

EducaGestor360 - Resumen del Proyecto

Visión General

EducaGestor360 es un sistema integral de gestión educativa diseñado para modernizar y optimizar los procesos administrativos y académicos de instituciones educativas. El proyecto busca proporcionar una solución completa que mejore la eficiencia operativa, la calidad educativa y la experiencia de todos los stakeholders involucrados en el proceso educativo.

Objetivo Principal

Desarrollar una plataforma de software robusta y escalable que simplifique y mejore la gestión administrativa y académica de instituciones educativas, facilitando la toma de decisiones y optimizando los procesos educativos a través de herramientas digitales integradas.

Alcance del Proyecto

Módulos Principales

1. **Gestión de Estudiantes**

- Registro y matrícula de estudiantes
- Seguimiento académico
- Gestión de asistencia
- Visualización de calificaciones
- Historial académico

2. **Gestión de Docentes**

- Registro y administración de docentes
- Asignación de cursos
- Gestión de calificaciones
- Creación y gestión de recursos educativos
- Control de asistencia

3. **Gestión de Cursos**

- Creación y configuración de cursos
- Asignación de docentes
- Gestión de recursos educativos
- Control de cupos y horarios
- Seguimiento de progreso

4. **Gestión Administrativa**

- Administración de usuarios y roles
- Gestión de instituciones educativas
- Reportes y estadísticas
- Configuración del sistema

Características Principales

- Interfaz intuitiva y responsive
- Sistema de roles y permisos
- Generación de reportes personalizados
- Notificaciones automáticas
- Integración con sistemas externos
- Respaldo automático de datos
- Soporte multiidioma

Arquitectura Técnica

Stack Tecnológico

- **Backend**: Java con Spring Boot
- **Frontend**: React.js

- **Base de Datos**: PostgreSQL
- **Autenticación**: JWT (JSON Web Tokens)
- **API**: RESTful
- **Control de Versiones**: Git

Estructura de la Base de Datos

El sistema utiliza una base de datos relacional con las siguientes entidades principales:

- Institución
- Estudiante
- Docente
- Curso
- Matrícula
- Calificación
- Asistencia
- Recurso Educativo

Flujos de Trabajo Principales

1. **Proceso de Matrícula**

```
```mermaid
```

```
graph TD
```

```
 A[Estudiante solicita matrícula] --> B[Verificación de datos]
```

```
 B --> C[Selección de cursos]
```

```
 C --> D[Confirmación de matrícula]
```

```
 D --> E[Generación de comprobante]
```

```
```
```

2. **Gestión de Calificaciones**

```
```mermaid
```

graph TD

A[Docente ingresa calificaciones] --> B[Sistema valida datos]

B --> C[Actualización de registros]

C --> D[Notificación a estudiantes]

D --> E[Generación de reportes]

...

## ## Políticas de Desarrollo

### 1. \*\*Estándares de Código\*\*

- Seguir principios SOLID
- Documentación inline en inglés
- Nomenclatura camelCase para variables y métodos
- Pruebas unitarias obligatorias

### 2. \*\*Control de Versiones\*\*

- Rama principal: `main`
- Rama de desarrollo: `develop`
- Ramas de feature: `feature/nombre-funcionalidad`
- Ramas de hotfix: `hotfix/descripcion-fix`

### 3. \*\*Proceso de Revisión\*\*

- Code reviews obligatorios
- Pull requests documentados
- Tests automatizados antes de merge

## ## Estado Actual y Próximos Pasos

### ### Módulos Implementados

- ☒ Gestión de usuarios y roles

- ☒ Registro de estudiantes y docentes
- ☒ Gestión básica de cursos
- ☒ Sistema de matrículas

#### ### En Desarrollo

- ☐ Sistema de calificaciones
- ☐ Gestión de recursos educativos
- ☐ Reportes avanzados

#### ### Próximas Funcionalidades

- ☐ Integración con sistemas de pago
- ☐ App móvil para estudiantes
- ☐ Sistema de mensajería interna
- ☐ Portal para padres de familia

#### ## Buenas Prácticas y Recomendaciones

##### 1. **\*\*Desarrollo\*\***

- Seguir el patrón MVC
- Implementar manejo de excepciones
- Mantener la documentación actualizada
- Realizar pruebas unitarias

##### 2. **\*\*Seguridad\*\***

- Validar todas las entradas de usuario
- Implementar autenticación en dos factores
- Encriptar datos sensibles
- Mantener logs de seguridad

### 3. **\*\*Rendimiento\*\***

- Optimizar consultas a la base de datos
- Implementar caché donde sea apropiado
- Monitorear el uso de recursos
- Realizar pruebas de carga

## ## Recursos y Documentación

### ### Documentación Técnica

- Diagramas UML
- API Documentation
- Manual de desarrollo
- Guías de implementación

### ### Herramientas de Desarrollo

- IDE recomendado: IntelliJ IDEA
- Herramientas de testing: JUnit, Jest
- Herramientas de documentación: Swagger
- Control de versiones: GitLab

## ## Contactos Clave

- **\*\*Project Manager\*\***: [Nombre del PM]
- **\*\*Tech Lead\*\***: [Nombre del Tech Lead]
- **\*\*Backend Lead\*\***: [Nombre del Backend Lead]
- **\*\*Frontend Lead\*\***: [Nombre del Frontend Lead]
- **\*\*QA Lead\*\***: [Nombre del QA Lead]

## ## Conclusión

EducaGestor360 es un proyecto ambicioso que busca transformar la gestión educativa mediante una solución tecnológica integral.