

# **PLANIFICACIÓN FOOTBALL FIELD-KINAL**

**Proyecto Final 2025**

**H.E - Software Development**

**Centro Educativo Técnico Laboral KINAL**



**IN6BM – 2025**



**Anthony Josue Escobar Ponce - 202022G**



**Aníbal Guillermo Herrera Ortiz – 2020324**

## **Sistema de gestión de canchas KINAL**

Nuestro proyecto busca desarrollar una API web escalable y funcional para la gestión del apartado de canchas en nuestro centro educativo KINAL de forma eficiente y moderna, El objetivo principal del proyecto es digitalizar y automatizar el apartado, pago y registro respecto al uso de la cancha escolar en base a un conjunto de tecnologías JavaScript utilizadas para desarrollar aplicaciones web completas (Stack MERN), desde el frontend hasta el backend, compuestas por MongoDB para el manejo de la base de datos para los registros y constancias de los movimientos en la aplicación web, NodeJs como entorno base para ejecutar JavaScript en el servidor para la conexión de los procesos lógicos del backend hacia la base de datos, ExpressJs como framework para el entorno desarrollado en la conexión con NodeJs para el manejo de las rutas del servidor con el frontend recibiendo sus solicitudes y respondiendo con los datos necesarios y ReactJs como librería de desarrollo de JavaScript para crear interfaces de usuario interactivas y rápidas como botones, formularios y etc..

**Objetivos Específicos:**

Implementar un sistema para las reservas de nuestra cancha institucional de futbol que de manera visual muestre los horarios disponibles según el mes en que se encuentren. Permitir pagos en línea mediante tarjeta de crédito o débito desde la misma plataforma y ahorrarnos complicaciones mayores. Integrar un sistema de noticias y avisos para comunicar eventos como nuestros torneos de futbol llamados Inter talleres entre algunos otros más. Ofrecer comprobantes y/o facturas de manera digital con toda la información necesaria detallada en este mismo apartado. Registrar datos de los torneos anuales y permitir inscripción de equipos desde la API en un futuro.”

# PLANIFICACION SCRUM

## TEAM:

Ambos integrantes formaremos parte del equipo de desarrollo, colaborando activamente en la construcción de la aplicación.

## ROLES PRINCIPALES:

### **Anthony Escobar:**

Product Owner, guiando la visión del producto y priorizando el backlog.

Scrum Master, quien facilitará las ceremonias SCRUM y ayudará a resolver obstáculos.

### **Aníbal Herrera:**

Desarrollador Principal, Desarrollador principal de Frontend de nuestra API web y developer encargado de estructuración final.

# **SPRINTS:**

## **SPRINT 1 (SEMANA 1): ANÁLISIS Y DISEÑO**

**Objetivo:** Establecer bases técnicas y funcionales del proyecto.

**Tareas Clave:**

### **1. Reunión inicial con stakeholders:**

- Identificar necesidades del profesor encargado y estudiantes.
- Documentar requisitos funcionales (reservas, pagos, noticias) y no funcionales (seguridad, usabilidad).

### **2. Investigación tecnológica:**

- Definir stack MERN (MongoDB, Express, React, Node.js).
- Seleccionar APIs de pago (Stripe/PayPal) y librerías (Axios, Mongoose, Bootstrap).

### **3. Diseño de base de datos:**

- Esquemas para:
  - Usuarios (estudiantes, profesores, administradores).
  - Reservas (fecha, hora, estado de pago).
  - Torneos (equipos, fechas, pagos).

### **4. Prototipado de interfaz:**

- Bocetos en Figma o papel: calendario de reservas, formulario de pago, panel de noticias.

### **5. Configuración inicial:**

- Crear repositorio Git (estructura de carpetas para frontend/backend).
- Instalar dependencias básicas (Express, React, Mongoose).

## **SPRINT 2 (SEMANA 2): BACKEND Y AUTENTICACIÓN**

**Objetivo:** Implementar lógica central del backend y sistema de usuarios.

**Tareas Clave:**

### **1. Desarrollo de API con Express:**

- Rutas para:
  - Autenticación (registro/login con JWT).
  - CRUD de reservas (crear, consultar, cancelar).
- Middlewares:
  - Validación de horarios (evitar solapamientos).
  - Protección de rutas por rol (estudiante vs. administrador).

### **2. Integración con MongoDB:**

- Conexión a Atlas o local.
- Operaciones CRUD para usuarios y reservas.

### **3. Pruebas iniciales:**

- Postman: Verificar endpoints de autenticación y reservas.
- Validar reglas de negocio (ej: un usuario no puede reservar más de 2 horas consecutivas).

## **SPRINT 3 (SEMANA 3): FRONTEND Y RESERVAS**

**Objetivo:** Construir interfaz de usuario y conectar con backend.

**Tareas Clave:**

### **1. Desarrollo de componentes en React:**

- Calendario interactivo (mostrar horarios ocupados/libres con react-big-calendar).
- Formulario de reserva (selección de fecha, hora y confirmación).

### **2. Integración con backend:**

- Llamadas API con Axios (fetch de horarios, envío de reservas).

### **3. Pagos en línea (sandbox):**

- Integración con Stripe/PayPal para simulacro de transacciones.

### **4. Generación de comprobantes:**

- PDF con datos de reserva (usuario, fecha, monto) mediante pdfkit.

## **SPRINT 4 (SEMANA 4): NOTICIAS Y TORNEOS**

**Objetivo:** Implementar módulos secundarios pero esenciales.

**Tareas Clave:**

### **1. Sección de noticias:**

- CRUD para administradores (publicar torneos, anuncios).
- Visualización en frontend (cards con fechas importantes).

### **2. Módulo de torneos:**

- Inscripción de equipos (formulario con datos de participantes).
- Relación con reservas (asignar cancha para partidos).

### **3. Mejoras de UI/UX:**

- Feedback de usuarios testeadores (ajustes en navegación).

## **SPRINT 5 (SEMANA 5): PRUEBAS Y DESPLIEGUE**

**Objetivo:** Estabilizar el producto y preparar entrega final.

**Tareas Clave:**

### **1. Pruebas integrales:**

- Manuales: Flujo completo de reserva-pago-comprobante.
- Automatizadas: Jest para backend (validar lógica de horarios).

### **2. Optimización:**

- Rendimiento de calendario (carga rápida de fechas).
- Seguridad: Validar tokens JWT y sanitización de inputs.

### **3. Despliegue local:**

- Configurar servidor interno de KINAL o Heroku para demo.

### **4. Documentación:**

- Manual de usuario (pasos para reservar, pagar, consultar torneos).
- Documentación técnica (estructura de API, variables de entorno).

**Revisión y entrega de avances cada 4 días para corrección de errores y validación de requerimientos a cumplir a continuación de manera previa a cada SPRINT.**



# PLANIFICACION DETALLADA

## Tecnologías Requeridas

1. Backend: Nodejs, Express ,JavaScript
2. Frontend: React,Vite, Bootstrap,CSS
3. Base de Datos: MongoDB
4. Autenticación: JWT para la gestión de sesiones de usuario
5. Herramientas de Gestión: GitHub

## Dependencias:

### BACKEND:

- express
- helmet
- cors
- morgan
- dotenv
- jsonwebtoken
- mongoose
- argon2
- express-validator
- swagger-jsdoc
- swagger-ui-express
- express-rate-limit

### FRONTEND:

- react-router-dom
- react-hot-toast
- axios
- prop-types

### **Definir endpoints básicos:**

- / kinalfootballfield/auth: Registro y login de usuarios.
- / kinalfootballfield /user: Gestión de usuarios (estudiantes, administradores).
- / kinalfootballfield /event: gestión de eventos de competencia en la canchas.
- / kinalfootballfield /field: manejo de canchas disponible sy apartados existentes dentro de la misma.
- / kinalfootballfield /reservation: gestión de reservaciones para el apartado de canchas por usuario y profesor.
- / kinalfootballfield /payment: gestión de tarjetas a utilizar para la realización del pago del apartado de canchas.
- **Revisiones Técnicas (Cada 3 días)**
  - **Día 3:** Revisión de modelos y relaciones (Usuario, Canchas, Eventos, Reservaciones y pagos).
  - **Día 6:** Revisión de configuración inicial (Express, conexión MongoDB, estructura de rutas).

# SEMANA 1-2:

## SPRINT 1 (SEMANA 1): AUTENTICACIÓN Y USUARIOS

**Objetivo:** Configurar sistema de autenticación y gestión de usuarios

**Tareas:**

1. Implementar modelo User con:

- Campos: name, phone, email, password, role (STUDENT/ADMINISTRATOR), academic, status
- Validaciones: email único, phone (8 caracteres)

2. Desarrollar controladores para:

- Registro (hash de contraseña con Argon2)
- Login (generación de JWT)
- Perfil de usuario (visualización/edición)

3. Middleware de autenticación JWT

- Endpoints funcionales: /auth/register, /auth/login, /users/profile
- Protección de rutas por rol

## **SPRINT 2 (SEMANA 2): GESTIÓN DE CANCHAS Y EVENTOS**

**Objetivo:** Implementar CRUD de canchas y eventos

**Tareas:**

1. Modelo Field:

- Campos: name (único), description, status

2. Modelo Event:

- Campos: name (único), description, dateFrom, dateTo, status

3. Controladores para:

- Crear/editar/eliminar canchas (solo ADMIN)
- Publicar eventos (torneos, mantenimientos)
- Endpoints: /fields, /events
- Validación de fechas para eventos

## SEMANA 3-4:

### SPRINT 3 (SEMANA 3): RESERVAS Y PAGOS

**Objetivo:** Sistema de reservas con integración de pagos

**Tareas:**

1. Modelo Reservation:

- Relaciones: student, payment, teacher
- Campos: date, timeRange, status

2. Modelo Payment:

- Validaciones: tarjeta válida (VISA/MASTERCARD), fecha futura

3. Controladores para:

- Reservas con verificación de disponibilidad
- Registro de tarjetas (enmascaramiento de datos)
- Endpoints: /reservations, /payments
- Integración con bancos locales (BAC, BAM, etc.)

## **SPRINT 4 (SEMANA 4): FRONTEND BÁSICO**

**Objetivo:** Interfaces para usuarios y administradores

**Tareas:**

### **1. Vistas en React:**

- Login/Registro
- Dashboard de reservas (calendario)
- Panel ADMIN (gestión de canchas/eventos)

### **2. Conexión con API:**

- Autenticación JWT en localStorage
- Fetch de reservas/eventos
- UI funcional para estudiantes y administradores
- Integración con endpoints existentes

# SEMANA 5:

## SPRINT 5 (SEMANA 5): PRUEBAS Y DESPLIEGUE

**Objetivo:** Estabilizar sistema para producción garantizando todas las funcionalidades clave operativas y estables.

### TAREAS PRINCIPALES

#### 1. Pruebas Exhaustivas

- **Backend (Postman):**
  - Verificar el 100% de los endpoints:
    - Autenticación (login/registro).
    - Gestión de canchas (creación/edición).
    - Reservas (disponibilidad, pagos).
    - Eventos/torneos.
  - Validar reglas de negocio:
    - Límite de reservas por estudiante.
    - Horarios permitidos (9:00 AM - 7:00 PM).
    - Cancelaciones con 2h de anticipación.
- **Frontend (Usabilidad):**
  - Pruebas con estudiantes reales:
    - Flujo completo: Reserva → Pago → Confirmación.
    - Visualización de eventos/torneos.
  - Detección de errores en consola y responsive design (móvil/escritorio).

#### 2. Despliegue

- **Backend:**
  - Vercel o servidor interno de KINAL.
  - Configurar variables de entorno (JWT\_SECRET, MongoDB Atlas).
- **Frontend:**
  - Firebase Hosting con dominio institucional.
  - Asegurar HTTPS y CORS.

#### 3. Documentación

- **Técnica:**
  - Especificación de API (endpoints, modelos, ejemplos).
  - Instrucciones para despliegue.
- **Usuario:**
  - Manual para estudiantes (cómo reservar).
  - Guía para administradores (gestión de canchas/eventos).

## HERRAMIENTAS ADICIONALES

- **Librerías:**
  - **Bootstrap/Tailwind:** Para garantizar diseño responsive.
  - **React Big Calendar:** Visualización de reservas.
- **Gestión:**
  - **Reuniones diarias:** 8:00 PM vía WhatsApp (avances/bloqueos).
  - **Revisiones técnicas:** Cada 3 días (ej: martes y viernes).

---

## CRITERIOS DE ACEPTACIÓN

- **Backend:**
  - 100% de endpoints probados en Postman.
  - Validación de roles y horarios funcional.
- **Frontend:**
  - Sin errores en consola.
  - Paneles diferenciados (Estudiante/Administrador).
- **Despliegue:**
  - Accesible via URL institucional.
  - Integración con pasarela de pagos (Stripe) operativa.

---

## CEREMONIAS SCRUM

- **Daily Stand-ups:** 8:00 PM (WhatsApp).
  - ¿Qué hice ayer?
  - ¿Qué haré hoy?
  - ¿Bloqueos?
- **Sprint Planning:** Lunes (asignar tareas para la semana).
- **Sprint Review:** Viernes (demostrar avances al Product Owner).
- **Retrospectiva:** Viernes post-review (mejoras para próximos sprints).