***Proyecto APT – Plataforma de aprendizaje con IA***

**Profesor:**

Arturo Guerra Castro

**Sección:**

003V

**Estudiante:**

Víctor Ayala Reyes

**Fecha:**

03/09/2025

# Índice

[1. Introducción 1](#_Toc202040877)

[2. Justificación de la Solución Big Data 1](#_Toc202040878)

[3. Selección y Justificación de la Arquitectura 2](#_Toc202040879)

[4. Ciclo de Vida del Dato en Streaming 4](#_Toc202040880)

[4.1 Conexión en Tiempo Real con la Fuente de Datos 4](#_Toc202040881)

[4.2 Procesamiento en Streaming con Dataflow 5](#_Toc202040882)

[4.3 Almacenamiento en el Data Lake 6](#_Toc202040883)

[4.4 Carga y Consulta en BigQuery 7](#_Toc202040884)

[4.5 Visualización Interactiva en Tiempo Real 8](#_Toc202040885)

[5. Anexo Técnico 9](#_Toc202040886)

[5.1 Control de Errores 9](#_Toc202040887)

[5.2 Control de Duplicidad de Datos 9](#_Toc202040888)

[5.3 Registro de Actividad 10](#_Toc202040889)

[5.4 Validación de Datos y Procesos 11](#_Toc202040890)

[10. Conclusiones 12](#_Toc202040891)

# 1. Descripción de proyecto APT

El proyecto consiste en el desarrollo de una plataforma digital de aprendizaje que unifica distintos métodos de estudio mediante inteligencia artificial. La solución incorpora seguimiento personalizado del progreso, permitiendo a cada usuario crear módulos de estudio y generar quizzes automáticos a partir del contenido que desea repasar.

Las evaluaciones generadas por IA refuerzan el aprendizaje y entregan métricas claras: tiempo de respuesta, porcentaje de aciertos, evolución por tema, frecuencia de estudio y otros indicadores de desempeño. El enfoque central del proyecto es coordinar efectivamente a un equipo de tres integrantes para entregar un producto funcional, priorizando la planificación, la comunicación y la calidad del resultado.

# 2. Relación del proyecto APT con las competencias del perfil de egreso

El proyecto permite evidenciar y desarrollar competencias clave del perfil de egreso:

* Gestión y trabajo en equipo (SCRUM): planificación iterativa, definición de roles, coordinación de tareas y mejora continua.
* Comunicación efectiva y documentación técnica: acuerdos de equipo, especificaciones de API, manual de usuario y bitácoras de decisiones.
* Desarrollo de aplicaciones backend: diseño e implementación de APIs REST con Java/Spring Boot y principios de arquitectura limpia.
* Modelamiento y administración de datos: diseño de esquema relacional en Oracle / PostgreSQL, consultas y optimización básica.
* Programación frontend y UX/UI: construcción de vistas en React, prototipado y validación de usabilidad.
* Infraestructura y despliegue en la nube: integración y publicación en GCP/Azure, manejo de variables de entorno y pipelines.
* Testing y aseguramiento de calidad (QA): pruebas unitarias, de integración y criterios de aceptación; enfoque preventivo de defectos.
* Innovación con IA aplicada: uso de NLP y generación de contenido para crear quizzes y visualizaciones pedagógicas.

# 3. Relación del proyecto con tus intereses profesionales

Mis principales intereses se centran en el desarrollo de software y la gestión de proyectos tecnológicos. Este proyecto es especialmente pertinente porque integra la gestión de personas dentro de un equipo de tecnología: alineación de objetivos, distribución de responsabilidades, seguimiento de avances y comunicación efectiva. Al combinar habilidades técnicas con liderazgo, el proyecto potencia tanto la calidad del producto como el crecimiento del equipo.

# 4. Argumento de factibilidad dentro de la asignatura

El proyecto es factible porque:

* **Alcance acotado y medible:** se prioriza un MVP con módulos esenciales (autenticación, creación de módulos, generación de quizzes y métricas básicas).
* **Recursos y stack conocidos:** Java/Spring Boot, Oracle/PostgreSQL, React y GCP/Azure son tecnologías abordables en el contexto de la asignatura.
* **Trabajo en equipo de tres integrantes:** permite dividir roles (gestión/PO, backend, frontend) y rotar responsabilidades para aprendizaje transversal.
* **Iteraciones cortas:** entregas parciales facilitan feedback temprano del docente y ajustes oportunos.
* **Riesgos controlados:** se contemplan prácticas de QA y criterios de “Definition of Done” para mantener la calidad.

# 5. Objetivos claros y coherentes

Los objetivos del proyecto están definidos para garantizar que el trabajo realizado sea alcanzable dentro de la asignatura y aporte un resultado concreto y de valor. Se busca avanzar desde la construcción de un MVP hasta su despliegue en la nube, siempre aplicando buenas prácticas de programación, QA y gestión. Los objetivos propuestos integran tanto el desarrollo técnico como la gestión de equipo, lo que permite cubrir las distintas dimensiones del perfil de egreso.

Objetivo 1 🡪 Entregar un MVP funcional que permita crear módulos de estudio, generar quizzes automáticos por IA y visualizar métricas básicas.

Objetivo 2 🡪 Implementar una API REST con autenticación y persistencia en Oracle/PostgreSQL, documentada (OpenAPI/Swagger).

Objetivo 3 🡪 Desarrollar una interfaz en React con flujo de usuario simple y prototipo validado de UX/UI.

Objetivo 4 🡪 Establecer prácticas de QA: ≥80% de cobertura de pruebas unitarias de backend, pruebas de integración de endpoints críticos y criterios de aceptación.

Objetivo 5 🡪 Desplegar el MVP en GCP/Azure con pipeline básico de CI/CD y monitoreo mínimo (logs y métricas de uptime).

Objetivo 6 🡪 Aplicar SCRUM con sprints, tableros de trabajo y retrospectivas, registrando métricas de proceso (burndown, throughput).

# 6. Propuesta metodológica de trabajo (SCRUM)

La metodología seleccionada para la ejecución del proyecto es **SCRUM**, dado que facilita la organización de tareas en equipos pequeños y fomenta la colaboración constante. Al trabajar con roles definidos, iteraciones cortas y reuniones periódicas, se asegura la transparencia del avance y la capacidad de adaptación a imprevistos. Además, SCRUM permite integrar prácticas de calidad desde el inicio, logrando un proceso más ordenado y con entregas continuas que generan valor en cada sprint.

**Roles (rotativos/semanales):**

* Product Owner (PO): prioriza backlog, define criterios de aceptación y valida incrementos.
* Scrum Master: facilita ceremonias, remueve impedimentos y vela por la mejora continua.
* Equipo de Desarrollo: implementación full-stack y QA.

**Ceremonias:**

* Sprint Planning (semanal): definición de objetivo del sprint y selección de historias.
* Daily (15 min): coordinación y actualización de bloqueos.
* Sprint Review: demostración del incremento al docente/pares.
* Retrospective: acciones concretas de mejora (máx. 3) para el siguiente sprint.

**Artefactos y herramientas:**

* Product Backlog y Sprint Backlog en Jira.

**Calidad (QA) integrada:**

* Pruebas unitarias (backend), pruebas de integración en endpoints críticos y testing exploratorio de UI.

# 7. Plan de trabajo para el proyecto APT

El plan de trabajo se organizará en sprints semanales, con actividades priorizadas y revisiones periódicas que permitan evaluar el progreso y realizar ajustes cuando sea necesario. Cada integrante del equipo asumirá responsabilidades definidas, promoviendo al mismo tiempo la colaboración y la rotación de tareas con el fin de favorecer un aprendizaje integral. El plan contempla etapas de análisis, desarrollo, pruebas y despliegue, asegurando que al finalizar el semestre se disponga de un producto funcional, validado y alineado con los objetivos del proyecto.

**1.- Desarrollo de software backend**

* Actividad/Tarea: Configuración del entorno y base de datos
* Descripción: Crear la base de datos PostgreSQL y configurar el backend en Spring Boot.
* Recursos: PostgreSQL, Spring Boot, GCP
* Duración: 4 semanas
* Responsable: Diego Allende
* Observaciones: Requiere coordinar endpoints iniciales con frontend.

**2.- Procesamiento de lenguaje natural (NLP)**

* Actividad/Tarea: Implementación de módulo IA
* Descripción: Desarrollar integración con modelos NLP para extraer conceptos clave de los textos.
* Recursos: GPT-4 API
* Duración: 4 semanas
* Responsable: Diego Allende
* Observaciones: Puede requerir pruebas adicionales para la coherencia pedagógica.

**3.- Desarrollo frontend**

* Actividad/Tarea: Diseño de interfaces principales
* Descripción: Construir pantallas para carga de material, quizzes y métricas.
* Recursos: React, CSS
* Duración: 3 semanas
* Responsable: Víctor Ayala
* Observaciones: Iterar diseño según retroalimentación.

**4.- Innovación y trabajo colaborativo**

* Actividad/Tarea: Integración backend–frontend
* Descripción: Conectar APIs REST del backend con interfaces del frontend.
* Recursos: Postman, GitHub, GCP
* Duración: 2 semanas
* Responsables: Diego & Víctor
* Observaciones: Depende de avances paralelos de ambos módulos.

**5.- QA y aseguramiento de calidad**

* Actividad/Tarea: Pruebas unitarias y de integración
* Descripción: Validar funcionamiento de módulos, coherencia de quizzes y métricas.
* Recursos: JUnit, Jest, Postman, QA plan
* Duración: 2 semanas
* Responsable: Greisy García
* Observaciones: Incluir feedback de pruebas con estudiantes.

**6.- Documentación y presentación**

* Actividad/Tarea: Elaboración de informe y PPT final
* Descripción: Documentar arquitectura, pruebas, resultados y preparar presentación.
* Recursos: Word, PowerPoint, GitHub Wiki
* Duración: 2 semanas
* Responsable: Greisy García
* Observaciones: Trabajo en paralelo a pruebas.

**7.- Gestión de proyecto (SCRUM)**

* Actividad/Tarea: Planificación y seguimiento
* Descripción: Reuniones semanales, revisión de backlog y retrospectivas.
* Recursos: Trello/Jira, GitHub, Google Meet.
* Duración: Durante todo el proyecto
* Responsables: Greisy (Scrum Master) + equipo
* Observaciones: Garantiza adaptación y control de avance.

# 8. Propuesta de evidencias del logro de actividades

Para demostrar el cumplimiento de los objetivos y la correcta ejecución del proyecto, se definieron diversas **evidencias objetivas** que respaldan cada etapa del trabajo. Estas evidencias incluyen artefactos técnicos (código, documentación, prototipos y pruebas), métricas de gestión (sprints, retrospectivas y tablero de tareas) y resultados funcionales (MVP desplegado en la nube). De esta forma, se asegura una trazabilidad clara entre lo planificado, lo ejecutado y lo entregado, mostrando el impacto del trabajo en equipo y la aplicación de las competencias desarrolladas en la carrera.

* Repositorio Git con historial de commits por integrante y pull requests revisados.
* Documentación técnica: diagrama de arquitectura, modelo de datos, endpoints (OpenAPI), guías de despliegue.
* Prototipos UX/UI: wireframes, mockups y evidencia de validaciones con usuarios/pares.
* Métricas de proceso: tablero de sprint, burndown, minutas de retrospectivas y acciones de mejora.
* Módulos funcionales: creación de módulos, generación de quizzes por IA y tablero de métricas.
* QA: reportes de cobertura, resultados de pruebas de integración y registro de defectos con su resolución.
* Despliegue en la nube: URL pública del MVP, logs de ejecución y evidencia de pipeline de CI/CD.
* Presentación/defensa final: demo del producto, relación con objetivos y lecciones aprendidas.