

Administrador de Licencias

Sistema de gestión de licencias desarrollado con Next.js 14, TypeScript, Prisma y Tailwind CSS.

Tecnologías Utilizadas

- **Framework:** Next.js 14 con App Router
- **Lenguaje:** TypeScript
- **Base de Datos:** Prisma ORM
- **Estilos:** Tailwind CSS
- **UI Components:** Radix UI + shadcn/ui
- **Gráficos:** Recharts
- **Gestión de Estado:** SWR
- **Iconos:** Lucide React

Instalación

```
# Clonar el repositorio
git clone [url-del-repositorio]
cd license-administrator

# Instalar dependencias
pnpm install

# Configurar variables de entorno
cp .env.example .env

# Configurar base de datos
npx prisma generate
npx prisma db push
npx prisma db seed

# Ejecutar en desarrollo
pnpm dev
```

Estructura del Proyecto

```
src/
├── app/                                # App Router
│   ├── api/                            # API Routes
│   │   ├── dashboard/stats/           # Estadísticas
│   │   ├── licenses/                  # CRUD licencias
│   │   └── users/                     # CRUD usuarios
│   ├── dashboard/                     # Páginas dashboard
│   ├── globals.css                    # Estilos globales
│   └── components/                    # Componentes
```

```
├── ui/                # UI base
├── overview/          # Dashboard
├── users/              # Usuarios
├── licenses/          # Licencias
├── hooks/             # Custom hooks
├── lib/               # Utilidades
└── test/              # Testing
```

Funcionalidades

Dashboard

- Estadísticas en tiempo real
- Gráficos de uso mensual
- Alertas de vencimiento
- Métricas de rendimiento

Gestión de Usuarios

- CRUD completo de usuarios
- Asignación de licencias
- Estados activo/inactivo
- Departamentos y roles

Gestión de Licencias

- Catálogo de licencias
- Información de costos
- Fechas de expiración
- Asignaciones por departamento

API Endpoints

Dashboard Stats

```
GET /api/dashboard/stats
```

Retorna estadísticas completas del dashboard.

Usuarios

```
GET /api/users      # Listar usuarios
POST /api/users     # Crear usuario
```

Licencias

```
GET /api/licenses      # Listar licencias
POST /api/licenses     # Crear licencia
GET /api/licenses/costs # Información costos
```

Componentes Principales

Dashboard

- **OverviewHeader**: Navegación superior
- **StatsChart**: Gráfico principal de líneas
- **BarStatsSummary**: Gráfico de barras
- **MostUsedLicenses**: Top licencias utilizadas

Cards de Resumen

- **UsersCard**: Total usuarios nuevos
- **LicenseCard**: Total licencias
- **ActiveUsersCard**: Usuarios activos
- **ExpiringSoonCard**: Próximas a vencer

Tablas

- **UsersTable**: Gestión de usuarios
- **LicenseTable**: Gestión de licencias
- **CostosTable**: Información financiera

Custom Hooks

```
// Estadísticas del dashboard
const { stats, isLoading, error } = useStats()

// Gestión de usuarios
const { users, isLoading, isError } = useUsers()

// Manejo de licencias
const { license, error, isLoading } = useLicenses()

// Información de costos
const { costs, isLoading, error, refetch } = useCosts()
```

Sistema de Estilos

Variables CSS Personalizadas

```
--color-primary: oklch(0.145 0 0);
--color-secondary: #9ac4fe;
```

```
--color-sidebarBg: #f9f9f9;  
--color-borderPrimary: #e5e5e5;
```

Componentes UI

- Botones con variantes
- Cards informativos
- Inputs de formulario
- Barras de progreso
- Componentes de gráficos

Páginas

- / - Login principal
- /dashboard - Dashboard principal
- /dashboard/users - Gestión usuarios
- /dashboard/licenses - Gestión licencias
- /dashboard/reports - Reportes
- /dashboard/assignments - Asignaciones
- /dashboard/contracts - Contratos
- /dashboard/settings - Configuración

Testing

```
# Scripts de prueba disponibles  
node src/test/test-users.mjs      # API usuarios  
node src/test/test-licenses.mjs   # API licencias  
node src/test/testDb.mjs          # Conexión DB
```

Docker

```
FROM node:20-alpine  
WORKDIR /app  
COPY package*.json ./  
RUN npm install  
COPY . .  
RUN npx prisma generate  
CMD ["npm", "run", "dev"]
```

Scripts

```
pnpm dev      # Desarrollo  
pnpm build    # Build producción
```

```
pnpm start      # Servidor producción
pnpm lint       # Linting
```

Métricas Rastreadas

- Total de licencias por proveedor
- Usuarios activos y nuevos
- Licencias próximas a expirar (30 días)
- Uso mensual por usuario
- Costos e instalaciones

Utilidades

Función **cn**

```
export function cn(...inputs: ClassValue[]) {
  return twMerge(clsx(inputs))
}
```

Cliente Prisma

Configurado para desarrollo y producción con pooling de conexiones.

Seguridad

- Validación de tipos con TypeScript
- Sanitización de inputs
- Manejo seguro de errores
- Variables de entorno protegidas

Responsive Design

- Diseño mobile-first
- Breakpoints optimizados
- Navegación adaptativa
- Componentes flexibles

Accesibilidad

- Componentes Radix UI
- Navegación por teclado
- ARIA labels
- Alto contraste

Próximos Pasos

- ☐ Implementar autenticación completa


- ☐ Añadir más proveedores de licencias
- ☐ Sistema de notificaciones push
- ☐ Exportación de reportes PDF
- ☐ API REST completa
- ☐ Tests automatizados

Licencia

Este proyecto está bajo la Licencia MIT.

Contribución

1. Fork el proyecto
2. Crear rama feature (`git checkout -b feature/AmazingFeature`)
3. Commit cambios (`git commit -m 'Add some AmazingFeature'`)
4. Push a la rama (`git push origin feature/AmazingFeature`)
5. Abrir Pull Request

Desarrollado con  usando Next.js y TypeScript