

# Brain Boost Capstone

***Integrantes:*** Macarena Bertero  
Ignacio León  
Claudio Valencia

## Índice

Contexto	3
Solución Propuesta y/o Factor Diferenciador	3
Objetivos del Proyecto	4
1. Objetivo General	4
2. Objetivos Específicos	4
Alcances del Proyecto	4
1. Funcionalidades Incluidas	4
2. Funcionalidades Excluidas	6
Relación con el Perfil de Egreso y los Intereses Profesionales	7
1. Intereses Profesionales	7
Requisitos del Proyecto	7
1. Requisitos Funcionales	7
2. Requisitos No Funcionales	9
Diagrama de Base de Datos	10
1. Diccionario de Datos	10
1.1. Tablas de Ubicación e Institución	10
1.2. Tablas de Contenido Académico	11
1.3. Tablas de Usuarios y Gamificación	12
1.4. Tablas de Carga y Auditoría	13
Metodología	14
Plan de Trabajo	15
Sprint 1: Pre-desarrollo (Del 16 al 30 de agosto)	15
Sprint 2: Base Tecnológica (Del 31 de agosto al 14 de septiembre)	16
Sprint 3: Integración con IA (Del 15 al 29 de septiembre)	16
Sprint 4: Lógica de Gamificación (Del 30 de septiembre al 13 de octubre)	17
Sprint 5: Frontend y Conexión (Del 14 al 27 de octubre)	18
Sprint 8 : Cierre de Proyecto (Del 25 de noviembre al 15 de diciembre)	19
Factibilidad del Proyecto	20
Arquitectura	21
Persistencia y Almacenamiento de Datos	21
Microservicios Principales	22
Tecnologías	23
1. Tecnologías a Utilizar	23
Cloud	23
1. Configuración de Seguridad	24
Conclusión	25

## Contexto

El presente documento describe la solución técnica y las especificaciones de un proyecto de desarrollo diseñado para mejorar el proceso de estudio de los alumnos del Duoc UC. Actualmente, el aprendizaje de contenidos teóricos en la educación superior a menudo se ve limitado por métodos de estudio pasivos y poco atractivos, como la simple lectura de textos o la memorización. Esta realidad genera una baja motivación en los estudiantes y dificulta la retención efectiva del conocimiento.

La necesidad que se identificó es transformar este proceso de aprendizaje en una experiencia interactiva y lúdica que no solo capte la atención del estudiante, sino que también adapte el contenido a sus necesidades individuales, fomentando un estudio más eficiente y dirigido. La solución propuesta busca resolver este problema aplicando tecnologías de vanguardia para convertir un proceso tradicional en una experiencia de aprendizaje gamificada y personalizada.

## Solución Propuesta y/o Factor Diferenciador

La solución propuesta se diferencia de las herramientas de estudio tradicionales al integrar un sistema de gamificación adaptativa impulsado por inteligencia artificial. A diferencia de las plataformas de cuestionarios genéricas, este proyecto utiliza la API de Google Gemini para analizar el rendimiento del usuario y generar preguntas y desafíos dinámicos que se enfocan en las áreas de debilidad del estudiante.

El factor diferenciador clave de nuestra aplicación es la personalización inteligente del aprendizaje. El sistema registra cada error y acierto del estudiante, creando un perfil de sus fortalezas y debilidades. Con base en este análisis, la inteligencia artificial no solo genera contenido nuevo, sino que también prioriza el refuerzo de los temas donde el usuario comete más errores, convirtiendo las deficiencias en oportunidades de mejora. Esto asegura que el tiempo de estudio sea eficiente y dirigido, transformando la experiencia de aprendizaje en un proceso continuo de adaptación y crecimiento, lo cual cumple con los criterios solicitados y ofrece una solución innovadora y efectiva a la problemática planteada.

## Objetivos del Proyecto

### 1. Objetivo General

Desarrollar una aplicación de aprendizaje adaptativo, basada en gamificación e inteligencia artificial, para los estudiantes del Duoc UC, con el fin de mejorar su retención de conocimiento en materias teóricas.

### 2. Objetivos Específicos

- **Gestión de Usuarios:** Implementar un sistema robusto de gestión de usuarios con tres roles diferenciados (Estudiante, Profesor, Administrador), asegurando su autenticación segura y la gestión de permisos.
- **Integración de Contenido Inteligente:** Diseñar y construir un microservicio (IA Service) que se integre con la API de Google Gemini para generar preguntas dinámicas y personalizadas, analizando el rendimiento del estudiante para identificar áreas de debilidad.
- **Implementación de Gamificación:** Desarrollar un módulo de juego que incorpore mecánicas como puntajes, rankings y desafíos, incentivando la participación activa del estudiante y proporcionando un entorno de aprendizaje lúdico y motivador.
- **Construcción de una Arquitectura Escalable:** Implementar una arquitectura de microservicios sobre Google Cloud Platform (GCP) con una base de datos relacional (PostgreSQL) y una no relacional, garantizando la escalabilidad y el mantenimiento a largo plazo de la solución.

## Alcances del Proyecto

### 1. Funcionalidades Incluidas

- **Gestión de Perfiles por Rol:** La aplicación incluirá tres tipos de roles de usuario: Estudiante, Profesor, y Administrador. Cada rol tendrá un conjunto de permisos y funcionalidades específicas.
- **Módulo de Estudiante:** Este perfil tendrá acceso exclusivo al módulo de juego. Los estudiantes podrán seleccionar materias precargadas según su carrera, jugar, y

visualizar su progreso. Podrán editar aspectos básicos de su perfil, como su foto o ícono.

- **Módulo de Profesor:** Los profesores tendrán acceso a un módulo de visualización en el cual podrán ver los rankings y el progreso de los estudiantes de sus cursos. Además, tendrán la capacidad de cargar contenido teórico (documentos, extractos, etc.) por materia. Este contenido es la fuente que posteriormente la Inteligencia Artificial (IA) analizará para generar preguntas dinámicas. Adicionalmente, el profesor podrá crear, editar y eliminar preguntas de forma manual en el banco de contenido teórico de las materias que tenga asignadas. Finalmente, tendrán la capacidad de cargar masivamente alumnos a las asignaturas que tengan a su cargo, facilitando la inscripción y la administración de grandes grupos de alumnos.
- **Módulo de Administrador:** Este rol tendrá los privilegios más altos, con acceso a todas las funcionalidades del sistema. Podrá gestionar usuarios, contenido, y configuraciones generales de la aplicación.
- **Módulo de Gamificación:** Implementación de mecánicas de juego como niveles, desafíos, puntajes, y un sistema de logros para mantener a los estudiantes motivados con el aprendizaje. El sistema registrará los errores de los usuarios, lo que permitirá generar cuestionarios de refuerzo adaptativos, enfocados en las áreas de debilidad detectadas.
- **Banco de Contenido Teórico:** La aplicación contendrá un banco de preguntas y respuestas relacionadas con las materias teóricas de la carrera, organizadas de manera jerárquica: una carrera tiene muchas asignaturas, y cada asignatura contiene muchas preguntas. Esto permite un manejo y una categorización del contenido clara.
- **Integración Inteligente con Google Gemini:** Se desarrollará un microservicio dedicado a la comunicación con la API de Gemini. Esta integración no solo generará preguntas dinámicas y personalizadas, sino que también analizará el historial de respuestas del usuario para identificar áreas de debilidad. La inteligencia artificial fomentará el contenido en el que el usuario comete más errores, reforzando así el aprendizaje de manera dirigida y eficiente. El sistema cuenta con un módulo de inteligencia que, al recibir la solicitud de un nuevo cuestionario, procesa el historial de rendimiento del usuario. Este análisis permite identificar los temas o conceptos en los que el estudiante ha tenido más dificultades. Con esta información, el sistema genera preguntas de refuerzo, asegurando que el contenido se adapte de forma inteligente y dirigida a las necesidades de aprendizaje del usuario. Este enfoque asegura un proceso de aprendizaje adaptativo, donde la inteligencia artificial utiliza el rendimiento del estudiante para dirigir y reforzar el conocimiento de manera personalizada.

- **Sistemas de Estadísticas y Analíticas:** Se registrarán las interacciones y el progreso del usuario en una base de datos no relacional para futuras analíticas, permitiendo comprender mejor los patrones de aprendizaje.
- **Versión web:** Se incluirá el desarrollo de una aplicación web cuyo alcance estará estrictamente limitado a las funcionalidades de gestión y administración de contenido, siendo la plataforma móvil la única destinada a la experiencia de juego. El administrador utilizará la versión web para gestionar usuarios, roles y configuraciones generales del sistema. Mientras que el perfil de profesor podrá cargar contenido teórico masivo, gestionar el banco de preguntas y cargar alumnos de manera masiva a las asignaturas asignadas.
- **Interfaz de Usuario (UI) y Experiencia de Usuario (UX) intuitiva:** El diseño se centrará en una interfaz limpia y una navegación sencilla, asegurando una experiencia de juego fluida y accesible.

## **2. Funcionalidades Excluidas**

- **Desarrollo para iOS:** El proyecto se enfocará exclusivamente en la plataforma Android utilizando lenguajes nativos. No se contempla una versión nativa para el sistema operativo iOS en esta etapa.
- **Versión de Escritorio:** No se desarrollarán versiones de la aplicación para programas de escritorio.
- **Integración con el Sistema Académico del Duoc UC:** El juego funcionará de manera independiente del sistema de registro y notas oficial de la institución.
- **Modo Multijugador en Tiempo Real:** Las funcionalidades multijugador se limitarán a tablas de clasificación (rankings) y desafíos asincrónicos, para mantener el foco en el aprendizaje individual.

## **Relación con el Perfil de Egreso y los Intereses Profesionales**

Este proyecto integra y aplica directamente las competencias del perfil de egreso de la carrera, demostrando la capacidad del equipo para enfrentar un desafío tecnológico real. La solución aborda de forma integral la creación de una propuesta informática, la cual se

desarrolla utilizando la metodología ágil Scrum para sistematizar el proceso. Esto se evidencia en la construcción de un modelo arquitectónico de microservicios y un modelo de datos escalable, que en conjunto permiten implementar una solución integral que automatiza y optimiza los procesos de aprendizaje.

El proyecto también demuestra la capacidad de programar consultas y rutinas para la manipulación de información, construir programas con buenas prácticas de codificación y realizar pruebas de calidad en cada iteración. Finalmente, se abordan las vulnerabilidades sistémicas en la sección de seguridad, demostrando un enfoque profesional y completo. La participación en este proyecto se alinea directamente con nuestros intereses de forjar una carrera en el desarrollo de software, la gestión de la nube y la inteligencia artificial, aplicando la capacidad de generar ideas innovadoras y resolver problemas de forma proactiva.

## 1. Intereses Profesionales

La participación en este proyecto permite al equipo adquirir experiencia práctica en áreas de alta demanda en el mercado laboral, como el desarrollo de software y arquitecturas de microservicios, el despliegue en la nube con GCP, la integración de inteligencia artificial y el desarrollo de bases de datos escalables. Esto se alinea directamente con los intereses de forjar una carrera como desarrollador de software con especialización en tecnologías de backend e IA.

# Requisitos del Proyecto

## 1. Requisitos Funcionales

El sistema debe permitir que los usuarios se autenticuen de manera segura, asignando roles con permisos específicos:

- **Requisitos del Estudiante:**

- El estudiante debe poder acceder al juego y seleccionar su carrera y materia.
- El sistema debe analizar el historial de respuestas del estudiante para identificar los temas o conceptos en los que comete más errores.
- El juego debe ajustar dinámicamente el contenido, priorizando la presentación de preguntas y desafíos sobre los temas que el estudiante necesita reforzar. Esto asegura un aprendizaje personalizado y enfocado en sus debilidades.

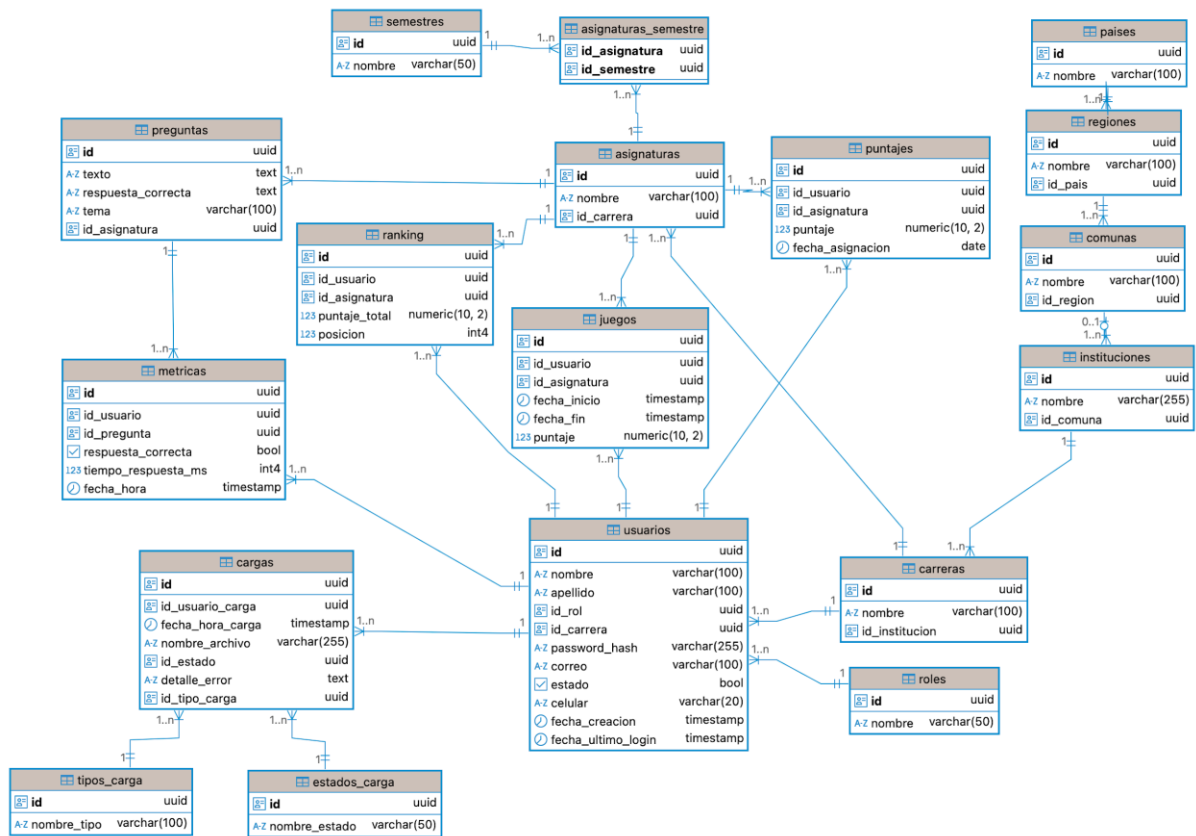
- El sistema debe mostrar al estudiante su puntaje, progreso individual y las áreas específicas de mejora.
- El estudiante debe poder editar su foto o ícono de perfil.
- **Requisitos del Profesor:**
  - El profesor debe tener acceso únicamente a los datos de los estudiantes y el contenido de las asignaturas que le han sido asignadas por el Administrador.
  - El sistema debe permitir al profesor ver los rankings y acceder a estadísticas de rendimiento detalladas del grupo y de los estudiantes individuales en sus cursos asignados.
  - El profesor debe tener la funcionalidad para cargar contenido teórico (ej. archivos PDF o texto) por materia. Este contenido debe ser almacenado para que el Módulo de IA lo analice posteriormente.
  - El profesor debe tener acceso a un módulo de administración para crear, editar o eliminar preguntas de forma manual en el banco de contenido teórico de las materias bajo su cargo.
  - El profesor debe tener la capacidad de cargar masivamente alumnos a las asignaturas que tenga asignadas, utilizando un archivo, lo que facilitará el proceso de inscripción
- **Requisitos del Administrador:**
  - El administrador debe tener privilegios para gestionar todos los aspectos de la aplicación, incluyendo usuarios, roles, materias y el contenido general.
- **Requisitos Generales:**
  - El sistema debe permitir al usuario iniciar una sesión de juego o desafío en cualquier momento.
  - El sistema debe presentar preguntas teóricas de forma aleatoria o temática.
  - El sistema debe validar la respuesta del usuario y proporcionar retroalimentación inmediata, utilizando la API de Google Gemini para generar respuestas dinámicas y personalizadas.
  - El sistema debe registrar y actualizar el puntaje y el progreso del usuario después de cada sesión de juego.
  - El sistema debe mostrar un ranking general y por curso de los puntajes más altos entre los usuarios.

## 2. Requisitos No Funcionales



- Rendimiento: El tiempo de respuesta de la aplicación para las interacciones del usuario no debe exceder los 2 segundos, y el tiempo de carga de las preguntas de Gemini no debe superar los 5 segundos.
- Seguridad: Las contraseñas de los usuarios deben ser encriptadas. La comunicación con los microservicios debe realizarse a través de canales seguros (HTTPS).
- Usabilidad: La interfaz de usuario debe ser intuitiva y fácil de usar para un público estudiantil.
- Escalabilidad: La arquitectura de microservicios debe permitir un crecimiento futuro en el número de usuarios y la adición de nuevos módulos sin comprometer el rendimiento general. %
- Disponibilidad: El sistema debe tener una disponibilidad del 99% para garantizar que los usuarios puedan estudiar en cualquier momento.

## Diagrama de Base de Datos



[capstonebdd- v2.png](#)

## 1. Diccionario de Datos

### 1.1. Tablas de Ubicación e Institución

- **países:** Almacena los países. Su objetivo es proporcionar una referencia de ubicación global para las instituciones.
  - id: Identificador único del país.
  - nombre: Nombre del país.
- **regiones:** Contiene las regiones de cada país. Ayuda a organizar la ubicación de manera jerárquica.
  - id: Identificador único de la región.
  - nombre: Nombre de la región.
  - id\_pais: Clave foránea que la relaciona con la tabla países.
- **comunas:** Almacena las comunas de cada región. Proporciona el nivel más específico de ubicación.
  - id: Identificador único de la comuna.

- nombre: Nombre de la comuna.
- id\_region: Clave foránea que la relaciona con la tabla regiones.
- **instituciones:** Almacena las instituciones educativas como Duoc UC.
  - id: Identificador único de la institución.
  - nombre: Nombre de la institución.
  - id\_comuna: Clave foránea que la relaciona con la tabla comunas.

## 1.2. Tablas de Contenido Académico

- **roles:** Contiene los roles de usuario (Estudiante, Profesor, Administrador). Es fundamental para la gestión de permisos en la aplicación.
  - id: Identificador único del rol.
  - nombre\_rol: Nombre del rol.
- **carreras:** Almacena las carreras que ofrece la institución.
  - id: Identificador único de la carrera.
  - nombre: Nombre de la carrera.
  - id\_institucion: Clave foránea que la relaciona con la tabla instituciones.
- **semestres:** Almacena los semestres académicos.
  - id: Identificador único del semestre.
  - nombre: Nombre del semestre (ej. "Semestre 1", "Semestre de Verano").
- **asignaturas:** Contiene las materias académicas. Cada asignatura pertenece a una carrera.
  - id: Identificador único de la asignatura.
  - nombre: Nombre de la asignatura.
  - id\_carrera: Clave foránea que la relaciona con la tabla carreras.
- **asignaturas\_semestre:** Esta es una tabla intermedia que resuelve la relación de muchos a muchos (N:M) entre asignaturas y semestres, permitiendo que una asignatura se imparta en varios semestres y viceversa.
  - id\_asignatura: Clave foránea que la relaciona con la tabla asignaturas.
  - id\_semestre: Clave foránea que la relaciona con la tabla semestres.

## 1.3. Tablas de Usuarios y Gamificación

- **usuarios:** Almacena la información de los usuarios. Se conecta directamente a carreras para indicar a qué carrera pertenece cada usuario.
  - id: Identificador único del usuario.

- nombre: Nombre del usuario.
- apellido: Apellido del usuario.
- rut: Rut del usuario.
- id\_rol: Clave foránea que la relaciona con la tabla roles.
- id\_carrera: Clave foránea que la relaciona con la tabla carreras.
- password\_hash: Contraseña encriptada para la autenticación.
- correo: Correo electrónico, usado como identificador único.
- estado: Indica si la cuenta está activa o no.
- celular: Número de celular del usuario.
- fecha\_creacion: Fecha en que se creó la cuenta.
- fecha\_ultimo\_login: Fecha del último acceso del usuario.
- **preguntas:** El banco de contenido teórico.
  - id: Identificador único de la pregunta.
  - texto: El enunciado de la pregunta.
  - respuesta\_correcta: La respuesta correcta de la pregunta.
  - tema: Tema de la pregunta (ej. "JOINS" en SQL).
  - id\_asignatura: Clave foránea que la relaciona con la tabla asignaturas.
- **juegos:** Registra cada sesión de juego. Es el historial de partidas del usuario.
  - id: Identificador único del juego/sesión.
  - id\_usuario: Clave foránea que la relaciona con la tabla usuarios.
  - id\_asignatura: Clave foránea que la relaciona con la tabla asignaturas.
  - fecha\_inicio: Momento en que inició el juego.
  - fecha\_fin: Momento en que terminó el juego.
  - puntaje: Puntaje final de la sesión de juego.
- **metricas:** Registra cada respuesta individual para analizar el rendimiento.
  - id: Identificador único de la métrica.
  - id\_usuario: Clave foránea que la relaciona con la tabla usuarios.
  - id\_pregunta: Clave foránea que la relaciona con la tabla preguntas.
  - respuesta\_correcta: Valor booleano (TRUE/FALSE) que indica si la respuesta fue correcta.
  - tiempo\_respuesta\_ms: Tiempo que el usuario tardó en responder.
  - fecha\_hora: Momento en que se registró la respuesta.
- **puntajes:** Almacena el puntaje acumulado de un usuario en una asignatura específica.
  - id: Identificador único del puntaje.
  - id\_usuario: Clave foránea que la relaciona con la tabla usuarios.

- id\_asignatura: Clave foránea que la relaciona con la tabla asignaturas.
- puntaje: Puntaje total acumulado.
- fecha\_asignacion: Fecha en que se registró o actualizó el puntaje.
- **ranking:** Una tabla de apoyo para optimizar el rendimiento de la visualización del ranking.
  - id: Identificador único del ranking.
  - id\_usuario: Clave foránea que la relaciona con la tabla usuarios.
  - id\_asignatura: Clave foránea que la relaciona con la tabla asignaturas.
  - puntaje\_total: Puntaje total del usuario en la asignatura.
  - posicion: Posición del usuario en el ranking.

#### 1.4. Tablas de Carga y Auditoría

- **estados\_carga:** Contiene los estados de una carga masiva (ej. "Completado", "Pendiente", "Fallido").
  - id: Identificador único del estado.
  - nombre\_estado: Nombre del estado de la carga.
- **tipos\_carga:** Almacena los tipos de cargas (ej. "Carga de Usuarios", "Carga de Contenido").
  - id: Identificador único del tipo.
  - nombre\_tipo: Nombre del tipo de carga.
- **cargas:** Registra el historial de todas las cargas masivas.
  - id: Identificador único de la carga.
  - id\_usuario\_carga: Clave foránea que la relaciona con la tabla usuarios.
  - fecha\_hora\_carga: Fecha y hora de la carga.
  - nombre\_archivo: Nombre del archivo original.
  - id\_estado: Clave foránea que la relaciona con la tabla estados\_carga.
  - detalle\_error: Detalles del error en caso de que la carga falle.
  - id\_tipo\_carga: Clave foránea que la relaciona con la tabla tipos\_carga.

## Metodología

Para la gestión y desarrollo de este proyecto, se ha seleccionado la metodología ágil Scrum. Esta elección se fundamenta en su capacidad para gestionar proyectos complejos de manera eficiente y flexible, lo que es ideal para un proyecto de desarrollo de software.

Las principales razones para la elección de Scrum son:

- **Adaptabilidad:** Permite reaccionar de manera ágil a los cambios en los requisitos del proyecto o a los hallazgos inesperados durante el desarrollo, asegurando que el producto final siempre cumpla con las necesidades del cliente.
- **Entrega de Valor Temprana y Continua:** El trabajo se divide en "sprints" cortos, lo que permite entregar funcionalidades operativas de forma incremental. Esto nos permitirá mostrar progreso de manera constante y obtener retroalimentación temprana del profesor.
- **Gestión del Riesgo:** Al abordar el desarrollo en iteraciones cortas, los riesgos se identifican y gestionan de forma proactiva, evitando que se conviertan en problemas mayores.
- **Transparencia:** Con la planificación y revisión de cada sprint, todos los involucrados en el proyecto tienen una visibilidad completa del estado del proyecto y de los avances logrados.

## Plan de Trabajo

El cronograma de trabajo para las siguientes etapas se detalla a continuación. Cada sprint tendrá una duración de dos semanas, lo que nos permitirá un desarrollo iterativo e incremental. La siguiente planificación está alineada con la Carta Gantt del proyecto.

### **Sprint 1: Pre-desarrollo (Del 16 al 30 de agosto)**

- **Objetivo:** Establecer las bases teóricas y de diseño del proyecto para asegurar un inicio de desarrollo sólido y bien planificado.
- **Actividades Clave:**
  - Definición de los alcances del proyecto.
  - Identificación de los requisitos funcionales y no funcionales.
  - Identificación de la problemática y justificación de la relevancia del proyecto.
  - Elección y justificación de la metodología de trabajo (Scrum).

- Preparación de los ambientes de desarrollo y de la infraestructura en la nube.
  - Diseño del modelo de la arquitectura.
  - Elaboración del diagrama de base de datos.
  - Selección y validación de las tecnologías a utilizar.
- 
- Facilitador: El *expertise* del equipo en Java y Spring Boot nos permite diseñar una arquitectura de microservicios con convenciones estandarizadas desde el inicio.
  - Obstaculizador: Ambigüedad o alcance no definido de la funcionalidad de la IA, lo cual podría forzar rediseños en la arquitectura
  - Mitigación: Priorizar la Prueba de Concepto del IA Service en el diseño. Se definirá un contrato *mock* de API entre el IA Service y el Content Service para que el desarrollo del *backend* pueda avanzar.

## **Sprint 2: Base Tecnológica (Del 31 de agosto al 14 de septiembre)**

- Objetivo: Configurar el entorno de desarrollo e implementar los servicios de autenticación y gestión de usuarios.
  - Backlog del Sprint:
    - Configuración y despliegue del servidor en Google Cloud Platform (GCP).
    - Desarrollo y despliegue del Auth Service.
    - Instalación y configuración de la base de datos en PostgreSQL
    - Pruebas unitarias de los servicios de autenticación.
    - Pruebas de integración entre el Auth Service y la base de datos.
- 
- Facilitador: Dominio de la Nube, experiencia previa del equipo en la configuración de la infraestructura en GCP y la gestión de la base de datos PostgreSQL.
  - Obstaculizador: Problemas de conectividad segura entre los microservicios (a través del BFF) y la instancia de PostgreSQL, exponiendo datos críticos.

- Mitigación: Configuración estricta de Firewall Rules de GCP. Se asegurará que la conexión a PostgreSQL sea solo interna, deshabilitando cualquier acceso público directo.

### **Sprint 3: Integración con IA (Del 15 al 29 de septiembre)**

- Objetivo: Desarrollar el servicio clave para la integración con la IA y la generación de contenido dinámico.
- Backlog del Sprint:
  - Desarrollo del IA Service y su integración inicial con la API de Google Gemini para la generación de preguntas dinámicas.
  - Implementación de la lógica para la generación de preguntas de refuerzo
  - Pruebas unitarias del IA Service y el User Service.
  - Pruebas de integración de la comunicación entre servicios.
- Facilitador: El IA Service es un microservicio independiente, permitiendo que el desarrollo del Auth y Content Service avance en paralelo sin dependencias críticas.
- Obstaculizador: Riesgo de Interrupción por Cuotas: Exceder los límites de la versión gratuita de la API de Google Gemini, lo que resultaría en una interrupción total del servicio de preguntas dinámicas y afectaría la experiencia de juego. Además, la latencia de la versión gratuita podría ser inconsistente.
- Mitigación:
  1. Implementar un mecanismo de *fallback* en el IA Service que, al fallar la conexión con Gemini (código de error 429 - Cuota Excedida), entregue una pregunta del banco de contenido estático.
  2. Utilizar mecanismos de caché para las preguntas generadas por la IA, reduciendo la necesidad de llamar a la API en partidas consecutivas.



#### **Sprint 4: Lógica de Gamificación (Del 30 de septiembre al 13 de octubre)**

- **Objetivo:** Implementar el motor de gamificación y la lógica de las partidas.
- **Backlog del Sprint:**
  - Desarrollo del Content Service para la gestión de asignaturas, materias y preguntas.
  - Implementación de la lógica de los juegos, incluyendo la puntuación y el seguimiento del progreso.
  - Creación de la lógica para la gestión de rankings y logros de usuario.
- **Facilitador:** La lógica se desarrolla en Spring Boot, que ofrece frameworks robustos para la implementación de lógica de negocio (puntajes, logros y rankings).
- **Obstaculizador:** Errores en las fórmulas de cálculo del progreso adaptativo o en la jerarquía del *ranking*, afectando la experiencia de juego y la credibilidad del sistema.
- **Mitigación:** Pruebas Unitarias exhaustivas y detalladas en la capa de servicio para validar todas las reglas de negocio. Se realizarán revisiones de código cruzadas.

#### **Sprint 5: Frontend y Conexión (Del 14 al 27 de octubre)**

- **Objetivo:** Iniciar el desarrollo del frontend de la aplicación móvil y asegurar la conexión con los servicios de backend.
- **Backlog del Sprint:**
  - Desarrollo de las pantallas principales de la aplicación móvil.
  - Conexión de la aplicación móvil con el Auth & User Service.
  - Pruebas de conectividad y funcionalidad del backend desde el frontend.
- **Facilitador:** El uso del BFF simplifica la lógica del *frontend*, ya que recibe datos agregados, reduciendo la complejidad de la conexión y las llamadas múltiples.
- **Obstaculizador:** Inconsistencia en la API expuesta por el BFF que rompan la comunicación con la aplicación móvil.

- Mitigación: Definición de Contratos de Datos inmutables entre el BFF y el Frontend, utilizando herramientas como Swagger/OpenAPI para documentar y estandarizar la API antes de iniciar la codificación.

### **Sprint 6: Desarrollo de la Experiencia (Del 28 de octubre al 10 de noviembre)**

- Objetivo: Completar el desarrollo del frontend para las funcionalidades de juego.
- Backlog del Sprint:
  - Desarrollo de las pantallas de juego y visualización de preguntas.
  - Implementación de la lógica de respuesta y la comunicación con el IA Service para obtener preguntas personalizadas.
  - Integración de los elementos de gamificación en la interfaz de usuario.
- Facilitador: El desarrollo modular nos permite enfocarnos al 100% en la interfaz de usuario y la experiencia de juego, ya que el *backend* ya está estable.
- Obstaculizador: **Problemas de rendimiento** en la aplicación móvil (lags o consumo excesivo de batería) debido a la manipulación de grandes paquetes de datos JSON.
- Mitigación: Optimizar el formato JSON que entrega el BFF para que sea lo más ligero posible, limitando solo a los datos estrictamente necesarios para la pantalla.

### **Sprint 7: Pruebas y Despliegue (Del 11 al 24 de noviembre)**

- Objetivo: Realizar pruebas de calidad y preparar el despliegue final.
- Backlog del Sprint:
  - Pruebas de calidad y usabilidad (UAT - User Acceptance Testing) con usuarios de prueba.
  - Pruebas de estrés y rendimiento en el backend.
  - Corrección de errores y bugs detectados en las pruebas.
  - Preparación del entorno de producción y del proceso de despliegue en GCP.
- Facilitador: Los Sprints anteriores incluyeron la creación de Pruebas Unitarias, simplificando la etapa de UAT.

- Obstaculizador: Descubrimiento de vulnerabilidades de seguridad no detectadas o fallos críticos en la carga masiva de alumnos o contenido.
- Mitigación: Pruebas de Penetración Básicas centradas en los *endpoints* críticos (Auth Service y Carga Masiva). Se asegurará que la validación de archivos de contenido sea estricta.

### **Sprint 8 : Cierre de Proyecto (Del 25 de noviembre al 15 de diciembre)**

- Objetivo: Finalizar el proyecto, entregar la documentación completa y preparar la presentación.
- Backlog del Sprint:
  - Redacción y consolidación del informe final de proyecto.
  - Elaboración de la presentación final y la demo de la aplicación.
  - Ajustes finales de despliegue para la presentación.
  - Preparación para el cierre formal del proyecto.
- Facilitador: El uso de la metodología Scrum garantiza que la documentación (Informe, Diagramas y Bitácoras) se ha generado de forma incremental a lo largo de los *Sprints*.
- Obstaculizador: Desviación de tiempo causada por ajustes de última hora en el código o fallas en la demostración final.
- Mitigación: Bloqueo del Código al final del Sprint 6. La última semana se dedicará exclusivamente a la documentación final, la preparación de la presentación y la realización de simulacros de la demo para asegurar su fluidez.

## **Factibilidad del Proyecto**

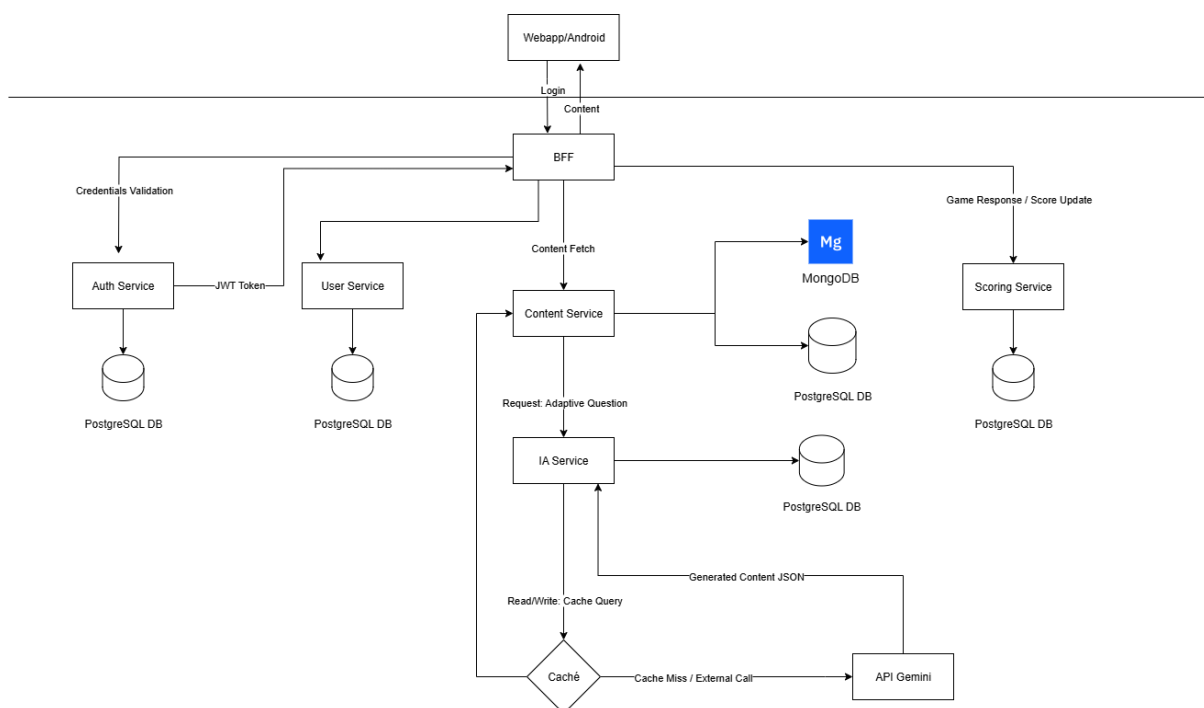
Este proyecto es factible de ser realizado, ya que se han tomado en cuenta los siguientes factores para garantizar su correcta ejecución.

- Alcance y Tiempo: El alcance del proyecto ha sido definido de manera precisa para ser completado en el plazo establecido. El plan de trabajo con la metodología Scrum, con sprints de dos semanas, nos permite monitorear el avance de manera constante y ajustar el plan si fuera necesario. El "Sprint 0" ya ha completado las fases de planificación y diseño, lo que nos da un inicio sólido en la fase de desarrollo.
- Recursos y Herramientas: La factibilidad del proyecto se sustenta en la capacidad técnica especializada del equipo. Poseemos conocimiento y experiencia previa en Java y Spring Boot. Esto garantiza que el *backend* será desarrollado con eficiencia, escalabilidad y siguiendo las mejores prácticas. Esta habilidad, junto con la elección

de tecnologías *open source* (Kotlin y PostgreSQL) y la utilización de la plataforma de nube Google Cloud Platform (GCP), asegura el acceso a todas las herramientas necesarias para el desarrollo y despliegue del proyecto sin incurrir en barreras de costos significativas.

- **Viabilidad de Mercado:** La propuesta de valor de este proyecto es altamente atractiva y factible, ya que no se han encontrado en el mercado local soluciones similares que integren de forma efectiva la gamificación con la inteligencia artificial para personalizar el aprendizaje. Esto representa una oportunidad única para desarrollar un producto innovador que satisface una necesidad real en el ámbito educativo.
- **Manejo de Riesgos:** La metodología Scrum nos permite identificar y mitigar riesgos de manera temprana. Por ejemplo, la integración con la API de Gemini, uno de los puntos más complejos, se ha planificado para el Sprint 2, lo que nos da suficiente tiempo para abordar cualquier dificultad técnica. Además, la modularidad de los microservicios asegura que un problema en un componente no detenga el desarrollo completo del sistema.

## Arquitectura



[arquitectura-v1.3.2.png](#)

El proyecto se desarrollará bajo una arquitectura de Microservicios, lo cual garantiza la escalabilidad horizontal y la modularidad de cada componente. El sistema se desplegará en la nube de Google Cloud Platform (GCP), utilizando el patrón Backend for Frontend (BFF) para optimizar el rendimiento de la aplicación móvil y la interfaz web.

El BFF actúa como el único punto de entrada para todos los clientes (aplicación móvil y versión web). Su función principal no es solo enrutar peticiones, sino también orquestar y consolidar la información proveniente de múltiples microservicios en una única respuesta eficiente. Este componente es esencial para la seguridad, ya que actúa como un perímetro que protege las claves privadas del sistema de ser expuestas al cliente final, mitigando

## Persistencia y Almacenamiento de Datos

El sistema utiliza dos tipos de almacenamiento, cada uno adaptado a un propósito específico para optimizar la carga de trabajo y el rendimiento:

- Base de Datos Transaccional (PostgreSQL): Esta es la base de datos principal y central del sistema. Se utiliza para almacenar datos críticos que requieren alta integridad referencial, incluyendo:
  - Información de usuarios y la asignación de roles.
  - Estructura definitiva del banco de contenido académico (carreras, asignaturas y las preguntas que el profesor gestiona mediante el CRUD).
  - Registros de la lógica de gamificación.
- Base de Datos No Relacional (MongoDB): Este almacenamiento se destina exclusivamente a la ingesta y el *staging* de datos voluminosos. Se utiliza solamente para recibir y almacenar el contenido teórico (archivos o texto) que el profesor carga de manera masiva. Una vez que este contenido es procesado por el IA Service y transformado en preguntas estructuradas, el contenido original puede ser gestionado, liberando esta base de datos de tareas transaccionales.

## Microservicios Principales

La lógica de negocio se divide en los siguientes microservicios independientes:

- Auth Service (Servicio de Autenticación): Es la autoridad de identidad del sistema. Se encarga de procesar el registro y el inicio de sesión, validando credenciales y generando el JSON Web Token (JWT), que se usa para la autorización del usuario.
- User Service (Servicio de Usuarios): Gestiona los perfiles de los usuarios y las asignaciones de roles (Estudiante, Profesor, Administrador), así como el mantenimiento del estado de cada perfil.
- Content Service (Servicio de Contenido): Maneja la fuente de verdad del contenido académico. Es responsable de la gestión CRUD del banco de preguntas estructuradas. Se comunica con MongoDB para la lectura de contenido bruto y con el IA Service para solicitar contenido adaptativo.
- IA Service (Servicio de Inteligencia Artificial): Actúa como un *proxy* seguro y es el motor de la personalización:
  - Recibe la solicitud Request: Adaptive Question desde el Content Service.
  - Gestiona el recurso mediante Read/Write: Cache Query, consultando primero el Caché para mitigar la latencia y los costos de la API externa.
  - Analiza el historial de errores del usuario para solicitar preguntas dinámicas a la API de Google Gemini y reforzar el aprendizaje.
- Scoring Service (Servicio de Puntuación y Ranking): Es el motor de cálculo de la lógica de gamificación. Se encarga de recibir las respuestas de los jugadores desde el BFF (Game Response / Score Update), calcular los puntajes y el progreso, y actualizar los registros de Rankings y Puntajes en la base de datos PostgreSQL.

## Tecnologías

Cada servicio se diseñará para ser independiente y se comunicará con otros a través de APIs bien definidas.

### 1. Tecnologías a Utilizar

- Lenguajes de Programación: Kotlin (con Java) para el desarrollo de la aplicación móvil nativa en Android.
- Base de Datos:
  - Relacional (Postgres): Para datos estructurados como perfiles de usuario y contenido de preguntas.
  - No Relacional (MongoDB): Para datos flexibles y dinámicos como estadísticas de juego y logs.

- Servicios de IA: Google Gemini API para la generación de contenido dinámico y retroalimentación inteligente.
- Servicios en la Nube y Despliegue: Se utilizarán plataformas de nube como Google Cloud Platform (GCP). Se empleará Docker para el empaquetado de cada microservicio.

## Cloud

El proyecto se sustenta en una arquitectura de microservicios desplegada en Google Cloud Platform (GCP). El backend principal se aloja en una instancia de Google Compute Engine (GCE), server-capstone-g3, que está configurada con 2 vCPUs y 1 GB de RAM, siendo adecuada para entornos de desarrollo y pruebas. Se ha configurado un servidor web Nginx para actuar como un proxy inverso para la aplicación de backend y servir contenido estático. La base de datos, PostgreSQL, se utiliza como la base de datos principal del sistema.

### 1. Configuración de Seguridad

La seguridad del sistema ha sido una prioridad, especialmente después de una reconfiguración completa para abordar vulnerabilidades iniciales. El equipo ha implementado medidas para dejar todo el sistema seguro en su configuración actual, gestionando la seguridad a través de las reglas de Firewall de GCP y una configuración estricta de los servicios.

- Acceso a la Base de Datos Restringido: La configuración inicial, que permitía conexiones remotas desde cualquier dirección IP, ha sido eliminada. Ahora, el acceso al puerto 5432 de PostgreSQL está restringido a un rango de IP específico para evitar conexiones no autorizadas.
- Autenticación Reforzada: El acceso SSH por contraseña ha sido deshabilitado, y ahora el acceso al servidor se realiza exclusivamente a través de claves SSH para garantizar una conexión segura. Para el acceso a la base de datos, se ha creado un nuevo usuario seguro (cvalencia) con una contraseña fuerte y aleatoria. La autenticación exige el método robusto scram-sha-256 para todas las conexiones.
- Recomendaciones para Futuro: Se ha definido un plan para el entorno de producción que garantiza una arquitectura aún más segura. El procesamiento de documentos con la API de Gemini se realizará en un servidor de backend que se comunicará con la

base de datos de forma local, evitando exponer las claves de la API. Se recomienda el despliegue de esta aplicación en Google Cloud Run para una mayor seguridad, escalabilidad y eficiencia de costos en un entorno de producción.

## Conclusión

Esta entrega marca la culminación formal de la fase de Planificación y Diseño del proyecto. Hemos establecido una arquitectura de software robusta basada en microservicios, la cual garantiza la escalabilidad y la modularidad necesarias para el sistema.

La implementación del patrón Backend for Frontend (BFF) es una decisión estratégica que no solo optimiza el rendimiento y la agregación de datos para la aplicación móvil, sino que también establece un perímetro de seguridad riguroso. Este perímetro es clave para proteger los secretos del sistema, como la API Key de Google Gemini. Asimismo, el sistema está diseñado para utilizar bases de datos especializadas: PostgreSQL para la integridad de los datos transaccionales y un almacenamiento NoSQL para la ingesta eficiente de contenido masivo.

Con el plan de trabajo validado y la Carta Gantt en ejecución bajo la metodología Scrum, el equipo se encuentra actualmente inmerso en la fase de desarrollo, aplicando nuestra experiencia en Java y Spring Boot para construir los microservicios. Confiamos en que la solidez de este diseño nos permitirá avanzar de manera eficiente hacia la materialización de esta solución educativa innovadora.