Autoevaluación Fase 1 – Definición Proyecto APT

Nombre: Nicolás Alexander Hinojosa Salazar  
Carrera: Ingeniería en Informática — Duoc UC  
Proyecto APT: Fundación Bienestar Permanente

# Abstract

El proyecto APT “Fundación Bienestar Permanente” propone el desarrollo de un ecosistema digital que facilite la generación y gestión de contenido audiovisual educativo mediante la integración de servicios de inteligencia artificial. La solución contempla un módulo de generación de contenido (guión, voz sintética y video automatizado) y una plataforma web compuesta por una landing pública y una intranet privada para la administración y distribución del material. El objetivo es reducir costos y tiempos de producción de materiales didácticos para la fundación, aumentar su visibilidad y permitir una difusión más efectiva de sus iniciativas. El desarrollo seguirá un enfoque ágil (Scrum) con entregas incrementales orientadas a un MVP viable y escalable.

# Descripción del Proyecto APT

El proyecto se compone de dos subsistemas principales:

1. Módulo de Generación de Contenido (IA): Orquestación de APIs de generación de texto, síntesis de voz y generación/edición de video para transformar ideas o breves instrucciones en guiones, narraciones y piezas audiovisuales respetando lineamientos de branding.
2. Plataforma Web e Intranet:
   * Página Web: Presentación institucional, campañas y material seleccionado.
   * Intranet privada: Panel de gestión para usuarios autorizados donde se sube, etiqueta, organiza, visualiza y descarga el contenido generado. Incluye control de accesos y trazabilidad de cambios.

La solución será modular, con enfoque en seguridad, escalabilidad y mantenibilidad.

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

El proyecto moviliza directamente las competencias del perfil de egreso, entre las que destaco:

* Construcción de modelos de datos: Diseño de modelo entidad-relación para contenidos, usuarios, metadatos y logs.
* Desarrollo de software y buenas prácticas: Implementación de backend, API REST y frontend, aplicando estándares de codificación y control de versiones.
* Gestión de proyectos informáticos: Planificación y control con Scrum (backlogs, sprints, revisiones).
* Pruebas y aseguramiento de calidad: Diseño y ejecución de pruebas unitarias, integración y aceptación.
* Seguridad y cumplimiento: Definición de control de acceso, políticas de protección de datos y revisión OWASP básica.

Estas actividades permiten demostrar capacidades tanto técnicas (programación, bases de datos) como transversales (gestión, comunicación con stakeholders).

# Relación del proyecto con mis intereses profesionales

Mis intereses se centran en arquitectura de software, desarrollo full-stack y aplicaciones de inteligencia artificial para resolver problemas reales. Este proyecto me permite:

* Aplicar conocimientos de modelado de datos y consultas avanzadas (mi área fuerte).
* Profundizar en integración de APIs de IA (área de alto interés: Deep Learning, Ciencia de Datos).
* Practicar la gestión de proyectos mediante metodologías ágiles, lo que coincide con mi interés en arquitectura y coordinación técnica.

Además, el impacto social del proyecto (trabajar con una fundación) refuerza mi motivación por desarrollar soluciones tecnológicas con propósito.

# Argumento de factibilidad dentro de la asignatura

El proyecto es factible considerando:

* Tiempo: El semestre permite organizar el trabajo en sprints cortos con entregas incrementales (MVP en la primera mitad del curso y mejoras en la segunda).
* Recursos técnicos: Herramientas y servicios accesibles (repositorios GitHub/GitLab, Figma, frameworks web, APIs de IA con planes gratuitos o académicos).
* Competencias del equipo: El equipo posee experiencia en bases de datos, desarrollo y AI (roles distribuidos).
* Riesgos y mitigación: Riesgo principal: gestión del tiempo y dependencia de APIs externas — mitigación: alcance acotado (MVP), pruebas locales/alternativas y planificación de backups.

Con una planificación rigurosa y control de alcance es viable entregar un MVP funcional y la documentación requerida para la evaluación.

# Objetivos

Objetivo General  
Desarrollar e implementar un ecosistema digital (módulo IA + plataforma web) que permita a Fundación Bienestar Permanente generar, gestionar y distribuir contenido audiovisual educativo de forma eficiente y consistente.

Objetivos Específicos

1. Levantar y priorizar requisitos con la fundación; definir criterios de aceptación.
2. Diseñar la arquitectura (frontend, backend, BD, integración IA) y estándares de seguridad.
3. Implementar un MVP que permita generar guiones y audio sintético, y gestionar contenidos básicos en la intranet.
4. Implementar pruebas unitarias, de integración y pruebas de aceptación con usuarios representativos.
5. Documentar la solución (manuales técnico y de usuario) y entregar plan de mantenimiento.

# Propuesta metodológica de trabajo

Se utilizará Scrum con las siguientes consideraciones prácticas:

* Duración de sprints: 2 semanas por sprint (ajustable).
* Roles: Product Owner (representante de la fundación / docente), Scrum Master (rotativo entre integrantes), Developers (yo y compañeros).
* Artefactos: Product Backlog (priorizado por valor), Sprint Backlog, Definition of Done (DoD) documentada, burndown charts.
* Eventos:
  + Sprint Planning: selección de user stories y tareas, estimación.
  + Daily Stand-up: 10–15 min diarias.
  + Sprint Review: demo al final de cada sprint con el cliente.
  + Sprint Retrospective: mejoras de proceso.
* Criterios de aceptación: cada historia incluye criterios de aceptación claros y pruebas asociadas.
* Control de calidad: integración continua (CI) con pipelines que ejecuten tests automáticos y despliegue a entorno de staging.

Este enfoque asegura adaptación rápida ante cambios y visibilidad constante para la fundación.

# Plan de trabajo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plan de Trabajo Proyecto APT | | | | | | |
| Competencia o unidades de competencias | Nombre de  Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable | Observaciones |
| *Levantamiento y toma de requerimientos* | *Reunión inicial con la fundación* | *Levantar información sobre necesidades, requerimientos y expectativas* | *Microsoft teams* | *1* | *CM* | *Puede requerir más de una sesión si el cliente ajusta requerimientos* |
| *Levantamiento y toma de requerimientos* | *Minutas y validación de requerimientos* | *Documentar acuerdos y validarlos con el cliente* | *Word/Docs* | *2* | *NH* | *Es posible que el cliente solicite cambios posteriores a la validación inicial; se recomienda mantener comunicación constante para evitar retrabajos* |
| *Gestión de proyectos informáticos* | *Definición del alcance y restricciones* | *Elaborar documento que establezca el alcance del sistema y sus limitaciones* | *Plantilla de alcance, Word/Docs* | *1* | *FS* | *Fundamental para evitar desviaciones posteriores* |
| *Gestión de proyectos informáticos* | *Elaboración del Product Backlog* | *Construcción del backlog con todas las funcionalidades* | *Trello/Jira* | *1* | *CM, NH, FS* | *El backlog puede cambiar en el transcurso del proyecto según nuevas prioridades; se debe mantener actualizado y validado con el equipo y cliente* |
| *Gestión de proyectos informáticos* | *Elaboración del Roadmap / Release Plan* | *Establecer entregas por sprint* | *Excel / Jira* | *2* | *CM, NH, FS* | *El cumplimiento de fechas dependerá de la correcta estimación de esfuerzos en cada sprint; se recomienda revisarlo y ajustarlo periódicamente* |
| *Construcción de modelo arquitectónico* | *Diseño de la arquitectura del sistema* | *Crear diagramas de despliegue, clases y casos de uso* | *Lucidchart* | *3* | *CM, NH, FS* | *Debe validarse con todo el equipo antes de continuar* |
| *Desarrollo de software* | *Diseño UI/UX* | *Elaboración de prototipos de pantallas (landing e intranet)* | *Figma, paleta de colores, branding* | *2* | *NH* | *Permite validación temprana con cliente* |
| *Diseño UI/UX* | *Estándares de diseño y desarrollo* | *Definir paleta, tipografía, convenciones de código* | *Docs/Markdown* | *2* | *CM* | *Una definición temprana de estándares facilita la consistencia, pero puede requerir ajustes si el cliente modifica lineamientos de branding* |
| *Desarrollo de software* | *Desarrollo de pagina web* | *Implementar página institucional con secciones de información, contacto y noticias* | *VSCode, GitHub* | *18* | *CM, NH, FS* | *Validar con cliente antes de integrar con intranet* |
| *Producción* | *Implementacion de pagina web* | *Desarrollo parte front end o vista del usuario* | *VSCode, GitHub* | *2* | *CM, NH, FS* | *Es la vista que tendrán usuarios visitantes de la web (importante definir diseño)* |
| *Construcción de modelos de datos* | *Modelado y creación de base de datos* | *Diseñar y crear base de datos para almacenar usuarios y contenidos generados* | *MySQL* | *2* | *CM, NH, FS* | *Requiere definir bien el modelo entidad-relación* |
| *Desarrollo de software* | *Desarrollo la intranet (login)* | *Crear funcionalidades de autenticación* | *VSCode, GitHub* | *3* | *CM, NH, FS* | *Es necesario validar cada avance con el cliente para asegurar que el diseño y contenido reflejen correctamente la identidad institucional* |
| *Desarrollo de software* | *Desarrollo de intranet (registro)* | *Crear funcionalidades CRUD de contenidos* | *VSCode, GitHub* | *4* | *CM, NH, FS* | *El módulo CRUD debe probarse en conjunto con la base de datos; podrían surgir ajustes en el modelado según necesidades reales del cliente* |
| *Desarrollo de software* | *Desarrollo de Intranet (Módulo para generar contenido)* | *Desarrollar sistema de creación con IA* | *VSCode, GitHub* | *6* | *CM, NH, FS* | *El módulo que genera contenidos debe consumir la api pero mostrarse siempre dentro de la misma web* |
| *Desarrollo de software* | *Desarrollo de intranet (Módulo de gestión de contenidos)* | *Desarrollar sistema de organización visual (tabla) de contenidos* | *VSCode, GitHub* | *3* | *CM, NH, FS* | *La gestión de contenidos se trabaja en tablas* |
| *Programación de rutinas* | *Integración con APIs de IA* | *Conectar la plataforma con APIs (GPT, ElevenLabs, Runway ML)* | *API keys, documentación oficial* | *11* | *CM, NH, FS* | *Riesgo de cambios en políticas de uso de APIs* |
| *Calidad de software* | *Elaboración del plan de pruebas* | *Definir pruebas unitarias, integración, aceptación* | *Word/Docs* | *3* | *CM, NH, FS* | *La cobertura de pruebas debe incluir tanto aspectos técnicos como de usabilidad; podría requerir ampliación si se incorporan nuevas funcionalidades* |
| *Calidad de software* | *Pruebas unitarias e integración* | *Elaborar casos de prueba y ejecutarlos* | *Postman* | *8* | *CM, NH, FS* | *Puede requerir ajustes iterativos en el código* |
| *Calidad de software* | *Informe de resultados de pruebas* | *Documentar resultados finales de QA* | *Word/Docs* | *3* | *CM* | *Es probable que se identifiquen defectos críticos que requieran correcciones; el informe debe actualizarse hasta la última iteración antes del despliegue* |
| *Seguridad informática* | *Implementación de medidas de seguridad* | *Aplicar autenticación, encriptación y controles de acceso* | *OWASP* | *3* | *CM, NH, FS* | *Crítico para resguardar datos de usuarios* |
| *Gestión de proyectos informáticos* | *Documentación final y manuales* | *Redactar manual de usuario, instalación y mantenimiento* | *Word/Docs, GitHub0* | *5* | *CM, NH, FS* | *Entregable obligatorio para cliente y evaluación* |

# Propuesta de evidencias y justificación

Evidencias de gestión y planificación

* Acta de constitución, Product Backlog, Roadmap / Release Plan, Sprint Backlogs, minutas de reunión.  
  *Justificación:* demuestran gestión, trazabilidad y priorización.

Evidencias de análisis y diseño

* Wireframes / prototipos (Figma), Diagramas UML (casos de uso, clases, secuencia), Modelo Entidad-Relación.  
  *Justificación:* muestran diseño validado antes de implementar.

Evidencias técnicas y de implementación

* Repositorio con código (GitHub/GitLab), ramas y tags; scripts de base de datos; pipelines CI/CD; contenedores (Docker).  
  *Justificación:* validan trabajo reproducible y control de versiones.

Evidencias de calidad

* Plan de pruebas, casos de prueba, reportes de ejecución, registro de incidencias (issues), resultados de pruebas de seguridad y checklist OWASP básico.  
  *Justificación:* certifican que el sistema cumple requisitos y es robusto.

Evidencias de entrega

* Demo funcional en entorno staging, manual de usuario, manual de instalación/configuración y plan de mantenimiento.  
  *Justificación:* permiten la aceptación por parte del cliente y la continuidad operativa.

Evidencias de valor / impacto

* Registro de métricas (número de contenidos generados, tiempo promedio de producción, uso por parte de la fundación), retroalimentación de usuarios y video demo.  
  *Justificación:* demuestran impacto y éxito de la solución.

# Aspectos formales y de autocontrol

En mi autoevaluación anterior identifiqué fortalezas en **Bases de Datos, modelado, consultas SQL, transformación de datos y gestión de proyectos**, y áreas a fortalecer en **pruebas de** certificación**, arquitectura avanzada e implementación de seguridad**. Planifico abordar esas brechas mediante: pair-programming con el equipo, sesiones de revisión de arquitectura, y la priorización de pruebas y checklist de seguridad desde los primeros sprints.

# Conclusión

La primera fase del proyecto APT me permitió alinear mis competencias académicas con un desafío social real. Confirmé que mis principales fortalezas se encuentran en el modelado de bases de datos, la transformación de información y la planificación de proyectos. Estas capacidades facilitaron la definición de una arquitectura viable y un roadmap realista para entregar un MVP dentro de los plazos establecidos por la asignatura. Asimismo, identifiqué áreas de mejora concretas en pruebas avanzadas, fortalecimiento de la seguridad e integración de servicios de inteligencia artificial de terceros. Para las siguientes fases me propongo dar mayor énfasis a la automatización de pruebas, la aplicación sistemática de listas de verificación de seguridad y la profundización en pruebas de integración, con el fin de elevar la calidad global del producto.