PROCESO SCRUM – ZOONIKA

Este documento evidencia la aplicación del marco de trabajo SCRUM para el desarrollo de la aplicación full-stack Zoonika.

1. Definición de Roles

- **Product Owner:** Juan Richard Mamani Vilca
- **Scrum Master:** Juan Richard Mamani Vilca
- **Development Team:** Grupo A

2. Product Backlog

El Product Backlog es una lista priorizada de todas las funcionalidades deseadas para el producto.

ID	Historia de Usuario / Funcionalidad	Prioridad	Estimación	Estado
HU-01	Registro de nuevos usuarios	Alta	5 pts	Terminado
HU-02	Inicio de sesión de usuarios con generación de token JWT	Alta	8 pts	Terminado
HU-03	Visualización de la lista de galerías y sus especialistas	Alta	5 pts	Terminado
HU-04	Visualización del detalle de una galería con sus comentarios	Alta	5 pts	Terminado
HU-05	Añadir comentarios y valoraciones a una galería (solo usuarios logueados)	Alta	8 pts	Terminado
HU-06	Editar comentarios propios	Media	3 pts	Terminado
HU-07	Eliminar comentarios propios	Media	2 pts	Terminado
HU-08	Visualización de la lista de especialistas	Media	2 pts	Terminado
HU-09	Actualización del perfil de usuario (nombre, email)	Media	3 pts	Terminado
HU-10	Creación de historial de inicio de sesión por usuario	Baja	3 pts	Terminado
HU-11	Implementación de pruebas de integración para la API	Alta	8 pts	Terminado
HU-12	Refactorización de la arquitectura (Capas: Rutas, Controladores, Servicios)	Alta	13 pts	Terminado

3. Sprint Backlogs

Sprint 1 (Duración: 2 semanas)

- **Objetivo del Sprint:** Establecer la arquitectura base del backend y la funcionalidad de autenticación de usuarios.

Historia de Usuario	Criterios de Aceptación	Tareas Técnicas	Responsable
HU-12: Refactor	- El código está separado en capas (rutas, controllers, services). - Se implementa un manejador de errores central.	- Crear estructura de directorios. - Mover lógica de rutas a controladores. - Implementar capa de servicios. - Centralizar instancia de Prisma.	Backend Dev
HU-01: Registro	- Formulario valida campos. - Usuario se guarda en BD con	- Crear modelo de Usuario en Prisma.	Backend Dev

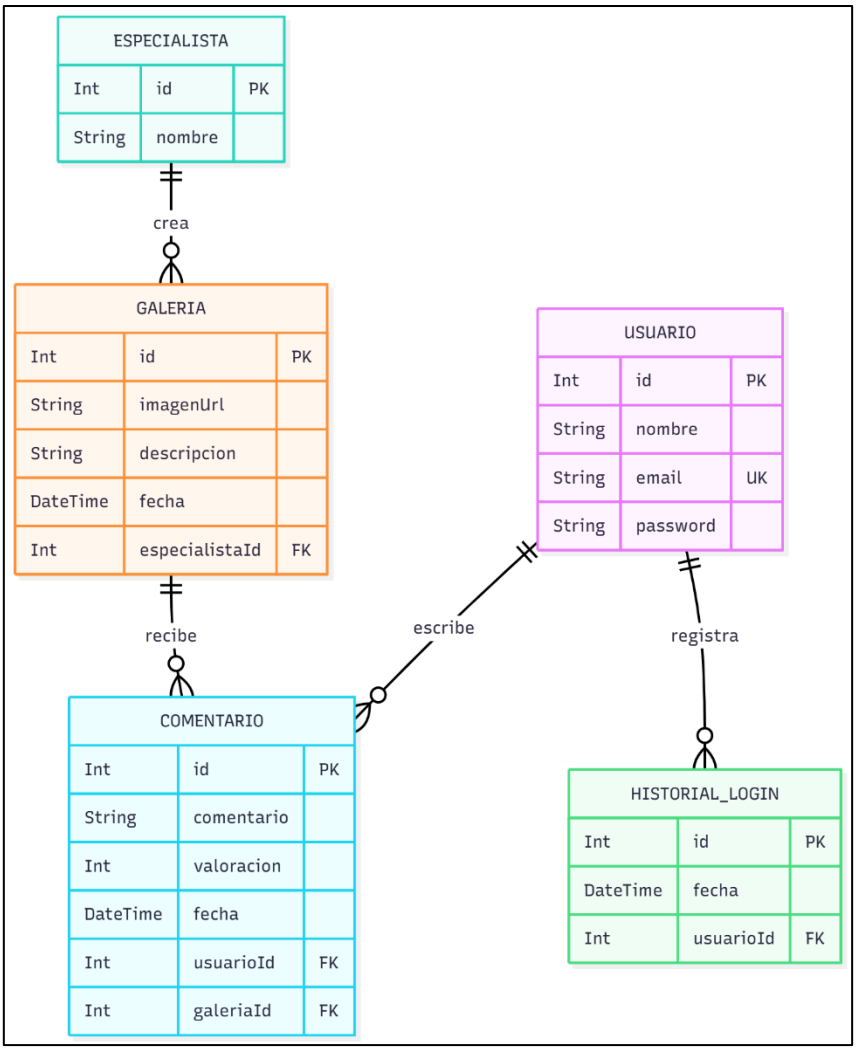
Historia de Usuario	Criterios de Aceptación	Tareas Técnicas	Responsable
	contraseña hasheada. - Email no puede ser duplicado.	- Implementar endpoint POST /auth/register. - Añadir lógica de hasheo con bcryptjs.	
HU-02: Login	- Usuario puede iniciar sesión con email/contraseña. - Se devuelve un token JWT válido con expiración.	- Implementar endpoint POST /auth/login. - Configurar y firmar tokens JWT. - Crear authMiddleware para verificar tokens.	Backend Dev
HU-11: Pruebas	- Las pruebas de autenticación pasan. - Se genera reporte de cobertura.	- Configurar Jest, Supertest y ts-jest. - Escribir pruebas para registro y login. - Configurar script npm test --coverage.	Backend Dev

Sprint 2 (Duración: 2 semanas)

- **Objetivo del Sprint:** Desarrollar las funcionalidades principales de visualización e interacción con galerías y comentarios.

Historia de Usuario	Criterios de Aceptación	Tareas Técnicas	Responsable
HU-03/04: Galerías	- Endpoints para listar y ver detalles de galerías. - El detalle incluye especialista y comentarios.	- Crear modelos Galeria, Especialista y Comentario en Prisma. - Implementar endpoints GET /galerias y /galerias/:id. - Construir vistas en Next.js para mostrar las galerías.	Full-Stack Dev
HU-05: Comentar	- Usuario logueado puede postear un comentario. - No se puede comentar dos veces en la misma galería.	- Implementar endpoint POST /comentarios protegido por authMiddleware. - Lógica en el servicio para verificar comentario existente. - Crear formulario en el frontend para enviar comentario.	Full-Stack Dev
HU-06/07: Editar/Eliminar	- Usuario solo puede editar/eliminar sus propios comentarios.	- Implementar endpoints PUT y DELETE en /comentarios/:id protegidos. - Lógica de autorización en los controladores. - Añadir botones y lógica en el frontend.	Full-Stack Dev
HU-10: Historial Login	- Cada login exitoso se guarda en la BD.	- Añadir modelo HistorialLogin a Prisma. - Ejecutar migración. - Actualizar authService para registrar el login.	Backend Dev

4. Diagrama de la Base de Datos (ERD)


ESPECIALISTA
Int id PK
String nombre
crea
GALERIA
Int id PK
String imagenUrl
String descripcion
DateTime fecha
Int especialistaId FK
recibe
USUARIO
Int id PK
String nombre
String email UK
String password
escribe
registra
COMENTARIO
Int id PK
String comentario
Int valoracion
DateTime fecha
Int usuarioId FK
Int galeriaId FK
HISTORIAL_LOGIN
Int id PK
DateTime fecha
Int usuarioId FK