}

1. **PARTE I**

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

|  |  |
| --- | --- |
| Nombre estudiante | **Erick Brandon San Martín Zuñiga** |
| Rut | **21.449.917-7** |
| Carrera | **Ingeniería en informática** |
| Sede | **Maipú** |

|  |  |
| --- | --- |
| Nombre estudiante | **John Andres Herrera Montes** |
| Rut | **21.367.897-3** |
| Carrera | **Ingeniería en informática** |
| Sede | **Maipú** |

|  |  |
| --- | --- |
| Nombre estudiante | **Alexander Javier Pulgar Allende** |
| Rut | **21.091.734-9** |
| Carrera | **Ingeniería en informática** |
| Sede | **Maipú** |

|  |
| --- |
| **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 | OfiSync: Sistema Integral de Gestión para Edificios |
| Área (s) de desempeño(s) | Desarrollo de Software  Análisis de Sistemas  Administración de Servicios Informáticos  Gestión de Proyectos Tecnológicos |
| Competencias | Administrar la configuración de ambientes, servicios de aplicaciones y bases de datos   * Configuración del VPS, base de datos centralizada y ambientes de desarrollo/pruebas.   Ofrecer propuestas de solución informática analizando de forma integral los procesos   * Definición de módulos según los perfiles (Administrador, Conserje, Aseo, Clientes).   Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento   * Desarrollo iterativo en sprints, control de versiones, pruebas y documentación.   Construir modelos de datos escalables   * Diseño y normalización de la base de datos centralizada.   Programar consultas o rutinas para manipular información en la base de datos   * CRUD de gastos comunes, reservas, inventarios, bitácoras y pagos.   Construir programas y rutinas de variada complejidad con buenas prácticas de codificación   * Desarrollo del portal web y aplicación móvil con integración a la base de datos.   Realizar pruebas de calidad tanto de los productos como de los procesos   * Testing unitario, integración, funcional y de carga.   Construir el modelo arquitectónico de la solución sistémica   * Definición de arquitectura multiplataforma (web + app + base de datos centralizada).   Implementar soluciones sistémicas integrales para automatizar procesos de negocio   * Automatización de gastos comunes, reservas, validación de pagos, control de inventarios y notificaciones.   Resolver vulnerabilidades sistémicas y cumplir normas de seguridad   * Validaciones de usuarios, cifrado de credenciales, pruebas de seguridad en la base de datos y aplicación. |

|  |
| --- |
| **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 | Este proyecto busca abordar una problemática real en la administración de edificios en Chile: la falta de digitalización y automatización en la gestión diaria de gastos comunes, reservas de servicios, bitácoras de limpieza, y registros de entradas y salidas. Actualmente, muchas comunidades aún utilizan procesos manuales, planillas de Excel o software no especializado que no se adapta a las necesidades específicas de cada rol (administrador, inquilino, personal de aseo y conserjes).  El proyecto se sitúa principalmente en la Región Metropolitana, especialmente en comunas como Las Condes, Providencia, Huechuraba y Vitacura, donde existe una alta concentración de edificios de oficinas y también comunas como Santiago Centro donde existe una gran concentración de edificios residenciales. Impactaría directamente a administradores de edificios, trabajadores de aseo, conserjes e inquilinos, mejorando su experiencia, eficiencia y trazabilidad en los procesos.  La elección de este tema responde tanto a su impacto real en el entorno laboral como a su alineación con tendencias actuales en transformación digital, automatización y servicios inteligentes. |
| Descripción del Proyecto APT | El proyecto consiste en desarrollar un sistema multiplataforma compuesto por:  Un portal web desarrollado con vistas personalizadas para:   * Administrador: cálculo y gestión de gastos comunes, recepción y validación de comprobantes de pago, manejo de reservas de servicios, control de inventario de insumos. * Conserje: registro de bitácoras de recepción, control de accesos. * Personal de aseo: administración del inventario de insumos y registro de actividades en la bitácora de limpieza.   Una aplicación móvil híbrida (Ionic + JavaScript) dirigida a:   * Clientes (Inquilinos): consultar el monto de gastos comunes, subir comprobantes de pago, y reservar servicios comunes del edificio.   Una base de datos centralizada, alojada en un VPS, que sincroniza en tiempo real la información entre el portal web y la aplicación móvil, asegurando consistencia y disponibilidad de los datos para todos los perfiles de usuario.  El objetivo del sistema es modernizar y automatizar los procesos de administración de edificios de oficinas, optimizando la eficiencia operativa y mejorando la experiencia de los distintos usuarios involucrados. |
| Pertinencia del proyecto con el perfil de egreso | El proyecto se relaciona directamente con el perfil de egreso ya que:   * Requiere del diseño y desarrollo de soluciones tecnológicas completas, considerando distintos tipos de usuarios y dispositivos. * Implica gestión de proyectos tecnológicos, planificación por sprint, control de entregables y seguimiento del avance. * Aplica principios de interoperabilidad, sincronización de datos, y seguridad informática, que forman parte del perfil de egreso. * Se basa en metodologías ágiles, como Scrum, que permiten entregar valor de forma iterativa y continua. * Las competencias seleccionadas son esenciales para el éxito del proyecto, ya que permiten abordar tanto la parte técnica como la de gestión. |
| Relación con los intereses profesionales | Este proyecto se alinea perfectamente con nuestros intereses profesionales en el desarrollo de software.  Nos permite trabajar en múltiples áreas del desarrollo como frontend, backend, bases de datos y arquitectura de sistemas, además de fortalecer nuestra experiencia con metodologías ágiles y desarrollo multiplataforma.  Su implementación nos dará la oportunidad de mejorar nuestras habilidades técnicas y ampliar nuestro portafolio con un proyecto sólido y funcional, útil para el mercado laboral. |
| Factibilidad de desarrollo del Proyecto APT | Fortalezas:   * El proyecto está bien delimitado en sus funcionalidades principales y cuenta con un backlog organizado por épicas e historias de usuario. * El tiempo estimado (10 semanas) es suficiente, dividido en 3 sprints, lo que permite una entrega progresiva de funcionalidades. * Las tecnologías utilizadas (Ionic, JavaScript, Django) son manejables y realistas para este contexto académico. * Los materiales requeridos son accesibles (software libre o con licencias académicas).   Debilidades/Potenciales dificultades:   * La sincronización entre las plataformas puede generar complejidad técnica. * Posibles dificultades con la conexión y configuración del VPS. * Requiere una planificación cuidadosa para no sobrecargar los sprints.   Solución:   * Mitigar estas debilidades aplicando metodologías ágiles y realizando pruebas frecuentes. * Priorizar funcionalidades críticas en los primeros sprints. * Consultar a docentes/tutores técnicos ante trabas técnicas. |

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 un sistema multiplataforma compuesto por un portal web y una aplicación móvil, que modernice y automatice la administración de edificios de oficinas, mediante la digitalización de procesos, la optimización de recursos y la mejora de la experiencia de todos los usuarios involucrados. |
| Objetivos específicos | Objetivos Estratégicos:   * Reducir los errores administrativos mediante la digitalización y automatización de procesos clave como gastos comunes, reservas y bitácoras. * Mejorar la transparencia y la confianza de los inquilinos en la gestión de pagos y reservas a través de un sistema accesible y seguro. * Ahorrar tiempo y recursos al optimizar los flujos de trabajo de administradores, conserjes y personal de aseo. * Aumentar la satisfacción de los usuarios al ofrecer una plataforma moderna, intuitiva y eficiente. * Desarrollar una solución escalable y flexible, capaz de adaptarse a futuras mejoras o ampliaciones funcionales. * Implementar metodologías ágiles que garanticen una gestión eficiente del desarrollo y entregas oportunas. * Fomentar una colaboración efectiva entre los miembros del equipo de desarrollo y los stakeholders.   Objetivos del Proyecto:   * Desarrollar un portal web con vistas diferenciadas para: * Administradores (gestión de gastos comunes, reservas, comprobantes, inventarios, agenda). * Conserjes (bitácoras de recepción, control de accesos, tareas asignadas). * Personal de aseo (registro de actividades y control de insumos). * Desarrollar una aplicación móvil orientada a clientes (inquilinos) con funcionalidades como: * Consulta del monto de gastos comunes. * Subida de comprobantes de pago. * Reserva de servicios comunes. * Asegurar la sincronización en tiempo real entre la aplicación móvil y el portal web a través de una base de datos centralizada en la nube. * Diseñar interfaces intuitivas y accesibles para todos los tipos de usuarios, asegurando facilidad de uso. * Implementar un sistema seguro de almacenamiento y consulta de datos, con respaldo en la nube. * Estructurar una base de datos robusta que permita gestionar gastos comunes, reservas, registros y bitácoras de manera eficiente. * Habilitar la carga y validación de comprobantes de pago para facilitar el control financiero de los administradores. * Implementar una bitácora digital para conserjes y personal de aseo, con la posibilidad de adjuntar fotos y observaciones. * Desarrollar un sistema de reservas eficiente, evitando conflictos de horarios y duplicaciones. * Crear un módulo de inventario de insumos de aseo, con alertas de stock bajo para facilitar su reposición. * Cumplir con los plazos establecidos para el desarrollo, sin comprometer la calidad técnica ni funcional del producto. |

|  |
| --- |
| **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 |
| Metodología Ágil: Scrum  El proyecto será desarrollado utilizando la metodología ágil Scrum, con el propósito de garantizar entregas frecuentes, funcionales y alineadas con las necesidades del cliente. Scrum facilitará la colaboración continua, una planificación dinámica y la capacidad de adaptación ante cambios en los requerimientos o prioridades.  Product Owner: Alexander Pulgar  Scrum Master: John Herrera  Developer Web 1: John Herrera  Developer Web 2: Erick San Martín  Developer App Móvil: Alexander Pulgar  DBA 1: Erick San Martín  DBA 2: Alexander Pulgar  DBA 3: John Herrera  QA 1: John Herrera  QA 2: Alexander Pulgar  QA 3: Erick San Martin |

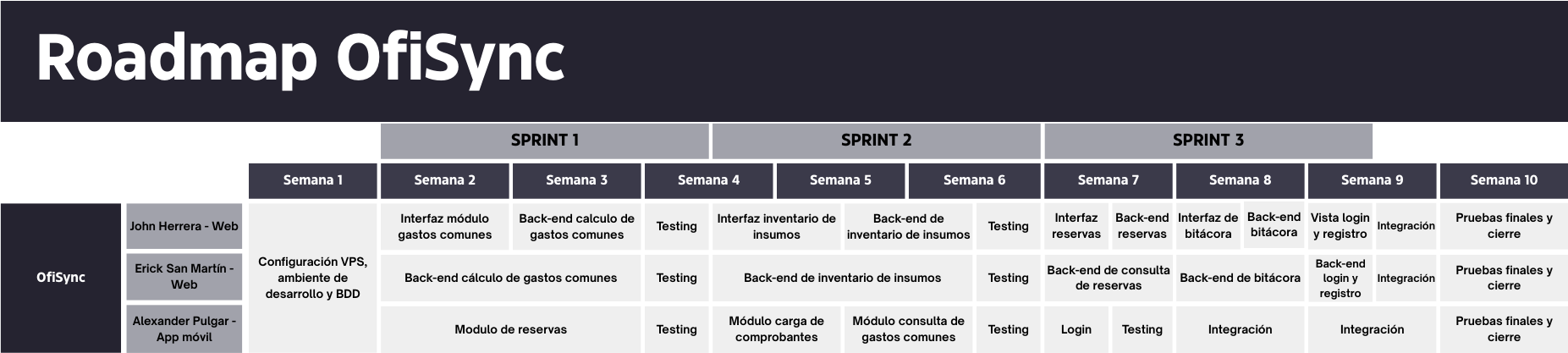
|  |
| --- |
| **6. Evidencias** |
| A continuación, describe qué evidencias serán evaluadas en el informe de avance y en el informe final de tu proyecto APT. Estas evidencias deben ser acordadas con tu docente. Se entenderá por evidencia los productos que se desarrollen durante el proyecto y cuyo propósito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

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo de evidencia**  **(avance o final)** | **Nombre de la evidencia** | **Descripción** | **Justificación** |
| Avance | Acta Scrum | Documento inicial donde se establecen los objetivos del proyecto, el equipo Scrum, roles y el alcance inicial. | Permite alinear al equipo y stakeholders en torno a un marco formal de inicio. |
| Avance | Product Backlog | Lista priorizada de historias de usuario y funcionalidades a desarrollar. | Garantiza que el equipo tenga claridad del alcance y prioridades. |
| Avance | Módulo de gastos comunes completo (web) | Entregable funcional que permite calcular, visualizar y administrar los gastos comunes en la plataforma web. | Representa un avance real del proyecto y valida la integración frontend-backend. |
| Avance | Módulo de reservas completo (móvil) | Entregable funcional que permite a los usuarios realizar reservas desde la aplicación móvil. | Asegura la disponibilidad temprana de una funcionalidad clave para los usuarios. |
| Avance | Sprint backlog del primer sprint. | Selección de historias de usuario y tareas técnicas definidas para el Sprint 1. | Muestra la planificación detallada de trabajo en el primer ciclo. |
| Avance | Acta de retrospectiva del primer sprint. | Documento que resume aprendizajes, problemas y mejoras detectadas al cierre del Sprint 1. | Permite al equipo reflexionar y mejorar continuamente. |
| Avance | Reporte de testing primer sprint. | Resultados de pruebas unitarias, integradas y de usuario del Sprint 1. | Asegura la calidad del avance entregado y da evidencia objetiva de validación. |
| Final | Sistema completo funcional. | Entrega final del producto con todas las funcionalidades web y móviles integradas. | Demuestra que el proyecto cumple con el alcance comprometido y está listo para producción. |
| Final | Sprint backlog resto de sprints. | Planificación de las tareas y funcionalidades trabajadas en los sprints posteriores. | Evidencia el control y seguimiento realizado durante todo el desarrollo. |
| Final | Acta de retrospectiva resto de sprints. | Documentos que resumen los aprendizajes de los sprints 2, 3. | Refuerza la mejora continua y la adaptación del proceso Scrum. |
| Final | Reporte de testing completo | Reporte consolidado de todas las pruebas realizadas al sistema completo. | Da garantía de calidad final y reduce riesgos de fallos en producción. |

|  |
| --- |
| **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 |
| Comunicación y coordinación de equipo. | Reuniones diarias | Reuniones de 15 min cada día para sincronizar avances, bloqueos y próximos pasos. | Videollamadas, chat, tablero Scrum. | 15 a 20 minutos por día/Todo el proyecto | Equipo completo | Clave para mantener alineación y comunicación. |
| Administrar la configuración de ambientes, servicios de aplicaciones y bases de datos. | Configuración VPS y ambientes | Configuración de servidor AWS, instalación de dependencias y setup de entornos de desarrollo/pruebas. | AWS, Docker, Git. | 2 días | Equipo completo | Actividad inicial obligatoria para todo el desarrollo. |
| Construir modelos de datos escalables / Programar consultas o rutinas para manipular información en la base de datos. | Configuración base de datos | Diseño y creación de la base centralizada en PostgreSQL. | PostgreSQL, pgAdmin. | 3 días | Equipo completo | Requiere definir esquema y relaciones iniciales. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Interfaz módulo gastos comunes (web) | Pantallas para visualización y edición de gastos comunes en Django frontend. | Django templates, CSS, JS. | 1 semana | John Herrera | Se conecta a API de gastos comunes. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Construir programas y rutinas de variada complejidad con buenas prácticas de codificación. | Backend cálculo de gastos comunes | Desarrollo de lógica y endpoints para el cálculo de gastos comunes. | Django REST, PostgreSQL. | 2 semanas | Erick San Martín y John Herrera | Integración directa con frontend y app. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Módulo de reservas (app móvil) | Desarrollo de pantallas y lógica de reservas en Ionic. | Ionic, Angular. | 2 semanas | Alexander Pulgar | Requiere endpoints backend de reservas. |
| Realizar pruebas de calidad tanto de los productos como de los procesos. | Testing inicial | Pruebas de los módulos de gastos comunes y reservas. | Pruebas manuales. | Media semana | Equipo completo | Validación del primer incremento del producto. |
| Realizar pruebas de calidad tanto de los productos como de los procesos. | Interfaz inventario de insumos (web) | Desarrollo de formulario web para CRUD de insumos. | Django templates, CSS | 1 semana | John Herrera | Validaciones de stock. |
| Construir modelos de datos escalables / Programar consultas o rutinas para manipular información en la base de datos. | Backend inventario de insumos | API CRUD para insumos con persistencia en PostgreSQL. | Django REST, PostgreSQL | 2 semanas | Erick San Martín y John Herrera | Se conecta con interfaz web. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Módulo carga de comprobantes (app móvil) | Función en la app para subir comprobantes (PDF/imagen). | Ionic, cámara/galería | 1 semana | Alexander Pulgar | Validar formato de archivo antes de subir |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Módulo consulta de gastos comunes (app móvil) | Pantalla en app para visualizar gastos comunes a pagar. | Ionic, Angular | 1 semana | Alexander Pulgar | Requiere API estable de gastos comunes. |
| Realizar pruebas de calidad tanto de los productos como de los procesos. | Testing intermedio | Pruebas de inventario, comprobantes y gastos comunes en móvil. | Pruebas manuales | Media semana | Equipo completo | Validaciones cruzadas de módulos. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Interfaz consulta de reservas (web) | Pantallas para que conserje/administrador vea reservas. | Django templates, CSS | Media semana | John Herrera | Se conecta al backend de reservas. |
| Construir programas y rutinas de variada complejidad / Implementar soluciones sistémicas integrales. | Backend consulta de reservas | Endpoints para devolver reservas al frontend web. | Django REST, PostgreSQL | 1 semana | Erick San Martín y John Herrera | Conexión con módulo de reservas móvil. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Interfaz bitácora (web) | Pantalla para notas, control de acceso, registro de bitácora. | Django templates, CSS | Media semana | John Herrera | Sin observaciones. |
| Construir programas y rutinas de variada complejidad / Programar consultas o rutinas para manipular información en la base de datos. | Backend bitácora | API CRUD para almacenar registros de bitácora. | Django REST, PostgreSQL | 1 semana | Erick San Martín y John Herrera | Debe garantizar persistencia y consulta rápida. |
| Desarrollar una solución de software utilizando técnicas sistemáticas de desarrollo y mantenimiento / Implementar soluciones sistémicas integrales. | Vista login y registro (web) | Desarrollo de pantallas de login y registro de usuarios. | Django templates, CSS, JS | Media semana | John Herrera | Requiere integración con backend. |
| Construir programas y rutinas de variada complejidad / Resolver vulnerabilidades sistémicas y cumplir normas de seguridad. | Backend login y registro | API de autenticación y roles de usuario. | Django REST, PostgreSQL | Media semana | Erick San Martín | Incluye validación y cifrado de credenciales. |
| Implementar soluciones sistémicas integrales / Resolver vulnerabilidades sistémicas y cumplir normas de seguridad. | Login app móvil | Implementación de login en la app móvil (Ionic). | Ionic, Angular | Media semana | Alexander Pulgar | Sin observaciones |
| Construir el modelo arquitectónico de la solución sistémica / Implementar soluciones sistémicas integrales. | Integración de módulos | Conexión de todos los módulos web, backend y app móvil. | Git, APIs REST, Postman | 2 semanas | Equipo completo | Puede requerir ajustes de endpoints y UI. |
| Realizar pruebas de calidad tanto de los productos como de los procesos / Implementar soluciones sistémicas integrales. | Testing final y cierre | Pruebas de regresión, funcionales y de usabilidad para el sistema completo. | Pruebas manuales | 1 semana | Equipo completo | Validación final previa a puesta en marcha. |

|  |
| --- |
| **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. |



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)