



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

LICENCIATURA EN CIENCIAS COMPUTACIONALES

TESIS

**DESARROLLO DE UN CURSO VIRTUAL
PARA EL APRENDIZAJE DE KOTLIN**

**Para obtener el título de
Licenciado en Ciencias Computacionales**

PRESENTA:

P.L.C.C. Moisés Cruz Pancardo

DIRECTORA:

Dra. Mireya Clavel Maqueda

CODIRECTOR:

Dr. Eduardo Cornejo Velázquez

Mineral de la Reforma, Hidalgo, octubre de 2025

Autorización de impresión



Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences

Mineral de la Reforma, Hgo., a 20 de octubre de 2025

Número de control: ICBI-D/1839/2025
Asunto: Autorización de impresión.

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

Con 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 al egresado de la Licenciatura en Ciencias Computacionales **Moisés Cruz Pancardo**, quien presenta el trabajo de titulación "**DESARROLLO DE UN CURSO VIRTUAL PARA EL APRENDIZAJE DE KOTLIN**", ha decidido, después de revisar fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; dicho trabajo en la reunión de sinodales, **autorizar la impresión del mismo**, una vez realizadas las correcciones acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente: Dr. Edgar Olguín Guzmán

Secretario: M.I.D. Norma Laura Salazar Viveros

Vocal: Dra. Mireya Clavel Maqueda

Suplente: Dr. Eduardo Comejo Velázquez

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

Atentamente
"Amor, Orden y Progreso"

Mtro. Gabriel Vergara Rodríguez
Director de ICBI



GVR/YCC

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. 40001
direccion_icbi@uaeh.edu.mx, vergara@uaeh.edu.mx

"Amor, Orden y Progreso"



2025



uaeh.edu.mx

Dedicatoria

En primer lugar, le dedico este triunfo a mi abuela: Mamá Lola. Quien, aunque ya no está físicamente conmigo, vive eternamente en mi corazón y en cada uno de mis logros. Ella fue mi guía, mi apoyo incondicional y el ejemplo más puro de amor y bondad. Desde siempre cuidó de mí con ternura, me enseñó la humildad, la sencillez y el valor de compartir. Me brindó todo sin esperar nada a cambio, siempre preocupándose por mi bienestar y alentándome a seguir adelante. Su ausencia me duele profundamente, pero también me impulsa a continuar, porque sé que estaría orgullosa de verme cumplir este sueño que tantas veces me motivó a perseguir. A ella, con todo mi amor y eterna gratitud.

A mis padres, Moises y Dolores quienes con su esfuerzo, sacrificio y amor incondicional me dieron la oportunidad de prepararme y forjar un camino sólido, siendo mi mayor ejemplo de perseverancia.

A mi padre, por su fortaleza, por su sabiduría en cada consejo, por escucharme aun en la distancia y enseñarme que los logros se construyen con disciplina. A mi madre, por su dedicación constante, por exigirme siempre dar lo mejor de mí y por creer en mi potencial incluso cuando yo dudaba. Ambos me enseñaron el valor del trabajo y la honestidad, y gracias a ello nunca me ha faltado nada más que el tiempo para agradecerles como merecen.

A mi hermana, mi confidente y compañera de aventuras, quien siempre ha estado a mi lado brindándome su apoyo, comprensión y alegría. Su presencia ha sido fundamental en los momentos difíciles, y espero que este logro sea también una inspiración para ella, un recordatorio de que todo esfuerzo tiene su recompensa y de que los sueños sí se cumplen.

A Dulce Martínez, por su compañía durante mi etapa universitaria, por su apoyo, consejos y paciencia en los momentos que lo necesité. Su presencia me ayudó a mantener el enfoque y a no perder el rumbo en los momentos de incertidumbre. Agradezco las enseñanzas y experiencias compartidas, las cuales contribuyeron a mi crecimiento personal y profesional.

A mis asesores, el Dr. Eduardo Cornejo Velázquez y la Mtra. Mireya Clavel Maqueda, por su guía académica, su paciencia y su calidad humana. Gracias por compartir sus conocimientos,

por confiar en mi proyecto, su orientación durante la elaboración de este trabajo y las oportunidades brindadas que me permitieron crecer tanto profesional como personalmente. Su apoyo fue esencial para culminar con éxito esta etapa, y siempre llevaré conmigo el aprendizaje que me dejaron.

Finalmente, dedico este logro a todas aquellas personas a quienes tengo el privilegio de llamar amigos y amigas. Gracias por su compañía, las risas, los consejos, la motivación constante y por estar presentes en los momentos en los que más los necesité. Cada palabra de aliento y cada gesto de apoyo fueron una chispa más que me ayudó a no rendirme.

A todos ustedes, gracias por formar parte de mi historia y por creer en mí. Este logro no solo es mío: es también suyo.

Resumen

En la actualidad, la programación móvil se ha convertido en un área de creciente importancia dentro de la formación académica y profesional, ya que promueve competencias digitales esenciales como el pensamiento lógico, la resolución de problemas y la creatividad. Sin embargo, para estudiantes principiantes, la programación suele representar un reto debido a la complejidad de los entornos de desarrollo y a la falta de recursos educativos accesibles. Ante este panorama, la educación virtual ofrece un medio flexible y atractivo para introducir a los alumnos en la programación mediante metodologías activas y recursos digitales interactivos.

Con base en esta necesidad, el presente trabajo describe el diseño, desarrollo y publicación de un curso en línea orientado a la enseñanza del lenguaje de programación Kotlin en Android Studio. El proyecto se fundamentó en el modelo instruccional ADDIE, complementado con el Aprendizaje Basado en Problemas (ABP) y con el uso de elementos lúdicos inspirados en la temática de Dragon Ball, con el propósito de fortalecer la motivación y el aprendizaje significativo.

Durante la construcción del curso se emplearon diversas herramientas tecnológicas: Moodle como Sistema Gestor de Aprendizaje, Hostinger como servicio de alojamiento para su publicación, y Laragon como entorno de pruebas para garantizar su funcionamiento previo a la liberación en línea. Además, se elaboraron materiales propios como videos, guiones explicativos e infografías creadas en Canva, así como actividades interactivas diseñadas con Lumi.

De manera complementaria, se utilizaron diagramas de flujo y de estado para estructurar procesos y contenidos, lo que aportó claridad a la planeación y permitió establecer un orden lógico en la enseñanza. Estos recursos, en conjunto, funcionan como una guía práctica que facilita la comprensión de los estudiantes y orienta su avance dentro del curso.

Como parte del diseño instruccional, se llevó a cabo una investigación previa para identificar áreas vulnerables en la enseñanza de la programación móvil, lo que permitió definir un perfil de estudiante adecuado. En este sentido, el curso se orienta principalmente a principiantes en programación, recomendando un rango de edad a partir de los 15 o 16 años, correspondiente

a nivel medio superior, dado que en este punto académico los estudiantes cuentan con habilidades básicas de lógica matemática y están en condiciones de utilizar entornos de desarrollo como Android Studio.

La propuesta incluye actividades diagnósticas, prácticas y evaluaciones finales, que acompañan al estudiante desde el aprendizaje de conceptos fundamentales hasta la creación de interfaces gráficas simples. Los resultados obtenidos reflejan que la estructura del curso permitió introducir a los participantes en la programación móvil de manera clara y atractiva, favoreciendo la motivación, la autonomía y el aprendizaje significativo.

Este proyecto constituye una aportación académica relevante al campo de la educación virtual en programación, al combinar metodologías pedagógicas, recursos tecnológicos y estrategias creativas en un curso integral, replicable y adaptable a diferentes contextos educativos, orientado al fortalecimiento de competencias digitales esenciales en estudiantes principiantes.

Abstract

Nowadays, mobile programming has become an area of growing importance in academic and professional training, as it fosters essential digital skills such as logical thinking, problem-solving, and creativity. However, for beginners, programming often represents a challenge due to the complexity of development environments and the lack of accessible educational resources. In this context, virtual education offers a flexible and attractive way to introduce students to programming through active methodologies and interactive digital resources.

Based on this need, this project describes the design, development, and deployment of an online course focused on teaching Kotlin programming language in Android Studio. The project was based on the ADDIE instructional design model, complemented with Problem-Based Learning (PBL) and the use of ludic elements inspired by Dragon Ball theme, intended to strengthen motivation and meaningful learning.

During the construction of the course, several technological tools were employed: Moodle as a Learning Management System, Hostinger as a hosting service for deployment and Laragon as a testing environment to ensure functionality before online deployment. In addition, original materials such as videos, explanatory scripts, and infographics created with Canva were produced, along with interactive activities designed with Lumi. Complementary resources such as flowcharts and state diagrams were also used to structure processes and contents, providing clarity in planning and ensuring logical order in instruction.

Together, these resources function as a practical guide that supports student understanding and progression throughout the course.

As part of the instructional design, a preliminary investigation was carried out to identify vulnerable areas in mobile programming education, which allowed the definition of a suitable student profile. In this regard, the course is mainly aimed at beginners in programming, recommending a minimum age of 15 or 16 years, corresponding to upper secondary level, since at this stage students generally possess basic logical-mathematical skills and can use development environments such as Android Studio.

The project includes diagnostic, practical, and final assignments that accompany students from learning fundamental concepts to creating simple graphical interfaces. The results showed that the course structure successfully introduced participants to mobile programming in a clear and attractive way, fostering motivation, autonomy, and meaningful learning.

This project constitutes a relevant academic contribution to the field of virtual education, by combining pedagogical methodologies and creative strategies in an integral, replicable, and adaptative course, aimed at strengthening essential digital skills in beginner students.

Índice

CAPÍTULO 1. Introducción	1
1.1 Antecedentes.....	3
1.2 Planteamiento del problema	5
1.3 Justificación del proyecto.....	7
1.4 Objetivos del proyecto	8
1.4.1 Objetivo General	8
1.4.2 Objetivos Específicos	8
1.5 Alcances y limitaciones	9
1.6 Herramientas Utilizadas	9
CAPÍTULO 2. Marco Teórico	12
2.1 El e-learning	12
2.2 Entorno Virtual de Aprendizaje	13
2.3 Sistemas de Gestión de Aprendizaje.....	18
2.4 Diseño Instruccional	20
2.4.1 Modelo ADDIE.....	22
2.4.2 Modelo ASSURE.....	23
2.4.3 Modelo GAGNÉ.....	25
2.4.4 Modelo de Dick y Carey	27
CAPÍTULO 3. Estado del Arte.	30
3.1 Android: programación de dispositivos móviles a través de ejemplos	30
3.2 Desarrollo de competencias en programación de aplicaciones móviles.....	31
3.3 Plataforma e-learning de apoyo al estudio en estudiantes universitarios	33
3.4 Capacitación para mejorar competencias de acompañamiento virtual.....	34
3.5 Programación informática y robótica en la enseñanza básica.....	35
3.6 Discusión	37
CAPÍTULO 4. Marco Metodológico.....	39
4.1 Metodología de diseño instruccional	39
4.2 Metodología de enseñanza	42
4.2.1 Aprendizaje basado en problemas	45
4.3 Desarrollo iterativo e incremental	47
CAPÍTULO 5. Desarrollo del curso	48

5.1 Análisis de recursos	48
5.2 Diseño de la iniciativa	56
5.3 Desarrollo de la secuencia didáctica	56
5.3.1 Patrones visuales y elección cromática	59
5.3.2 Backlog del curso	61
5.4 Ciclos de elaboración de recursos.....	64
5.4.1 Implementación del curso.....	66
5.4.2 Construcción de imágenes	66
5.4.3 Creación de videos.....	67
5.4.4 Creación de elementos lúdicos.....	73
5.5 Implementación del curso.....	76
5.6 Configuración local de desarrollo	77
5.7 Evaluación del curso.....	82
5.7.1 Evaluación del aprendizaje de los participantes	88
5.8 Publicación del curso	90
CAPÍTULO 6. Resultados y conclusiones.....	94
6.1 Resultados.....	94
6.2 Conclusión.....	97
6.3 Trabajos futuros.....	98
Referencias Bibliográficas	100
Anexos.....	104

Índice de figuras

Figura 1 Etapas de ADDIE.....	22
Figura 2 Etapas de ASSURE.....	24
Figura 3 Etapas de Gagné.....	26
Figura 4 Etapas de Dick y Carey	28
Figura 5 Pantalla de inicio de GoKoan	34
Figura 6 Ventajas del aprendizaje basado en resolución de problemas	46
Figura 7 Diagrama de casos de uso módulo 3	54
Figura 8 Esquema de aplicación ABP	55
Figura 9 Diagrama de estado de la secuencia didáctica.....	57
Figura 10 Diagrama de actividades: sopa de letras.....	58
Figura 11 Paleta de colores	60
Figura 12 Diseño de portada	61
Figura 13 Entorno de trabajo de Canva	67
Figura 14 Entorno de trabajo de Adobe Express	67
Figura 15 Diseño de video de bienvenida.....	72
Figura 16 Entorno de trabajo de DFD	73
Figura 17 Entorno de trabajo de Lumi	74
Figura 18 Entorno de trabajo de Android Studio.....	76
Figura 19 Mapa de navegación: Prueba 3 “Transformaciones”	77
Figura 20 Descarga de Laragon	79
Figura 21 Directorio de Laragon	79
Figura 22 Ubicación de directorio laragon/www.....	80
Figura 23 Panel de control de Laragon.....	80
Figura 24 Entorno de trabajo de phpMyAdmin	81
Figura 25 Instalación de Moodle.....	82
Figura 26 Pantalla de inicio de Moodle	82
Figura 27 Formulario de registro de alumnos	88
Figura 28 Evaluación diagnóstica	89
Figura 29 Evaluación final.....	90
Figura 30 Panel principal de Hostinger	91

Figura 31 Panel de Administración de Hostinger	91
Figura 32 Sitio oficial de Moodle	92
Figura 33 Descomprensión de archivos.....	92
Figura 34 Concluir instalación de Moodle	93
Figura 35 Contenido del módulo 2.....	95
Figura 36 Prueba 1: Los juegos de Cell.....	96

Índice de tablas

Tabla 1 Sistemas de gestión de aprendizaje	19
Tabla 2 Comparativo de modelos instruccionales	40
Tabla 3 Comparativo de metodologías de enseñanza.....	43
Tabla 4 Caso de uso: Manipulando datos en Kotlin.....	52
Tabla 5 Actividades y prioridades del módulo 3.....	62
Tabla 6 Guion de video: widgets.....	68
Tabla 7 Parámetros de evaluación EVA	84

CAPÍTULO 1. Introducción

En la actualidad, la programación se enuncia como una competencia esencial para que las personas puedan desenvolverse en un entorno cada vez más digitalizado. La evolución de las aplicaciones móviles, en particular, ha experimentado un crecimiento ininterrumpido debido al uso masivo de dispositivos inteligentes y a la demanda de soluciones innovadoras en sectores como la educación, el comercio, la salud y el entretenimiento.

En este escenario, la programación implica adquirir conocimientos técnicos, así como desarrollar habilidades de pensamiento lógico, resolución de problemas y creatividad.

A lo largo de los años Android se ha convertido en el sistema operativo más usado a nivel mundial, ya sea por el bajo costo de los móviles, o por la cantidad de contenido disponible que tenemos a nuestra disposición (Guimerá Orozco, 2018).

En el ámbito del desarrollo móvil, Kotlin ha ganado relevancia desde que Google lo adoptó como lenguaje oficial para Android en 2017. Su lógica de programación es amigable, desde crear una variable hasta realizar toda una función, un microservicio, un web services o cualquier cosa que demande más tiempo desarrollarlo, es muy sencillo de realizar, una vez que se aprende la sintaxis del lenguaje el resto se avanza de manera rápida y de esta forma Kotlin puede ser usado cuando es necesario lanzar un programa en el menor tiempo posible (María & De Alcalá, 2021).

Se considera una opción idónea para introducir a estudiantes principiantes en la programación de aplicaciones móviles. Android Studio, como Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés), proporciona un conjunto de herramientas que facilita el diseño, codificación, depuración y prueba de aplicaciones, potenciando el aprendizaje práctico.

Por otra parte, la evolución de las Tecnologías de la Información y la Comunicación (TIC) ha facilitado la generación y difusión del conocimiento, impulsando nuevos enfoques educativos que fomentan el pensamiento lógico, la resolución de problemas y la creatividad.

La educación virtual, sustentada en plataformas digitales y Sistemas Gestores de Aprendizaje (SGA) que buscan optimizar el aprendizaje en línea, ha permitido el desarrollo de entornos más dinámicos, accesibles y adaptables a las necesidades de la sociedad actual de manera estructurada.

Estos sistemas permiten organizar el contenido en módulos o unidades de aprendizaje, facilitando la integración de actividades interactivas, evaluaciones continuas y recursos multimedia que favorecen un aprendizaje más dinámico, debido a la simplicidad otorgada para acceder y consumir contenido educativo desde cualquier ubicación mientras se tenga un dispositivo inteligente con acceso a internet, ventajas por las se favorece el aprendizaje autónomo y la posibilidad de avanzar a un ritmo personalizado.

Estas características eliminan barreras geográficas y temporales, favoreciendo que personas de distintos perfiles accedan a formación especializada.

El reto que enfrentan las instituciones educativas y los docentes radica en diseñar estrategias efectivas para la enseñanza de la programación, especialmente considerando que muchos estudiantes tienen su primer acercamiento a esta disciplina en la educación superior y pueden experimentar dificultades en su aprendizaje.

Ante esta problemática, surge la necesidad de estrategias innovadoras que no solo optimizan la enseñanza, sino que también motivan a los estudiantes a continuar desarrollando sus habilidades en un entornos inclusivos y adaptables.

Otro aspecto clave de la educación virtual es la incorporación de metodologías activas, que promueven la participación activa de los estudiantes y estimulan el desarrollo de habilidades de resolución de problemas, así como el pensamiento crítico.

La educación virtual toma lugar como un modelo asequible, versátil y cada vez más extendido, permitiendo a los estudiantes de diversas ubicaciones geográficas acceder a contenidos educativos de calidad, eliminando barreras físicas y facilitando el aprendizaje autónomo.

Este modelo ha sido especialmente impulsado por la creciente transformación digital, donde la demanda de habilidades técnicas y competencias digitales sigue en aumento.

El desarrollo de un curso virtual que combine Kotlin, Android Studio y Moodle representa una oportunidad para ofrecer una experiencia de aprendizaje accesible, motivadora y adaptada a principiantes.

El uso de SGA actualmente se ha tornado indispensable en el proceso de enseñanza y aprendizaje; es importante tomar en cuenta tanto la usabilidad como la experiencia de usuario en el diseño de estos sistemas (García, 2021).

1.1 Antecedentes

En los últimos años, la educación a distancia ha ganado relevancia en diversas partes del mundo y México no ha sido la excepción. La educación a distancia en México ha experimentado un crecimiento significativo, impulsado principalmente por el avance de las TIC, que facilitan la creación de plataformas virtuales y cursos en línea. Dichas herramientas permiten que los estudiantes, independientemente de su ubicación geográfica, accedan a contenidos educativos sin las limitaciones del espacio y del tiempo.

La educación a distancia en México logra desafiar retos como la desigualdad en el acceso a tecnologías y la falta de capacitación docente en el uso de estas herramientas, demostrando ser una alternativa eficaz para ampliar las oportunidades educativas en contextos geográficos dispersos y en sectores de la población que, de otro modo, no tendrían acceso a una educación de calidad. (Cázales & Granados, 2017)

La integración de la educación a distancia se ha consolidado no solo en el ámbito de la educación superior, sino también en la formación continua y la capacitación técnica, áreas en las que las herramientas digitales han abierto nuevas posibilidades para que los estudiantes adquieran habilidades clave para el desarrollo profesional.

Es posible destacar que la educación a distancia tiene la capacidad para adaptarse a las necesidades de los estudiantes mediante el uso de metodologías como el Aprendizaje Basado en Problemas (ABP), que fomenta la aplicación de conceptos a situaciones reales.

En la actualidad, la incorporación de las TIC en la educación superior ha transformado los métodos de enseñanza y aprendizaje, brindando una mayor flexibilidad y accesibilidad en la formación académica, también fomentan el desarrollo de competencias digitales en docentes y estudiantes, lo que contribuye a una educación más dinámica y contextualizada.

Es importante destacar que la capacitación docente en el uso de tecnologías es esencial para aprovechar al máximo los recursos digitales y mejorar la enseñanza en línea.

De igual modo, las TIC han generado una transformación en la manera en que los docentes diseñan y estructuran sus estrategias de enseñanza, permitiendo la creación de modelos de aprendizaje más personalizados y adaptativos (Quiroz & Quiroz, 2019)

En este sentido, las plataformas de aprendizaje como Moodle se convierten en herramientas fundamentales para facilitar la interacción entre los estudiantes y los contenidos, brindando un espacio estructurado para el desarrollo de habilidades teóricas y prácticas.

El uso de herramientas digitales facilita la gestión del conocimiento y promueve entornos interactivos que mejoran la calidad educativa.

La relevancia de este enfoque en la educación superior radica en su capacidad de eliminar barreras geográficas y temporales, promoviendo el acceso equitativo a la educación. Además, la implementación de TIC en los procesos de enseñanza-aprendizaje no solo mejora la calidad educativa, sino que también fomenta habilidades esenciales para la sociedad digital actual, como la autonomía en el aprendizaje, la resolución de problemas y el pensamiento crítico.

En el ámbito de la educación en programación, la integración de Entornos Virtuales de Aprendizaje (EVA) ha demostrado ser una estrategia efectiva para mejorar la comprensión de conceptos y fortalecer el desarrollo de habilidades computacionales.

La implementación de estas plataformas facilita el aprendizaje autónomo, permitiendo la retroalimentación inmediata y fomenta la resolución de problemas a través de ejercicios prácticos estructurados.

El surgimiento del movimiento de Software Libre (SL) propone un nuevo modelo de trabajo sobre las libertades de los usuarios y la propiedad intelectual del software desarrollado. En la actualidad, ha demostrado tener viabilidad tanto técnica como económica y las estadísticas reflejan un crecimiento sostenido de su uso (Aguirre, 2015)

Los entornos digitales no solo optimizan el proceso de enseñanza, sino que también incentivan la participación de los estudiantes, promoviendo la interacción y el trabajo colaborativo. Además, resaltan que el acceso a recursos virtuales bien diseñados mejora significativamente la motivación y el rendimiento académico de los estudiantes, factores clave para el éxito en el desarrollo de habilidades de programación.

Este antecedente resulta relevante para este trabajo, ya que respalda la necesidad de emplear herramientas digitales en la enseñanza de la programación. Es evidente que la adopción de metodologías innovadoras basadas en las TIC contribuye a una formación más efectiva y accesible para los estudiantes, alineándose con las tendencias actuales en educación digital. la educación virtual presenta grandes desafíos para crear ambientes de aprendizaje más reales, con mayor interacción entre estudiantes y entre estudiantes y profesores y de calidad, sostenidos dentro de un marco teórico de la psicología educativa (Carrillo, 2018).

1.2 Planteamiento del problema

En la era digital con mundo cada vez más interconectado a través de los dispositivos móviles, la programación móvil se ha convertido en una habilidad esencial, ya que constituye una base para el desarrollo aplicaciones y soluciones tecnológicas que impulsan diversos sectores productivos, educativos y sociales. Sin embargo, su aprendizaje puede presentar un reto para los principiantes, enfrentando barreras relacionadas con la falta de accesibilidad, enfoques pedagógicos inadecuados y recursos limitados debido a que suelen presentar contenidos

fragmentados, un nivel de complejidad elevado o una falta de contextualización que dificulta el aprendizaje autónomo y progresivo.

A pesar de existir diversas herramientas de aprendizaje tales como tutoriales, videos y cursos sobre programación en Kotlin, la mayoría se centra en transmitir contenidos de manera desarticulada, sin integrar procesos de retroalimentación, evaluación ni acompañamiento que permitan al estudiante verificar sus avances y consolidar su aprendizaje. Esta ausencia de estructura limita la posibilidad de desarrollar competencias prácticas, pensamiento lógico y habilidades de resolución de problemas, indispensables en la formación inicial de un programador.

Esta problemática se acentúa en principiantes de la programación, que requieren materiales claros, ejemplos cercanos y un entorno virtual que no solo les brinde información, sino que también los motive y guíe en su proceso de aprendizaje.

Los sistemas gestores de aprendizaje (SGA), como Moodle, ofrecen la posibilidad de organizar los contenidos en módulos, integrar actividades interactivas, aplicar evaluaciones y facilitar la retroalimentación. No obstante, su potencial depende en gran medida de la forma en que se diseñen los cursos y de la implementación de metodologías activas que conviertan al estudiante en protagonista de su propio aprendizaje. El uso de las TIC influyen en los procesos educativos ofreciendo el uso de modelos innovadores de enseñanza-aprendizaje de manera alcanzable y maleable.

En este contexto, surge la necesidad de desarrollar herramientas educativas innovadoras que no solo introduzcan a los estudiantes en el mundo de la programación, sino que también brindan un entorno dinámico en el que puedan experimentar, resolver problemas y construir soluciones tecnológicas reales. La implementación de un curso bien estructurado en un EVA puede marcar la diferencia en la adquisición de habilidades clave como la resolución de problemas, el pensamiento crítico y la autonomía en el aprendizaje.

Responder a estas necesidades es fundamental para fortalecer un sistema educativo inclusivo y adaptado a las realidades de una sociedad en constante cambio. Este enfoque no solo busca satisfacer las demandas del mercado laboral, sino también formar individuos capaces de

aprovechar las ventajas de las TIC para enfrentar los retos actuales y futuros en el ámbito tecnológico.

La falta de propuestas educativas diseñadas específicamente para principiantes en programación móvil, con materiales accesibles, estructurados y motivadores, constituye una brecha importante en el proceso de formación digital. De no atenderse esta necesidad, se corre el riesgo de que los estudiantes se enfrenten a experiencias poco satisfactorias, que conduzcan al desinterés, la deserción y la falta de preparación en un área que resulta clave para el desarrollo académico y profesional en la era digital.

La aparición de nuevas formas de organización económica, social, política y cultural trae consigo nuevas maneras de trabajar, comunicarnos, aprender, pensar y vivir. A este nuevo orden social se lo llama Sociedad de la Información. Las TIC ofrecen la posibilidad de reconstruir y reinterpretar las posibilidades de enseñar, de acuerdo con el marco socio - educativo - cultural de referencia (Aguirre, 2015).

1.3 Justificación del proyecto

El aprendizaje de la programación móvil representa un reto significativo para los estudiantes principiantes, debido a la complejidad de los entornos de desarrollo y a la falta de materiales didácticos estructurados que faciliten la comprensión progresiva de los contenidos. A pesar de existir gran variedad de recursos en línea, la mayoría presenta una secuencia pedagógica limitada, con contenidos fragmentados y escasa retroalimentación, lo que dificulta el aprendizaje autónomo y reduce la motivación de los estudiantes en sus primeros acercamientos a la programación.

Ante este panorama, la educación virtual se presenta como una alternativa eficaz para atender las necesidades de quienes inician en la programación. Los entornos digitales de aprendizaje permiten combinar flexibilidad, interactividad y acompañamiento, ofreciendo un espacio estructurado en el que los estudiantes puedan avanzar de manera guiada y práctica. Su aplicación en el campo de la programación móvil resulta especialmente pertinente, ya que posibilita integrar recursos multimedia, actividades interactivas y evaluaciones automatizadas que facilitan la comprensión de conceptos abstractos.

Por ello, este proyecto se justifica en la necesidad de ofrecer una propuesta educativa innovadora que contribuya al fortalecimiento de las competencias digitales básicas, particularmente en el ámbito de la programación móvil. A través del uso de recursos tecnológicos y estrategias didácticas adecuadas, se busca generar un impacto positivo en la formación inicial de los futuros programadores, promoviendo la autonomía, la motivación y la continuidad del aprendizaje en un campo esencial para el desarrollo tecnológico actual.

1.4 Objetivos del proyecto

1.4.1 Objetivo General

Desarrollar un curso virtual interactivo, implementado en Moodle, para la enseñanza de los fundamentos del lenguaje de programación Kotlin a nivel básico, fomentando el aprendizaje autónomo, el pensamiento lógico y la resolución de problemas, a través de una metodología didáctica que combine enfoques activos y actividades prácticas que favorezcan la comprensión y participación de los estudiantes.

1.4.2 Objetivos Específicos

- Diseñar los contenidos educativos del curso mediante una metodología de diseño instruccional adecuada.
- Aplicar metodologías activas que fomenten el aprendizaje autónomo y la resolución de problemas en los estudiantes.
- Implementar el curso en la plataforma Moodle y publicarlo en línea para garantizar su accesibilidad.
- Evaluar la efectividad del curso a través del desempeño y la retroalimentación de los estudiantes.

1.5 Alcances y limitaciones

Los alcances de este proyecto hacen énfasis en el desarrollo del curso en nivel básico diseñado principalmente para principiantes, implementado en la plataforma Moodle. El curso integra contenidos teóricos y prácticos, actividades interactivas, evaluaciones y retroalimentación, lo que brinda a los participantes una experiencia formativa estructurada y motivadora. Asimismo, la incorporación de metodologías activas como el ABP y recursos lúdicos que favorecen la permanencia y el interés de los estudiantes.

De manera adicional, el proyecto alcanzó la implementación de recursos digitales complementarios, tales como diagramas de flujo y de estado, videos explicativos y materiales elaborados en Canva, Lumi, Adobe Express, DFD, que enriquecieron el curso y lo hicieron más accesible. Este conjunto de elementos contribuyó al fortalecimiento de competencias básicas en programación y pensamiento lógico, constituyendo una base sólida para futuros aprendizajes en el área del desarrollo de aplicaciones móviles.

Entre las principales limitaciones se encuentra que el curso está orientado únicamente a un nivel introductorio, por lo que no aborda temáticas avanzadas de programación en Kotlin ni el desarrollo de aplicaciones móviles de gran complejidad.

Asimismo, la propuesta se enfocó en un rango de estudiantes principiantes, lo que restringe su aplicación inmediata en perfiles más avanzados. Además, la dependencia de recursos tecnológicos como el acceso a dispositivos inteligentes y a una conexión estable a internet constituyó un factor que pudo influir en la participación de algunos usuarios.

1.6 Herramientas Utilizadas

- **Moodle** versión 4.3 es una plataforma de aprendizaje diseñada para proporcionar a educadores, administradores y estudiantes un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados (*Acerca De Moodle - MoodleDocs*, s.f.). Permitió organizar los contenidos en módulos, integrar evaluaciones, aplicar actividades interactivas y dar seguimiento al progreso de los participantes.
- **Canva** es una plataforma de diseño y comunicación visual cuya misión es poner el poder del diseño al alcance de todo el mundo, para que cualquier persona pueda

diseñar lo que quiera y publicarlo donde quiera (Canva, s.f.). Utilizada para la elaboración de materiales gráficos y recursos visuales que hicieron más atractivos los materiales, reforzando la motivación de los estudiantes y aportando claridad en la presentación de conceptos.

- **Android Studio**, IDE oficial de Android, fue creado exclusivamente a fin de acelerar el desarrollo y ayudarte a compilar Apps de alta calidad para todos los dispositivos Android (*Introducción A Android Studio*, s. f.). Constituyó la herramienta central de programación, ya que permitió trabajar directamente con el lenguaje Kotlin y practicar la creación de interfaces gráficas básicas, logrando que el aprendizaje fuera práctico y aplicado.
- **DFD** es un software diseñado para construir y analizar algoritmos. Permite crear diagramas de flujo de datos para representar algoritmos de programación estructurada a partir de herramientas de edición que ofrece el programa (Ferreira Gómez, 2018). Se empleó en la elaboración de diagramas de flujo, que sirvieron como apoyo visual en el curso para representar procesos lógicos y reforzar la comprensión de los algoritmos.
- **Hostinger** es un proveedor de alojamiento y dominios web que ofrece servicios accesibles para publicar y administrar sitios en línea (Hostinger, 2025.). Herramienta utilizada para la publicación del curso en un dominio propio, permitiendo su accesibilidad a través de un entorno virtual estable y replicable.
- **Laragon** es una herramienta gratuita y portable que crea un entorno de desarrollo local rápido, flexible y potente en Windows, ideal para desarrollar aplicaciones web con PHP, Node.js, Python, Go y Ruby. Integra servidor web (Apache, Nginx), base de datos (MySQL), Git, Node.js y otras tecnologías, simplificando la configuración y acelerando la productividad del desarrollador (LeoKhoa, s.f.). Facilitó la ejecución de pruebas de configuración y funcionamiento en un entorno local antes de la publicación definitiva en línea, lo que facilitó la detección y resolución de posibles errores.

- **Lumi** es una herramienta de código abierto para la creación de contenido interactivo HTML5, diseñada para que educadores y profesionales del aprendizaje electrónico puedan crear, compartir y reutilizar recursos educativos (Educación Lumi, 2024). Fue de utilidad para generar actividades dinámicas y evaluaciones interactivas que enriquecieron la experiencia del estudiante, favoreciendo la participación activa y el aprendizaje autónomo.

CAPÍTULO 2. Marco Teórico.

Este capítulo aborda temas primordiales para presentar el entendimiento de la información acerca de los temas vistos en los segmentos posteriores. Teniendo como propósito detallar la evolución de los dispositivos móviles en la vida cotidiana así mismo estudiar la estructura de un SGA y su importancia en el aprendizaje autónomo virtual.

La enseñanza de la programación móvil ha sido abordada en diversas investigaciones y recursos bibliográficos, donde se destaca la importancia de metodologías que facilitan la formación y desarrollo de habilidades prácticas, para el desarrollo de aplicaciones móviles. Existen múltiples enfoques que integran el uso de herramientas como Android Studio y lenguajes como Kotlin, promoviendo estrategias de aprendizaje autónomo.

2.1 El e-learning

Traducido al español como aprendizaje en línea es un modelo de educación derivado del uso de tecnologías digitales para la enseñanza y el aprendizaje, caracterizado por permitir el acceso a contenidos educativos a través de plataformas en línea, que facilitan el acceso a contenido educativo, así como la interacción entre estudiantes y docentes, impulsando la formación a distancia y brindando flexibilidad en el proceso de aprendizaje (Hernández et al., 2024).

El e-learning no trata solamente de tomar un curso y colocarlo en una computadora, se trata de una combinación de recursos, interactividad, apoyo y actividades de aprendizaje estructuradas. Para realizar todo este proceso es necesario conocer las posibilidades y limitaciones que el soporte informático o plataforma virtual nos ofrece (Belloch, 2012).

A continuación, se describen algunas características:

- **Accesibilidad:** permite a los estudiantes aprender desde cualquier lugar y en cualquier momento.
- **Uso de Tecnologías de la Información y la Comunicación (TIC):** utiliza plataformas virtuales, videos, foros, videoconferencias y otros recursos digitales.

- **Aprendizaje autónomo:** el estudiante gestiona su tiempo y ritmo de aprendizaje.
- **Interactividad:** facilita la comunicación entre estudiantes y docentes mediante chats, foros y herramientas colaborativas.
- **Evaluación y seguimiento:** permite el monitoreo del progreso mediante cuestionarios, exámenes en línea y reportes de desempeño.

Es necesario hacer una diferencia importante entre los modelos tradicionales de educación a distancia y el e-learning. Inicialmente, el e-learning estaba vinculado al uso de los ordenadores personales, en la actualidad está sostenido por los sistemas distribuidos que han favorecido las redes de aprendizaje y el acceso a las plataformas desde cualquier dispositivo. Así, el uso de dispositivos móviles se ha integrado en los diseños educativos y las actividades de aprendizaje en línea.

La educación a distancia puede o no utilizar tecnología garantizando el estudio independiente sin necesidad de una intervención continua por parte del docente. En el caso del e-learning, se comparte la no presencialidad del modelo, pero hace énfasis en la utilización de Internet como sistema de acceso a los contenidos y a las actividades de la formación, siendo la interacción y la comunicación una parte fundamental en este tipo de formación.

La evolución del e-learning está vinculada a las transformaciones tecnológicas y a los cambios relacionados con los espacios y tiempos para la formación y el aprendizaje.

2.2 Entorno Virtual de Aprendizaje

Un Entorno Virtual de Aprendizaje (EVA) se puede definir como una plataforma digital cuyo objetivo es facilitar la enseñanza y el aprendizaje a través de herramientas tecnológicas, ofreciendo accesibilidad, gestión académica y recursos interactivos. Permitiendo la interacción entre estudiantes y docentes mediante recursos como foros, materiales didácticos, evaluaciones y actividades interactivas.

Los EVA no solo proporcionan un espacio digital para el aprendizaje, sino que además integran herramientas técnicas y pedagógicas que facilitan una experiencia educativa eficaz, flexible y accesible.

Los ambientes virtuales de aprendizaje tienen cada vez mayor preponderancia en el proceso de enseñanza. La incursión tecnológica se ha convertido en un reto para los modelos pedagógicos. Es por ello que surge la imperiosa necesidad de aplicar estrategias innovadoras que propicien el desarrollo de competencias que puedan fomentar en los estudiantes su capacidad crítica y reflexiva de conocimientos básicos en distintos ámbitos (Cedeño Romero & Murillo Moreira, 2019).

A continuación, se mencionan características relevantes de los EVA:

- Accesibles a través de navegadores web, con protección mediante contraseñas o claves de acceso.
- Compatibles con tecnologías web 1.0 y 2.0, asegurando una integración moderna.
- Interfaces gráficas intuitivas que organizan los módulos de estructura moderada.
- Módulos para la gestión de cursos, administración académica, calendarios, materiales digitales, actividades y evaluación del aprendizaje.
- Adaptabilidad a las necesidades del usuario mediante roles específicos (administrador, profesor, tutor y estudiante) con privilegios personalizados.
- Comunicación e interacción efectiva entre estudiantes y docentes.
- Diversidad de actividades implementadas en los cursos.
- Recursos para el seguimiento y evaluación del progreso de los estudiantes.

Al momento de seleccionar una plataforma o herramienta tecnológica es posible clasificar dos tipos de características fundamentales: técnicas y pedagógicas, a continuación, se describen los aspectos considerados.

1. Características técnicas:

- **Tipo de licencia:** se destacan tres tipos de licencia que pueden encontrarse en las plataformas educativas:
 - **Licencia propietaria:** es necesario pagar una suscripción o adquisición, debido a que el software es desarrollado y controlado por una empresa, limitando su modificación o distribución.
 - **Licencia gratuita:** no representa un costo para el usuario, pero sin embargo puede tener limitaciones en cuanto a funcionalidades o uso comercial, y no siempre permite modificar el código.
 - **Licencia de código abierto:** proporciona acceso al código fuente del sistema, otorgando la posibilidad de modificar, adaptar y redistribuir de acuerdo con las necesidades del usuario o institución.
 -
- **Idioma y soporte multilingüe:** es primordial que la plataforma ofrezca compatibilidad con distintos idiomas, para permitir la personalización del contenido según el público objetivo, mejorando la accesibilidad y la experiencia de los estudiantes de distintos contextos culturales y lingüísticos.
- **Sistema operativo y tecnología empleada:** se debe garantizar la compatibilidad del sistema con los dispositivos y plataformas utilizados por los usuarios, esto incluye evaluar si la herramienta funciona adecuadamente en distintos sistemas operativos (como Windows, macOS, Linux o Android) y si las tecnologías que emplea (como bases de datos, lenguajes de programación o servidores) son compatibles con los recursos disponibles en la organización.

- **Documentación de apoyo:** verificar la existencia de documentación clara y detallada para los distintos tipos de usuarios (administradores, docentes y estudiantes) con el objetivo de garantizar un uso eficiente de la plataforma. Esta documentación puede abarcar áreas como manuales, tutoriales, guías en línea y preguntas frecuentes.
- **Comunidad de usuarios:** es factible que la plataforma disponga de una comunidad activa de usuarios y técnicos. Dando soporte de soluciones a problemas comunes, actualizaciones, mejoras constantes y asesoría en la implementación o personalización del sistema, favoreciendo el intercambio de experiencias y buenas prácticas, lo que contribuye al enriquecimiento del proceso de enseñanza-aprendizaje, asegura un respaldo técnico continuo y fortalece la sostenibilidad de la plataforma a largo plazo.

2. Características pedagógicas:

- **Gestión y administración del curso:** el sistema debe ofrecer funcionalidades que permitan organizar y controlar eficientemente los cursos, facilitando la planificación de contenidos, la asignación de actividades y el seguimiento del progreso general.
- **Comunicación e interacción entre usuarios:** debe contar con herramientas que promuevan la interacción entre docentes y estudiantes, así como entre los propios participantes, a través de foros, chats, mensajería interna o videoconferencias, facilitando la colaboración, el intercambio de ideas y la resolución de dudas en tiempo real, fortaleciendo así el aprendizaje colectivo y el sentido de comunidad dentro del entorno virtual.
- **Desarrollo e integración de contenidos:** debe otorgar la facilidad de la creación, integración y modificación de materiales educativos de forma sencilla y flexible, incluyendo imágenes, enlaces, videos y otros recursos multimedia que enriquezcan el proceso de aprendizaje.
- **Diseño de actividades interactivas:** Una buena plataforma educativa debe facilitar la incorporación de actividades dinámicas que motiven la participación del estudiante, tales como cuestionarios, juegos didácticos, simulaciones, videos interactivos o retos basados en problemas.

- **Fomento del trabajo colaborativo:** Debe incluir funciones que estimulen la cooperación entre los estudiantes, como proyectos en grupo, wikis o espacios de discusión compartida, fortaleciendo así las habilidades de trabajo en equipo y el aprendizaje social.
- **Evaluación y seguimiento del aprendizaje:** La posibilidad de aplicar evaluaciones y monitorear el desempeño de los estudiantes es clave. Esto puede lograrse mediante calificaciones automatizadas, reportes de avance y retroalimentación personalizada.
- **Entornos personalizables:** Finalmente, es importante que cada usuario tenga la posibilidad de adaptar el entorno de aprendizaje a sus propias necesidades. Esto puede incluir la personalización del idioma, accesibilidad visual, organización del contenido y otras preferencias que contribuyan a una experiencia más cómoda y eficaz.

Existen cuatro características básicas y esenciales, que cualquier plataforma de e-learning debería tener:

- **Interactividad:** conseguir que la persona que está usando la plataforma tenga conciencia de que es el protagonista de su formación.
- **Flexibilidad:** conjunto de funcionalidades que permiten que el sistema de e-learning tenga una adaptación fácil en la organización donde se quiere implantar, en relación con la estructura institucional, los planes de estudio de la institución y, por último, a los contenidos y estilos pedagógicos de la organización.
- **Escalabilidad:** capacidad de la plataforma de e-learning de funcionar igualmente con un número pequeño o grande de usuarios.
- **Estandarización:** Posibilidad de importar y exportar cursos en formatos estándar como SCORM.

2.3 Sistemas de Gestión de Aprendizaje

Un Sistema de Gestión del Aprendizaje (SGA), también conocido como Learning Management System (LMS), es una plataforma tecnológica alojada en la web, cuyo propósito es administrar, distribuir y evaluar procesos de enseñanza-aprendizaje en entornos educativos y corporativos. Su función principal es facilitar la creación, organización y seguimiento de cursos en línea, permitiendo a los docentes estructurar contenidos, gestionar evaluaciones y monitorear el progreso de los estudiantes.

Los SGA han revolucionado la enseñanza al permitir que el proceso educativo trascienda las barreras físicas y temporales. Gracias a estas plataformas, es posible implementar metodologías de aprendizaje adaptativo, enseñanza personalizada y educación autogestionada. Además, integra herramientas compatibles con estándares como SCORM y xAPI, garantizando la interoperabilidad de los contenidos educativos en diferentes plataformas.

Características de un SGA

Los SGA poseen diversas características que los hacen esenciales en la educación a distancia y en modelos de aprendizaje híbrido. Algunas de sus principales funcionalidades incluyen:

- **Gestión de usuarios y roles:** Permiten la asignación de perfiles como administradores, docentes, tutores y estudiantes, cada uno con diferentes niveles de acceso y permisos dentro del sistema.
- **Organización de contenidos:** Facilitan la carga y estructuración de materiales de aprendizaje en diferentes formatos (textos, vídeos, enlaces, cuestionarios interactivos, etc.).
- **Evaluación y seguimiento:** Integra herramientas para la creación de pruebas, tareas y cuestionarios, además de mecanismos de retroalimentación y análisis de desempeño.
- **Comunicación e interacción:** Disponen de foros, chats y sistemas de mensajería para fomentar la interacción entre estudiantes y docentes.
- **Reportes y análisis:** Proporcionan datos sobre la participación y el progreso de los estudiantes, facilitando la toma de decisiones pedagógicas.

En la Tabla 1 se hace una comparación de algunos SGA.

Tabla 1

Sistemas de gestión de aprendizaje

Nombre	Ventajas	Desventajas
Moodle https://download.moodle.org/	• Código abierto y gratuito. • Amplia comunidad de usuarios y desarrolladores. • Soporta múltiples idiomas. • Compatible con diversos dispositivos.	• Curva de aprendizaje pronunciada. • Instalación y mantenimiento complejos. • La interfaz puede resultar poco intuitiva.
Chamilo https://chamilo.org/download/	• Código abierto y gratuito. • Interfaz amigable y fácil de usar. • Compatible con dispositivos móviles. • Buena integración con otras aplicaciones.	• Comunidad de usuarios pequeña. • Funcionalidades avanzadas limitadas. • Menos extensible que algunas plataformas comerciales.
Google Classroom https://classroom.google.com/	• Propietario y gratuito. • Integración con herramientas de Google. • Interfaz sencilla y fácil de usar. • Accesible desde cualquier dispositivo con internet.	• Funcionalidades limitadas para cursos complejos. • Dependencia de Google. • Falta de soporte técnico dedicado.
Edmodo https://new.edmodo.com/	• Propietario y gratuito. • Interfaz similar a redes sociales, lo que facilita su uso. • Comunicación fluida entre estudiantes y docentes. • Herramientas para seguimiento del progreso académico.	• Limitaciones en la personalización. • Algunas funciones avanzadas requieren pago. • Límite de almacenamiento de datos

Tabla 1

Comparación de sistemas de gestión de aprendizaje (continuación)

Nombre	Ventajas	Desventajas
Blackboard Learn https://www.blackboard.com/teaching-learning/learning-management/blackboard-learn	• Amplias funcionalidades para la gestión educativa. • Herramientas avanzadas de evaluación y seguimiento. • Integración con diversas plataformas y servicios.	• Costos elevados de licencia. • Requiere capacitación. • Interfaz compleja para nuevos usuarios.
Canvas LMS https://www.instructure.com/canvas/	• Propietario y de pago. • Interfaz moderna y fácil de usar. • Altamente personalizable. • Soporte técnico y actualizaciones constantes.	• Costos asociados a la licencia. • Dependencia del proveedor para mantenimiento y soporte.

2.4 Diseño Instruccional

Los cambios en la Educación ocasionados por la integración de las Tecnologías de la Información y la Comunicación TIC, han hecho que las instituciones educativas estén interesadas en la utilización de nuevas metodologías en el proceso educativo. Esta integración hace que dichas instituciones reflexionen sobre la necesidad de producir materiales educativos y procesos de formación, adaptados a las nuevas necesidades del entorno educativo y de los estudiantes de hoy (Aguedelo, 2009).

El diseño instruccional (DI) es un procedimiento para desarrollar un programa educativo o de capacitación, currículos o cursos de manera secuencial y auténtica. Mediante un proceso sistemático orientado a la planificación y estructuración, cuyo objetivo es optimizar la enseñanza y garantizar el cumplimiento de los objetivos educativos (Branch & Merrill, 2011).

El DI implica la selección y aplicación de métodos, materiales y estrategias didácticas que facilitan la adquisición de conocimientos y habilidades en los estudiantes. Con el avance de la tecnología este concepto ha evolucionado integrando herramientas digitales y TIC para mejorar la calidad de la educación, fomentando la construcción del conocimiento y potenciando la interacción y el aprendizaje autónomo.

El planeamiento de la actividad del modelo instruccional consiste en el control detallado del ambiente y los procesos internos del estudiante, para que durante su ejecución se emplee elementos del ambiente como son el docente, alumno y contenidos (Quispe Zabaleta, 2019).

Es posible identificar diversos modelos de diseño instruccional, estos modelos son esenciales en la educación presencial y virtual, ya que proporcionan una base sistemática para organizar los materiales, definir objetivos de aprendizaje y seleccionar las mejores estrategias pedagógicas para lograr los resultados esperados. Entre sus principales características se encuentran:

- **Estructura clara:** organiza el contenido en módulos o unidades que faciliten la comprensión.
- **Orientado a objetivos:** definir qué conocimientos y habilidades deben adquirirse.
- **Adaptabilidad:** se puede aplicar en distintos niveles educativos y contextos.
- **Uso de estrategias didácticas:** integra herramientas y metodologías que favorecen el aprendizaje activo.
- **Evaluación del aprendizaje:** permite medir el progreso y ajustar el curso según sea necesario.

Se plantea como un proceso sistémico con actividades interrelacionadas que nos permiten crear ambientes que realmente faciliten, de forma mediada, los procesos de construcción del conocimiento. Si estos ambientes de aprendizaje no utilizan un diseño instruccional adecuado a la modalidad virtual no seguirán una planificación apropiada del proceso formativo con una propuesta didáctica definida y, por ello, los beneficios de las actividades de aprendizaje pueden verse disminuidos notablemente. Por tanto, el diseño instruccional no debe dejarse de lado en la producción e implementación de ningún recurso educativo o ambiente virtual de aprendizaje, sino que sirve como garantía de rigor y validez de todo el proceso (Belloch, 2017).

2.4.1 Modelo ADDIE

Es de los modelos más reconocidos y utilizados, se caracteriza por ser flexible, adaptable y permitir la retroalimentación en cada fase. Sirve como guía estructurada para diseñar, desarrollar y evaluar cursos o programas educativos, garantizando que los contenidos, métodos y recursos estén alineados con los objetivos de aprendizaje y las necesidades de los estudiantes. En la Figura 1 se presentan las fases de este modelo

Figura 1

Etapas de ADDIE



Nota. Elaboración propia.

Su nombre está basado en sus cinco fases:

1. **Análisis:** centrado en “qué” enseñar, analizando y detectando las características del aprendizaje y las necesidades de los estudiantes, los objetivos y las habilidades a ser logradas. Además, también hace un análisis de los recursos existentes, las particularidades de la audiencia a la que se dirigirá el programa y otras consideraciones, cuyo resultado será la descripción de una situación y sus necesidades.

2. **Diseño:** de acuerdo con los resultados de la fase de análisis procede a planear el “cómo” enseñarlo mediante una estrategia y el desarrollo de la instrucción. Delimita la forma de alcanzar las metas educativas y ampliar sus fundamentos
3. **Desarrollo:** comienza con la creación de los planes de las lecciones que estudiarán los estudiantes, así como materiales didácticos sobre los que trabajarán. Se desarrollan las categorizaciones, instrucciones, documentos de apoyo y todos los medios que serán usados como acompañamiento de la instrucción.
4. **Implementación:** se realiza la ejecución y puesta en práctica de la acción formativa con la participación de los alumnos.
5. **Evaluación:** se ejecuta la evaluación de los resultados de los estudiantes, además de analizarlos. Prosiguiendo con la evaluación de la eficacia y la eficiencia de la instrucción, la evaluación debe estar presente durante todo el proceso de diseño instruccional dentro de las fases, entre las fases, y después de la implementación.

2.4.2 Modelo ASSURE

El modelo ASSURE se especializa en la implementación de tecnología en la enseñanza, permite planificar y ejecutar clases o cursos de manera efectiva, asegurando que los recursos tecnológicos y didácticos se usen estratégicamente para que los estudiantes aprendan de manera activa y significativa. En la Figura 2 se presentan las fases de este modelo.

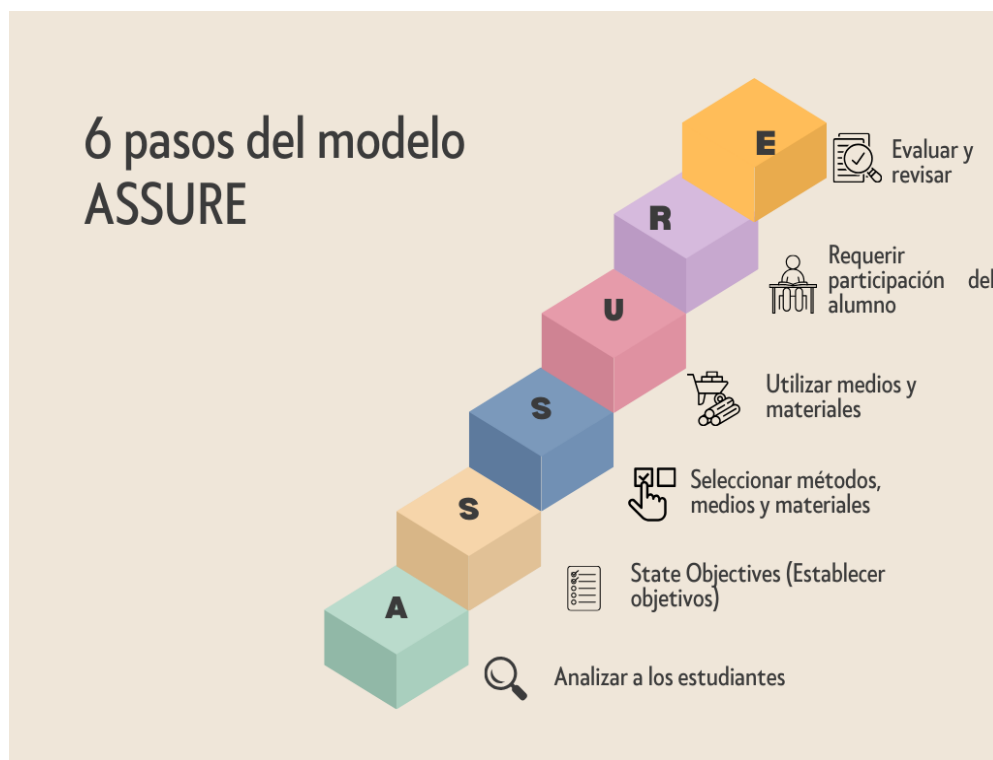
A continuación, se definen sus seis fases:

- **Analizar a los estudiantes:** identificar las características del público objetivo, como su nivel de conocimientos previos, estilos de aprendizaje y habilidades.
- **Establecer objetivos (State Objectives):** definición de los objetivos de aprendizaje en términos medibles.
- **Seleccionar métodos, medios y materiales:** deben elegirse estrategias didácticas y recursos tecnológicos adecuados.

- **Utilizar medios y materiales:** implementación de los recursos en la enseñanza.
- **Requerir participación del alumno:** diseño de actividades interactivas que promuevan el aprendizaje activo.
- **Evaluar y revisar:** se mide la efectividad del proceso y se realizan mejoras.

Figura 2

Etapas de ASSURE



Nota. Elaboración propia.

El modelo ASSURE, es flexible, fácil de diseñar y útil en cualquier ambiente de aprendizaje. Contribuye a mejorar la planeación de los cursos, seleccionar los medios y recursos, además de que permite que los profesores desarrollen su propio material. Facilita el logro de los objetivos y por ende el éxito del aprendizaje del estudiante, porque permite durante el proceso evaluar y retroalimentar los avances en su aprendizaje (Benítez Lima, 2010).

2.4.3 Modelo GAGNÉ

Ligada a procesos cognitivos y conductuales, Gagné establece que las tareas de aprendizaje de las habilidades intelectuales se pueden organizar en base a: “reconocimiento de estímulos, generación de respuestas, seguimiento de procedimientos, uso de terminología, discriminación, formación de conceptos, aplicación de reglas y resolución de problemas”.

A continuación, se definen las fases de Gagné:

- **Ganar la atención:** es necesario generar interés con un estímulo visual y sonoro, para provocar la curiosidad de los estudiantes. (videos, preguntas, problemas, ejemplos llamativos).
- **Informar los objetivos:** se explica a los estudiantes lo que se espera que logren al final del curso, provocando la expectativa con el objetivo de completar alguna tarea.
- **Estimular recuerdos conocimiento previo:** Se asocia el conocimiento nuevo con el conocimiento o experiencia personal, buscando el diálogo, la discusión, argumentación.
- **Presentar el material:** el contenido se organiza de manera que pueda ayudar a adquirir nuevos conocimientos de manera agradable usando diversos medios.
- **Producir la ejecución:** permite codificar la información usando diversas estrategias para desarrollar la nueva habilidad adquirida.
- **Dar feedback:** el docente proporciona comentarios y un conjunto de ayudas para reflexionar sobre lo aprendido.
- **Evaluar el desempeño:** los estudiantes demuestran lo aprendido sin entrenamiento previo, solo depende de sus capacidades y del conocimiento internalizado.
- **Promover la retención la transferencia:** consiste en adquirir conocimientos y habilidades para solucionar otros problemas, estos se pueden adquirir por retención.
- **Guiar el proceso:** el docente desarrolla estrategias para monitorear el proceso de aprendizaje.

El modelo de Gagné consta de 9 pasos, mismos que se muestran en la Figura 3.

Figura 3

Etapas de Gagné



Nota. Elaboración propia.

2.4.4 Modelo de Dick y Carey

El modelo de Dick y Carey fue desarrollado por Walter Dick y Lou Carey en 1978. Este enfoque considera la instrucción como un sistema integral, dentro de la cual cada uno de sus elementos se encuentra interrelacionado de manera que un cambio en alguno de ellos repercute en el resto. Se distingue por su carácter detallado y prescriptivo, lo que lo convierte en un referente ampliamente utilizado en el ámbito del diseño instruccional.

El modelo se estructura en nueve fases secuenciales que guían el desarrollo de la instrucción:

- Identificación de metas instruccionales: definir con claridad los objetivos generales de aprendizaje.
- Análisis instruccional: determinar las habilidades y conocimientos previos necesarios.
- Análisis de los aprendices y del contexto: considerar las características de los estudiantes y el entorno educativo.
- Redacción de objetivos de desempeño: especificar lo que se espera que logren los estudiantes al finalizar el proceso.
- Desarrollo de instrumentos de evaluación: diseñar pruebas o mecanismos para medir los logros de aprendizaje.
- Desarrollo de la estrategia instruccional: establecer métodos y actividades de enseñanza adecuados.
- Desarrollo y selección de materiales de instrucción: crear o seleccionar los recursos didácticos pertinentes.
- Diseño y desarrollo de la evaluación formativa: aplicar evaluaciones durante la instrucción para realizar ajustes.

- Diseño y desarrollo de la evaluación sumativa: evaluar de manera global la efectividad del curso.

A continuación, en la Figura 4 se muestran las etapas de Dick y Carey.

Figura 4

Etapas de Dick y Carey



Nota. Elaboración propia.

El modelo de Dick y Carey se caracteriza por su precisión metodológica, lo que permite un control riguroso en el diseño de programas de enseñanza, garantizando la coherencia entre los objetivos, las actividades y las evaluaciones. Si bien puede considerarse más complejo y demandante en comparación con otros modelos como ADDIE, resulta especialmente útil en contextos donde se busca asegurar una planeación minuciosa y una evaluación continua del proceso de aprendizaje.

La formación de profesores en el diseño instruccional, así como la capacitación a estudiantes son aspectos que no deben obviarse. Implica no sólo un compromiso individual, sino denota la necesidad de que todo proyecto de diseño instruccional debe vincularse a estrategias institucionales que clarifiquen tanto intenciones como procesos de gestión para el aprendizaje y faciliten no sólo aspectos de conexión e infraestructura, si no de procedimiento o clima organizacional que apoye el desarrollo de las actividades del proceso enseñanza-aprendizaje (Esquivel Gámez, 2014).

CAPÍTULO 3. Estado del Arte.

En este capítulo se analizan estudios y materiales previos que han contribuido a la enseñanza de la programación en entornos virtuales, identificando fortalezas y áreas de mejora en la implementación de cursos diseñados para este propósito.

3.1 Android: programación de dispositivos móviles a través de ejemplos

La vasta extensión de software de desarrollo de Android (SDK) puede resultar abrumadora para quien se acerca por primera vez a este sistema. La bibliografía existente suele estar dirigida a profesionales y programadores experimentados, los libros introductorios suelen estar enfocados a construir aplicaciones completas, mismas que debido a su complejidad se vuelven frustrantes para los estudiantes ya que requiere conocimientos previos (Soriano, 2019).

Soriano decide publicar un libro en el cual, a través de ejemplos prácticos, facilita el aprendizaje de la programación en Android, abordando desde conceptos fundamentales hasta el desarrollo de aplicaciones funcionales, el autor estructura el contenido de manera didáctica, permitiendo a los lectores comprender la lógica de la programación móvil mediante la implementación progresiva de ejercicios. Además, destaca el uso de Android Studio y Java como herramientas clave para el desarrollo de aplicaciones, promoviendo un enfoque basado en la práctica para fortalecer el aprendizaje.

Se implementa un enfoque práctico que promueve la enseñanza del desarrollo de aplicaciones en Android, a través del aprendizaje basado en ejemplos directamente con código, brindando una estructura progresiva que permita a los estudiantes comprender desde los fundamentos del desarrollo móvil hasta conceptos avanzados, promoviendo un aprendizaje escalonado y efectivo.

Obteniendo resultados favorables, debido a la facilidad de asimilación de conocimientos y optimización de la capacidad de resolución de problemas en el desarrollo de aplicaciones móviles.

Se destaca el uso de Android Studio como entorno de desarrollo, ya que proporciona las herramientas necesarias para la creación, prueba y optimización de aplicaciones, facilitando el proceso de aprendizaje y mejorando la productividad del estudiante. Además, destaca el uso de XML para la definición de interfaces gráficas y la implementación de Kotlin y Java como lenguajes de programación principales en el desarrollo de aplicaciones Android, haciendo énfasis en las ventajas y diferencias entre ambos.

Se incorporan componentes esenciales para la construcción de aplicaciones móviles, como, por ejemplo: *activities (actividades)*, *fragments (fragmentos)*, servicios y bases de datos locales. Estos elementos son presentados con ejemplos detallados, permitiendo a los estudiantes no sólo comprender su funcionamiento, sino que también les brinda el conocimiento para aplicarlos en la creación de proyectos propios. Asimismo, se enfatiza el uso de buenas prácticas en programación, incluyendo el manejo eficiente de recursos, la optimización del rendimiento de las aplicaciones y el cumplimiento de los lineamientos de diseño establecidos por Google en su *Material Design*.

También se aborda la necesidad de integrar en la enseñanza estrategias didácticas que favorezcan el aprendizaje autónomo, proporcionando guías y ejercicios que permitan a los estudiantes desarrollar su capacidad de resolución de problemas. En este sentido, su metodología se alinea con modelos instruccionales orientados a la enseñanza de programación en entornos digitales, facilitando la adaptación del contenido a diferentes niveles de experiencia y aprendizaje.

La relevancia del trabajo de Soriano (2019) radica en su contribución a la formación de nuevos desarrolladores de aplicaciones móviles mediante un enfoque práctico y accesible. Su propuesta didáctica respalda la enseñanza de la programación en entornos virtuales de aprendizaje, convirtiéndose en un referente valioso para la estructuración de cursos de desarrollo móvil que buscan combinar teoría.

3.2 Desarrollo de competencias en programación de aplicaciones móviles.

Barrera, Vega y Morales (2020) llevaron a cabo una investigación orientada al desarrollo de competencias digitales en programación de aplicaciones móviles en estudiantes de noveno

grado, con el propósito de responder a la creciente necesidad de fortalecer estas habilidades desde niveles educativos tempranos. La enseñanza de la programación enfrenta diversos desafíos, como la falta de metodologías didácticas efectivas, la escasez de recursos accesibles y la percepción de la programación como una disciplina compleja y difícil de abordar. Ante este panorama, los autores llevaron a cabo un análisis para posteriormente implementar tres estrategias pedagógicas innovadoras con el fin de optimizar el aprendizaje y facilitar la comprensión de conceptos clave en el desarrollo de aplicaciones móviles.

La investigación se estructuró en torno a la aplicación de Moodle como plataforma de aprendizaje en línea, el diseño de una página web educativa que centraliza los recursos didácticos y la utilización de App Inventor como una herramienta visual que permite a los estudiantes programar de manera intuitiva y sin necesidad de conocimientos avanzados en lenguajes de programación tradicionales, dichas estrategias son organizadas en seis módulos temáticos que abarcan desde conceptos básicos de programación hasta el desarrollo de aplicaciones funcionales, brindando a los estudiantes un entorno de aprendizaje estructurado y progresivo.

Integrar las TIC para enseñar programación de aplicaciones móviles, Apps, permite desarrollar en los estudiantes, competencias computacionales clave para la ciudadanía digital. Los resultados obtenidos mostraron un impacto positivo en el aprendizaje, destacando que los estudiantes no solo mejoraron sus competencias técnicas en programación, sino que también desarrollaron habilidades transversales como el pensamiento lógico, la resolución de problemas, la creatividad y el trabajo en equipo. Asimismo, se observará un incremento en la autonomía de los alumnos, quienes fueron capaces de enfrentar desafíos de programación con mayor seguridad y confianza en sus habilidades.

Se concluye que la enseñanza de la programación mediante entornos digitales estructurados y herramientas accesibles es una estrategia efectiva para fomentar el interés y la participación de los estudiantes, debido a la combinación de plataformas de educación virtual con software de desarrollo intuitivo fomentando un aprendizaje más dinámico y significativo, finalmente resultando en la formación de una nueva generación de programadores con habilidades fundamentales.

3.3 Plataforma e-learning de apoyo al estudio en estudiantes universitarios

Los autores (Nácher *et al.*, 2018) presentan en su estudio la plataforma GoKoan, una herramienta de e-learning diseñada para optimizar el estudio en estudiantes universitarios mediante el uso de metodologías innovadoras y tecnologías adaptativas. Su trabajo surge de la necesidad de mejorar los procesos de aprendizaje en educación superior, proporcionando a los alumnos un entorno flexible y personalizado que facilite la adquisición y retención del conocimiento.

Se hace énfasis en el desarrollo y evaluación de la plataforma GoKoan, misma que incorpora un sistema de planificación inteligente basado en técnicas de inteligencia artificial y adaptabilidad al ritmo de cada estudiante. Organizando los contenidos de manera estructurada, permitiendo a los usuarios gestionar su tiempo de estudio de forma eficiente y ofreciendo estrategias de aprendizaje personalizadas según su progreso y desempeño. Además, integra elementos de gamificación para aumentar la motivación y la participación de los estudiantes.

Uno de los principales avances del estudio fue la validación de la efectividad de la plataforma como un recurso educativo innovador en el ámbito educativo. Los resultados de la investigación evidenciaron que el uso de plataformas de e-learning con enfoques personalizados puede generar un impacto positivo en el aprendizaje de los estudiantes, promoviendo una mayor autonomía y compromiso con su formación académica. A su vez, se identificó que la combinación de estrategias como la planificación automatizada, la gamificación y la adaptabilidad favorecen un aprendizaje más eficiente y motivador.

La implementación de herramientas tecnológicas avanzadas en educación superior no solo facilita el acceso al conocimiento, sino que también transforma la manera en que los estudiantes interactúan con los contenidos académicos. La plataforma GoKoan representa un modelo de innovación en el ámbito del e-learning, demostrando que la integración de metodologías adaptativas y sistemas de aprendizaje personalizados puede mejorar significativamente los resultados educativos. A continuación, en la Figura 5 se muestra la pantalla de inicio de GoKoan.

Figura 5

Pantalla de inicio de GoKoan



Fuente. <https://www.gokoan.com> (2025)

3.4 Capacitación para mejorar competencias de acompañamiento virtual.

Debido a la necesidad de fortalecer las habilidades pedagógicas en entornos digitales, así como el creciente uso de plataformas de e-learning en la educación universitaria. Chujutalli (2023) desarrolló un estudio enfocado en el diseño de un curso de capacitación a distancia con el propósito de mejorar las competencias de acompañamiento virtual en docentes universitarios. Busca dar solución a la problemática de que muchos docentes, aunque expertos en sus respectivas áreas de conocimiento, carecen de formación específica en estrategias didácticas para la enseñanza en línea, lo que puede afectar la calidad del aprendizaje de los estudiantes.

El propósito del estudio es la elaboración de un curso estructurado bajo un modelo de educación a distancia, con un enfoque en metodologías activas y herramientas tecnológicas que faciliten la interacción y el seguimiento del aprendizaje. Se buscó que los docentes puedan adquirir competencias clave para gestionar entornos virtuales de enseñanza, fomentar la participación de los estudiantes y aplicar estrategias efectivas de acompañamiento en línea. Entre los principales avances, se destaca la estructuración del curso en módulos progresivos que abarcan desde la introducción a las plataformas virtuales hasta el diseño de actividades

interactivas y la evaluación del aprendizaje en entornos digitales. Además, se incorporan recursos multimedia y estrategias de aprendizaje colaborativo para garantizar una experiencia formativa dinámica y efectiva.

Hoy en día se ha vuelto indispensable la formación docente en competencias digitales que permitan garantizar un proceso educativo de calidad en entornos virtuales. Teniendo en cuenta que un curso de capacitación bien diseñado puede contribuir significativamente al desarrollo de habilidades de acompañamiento virtual, favoreciendo una enseñanza más efectiva y adaptada a las exigencias del e-learning.

Los resultados obtenidos evidenciaron mejoras en la capacidad de los participantes para gestionar entornos virtuales de aprendizaje, aplicar metodologías innovadoras y utilizar plataformas digitales de manera eficiente. Se observó que, tras la implementación del curso, los docentes mostraron mayor confianza en su labor de acompañamiento virtual y en la aplicación de estrategias didácticas centradas en el estudiante.

3.5 Programación informática y robótica en la enseñanza básica.

Delgado (2015) explora la incorporación de la programación informática y la robótica en la enseñanza básica, subrayando su papel en el desarrollo del pensamiento lógico, la resolución de problemas y la creatividad en los estudiantes. Su estudio parte de la premisa de que la educación en programación no solo debe estar reservada para niveles avanzados, sino que puede integrarse desde etapas tempranas como un componente clave en la formación de competencias digitales.

El autor destaca que la programación permite a los estudiantes interactuar activamente con la tecnología, pasando de ser meros consumidores a creadores de soluciones innovadoras. Para ello, examina distintos enfoques pedagógicos que han sido implementados en diversas instituciones educativas, haciendo especial énfasis en el uso de herramientas visuales como Scratch y entornos de programación basados en bloques. Estas plataformas, diseñadas específicamente para principiantes, facilitan la comprensión de estructuras algorítmicas sin

necesidad de conocimientos avanzados de sintaxis, permitiendo que los alumnos desarrollen proyectos de manera intuitiva.

Uno de los avances claves señalados en la investigación es la posibilidad de integrar la programación y la robótica como un eje transversal dentro del currículo escolar. Delgado (2015) argumenta que estas disciplinas no sólo fortalecen el aprendizaje en asignaturas como matemáticas y ciencias, sino que también promueven habilidades esenciales para el siglo XXI, como el pensamiento computacional y la capacidad de abstracción. En este contexto, la robótica educativa se presenta como un recurso valioso para reforzar conceptos matemáticos y físicos mediante la experimentación práctica, fomentando el aprendizaje basado en la exploración y el descubrimiento.

Entre las motivaciones que impulsan este estudio, el autor enfatiza la creciente demanda de habilidades digitales en el mercado laboral y la necesidad de que los sistemas educativos se adapten a este nuevo panorama. Además, resalta la importancia de capacitar a los docentes en metodologías de enseñanza de la programación, pues su desconocimiento sobre estas herramientas puede convertirse en una barrera para su implementación efectiva en el aula.

Los resultados obtenidos muestran que los estudiantes que han participado en programas de enseñanza de programación y robótica han desarrollado una mejor capacidad para estructurar problemas, analizar soluciones y aplicar el pensamiento algorítmico en diferentes contextos. Además, se ha observado un incremento en su interés por la tecnología y la innovación, así como una mayor disposición para resolver desafíos de manera autónoma.

En conclusión, la investigación de Delgado (2015) evidencia que la enseñanza de la programación y la robótica desde la educación básica no solo es posible, sino altamente beneficiosa. Sin embargo, también señala que su implementación requiere de estrategias adecuadas, infraestructura tecnológica suficiente y una formación docente continua que garantice el aprovechamiento óptimo de estas herramientas en los procesos educativos.

3.6 Discusión

La instrucción en el desarrollo de aplicaciones móviles se ha abordado desde múltiples perspectivas. derivado de las diversas problemáticas en torno a la enseñanza de la programación y el desarrollo de competencias digitales, así como diferentes estrategias para abordarlas. Se analizan 5 diferentes estudios que ofrecen un enfoque particular, evidenciando la importancia de adaptar las metodologías educativas a los avances tecnológicos y a las necesidades de los estudiantes.

La primera iniciativa, llamada “Android: programación de dispositivos móviles a través de ejemplos”. identifica la barrera existente dentro del aprendizaje de la programación para el desarrollo de aplicaciones móviles, especialmente por la falta de materiales didácticos que combinan teoría y práctica de manera efectiva. Como respuesta, su libro presenta una metodología basada en ejemplos prácticos, utilizando entornos de desarrollo como Android Studio, facilitando así la adquisición de habilidades técnicas a través de ejercicios aplicados a casos reales.

Por su parte la iniciativa “Desarrollo de competencias digitales en programación de aplicaciones móviles en estudiantes de noveno grado a través de tres estrategias pedagógicas” se centra en la dificultad que tienen los estudiantes de un grupo de noveno grado para desarrollar competencias digitales en el área de programación de móvil debido a la falta de estrategias pedagógicas adecuadas. Como solución a esta problemática los autores deciden implementar tres enfoques metodológicos distintos: enseñanza basada en proyectos, aprendizaje colaborativo y uso de entornos de desarrollo accesibles, demostrando que la combinación de estas estrategias mejora significativamente el desempeño de los estudiantes en programación.

La siguiente iniciativa enfoca su estudio en la optimización de las estrategias de estudio en educación superior, identificando la dificultad que enfrentan los estudiantes para organizar y gestionar su aprendizaje de manera autónoma. Como solución, desarrollan *GoKoan*, una plataforma e-learning basada en técnicas de aprendizaje adaptativo que permite personalizar el proceso de estudio según el perfil del estudiante, fomentando así una mayor retención del conocimiento.

Por su parte, la cuarta iniciativa aborda la necesidad de mejorar las competencias de acompañamiento virtual en docentes universitarios, ya que la falta de formación en metodologías e-learning limita la efectividad de la educación a distancia. Para enfrentar este desafío, se implementa un curso de capacitación a distancia enfocado en fortalecer las habilidades pedagógicas de los docentes dentro de entornos virtuales de aprendizaje.

Finalmente, la iniciativa “Programación informática y robótica en la enseñanza básica” destaca la falta de integración de la programación y la robótica en la educación básica, lo que limita el desarrollo del pensamiento lógico y computacional desde etapas tempranas. Para solventar esta problemática, propone la inclusión de herramientas como Scratch y entornos de programación visuales, los cuales permiten a los estudiantes adquirir conocimientos de programación de manera intuitiva y sin barreras sintácticas complejas.

En conjunto, estos estudios evidencian la necesidad de adaptar los métodos de enseñanza de la programación a las necesidades de los estudiantes. Desde la incorporación de recursos didácticos y entornos adaptativos hasta la capacitación docente y la implementación de metodologías activas, cada enfoque contribuye a la construcción de un modelo educativo más dinámico y accesible para la formación en competencias digitales.

CAPÍTULO 4. Marco Metodológico

Para la realización de este proyecto, se llevó a cabo un análisis exhaustivo de diversas metodologías de trabajo y enseñanza, con el propósito de identificar aquellas que mejor se adaptan a los objetivos planteados. Este proceso permite seleccionar tres metodologías de trabajo encargadas de asegurar una implementación efectiva y accesible.

A continuación, se detalla cómo se incorporaron de manera específica:

1. El modelo ADDIE se adoptó como guía estructural del proyecto, ya que asegura que cada etapa del curso esté alineada con los objetivos de aprendizaje, promoviendo una planificación sistemática y la posibilidad de realizar ajustes continuos para optimizar la propuesta educativa.
2. ABP promovió el aprendizaje significativo y el desarrollo de competencias prácticas. Mediante la resolución de problemas vinculados con la programación móvil, los estudiantes aplican los conceptos teóricos en situaciones concretas, fortaleciendo así el pensamiento crítico, la autonomía y la capacidad de resolver retos con el lenguaje Kotlin en Android Studio.
3. El desarrollo iterativo e incremental se implementó como un esquema metodológico que facilita la construcción progresiva de conocimientos. A través de este enfoque, los contenidos se introducen gradualmente, incrementando la complejidad de los ejercicios y reforzando lo aprendido en cada etapa. Esto favorece un aprendizaje escalonado, que asegura la comprensión de los fundamentos antes de avanzar hacia nuevos desafíos en la programación móvil.

4.1 Metodología de diseño instruccional

Los modelos de diseño instruccional proporcionan una serie de pasos que permiten dar orden y organización de manera general a la estructura de un curso, dando como resultado la elaboración de un material preciso y conciso, haciendo énfasis en su público objetivo. Existen varios modelos de diseño instruccional, cada uno destaca por la cantidad de pasos, así como la definición de estos, surge la necesidad de una comparación entre algunos de los modelos

más conocidos tomando en cuenta sus características, así como las ventajas y desventajas que brindan al implementar un curso virtual. La comparación se muestra en la Tabla 2.

Tabla 2

Comparativo de modelos instruccionales

Modelo	Características	Ventajas	Desventajas
ADDIE	• Se compone de 5 fases. • Es un modelo de secuencia y estructurado. • Se enfoca en el diseño instructivo para la creación de cursos y materiales educativos.	• Enfoque sistemático y organizado. • Adaptable a las necesidades de los estudiantes. • Define metas concretas y medibles. • Mejora los materiales instruccionales. • Fomenta la colaboración entre diseñadores, docentes y estudiantes. • Evalúa la eficacia del plan de aprendizaje. • Asegura relevancia y efectividad del contenido con el tiempo.	• Es lineal y asume que cada fase se completa antes de pasar a la siguiente. • Difícil de adaptar a contextos y necesidades específicas. • Implementar el modelo puede ser costoso y demanda mucho tiempo. • Poco realista y práctico en algunas situaciones. • inadecuado para ciertos tipos de entornos de aprendizaje.
ASSURE	• Se compone de 6 fases. • Permite la interacción de la tecnología con los medios de enseñanza.	• Fomenta el uso de tecnologías y recursos multimedia. • Considera las características del estudiante. • Permite personalizar el aprendizaje. • Promueve la participación activa.	• Puede ser menos efectivo sin entornos tecnológicos. • Puede ser más complejo de aplicar en grandes grupos.

Tabla 2*Comparativo de modelos instruccionales (continuación)*

Modelo	Características	Ventajas	Desventajas
GAGNÉ	- Se compone de 9 fases. - Hace énfasis en el procesamiento cognitivo del estudiante. - Es útil para la enseñanza estructurada en diferentes niveles. - Se centra en cómo se recibe y retiene la información.	- Se adapta a diferentes estilos de aprendizaje. - Fomenta una enseñanza clara y estructurada. - Es aplicable a múltiples disciplinas y niveles educativos. - Se puede integrar con otros modelos.	- Flexibilidad limitada - Compatibilidad limitada con la tecnología. - No prioriza el aprendizaje colaborativo o basado en proyectos.
Dick y Carey	- Se compone de nueve fases secuenciales que conciben la instrucción como un sistema en el que los objetivos, actividades, materiales, estudiantes y evaluaciones se encuentran interrelacionados. - Permite la enseñanza como un sistema integral en el que cada componente está interrelacionado, lo que permite una planeación rigurosa y sistemática del proceso educativo.	- Ofrece Coherencia y alineación entre objetivos, actividades, materiales y evaluaciones. - Favorece la retroalimentación constante gracias a la evaluación formativa. - Claridad en el proceso al dividir la instrucción en fases específicas. - Es replicable y adaptable en diferentes contextos educativos, incluidos entornos virtuales.	- Requiere tiempo y recursos significativos. - Puede ser poco flexible en contextos educativos con limitaciones técnicas o de personal. - El nivel de detalle puede generar sobrecarga administrativa en la planeación. - No siempre resulta práctico para cursos cortos o proyectos con plazos reducidos.

Basándose en la comparación realizada entre los modelos de diseño instruccional realizada, se optó por la elección del modelo más adecuado considerando el cumplimiento de dos criterios:

1. **Simplicidad y eficiencia:** el modelo no debe ser excesivamente complejo ni extenso, debido a que podría dificultar su implementación por parte de los docentes. Un modelo demasiado largo o detallado podría provocar que ciertos pasos sean omitidos o aplicados de manera incorrecta.
2. **Evaluación continua:** es fundamental que contemple un proceso de evaluar de manera constante a los estudiantes operando bajo esquemas de evaluación periódica y registro de notas.

Tomando en cuenta estos criterios, el modelo **ADDIE** se presenta como la opción más adecuada, debido a que cumple con ambas condiciones. En consecuencia, los otros modelos incluyen una gran cantidad de pasos, dificultando su aplicación.

Es importante mencionar que ADDIE es más flexible y adaptable para diversos objetivos debido a que ofrece una estructura que promueve el desarrollo de una gran variedad de interacciones de instrucción, permitiendo su integración con distintas metodologías, incluidas aquellas basadas en TIC. Su estructura modular facilita su implementación sin comprometer la calidad del proceso de enseñanza-aprendizaje. Los resultados obtenidos en cada fase permiten al diseñador regresar a cualquier fase anterior.

4.2 Metodología de enseñanza

En el ámbito de la programación, donde la práctica y la resolución de problemas resultan fundamentales, la selección de una metodología pertinente se convierte en un factor decisivo para favorecer un aprendizaje significativo.

Las metodologías de enseñanza constituyen un elemento esencial en el diseño instruccional, pues orientan la manera en que se organiza el proceso de aprendizaje y la interacción entre docentes, estudiantes y contenidos. Su adecuada elección no solo influye en la adquisición de

conocimientos teóricos, sino también en el desarrollo de habilidades prácticas, el pensamiento crítico y la autonomía del estudiante.

En la Tabla 3 se presenta un cuadro comparativo de las metodologías de enseñanza: aprendizaje basado en problemas, aprendizaje basado en proyectos, aprendizaje experiencial y aprendizaje por descubrimiento.

Tabla 3

Comparativo de metodologías de enseñanza

Metodología	Características	Ventajas	Desventajas
Aprendizaje basado en problemas	• Se basa en la resolución de problemas reales o simulados. • Promueve la investigación y la aplicación del conocimiento. • Los alumnos trabajan de manera autónoma o en grupos.	• Fomenta el pensamiento crítico y la autonomía. Relaciona la teoría con la práctica. • Mejora la capacidad de análisis y resolución de problemas.	• Puede ser difícil para estudiantes sin conocimientos previos. • Requiere más tiempo y esfuerzo del instructor para guiar el proceso.
Aprendizaje basado en proyectos	• Los estudiantes trabajan en un proyecto final a lo largo del curso. • Integra múltiples habilidades y conocimientos. • El aprendizaje se da a través de la práctica.	• Motiva a los estudiantes al ver resultados tangibles. • Refuerza el aprendizaje a largo plazo. • Facilita el trabajo colaborativo y la creatividad.	• Puede generar frustración si no hay una guía clara. • La evaluación del aprendizaje puede ser subjetiva.
Aprendizaje experiencial	• Se basa en la experimentación con herramientas y software. • Requiere reflexión sobre la experiencia para consolidar el aprendizaje. • Integra ciclos de prueba y error.	• Facilita la asimilación de conocimientos prácticos. • Aumenta el compromiso y la participación del estudiante. • Permite aprender a partir de los errores.	• No siempre es adecuado para todos los estilos de aprendizaje. • Puede ser difícil de implementar en cursos teóricos.

Tabla 3*Comparativo de metodologías de enseñanza (continuación)*

Metodología	Características	Ventajas	Desventajas
Aprendizaje por descubrimiento	• Los alumnos construyen su propio conocimiento a partir de la exploración. • Se basa en la curiosidad y la experimentación.	• Desarrolla habilidades de autoaprendizaje. • Estimula la creatividad y el pensamiento crítico. • Genera un aprendizaje más significativo.	• Puede ser frustrante si el estudiante no encuentra las respuestas. • Requiere mayor planificación y seguimiento por parte del instructor.

Cada metodología analizada se distingue por su enfoque en la interacción, la resolución de problemas y la participación activa del estudiante en el proceso de aprendizaje. Dado que no todas las estrategias de enseñanza son igualmente efectivas para todos los contextos, es fundamental compararlas y analizar sus características, ventajas y desventajas para seleccionar la más adecuada en la enseñanza de la programación.

Considerando las características del curso, el aprendizaje basado en problemas destaca como la metodología de enseñanza más adecuada, debido a que permite a los estudiantes desarrollar habilidades de programación de manera práctica y contextualizada, fomentando la autonomía y el pensamiento crítico al presentar a los estudiantes problemas reales que deben resolver mediante la implementación de sus habilidades y conocimientos adquiridos.

Además, dicha metodología tiene un enfoque flexible que se adapta a distintos niveles de experiencia y permite integrar herramientas tecnológicas, alineándose con los objetivos del curso. Su estructura centrada en la resolución de problema facilita la aplicación del conocimiento en escenarios reales, mejorando la retención del aprendizaje y fortaleciendo la capacidad de los estudiantes para enfrentarse a desafíos propios del desarrollo de software.

4.2.1 Aprendizaje basado en problemas

El Aprendizaje Basado en Problemas (ABP) es una metodología educativa que pone en prioridad el aprendizaje del estudiante, la investigación y la reflexión, instruyendo a los alumnos hacia la resolución de problemas planteados por el profesor. En contraste con la tradicional exposición de contenidos por parte de la docente seguida de actividades prácticas, el ABP impulsa un enfoque más participativo, donde los estudiantes se involucran activamente en la resolución.

La propuesta metodológica ABP se ha convertido en una alternativa atractiva para ejecutar los cambios en los modelos educativos de la educación superior, en particular porque su entorno de aprendizaje es propicio para el desarrollo de habilidades de pensamiento de orden superior, como el pensamiento crítico (Bueno,2018).

Esta metodología se basa en la idea de que el conocimiento se construye de manera más efectiva cuando los estudiantes aplican conceptos teóricos en contextos prácticos. Además, promueve el trabajo colaborativo y el desarrollo de habilidades analíticas, ya que los estudiantes deben formular hipótesis, explorar posibles soluciones y evaluar sus resultados.

Los componentes centrales del modelo: contenido, contexto y conexión, tienen como funciones principales establecer las bases del problema, de tal manera que guarden relación directa con las metas y objetivos de aprendizaje previstos; contextualizar el dominio de conocimiento y, además, guiar a los estudiantes a construir marcos conceptuales integrados (Bueno, 2018). En la Figura 6 se muestran las ventajas de ABP.

De acuerdo con diversos autores, se puede establecer que las ventajas de ABP son las siguientes:

- Los estudiantes adquieren conocimientos como resultado de la resolución de problemas reales, impulsando su retención y comprensión.
- Promueve el análisis, la síntesis y la evaluación de información para encontrar soluciones efectivas.

- Los estudiantes asumen un rol activo en su aprendizaje, desarrollando la capacidad de investigar en fuentes de información confiables para posteriormente tomar decisiones fundamentando sus soluciones, impulsando así el aprendizaje de manera independiente.
- Al enfrentar problemas en conjunto, los participantes fortalecen la comunicación, la colaboración y el trabajo en equipo.
- La conexión con problemas relacionados a su campo profesional, los estudiantes adquieren habilidades prácticas para su desempeño, impulsado por aprendizaje más atractivo y relevante para los estudiantes.
- Puede aplicarse dentro de una amplia gama de disciplinas, tales como la programación hasta las ciencias sociales y médicas.

Figura 6

Ventajas del aprendizaje basado en la resolución de problemas



Fuente. Aulaplaneta (2015).

4.3 Desarrollo iterativo e incremental

Es una metodología ampliamente utilizada en la ingeniería de software y en el diseño instruccional. Este enfoque permite la construcción progresiva de sistemas mediante ciclos repetidos de planificación, desarrollo y evaluación. En lugar de desarrollar un producto completo desde el inicio, se realizan entregas parciales y funcionales en cada iteración, lo que facilita la incorporación de mejoras continuas basadas en la retroalimentación obtenida.

En el ámbito educativo, esta metodología es especialmente útil para la integración de Recursos Educativos Abiertos (REA) en el diseño instruccional y en las estrategias pedagógicas dentro de las modalidades de educación a distancia. Además, el uso de modelos iterativos e incrementales ha sido propuesto para la evaluación por competencias de los aprendizajes, permitiendo la metaevaluación del desempeño de los docentes y contribuyendo al desarrollo de la conciencia y reflexividad sobre su nivel de desarrollo en la competencia evaluativa (Jiménez et al., 2014).

En este proyecto, el desarrollo iterativo e incremental se implementa mediante la constante revisión y optimización de los módulos y actividades. Cada iteración incorpora ajustes basados en la retroalimentación de los participantes, asegurando que los contenidos sean comprensibles y que las actividades estén alineadas con los objetivos de aprendizaje. Este enfoque es especialmente útil en la enseñanza de programación, donde la evolución de las tecnologías exige que los materiales educativos sean dinámicos y adaptables.

Además, esta metodología facilita la integración de nuevas herramientas digitales y estrategias pedagógicas a lo largo del proceso. El desarrollo iterativo permite una mayor flexibilidad en la planificación y ejecución de cursos en línea, ya que se pueden realizar mejoras sin comprometer la estructura general del programa.

En conclusión, el desarrollo iterativo e incremental representa un enfoque efectivo para el diseño del presente curso, garantizando que cada módulo sea validado antes de avanzar al siguiente y permitiendo ajustes continuos para mejorar la experiencia de aprendizaje. Su implementación contribuye a un proceso de enseñanza-aprendizaje dinámico, centrado en la mejora progresiva de los contenidos y la interacción con los estudiantes.

CAPÍTULO 5. Desarrollo del curso.

En relación con la problemática identificada y los objetivos propuestos, se diseñó y desarrolló un curso en línea sobre programación básica en Kotlin, fundamentado en el Aprendizaje Basado en Problemas (ABP). Este enfoque permite a los estudiantes aprender mediante la resolución de situaciones reales, fomentando el pensamiento crítico y la aplicación práctica del conocimiento.

Además, se empleó el modelo ADDIE como guía en el diseño instruccional, asegurando una estructura clara y un proceso iterativo que facilita la evaluación y mejora continua del curso.

El curso desarrollado incluye los siguientes elementos:

- Contenido educativo estructurado y accesible.
- Ejercicios prácticos que refuercen el aprendizaje.
- Mecanismos de evaluación y retroalimentación efectiva.
- Acceso a herramientas de desarrollo que facilitan la programación en Kotlin.

En las siguientes secciones se describe el proceso de desarrollo del curso propuesto.

5.1 Análisis de recursos

En la fase de análisis del modelo ADDIE, se llevó a cabo un estudio detallado a través de fuentes de Internet, así como libros que permitió comprender las necesidades de aprendizaje y los requisitos del curso. Se identificó que existe una carencia de materiales accesibles y didácticos para la enseñanza de programación en Kotlin, especialmente para principiantes.

Además, se reconoce que los estudiantes suelen enfrentar dificultades en la comprensión de conceptos básicos de programación y en la aplicación práctica de estos conocimientos en un entorno de desarrollo real. Lo expuesto tiene como objetivo definir con claridad los

fundamentos del curso, los recursos didácticos digitales y las actividades que deberán realizar los estudiantes a partir del paradigma de ABP.

El curso desarrollado está compuesto de diez módulos estructurados de manera progresiva para guiar a los participantes en el aprendizaje de los conceptos básicos de Kotlin. A lo largo del curso, los estudiantes enfrentan desafíos que los llevan a comprender los fundamentos de la programación, desarrollar interfaces gráficas, gestionar y aplicar sus conocimientos en un problema basado en la metodología ABP.

Cada módulo está diseñado para guiar y fortalecer el aprendizaje mediante ejercicios prácticos y actividades dinámicas, asegurando que los participantes adquieran conocimientos teóricos y habilidades aplicables en el desarrollo de aplicaciones móviles. A continuación, se muestra una breve descripción de cada módulo:

- **Bienvenida**

Ofrece una introducción al curso describiendo las herramientas a utilizar: Kotlin y Android Studio.

- **Módulo 1: El despertar del programador**

Los estudiantes comparten su conocimiento previo acerca de conceptos básicos de la programación móvil, se familiarizan con la relación entre Kotlin y Android Studio dentro del desarrollo de aplicaciones móviles.

- **Módulo 2: Los tipos de datos emergen**

Los estudiantes exploran los tipos de datos y variables esenciales en Kotlin, comprendiendo su declaración y uso en la programación. A través de actividades interactivas, identifican y manipulan valores en una interfaz, lo que les permite familiarizarse con el entorno de Android Studio. Finalmente, consolidan sus conocimientos mediante ejercicios prácticos que los llevan a realizar su primera declaración de variables en Kotlin de manera autónoma.

- **Módulo 3: El entrenamiento aumenta de nivel**

Los participantes profundizan en el uso de operadores aritméticos en Kotlin, comprendiendo su aplicación a través de ejemplos prácticos. Además, exploran de manera autónoma otros operadores y sus usos en distintos contextos. Para consolidar el aprendizaje, el módulo culmina con dos pruebas con problemas aritméticos que ponen a prueba su ingenio y conocimientos, brindándoles la oportunidad de resolver problemas y comparar sus soluciones con las respuestas correctas proporcionadas por el curso.

- **Módulo 4: Fusión programática**

Los estudiantes exploran el concepto de funciones en Kotlin, comprendiendo sus elementos y los distintos tipos de funciones disponibles en el lenguaje. A través de un ejercicio práctico, aplican lo aprendido desarrollando una función para resolver un problema específico. Finalmente, consolidan su conocimiento mediante un video interactivo que, de manera dinámica, presenta preguntas sobre el tema, reforzando así su comprensión.

- **Módulo 5: Decisiones al límite**

Los participantes son felicitados por alcanzar la mitad del curso, además exploran el uso de las estructuras condicionales IF & WHEN, fundamentales en la programación con Kotlin. A través de ejemplos prácticos, comprenderán su funcionamiento y aplicabilidad en distintos escenarios. Posteriormente, pondrán a prueba sus conocimientos con una evaluación práctica que desafiará las habilidades adquiridas. Finalizando con un examen teórico para consolidar su aprendizaje.

- **Módulo 6: La transformación comienza**

Los estudiantes analizan el concepto de interfaz gráfica y sus principales componentes. Además, se familiarizan con los fundamentos básicos de XML, comprendiendo su relevancia en el diseño de interfaces en Android Studio. Para reforzar el aprendizaje, se realizará una evaluación mediante una sopa de letras que los participantes deberán resolver. Finalmente, desarrollarán una actividad creativa en la que imaginan y dibujan una interfaz gráfica, explicando sus elementos con base en lo aprendido en la lección.

- **Módulo 7: Una nueva dimensión**

Los participantes exploran los conceptos clave de los *Activities* en Android, comprendiendo su definición, usos y tipos principales. A través de ejemplos prácticos, se les guiará para entender su funcionamiento y aplicación en el desarrollo de aplicaciones móviles. Además, se incluirá un video instructivo donde aprenderán a crear un *Activity* en Android Studio y a generar un simulador virtual, sentando las bases para el desarrollo de sus propias aplicaciones.

- **Módulo 8: La fase de interfaz ha comenzado**

Los participantes identifican qué es un widget dentro del desarrollo de interfaces gráficas en Android, explorando los widgets más comunes como TextView, ImageView, Button y EditText, aprendiendo a declararlos correctamente en XML. Además, estudiarán su comportamiento a través de la implementación de algunos atributos como text, hint, src, layout_width, layout_height, entre otros, comprendiendo cómo estos afectan su comportamiento y apariencia, reforzando su aprendizaje con evaluaciones a través de preguntas teóricas y una actividad de observación práctica que les permitirá reconocer y comprender el uso de cada widget dentro de una interfaz ya construida.

- **Módulo 9: Toca para atacar**

Los estudiantes aprenden a utilizar herramientas que les permitan interactuar con los usuarios, como EditText para recibir datos del usuario y Button para ejecutar acciones, incluyendo la navegación entre pantallas mediante activities usando Intent. A través de un ejercicio práctico, desarrollarán una pantalla de bienvenida donde el usuario escribirá su nombre, el cual será enviado a una segunda pantalla que mostrará un saludo personalizado. Esta actividad les permitirá comprender cómo capturar datos y transferir información entre distintas pantallas de forma sencilla y funcional.

- **Módulo 10: La batalla final**

En esta etapa culminante, los participantes integran todos los conocimientos adquiridos mediante el desarrollo de una aplicación temática basada en la serie de anime Dragon Ball. Se diseña un menú en el que podrán seleccionar una opción, permitir a los usuarios elegir a un maestro de entrenamiento e ingresar el nombre de su personaje. Finalmente,

al presionar un botón, se generará un número aleatorio que representará su “nivel de pelea” acompañado de frases divertidas como “¡Es de más de 8000!” o “¡Ni Mr. Satán tiene ese nivel!”. Esta actividad busca reforzar el uso de widgets, lógica, navegación y diseño atractivo de interfaces.

En base a un análisis detallado de los requerimientos del curso y para garantizar una planificación eficiente y estructurada, se llevó a cabo el desarrollo de descripciones de las características a implementar, para garantizar una planificación eficiente y estructurada de los recursos utilizados en el curso.

En la Tabla 4 se detalla el caso de uso del módulo 3 “Manipulando datos en Kotlin”, describiendo la interacción del estudiante con el módulo y las características presentadas a los participantes.

Tabla 4

Caso de uso: Manipulando datos en Kotlin

Módulo	Estudiante
Funcionalidad	El curso permite analizar el contenido y resolver las pruebas propuestas correspondientes al módulo.
Portada	Visualiza la portada del módulo.
Ingreso al módulo	Accede al contenido del módulo.
Ingreso a infografía	Ingresa a la infografía que contiene la definición de operadores, así como sus tipos.
Ingreso a tabla	Accede a una tabla con los operadores aritméticos, operaciones que realiza y ejemplos.
Ingreso a tabla	Visualiza una tabla con operadores compuestos, la expresión que simplifica y ejemplo de cómo sería la expresión completa.
Ingreso a foro	Participa en un foro donde deberán compartir otros ejemplos de operadores, realizando una comparación entre ellos.
Acceso a prueba	Realiza una actividad donde se les plantea un problema que deberán resolver con operaciones aritméticas.
Visualiza problemática	Visualiza el video con el problema propuesto.
Lee instrucciones	Lee las instrucciones de la prueba presentada.

Tabla 4

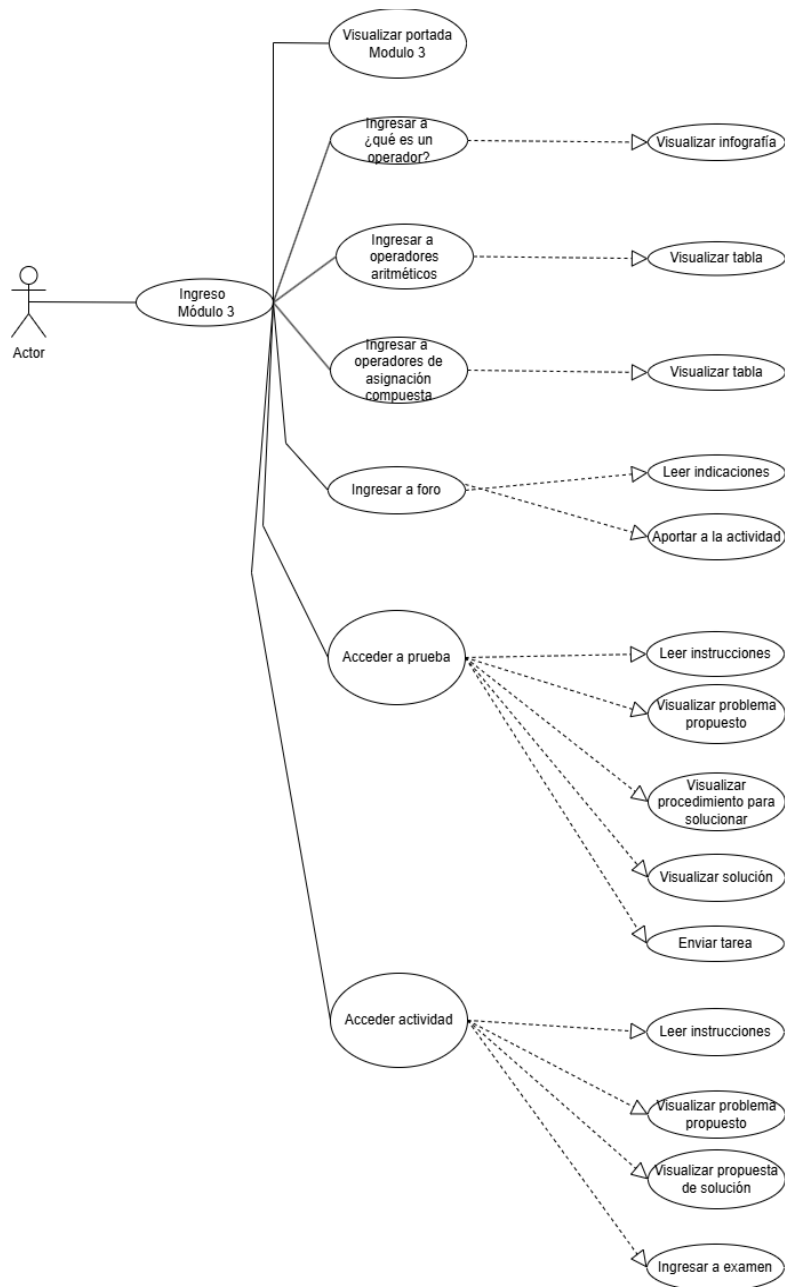
Caso de uso: Manipulando datos en Kotlin (continuación)

Módulo	Estudiante
Visualiza secuencia de solución	Accede a la solución que se debe realizar mediante un diagrama de flujo.
Visualiza propuesta de solución	Estudia un video donde se aprecia una correcta solución en código Kotlin.
Envío de respuestas	Entrega la tarea solicitada.
Acceso a actividad	Accede a la actividad manipulando datos.
Lee instrucciones	Lee instrucciones de la actividad.
Visualiza problema	Analiza el problema propuesto.
Visualiza solución	Estudia la solución del problema.
Solución a examen	Ingresa a un examen que evalúa su aprendizaje a lo largo del curso.

Por otra parte, en la Figura 7 se muestra un diagrama de caso de uso en UML que describe el funcionamiento del módulo 3 manipulando datos en Kotlin.

Figura 7

Diagrama de casos de uso módulo 3

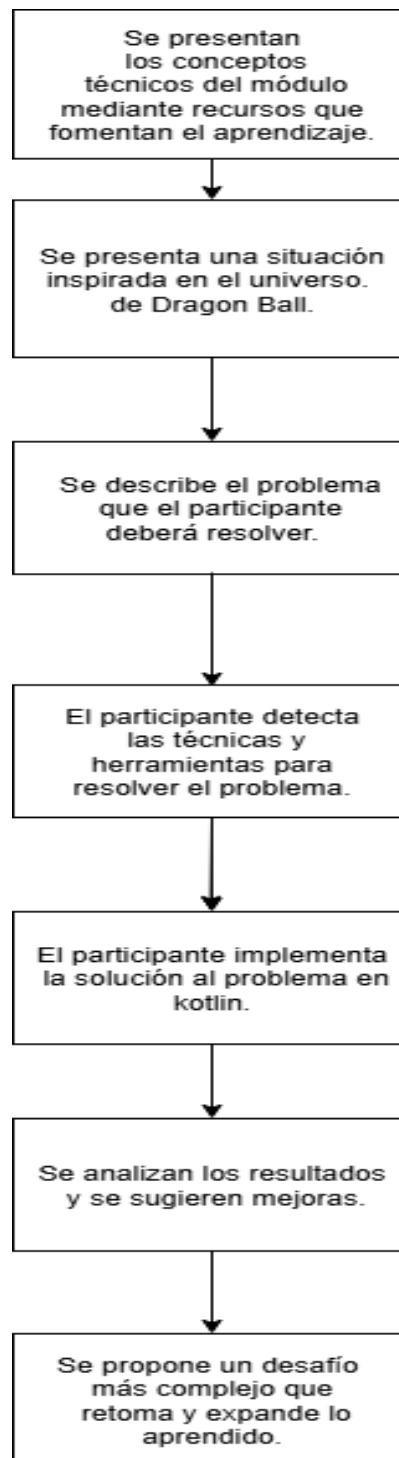


Nota. Elaboración propia.

En la Figura 8 se presenta un esquema que muestra la forma en que se presentan los problemas a los participantes y la estrategia seguida aplicando el ABP en el desarrollo del curso.

Figura 8

Esquema de aplicación de ABP



Nota. Elaboración propia.

5.2 Diseño de la iniciativa

Durante la fase de diseño de la propuesta de solución se llevaron a cabo diversas tareas:

- Selección del sistema de aprendizaje virtual a utilizar.
- Desarrollo de la secuencia didáctica.
- Identificación de materiales didácticos a elaborar.
- Detección de actividades interactivas a implementar.
- Elección de herramientas digitales para el desarrollo de recursos.

5.3 Desarrollo de la secuencia didáctica

La estructura pedagógica abarca un total de 10 módulos planificados de manera lineal que permiten producir las definiciones y dinámicas del curso. Empleando un enfoque progresivo y cíclico que dirige a los participantes en la solución de problemas, creación de código y la validación de los resultados obtenidos.

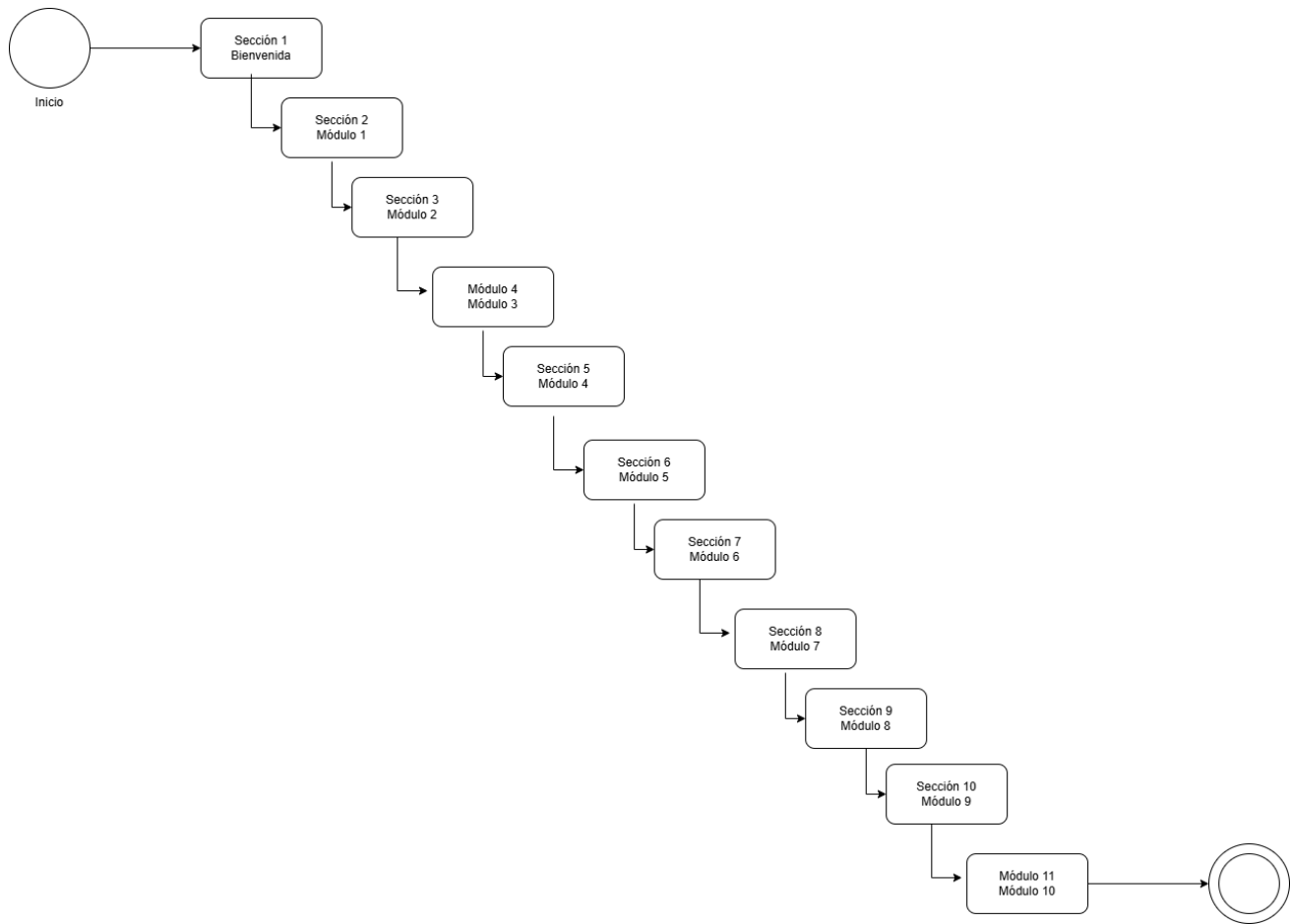
En la Figura 9 se muestra un diagrama de estados de la secuencia didáctica implementada.

La sección inicial, de carácter introductorio, tiene como objetivo brindar a los estudiantes una bienvenida formal al curso, proporcionando información clave para su comienzo. Los módulos presentados posteriormente fueron diseñados en base a los recursos didácticos, de entre los cuales destacan cuestionarios, actividades interactivas y tareas que los participantes deben concretar.

A lo largo del curso se implementó una estrategia de evaluación continua que promueve el aprendizaje activo, integrando en cada módulo elementos que fomentan el aprendizaje basado en problemas. Permitiendo consolidar los conocimientos teóricos, además de motivar la aplicación práctica de los contenidos promoviendo el pensamiento crítico y el desarrollo progresivo de habilidades en programación con Kotlin.

Figura 9

Diagrama de estados de la secuencia didáctica



Nota. Elaboración propia.

La distribución de estos métodos de evaluación se estructuró de acuerdo con la naturaleza y el nivel de complejidad de cada módulo.

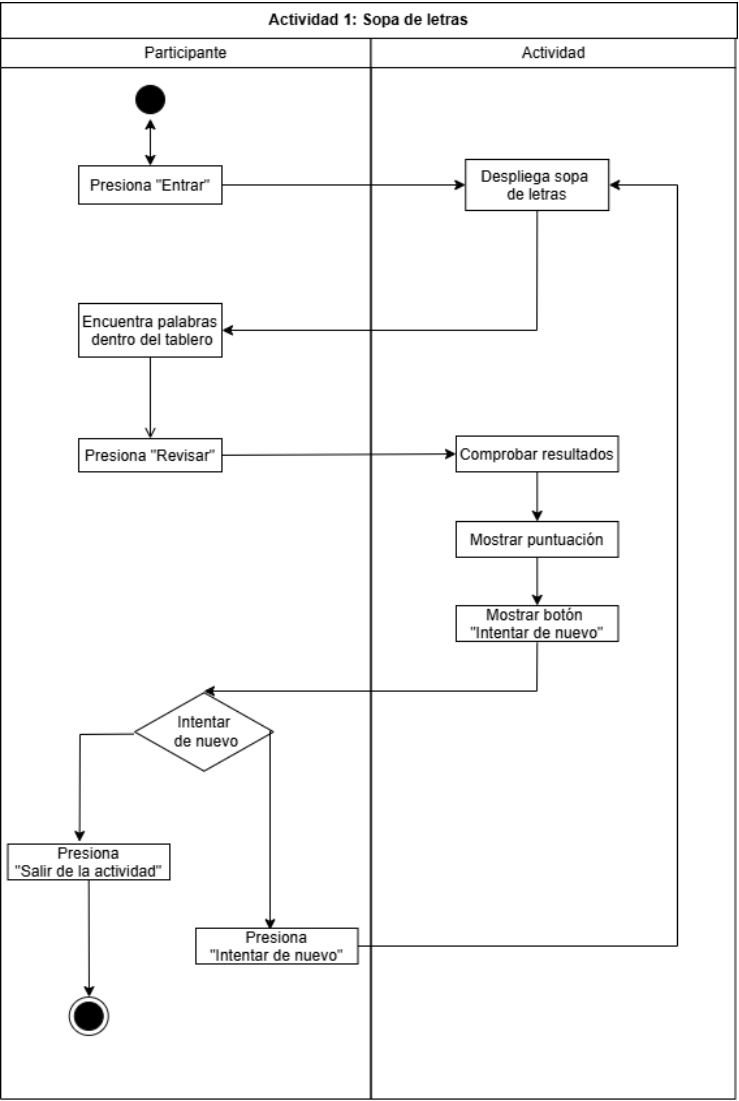
Este enfoque busca que el alumno no solo memorice contenido, sino que desarrolle la capacidad de resolver problemas reales mediante el uso progresivo y contextualizado de Kotlin, involucrándose de forma activa en su propio proceso de formación. Así, la evaluación funge como una herramienta didáctica continua e integrada al diseño del curso.

En la Figura 10 se presenta el diagrama de actividades UML que corresponde a la actividad 1 “Sopa de letras” dentro del módulo 1 “Kotlin y Android Studio”. Se describe la secuencia lógica de acciones, decisiones y condiciones que sigue el estudiante al interactuar con la actividad. Este flujo busca representar de forma visual la estructura del problema planteado, así como la

lógica detrás de su desarrollo, reforzando el enfoque del curso en el aprendizaje práctico y progresivo.

El propósito principal es favorecer la comprensión de los procesos de programación mediante Kotlin dentro de Android Studio. Asimismo, contribuye a mantener el interés del participante y se busca apoyar en la identificación de los elementos fundamentales del entorno y fomentar su autonomía desde las primeras fases del desarrollo de aplicaciones móviles.

Figura 10
Diagrama de actividades: sopa de letras



Nota. Elaboración propia.

5.3.1 Patrones visuales y elección cromática

La selección de colores utilizados dentro del curso corresponde a una estrategia didáctica y visual planificada, cuyo objetivo es promover la comprensión de los temas y mantener una coherencia estética a lo largo de todos los módulos.

Los tonos verdes predominan en los recursos destinados a la transmisión directa del conocimiento, como portadas, presentaciones, videos, imágenes ilustrativas e infografías.

La gama cromática fue elegida por su asociación con el crecimiento, la claridad mental y la armonía, cualidades que favorecen un entorno propicio para el aprendizaje, reforzando visualmente el carácter instructivo del contenido, aportando frescura, orden y energía a los materiales.

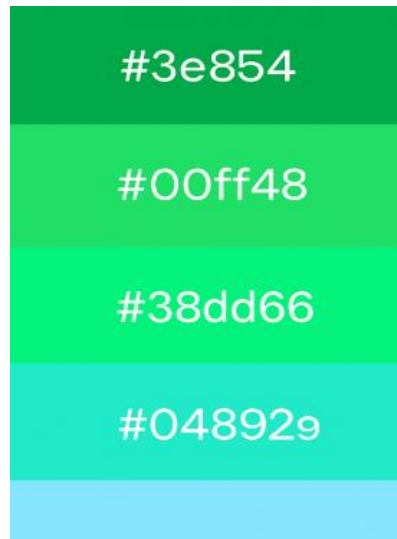
Por otro lado, los colores turquesa y cian se implementaron en los ejercicios prácticos, teniendo como función generar un entorno visual estimulante en ejercicios que los participantes deben analizar y resolver. Estos tonos son utilizados por su semejanza con colores característicos de elementos gráficos de la serie anime de *Dragon Ball*, resultando visualmente atractivos, además de evocar un vínculo emocional con el anime, fortaleciendo así la motivación y el interés en los problemas propuestos.

En la Figura 11 se muestra la paleta de colores que además de cumplir una función estética, así como pedagógica, establece una separación clara entre contenido instructivo y contenido práctico, fortaleciendo así la navegación, comprensión y participación activa.

Para la creación de las portadas mostradas al inicio de cada módulo se utiliza como base los colores y elementos que reflejan el inicio de una nueva aventura, como lo es en el caso de los videojuegos. Este diseño busca reflejar la identidad visual y temática del curso, así como una conexión con la actual era digital, alineándose con los intereses del público objetivo y el enfoque lúdico del aprendizaje.

Figura 11

Paleta de colores



Nota. Elaboración propia.

El patrón de diseño definido para esta ilustración mostrada incluye los siguientes elementos clave:

- Fondo de color verde: Predomina en el fondo y refuerza la idea de progreso, tecnología y creatividad, además de relacionarse con los colores representativos del IDE Android Studio.
- Estilo retro-gamer: Se emplean gráficos pixelados, por ejemplo: íconos de corazones, indicadores de "vidas" y un botón "START" al centro, simulando la interfaz de un videojuego.
- Iconografía Android y Kotlin: La presencia del ícono de Android dentro de un dispositivo móvil, así como el símbolo de toque digital, hace alusión directa a la temática del curso y al entorno de desarrollo.
- Autores del curso: En la parte inferior, se indican claramente los responsables académicos: autor, directora y codirector. Esta sección tiene un diseño alineado al estilo lúdico y refuerza la credibilidad del contenido.

- Elemento digital: La inclusión de un ícono de computadora retro hace referencia al componente técnico del curso, reforzando la relación con la programación y el aprendizaje virtual.

Este patrón de diseño se utilizó en otras portadas o secciones del curso como sello distintivo visual, combinando elementos gráficos, colores temáticos y jerarquía de la información.

En la Figura 12, se muestra la portada del curso.

Figura 12

Diseño de portada



Nota. Elaboración propia.

5.3.2 Backlog del curso

Con el propósito de desarrollar los módulos contemplados dentro del curso, y complementando la metodología de trabajo se desarrolló el backlog fungiendo como herramienta de planificación.

Dentro del backlog de cada módulo se definieron las actividades a realizar para la elaboración de materiales digitales por construir, contemplando los elementos clave para alcanzar los

objetivos didácticos planteados. En la Tabla 5 se muestra la secuencia de actividades, así como las prioridades asignadas para producir los materiales del módulo 3.

Tabla 5

Actividades y prioridades del módulo 3

Material Educativo Digital	Descripción	Prioridad
Incorporar portada	Elaborar una ilustración central para el módulo	1
Recopilar información acerca de operadores aritméticos	Investigar en internet la definición y ejemplos de operadores aritméticos en Kotlin.	2
Diseñar infografía: ¿Qué es un operador?	Con la información recabada, se elabora la infografía con conceptos y definiciones acorde al módulo, procurando el aprendizaje de los participantes.	3
Insertar infografía ¿Qué es un operador?	Posterior al desarrollo de la infografía, es insertada en el módulo.	4
Generar tabla: Operadores aritméticos	En base a los datos obtenidos, se genera una tabla con la descripción y ejemplos de operadores aritméticos utilizados en Kotlin.	5
Insertar tabla: operadores aritméticos.	Se añade al módulo la tabla generada anteriormente	6
Generar tabla: Operadores complejos	Diseñar una tabla con algunos ejemplos de operadores encargados de simplificar la sintaxis de operaciones complejas en Kotlin	7
Añadir tabla: operadores complejos	Se integra al módulo la tabla: operadores complejos	8
Diseñar instrucciones para foro	Se redactan instrucciones para que los alumnos investiguen por su propia cuenta	9
Implementar foro	Se inserta la instrucción del foro y es activado	10
Creación de ejercicio de prueba	En base a lo visto dentro del módulo, se desarrolla un ejercicio práctico que los alumnos deberán analizar y resolver	11

Tabla 5*Actividades y prioridades del módulo 3 (continuación)*

Material Educativo Digital	Descripción	Prioridad
Diseñar instrucciones	Se genera una lista de instrucciones que los participantes deben seguir.	12
Escribir guion para el video: Ejercicio práctico	Redactar un guion que presente a los participantes el problema del ejercicio	1
Grabar audios para video	Acorde al guion, se realiza la grabación de los audios necesarios para el desarrollo del video	14
Optimización del audio	Se mejora la calidad del audio grabado, así como realizar las modificaciones pertinentes.	15
Desarrollar video: ejercicio practico	Se realiza la selección de materiales complementarios para la creación del video, incorporando todos los elementos necesarios	16
Escribir guion para video: diagrama de flujo	Redactar un guion que muestra el análisis del problema presentado	17
Grabar audio para video	Utilizando como base el guion redactado, se comienza con la grabación de audio	18
Optimizar audio para video	Se mejora la calidad de los audios y se realizan los cortes necesarios.	19
Desarrollar video: diagrama de flujo	Se produce el video incorporando los elementos reunidos.	20
Escribir guion para video: solución en Kotlin.	Escribir un guion para mostrar una correcta solución del problema en Kotlin a los participantes	21
Grabar audio para el video	Con base en el guion, se graba el audio.	22
Optimizar audio	Se mejora la calidad de los audios	23
Desarrollar video	Con los materiales generados, se produce el video	24
Evaluar desempeño de los participantes	Se indica a los participantes la entrega de una tarea para evaluar su aprendizaje	25
Crear actividad didáctica	Tomando en cuenta los conocimientos y habilidades adquiridas por los participantes durante el desarrollo del módulo se implementa una actividad donde se pone a prueba el aprendizaje de los participantes	26

Tabla 5*Actividades y prioridades del módulo 3 (continuación)*

Material Educativo Digital	Descripción	Prioridad
Generar instrucciones	Se generan instrucciones que deben seguir los participantes para desarrollar la actividad	27
Redactar guion para video: ejercicio práctico.	Escribir un guion para presentar a los participantes el problema que deberán resolver en este segmento	28
Grabar audio para el video	En base al guion, grabar audio	29
Optimizar audio	Mejorar la calidad del audio y realizar ajustes pertinentes.	30
Desarrollar video: ejercicio práctico	Desarrollar el video, insertando los materiales recabados	31
Redactar guion para video: solución en kotlin.	Escribir un guion para presentar a los participantes el problema que deberán resolver en este segmento.	32
Grabar audio para el video	En base al guion, grabar audio	33
Optimizar audio	Mejorar la calidad del audio y realizar ajustes pertinentes	34
Desarrollar video: solución en Kotlin	Desarrollar el video, insertando los materiales recabados	35
Evaluar aprendizaje de los participantes	Se aplica un examen que evalúa el razonamiento que desempeñan durante la resolución del problema planteado.	36

5.4 Ciclos de elaboración de recursos

Con el propósito de dar estructura al desarrollo de cada módulo y de sus recursos, se llevó a cabo una planificación minuciosa contemplando las secciones y las actividades involucradas.

Fase de planeación

- Definición del alcance: Se determinó el tamaño de cada módulo en el curso, identificando los contenidos específicos que se iban a abordar para cumplir con los objetivos de aprendizaje planteados.

- Calendarización: Se estableció un cronograma general con el objetivo de establecer la duración del desarrollo de cada módulo y la fecha límite para tener listos tanto el material de enseñanza como las actividades de evaluación.

Distribución por módulo

- Distribución de actividades: Se identificaron las actividades necesarias para desarrollar cada módulo y sus componentes, preparando vídeos, ejercicios prácticos, textos guía o pruebas de evaluación
- Cálculo de duración: tomando en cuenta la complejidad de los contenidos, así como las habilidades que los estudiantes irán adquiriendo en el procedimiento de enseñanza. se calculó el tiempo requerido para llevar a cabo cada tarea.
- Distribución de medios disponibles: Se realizó una clasificación de los recursos materiales y tecnológicos requeridos para el desarrollo del proyecto, obteniendo el procedimiento de enseñanza de forma adecuada.

Revisión y aprobación del curso

Una vez finalizada la construcción de los contenidos, se llevó a cabo una revisión final, a cargo del autor, con el objetivo de analizar el procedimiento, asegurando que estuviera correcto, sin errores y listo para implementarse. Los puntos revisados son los siguientes:

- Revisión de contenidos: se analizó el contenido de los módulos del curso, haciendo énfasis en la coherencia de los contenidos, así como la calidad pedagógica de cada unidad, buscando determinar la existencia de errores, inconsistencias o la optimización del contenido.
- Revisión de actividades: se realizó un análisis de las actividades propuestas para los participantes, revisando la claridad de las instrucciones, tanto en las actividades como

en los ejercicios prácticos, además de comprobar que las actividades corresponden con a los objetivos de aprendizaje planteados.

- Revisión de recursos: Se estudió la congruencia de los recursos y el procedimiento de enseñanza elegido.
- Aprobación: Tras realizar una exhaustiva revisión del curso e implementar las optimizaciones pertinentes a cada módulo del curso, cada módulo fue presentado a los directores del proyecto, esperando la validación definitiva previa a su lanzamiento.

5.4.1 Implementación del curso

Para llevar a cabo la elaboración de materiales y recursos pedagógicos digitales, se empleó una variedad de herramientas aptas para obtener los resultados establecidos.

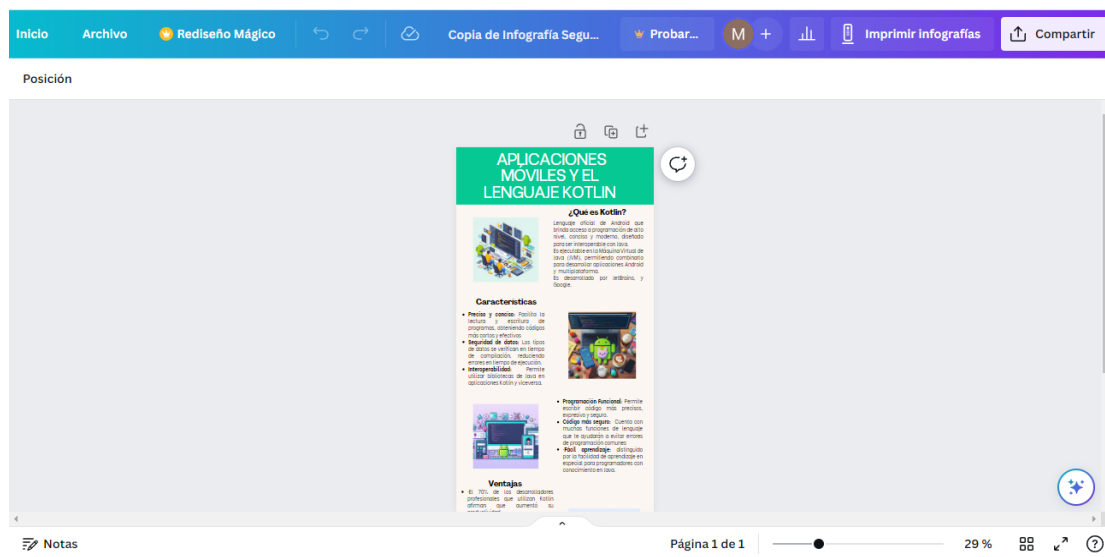
A continuación, se describen las herramientas utilizadas durante la producción de los recursos del proyecto:

5.4.2 Construcción de imágenes

Canva, Herramienta gratuita de diseño gráfico en línea (Canva s.f.) proporciona una gran variedad de funciones que permiten desde crear, editar y personalizar una amplia gama de recursos digitales, permitiendo la interacción con videos, imágenes, música, entre otros recursos multimedia con el objetivo de crear materiales llamativos. Además, cuenta con la posibilidad de editar videos de manera profesional. En la Figura 13 se muestra el ámbito de desarrollo de Canva con el diseño de una infografía.

Figura 13

Entorno de trabajo de Canva



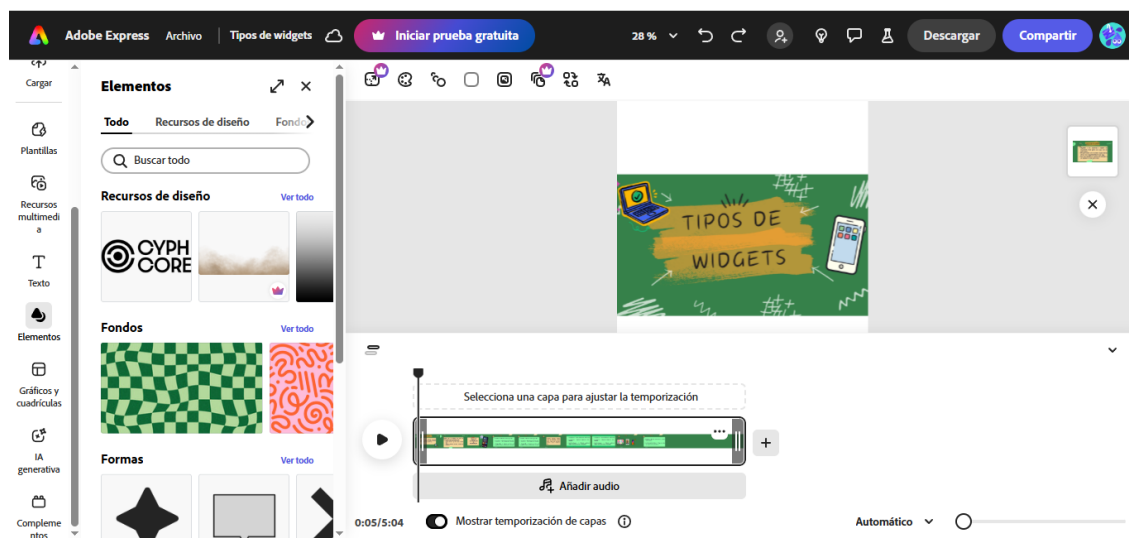
Nota. Elaboración propia.

5.4.3 Creación de videos.

Adobe Premiere Express, herramienta de uso libre para la edición de videos en línea. Permite la producción de videos complejos, proporcionando funciones que incluyen la transición de vídeo, inclusión de textos, animaciones y audios. En la Figura 14 se observa el entorno de desarrollo de Adobe Premiere Express con el diseño de un video.

Figura 14

Entorno de trabajo de Adobe Express



Nota. Elaboración propia.

En la Tabla 6 se presenta el guion que se utilizó para el desarrollo del video para la sección de widgets.

Tabla 6

Guion de video: widgets

Escena	Descripción	Duración (segundos)
1	Pantalla de inicio con el título: Tipos de widgets	10
2	Saludo y bienvenida a los participantes	10
3	Instructor: En esta nueva fase de entrenamiento, si has llegado hasta aquí ya conoces lo básico de Android Studio y estás listo para dar un gran salto, crear tu primera interfaz gráfica	10
4	Muestra pantalla de texto	12
	Para empezar, definiremos que es un widget. Es cualquier elemento visual con el que los usuarios pueden interactuar en una aplicación. Imagina que los widgets son como bloques de construcción de una aplicación.	12
5	Muestra pantalla de texto	15
	Algunos nos muestran información, otros nos permiten al usuario escribir algo, y algunos otros realizan acciones cuando los presionas. Es decir, sin widgets, una aplicación no tendría botones, texto o imágenes. Sería un espacio vacío	15
6	Muestra pantalla de texto	17
	Algunos de los beneficios de los widgets son ayudar a los usuarios a lograr un recorrido clave para una aplicación con un solo toque. Proporcionar un resumen rápido de los datos más importantes de una aplicación. De igual manera, son una excelente manera de atraer a un usuario a una aplicación	17
7	Muestra pantalla de Título	12
	Existe una gran variedad de widgets para programar en Android Studio, pero debido a su complejidad, sólo abordaremos algunos de los más sobresalientes. Primero tenemos a los widgets de texto y visualización de información	12

Tabla 6*Guion de video: widgets (continuación)*

Escena	Descripción	Duración (segundos)
8	Muestra pantalla de texto	6
	Estos widgets nos permiten mostrar información al usuario como títulos, descripciones o resultados	6
9	Muestra pantalla de inicio	42
	El primero es TextView. Este es utilizado para mostrar el texto al usuario. Puede ser un título, una descripción, una frase o el resultado de una operación. El texto que ves en muchas aplicaciones, pudiendo ser como Bienvenido, Iniciar sesión, Escribe tu nombre	42
	Normalmente se muestra con un TextView. ImageView. Este widget sirve para mostrar imágenes en la pantalla. Puedes utilizarlo para poner un logo, un personaje, una foto o cualquier imagen que quieras mostrar dentro de tu aplicación. Tenemos también ProgressBar. Este widget muestra una barra que indica el progreso o carga. Es útil cuando tu aplicación está haciendo algo que tarda un poco, como descargar o procesar datos. Así, el usuario sabe que debe esperar un momento.	
10	A continuación, te muestro algunos ejemplos que permiten identificar los widgets en una pantalla	6
	Muestra pantalla con ejemplos	5
11	Muestra pantalla de título	3
	Ahora hablaremos de los Widgets Contenedores	3
12	Muestra pantalla de texto	16
	Estos nos permiten organizar los elementos dentro de la pantalla. Estos widgets sirven para organizar y acomodar otros widgets dentro de la pantalla, como si fueran una caja que contiene botones, texto, imágenes, etc.	20
13	Muestra pantalla de texto	50

Tabla 6*Guion de video: widgets (continuación)*

Escena	Descripción	Duración (segundos)
	Tenemos <code>FrameLayout</code>, este es un tipo de layout que pone los widgets uno encima del otro, como si los apilaras en una sola pila. El primer widget que cargas se queda al fondo y los siguientes se quedan encima. Es útil cuando solo quieres mostrar un widget a la vez o cuando necesitas que uno cubra al otro. Por ejemplo, una imagen de fondo con un botón encima. <code>LinearLayout</code>. Este layout organiza los widgets en línea, ya sea de arriba hacia abajo o de izquierda a derecha, dependiendo de la orientación que le pongas. Es muy fácil de utilizar cuando quieres que los elementos estén bien ordenados uno tras otro. <code>ConstraintLayout</code>. Esto nos da más libertad para acomodar los widgets, porque puedes acomodarlos donde tú quieras que se coloque, poniendo reglas o restricciones, hablando de constraints. Pudieras hacer que un botón esté centrado, pegado a otro, o que siempre se mantenga a cierta distancia del board Es ideal para que tus diseños sean más complejos.	52
14	De igual manera, a continuación, te muestro algunos ejemplos de cómo identificar estos widgets	8
	Muestra pantalla con ejemplos	8
15	Muestra pantalla de título	6
	Continuamos con widgets de entrada de datos	3
	Estos permiten al usuario escribir información directamente en la aplicación, como su nombre, correo, ciudad o cualquier otro dato, siendo esenciales para recolectar información del usuario	
	<code>EditText</code> es una caja de texto vacía, donde el usuario puede escribir libremente. <code>AutocompleteTextView</code>. Esta es una caja de texto que muestra sugerencias mientras el usuario escribe, ayudando a completar lo que se está escribiendo más rápidamente. De igual manera te muestro algunos ejemplos.	
	Muestra pantalla de título	3
	Continuamos con widgets de botones e interacción	3

Tabla 6*Guion de video: widgets (continuación)*

Escena	Descripción	Duración (segundos)
	Son elementos que el usuario presiona para realizar una acción	4
	Tenemos, por ejemplo, Button, que simplemente es un botón que al presionarlo realiza una acción. ImageButton. Este de igual manera es un botón, con la diferencia de que es una imagen en lugar de texto. ToggleButton. Este es un botón de encendido o apagado, dependiendo de la posición. Y tenemos también Switch. Es muy parecido a ToggleButton, pero con un diseño más moderno	17
	De igual manera aquí puedes apreciar algunos ejemplos.	5
	Muestra pantalla con ejemplos	5
	Finalmente estudiaremos los widgets de selección y navegación	4
	Muestra pantalla de título	4
	Permiten al usuario elegir opciones o moverse entre pantallas	3
	Muestra pantalla de texto	3
	Muestra pantalla de texto	20
	Por ejemplo, están Spinner, un menú desplegable con opciones. CheckBox. son casillas de verificación para seleccionar múltiples opciones. RadioButton y RadioGroup. Estos son botones de opción para seleccionar solo una opción dentro de un grupo. Y también tenemos Slider. Este es una barra deslizante que permite seleccionar un valor. De igual manera aquí puedes apreciar algunos ejemplos	20
	Muestra pantalla de ejemplos	
	Así que hemos llegado con el video de hoy. Te invito a seguir aprendiendo. Y recuerda, el entrenamiento continúa	7
	Despedida	

En la Figura 15 se muestran los recursos empleados para la producción del video de bienvenida del curso.

Figura 15

Diseño de video de bienvenida



Nota. Elaboración propia.

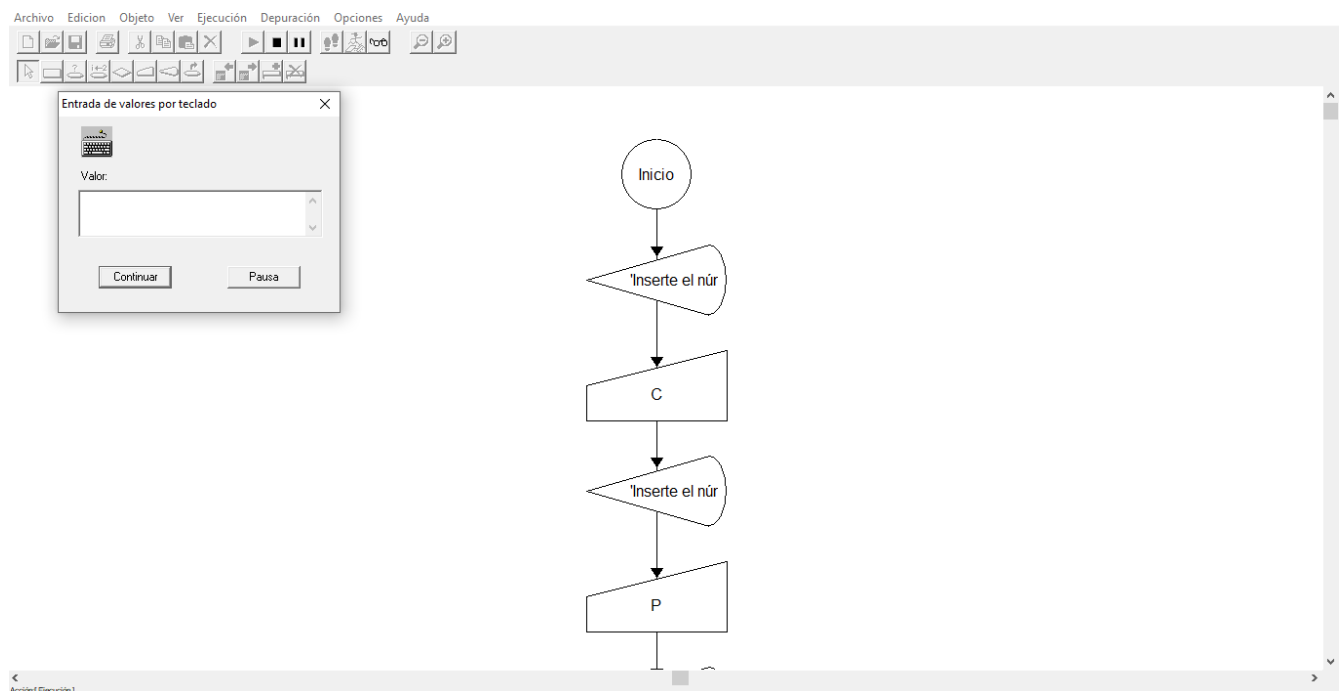
5.4.4 Creación de elementos lúdicos

En esta sección se describen las herramientas utilizadas y los elementos lúdicos construidos para ser incorporados al curso de este proyecto.

DFD. Es una herramienta utilizada en el entorno educativo, permite diseñar, ejecutar y verificar algoritmos de manera gráfica mediante diagramas de flujo, previo a la ejecución de código. En la figura 16 se presenta el ámbito de desarrollo de la aplicación en la creación de un diagrama de flujo en DFD.

Figura 16

Entorno de trabajo de DFD



Nota. Elaboración propia.

Lumi es una aplicación que permite la creación, organización y finalmente compartir contenidos didácticos e interactivos de manera simple y práctica. Facilita el desarrollo de una gran variedad de recursos multimedia.

Utiliza estándares como H5P, lo que la vuelve una plataforma intuitiva en la que desarrolladores de contenido pueden construir actividades, ejercicios, presentaciones, vídeos

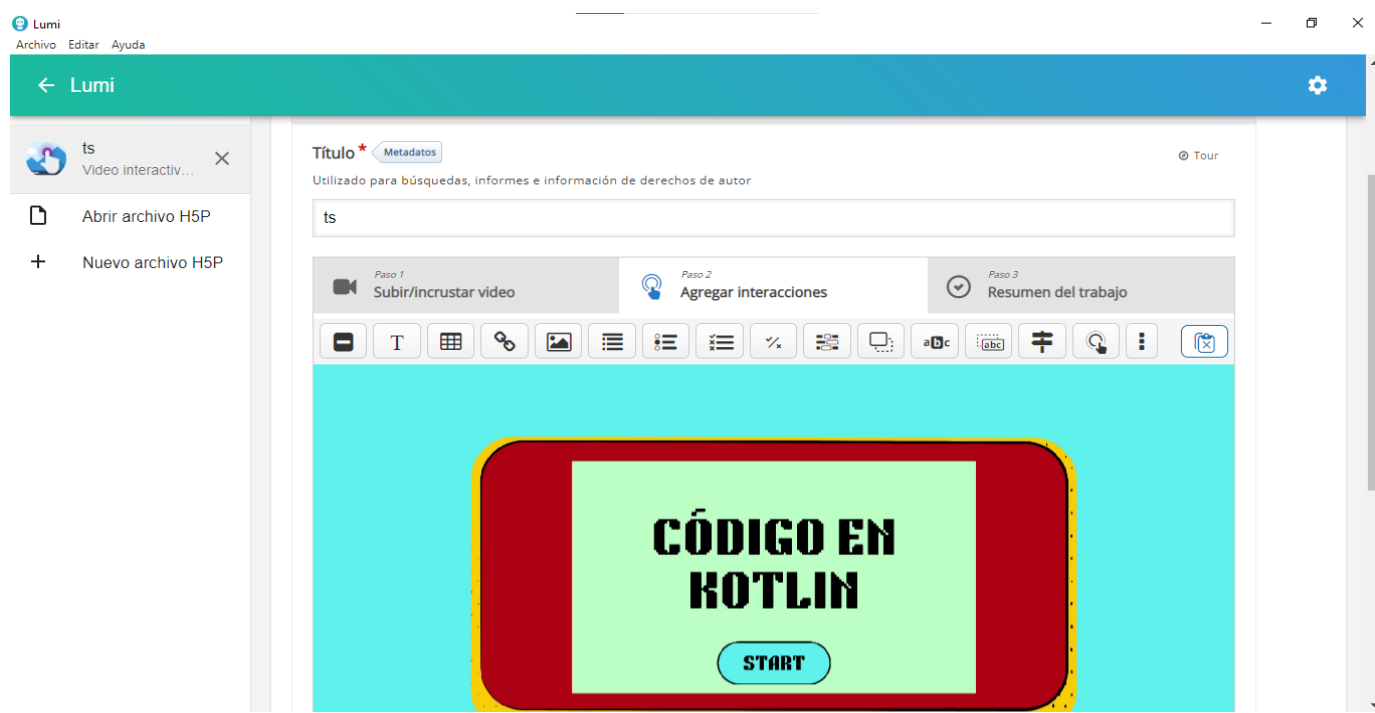
interactivos, pruebas o juegos pedagógicos, sin necesidad de tener conocimientos avanzados de programación.

Finalmente, permite visualizar, previsualizar y exportar el material en diversos formatos (como HTML), proporcionándole una gran portabilidad, debido a que permite integrar el resultado en diferentes entornos de aprendizaje, como Sistemas de Gestión de Aprendizaje o páginas web, así como usarse sin conexión a internet.

En la Figura 17 se presenta el ámbito de desarrollo de Lumi durante la elaboración de materiales didácticos.

Figura 17

Entorno de trabajo de Lumi



Nota. Elaboración propia.

Del conjunto de materiales dinámicos que proporciona Lumi, se integraron al curso los siguientes tipos:

- Video interactivo (Interactive Video): Permite agregar interacciones a un video agregando explicaciones, imágenes adicionales, tablas, complete el espacio en blanco y preguntas de opción múltiple.

- Crucigrama (Cross Word): Facilita la creación y personalización de un crucigrama atractivo para la audiencia. Permite la modificación de los colores, cargar una imagen de fondo, otorgar la puntuación además de incrustar las palabras de manera aleatoria para que la audiencia obtenga un nuevo crucigrama cada vez que lo desee.
- Encuentra las palabras (Find the words): Permite crear una lista de palabras que se integran en una cuadrícula. Posteriormente los participantes deben encontrar y seleccionar las palabras en la cuadrícula.
- Arrastrar y soltar (Drag and Drop): permite a los participantes relacionar dos o más elementos y establecer conexiones lógicas de forma visual.

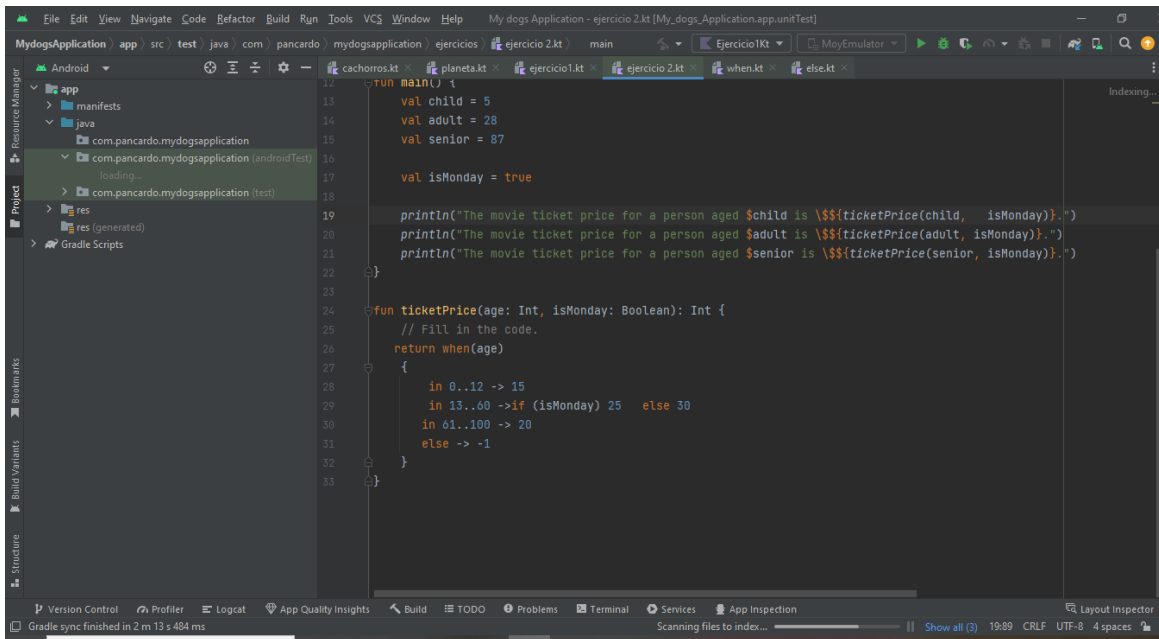
Android Studio es el IDE oficial de Android, dirigido a desarrolladores y a estudiantes de programación.

Proporciona herramientas avanzadas que permiten diseñar, implementar, depurar y ejecutar aplicaciones en Android, a través de la edición de interfaces gráficas y análisis de rendimiento, resultando así el desarrollo de apps de forma más rápida y eficiente.

En la Figura 18 se observa el entorno de trabajo de Android Studio durante el desarrollo de contenido.

Figura 18

Entorno de trabajo de Android Studio



Nota. Elaboración propia.

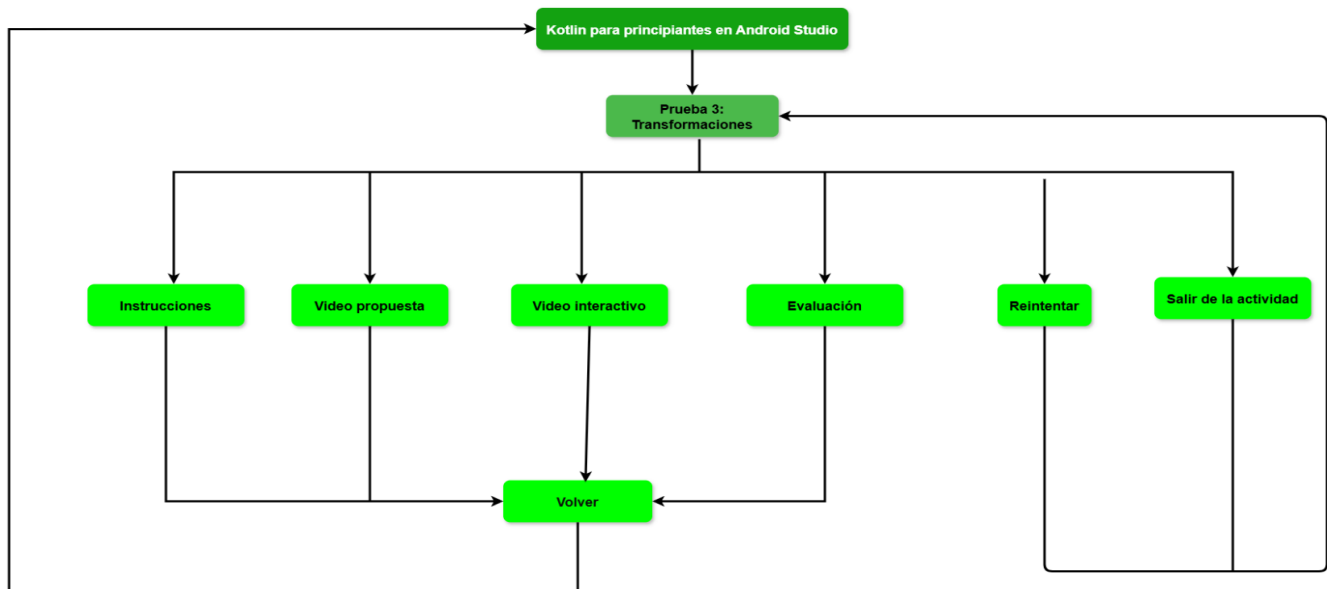
5.5 Implementación del curso

Tras haber concluido con el desarrollo de los elementos de enseñanza, se llevó a cabo la incorporación y ajustes de los módulos del curso dentro del entorno del SGA Moodle.

Con el objetivo de integrar los módulos en Moodle se optó por elaborar mapas de navegación que permitieron organizar los componentes y la relación con los participantes. En la Figura 19 se muestra un mapa de navegación que detalla la intervención de los recursos de la prueba 3 “Transformaciones”.

Figura 19

Mapa de Navegación: Prueba 3 "Transformaciones"



Nota. Elaboración propia.

5.6 Configuración local de desarrollo

En el ámbito de proyectos informáticos, es de suma utilidad contar con un entorno local de trabajo que permita diseñar, implementar, probar y ajustar recursos de manera que no afecte los entornos de producción.

Este tipo de prácticas proporcionan un espacio controlado para la depuración, validación funcional y corrección de errores previo a su despliegue final. Además, el uso de ramas en sistemas de control de versiones como Git contribuye a una gestión ordenada del proyecto, permitiendo implementar mejoras de forma paralela, revisar cambios y mantener la estabilidad del producto principal.

Siguiendo esta línea de trabajo, se configuró un entorno de trabajo local que permitió desarrollar los módulos de manera progresiva y controlada, dicha estrategia facilitó integrar recursos educativos, así como validar los componentes didácticos y la mejora continua del contenido, garantizando la calidad del curso antes de su implementación final en línea.

Se optó por utilizar Laragon, una herramienta de desarrollo local que destaca por facilitar su instalación, ligereza y compatibilidad con múltiples tecnologías, ofrece una experiencia más robusta y profesional al integrar servicios como Apache, MariaDB, PHP, Node.js, y Git en un solo paquete auto contenible.

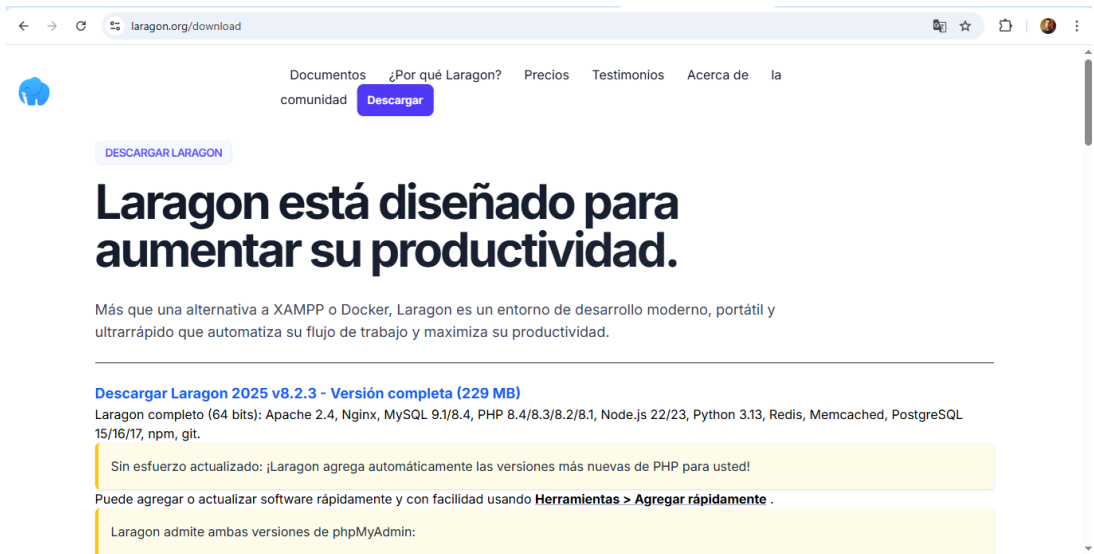
Una de sus principales ventajas es su portabilidad, ya que puede ejecutarse sin instalación previa, incluso desde una unidad externa. Además, su interfaz amigable y su terminal avanzada facilitan la administración del entorno, permitiendo al desarrollador centrarse en la construcción de la solución sin perder tiempo en configuraciones técnicas complejas.

En el contexto de este proyecto, Laragon resultó especialmente útil para ejecutar de manera local la plataforma Moodle, ya que proporciona los servicios necesarios para su correcto funcionamiento sin necesidad de servidores remotos. De esta manera, se garantiza un entorno de trabajo seguro, eficiente y funcional para el desarrollo y evaluación de los módulos del curso.

Para configurar el entorno local de trabajo, se siguieron los siguientes pasos:

1. Descargar el instalador: Se accede al sitio web oficial de Laragon (<https://laragon.org/>) y acceder al apartado de descargas, posteriormente se descarga la versión más reciente de la aplicación. Como se muestra en la Figura 20.

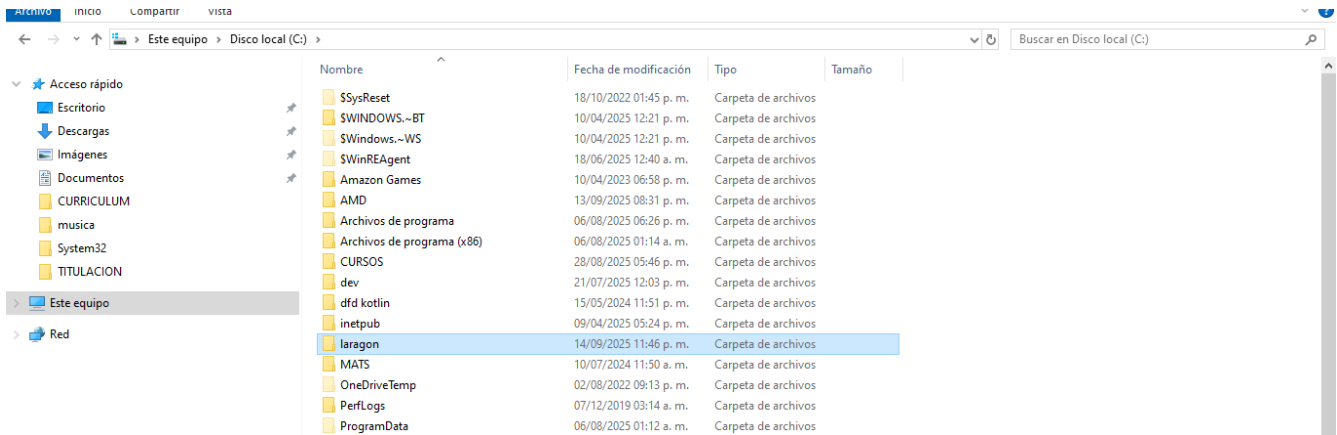
Figura 20
Descarga de Laragon



Nota. Elaboración propia.

2. Instalar la aplicación: Una vez finalizada la descarga se debe ejecutar el instalador, se aceptan términos y condiciones, a continuación, se asigna la ruta de instalación que deberá ser: C:\laragon como se muestra en la Figura 21. Finalmente se eligen los componentes que se desea agregar, en este caso se continuó con la selección por defecto.

Figura 21
Directorio de Laragon

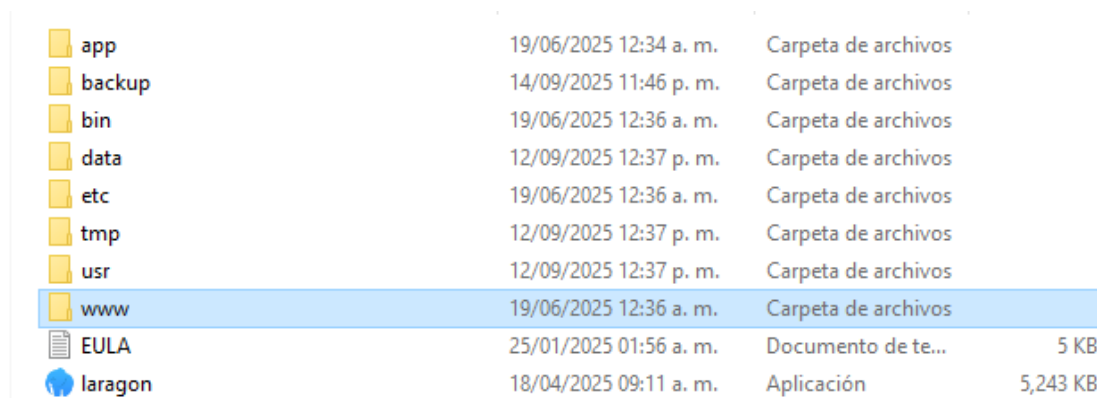


Nota. Elaboración propia.

3. Preparar Moodle: Acceder al apartado de descargas del sitio oficial de Moodle (<https://download.moodle.org/>). El archivo descargado debe colocarse dentro de la carpeta denominada www, como se muestra en la Figura 22.

Figura 22

Ubicación de directorio laragon/www



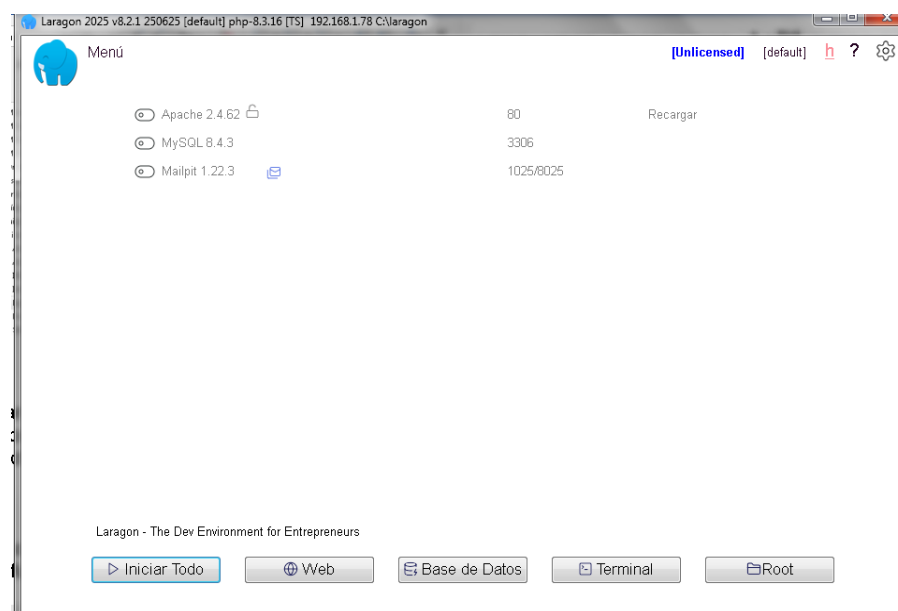
app	19/06/2025 12:34 a. m.	Carpeta de archivos	
backup	14/09/2025 11:46 p. m.	Carpeta de archivos	
bin	19/06/2025 12:36 a. m.	Carpeta de archivos	
data	12/09/2025 12:37 p. m.	Carpeta de archivos	
etc	19/06/2025 12:36 a. m.	Carpeta de archivos	
tmp	12/09/2025 12:37 p. m.	Carpeta de archivos	
usr	12/09/2025 12:37 p. m.	Carpeta de archivos	
www	19/06/2025 12:36 a. m.	Carpeta de archivos	
EULA	25/01/2025 01:56 a. m.	Documento de te...	5 KB
laragon	18/04/2025 09:11 a. m.	Aplicación	5,243 KB

Nota. Elaboración propia.

4. Iniciar servicios de Laragon: Se accede al panel, hacer clic en "Iniciar todo" para arrancar todos los servicios (Apache, MySQL, etc.), para comprobar el correcto funcionamiento. En la Figura 23 se muestra el panel de control.

Figura 23

Panel de control de Laragon

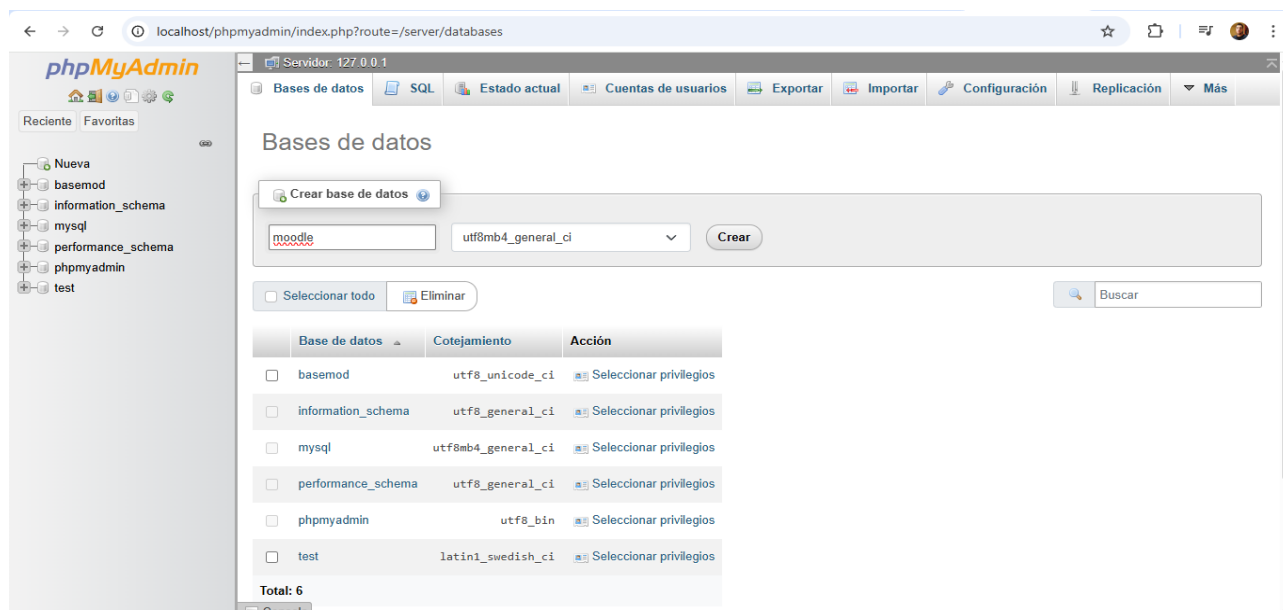


Nota. Elaboración propia.

5. Crear base de datos: Después de iniciar los servicios de Laragon correctamente, se deberá abrir un navegador e ingresar al enlace <http://localhost/phpmyadmin> donde realiza la creación de una base de datos vacía con el nombre “Moodle”. Como se presenta en la Figura 24.

Figura 24

Entorno de trabajo de phpMyAdmin



Nota. Elaboración propia.

6. Configurar Moodle: finalmente se debe acceder al enlace <http://localhost/moodle> y se siguen los pasos para llevar a cabo la instalación, configurando los ajustes pertinentes. La página derivada debería ser igual a la que se muestra en la Figura 25.

Figura 25

Instalación de Moodle

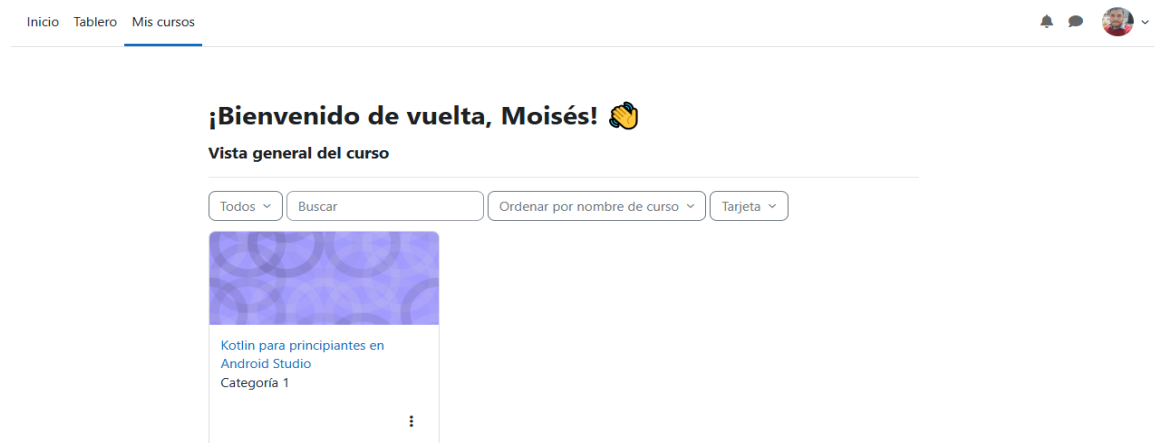


Nota. Elaboración propia.

7. Interfaz de Moodle: Al concluir con éxito el proceso de instalación, Moodle nos lleva directamente a su página de inicio donde podremos comenzar a diseñar y gestionar nuestro curso. En la Figura 26 se muestra el tablero de inicio.

Figura 26

Pantalla de inicio de Moodle



Nota. Elaboración propia.

5.7 Evaluación del curso

Consecuente a la implementación de la metodología ADDIE durante el desarrollo del curso, es pertinente realizar una evaluación al curso diseñado. Tomando como referencia el instrumento

de seguimiento y evaluación de cursos virtuales propuesto por Hugo Trejo González (2022), publicado en la revista EDUTEC, que tiene como objetivo valorar cursos implementados en entornos virtuales de aprendizaje.

Este instrumento resultó pertinente para el presente trabajo por su enfoque integral en la valoración de entornos educativos virtuales, ya que considera seis criterios principales: diseño instruccional, objetivos de aprendizaje y enfoque pedagógico, arquitectura de la instrucción, diseño multimedia y usabilidad, interacciones y comportamiento (netiquetas). Estos criterios fueron adaptados al contexto del curso diseñado en Moodle, de acuerdo con el modelo ADDIE empleado en este proyecto.

Esta herramienta contempla cinco dimensiones fundamentales: diseño instruccional, enfoque pedagógico, arquitectura de la instrucción, diseño web y netiqueta, mismas que en conjunto permitieron realizar la evaluación del curso. Ofreciendo una visión integral de la calidad educativa y técnica de los cursos en línea, a continuación, se describen los aspectos considerados

- Diseño instruccional: Analiza la coherencia entre los objetivos de aprendizaje, los contenidos de cada módulo y las actividades propuestas.
- Enfoque pedagógico: Evalúa la pertinencia del uso de las metodologías considerando su capacidad para motivar al estudiante, fomentar el pensamiento crítico y establecer conexiones entre teoría y práctica, principalmente a través de actividades interactivas y contextos aplicados relacionados con la programación.
- Arquitectura de la instrucción: Contempla la organización de los contenidos dentro del Sistema de Gestión del Aprendizaje, cuidando la secuencia lógica, la progresión por niveles de dificultad y la accesibilidad del material didáctico.
- Usabilidad y accesibilidad: Valora la presentación gráfica del curso, incluyendo paleta de colores, estructura visual, navegación intuitiva y legibilidad. Priorizando la coherencia visual y la usabilidad como factores clave para mantener el interés del estudiante.

- Interacciones y comportamiento: Estudia la interacción adecuada entre los participantes, incluyendo reglas de participación en foros y canales de comunicación asincrónica.

La aplicación del instrumento fue realizada por el autor del curso, en su calidad de responsable de la propuesta, con base en la evidencia obtenida durante la implementación. Para ello se llevó un registro de estudiantes, quienes resolvieron tres pruebas clave del curso las cuales son: evaluación diagnóstica, evaluación intermedia y evaluación final permitiendo analizar la calidad de los contenidos, el grado de motivación alcanzado y la efectividad de las actividades propuestas. Las capturas de pantalla de los formularios se presentan en los Anexos A, B y C, respectivamente.

En la Tabla 7 se presenta la síntesis de los resultados obtenidos tras la aplicación de la herramienta para evaluar el EVA implementado. Puntuación otorgada por el responsable del curso.

Tabla 7

Parámetros de evaluación EVA

Criterios	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Diseño instruccional					
1. El modelo instruccional propuesto es factible y se adapta a la institución en cuanto al enfoque pedagógico e infraestructura.				X	

Tabla 7*Parámetros de evaluación EVA (continuación)*

Criterios	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
2. Se respetan las etapas de diseño instruccional o se adaptan con el objetivo de dar congruencia a la propuesta pedagógica.					X
Enfoque pedagógico					
3. Los contenidos corresponden con el enfoque pedagógico elegido y con las necesidades planteadas en las fases del modelo instruccional.					X
Arquitectura de la instrucción					
4. El diseño de la instrucción es acorde con el paradigma de aprendizaje a nivel didáctico.				X	

Tabla 7*Parámetros de evaluación EVA (continuación)*

Criterios	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Usabilidad y accesibilidad					
5. La plataforma y los materiales facilitan el acceso al	5. La plataforma y los materiales facilitan el acceso al	5. La plataforma y los materiales facilitan el acceso al	5. La plataforma y los materiales facilitan el acceso al	5. La plataforma y los materiales facilitan el acceso al	5. La plataforma y los materiales facilitan el acceso al
5.1 Aprendizaje					X
5.2 Operatividad					X
5.3 Satisfacción					X
5.4 Contenido				X	
5.5 Eficiencia				X	
5.6 Eficacia					X
Interacciones y comportamiento					
6. Los docentes y estudiantes demuestran buenas prácticas en el entorno virtual: uso adecuado de la plataforma, respeto a las interacciones y a los horarios de trabajo.					X

A partir de dichos criterios, se evaluó la coherencia pedagógica, la estructura del diseño y la funcionalidad del entorno virtual, considerando tanto la experiencia del docente como los resultados obtenidos por los participantes durante el desarrollo del curso.

Tras el análisis de los resultados obtenidos es posible destacar los siguientes puntos:

- Los contenidos del curso están directamente vinculados a los objetivos planteados presentando un alto nivel de coherencia y calidad en su contenido, contribuyendo de manera efectiva al desarrollo de las competencias definidas.
- Las tareas promueven la participación activa, la autoevaluación y la reflexión, integrando un enfoque motivador mediante ejemplos y problemáticas cercanas a los intereses de los estudiantes.
- La estructura del curso permite avanzar de manera ordenada desde conceptos básicos hacia retos más complejos, facilitando el aprendizaje progresivo.
- Se incluyen videos explicativos y materiales de referencia que fortalecen la comprensión y la práctica autónoma.

El parámetro con menor valoración relativa fue la arquitectura de la instrucción, calificado como “De acuerdo”, lo que indica oportunidades de mejora en la organización de algunos contenidos y en la optimización de recursos multimedia.

La evaluación del curso “kotlin para principiantes en Android Studio” no solo sirvió como diagnóstico de calidad, sino también como un antecedente para futuras mejoras, permitiendo identificar fortalezas y áreas susceptibles de mejora del curso, garantizando con ello una experiencia de aprendizaje más efectiva, inclusiva y orientada al desarrollo de competencias digitales.

Es importante recordar que no se pretende limitar la evaluación a los 6 criterios de la Tabla y se podrían integrar la descripción de fases del modelo elegido. Asimismo, puede ser adaptado para entornos presenciales, en donde las interacciones y el espacio físico juegan un papel esencial en la enseñanza-aprendizaje (Trejo, 2022).

5.7.1 Evaluación del aprendizaje de los participantes

Para complementar el diagnóstico del curso como entorno virtual de aprendizaje, surgió la necesidad de evaluar el aprendizaje que logran alcanzar los participantes inscritos en el curso *“Kotlin para principiantes en Android Studio”*. Determinando la creación e implementación de tres instrumentos de evaluación aplicados en diferentes momentos clave del proceso formativo, con el objetivo de medir los conocimientos previos y finales, así como obtener una retroalimentación sobre la experiencia de aprendizaje. A continuación, se describen los instrumentos utilizados durante dicha evaluación de conocimiento.

- Formulario de inscripción y evaluación diagnóstica general: Previo a la inscripción en Moodle, los participantes recibieron y completaron un formulario en Google Forms, que cumplió una doble función: filtrar a los interesados genuinos en participar en el curso, además de evaluar sus conocimientos generales dentro del área de la programación. Permitiendo conocer el nivel de partida de los estudiantes y establecer una línea base para el contenido del curso, así como sus herramientas de evaluación. En la Figura 27 se muestra la pantalla de acceso al formulario de inscripción al curso *“kotlin para principiantes en Android Studio”*. Para consultar de manera detallada el resultado del formulario consultar el anexo A.

Figura 27

Formulario de registro alumnos

← → ↻ docs.google.com/forms/d/e/1FAIpQLSfznVafUJE-dvlf-2hL-ZkNH3b1qIVG38ipA7hqajXvAX3N9Q/viewform ☆ ⌵ ⓘ ⋮

Inscripción y evaluación diagnóstica - Curso: kotlin para principiantes en Android Studio

Este formulario tiene como objetivo inscribirte oficialmente al curso "Kotlin para principiantes" y conocer tu nivel inicial sobre programación móvil con Android Studio. Por favor, responde las preguntas con sinceridad y sin ayuda externa. Una vez inscritos al curso, les pido de la manera más atenta, que concluyan el curso en la fecha límite del jueves 31 de Julio

cr419703@uaeh.edu.mx [Cambiar de cuenta](#)

No compartido

[Siguiente](#) [Borrar formulario](#)

Nunca envíes contraseñas a través de Formularios de Google.
Este formulario se creó en uaeh. - [Propietario del formulario de contacto](#)

Google Formularios

Nota. Elaboración propia.

- **Formulario de diagnóstico:** Una vez que los participantes están inscritos en el curso y acceden a la plataforma Moodle, deben responder un formulario cuyo objetivo es analizar el conocimiento previo de los participantes en programación móvil, con énfasis en conceptos básicos de Android Studio y Kotlin, logrando así determinar el grado de familiaridad que los participantes tenían con la temática central del curso. En la Figura 28 se muestra la pantalla de acceso al formulario “conocimientos previos”. Para consultar a detalle la información recabada por el formulario, revisar Anexo B.

Figura 28

Evaluación diagnóstica

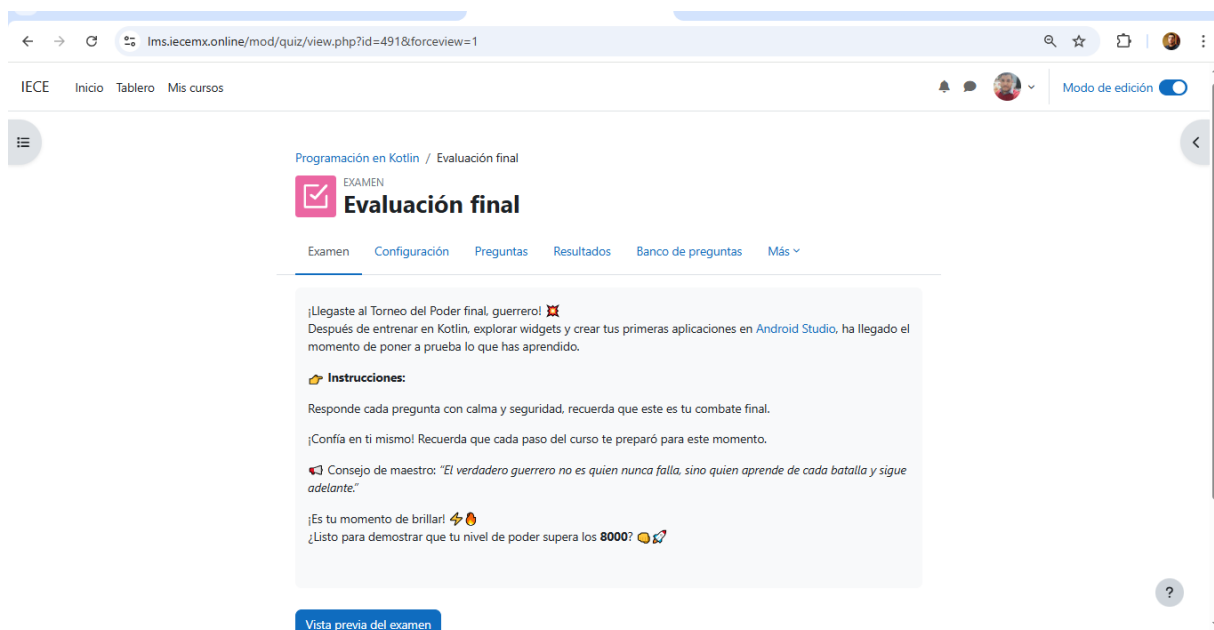


Nota. Elaboración propia.

- **Evaluación final de aprendizaje:** En el módulo final del curso, los participantes deben realizar una evaluación que tiene como objetivo de medir el conocimiento y habilidades adquiridos a lo largo del curso. Esta prueba permite comparar los resultados obtenidos con los diagnósticos iniciales, evidenciando el progreso de los estudiantes. En la Figura 29 se muestra la pantalla de acceso a la evaluación final. Para consultar a detalle la información reunida por el formulario, consultar el Anexo C.

Figura 29

Evaluación final



Nota. Elaboración propia.

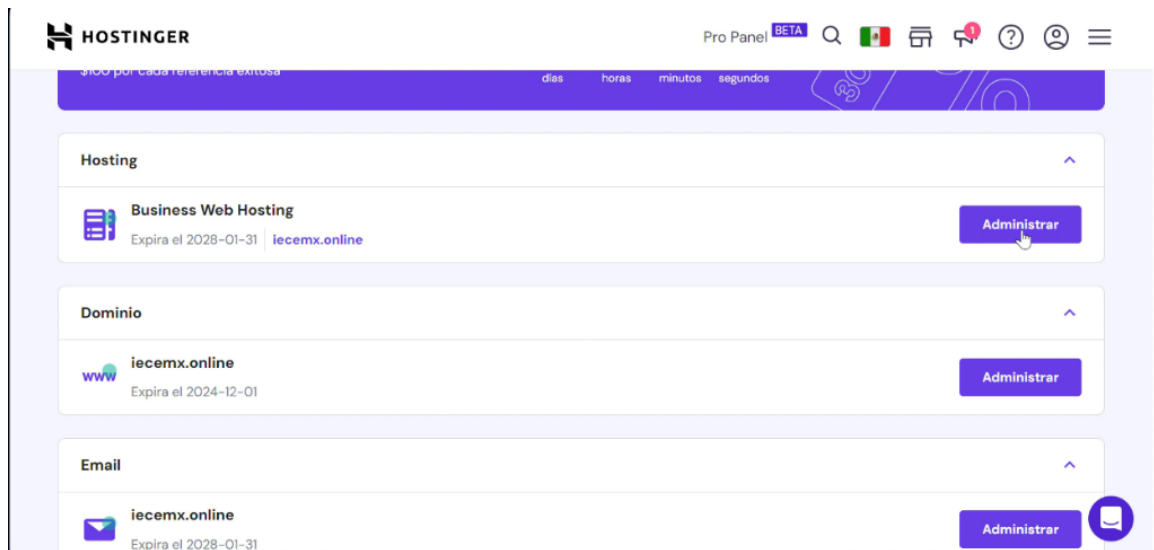
5.8 Publicación del curso

Con la finalidad de impartir el curso en modalidad virtual y permitir que los participantes accedan desde cualquier dispositivo con conexión a internet, se tomó la decisión de adquirir un dominio web a través del proveedor Hostinger. Permitiendo disponer de un espacio en línea propio, estable y de fácil acceso para alojar los contenidos y recursos educativos desarrollados. Una vez realizada la compra del dominio, fue necesario realizar la instalación de Moodle. A continuación, se describe el procedimiento seguido para la instalación:

1. Entrada al panel principal de configuración: se inicia sesión en la cuenta registrada como administrador a la sección de panel de control, como se muestra en la Figura 30

Figura 30

Panel principal de Hostinger

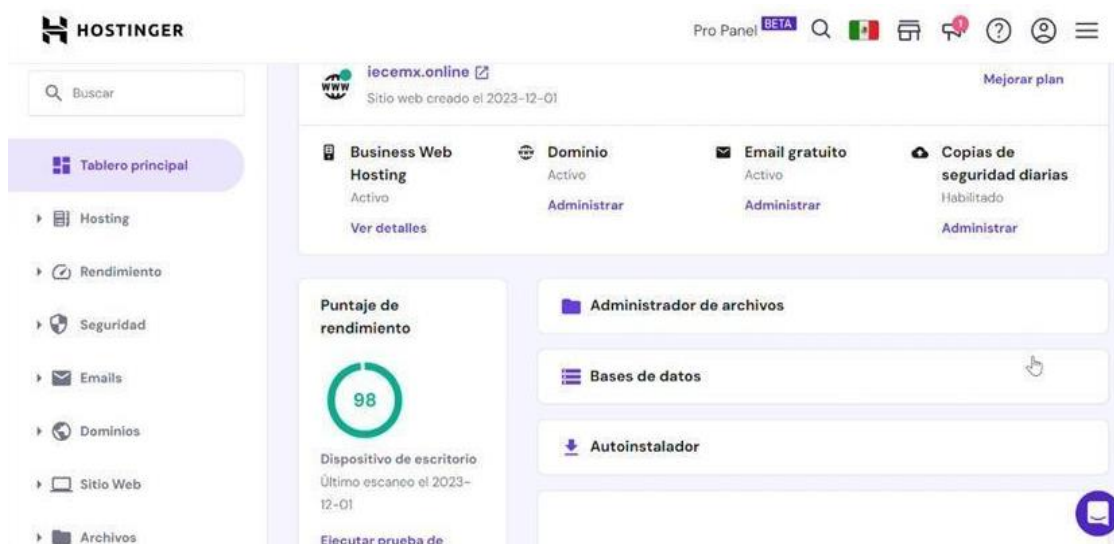


Nota. Elaboración propia.

2. Gestor de archivos: Dentro del apartado denominado “Business Web Hosting”, se seleccionó la opción “Administrar”, lo cual permitió ingresar al tablero de control del servicio. Desde este panel, se procedió a localizar y acceder al “Administrador de archivos”, herramienta que facilita la gestión directa de los directorios del hosting. En la Figura 31, se presenta el tablero de administrador.

Figura 31

Panel de Administración de Hostinger

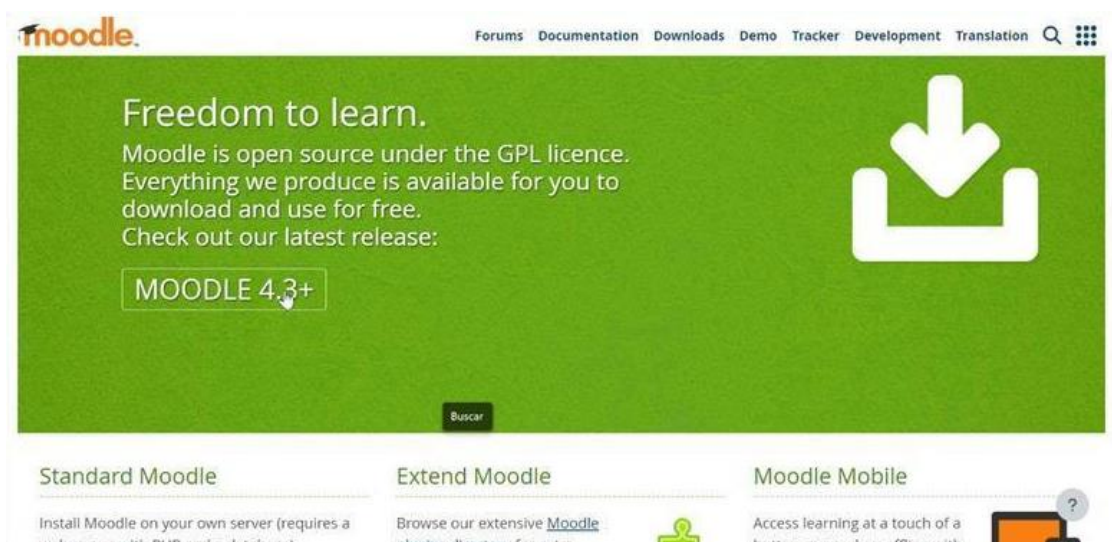


Nota. Elaboración propia.

3. Descargar Moodle: se descarga el instalador de Moodle, accediendo al sitio oficial (<https://download.moodle.org/>) como se muestra en la Figura 32, para subir los archivos descargados al directorio correspondiente.

Figura 32

Sitio oficial de Moodle

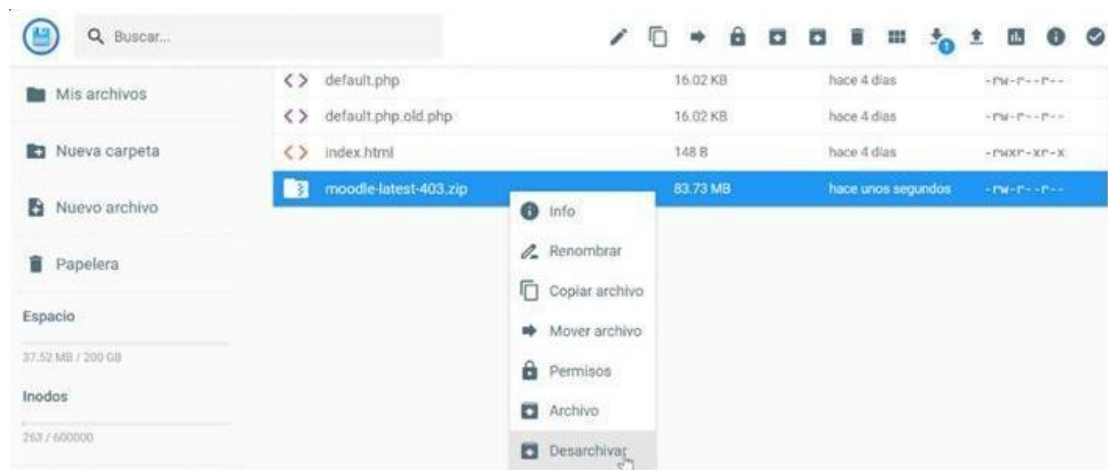


Nota. Elaboración propia.

4. Se accede a la carpeta “public_html” y a continuación se sube el archivo comprimido en formato ZIP que contiene los archivos de Moodle. Tras completar la carga, se utiliza la función “Desarchivar” para extraer su contenido, como se muestra en la Figura 33 .

Figura 33

Descompresión de archivos



Nota. Elaboración propia.

Al completar la descompresión del archivo ZIP, se creará una carpeta denominada "Moodle". El contenido de dicha carpeta se reorganiza y se transfiere a un nuevo directorio previamente creado con el nombre "LMS", que funcionará como el contenedor principal del sistema.

5. Base de datos: Desde el panel principal del proveedor, se accede a la sección correspondiente a Bases de datos, donde se procede a crear una nueva base de datos destinada a Moodle. Durante este proceso, también se genera un usuario con los permisos necesarios para gestionar dicha base, asegurando el vínculo adecuado entre el sistema y la base de datos creada
6. Configuración de Moodle: tras haber finalizado los pasos anteriores, se accede desde el navegador web a la dirección del dominio en la que se alojaron los archivos del LMS Moodle. Al ingresar, se inicia automáticamente el asistente de instalación, Figura 34, el cual guía paso a paso en la configuración inicial del sistema, incluyendo la selección del idioma, la validación de los requisitos del servidor y la conexión con la base de datos previamente creada.

Figura 34

Concluir instalación de Moodle



Nota. Elaboración propia.

7. Configuración final: se establecen las configuraciones finales de Moodle para comenzar a trabajar con la plataforma.

CAPÍTULO 6. Resultados y conclusiones

6.1 Resultados

Para valorar la calidad del entorno virtual diseñado, se aplicaron tres instrumentos de evaluación durante el desarrollo del curso: un examen de inscripción, un examen diagnóstico y una evaluación final. Estos instrumentos permitieron identificar el nivel de conocimientos previos, el progreso alcanzado y la consolidación de las competencias al finalizar el curso.

Cada evaluación se elaboró en función de los contenidos trabajados en los módulos, abordando temas relacionados con la lógica de programación, el uso de widgets y la creación de interfaces gráficas en Android Studio con Kotlin. Las pruebas se aplicaron directamente en la plataforma Moodle, lo que facilitó el registro automático de resultados y la comparación del desempeño individual y grupal. Los resultados de las evaluaciones se evidencian en los anexos del documento

El desarrollo e implementación del curso virtual de programación básica en Kotlin con Android Studio se llevó a cabo exitosamente en la plataforma Moodle, y fue publicado en un dominio particular alojado en Hostinger, lo que facilitó el acceso desde cualquier ubicación geográfica y dispositivo con conexión a internet.

La implementación de la metodología ADDIE facilitó el diseño de un curso introductorio a la programación móvil, permitiendo a los participantes adquirir conocimientos, técnicas y herramientas a través de la solución de problemas hasta lograr desarrollar una interfaz gráfica funcional.

La estructura general del curso, diseñada siguiendo el modelo instruccional ADDIE, permitió organizar los contenidos de manera progresiva y secuencial, respondiendo a las necesidades de un público principiante. Comenzando por un análisis minucioso de los requerimientos de material y los lineamientos de enseñanza. Siendo una base para determinar los elementos incluidos dentro de cada módulo abarcando desde el reconocimiento de los elementos básicos de Android Studio, hasta la creación de interfaces gráficas simples y la integración de lógica de programación para el manejo de datos y eventos.

Dentro de la fase de diseño se estructuraron los módulos del curso de forma progresiva, enriqueciendo la experiencia educativa mediante la incorporación de recursos multimedia como videos, infografías y actividades interactivas desarrolladas con herramientas como Lumi, además de evaluaciones diagnóstica y final para medir el progreso y la comprensión de los participantes. En la Figura 35 se exponen los elementos que componen el módulo 2 “Los tipos de datos emergen”.

Figura 35

Contenido del módulo 2

MÓDULO 2
LOS TIPOS DE DATOS EMERGEN
Analiza el KI de la información

Tipos de Datos Finalización

¿Qué es una variable? Finalización

Clasificando Variables en una App Finalización

Actividad 2: Exploración de Tipos de Datos en Aplicaciones
Abiertos: lunes, 11 de diciembre de 2023, 00:00
Pendientes: lunes, 18 de diciembre de 2023, 00:00
Finalización

No disponible, a menos que: La actividad anterior con finalización está marcada como completada
Editar restricciones

Actividad 3: Crea tu proyecto Finalización

No disponible, a menos que: La actividad anterior con finalización está marcada como completada
Editar restricciones

Actividad 4: Declarando variables Finalización

No disponible, a menos que: La actividad anterior con finalización está marcada como completada
Editar restricciones

Nota. Elaboración propia.

De igual manera en la Figura 36 se presentan la secuencia de la prueba 1 “Los juegos de Cell” dentro del módulo 3, donde se incluyen 3 videos, de los cuales, el primero presenta un ejercicio práctico que los participantes deben resolver por su propia cuenta para posteriormente programar una solución en código kotlin. El segundo video muestra la secuencia seguida dentro de un algoritmo para solucionar el problema. Por último, el video final muestra los pasos a seguir para desarrollar el código correspondiente.

Figura 36

Prueba 1: Los juegos de Cell

 **Prueba 1: Los Juegos de Cell**

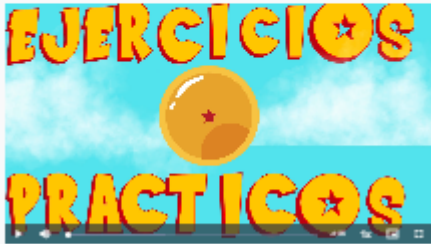
[Tema](#) [Configuración](#) [Calificación avanzada](#) [Más v](#)

Abierto: martes, 15 de julio de 2025, 00:00
Pendiente: sábado, 2 de agosto de 2025, 00:00

¡Bienvenidos a este nuevo reto!

La mejor forma de aprender a programar es resolviendo problemas. En esta sección, enfrentarán un desafío diseñado para ayudarlos a mejorar su pensamiento lógico y su capacidad para estructurar soluciones en Kotlin.

Paso 1: Analiza el problema
Mira el siguiente video donde se te presenta un problema práctico.



Paso 2: Analiza la solución
Analiza el siguiente video, el cual contiene la solución en un diagrama de flujo.



Paso 3: Implementa tu código en Kotlin
Una vez que tengas diseñada tu solución, escribe tu código en Kotlin y realiza las pruebas pertinentes para asegurarte que funciona correctamente.

Paso 4: Compara tu solución
Finalmente, en el tercer video, veas la solución programada en Kotlin. Comprueba cómo se traduce el diagrama de flujo en código y revisa si hay diferencias con tu solución.



Entrega tu archivo: Adjunta el código fuente (.kt) y opcionalmente un archivo de texto con una breve explicación de cómo funciona tu solución.

Nota. Elaboración propia.

Cabe señalar que, a pesar de no aplicar un instrumento formal de evaluación de satisfacción de los usuarios, las observaciones registradas durante el desarrollo del curso y el desempeño mostrado por los participantes permiten inferir una percepción positiva respecto a la claridad de los contenidos y la funcionalidad del entorno virtual, evidenciando que, aunque algunos de los estudiantes carecían de conocimientos previos fue evidente. un incremento en la participación y en la capacidad de resolución autónoma de las actividades prácticas.

Cabe destacar que la inclusión de temáticas motivadoras basadas en la saga Dragon Ball contribuyó notablemente a mantener el interés y la motivación durante el proceso de aprendizaje, fortaleciendo el compromiso de los estudiantes.

6.2 Conclusión

En síntesis, los resultados obtenidos a partir de los tres exámenes aplicados evidencian la efectividad del curso y la pertinencia del modelo instruccional utilizado, confirmando que el entorno Moodle, junto con el enfoque basado en el modelo ADDIE, facilitó una experiencia de aprendizaje estructurada, accesible y motivadora.

Los indicadores derivados de las evaluaciones permiten concluir que el curso logró cumplir con sus objetivos formativos, fortaleciendo las competencias iniciales en programación móvil.

El desarrollo del curso virtual de programación en Kotlin con Android Studio permitió comprobar la viabilidad de diseñar una propuesta educativa innovadora y accesible para estudiantes principiantes en programación móvil. La implementación del modelo ADDIE fue esencial para estructurar cada fase de análisis, diseño, desarrollo, implementación y evaluación, asegurando la coherencia del proyecto y la pertinencia de los contenidos.

La integración de ABP favoreció el aprendizaje significativo, pues situó a los estudiantes frente a retos prácticos inspirados en la temática de Dragon Ball. Este enfoque lúdico, complementado con elementos de gamificación, resultó clave para mantener la motivación de los participantes y consolidar una experiencia dinámica en el proceso de enseñanza-aprendizaje.

El uso de Moodle como Sistema Gestor de Aprendizaje, apoyado por el servicio de alojamiento Hostinger, garantizó la accesibilidad del curso desde distintos dispositivos y ubicaciones. Esta

infraestructura tecnológica brindó un entorno estable, replicable y flexible, lo que facilita su aplicación en futuros proyectos educativos.

Asimismo, el empleo de herramientas de diseño como Canva permitió elaborar materiales visuales atractivos y comprensibles, mientras que la producción de videos y actividades interactivas mediante Lumi enriqueció la dinámica de cada módulo. Los diagramas de flujo y de estado, elaborados para planificar procesos y representar la lógica de programación, contribuyeron a dar claridad al desarrollo instruccional y a reforzar la comprensión de los estudiantes sobre la estructura de los algoritmos.

En el ámbito práctico, Android Studio se consolidó como la herramienta central del curso, al permitir que los estudiantes experimentaran directamente con la programación en Kotlin y la creación de interfaces gráficas básicas. Este acercamiento práctico confirmó que, aun con un enfoque introductorio, es posible sentar bases sólidas en programación móvil y motivar la exploración de desarrollos más complejos en etapas posteriores.

La experiencia de producción también fortaleció la capacidad de gestión de recursos y la planeación didáctica. La construcción de videos educativos, infografías, guiones narrativos y simulaciones no solo aportó al curso un carácter interactivo, sino que evidenció la importancia de integrar recursos multimedia para atender diferentes estilos de aprendizaje.

En conjunto, el curso constituye una aportación académica significativa al campo de la educación virtual en programación. La combinación de metodologías pedagógicas, recursos digitales, infraestructura tecnológica y estrategias creativas demostró ser efectiva para introducir a los estudiantes en la programación móvil de forma clara y atractiva. Finalmente, se concluye que el proyecto es replicable y adaptable a diversos contextos educativos, consolidándose como un modelo innovador para el fortalecimiento de competencias digitales en entornos virtuales de aprendizaje.

6.3 Trabajos futuros

Con base en la experiencia obtenida durante el desarrollo e implementación del curso y considerando los resultados alcanzados en términos de aprendizaje y participación, se proponen las siguientes líneas de trabajo para su mejora:

- **Ampliación de contenidos:** Se plantea la inclusión de módulos intermedios y avanzados que aborden temas fundamentales para el desarrollo completo de aplicaciones móviles, tales como la integración y gestión de bases de datos, el consumo de APIs externas, el manejo de notificaciones push, así como la optimización y publicación de aplicaciones en la tienda Google Play.
- **Evaluación continua y adaptativa:** Se propone implementar sistemas de evaluación que proporcionen retroalimentación inmediata y personalizada, permitiendo a los estudiantes identificar errores y dificultades en tiempo real, y recibir sugerencias específicas para su corrección.
- **Fomento del trabajo colaborativo:** Se recomienda desarrollar proyectos grupales en los que los participantes apliquen de manera conjunta los conocimientos adquiridos, promoviendo habilidades blandas, el trabajo en equipo y la resolución colaborativa de problemas.

Estas líneas de trabajo buscan fortalecer la propuesta educativa, ampliando su alcance y efectividad, y asegurando que se mantenga alineada con las exigencias y tendencias actuales en la formación en programación móvil.

Referencias Bibliográficas

- Acerca de Moodle - MoodleDocs*. (2024, 25 octubre). Moodle. Recuperado 26 de febrero de 2025, de https://docs.moodle.org/all/es/Acerca_de_Moodle
- Agudelo, M. (2009). Importancia del diseño instruccional en ambientes virtuales. En M. Agudelo, *Nuevas ideas en informática educativa* (págs. 118-127). Medellín.
- Aguirre, J. F. (2015). Una experiencia para fortalecer los procesos de enseñanza de la programación mediante el uso de entornos virtuales de aprendizaje. *Enseñanza Y Aprendizaje De Ingeniería De Computadores*. <https://doi.org/10.30827/digibug.36570>
- Android Developers. (2023). *Developer documentation*. Google. <https://developer.android.com>
- Aulaplaneta. (2015). *¿Qué es un entorno virtual de aprendizaje (EVA)?* Grupo Planeta. <https://www.aulaplaneta.com/2015/01/29/recursos-tic/que-es-un-entorno-virtual-de-aprendizaje-eva>
- Barrera, C. C. F., Vega, J. A. N., & Morales, F. H. F. (2020). Desarrollo de competencias digitales en programación de aplicaciones móviles en estudiantes de noveno grado a través de tres estrategias pedagógicas. *Boletín Redipe*, 9(4), 179-191.
- Belloch, C. (2012). Entornos virtuales de aprendizaje. *Valencia: universidad de Valencia*.
- Belloch, C. (2017). Diseño instruccional.
- Boneu, J. M. (2007). Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos. *Rev. U. Soc. Conocimiento*, 4, 36.
- Branch, R., & Merrill, M. D. (2011). Characteristics of instructional design models. In R. Reiser & J. Dempsey (Eds.), *Trends and issues in instructional design and technology* (3rd ed., pp. 8-17). Upper Saddle River, NJ: Merrill-Prentice Hall.
- Bueno, P. M. (2018). *Aprendizaje basado en problemas (ABP) y habilidades de pensamiento crítico, ¿una relación vinculante?* Obtenido de <https://www.redalyc.org/journal/2170/217059664008/html/#:~:text=Los%20componentes%20centrales%20del%20modelo,guiar%20a%20los%20estudiantes%20a>
- Canva. (s.f.). *Canva*. Obtenido de https://www.canva.com/es_es/about/
- Carrillo, M. J., & Carlos, R. G. L. (2018). *Diseñando el aprendizaje desde el Modelo ADDIE* (Bachelor's thesis, Universidad de La Sabana).

- Cázales, Z. N., & Granados, H. M. M. (2017). Panorama de la educación a distancia en México. *Latinoamericana De Estudios Educativos*, 13(1), 65–82. <https://doi.org/10.17151/rlee.2017.13.1.4>
- Cedeño Romero, E. L., & Murillo Moreira, J. A. (2019). Entornos virtuales de aprendizaje y su rol innovador en el proceso de enseñanza. *Revista de Ciencias Humanísticas y Sociales (ReHuSo)*, 4(1), 138-148.
- CHUJUTALLI, K. C. T. DISEÑO DE UN CURSO DE CAPACITACIÓN A DISTANCIA PARA MEJORAR LAS COMPETENCIAS DE ACOMPAÑAMIENTO VIRTUAL EN DOCENTES UNIVERSITARIOS.
- Degiovanni, H. a. R. (2025, February 17). *Educación en línea en México: Un Análisis desde sus orígenes hasta el Modelo Educativo MIRA*. Universidad Loyola De América. <https://universidadloyola.edu.mx/educacion-en-linea-en-mexico-un-analisis-desde-sus-origenes-hasta-el-modelo-educativo-mira/>
- Delgado, J. M. C. (2015). Programación informática y robótica en la enseñanza básica. *Avances en supervisión educativa*, (24).
- Dick, W. y Carey, LM (1977). Evaluación de necesidades y diseño instruccional. *Tecnología educativa* , 17 (11), 53-59.
- Educación Lumi. (2024, 21 mayo). *Crea H5P y aloja tu contenido en Lumi - Lumi Education*. <https://lumi.education/es/>
- Galán, Y. I. J., Jaime, J. H., & Contreras, E. H. (2014). Propuesta de un modelo iterativo e incremental para la evaluación por competencias de los aprendizajes. *Revista Iberoamericana de Producción Académica y Gestión Educativa*, 1(2), 1-23.
- García, J. d. (2021). *Estudio de la experiencia de usuario en los sistemas de gestión del aprendizaje*. IE revista de investigación educativa de la REDIECH.
- González, H. T. (2022). Instrumento de evaluación para el desarrollo de cursos en entornos virtuales de enseñanza-aprendizaje. *Edutec, Revista Electrónica de Tecnología Educativa*, (79), 30-45.
- GUIMERÁ OROZCO, A. R. Í. S. T. I. D. E. S. (2018). *Iniciación a Android en Kotlin. Casos prácticos*. Ediciones Paraninfo, SA.
- Hernández, D. D. M., Maqueda, M. C., Velázquez, E. C., & Maldonado, O. D. R. I. (2024). Ambiente virtual para la enseñanza de fracciones en educación básica. *Ingenio y*

- Conciencia Boletín Científico de la Escuela Superior Ciudad Sahagún, 11(22), 97-107.
<https://doi.org/10.29057/escs.v11i22.11893>
- Hostinger. (2025, 5 de marzo). *¿Qué es un hosting y cómo funciona?* Recuperado el 24 de agosto de 2025, de <https://www.hostinger.com/es/tutoriales/que-es-un-hosting>
- Introducción a Android Studio.* (s. f.). Android Developers.
<https://developer.android.com/studio/intro?hl=es-419>
- Kurt, S. (2015). Dick and Carey Instructional Model. *Educational Technology*. Recuperado de <https://educationaltechnology.net/dick-and-carey-instructional-model/>
- LeoKhoa. (s.f.). *Laragon*. Laragon. <https://laragon.org/>
- Lezcano, L., & Vilanova, G. (2017). Instrumentos de evaluación de aprendizaje en entornos virtuales. Perspectiva de estudiantes y aportes de docentes. *Informes Científicos Técnicos-UNPA*, 9(1), 1-36.
- Lima, MGB (2010). El modelo de diseño instructivo ASSURE aplicado a la educación a distancia. *Tlatemoani: revista académica de investigación*, (1), 9.
- María, G. M. J., & De Alcalá, U. (2021). *Análisis del lenguaje Kotlin mediante el desarrollo de una aplicación web y móvil*. E_BUAH. Biblioteca Digital Universidad de Alcalá.
<https://ebuah.uah.es/dspace/handle/10017/49939>
- Morales-González, B., Edel-Navarro, R., & Aguirre-Aguilar, G. (2014). Modelo ADDIE (análisis, diseño, desarrollo, implementación y evaluación): Su aplicación en ambientes educativos. *Los modelos tecno-educativos, revolucionando el aprendizaje del siglo XXI*, 33-46.
- Nácher, M., Badenes Ribera, L., Torrijos, C., Ballesteros, M., & Cebadera, E. (2018). GoKoan: Una plataforma e-learning de apoyo al estudio en estudiantes universitarios. In *IN-RED 2018. IV Congreso Nacional de Innovación Educativa y Docencia en Red* (pp. 184-198). Editorial Universitat Politècnica de València. Recuperado de <https://riunet.upv.es/handle/10251/113174>
- Quiroz, D. L. Z., & Quiroz, M. S. Z. (2019). *Las tecnologías de la información y las comunicaciones (tics) en la educación superior: consideraciones teóricas*.
<https://refcale.uleam.edu.ec/index.php/refcale/article/view/2750>
- Quispe Zabaleta, K. (2019). *Gagné y la programación instruccional* (Monografía de pregrado). Universidad Nacional de Educación Enrique Guzmán y Valle, Lima, Perú.

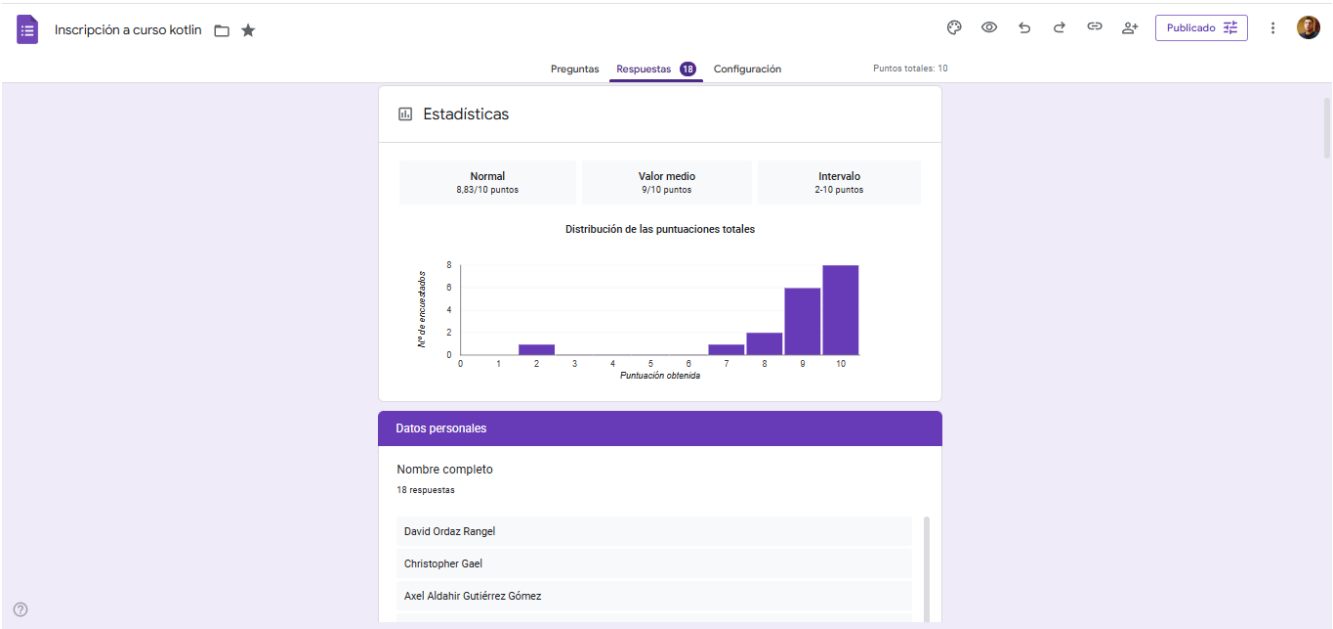
Software de grabación y edición de audio | Adobe Audition. (s.f.). Obtenido de Adobe:
<https://www.adobe.com/mx/products/audition.html>

Soriano, J. E. A. (2019). *Android: Programación de dispositivos móviles a través de ejemplos*. Marcombo.

Ventajas del aprendizaje basado en la resolución de problemas [Infografía] | Aulaplaneta. (2015, 25 agosto). <https://www.aulaplaneta.com/2015/08/25/recursos-tic/ventajas-del-aprendizaje-basado-en-la-resolucion-de-problemas>

Anexos

Resultados de formulario de inscripción



Resultados de evaluación diagnóstico

	Nombre / Apellido(s)	Dirección Email	Estado	Comenzado en	Finalizado	Tiempo empleado	Calificación/100.00	Q. 1 /8.00	Q. 2 /8.00	Q. 3 /4.00	Q. 4 /4.00	Q. 5 /10.00	Q. 6 /5.00	Q. 7 /8.00	Q. 8 /4.00	Q. 9 /8.00	Q. 10 /4.00	Q. 11 /10.00	Q. 12 /7.00	Q. 13 /8.00	Q. 14 /4.00	Q. 15 /8.00
☐	David Ordaz Rangel	or475179@uaeh.edu.mx	Terminados	15 de julio de 2025 21:53	15 de julio de 2025 22:05	11 mins 15 segundos	72.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✗ 0.00
☐	Angel De Jesus Atanasio Diaz De Jesus Atanasio Diaz	atd98414@uaeh.edu.mx	Terminados	16 de julio de 2025 07:19	16 de julio de 2025 07:23	3 mins 19 segundos	80.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00
☐	Axel Aldahir Gutiérrez Gómez	gu400871@uaeh.edu.mx	Terminados	18 de julio de 2025 16:48	18 de julio de 2025 16:53	4 mins 26 segundos	84.00	✓ 8.00	✓ 8.00	✓ 4.00	✗ 0.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00
☐	Ashley Torres perez	tot476602@uaeh.edu.mx	Terminados	20 de julio de 2025 00:23	20 de julio de 2025 00:38	15 mins 9 segundos	68.00	✗ 0.00	✓ 8.00	✓ 4.00	✗ 0.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✗ 0.00
☐	Roberto Estrada Tovar	es464681@uaeh.edu.mx	Terminados	22 de julio de 2025 13:37	22 de julio de 2025 13:56	18 mins 38 segundos	77.33	✓ 8.00	✓ 8.00	✓ 4.00	✗ 0.00	3.33	✗ -	✓ 8.00	✓ 4.00	✓ 8.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00
☐	Luz Aketzali Vargas Cruz	va525687@uaeh.edu.mx	Terminados	25 de agosto de 2025 16:23	25 de agosto de 2025 16:26	3 mins 29 segundos	42.00	✗ 0.00	✗ 0.00	✗ 0.00	✓ 4.00	✗ 0.00	✗ 0.00	✓ 8.00	✓ 4.00	✗ 0.00	✓ 4.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✗ 0.00
☐	Luz Aketzali Vargas Cruz		Terminados	25 de agosto de 2025 16:28	25 de agosto de 2025 16:32	3 mins 47 segundos	57.33	✗ 0.00	✓ 8.00	✗ 0.00	✓ 4.00	3.33	✗ 0.00	✓ 8.00	✓ 4.00	✗ 0.00	✗ 0.00	✓ 10.00	✗ 0.00	✓ 8.00	✓ 4.00	✓ 8.00
	Promedio general						68.67 (7)	2.29 (7)	6.86 (7)	2.86 (7)	2.29 (7)	6.67 (7)	0.00 (7)	8.00 (7)	4.00 (7)	5.71 (7)	3.43 (7)	10.00 (7)	0.00 (7)	8.00 (7)	4.00 (7)	4.57 (7)

Resultados de evaluación final

[illegible]