}

**Guía1. Definición Proyecto APT**

**Asignatura Capstone**

1. **PARTE I**

|  |
| --- |
| **1. Antecedentes Personales** |
| A continuación, se presenta una tabla en la que debes completar la información solicitada. |

|  |  |
| --- | --- |
| Nombre estudiante | * **Felipe Torres** * **Manuel Miqueles** * **Nicolas Fernández** * **Liam Ley** |
| Rut | * **19.406.513-2** * **20.224.356-8** * **21.193.631-2** * **22.606.024-3** |
| Carrera | **Ingeniería en Informática** |
| Sede | **Duoc UC Puente Alto** |

|  |
| --- |
| **2. Descripción Proyecto APT** |
| En la descripción debes señalar brevemente el nombre de tu proyecto APT y las competencias del perfil de egreso que vas a poner en práctica. Si en tu carrera están definidas las áreas de desempeño, también menciona a qué áreas de desempeño está vinculado el proyecto. |

|  |  |
| --- | --- |
| Nombre del proyecto | EncryptU |
| Área (s) de desempeño(s) | Programación de aplicaciones de escritorio, desarrollo web, seguridad informática. |
| Competencias | * Análisis y levantamiento de requerimientos de software. * Programación de soluciones seguras en entornos multiplataforma. * Desarrollo de base de datos y APIs REST. * Aseguramiento de la calidad y gestión de proyectos. |

|  |
| --- |
| **3. Fundamentación Proyecto APT** |
| A continuación, se presentan distintos campos que debes completar con la información solicitada. Esta sección busca que describas en detalle tu proyecto y justifiques su relevancia y pertinencia. |

|  |  |
| --- | --- |
| Relevancia del proyecto APT | Actualmente, la empresa Syntec se especializa en el desarrollo de soluciones tecnológicas para clientes institucionales y privados. En el contexto del proyecto EncryptU, se busca abordar una problemática creciente: la falta de herramientas accesibles y seguras para proteger información sensible, especialmente entre usuarios con baja experiencia tecnológica.  La selección del tema responde a la necesidad de crear una solución informática que permita el cifrado y gestión de contraseñas y datos personales mediante una aplicación de escritorio complementada por una plataforma web. Este enfoque es altamente pertinente en el campo laboral, ya que responde a desafíos reales en ciberseguridad, accesibilidad digital y protección de datos, áreas críticas en el desarrollo de software moderno.  Los principales beneficiarios del proyecto serán usuarios finales como adultos mayores, estudiantes y trabajadores que manejan información confidencial sin conocimientos técnicos avanzados. La solución propuesta busca democratizar el acceso a herramientas de encriptación, reduciendo riesgos de vulneración y mejorando la seguridad digital personal. |
| Descripción del Proyecto APT | El objetivo del proyecto EncryptU es desarrollar una aplicación de escritorio con arquitectura en capas que permita cifrar, descifrar y gestionar contraseñas de forma segura. Esta solución se complementa con una aplicación web secundaria para gestión de tickets y retroalimentación.  El sistema contará con autenticación segura, gestión de usuarios y roles, mantenedores de claves, reportes exportables y notificaciones automáticas. Además, se integrará mediante API REST para garantizar interoperabilidad entre ambas plataformas. |
| Pertinencia del proyecto con el perfil de egreso | Este proyecto se relaciona directamente con el perfil de egreso de la carrera de informática, ya que exige competencias como:   * Levantamiento y análisis de requerimientos * Diseño y desarrollo de sistemas computacionales * Implementación de soluciones tecnológicas seguras * Aseguramiento de calidad y pruebas funcionales * Gestión de proyectos informáticos bajo metodologías estructuradas   Estas competencias son esenciales para abordar la problemática planteada, permitiendo una ejecución técnica sólida y una documentación profesional del sistema. |
| Relación con los intereses profesionales | Nuestros intereses profesionales están enfocados en el desarrollo de soluciones informáticas que combinen seguridad, usabilidad y diseño funcional. El proyecto EncryptU refleja estos intereses al integrar tecnologías modernas como AES-256, Django, Oracle SQL y frameworks web, en un entorno de desarrollo realista.  La experiencia adquirida en este proyecto será clave para enfrentar los desafíos del mundo laboral, especialmente en áreas como ciberseguridad, desarrollo multiplataforma y experiencia de usuario. |
| Factibilidad de desarrollo del Proyecto APT | El proyecto se desarrollará entre agosto y diciembre de 2025, con una duración total de 85 días hábiles, distribuidos en fases de planificación, desarrollo, pruebas y cierre, según el cronograma Gantt.  Recursos disponibles:   1. Computadora personal para desarrollo y documentación 2. Visual Studio Code (HTML, CSS, JavaScript) 3. Framework Django 4.2.4 con Python 3.10.11, Bootstrap 5 y SweetAlert2 4. Base de datos Oracle SQL Developer 19c   No se identifican factores externos que dificulten el desarrollo, ya que el caso el equipo cuenta con experiencia previa en las herramientas seleccionadas. La planificación detallada y el enfoque modular del sistema aseguran la viabilidad del proyecto dentro del plazo establecido. |

1. **PARTE II**

|  |
| --- |
| **4. Objetivos** |
| En este apartado debes definir objetivos generales y específicos del Proyecto APT. Es importante aclarar que los objetivos se deben plantear en forma clara, concisa y sin dar mayores explicaciones, es decir, deben entenderse por sí solos. Se sugiere redactarlos utilizando un verbo en infinitivo, pues ello obliga a precisar acciones concretas. |

|  |  |
| --- | --- |
| Objetivo general | Desarrollar una aplicación de escritorio multiplataforma con soporte web que permita a usuarios no técnicos proteger su información mediante encriptación segura, simple y accesible. |
| Objetivos específicos | * Implementar autenticación y gestión de roles. * Desarrollar módulo de cifrado/descifrado con AES/Fernet. * Crear mantenedores de usuarios, claves y algoritmos. * Generar reportes exportables en PDF/XLS. * Integrar API REST para interoperabilidad con la app web secundaria. * Desarrollar aplicación web secundaria para gestión de feedback. * Garantizar seguridad de transporte y almacenamiento (TLS + encriptación). |

|  |
| --- |
| **5. Metodología** |
| En el siguiente apartado deberás describir la metodología, propia de tu disciplina, que utilizarás para resolver el proyecto APT antes descrito, incluyendo las etapas y métodos de trabajo. |

|  |
| --- |
| Descripción de la Metodología |
| Para abordar la problemática identificada —la dificultad que tienen adultos mayores y usuarios no técnicos para proteger su información sensible con herramientas de cifrado complejas—, el proyecto **EncryptU** se desarrollará bajo la **metodología en Cascada (Waterfall)**, la cual resulta adecuada dado que:   * El problema está claramente delimitado. * Se requiere un control riguroso de entregables en plazos establecidos. * La asignatura contempla fases de avance estructuradas.   **Fases de la metodología**   1. **Planificación y levantamiento de requerimientos**    * Actividades: elaboración del Acta de Constitución, levantamiento de requerimientos funcionales y no funcionales, definición de casos de uso.    * Producto: Documento de Requerimientos (ERS). 2. **Diseño**    * Actividades: creación de mockups, definición de la arquitectura del sistema (DAS), diseño del modelo de base de datos.    * Producto: Documento DAS, modelo ER. 3. **Desarrollo**    * Actividades: programación de la aplicación de escritorio (EncryptU) en Python y de la aplicación web secundaria con Flask y React, integración con base de datos y API REST.    * Producto: Aplicación de escritorio funcional + aplicación web secundaria. 4. **Pruebas y QA**    * Actividades: pruebas funcionales, pruebas de seguridad (cifrado AES/Fernet, autenticación), pruebas de usabilidad con usuarios representativos.    * Producto: Reportes de pruebas y validación de requerimientos. 5. **Implementación y Cierre**    * Actividades: elaboración de manual de usuario, presentación final, entrega de documentación y acta de cierre.    * Producto: Manual de usuario, informe final, entrega oficial del sistema.   **Definición de funciones, tareas y responsabilidades**  El equipo de desarrollo estará conformado por **5 roles principales**, cada uno con responsabilidades específicas:   * **Jefe de Proyecto (Felipe Torres)**   + Coordinar y supervisar todas las fases del proyecto.   + Gestionar la planificación, cronograma y control de costos.   + Validar entregables con el cliente y stakeholders.   + Liderar la presentación final y la documentación de cierre. * **Analista Programador (Liam Ley)**   + Desarrollar la aplicación de escritorio (EncryptU) en Python.   + Implementar el módulo de cifrado/descifrado con AES/Fernet.   + Asegurar la integración de los módulos internos de la aplicación. * **DBA – Administrador de Base de Datos (Manuel Miqueles)**   + Diseñar y modelar la base de datos en PostgreSQL/SQLite.   + Implementar scripts de tablas y consultas necesarias.   + Optimizar la seguridad y rendimiento del almacenamiento de datos. * **QA / Tester (Nicolás Fernández)**   + Elaborar plan de pruebas funcionales y no funcionales.   + Ejecutar pruebas de seguridad, usabilidad y rendimiento.   + Documentar incidencias y verificar su corrección. * **Diseñador UI/UX (José Rojas)**   + Crear mockups y flujos de interacción del sistema.   + Asegurar la usabilidad y accesibilidad de la aplicación para usuarios no técnicos.   + Diseñar interfaz gráfica amigable con íconos grandes, contrastes y textos claros.   **Enfoque en el problema identificado**  La metodología permitirá:   * **Controlar el cumplimiento del objetivo principal**: entregar una solución accesible, simple y segura. * **Dividir tareas según especialidad**: cada miembro aporta su conocimiento (programación, bases de datos, diseño, QA). * **Evitar ambigüedades**: al trabajar con fases cerradas y entregables claros, se asegura la trazabilidad del proyecto. * **Reducir riesgos**: al realizar pruebas de seguridad y usabilidad antes del cierre, se valida que la solución responde a la necesidad real del público objetivo (adultos mayores y usuarios no técnicos). |

|  |
| --- |
| **6. Evidencias** |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito sea visibilizar o documentar cómo se ha implementado el trabajo. |

|  |  |  |  |
| --- | --- | --- | --- |
| Tipo de evidencia | Nombre de la evidencia | Descripción | Justificación |
| Avance | Acta de Constitución | Documento inicial que define los objetivos, alcance, recursos, presupuesto, cronograma y riesgos del proyecto. | Establecer la base formal del proyecto, delimitando sus alcances y compromisos. |
| Avance | Planilla de Requerimientos | Hoja de cálculo con requisitos funcionales y no funcionales, priorización y responsables. | Facilitar la trazabilidad y control de requerimientos en todo el ciclo de vida. |
| Avance | Documento ERS (Especificación de Requerimientos de Software) | Documento que describe requerimientos funcionales, no funcionales, casos de uso y criterios de aceptación. | Garantizar que el sistema cumpla con las necesidades identificadas de los usuarios. |
| Avance | Documento de Caso de Uso extendido | Casos de uso detallados con actores, flujos alternativos, excepciones y reglas de negocio. | Asegurar la correcta interpretación de interacciones entre usuarios y sistema. |
| Avance | Carta Gantt / Roadmap | Cronograma del proyecto, en formato Gantt o roadmap según la metodología. | Planificar tiempos, dependencias y responsables de cada fase. |
| Avance | EDT – Costos | Estructura de desglose del trabajo con asignación de costos estimados. | Tener claridad de los recursos financieros y horas-hombre requeridas. |
| Avance | Documento de avance de sprint (si aplica ágil) | Registro de entregables y avances logrados en cada sprint. | Dar visibilidad periódica al progreso en metodologías ágiles. |
| Avance | Documento proceso de negocio (TO-BE) | Modelo del proceso de negocio esperado tras la implementación. | Definir cómo cambiarán los flujos actuales con el nuevo sistema. |
| Avance | Mockups del sistema completo | Prototipo visual de pantallas y flujos de interacción de la aplicación. | Validar con el cliente y usuarios la usabilidad antes del desarrollo. |
| Avance | Documento de Arquitectura (DAS) | Definición de arquitectura del sistema: capas, base de datos, APIs REST y componentes técnicos. | Asegurar escalabilidad, seguridad y buenas prácticas de desarrollo. |
| Avance | Modelo de Base de Datos (Scripts) | Scripts de creación de tablas. | Garantizar consistencia y seguridad en el almacenamiento de información. |
| Avance | Diccionario de Datos | Definición clara de cada campo, formato, unidad y origen de datos. | Evitar ambigüedad en la interpretación de la información del sistema. |
| Avance | Matriz RACI | Tabla que asigna responsables (R), aprobadores (A), consultados (C) e informados (I). | Asegurar claridad en la asignación de roles y responsabilidades. |
| Avance | Matriz de Riesgo | Identificación de riesgos, impacto, probabilidad y planes de mitigación. | Reducir incertidumbre y preparar acciones preventivas. |
| Avance | Matriz de Control de Cambios | Registro formal de solicitudes de cambio con impacto en alcance, costo y tiempo. | Mantener el control sobre modificaciones al proyecto. |
| Avance | Minutas de Reuniones | Documentación de acuerdos, compromisos y temas tratados en reuniones. | Asegurar trazabilidad de decisiones y compromisos asumidos. |
| Avance | Plan de Pruebas | Documento que define alcance, criterios, tipos de pruebas y cronograma. | Planificar y organizar adecuadamente la validación del sistema. |
| Avance | Reportes de Pruebas intermedias | Evidencias de pruebas unitarias y funcionales realizadas en fases iniciales. | Validar que cada módulo cumple antes de integrarse al sistema. |
| Avance | Matriz de Pruebas de Base de Datos | Casos de prueba específicos para validación de consultas, integridad y seguridad. | Asegurar la confiabilidad y consistencia de la información almacenada. |
| Avance | Planilla Registro de Defectos | Registro de errores encontrados, severidad, estado y responsable de corrección. | Dar trazabilidad y priorización al ciclo de corrección de errores. |
| Avance | Verificación de Alcances | Documento comparativo entre alcance planificado y entregado. | Asegurar cumplimiento de objetivos y detectar desviaciones. |
| Final | Aplicación de Escritorio EncryptU | Producto final en Python con GUI (Tkinter/ttkbootstrap), módulos de cifrado/descifrado, autenticación y gestión de usuarios. | Entregar solución concreta para la protección accesible y segura de información. |
| Final | Aplicación Web secundaria | Sistema complementario en Flask + React para gestión de tickets de soporte y notificaciones. | Facilitar retroalimentación del usuario y trazabilidad del uso del sistema. |
| Final | Plan de Capacitación | Documento/plan para entrenar usuarios finales y administradores del sistema. | Asegurar adopción y uso eficiente de la solución entregada. |
| Final | Manual de Usuario | Guía de instalación, configuración y uso del sistema. | Proveer documentación clara y accesible para el usuario final. |
| Final | Informe de Cierre del Proyecto | Documento final que resume logros, métricas, lecciones aprendidas y formaliza cierre. | Entregar balance global del proyecto y consolidar aprendizaje organizacional. |
| **7. Plan de Trabajo** |
| En la siguiente tabla define la planificación de tu Proyecto APT de acuerdo a lo requerido. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Plan de Trabajo Proyecto APT** | | | | | | |
| Competencia o unidades de competencias | Nombre de Actividades/Tareas | Descripción Actividades/Tareas | Recursos | Duración de la actividad | Responsable[[1]](#footnote-1) | Observaciones |
| Gestión de proyectos | Acta de Constitución | Elaborar documento base que define objetivos, alcance, presupuesto, cronograma y riesgos. | PC, plantillas de acta | 3 | Felipe Torres (Jefe de Proyecto) | Documento clave para formalizar el inicio del proyecto. |
| Gestión de proyectos | Definición de Requisitos (ERS) | Identificar y documentar requerimientos funcionales y no funcionales. | Entrevistas, PC, plantillas ERS | 4 | Felipe + Liam | Requiere validación con el cliente. |
| Gestión de proyectos | EDT + Matriz RACI | Desglose de estructura de trabajo y asignación de roles. | Plantilla EDT + Excel | 1 | Felipe Torres | Facilita la gestión y asignación de tareas. |
| Análisis | Documento de Arquitectura SW (DAS) | Definición de capas, APIs, BD y componentes técnicos. | PC, software de modelado | 3 | Manuel Miqueles (DBA) | Garantiza orden y calidad técnica. |
| Análisis | Casos de Uso | Modelar interacciones entre usuarios y sistema. | Plantillas de casos de uso | 4 | Felipe + QA (Nicolás) | Insumo para pruebas y validación funcional. |
| Diseño UI/UX | Mockups y Prototipo | Diseñar interfaces y flujos para validar usabilidad. | Figma, PC | 4 | Manuel Miqueles (Diseñador) | Validación temprana con cliente. |
| Desarrollo Web | Front-End Web (Administrador) | Construcción de interfaz web administrativa en React. | PC, VSCode, React | 19 | Liam | Requiere revisión de usabilidad. |
| Desarrollo Web | Back-End Web (Administrador) | Construcción de API y lógica de negocio en Flask. | PC, Python, Flask | 26 | Liam | Integración con BD y front-end. |
| Desarrollo Escritorio | Front-End App Escritorio | Interfaz gráfica de escritorio con Tkinter/ttkbootstrap. | Python, VSCode | 19 | Felipe | Orientada a adultos mayores (usabilidad). |
| Desarrollo Escritorio | Back-End App Escritorio | Implementación de lógica interna (cifrado, gestión usuarios). | Python, SQLite | 26 | Felipe | Uso de POO y buenas prácticas. |
| Seguridad | Implementación AES-256 y Encriptado único | Desarrollo de módulos de cifrado y descifrado con Fernet/AES. | Python, librería cryptography | 12 | Felipe + Liam | Fundamental para garantizar la seguridad de datos. |
| Integración | Implementación APIs | Desarrollo e integración de servicios REST. | Flask, Postman | 12 | Liam | Permite comunicación entre escritorio y web. |
| Bases de Datos | Implementación BD | Creación de tablas y consultas en PostgreSQL. | SQL Developer, PC | 12 | Manuel Miqueles | Asegurar consistencia e integridad. |
| UI/UX | Desarrollo de interfaz | Ajustes visuales, usabilidad y accesibilidad. | Figma, Python | 19 | Manuel Miqueles | Adaptar a usuarios no técnicos. |
| QA / Seguridad | Plan de Pruebas | Definir pruebas funcionales, no funcionales y de seguridad. | PC, documentación | 1 | Nicolás Fernández (QA) | Define métricas y alcance de pruebas. |
| QA / Seguridad | Pentesting y pruebas de seguridad | Evaluación de vulnerabilidades, cifrado y accesos. | Kali Linux, PC | 3 | Nicolás Fernández | Posibles ajustes tras hallazgos. |
| QA / Seguridad | Pruebas de Usabilidad | Validación con usuarios representativos (adultos mayores). | Prototipo, usuarios reales | 4 | Manuel Miqueles + QA | Medir satisfacción y facilidad de uso. |
| QA | Auditoría y Certificación preliminar | Revisión integral de cumplimiento de requerimientos. | Checklist, PC | 2 | Felipe + QA | Requisito previo al lanzamiento. |
| Implementación | Plan de despliegue | Estrategia de lanzamiento en fases y control de riesgos. | Documentación | 2 | Felipe | Minimiza impacto de fallos. |
| Implementación | Despliegue en fases y Go-Live | Migración controlada y puesta en producción. | Hosting, servidores | 4 | Equipo completo | Fase crítica, requiere monitoreo. |
| Cierre | Optimización post-producción | Correcciones y mejoras tras Go-Live. | PC, logs | 7 | Felipe + Liam | Estabilización del sistema. |
| Cierre | Manual de Usuario | Elaboración de guía de uso paso a paso. | ReportLab, Word | 1 | Manuel Miqueles | Orientado a usuarios sin conocimientos técnicos. |
| Cierre | Informe final y Cierre formal | Documento de cierre con resultados, aprendizajes y entregables. | PC, plantillas | 1 | Felipe | Marca el cierre oficial del proyecto. |

|  |
| --- |
| **8. Carta Gantt** |
| Busca un formato de Carta Gantt que te acomode y organiza en este las actividades planificadas en el punto anterior considerando el periodo asignado para el desarrollo de tu Proyecto APT. Debes mantener la temporalidad del periodo académico en el desarrollo de las tres fases que contempla la Asignatura de Portafolio de Título. |

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre de tarea | Duración | Comienzo | Fin |
| **Inicio de proyecto** | **85 días** | **lun 11-08-25** | **vie 05-12-25** |
| **Planificacion** | **5 días** | **lun 11-08-25** | **vie 15-08-25** |
| Definicion de requisitos | 3 días | lun 11-08-25 | mié 13-08-25 |
| Asignacion de recursos | 2 días | lun 11-08-25 | mar 12-08-25 |
| Plantilla de Requerimiento | 4 días | lun 11-08-25 | jue 14-08-25 |
| EDT | 1 día | lun 11-08-25 | lun 11-08-25 |
| Matriz RACI | 1 día | lun 11-08-25 | lun 11-08-25 |
| Plantilla acta de constitucion | 3 días | lun 11-08-25 | mié 13-08-25 |
| Captura de requerimientos especificos | 4 días | lun 11-08-25 | jue 14-08-25 |
| Documento de arquitectura SW | 3 días | lun 11-08-25 | mié 13-08-25 |
| Documento Casos de uso | 4 días | lun 11-08-25 | jue 14-08-25 |
| Prototipo | 4 días | lun 11-08-25 | jue 14-08-25 |
| Propuesta ERS | 3 días | lun 11-08-25 | mié 13-08-25 |
| Monitoreo de riesgos | 2 días | lun 11-08-25 | mar 12-08-25 |
| Plantilla de costos | 1 día | vie 15-08-25 | vie 15-08-25 |
| **Desarrollo de proyecto** | **65 días** | **lun 11-08-25** | **vie 07-11-25** |
| Front-End Web Administrador | 15 días | lun 11-08-25 | vie 29-08-25 |
| Back-End Web Administrador | 20 días | lun 11-08-25 | vie 05-09-25 |
| Front-End App Escritorio | 15 días | lun 11-08-25 | vie 29-08-25 |
| Back-End App Escritorio | 20 días | lun 11-08-25 | vie 05-09-25 |
| Implementación AES-256 y Encriptado unico | 10 días | lun 08-09-25 | vie 19-09-25 |
| Implementación APIs | 10 días | lun 22-09-25 | vie 03-10-25 |
| Implementación de BD | 10 días | lun 06-10-25 | vie 17-10-25 |
| Desarrollo de interfaz | 15 días | lun 20-10-25 | vie 07-11-25 |
| **Pruebas de seguridad** | **8 días** | **lun 10-11-25** | **mié 19-11-25** |
| Plan de pruebas | 1 día | lun 10-11-25 | lun 10-11-25 |
| Pruebas de seguridad | 1 día | mar 11-11-25 | mar 11-11-25 |
| Pentesting | 2 días | mié 12-11-25 | jue 13-11-25 |
| Usabilidad | 2 días | vie 14-11-25 | lun 17-11-25 |
| Auditoría | 1 día | mar 18-11-25 | mar 18-11-25 |
| Certificación Preliminar | 1 día | mié 19-11-25 | mié 19-11-25 |
| **Lanzamiento a produccion** | **6 días** | **jue 20-11-25** | **jue 27-11-25** |
| Plan de despliegue | 2 días | jue 20-11-25 | vie 21-11-25 |
| Despliegue en fases | 2 días | lun 24-11-25 | mar 25-11-25 |
| Go live | 2 días | mié 26-11-25 | jue 27-11-25 |
| **Post-Produccion** | **6 días** | **vie 28-11-25** | **vie 05-12-25** |
| Optimizacion continua(estabilizacion) | 1 día | vie 28-11-25 | vie 28-11-25 |
| Auditoria post-implementación | 1 día | lun 01-12-25 | lun 01-12-25 |
| Manual de usuario | 1 día | mar 02-12-25 | mar 02-12-25 |
| Mantencion | 2 días | mié 03-12-25 | jue 04-12-25 |
| Cierre formal del proyecto | 1 día | vie 05-12-25 | vie 05-12-25 |

Gráfico, Escala de tiempo

El contenido generado por IA puede ser incorrecto.

1. En caso de que el Proyecto APT sea grupal, en esta columna deben indicar el nombre de los responsables de cada tarea o actividad. Esto posteriormente permitirá diferenciar la evaluación por cada integrante. [↑](#footnote-ref-1)