seleccionar la fecha en la que la aplicación del examen dará inicio.
4. *Fecha de finalización.* Al hacer clic en el ícono del calendario, se abrirá un panel que permitirá seleccionar la fecha en la que la aplicación del examen finalizará.
5. *Duración.* Se deberá indicar con horas, minutos y segundos el tiempo total que durará la aplicación del examen.
6. *Número máximo de alumnos.* Se deberá indicar, con número, el número máximo de alumnos que pueden aplicar el examen.
7. *Número de preguntas.* Se deberá indicar, con número, la cantidad de preguntas totales que el examen tendrá.
8. *Habilidades a evaluar.* Al hacer clic se abrirá una lista desplegable en la cual podrá seleccionar la habilidad que el examen evaluará.
9. *Tipo/Propósito.* Al hacer clic se abrirá una lista desplegable en la cual podrá seleccionar el propósito para el que estará destinado el examen.
10. *Modalidad.* Al hacer clic se abrirá una lista desplegable en la cual podrá seleccionar la modalidad en la que se aplicará el examen.
11. *Semestre.* Al hacer clic se abrirá una lista desplegable en la cual podrá seleccionar el semestre para el que estará dirigido el examen.
12. *Idioma.* Al hacer clic se abrirá una lista desplegable en la cual podrá seleccionar el idioma que evaluará el examen.
13. *Materia.* Al hacer clic se abrirá una lista desplegable en la cual podrá seleccionar la asignatura a la que corresponderá el examen.
14. *Unidad.* Al hacer clic se abrirá una lista desplegable en la cual podrá seleccionar la unidad del programa temático que evaluará la asignatura.

En la figura 49 se aprecia un ejemplo de llenado de varios de estos campos para crear un nuevo examen. En caso de que se llegase a introducir un valor erróneo o inválido, el sistema lo notificará mediante mensajes de alerta y le indicará el problema a corregir debajo del campo donde se introdujo el valor.

8. APÉNDICES

FECHA Y HORA	Hora de inicio: 10:00 a. m. <input type="text"/>	Hora de finalización: 11:00 a. m. <input type="text"/>
	Fecha de inicio: 26/09/2023 <input type="text"/>	Fecha de finalización: 26/09/2023 <input type="text"/>
	Duración: 1 : 30 : 0 Horas Minutos Segundos	
<hr/>		
ALUMNO	Número máximo de alumnos: 40 <input type="text"/>	
<hr/>		
ESPECIFICACIONES	Número de preguntas: 24 <input type="text"/>	Tipo / Propósito: Parcial <input type="text"/>

FIGURA 49. EJEMPLO DE LLENADO DE FORMULARIO PARA CREAR UN NUEVO EXAMEN

Es importante mencionar que, si se selecciona el tipo de examen *Egreso* en el campo de *Tipo/Propósito*, no será posible completar los campos de *Semestre*, *Materia* y *Unidad*. La razón de ello es que, si un alumno está próximo a presentar un examen con fines de egreso, es porque no se encuentra cursando algún semestre y, por ende, no tiene asociada una asignatura y una unidad de programa temático. En la figura 50 se observa que cuando se selecciona el tipo de examen “Egreso”, en automático se bloquean los campos ya mencionados.

ESPECIFICACIONES	Número de preguntas: Ej: 20 <input type="text"/>	Tipo / Propósito: Egreso <input type="text"/>
	Habilidad a evaluar: Elegir Habilidad <input type="text"/>	Modalidad: Elegir Modalidad <input type="text"/>
	Semestre: Elegir Semestre <input type="text"/>	
<hr/>		
IDIOMA Y ASIGNATURA	Idioma: Elegir Idioma <input type="text"/>	Materia: Elegir Materia <input type="text"/>
	Unidad: Elegir Unidad <input type="text"/>	

FIGURA 50. CAMPOS DISPONIBLES Y NO DISPONIBLES PARA LA CREACIÓN DE UN EXAMEN CON FINES DE EGRESO

Una vez que todos los campos sean llenados correctamente, el examen estará listo para ser creado haciendo clic en el botón llamado *Crear examen* ubicado en la parte inferior de la página. Posteriormente y de forma automática, se le direccionará al listado de exámenes en donde podrá visualizar en forma de lista el examen que acaba de crear, sus características y algunas acciones que puede ejercer sobre ese examen.

8.1.4 Abrir listado de exámenes creados

Para acceder a esta ventana, basta con ir al escritorio del sistema y seleccionar la opción *Ver exámenes creados*.

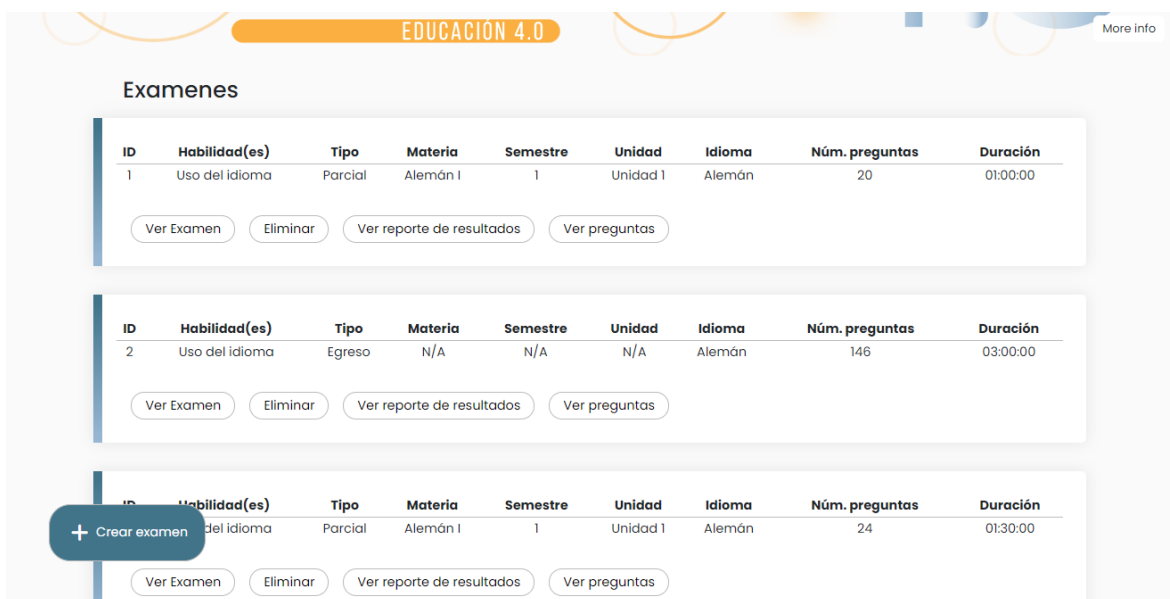
Si no existe ningún examen creado, se le presentará un mensaje indicando que no existe ninguna evaluación registrada y se le invitará a comenzar creando una nueva. Una vez que se tenga registro de por lo menos un examen, cada examen será listado mostrando información sobre él en forma de tabla y tendrá sus propios botones de acción. A fin de que cada examen pueda ser identificado, se presentan los siguientes datos en forma de tabla en cada uno:

- a. *ID*. Número de identificación del examen.
- b. *Habilidades*. Especifica las habilidades que evalúa el examen.
- c. *Tipo/Propósito*. Indica el tipo de examen o el propósito para el que fue creado.
- d. *Materia*. Indica la materia o asignatura para la que fue creado.
- e. *Semestre*. Indica el semestre al que va dirigido.
- f. *Unidad*. Unidad que evalúa el examen.
- g. *Idioma*. Idioma del examen.
- h. *Núm. de preguntas*. Número total de preguntas que contiene el examen.
- i. *Duración*. Tiempo total del examen para ser respondido.

Además, en cada uno de ellos se añaden 4 botones:

- a. *Ver examen*. Permite visualizar el examen de la forma en cómo lo vería el alumno en el momento de responderlo.
- b. *Eliminar*. Permite eliminar ese examen.
- c. *Ver reportes de resultados*. Permite generar reportes con los resultados registrados por los alumnos.
- d. *Ver preguntas*. Esta opción le permite ver un nuevo listado en el cual se compone de todas las preguntas que le fueron asignadas al examen tomando como base las características especificadas en el formulario para crear el examen.

Un ejemplo de lo anterior puede ser apreciado en la figura 51.



EDUCACIÓN 4.0

More info

Exámenes

ID	Habilidad(es)	Tipo	Materia	Semestre	Unidad	Idioma	Núm. preguntas	Duración	
1	Uso del idioma	Parcial	Alemán I	1	Unidad 1	Alemán	20	01:00:00	
Ver Examen Eliminar Ver reporte de resultados Ver preguntas									
2	Uso del idioma	Egreso	N/A	N/A	N/A	Alemán	146	03:00:00	
Ver Examen Eliminar Ver reporte de resultados Ver preguntas									
+	Crear examen	Uso del idioma	Parcial	Alemán I	1	Unidad 1	Alemán	24	01:30:00
Ver Examen Eliminar Ver reporte de resultados Ver preguntas									

FIGURA 51. LISTADO DE EXÁMENES CREADOS

8.1.4.1 Visualizar examen

Al presionar el botón “Ver examen”, se direccionará al usuario a la ventana donde se le mostrará el examen de la misma forma en que los alumnos podrán visualizarlo en el momento que ellos realicen sus evaluaciones.

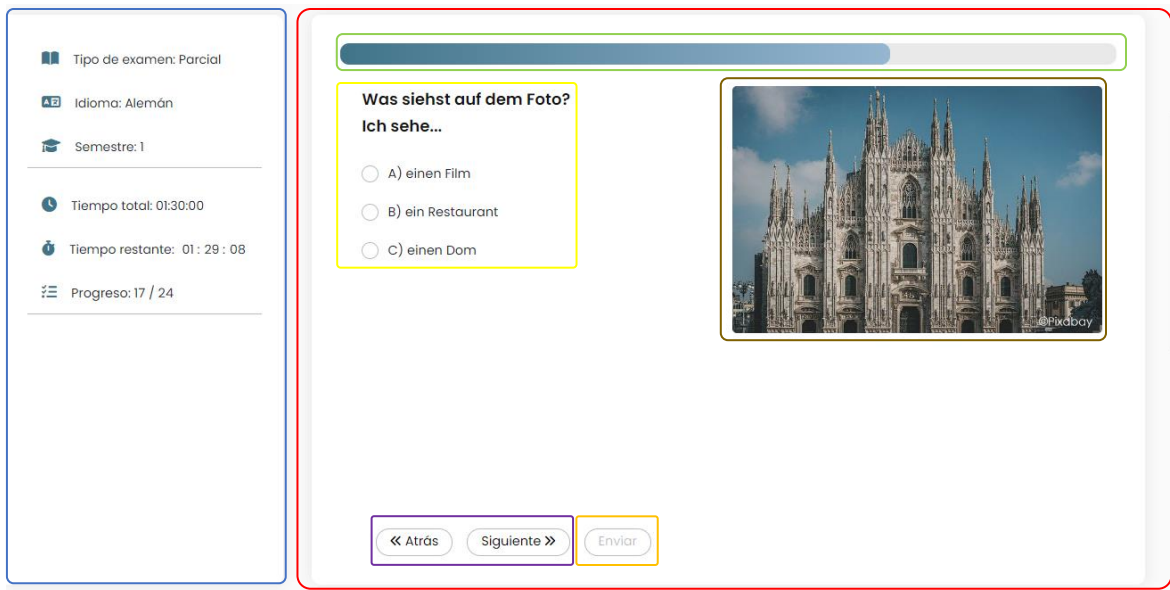


FIGURA 52. VISTA DE LA VENTANA DONDE EL ALUMNO PODRÁ REALIZAR SU EXAMEN

Esta ventana está conformada por dos paneles. El primero, ubicado del lado izquierdo y marcado con color azul en la figura 52, expresa información general del examen en 3 partes:

1. Datos generales, como la asignatura a la está dirigido el examen, el idioma que evalúa, y el semestre en el que se encuentra actualmente el alumno, el cual se indica mediante el número correspondiente o “N/A” que significa que no aplica (como es el caso de los alumnos egresados).
2. El tiempo total disponible y el tiempo restante para responder la evaluación.
3. El progreso, que muestra la pregunta en la que el alumno se encuentra situado.

El segundo panel, marcado con color rojo, ubicado del lado derecho, sirve de contenedor para el examen en sí. Este panel está conformado por una barra de progreso que, al igual que el indicador del panel izquierdo, incrementa o decrementa dependiendo del número de pregunta en la que el alumno se encuentra ubicado.

Barra de progreso (zona verde).

La barra de progreso en la parte superior indica de manera gráfica en qué pregunta se encuentra a cada momento. Conforme el alumno avance o retroceda en las preguntas, el tamaño de la barra color azul incrementará o decrementará.

Pregunta y respuesta (zona amarilla).

En esta zona se encuentran la pregunta actual y las posibles opciones de respuesta en función del tipo de pregunta y siempre que el tipo de pregunta aplique. Si es una pregunta de opción múltiple, las opciones se presentan como *radio buttons* o mediante una lista desplegable según sea el caso; si es una pregunta de selección múltiple, las opciones se enlistan para su selección con casillas; en caso de que la pregunta sea de tipo abierta, se añadirá un cuadro de texto donde el alumno podrá ingresar su respuesta.

Elemento multimedia (zona marrón)

En caso de que la pregunta actual tenga un elemento multimedia relacionado, será mostrado del lado izquierdo de la pregunta. Si se trata de un video o un audio, este podrá ser reproducido únicamente un par de veces.

Enviar examen (zona naranja)

Una vez finalizado el examen, el estudiante podrá enviar su examen haciendo clic en el botón llamado *Enviar*. Si el alumno aún tiene preguntas sin responder, el examen se lo notificará y podrá volver a su evaluación para darles respuesta siempre que aún haya tiempo restante disponible. En caso contrario, se le pedirá que confirme su envío. Si acepta, el examen será enviado para ser calificado.

Si el alumno agota su tiempo restante disponible para contestar su examen, el sistema se lo notificará en un mensaje y enviará el examen para ser calificado, aún si existen preguntas sin respuesta.

Navegación (zona morada)

Para avanzar y retroceder a través de las preguntas, el alumno podrá utilizar los botones de Atrás y Siguiente ubicados en la parte inferior izquierda del examen.

8.1.4.2 Eliminar un examen

Para eliminar un examen, basta con posicionarse en el examen que se desea eliminar y presionar el botón de *Eliminar*. El sistema le mostrará un mensaje de confirmación que preguntará si realmente desea eliminar ese examen. Si la respuesta es afirmativa, el examen será eliminado junto con los resultados obtenidos por los alumnos, y le será notificado que el examen ha sido eliminado exitosamente. En la figura 53 se observa la ventana de confirmación mencionada que aparece en cuanto se intenta eliminar un examen.

8. APÉNDICES

localhost dice
¿Eliminar este examen?

Aceptar Cancelar

ID	Habilidad(es)	Tipo	Materia	Semestre	Unidad	Idioma	Núm. preguntas
2	Uso del idioma	Egreso	N/A	N/A	N/A	Alemán	146

FIGURA 53. MENSAJE DE CONFIRMACIÓN PARA ELIMINAR UN EXAMEN

8.1.4.3 Ver reportes de resultados

Para ver los reportes de los resultados obtenidos por los alumnos, basta con acceder al listado de exámenes y presionar el botón *Ver reporte de resultados* del examen del cual se desea conocer esta información. Así, el sistema lo redireccionará a la ventana donde podrá consultar esta información ordenada de la forma que el usuario requiera, ya sea por día, mes o semana, y con la posibilidad de imprimirla. En la figura 54 se muestra un ejemplo de los resultados obtenidos por un alumno en un determinado examen ordenado por semana.

Ordenar los datos por: Semana

Resumen

ID examen:	2
Fecha de inicio:	2023-10-11
Fecha de término:	2023-10-11
Habilidades evaluadas:	Uso del idioma
Tipo/propósito:	Parcial
Asignatura:	Alemán I
Semestre:	1
Idioma:	Alemán

Resultados

Fecha: 8/10/2023 hasta 14/10/2023

ID alumno	Apellido paterno	Apellido materno	Nombre	Calificación	Fecha de entrega
328174	Smith	Smith	John	3.5	11/10/2023, 20:06

Imprimir Volver al listado

FIGURA 54. REPORTE DE RESULTADOS ORDENADO POR SEMANA

8.1.4.4 Abrir listado de preguntas asignadas al examen

El siguiente botón corresponde al del listado de preguntas asociadas a un examen. Al hacer clic en este botón, se direccionará a una nueva ventana, mostrada en la figura 55, en donde se presenta información de cada una de las preguntas asignadas al examen de manera general en forma de tabla.

La información de cada pregunta dentro de la tabla es la siguiente:

- ID*. Identificador de cada pregunta asignada al examen.
- Pregunta*. Muestra el texto de la pregunta.
- Opciones*. Muestra las posibles respuestas que puede tener la pregunta (excepto en las preguntas de tipo abierta).
- Respuesta*. Muestra la respuesta correcta de la pregunta (excepto en las preguntas de tipo abierta).

Además, se añade un botón de *Ver detalles* en cada pregunta, el cual le redireccionará a una nueva página en donde se muestran más detalles de la pregunta (Ver sección A.4.4.1).

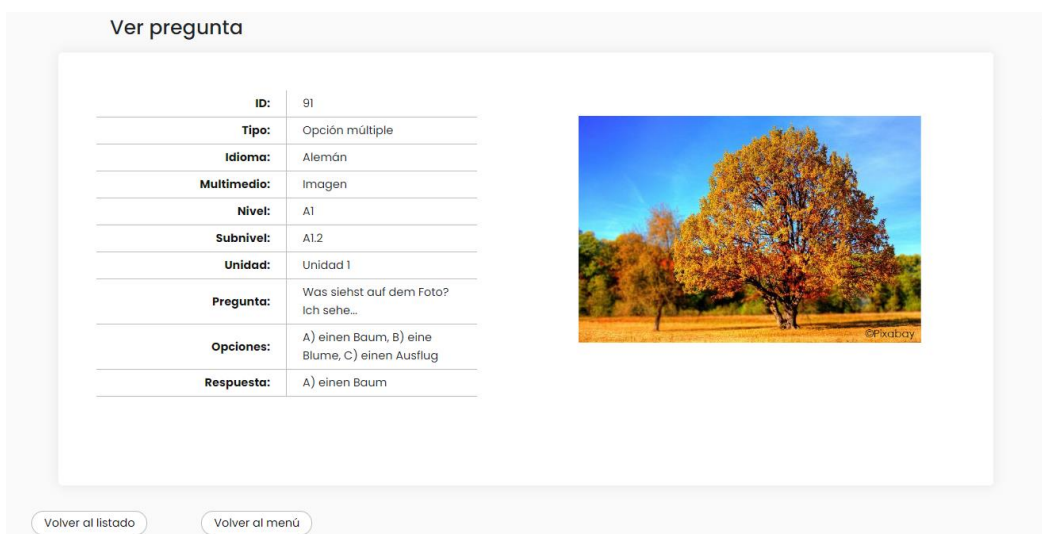
Preguntas asignadas

ID pregunta	Pregunta	Opciones	Respuesta	Acciones
26	Was siehst auf dem Foto? Ich sehe...	A) ein Telefon, B) einen Schalter, C) einen Geburtstag	C) einen Geburtstag	Ver detalles
29	Ergänzen Sie das passende Wort. die Adresse - die Straße - der Absender - das Land - die Briefmarke - die Stadt - die Hausnummer - der Empfänger - die Postleitzahl.			Ver detalles
30	Ergänzen Sie das passende Wort. die Adresse - die Straße - der Absender - das Land - die Briefmarke - die Stadt - die Hausnummer - der Empfänger - die Postleitzahl.			Ver detalles
41	Sie können mich auf dem	A) bar, B) Handy, C) Pass, D) schreiben	A) bar	Ver detalles
42	Ich habe keine Kreditkarte, ich zahle	A) bar, B) Handy, C) Pass, D) schreiben	B) Handy	Ver detalles
44	Was siehst auf dem Foto? Ich sehe...	A) die Kreditkarte, B) die Putzmittel, C) die Lebensmittel	C) die Lebensmittel	Ver detalles
45	Was siehst auf dem Foto? Ich sehe...	A) eine Verkäuferin, B) eine Kundin, C) ein Angebot	A) eine Verkäuferin	Ver detalles
52	Wo sind die Leute? Schreiben Sie den Ort im Restaurant - im Supermarkt - im Buchladen - im Obst- und Gemüseladen - im Café - in der Bäckerei - auf der Post - auf der Bank - im Möbelgeschäft Ich brauche fünf Briefmarken.			Ver detalles
55	Was für ein ...	A) Im 4. Stock, B) Ja, aber nicht in dieser ... C) ...	B) Ja, aber nicht in	Ver

FIGURA 55. EJEMPLO DE LISTADO DE LAS PREGUNTAS DE UN EXAMEN

8.1.4.4.1 Visualizar detalles de una pregunta

Como se explicó en el punto anterior, cada pregunta cuenta con un botón de *Ver detalles*, el cual direccionará a una nueva ventana donde se presenta información más detallada de la pregunta. En la figura 56 se puede observar un ejemplo de los detalles de una pregunta.



The screenshot shows a window titled "Ver pregunta" with a table of details and an image of a tree. The table contains the following information:

ID:	91
Tipo:	Opción múltiple
Idioma:	Alemán
Multimedia:	Imagen
Nivel:	A1
Subnivel:	A1.2
Unidad:	Unidad 1
Pregunta:	Was siehst auf dem Foto? Ich sehe...
Opciones:	A) einen Baum, B) eine Blume, C) einen Ausflug
Respuesta:	A) einen Baum

The image on the right shows a large tree with autumn foliage in a field under a blue sky. The image is credited to "SPixabay".

At the bottom of the window, there are two buttons: "Volver al listado" and "Volver al menú".

FIGURA 56. INFORMACIÓN DE UNA PREGUNTA EN ESPECÍFICO

Esta información consta de lo siguiente:

- ID*. Identificador único de la pregunta.
- Tipo*. Indica el tipo o propósito del examen.
- Idioma*. Especifica el idioma al que pertenece la pregunta.
- Multimedia*. Indica el tipo de multimedia que tiene asociado (en caso de que lo tuviera).
- Nivel*. Indica el nivel al que pertenece la pregunta.
- Subnivel*. Indica el subnivel al que pertenece la pregunta.
- Unidad*. Especifica la unidad del programa temático al que pertenece la pregunta.
- Pregunta*. Texto de la pregunta.
- Opciones*. Muestra las posibles respuestas que puede tener una pregunta (en caso de que aplicara por el tipo de pregunta).
- Respuesta*. Indica la respuesta correcta a la pregunta.

En caso de que la pregunta tuviese un elemento multimedia asociado, se mostrará del lado derecho de la tabla con la información de la pregunta y, a diferencia de la presentación de los multimedia en el examen, los audios y videos en esta ventana no tienen un número limitado de reproducciones.

8.1.5 Ver resultados

En este módulo, es posible acceder a los resultados obtenidos por un alumno en su examen de dos formas. La primera forma es visualizar los resultados en el momento que se envía el examen: en cuanto se presiona el botón enviar y se confirma su envío, el sistema realizará las operaciones necesarias para obtener la calificación y sugerencias de estudio, y serán mostradas junto con la retroalimentación de la evaluación en una nueva ventana. La segunda forma de ver los resultados es cuando el examen ya ha sido enviado y se intenta acceder de nuevo al examen. Si se da clic en el botón *Ver Examen* en la ventana del listado de exámenes, en lugar de abrir el examen, se mostrará un mensaje que indica que el examen ya ha sido resuelto pero que es posible ver sus resultados presionando el botón de *Ver resultados*. Esta situación puede observarse en la figura 57. Si este botón es presionado, se mostrará la ventana donde se encuentra toda la información referente al resultado del alumno como son un resumen general de su rendimiento, la retroalimentación de sus respuestas ingresadas y el listado de sugerencias de estudio. En esta ventana de resultados, si se presiona el botón de *Guardar resultados en PDF*, le permitirá guardar en un archivo de extensión *.pdf* estas sugerencias dadas por el sistema para estudiar. En la figura 58 se observa una vista general de los resultados obtenidos por un alumno.

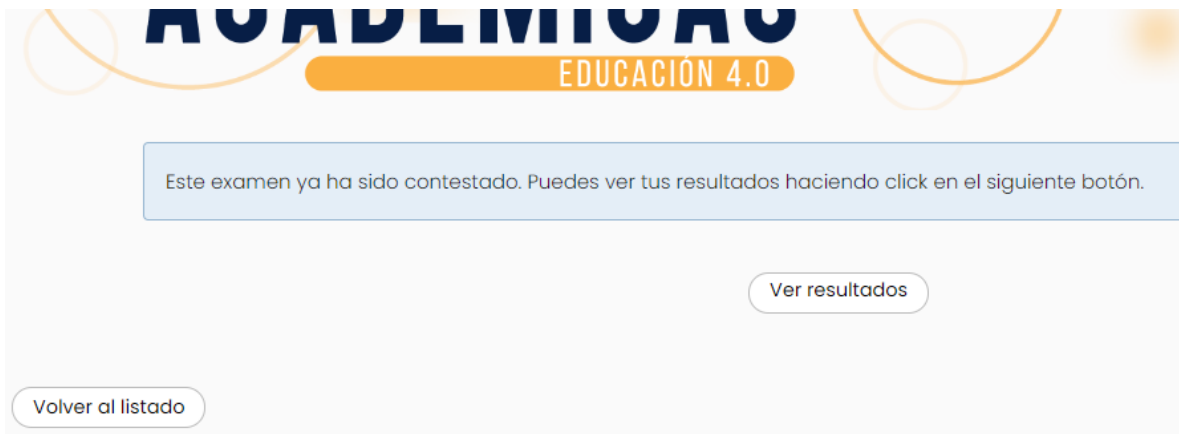


FIGURA 57. MENSAJE QUE APARECE CUANDO SE INTENTA ACCEDER A UN EXAMEN QUE YA FUE RESPONDIDO

Resumen de resultados

Fecha de inicio:	06/10/2023, 21:10 hrs.
Estado:	Finalizado
Finalizado en:	06/10/2023, 21:23 hrs.
Tiempo empleado:	00:04:26 hrs
Puntos:	9/24
Calificación:	3.75

Resumen

1) Was siehst auf dem Foto?
Ich sehe...

- A) ein Telefon
- B) einen Schalter
- C) einen Geburtstag

Respuesta correcta: C) einen Geburtstag




FIGURA 58. VISTA GENERAL DE LOS RESULTADOS OBTENIDOS EN UN EXAMEN

8.2 B. Cuestionario de validación para la Plataforma para la creación de exámenes aplicados a alumnos en el Centro de Lenguas

El conjunto de preguntas presentadas en este apéndice conforma el cuestionario cuyo fin fue el de conocer la situación de los factores de funcionalidad, usabilidad, confiabilidad y eficiencia que la plataforma ofrece a los usuarios. Mediante este, se pretendió calcular un resultado general que indicara el nivel de satisfacción de estos últimos.

8.2.1 Funcionalidad

1. El examen creado fue acorde a los parámetros establecidos.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

2. Los parámetros requeridos son los apropiados para la creación de un examen.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

3. El examen digital creado se adhiere a las características que tendría un examen físico de idiomas.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

4. Las ventanas y botones que se presentan para la visualización de la información y su navegación son adecuadas.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

5. La información solicitada sobre un estudiante, examen o pregunta son veraces.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

8.2.2 Usabilidad

6. Requirió mínimo esfuerzo para reconocer las funciones de la plataforma.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

7. Requirió mínimo esfuerzo para aprender a utilizar la plataforma para la resolución de un examen.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

8. Requirió mínimo esfuerzo para aprender a utilizar la plataforma para la creación de un examen.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

9. La interfaz utiliza una paleta de colores, la distribución de botones y textos, y el tamaño y tipo de fuente son adecuados.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

10. La plataforma creada y sus funciones son las ideales para la creación y aplicación de un examen virtual.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

11. La plataforma podría sustituir/o servir de alternativa a la aplicación de un examen físico (papel).

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

8.2.3 Confiabilidad

12. Es posible recuperar un examen con respuestas ya ingresadas por un estudiante aún después de una falla.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

13. La tasa de errores encontrados en la plataforma es baja.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

8.2.4 Eficiencia

14. El tiempo de respuesta para la creación de un examen es igual o menor al esperado.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

15. El tiempo de respuesta para la visualización de un examen es igual o menor al esperado.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

16. El tiempo de respuesta para la visualización de las listas de exámenes y alumnos, así como la visualización de sus atributos de cada uno, es igual o menor al esperado.

- Totalmente de acuerdo
- De acuerdo
- Regular
- En desacuerdo
- Totalmente en desacuerdo

8.3 C. Modelo Entidad-Relación del Sistema de proceso de evaluación de perfil lingüístico

8.4 D. Modelo Relacional del Sistema de proceso de evaluación de perfil lingüístico

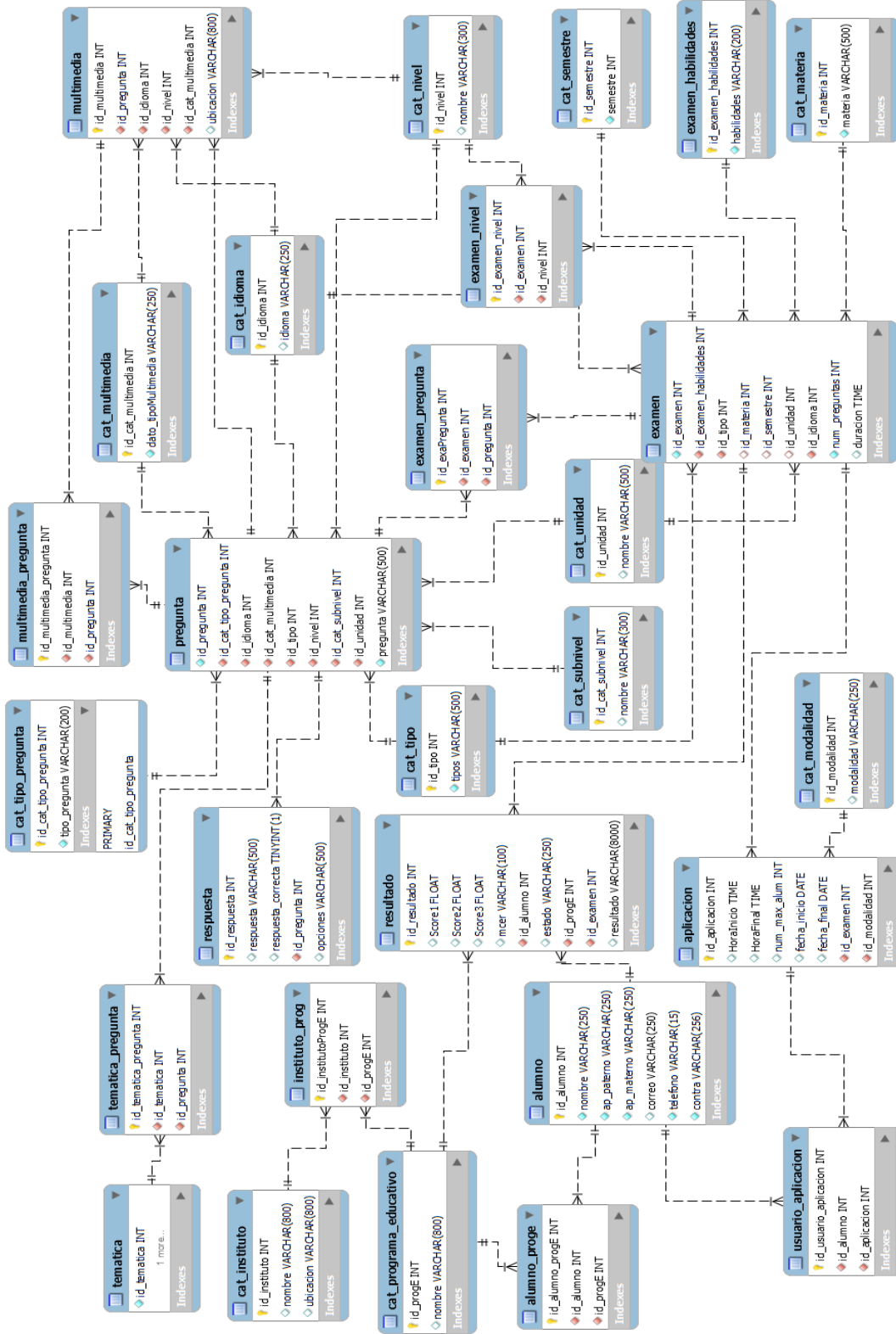


FIGURA 60. MODELO RELACIONAL

9. Anexos

9.1 Anexo 1

```
import { Question } from "../models/Question.js";
import { data } from "./data.js";

export const questions = data.map(
  (question) =>
    new Question(question.question, question.choices, question.answer)
);
```

Código fuente obtenido de GitHub (Fazt Web, 2021) utilizado como ejemplo para la conversión de datos recibidos a objetos de la clase *Question*.

```
class Question {
  /**
   *
   * @param {string} text The Text of the Question
   * @param {string[]} choices a lis of choices for the question
   * @param {string} answer the answeof of the question
   */
  constructor(text, choices, answer) {
    this.text = text;
    this.choices = choices;
    this.answer = answer;
  }

  /**
   *
   * @param {string} choice the choice selected
   * @returns {boolean} returns if the choice is correct
   */
  correctAnswer(choice) {
    return choice === this.answer;
  }
}

export { Question };
```

Código fuente obtenido de GitHub (Fazt Web, 2021) utilizado como ejemplo para la creación de la clase *Question* y su constructor en este sistema.

```
//@ts-check
import { Question } from "./Question.js";

export class Quiz {
  score = 0;
  questionIndex = 0;

  /**
   *
   * @param {Question[]} questions
   */
  constructor(questions) {
    this.questions = questions;
  }

  /**
   *
   * @returns {Question} the question found
   */
  getQuestionIndex() {
    return this.questions[this.questionIndex];
  }

  isEnded() {
    return this.questions.length === this.questionIndex;
  }

  guess(answer) {
    if (this.getQuestionIndex().correctAnswer(answer)) {
      this.score++;
    }
    this.questionIndex++;
  }
}
```

Código fuente obtenido de GitHub (Fazt Web, 2021) utilizado como ejemplo para la construcción de métodos para la clase *Exam* del sistema.

```
export class UI {
  constructor() {}

  /**
   *
   * @param {string} text
   */
  showQuestion(text) {
    const questionTitle = document.getElementById("question");
    questionTitle.innerHTML = text;
  }

  /**
   *
   * @param {string[]} choices
   */
  showChoices(choices, callback) {
    const choicesContainer = document.getElementById("choices");
    choicesContainer.innerHTML = "";

    for (let i = 0; i < choices.length; i++) {
      const button = document.createElement("button");
      button.addEventListener("click", () => callback(choices[i]));
      button.className = "button";
      button.innerText = choices[i];

      choicesContainer.append(button);
    }
  }

  showScores(score) {
    const gameOverHTML = `
      <h1>Result</h1>
      <h2 id="score">Your scores: ${score}</h2>
    `;
    // <h2 id="score">Your scores: ${quiz.score}</h2>

    const element = document.getElementById("quiz");
    element.innerHTML = gameOverHTML;
  }

  showProgress(currentIndex, total) {
    var element = document.getElementById("progress");
    element.innerHTML = `Question ${currentIndex} of ${total}`;
  }
}
```

Código fuente obtenido de GitHub (Fazt Web, 2021) utilizado como ejemplo para la construcción de la clase *UI* que mostraría varios de los elementos de los exámenes (pregunta, opciones, etc.).

```
//@ts-check
import { Quiz } from "../models/Quiz.js";
import { UI } from "../models/UI.js";
import { questions } from "../data/questions.js";

// Rrendering the page
const renderPage = (quiz, ui) => {
  if (quiz.isEnded()) {
    ui.showScores(quiz.score);
  } else {
    console.log(quiz);
    ui.showQuestion(quiz.getQuestionIndex().text);
    ui.showProgress(quiz.questionIndex + 1, quiz.questions.length);
    ui.showChoices(quiz.getQuestionIndex().choices, (currentChoice) => {
      quiz.guess(currentChoice);
      renderPage(quiz, ui);
    });
  }
};

function main() {
  const quiz = new Quiz(questions);
  const ui = new UI();

  renderPage(quiz, ui);
}

main();
```

Código fuente obtenido de GitHub (Fazt Web, 2021) utilizado como ejemplo para la creación del archivo que contendría toda la lógica referente al funcionamiento de los exámenes.