






Desarrollo de un curso virtual para el aprendizaje de Kotlin Development of a virtual course for learning Kotlin

Moisés Cruz-Pancardo ^a, Mireya Clavel-Maqueda ^a, Edgar Olgún Guzmán ^a, Norma Laura Salazar-Viveros ^a,
Eduardo Cornejo-Velázquez ^{a,*}

^a Área Académica de Computación y Electrónica, Universidad Autónoma del Estado de Hidalgo, 42184, Pachuca, Hidalgo, México.

Resumen

El mercado laboral de desarrollo de aplicaciones móviles ha experimentado un crecimiento constante impulsado por el uso masivo de dispositivos inteligentes y la necesidad de soluciones innovadoras en diversos sectores. Sin embargo, los estudiantes principiantes suelen enfrentar dificultades debido a la complejidad de los entornos de desarrollo y a la falta de recursos educativos accesibles. Este trabajo describe el diseño e implementación de un curso virtual para la enseñanza del lenguaje de programación Kotlin, basado en el modelo instruccional ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación) e integrado con estrategias de aprendizaje basado en problemas y elementos lúdicos inspirados en el anime *Dragon Ball*. La secuencia didáctica propuesta permitió introducir a los participantes en la programación móvil para la plataforma Android mediante la combinación de metodologías pedagógicas, recursos tecnológicos y actividades prácticas. El curso resultante es integral, replicable y adaptable a distintos contextos educativos, y está orientado al fortalecimiento de competencias digitales fundamentales en los estudiantes. *Palabras Clave:* aplicaciones móviles, aprendizaje basado en problemas, enseñanza lúdica, experiencia de usuario

Abstract

The mobile application development job market has experienced continuous growth driven by the widespread use of smart devices and the demand for innovative solutions across various sectors. However, beginner students often face difficulties due to the complexity of development environments and the lack of accessible educational resources. This paper describes the design and implementation of a virtual course for teaching the Kotlin programming language, based on the ADDIE (Analysis, Design, Development, Implementation, and Evaluation) instructional model and integrated with problem-based learning strategies and playful elements inspired by the *Dragon Ball* anime. The proposed teaching sequence introduced participants to mobile programming for the Android platform through a combination of pedagogical methodologies, technological resources, and practical activities. The resulting course is comprehensive, replicable, and adaptable to different educational contexts, and is aimed at strengthening essential digital competencies in students.

Keywords: mobile applications, problem-based learning, playful teaching, user experience

1. Introducción

La programación es una competencia esencial para que los egresados universitarios puedan desenvolverse en un mundo cada vez más digitalizado. El desarrollo de aplicaciones móviles ha tenido un crecimiento sostenido impulsado por el uso masivo de dispositivos inteligentes. En este contexto, Android se ha posicionado como la plataforma más utilizada a nivel mundial debido a la amplia disponibilidad de dispositivos y a su bajo costo de adquisición (Guimerá, 2018).

En el ámbito del desarrollo móvil, el lenguaje de programación Kotlin ha ganado relevancia desde que Google lo adoptó como lenguaje oficial para Android en 2017

(Oliveira et al., 2020). Su sintaxis clara y su lógica de programación amigable permiten implementar funciones, microservicios o servicios web de forma eficiente, reduciendo tiempos de desarrollo (Monge, 2021; Góis-Mateus y Martínez, 2019).

Sin embargo, aprender a desarrollar aplicaciones móviles con Kotlin representa un reto para estudiantes principiantes, especialmente por la complejidad del entorno Android y la ausencia de una guía pedagógica que acompañe sus primeros pasos.

Si bien existen numerosos recursos de aprendizaje en Internet —tutoriales, videos y cursos independientes—,

*Autor para la correspondencia: ecornejo@uaeh.edu.mx

Correo electrónico: cr419703@uaeh.edu.mx (Moisés Cruz-Pancardo), mclavel@uaeh.edu.mx (Mireya Clavel-Maqueda), eolguin@uaeh.edu.mx (Edgar Olgún Guzmán), normalis@uaeh.edu.mx (Norma Laura Salazar-Viveros), ecornejo@uaeh.edu.mx (Eduardo Cornejo-Velázquez)

muchos de ellos presentan una estructura desarticulada y carecen de retroalimentación, evaluación continua o acompañamiento docente, elementos indispensables para consolidar aprendizajes significativos. Actualmente existen alternativas como Codecademy, Udemy, Coursera u Omegaup, esta última ampliamente utilizada en México y Latinoamérica para el fortalecimiento de habilidades algorítmicas (Omegaup, 2024).

Sin embargo, estas plataformas no siempre integran secuencias didácticas progresivas, actividades prácticas contextualizadas o mecanismos formales de seguimiento que faciliten el aprendizaje de programación en un entorno profesional como Android Studio.

Estudios recientes destacan que la efectividad en la enseñanza de la programación aumenta cuando se incorpora retroalimentación inmediata, materiales interactivos y una progresión clara de los contenidos (Cedeño y Murillo, 2019; Martínez-Hernández et al., 2024).

Por ello, los docentes enfrentan el desafío de diseñar estrategias educativas que permitan introducir de manera estructurada a los estudiantes en el desarrollo de aplicaciones móviles, especialmente durante sus primeros acercamientos a la programación.

La educación virtual se ha consolidado como un modelo asequible, versátil y ampliamente adoptado, al permitir que estudiantes de distintas regiones accedan a materiales formativos de calidad sin limitaciones geográficas (Martínez-Hernández et al., 2024).

En este marco, los Entornos Virtuales de Aprendizaje (EVA) han demostrado ser una herramienta eficaz para mejorar la comprensión de conceptos y fortalecer tanto habilidades tecnológicas como personales. Su implementación facilita el aprendizaje autónomo, la retroalimentación inmediata y la resolución de problemas mediante ejercicios prácticos estructurados (Cedeño y Murillo, 2019).

En este trabajo se describe el desarrollo de un curso virtual interactivo para la enseñanza de los fundamentos del lenguaje de programación Kotlin a nivel básico, con el propósito de fomentar el aprendizaje autónomo, el pensamiento lógico y la resolución de problemas. La propuesta combina enfoques activos y actividades prácticas con la finalidad de fortalecer la comprensión, participación y motivación de los estudiantes.

Para la implementación del curso se utilizó la plataforma Open Source Moodle (Moodle, 2025) como Sistema de Gestión del Aprendizaje (LMS, por sus siglas en inglés). Esta plataforma permite integrar contenidos educativos en diversos formatos y trabajar en entornos locales durante la etapa de desarrollo (Aguado, 2021).

Moodle facilitó la incorporación de contenidos teóricos y prácticos, actividades interactivas, evaluaciones y mecanismos de retroalimentación, complementados con la estrategia de Aprendizaje Basado en Problemas (ABP) y elementos lúdicos diseñados para mantener el interés de los participantes.

2. Diseño metodológico

Para guiar el desarrollo de materiales educativos de calidad es importante utilizar herramientas metodológicas que permitan organizar y planificar las actividades del proceso de creación.

Se utilizó el modelo ADDIE (Análisis, Diseño, Desarrollo, Implementación y Evaluación) como guía en el diseño instruccional. En la Figura 1 se presentan las 5 fases del modelo (Templos, 2020) que posibilitan un proceso de creación flexible, adaptable y que permite la retroalimentación en cada fase.



Figura 1: Etapas del modelo ADDIE basado en García (2020).

2.1. Aprendizaje basado en Problemas

El Aprendizaje basado en Problemas (ABP) es una metodología activa que prioriza el aprendizaje del estudiante, la investigación y la reflexión. Impulsa un enfoque más participativo de los estudiantes quienes se involucran activamente en la resolución de problemas planteados por el profesor (Bueno, 2018).

La idea principal de la estrategia ABP es que el conocimiento se construye de manera más efectiva cuando los estudiantes aplican los conceptos teóricos en contextos prácticos. Además, promueve el trabajo colaborativo y el desarrollo de habilidades analíticas. Algunas de las ventajas de la estrategia ABP son:

- Los estudiantes adquieren conocimientos como resultado de la resolución de problemas reales, impulsando su retención y comprensión.
- Promueve el análisis, síntesis y evaluación de la información para encontrar soluciones efectivas.
- Los estudiantes asumen un rol activo en su aprendizaje de búsqueda de información, toma de decisiones y fundamentación de sus soluciones.

2.2. Desarrollo iterativo e incremental

El desarrollo iterativo e incremental es una metodología ampliamente utilizada en la ingeniería de software (Merchán-Narváez et al., 2024; Fernández-Soto y Alfaro-Pérez, 2020). Permite la construcción progresiva de los contenidos del curso

mediante ciclos repetidos de planificación, desarrollo y evaluación.

En lugar de desarrollar el curso completo desde el inicio, se realizaron productos parciales y funcionales en cada iteración, lo que facilitó la incorporación de mejoras continuas basadas en la evaluación y retroalimentación.

Este enfoque garantizó que cada módulo fuera validado antes de avanzar al siguiente y realizar ajustes de manera continua para mejorar la experiencia de aprendizaje. Su implementación contribuyó a un proceso de enseñanza - aprendizaje dinámico, centrado en la mejora progresiva de los contenidos y la interacción con los estudiantes.

3. Desarrollo del curso

Para del desarrollo del curso se utilizó la metodología ADDIE complementado con estrategias de ABP. El curso virtual desarrollado incluye los siguientes elementos:

- Contenido educativo estructurado y organizado.
- Ejercicios prácticos que refuerzan el aprendizaje.
- Mecanismos de evaluación y retroalimentación efectiva.
- Acceso a herramientas de desarrollo para trabajar en ejercicios de programación en Kotlin.

3.1. Análisis

Se realizó un análisis detallado de fuentes de Internet y libros para comprender las necesidades de aprendizaje y fundamentar 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.

3.2. Diseño

El curso desarrollado está compuesto de diez módulos organizados en la secuencia didáctica que organiza los contenidos y dificultad de manera incremental y progresiva para guiar a los participantes en el aprendizaje de los conceptos básicos.

Cada módulo está diseñado para guiar y fortalecer el aprendizaje mediante ejercicios y actividades dinámicas, asegurando que los participantes adquieran conocimientos teóricos y habilidades aplicables en el desarrollo de aplicaciones móviles.

En la Figura 2, se presenta el diagrama del caso de uso del módulo 3: Manipulando datos en Kotlin. Se describen los elementos que integran el módulo y la forma en que se presentan los contenidos teóricos, la interacción con los participantes en el curso, la actividad y evaluación propuesta.

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 específico planteado. Los problemas o desafíos de programación a resolver están inspirados en el anime *Dragon Ball* y los participantes implementan la solución en entorno de Android Studio escribiendo códigos en Kotlin.



Figura 2: Caso de uso "Manipulando datos en Kotlin".

Para el diseño de las actividades se consideran elementos lúdicos para mantener motivados a los participantes. En la Figura 3 se presenta el diagrama de actividades de la evaluación del módulo 1 "Kotlin y Android Studio" mediante una sopa de letras.

El diseño de los elementos busca favorecer la comprensión de los procesos construcción de programas con Kotlin en el ambiente de Android Studio. Para mantener el interés del participante, trabaja en los elementos fundamentales del entorno y se fomenta su autonomía desde los primeros módulos en el desarrollo de aplicaciones móviles.

3.3. Desarrollo

Para el desarrollo de los contenidos del curso se realizó la selección de los patrones visuales y la gama cromática. Se utilizan tonos verdes por su asociación con el crecimiento, claridad mental y la armonía, características que favorecen el carácter instructivo de los contenidos.

Para conservar el atractivo visual de los elementos gráficos del anime se manejan colores turquesa y cian en los elementos para el desarrollo de los ejercicios prácticos.

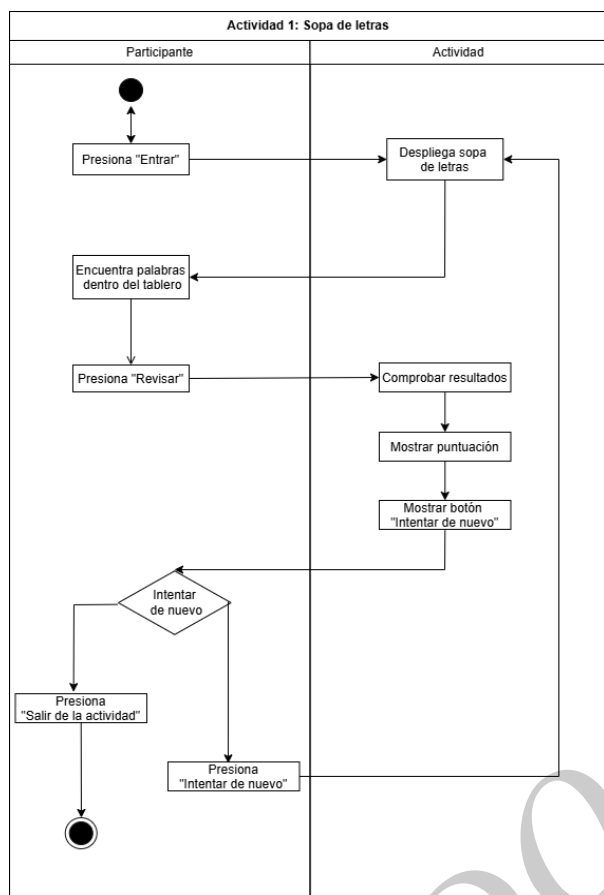


Figura 3: Diagrama de actividades: "Sopa de letras".

Con el propósito de desarrollar los contenidos de los módulos del curso se definió el backlog del proyecto para planificar y dar seguimiento a las actividades, entregables, evaluaciones y retroalimentación durante todo el proceso de desarrollo.

Para la construcción de los materiales y recursos pedagógicos en formatos digitales se utilizaron diversas herramientas especializadas de acceso gratuito: Canva para las imágenes e infografías, Adobe Premiere Express para videos, DFD para algoritmos, Lumi para contenidos interactivos y el IDE Android Studio para la creación de códigos.

En la Figura 4 se presenta la portada principal del curso donde se pueden observar el estilo de los elementos gráficos, textuales e iconográficos elegidos para los contenidos.



Figura 4: Portada principal del curso.

3.4. Implementación

Los contenidos digitales construidos se integraron en la plataforma Moodle, con el objetivo de organizar los componentes y asegurar una experiencia de usuario. Se definieron mapas de navegación para cada módulo, con base en la secuencia didáctica diseñada para abordar los contenidos de manera progresiva y secuencial.

Con el propósito de configurar un entorno de desarrollo controlado para la depuración, validación funcional y corrección de errores previo al despliegue final se habilitó un entorno local de desarrollo en Laragon dentro del que se hizo la instalación del LMS Moodle. Además, se implementó el control de versiones con Git para una gestión ordenada del proyecto en la implementación de mejoras, revisión de cambios y aseguramiento de la calidad del producto final.

Respecto a la integración de componentes, en la Figura 5 se presentan los contenidos del módulo 2 "Los tipos de datos emergen" para ejemplificar la integración de los videos, infografías y actividades interactivas, así como de las evaluaciones diagnósticas y formativas para medir el progreso de los participantes.

La estrategia ABP se implementó en las secciones en las que se plantea a los participantes el reto a resolver. De forma general, se estableció que, en estas secciones, se presentara el problema a través del diseño de algoritmos y la construcción de códigos en Kotlin.

En la Figura 6 se presenta el reto "Los juegos de Cell", incluido dentro del módulo 3. Se incluyen 3 videos, en el primero se presenta el problema que el participante debe resolver por su cuenta; en el segundo video, se describe el algoritmo que se utiliza para tener una solución y el tercer video se presentan los pasos para construir el código que corresponde con el algoritmo.

El curso que se encuentra en línea en la dirección <https://lms.iecemx.online/enrol/index.php?id=2> y se ha habilitado el acceso como usuario invitado con la contraseña Temporal01.

3.5. Evaluación

Para la evaluación del curso se aplicó el instrumento de seguimiento y evaluación de cursos virtuales propuesto por González (2022). El instrumento considera los siguientes elementos: diseño instruccional, enfoque pedagógico, arquitectura de la instrucción, usabilidad y accesibilidad e interacciones y comportamiento; por lo que permite asegurar la calidad del producto e identificar mejoras.

Para complementar la evaluación se habilitó un formulario de inscripción junto con una evaluación diagnóstica que es llenado por las personas que participan en el curso. Al finalizar el curso, los participantes realizan una evaluación final con el objetivo de validar los conocimientos y habilidades adquiridos, además de realizar la comparación con las evaluaciones diagnósticas para evaluar el progreso de cada participante.

evaluación diagnóstica para conocer el nivel de conocimientos previos de los participantes respecto a conceptos básicos de programación, manejo de Android Studio y fundamentos de Kotlin.

Al finalizar el curso, se aplicó una evaluación final que permitió comparar el nivel de progreso de los participantes. En términos generales, se observaron mejoras en la comprensión de estructuras básicas del lenguaje, el uso de widgets y la creación de interfaces gráficas simples.

Asimismo, mediante los registros de la plataforma Moodle se identificó una mayor participación en los módulos donde se integraron actividades basadas en problemas, lo que sugiere un impacto positivo de la estrategia ABP en la motivación y permanencia de los participantes.

▼ Módulo 2: Los tipos de datos emergen

Finalización ▼

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 Finalización ▼
Abiertos: lunes, 11 de diciembre de 2023, 00:00 Pendientes: lunes, 18 de diciembre de 2023, 00:00

Actividad 3: Crea tu proyecto Finalización ▼

Actividad 4: Declarando variables Finalización ▼

Figura 5: Contenidos del módulo “Los tipos de datos emergen”.

Los instrumentos utilizados permitieron identificar el nivel de conocimientos previos, el progreso alcanzado y la consolidación de los conocimientos al finalizar el curso.

Las evaluaciones de contenido abordan los temas de lógica de programación, el uso de widgets y la creación de interfaces gráficas. Las pruebas asociadas a los contenidos se habilitaron directamente en la plataforma Moodle lo que permitió el registro automático de resultados y la comparación del desempeño individual y grupal.

4. Resultados

La primera implementación del curso permitió identificar diversas oportunidades de mejora en los materiales, actividades y dinámicas de interacción. Se habilitó una

Prueba 1: Los Juegos de Cell

¡Bienvenidos a este nuevo reto!

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

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

✅ 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, verás 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.

¡Es hora de demostrar tus habilidades como programador Zi! 🎮

Figura 6: Reto “Los juegos de Cell”.

La retroalimentación cualitativa, obtenida mediante formularios adicionales, destacó que los elementos visuales y lúdicos inspirados en el anime facilitaron la comprensión y aumentaron el interés por las actividades prácticas. Estos resultados preliminares permiten considerar a la propuesta como una alternativa viable y atractiva para introducir a estudiantes principiantes en el desarrollo móvil con Kotlin.

Conclusiones

El modelo instruccional ADDIE permitió organizar los contenidos del curso de manera progresiva y secuencial en correspondencia al diseño definido para el curso. El análisis de requerimientos junto con los lineamientos establecidos por la secuencia didáctica para cada módulo del curso permitió atender de mejor manera las necesidades de los participantes.

La experiencia educativa de los participantes fue enriquecida con la incorporación de elementos multimedia como videos, infografías y actividades interactivas para fortalecer los conocimientos de los elementos básicos de Kotlin y Android Studio, la creación de interfaces gráficas y el desarrollo de la lógica de programación para el manejo de datos y eventos.

El curso desarrollado constituye una aportación académica al campo de la educación virtual en el área de la programación. La combinación de metodologías pedagógicas, recursos digitales, infraestructura tecnológica y estrategias creativas resultó ser efectiva para introducir a los estudiantes principiantes en la programación de aplicaciones móviles.

Referencias

Aguado, I. (2021). eXelearning como herramienta para la virtualización de la enseñanza: el diseño de objetos de aprendizaje para el estudio del paisaje urbano. *Ikastorratza. e-Revista de didáctica*

- Bueno, P. M. (2018). *Aprendizaje basado en problemas (ABP) y habilidades de pensamiento crítico: ¿una relación vinculante?* Revista Atlante: Cuadernos de Educación y Desarrollo. <https://www.eumed.net/rev/atlanter/2018/08/aprendizaje-problemas-pensamiento.html>
- Cedeño, E. y Murillo, J. (2019). Entornos virtuales de aprendizaje y su rol innovador en el proceso de enseñanza. *Rehuso*, 4(1), 138-148. <https://doi.org/10.33936/rehuso.v4i1.2156>
- Fernández-Soto, E., y Alfaro-Pérez, D. (2020). El modelo iterativo e incremental para el desarrollo de la aplicación de realidad aumentada Amón_RA. *Tecnología en Marcha*, 33(8), 165-177. <https://doi.org/10.18845/tm.v33i8.5153>
- García, R. (2020). La importancia de la aplicación del modelo instruccional ADDIE en la archivística. *Tlatemoani: revista académica de investigación*, 11(33):95-108.
- Góis-Mateus, B. y Martínez, M. (2019). An empirical study on quality of Android applications written in Kotlin language. *Empirical Software Engineering*, 24(6), 3356-3393. <https://doi.org/10.1007/s10664-019-09727-4>
- 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, O. A. (2018). *Iniciación a Android en Kotlin. Casos prácticos*. Ediciones Paraninfo, SA.
- Martínez-Hernández, D. D., Clavel-Maqueda, M., Cornejo-Velázquez, E., e Islas-Maldonado, O. D. R. (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.
- Merchán-Narváez, N. J., Palma-Peralta, E. E., Poma-Japón, D. X. (2024). Comparación de metodologías ágiles para el desarrollo de software. *MQRInvestigar*, 8(1), 5052-5074.
- Monge, L. S. P. (2021). Análisis del lenguaje Kotlin mediante el desarrollo de una aplicación web y móvil. *Biblioteca Digital Universidad de Alcalá*. <https://ebuah.uah.es/dspace/handle/10017/49939>
- Moodle. (2025). Moodle LMS. <https://moodle.org>
- Oliveira, V., Teixeira, L., Ebert, F. (2020). On the adoption of Kotlin on android development: A triangulation study. In 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 206-216).
- Omegaup. (2024). Programando un mejor futuro para todos. <https://omegaup.com>
- Templos, L. (2020). Modelo instruccional ADDIE. *Logos Boletín Científico de la Escuela Preparatoria No.2*, 7(14):24-26.