Manual del Sistema - Club*Enduro*

# Descripción del Proyecto

Este sistema web permite a los miembros de un club de enduro realizar aportes económicos voluntarios, los cuales son administrados por otros miembros “Tesoreros” o por un administrador. El fondo recaudado se utiliza para cubrir gastos comunes, tales como el mantenimiento de las rutas (picas), la compra de herramientas (ej. machetes para el personal de limpieza), entre otros. El sistema lleva un control de depósitos, gastos y miembros.

# Tecnologías Usadas

|  |  |
| --- | --- |
| Categoría | Tecnología |
| Frontend | HTML5, CSS3, TailwindCSS, JavaScript |
| Backend | Node.js, Express.js |
| Base de Datos | MySQL |
| ORM | Sequelize |
| Entorno | dotenv |
| Logging | morgan |

# Estructura de Carpetas

club-enduro/  
│  
├── frontend/  
│ ├── index.html  
│ ├── css/  
│ │ └── styles.css  
│ ├── js/  
│ │ └── app.js  
├── backend/  
│ ├── controllers/  
│ │ ├── depositosController.js  
│ │ └── gastosController.js  
│ ├── models/  
│ │ ├── deposito.js  
│ │ └── gasto.js  
│ ├── routes/  
│ │ ├── depositosRoutes.js  
│ │ └── gastosRoutes.js  
│ ├── config/  
│ │ └── conexDb.js  
│ └── link.js  
├── .env  
├── package.json  
└── README.md

# Configuración del Entorno

Archivo .env

PORT=3000  
DB\_HOST=localhost  
DB\_USER=root  
DB\_PASSWORD=””   
DB\_NAME=club\_enduro

Migraciones y Seeders

Migrar tablas: npx sequelize-cli db:migrate

Insertar datos iniciales: npx sequelize-cli db:seed:all

# Endpoints Importantes (API REST)

|  |  |  |
| --- | --- | --- |
| Método | Endpoint | Descripción |
| GET | /api/depositos | Lista todos los depósitos |
| POST | /api/depositos | Registrar un nuevo depósito |
| GET | /api/gastos | Lista todos los gastos realizados |
| POST | /api/gastos | Registrar un nuevo gasto |
| GET | /api/miembros | Lista de miembros del club |
| POST | /api/miembros | Registrar un nuevo miembro |

# Procedimientos del Sistema

## Backup de la Base de Datos

mysqldump -u root -p club\_enduro > backup\_club\_enduro.sql

Recomendación: configurar un cron job para realizar respaldos periódicos.

## Registro de Logs

Instala morgan con: npm install morgan

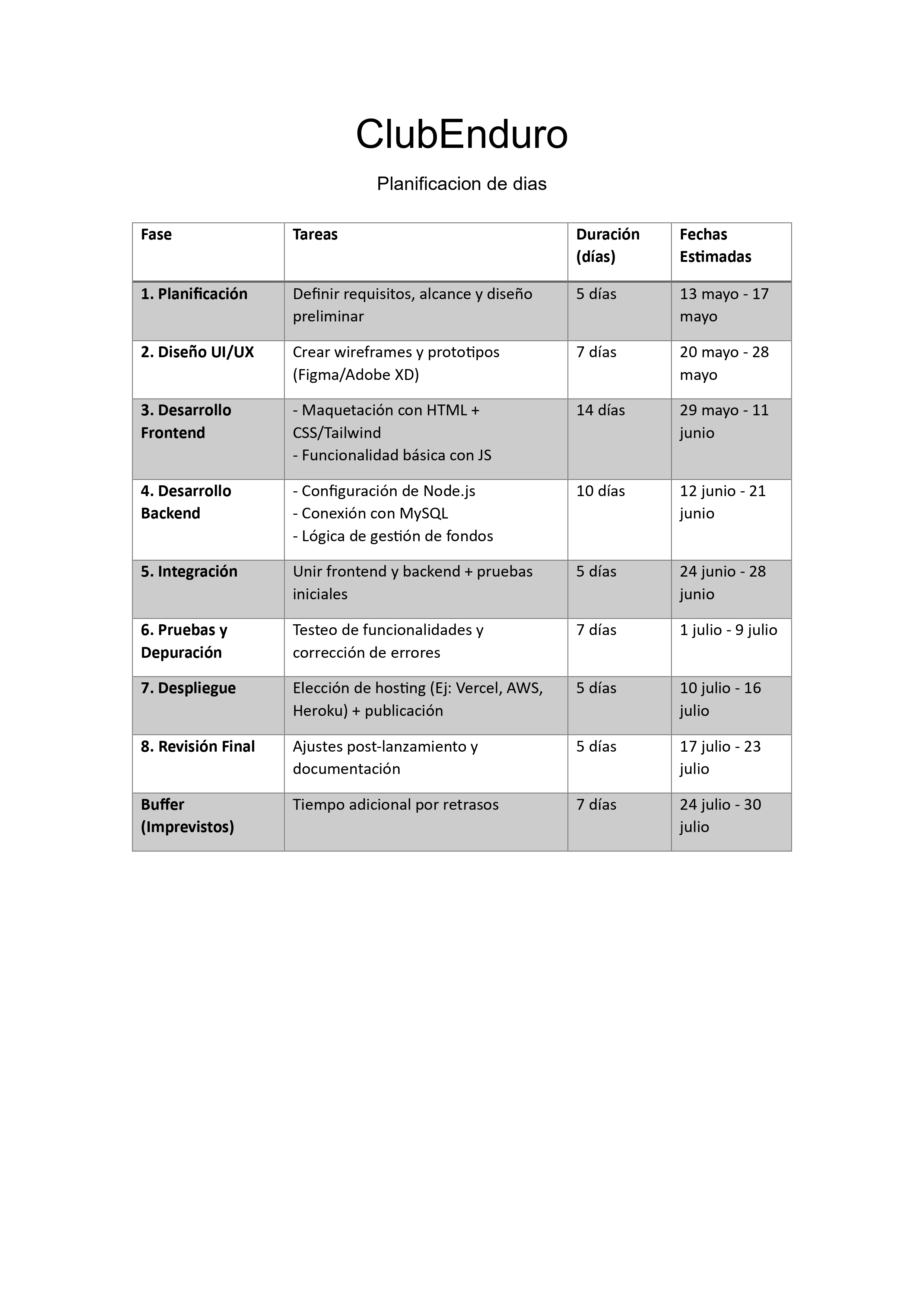
Y configúralo en server.js:  
const morgan = require('morgan');  
app.use(morgan('dev'));

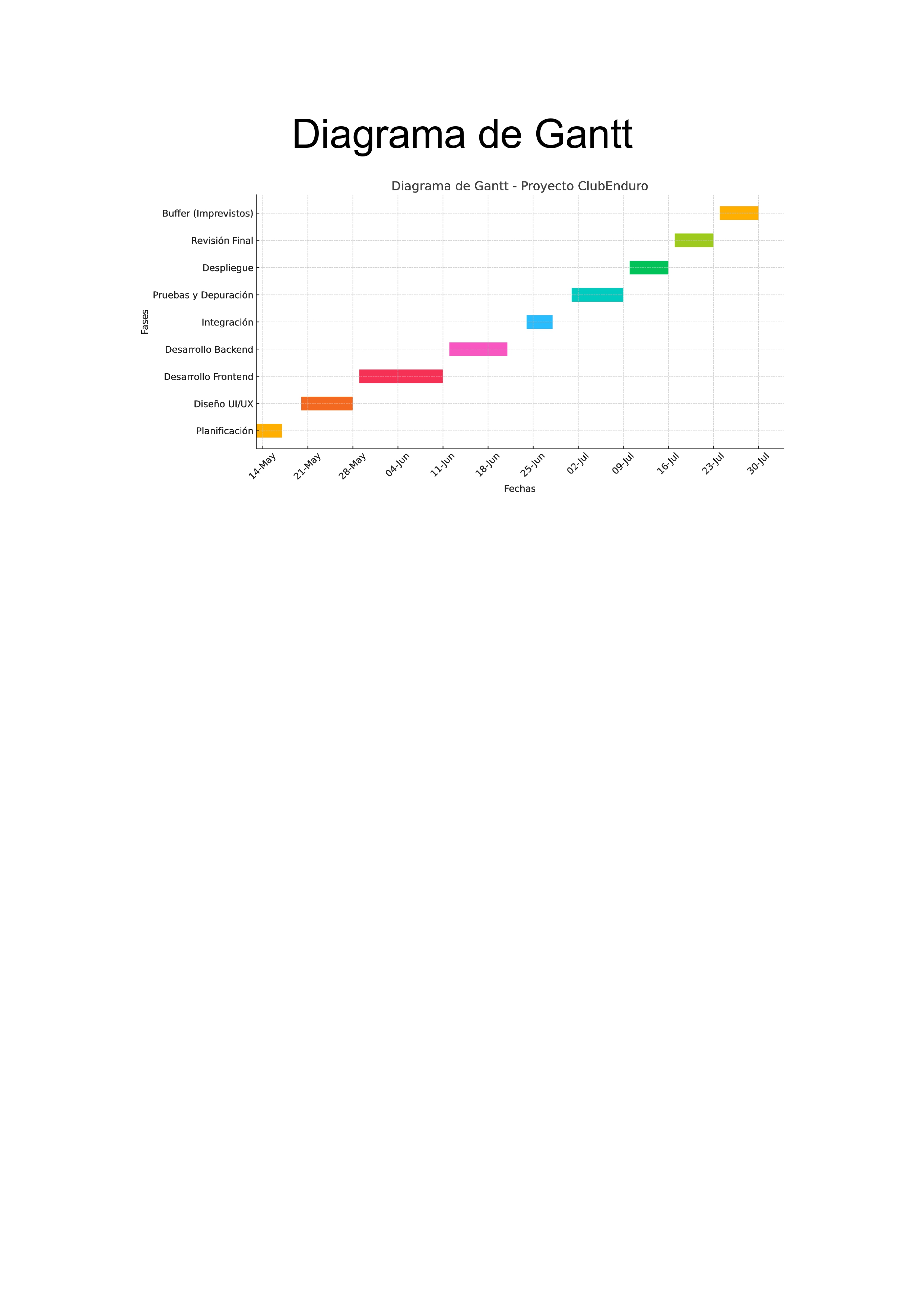
## Actualizaciones del Sistema

1. Actualizar dependencias: npm update  
2. Actualizar base de datos: npx sequelize-cli db:migrate  
3. Registrar cambios en CHANGELOG.md  
4. Notificar al equipo si hay cambios estructurales

Anexos

Diagrama de Gantt





Modelo entidad relación (MER)

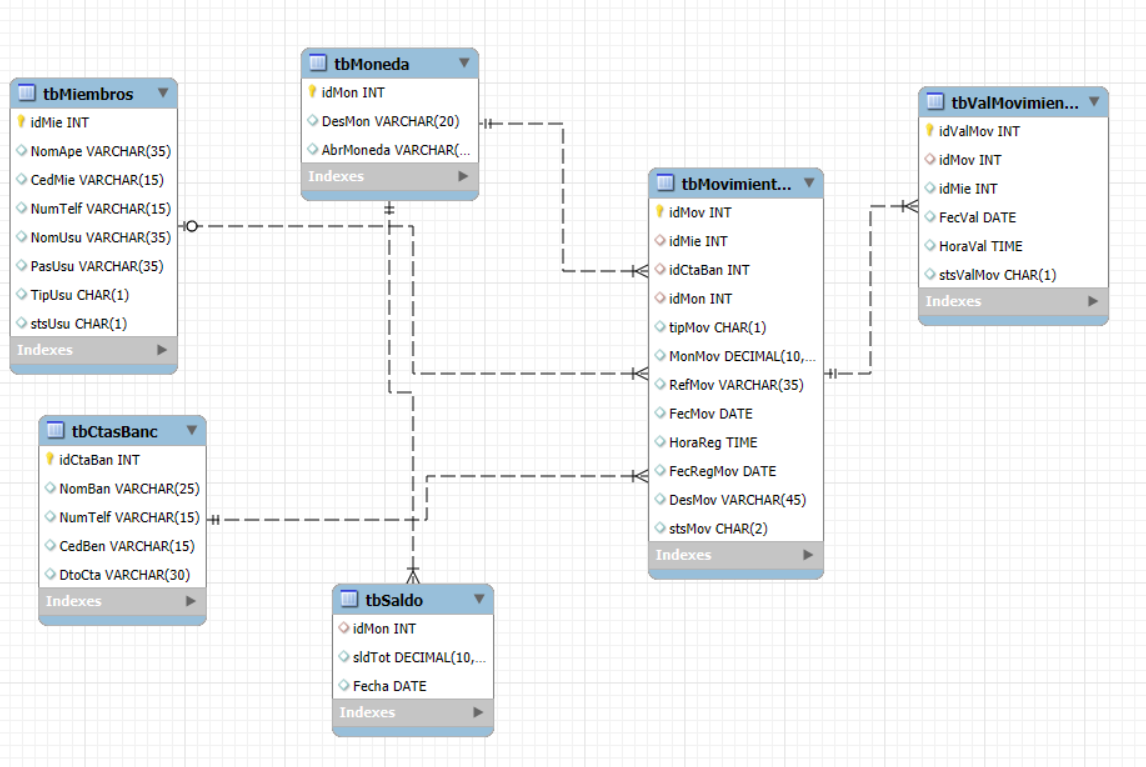
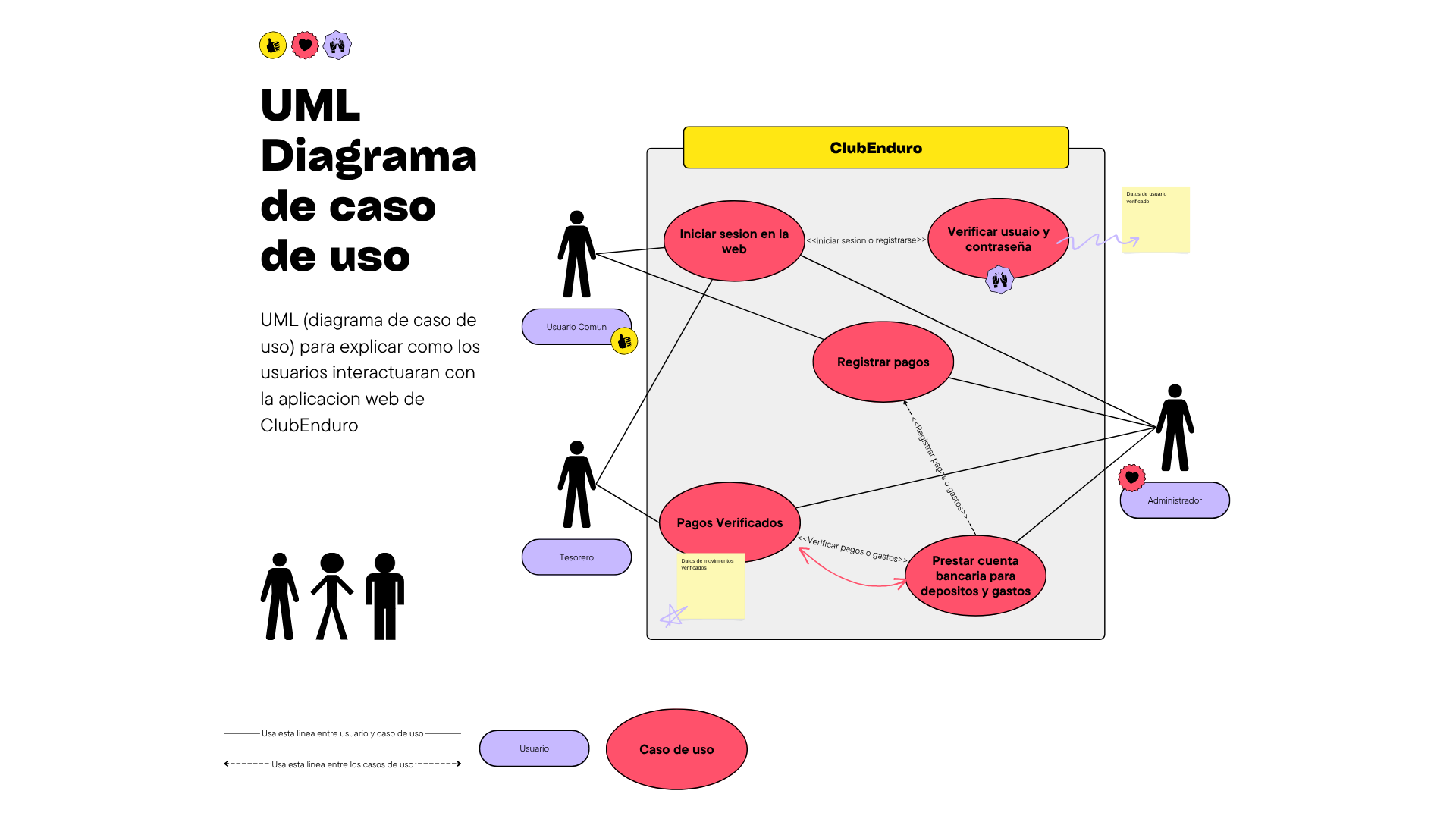


Diagrama de caso de uso (UML)



Carta de navegación (Site Map)