PLAN DE DESARROLLO DETALLADO

Sistema de Inventarios y Seguimiento - Telmex

Stack: React + Node.js + PostgreSQL

📋 FASES DE DESARROLLO

FASE 1: ANÁLISIS Y SETUP (Semanas 1-3)

Semana 1: Análisis de Requerimientos

🎯 Objetivos:
- Entender completamente el proceso actual
- Definir casos de uso específicos
- Identificar usuarios del sistema
- Documentar flujos de trabajo

📋 Actividades:
- Entrevistar al personal del área
- Mapear proceso actual (bitácora → Excel)
- Definir tipos de piezas y categorías
- Identificar información de envíos requerida
- Crear user stories y acceptance criteria

🎁 Entregables:
- Documento de requerimientos funcionales
- Diagrama de procesos actual vs propuesto
- Lista de user stories priorizadas

Semana 2: Diseño de Sistema

🎯 Objetivos:
- Diseñar arquitectura del sistema
- Modelar base de datos
- Crear mockups de interfaces
- Definir APIs necesarias

📋 Actividades:
- Diseño de entidad-relación (ER)
- Normalización de base de datos
- Wireframes de pantallas principales
- Definición de endpoints REST
- Selección de paleta de colores corporativa

🎁 Entregables:
- Diagrama ER de base de datos
- Mockups de interfaces principales
- Documentación de API endpoints
- Arquitectura del sistema

Semana 3: Setup del Entorno

🎯 Objetivos:
- Configurar entorno de desarrollo
- Preparar estructura de proyecto
- Setup de base de datos
- Configurar herramientas

📋 Actividades:
- Instalar Node.js, npm, PostgreSQL
- Crear repositorios Git (frontend/backend)
- Setup inicial React + Material-UI
- Setup inicial Express.js
- Configurar VS Code con extensiones
- Crear base de datos inicial

🎁 Entregables:
- Proyecto inicial funcionando
- Base de datos con estructura básica
- Documentación de setup

FASE 2: DESARROLLO BACKEND (Semanas 4-7)

Semana 4: Base de Datos y Modelos

🎯 Objetivos:
- Implementar esquema de base de datos
- Crear modelos de datos
- Setup de migraciones
- Datos de prueba

📋 Actividades:
- Crear tablas: usuarios, piezas, inventario, envios, seguimiento
- Implementar relaciones y constraints
- Crear seeders con datos de prueba
- Setup de pool de conexiones
- Implementar modelo de auditoría

🔧 Tecnologías:
- PostgreSQL
- pg (driver PostgreSQL)
- Sequelize ORM (opcional)

📁 Estructura sugerida:
```
/backend
  /models
  /migrations
  /seeders
  /config
```

Semana 5: APIs de Inventario

🎯 Objetivos:
- Desarrollar CRUD de inventarios
- Implementar validaciones
- Manejo de errores
- Testing básico

📋 Actividades:
- GET /api/inventario (listar con paginación)
- POST /api/inventario (crear entrada)
- PUT /api/inventario/:id (actualizar)
- DELETE /api/inventario/:id (dar de baja)
- GET /api/inventario/search (búsqueda)
- Validación de datos de entrada
- Middleware de manejo de errores

🧪 Testing:
- Crear colección Postman
- Probar todos los endpoints
- Validar respuestas y códigos de estado

Semana 6: APIs de Envíos y Seguimiento

🎯 Objetivos:
- Desarrollar funcionalidad de envíos
- Sistema de seguimiento
- Generación de códigos de rastreo
- Estados de envío

📋 Actividades:
- POST /api/envios (crear envío)
- GET /api/envios (listar envíos)
- PUT /api/envios/:id/status (actualizar estado)
- GET /api/envios/:codigo/tracking (seguimiento)
- Sistema de códigos únicos
- Estados: Pendiente, En tránsito, Entregado, Devuelto

🔧 Funcionalidades especiales:
- Generador de códigos de rastreo
- Historial de cambios de estado
- Notificaciones por email (opcional)

Semana 7: Autenticación y Reportes

🎯 Objetivos:
- Sistema de login/logout
- Protección de rutas
- APIs de reportes
- Logs de auditoría

📋 Actividades:
- POST /api/auth/login
- POST /api/auth/logout
- JWT middleware
- GET /api/reportes/inventario
- GET /api/reportes/envios
- GET /api/reportes/dashboard
- Sistema de logs

🔒 Seguridad:
- Hash de contraseñas con bcrypt
- Validación y sanitización
- Rate limiting
- CORS configurado

FASE 3: DESARROLLO FRONTEND (Semanas 8-11)

Semana 8: Setup React y Navegación

🎯 Objetivos:
- Estructura base de React
- Sistema de navegación
- Layout principal
- Componentes base

📋 Actividades:
- Setup React Router
- Crear layout principal con sidebar
- Implementar header con navegación
- Crear componentes reutilizables
- Setup de Material-UI theme
- Responsive design base

📁 Estructura sugerida:
```
/frontend/src
  /components
  /pages
  /services
  /utils
  /styles
```

Semana 9: Pantallas de Inventario

🎯 Objetivos:
- Lista de inventario
- Formularios CRUD
- Búsqueda y filtros
- Paginación

📋 Actividades:
- Página lista de inventario con tabla
- Modal/página para agregar pieza
- Modal/página para editar pieza
- Barra de búsqueda
- Filtros por categoría, estado
- Paginación con Material-UI
- Validación de formularios

🎨 Componentes Material-UI:
- DataGrid o Table
- TextField, Select
- Button, IconButton
- Dialog, Snackbar
- Pagination

Semana 10: Pantallas de Envíos

🎯 Objetivos:
- Gestión de envíos
- Seguimiento de paquetes
- Actualización de estados
- Timeline de eventos

📋 Actividades:
- Lista de envíos con filtros
- Formulario crear nuevo envío
- Página de detalle de envío
- Timeline de seguimiento
- Actualización de estados
- Búsqueda por código de rastreo

🎨 Componentes especiales:
- Stepper para estados de envío
- Timeline para historial
- Chips para estados
- Cards para información de envío

Semana 11: Dashboard y Reportes

🎯 Objetivos:
- Dashboard principal
- Gráficos estadísticos
- Generación de reportes
- Exportación de datos

📋 Actividades:
- Dashboard con métricas principales
- Gráficos de inventario por categoría
- Gráficos de envíos por estado/tiempo
- Reportes exportables (PDF/Excel)
- Filtros por fecha
- Widgets informativos

📊 Librerías de gráficos:
- Chart.js o Recharts
- Material-UI DataGrid para tablas
- jsPDF para exportar PDF
- xlsx para exportar Excel

FASE 4: INTEGRACIÓN Y TESTING (Semanas 12-14)

Semana 12: Integración Frontend-Backend

🎯 Objetivos:
- Conectar frontend con APIs
- Manejo de estados global
- Loading states
- Error handling

📋 Actividades:
- Setup Axios interceptors
- Context API o Redux para estado global
- Componentes de Loading
- Manejo de errores con Snackbars
- Validación del flujo completo
- Optimización de requests

🔧 Herramientas:
- Axios para HTTP requests
- Context API para estado global
- React Hook Form para formularios

Semana 13: Testing y Depuración

🎯 Objetivos:
- Testing exhaustivo
- Corrección de bugs
- Optimización de rendimiento
- Validación de seguridad

📋 Actividades:
- Testing manual de todos los flujos
- Casos edge cases
- Testing de rendimiento
- Validación de seguridad
- Optimización de consultas BD
- Code review y refactoring

🧪 Testing checklist:
- CRUD completo de inventario
- Proceso completo de envíos
- Autenticación y autorización
- Responsive design
- Validaciones de formularios

Semana 14: Despliegue y Documentación

🎯 Objetivos:
- Deploy en producción
- Documentación técnica
- Manual de usuario
- Capacitación del personal

📋 Actividades:
- Setup servidor de producción
- Configurar base de datos producción
- Deploy con variables de entorno
- SSL/HTTPS configurado
- Manual técnico
- Manual de usuario
- Video tutoriales (opcional)

🚀 Plataformas de deploy:
- Backend: Heroku, DigitalOcean, Railway
- Frontend: Netlify, Vercel
- Base de datos: Heroku Postgres, DigitalOcean

FASE 5: EVALUACIÓN Y CIERRE (Semanas 15-16)

Semana 15: Evaluación y Métricas

🎯 Objetivos:
- Medición de resultados
- Comparación con objetivos
- Feedback de usuarios
- Métricas de rendimiento

📋 Actividades:
- Recopilar métricas de uso
- Encuestas de satisfacción
- Comparar tiempos antes/después
- Análisis de errores reducidos
- Documentar lecciones aprendidas

Semana 16: Presentación Final

🎯 Objetivos:
- Presentación ejecutiva
- Demostración del sistema
- Entrega de documentación
- Plan de mantenimiento

📋 Actividades:
- Preparar presentación PowerPoint
- Demo en vivo del sistema
- Entrega de código fuente
- Plan de mantenimiento
- Recomendaciones futuras

🛠️ HERRAMIENTAS DE DESARROLLO DIARIAS

Setup Inicial Requerido

```bash
# Backend
npm init -y
npm install express cors helmet dotenv bcryptjs jsonwebtoken
npm install pg sequelize
npm install -D nodemon

# Frontend
npx create-react-app telmex-inventory
npm install @mui/material @emotion/react @emotion/styled
npm install @mui/icons-material @mui/x-data-grid
npm install axios react-router-dom
npm install chart.js react-chartjs-2
```

Estructura de Carpetas Final

```
proyecto-telmex/
├── backend/
│   ├── controllers/
│   ├── models/
│   ├── routes/
│   ├── middleware/
│   ├── config/
│   └── server.js
├── frontend/
│   ├── public/
```

```
│   └── src/
│       ├── components/
│       ├── pages/
│       ├── services/
│       └── utils/
└── docs/
    ├── technical/
    ├── user-manual/
    └── presentation/
```

📊 MÉTRICAS DE ÉXITO

Objetivos Cuantificables

- ✅ Reducir tiempo de registro: **de 15 min → 3 min** (80% mejora)
- ✅ Eliminar errores de transcripción: **100% → 0%**
- ✅ Tiempo de consulta: **de 5 min → 30 seg** (90% mejora)
- ✅ Generación de reportes: **de 2 horas → 5 min** (95% mejora)

Funcionalidades Mínimas Viable (MVP)

1. ✅ Login de usuarios
2. ✅ CRUD de inventario
3. ✅ Crear y rastrear envíos
4. ✅ Dashboard básico
5. ✅ Reportes exportables

¿Te parece bien este plan? ¿Hay alguna fase o semana específica en la que te gustaría que profundice más?