A white and blue document with black text

Description automatically generated

A continuación encontrarás una planificación de alto nivel para un proyecto de 16 semanas (8 Sprints de 2 semanas cada uno) que aborda 5 épicas: la primera corresponde a la configuración inicial y al levantamiento de un MVP, y las cuatro restantes a los requerimientos funcionales.

Visión General de las Épicas

Épica 1: Configuración inicial y MVP.

Enfoque en la puesta en marcha de la infraestructura técnica, herramientas, despliegues iniciales y un prototipo funcional mínimo (MVP).

Épica 2: Registro y notificación de entregas.

Incluye todo lo referente al ingreso de paquetes, asignación a departamentos y notificaciones automáticas.

Épica 3: Historial y seguimiento de encomiendas.

Registro detallado y seguimiento de paquetes, con la posibilidad de notificaciones y estados de pendiente/entregado/retirado.

Épica 4: Control de acceso a correspondencia.

Verificación digital (QR/códigos únicos), registro de quién retira la encomienda.

Épica 5: Panel de administración para conserjes.

Dashboard, gestión centralizada de paquetes pendientes, reclamos y administración básica.

Planificación de Sprints

- Sprint 1 (Semanas 1-2):

- Trabajar principalmente en Épica 1 (Configuración inicial, MVP básico).

- Sprint 2 (Semanas 3-4):

- Continuar y finalizar Épica 1 si algo quedó pendiente.

- Iniciar Épica 2 (Registro y notificación de entregas).

- Sprint 3 (Semanas 5-6):

- Completar Épica 2.

- Iniciar Épica 3 (Historial y seguimiento de encomiendas).

- Sprint 4 (Semanas 7-8):

- Completar Épica 3.

- Iniciar Épica 4 (Control de acceso a correspondencia).

- Sprint 5 (Semanas 9-10):

- Completar Épica 4.

- Sprint 6 (Semanas 11-12):

- Trabajar en Épica 5 (Panel de administración para conserjes).

- Sprint 7 (Semanas 13-14):

- Finalizar Épica 5.

- Integración, pruebas e iteraciones de mejora.

- Sprint 8 (Semanas 15-16):

- Ajustes finales, estabilización, pruebas de aceptación del usuario (UAT) y despliegue a producción.

Detalle por Épica

ÉPICA 1: Configuración inicial y levantamiento de MVP

Historias de Usuario

1. Como líder técnico quiero configurar el repositorio del proyecto y el entorno de desarrollo (CI/CD, control de versiones). Para asegurar que todos los miembros del equipo trabajen con la misma base y se puedan hacer despliegues iniciales de forma consistente.

2. Como desarrollador quiero tener un esqueleto básico de la aplicación (backend y frontend). Para poder implementar las funcionalidades mínimas y construir sobre una arquitectura sólida.

3. Como Product Owner quiero un MVP que muestre la estructura de menú/navegación y una pantalla de prueba. Para validar con los stakeholders la interfaz inicial y el flujo general.

Tareas Principales

- T1.1 Configuración de repositorio

- Crear repositorios en GitHub.

- Definir ramificaciones (branches) y políticas de merge (pull/merge requests).

- T1.2 Pipeline de Integración Continua

- Configurar scripts para compilación y test (p. ej. GitHub Actions, Jenkins, CircleCI).

- Preparar entornos (staging, producción).

- T1.3 Arquitectura base

- Definir y documentar la arquitectura (capas de backend, framework frontend, base de datos).

- Crear proyectos base (ej. Spring Boot + React, Node.js + Vue, etc.).

- T1.4 Despliegue MVP

- Configurar un hosting o servidor de prueba.

- Subir la primera versión con navegación mínima (login o pantalla de bienvenida “Hello World”).

ÉPICA 2: Registro y notificación de entregas

Historias de Usuario

1. Como conserje quiero ingresar al sistema nuevos paquetes con el número de departamento para registrar el ingreso de una encomienda y vincularla a un residente.

2. Como residente quiero recibir una notificación cuando un paquete ha llegado a mi nombre para saber que debo recogerlo a la brevedad.

3. Como sistema quiero guardar la fecha y hora exacta en que se ingresa el paquete para mantener un historial confiable de entregas.

Tareas Principales

- T2.1 Modelado y creación de la tabla “Paquetes”

- Definir campos: ID, número de departamento, fecha/hora de ingreso, estado inicial, etc.

- Configurar migraciones o script SQL.

- T2.2 Lógica de registro de nuevos paquetes

- Endpoint/Servicio de backend para alta de un nuevo paquete.

- Formulario de frontend para que el conserje ingrese la información.

- T2.3 Notificaciones automáticas

- Integrar un servicio de notificaciones (correo, push notification o SMS).

- Configurar triggers al momento de registrar el paquete.

- T2.4 Validaciones y pruebas

- Asegurar que se ingresen campos obligatorios (departamento, fecha/hora).

- Pruebas unitarias y de integración para comprobar el correcto registro y notificación.

ÉPICA 3: Historial y seguimiento de encomiendas

Historias de Usuario

1. Como residente quiero ver un historial de mis encomiendas con fechas de recepción y retiro para llevar un control de cuándo llegaron y cuándo las recogí.

2. Como conserje quiero poder consultar rápidamente el estado de cada encomienda (pendiente, en reclamo, entregado) para brindar información precisa a los residentes.

3. Como sistema quiero enviar notificaciones insistentes en paquetes que no han sido retirados para asegurar que los residentes retiren a tiempo sus encomiendas (especialmente si son perecederas).

Tareas Principales

- T3.1 Diseño de la tabla o vista “Historial”

- Incluir campos de tracking: fecha ingreso, fecha retiro, estado, usuario que retira.

- Relacionar con la tabla de “Paquetes”.

- T3.2 Interfaces de consulta

- Frontend: pantalla de historial de encomiendas.

- Filtros y búsqueda (ej: por rango de fechas, por departamento, por estado).

- T3.3 Notificaciones recurrentes

- Cron job o servicio programado que revise paquetes en estado “pendiente” y envíe recordatorios.

- Configurar frecuencia e integración con API de mensajería/correo.

- T3.4 Pruebas y validaciones

- Revisar correcto funcionamiento de la actualización de estados.

- Asegurar la privacidad: sólo el residente puede ver su historial.

ÉPICA 4: Control de acceso a correspondencia

Historias de Usuario

1.Como residente quiero un código QR/código único para retirar mi paquete para autenticarme de forma rápida y evitar que alguien más retire mi encomienda.

2. Como conserje quiero escanear/verificar el código al entregar el paquete Para asegurarme de que la persona que lo retira sea la correcta.

3. Como administrador del sistema quiero llevar un registro de quién retira cada paquete (nombre, fecha y hora) para tener trazabilidad en caso de reclamos o incidentes.

Tareas Principales

- T4.1 Generación de códigos QR o tokens

- Integrar una librería de generación de QR/códigos.

- Asociar el código único al registro del paquete.

- T4.2 Lectura y validación

- Implementar un flujo en el frontend (o app móvil si aplica) que permita escanear/validar el código.

- Verificar en la base de datos el estado y la pertenencia del paquete.

- T4.3 Registro de retiro

- Endpoint para marcar encomienda como “retirada”, capturando la fecha y el usuario que la retira.

- Actualización de la información en tiempo real en la base de datos.

- T4.4 Pruebas de seguridad

- Validar que no se pueda reutilizar un código ya escaneado.

- Manejo de casos de error (código inválido, expirado, etc.).

ÉPICA 5: Panel de administración para conserjes

Historias de Usuario

1. Como conserje quiero ver en un solo lugar los paquetes pendientes de retiro, los recién ingresados y los reclamados para tener una vista unificada que me facilite la administración diaria.

2. Como conserje quiero marcar rápidamente los paquetes como entregados y cerrar reclamos para actualizar el estado sin complicaciones y mantener la base de datos al día.

3. Como administrador del edificio quiero ver estadísticas (ej: promedio de permanencia de encomiendas, tasa de reclamos) para gestionar mejor los recursos y tiempos de la administración.

Tareas Principales

- T5.1 Creación del dashboard

- Diseño e implementación de la pantalla principal para el conserje:

- Listado de paquetes pendientes.

- Notificaciones de nuevos ingresos.

- Sección de reclamos abiertos.

- T5.2 Funcionalidades de actualización rápida

- Botones o acciones para “Marcar como entregado” o “Atender reclamo”.

- Actualización inmediata en la base de datos.

- T5.3 Estadísticas y reportes

- Integrar un módulo de reportes con métricas básicas (promedio de permanencia, cantidad de reclamos, etc.).

- Generación de gráficas y visualizaciones (o exportación a CSV).

- T5.4 Control de roles y permisos

- Asegurar que únicamente conserjes o administradores vean y modifiquen esta sección.

- Manejo de autenticación y autorización.

Integración, Pruebas Finales y Despliegue

Después de completar el desarrollo de las épicas, se recomienda:

1. Pruebas de Integración y UAT

- A partir del Sprint 7, realizar pruebas integrales que cubran toda la funcionalidad end-to-end.

- Involucrar a usuarios reales (conserjes, residentes simulados) para retroalimentación.

2. Refinamiento de backlog y correcciones

- Ajustar detalles de UX, correcciones de bugs y mejoras detectadas durante las pruebas.

3. Despliegue a Producción (Sprint 8)

- Planificar la ventana de mantenimiento (si aplica).

- Verificar la documentación de instalación y manuales de usuario (conserjes, administradores).

4. Retrospectiva final

- Reunirse con el equipo para documentar lecciones aprendidas y sugerir mejoras en futuros proyectos.