SCHOOLME

ACADEMICS

Manual Tecnico Portal Web

Centro de la industria, la empresa y los servicios
ADSO 2900177

Tabla de contenido

Grupo de Desarollo SchollMe
Jesus Fernando Carvajal jesusanacona017@gmail.com
Santiago Chaparro Riaño alfasan1481@gmail.com

Manual Técnico - SchoolMe Portal

1. Introducción

El proyecto **SchoolMe Portal** es una aplicación web de gestión escolar desarrollada con Angular 19. Esta plataforma permite la administración integral de instituciones educativas, incluyendo la gestión de usuarios, estudiantes, docentes, cargas académicas, agendas y configuraciones de seguridad basadas en roles.

La aplicación está diseñada para facilitar la comunicación y organización entre directores de grupo, docentes, estudiantes y acudientes, proporcionando herramientas para el seguimiento académico y administrativo.

Características principales:

- Gestión de usuarios con roles y permisos
- Administración de estudiantes y docentes
- Asignación de cargas académicas
- Configuración de agendas y horarios
- Sistema de autenticación seguro con JWT
- Interfaz responsiva con Taiga UI y Angular Material

2. Arquitectura del Sistema

Arquitectura General

El proyecto sigue una arquitectura cliente-servidor:

- **Frontend:** Aplicación Angular SPA (Single Page Application)
- **Backend:** API REST externa alojada en `https://api.schoolme.space/api`
- **Base de Datos:** Gestionada por el backend (detalles no disponibles en este repositorio)
- **Despliegue:** Contenedorizado con Docker y Nginx

Arquitectura del Frontend

- **Framework:** Angular 19 con Standalone Components
- **UI Libraries:** Taiga UI, Angular Material, Bootstrap
- **Estado:** Gestión basada en servicios inyectables
- **Ruteo:** Angular Router con guards de autenticación
- **HTTP Client:** Interceptors para autenticación automática

Componentes Arquitectónicos

- **Módulos de Seguridad:** Autenticación, autorización y gestión de usuarios
- **Módulos de Parámetros:** Configuración de entidades básicas (estudiantes, docentes, grupos)
- **Módulos de Negocio:** Lógica específica del dominio educativo

- **Layout:** Dashboard con navegación lateral y componentes reutilizables

3. Requisitos Técnicos

Requisitos del Sistema

- **Node.js:** v20.x o superior
- **npm:** v9.x o superior (incluido con Node.js)
- **Angular CLI:** v19.x
- **Sistema Operativo:** Windows, macOS o Linux

Dependencias Principales

- **Angular:** 19.0.0 (Core, Router, Forms, CDK, Material)
- **Taiga UI:** 4.60.0 (Componentes UI principales)
- **Bootstrap:** 5.3.5 (Estilos base)
- **JWT Decode:** 4.0.0 (Decodificación de tokens)
- **Chart.js/ng2-charts:** 4.5.1/8.0.0 (Gráficos)
- **SweetAlert2:** 11.19.1 (Alertas modales)

Navegadores Soportados

- Chrome 90+
- Firefox 88+
- Safari 14+
- Edge 90+

4. Instalación y Configuración

Instalación

1. **Clonar el repositorio:**

```
git clone <url-del-repositorio>
cd schoolme-portal
```

2. **Instalar dependencias:**

```
npm install --legacy-peer-deps
```

3. **Configurar entorno:**
    - Copiar `src/environments/environment.ts` según el entorno
    - Para desarrollo: usar `environment.development.ts`
    - Verificar URL de API: `https://api.schoolme.space/api`
4. **Ejecutar en desarrollo:**

Grupo de Desarollo SchollMe
Jesus Fernando Carvajal jesusanacona017@gmail.com
Santiago Chaparro Riaño alfasan1481@gmail.com

```
npm start
# o
ng serve
```

Acceder en: `http://localhost:4200`

Configuración de Entorno

Los archivos de entorno contienen:

- `apiUrl`: URL base de la API REST
- `uri`: Dirección IP para desarrollo local (opcional)

5. Estructura del Proyecto

```
schoolme-portal/
├── .angular/                 # Cache de Angular
├── .vscode/                  # Configuración VS Code
├── public/                   # Assets estáticos
│   ├── icons/               # Iconos de la aplicación
│   ├── img/                 # Imágenes
│   └── logos/               # Logos del colegio
├── src/
│   ├── app/
│   │   ├── auth/            # Módulos de autenticación
│   │   │   ├── guards/      # Guards de rutas
│   │   ├── components/      # Componentes reutilizables
│   │   │   ├── loader/      # Componente de carga
│   │   │   ├── modal-* /    # Modales genéricos
│   │   │   ├── nav/         # Navegación
│   │   │   └── sidebar/     # Barra lateral
│   │   ├── global/          # Servicios y módulos globales
│   │   ├── layout/          # Layout principal (dashboard)
│   │   ├── models/          # Interfaces TypeScript
│   │   │   ├── business/    # Modelos de negocio
│   │   │   ├── global/      # Modelos globales
│   │   │   ├── helpers/     # Utilidades
│   │   │   ├── parameters/  # Parámetros del sistema
│   │   │   └── security/    # Modelos de seguridad
│   │   ├── page/            # Páginas/componentes de rutas
│   │   │   ├── business/    # Páginas de negocio
│   │   │   ├── paramaters/  # Páginas de parámetros
│   │   │   ├── security/    # Páginas de seguridad
│   │   │   └── home/        # Páginas principales
│   │   └── service/         # Servicios de negocio
│   ├── environments/        # Configuración por entorno
│   └── styles.css           # Estilos globales
├── Dockerfile               # Configuración Docker
├── nginx.conf               # Configuración Nginx
├── angular.json             # Configuración Angular CLI
├── package.json             # Dependencias y scripts
└── tsconfig.*.json          # Configuración TypeScript
```

6. Gestión de Dependencias

Scripts Disponibles

```
{
  "start": "ng serve",
```


Grupo de Desarollo SchollMe
Jesus Fernando Carvajal jesusanacona017@gmail.com
Santiago Chaparro Riaño alfasan1481@gmail.com

```
  "build": "ng build",
  "watch": "ng build --watch --configuration development",
  "test": "ng test"
}
```

Dependencias Críticas

- **@angular/core:** Framework base
- **@taiga-ui/core:** Biblioteca de componentes UI
- **@angular/material:** Componentes Material Design
- **rxjs:** Programación reactiva
- **jwt-decode:** Manejo de tokens JWT

Gestión de Versiones

- Usar `npm install` para instalar dependencias
- `--legacy-peer-deps` para resolver conflictos de versiones
- Actualizaciones críticas requieren testing exhaustivo

7. Autenticación y Seguridad

Mecanismo de Autenticación

- **Tipo:** JWT (JSON Web Tokens)
- **Almacenamiento:** LocalStorage/SessionStorage
- **Interceptor:** Automático en todas las peticiones HTTP

Guards de Seguridad

- **esAdminGuard:** Verifica autenticación para rutas protegidas
- Redirección automática a `/login` si no autenticado

Roles y Permisos

- Sistema basado en roles con permisos granulares
- Modelos: User, Rol, Permission, Module, Form
- Asignación dinámica de permisos por rol

Seguridad Adicional

- Validación de tokens en cada petición
- Protección CSRF implícita en JWT
- Sanitización de inputs con Angular Forms

8. Especificación de API REST

URL Base

```
https://api.schoolme.space/api
```

Endpoints Principales

Basado en los modelos y servicios del frontend:

Seguridad

- `POST /auth/login` - Autenticación
- `GET /users` - Listar usuarios
- `POST /users` - Crear usuario
- `PUT /users/{id}` - Actualizar usuario

Estudiantes

- `GET /students` - Listar estudiantes
- `POST /students` - Crear estudiante
- `PUT /students/{id}` - Actualizar estudiante

Docentes

- `GET /teachers` - Listar docentes
- `POST /teachers` - Crear docente

Cargas Académicas

- `GET /academic-loads` - Listar cargas académicas
- `POST /academic-loads` - Crear carga académica

Agendas

- `GET /agendas` - Listar agendas
- `POST /agendas` - Crear agenda

Formatos de Request/Response

- **Content-Type:** `application/json`
- **Authorization:** `Bearer {token}`
- **Responses:** Estructuras JSON con campos `data`, `message`, `status`

9. Modelos de Datos

Modelos de Seguridad

```
interface User {
  id: number;
  email: string;
  password: string;
  photo: string;
  personId: number;
  status: number;
}

interface Rol {
```


Grupo de Desarollo SchollMe
Jesus Fernando Carvajal jesusanacona017@gmail.com
Santiago Chaparro Riaño alfasan1481@gmail.com

```
  id: number;
  name: string;
  description: string;
  status: number;
}

interface Permission {
  id: number;
  name: string;
  description: string;
}
```

Modelos de Parámetros

```
interface Student {
  id: number;
  personId: number;
  fullName: string;
  groupId: number;
  documentTypeId: string;
  identification: string;
  status: number;
}

interface Teacher {
  id: number;
  personId: number;
  fullName: string;
  phone: number;
  documentTypeId: string;
  identification: string;
  status: number;
}
```

Modelos de Negocio

```
interface AcademicLoad {
  id: number;
  teacherId: number;
  subjectId: number;
  groupId: number;
  hours: number;
}

interface Agenda {
  id: number;
  title: string;
  description: string;
  date: Date;
  teacherId: number;
}
```

10. Flujos de Negocio

Gestión de Usuarios

1. Creación de persona (datos básicos)
2. Creación de usuario con email/password

3. Asignación de roles y permisos
4. Activación/desactivación de cuenta

Matrícula de Estudiantes

1. Registro de datos personales
2. Asignación de grupo/aula
3. Creación de registro académico
4. Configuración de acudientes

Asignación de Cargas Académicas

1. Definición de materias por grado
2. Asignación de docentes a materias
3. Distribución de horas semanales
4. Validación de conflictos de horario

Gestión de Agendas

1. Configuración de agenda por docente
2. Asignación de actividades diarias
3. Registro de observaciones
4. Seguimiento por directores de grupo

11. Pruebas y Validación

Framework de Testing

- **Unit Tests:** Karma + Jasmine
- **Configuración:** `tsconfig.spec.json`
- **Comando:** `npm test`

Cobertura de Pruebas

- Componentes principales
- Servicios de negocio
- Guards de autenticación
- Utilidades y helpers

Validación Manual

- Testing de UI/UX en diferentes navegadores
- Validación de flujos de negocio
- Pruebas de integración con API

Grupo de Desarollo SchollMe
Jesus Fernando Carvajal jesusanacona017@gmail.com
Santiago Chaparro Riaño alfasan1481@gmail.com

Estrategia de QA

- Tests unitarios para lógica crítica
- Tests de integración para servicios
- Validación manual de funcionalidades complejas

12. Compilación y Despliegue

Compilación de Producción

```
ng build --configuration production
```

- Optimización automática
- Minificación de código
- Presupuesto: máximo 5MB inicial, 8KB por componente

Despliegue con Docker

```
# Construir imagen
docker build -t schoolme-portal .

# Ejecutar contenedor
docker run -p 80:80 schoolme-portal
```

Configuración de Nginx

- Servidor web para archivos estáticos
- Configuración SPA con fallback a `index.html`
- Compresión gzip activada
- Puerto 80 expuesto

Variables de Entorno

- Producción: `environment.ts` con URL de API real
- Desarrollo: `environment.development.ts` con configuración local

13. Resolución de Problemas

Problemas Comunes

Error de Compilación

- Verificar versiones de Node.js y Angular CLI
- Limpiar cache: `rm -rf node_modules && npm install`
- Verificar `angular.json` y `tsconfig.json`

Problemas de Autenticación

- Verificar token JWT en LocalStorage
- Comprobar expiración del token

Grupo de Desarollo SchollMe
Jesus Fernando Carvajal jesusanacona017@gmail.com
Santiago Chaparro Riaño alfasan1481@gmail.com

- Validar URL de API en entorno

Errores de API

- Verificar conectividad a `https://api.schoolme.space/api`
- Revisar headers de autorización
- Validar formato de requests JSON

Problemas de UI

- Verificar carga de estilos Taiga UI
- Comprobar dependencias de Bootstrap
- Validar responsive design

Debugging

- Usar DevTools del navegador
- Console logs en servicios
- Network tab para requests HTTP
- Angular DevTools para componentes

14. Mantenimiento y Monitoreo

Monitoreo de Rendimiento

- Tamaño de bundle (máximo 5MB)
- Tiempo de carga inicial
- Uso de memoria del navegador
- Errores de JavaScript en consola

Actualizaciones

- Mantener Angular y dependencias actualizadas
- Revisar compatibilidad de Taiga UI
- Actualizar Node.js según necesidades
- Backup de configuración antes de cambios

Logs y Alertas

- Monitoreo de errores en producción
- Logs de autenticación fallida
- Alertas de rendimiento degradado
- Seguimiento de uso de funcionalidades

Mantenimiento Preventivo

- Limpieza periódica de cache

Grupo de Desarollo SchollMe
Jesus Fernando Carvajal jesusanacona017@gmail.com
Santiago Chaparro Riaño alfasan1481@gmail.com

- Actualización de certificados SSL
- Revisión de permisos de archivos
- Backup de base de datos (backend)

15. Referencias y Contacto

Documentación Oficial

- Angular Documentation
- Taiga UI Documentation
- Angular Material
- Bootstrap Documentation

Herramientas y Librerías

- RxJS
- JWT.io
- Chart.js

Contacto

- **Desarrollador:** Jesús Fernando Carvajal Anacona
- **Proyecto:** SchoolMe Portal
- **Versión:** 0.0.0
- **Fecha:** Diciembre 2024

Notas Adicionales

- Proyecto generado con Angular CLI v19.0.6
- Configuración personalizada para institución educativa
- Arquitectura modular y escalable