

Descripción

Desarrolle un portal para estudiantes que les permita ver su información académica, materias inscritas y registro de asistencias. Utilizará una API RESTful con autenticación mediante API key y se implementará utilizando React.

Requisitos Técnicos

- Implementar la aplicación usando React con componentes funcionales.
- Crear componentes reutilizables para elementos comunes (tarjetas, tablas, etc.).
- Implementar el sistema de rutas con React Router.
- Manejar estados con hooks de React (useState, useEffect, useContext, etc.).
- Consumir la API proporcionada usando fetch o axios.
- Incluir manejo básico de errores y estados de carga.
- **IMPORTANTE:** Implementar protección de rutas para que no se pueda acceder sin una API key válida.

Estructura de Componentes

Deberá crear al menos los siguientes componentes:

1. **Login:** Formulario para ingreso de API key.
2. **StudentSelector:** Componente para seleccionar un estudiante.
3. **Navbar:** Barra de navegación con opciones y perfil.
4. **Dashboard:** Vista principal con resumen de información.
5. **SubjectsList:** Lista de materias inscritas con filtrado.
6. **AbsencesSummary:** Resumen de asistencias con estadísticas.
7. **AbsencesTable:** Tabla detallada de inasistencias.
8. **ProtectedRoute:** Componente HOC para proteger rutas.

Sistema de Rutas

Implementar las siguientes rutas con React Router:

- **/login** - Página de autenticación
- **/select-student** - Selección de estudiante
- **/dashboard** - Dashboard principal

- **/subjects** - Lista de materias inscritas
- **/absences** - Registro de asistencias

Endpoints Disponibles

- **GET /students/** - Listar estudiantes
- **GET /students/{student_id}/** - Obtener perfil detallado de un estudiante
- **GET /subjects/** - Listar materias
- **GET /enrollments/?student_id={id}** - Obtener materias en las que está inscrito
- **GET /absences/?student_id={id}** - Obtener registro de inasistencias
- **GET /students/{student_id}/absence_summary/** - Obtener resumen de asistencias agrupado por materia

Autenticación

Todas las peticiones protegidas deben incluir el header `x-api-key` con el valor proporcionado.

Funcionalidades Requeridas

Página de Login o Selección de Estudiante

- Formulario para ingresar la API key.
- Selector de estudiante después de autenticación exitosa.
- Almacenar la API key en localStorage o sessionStorage para persistencia.

Página de Dashboard

- Mostrar información básica del estudiante seleccionado (nombre, email, DNI, etc.).
- Resumen de materias inscritas (cantidad, nombres, etc.).
- Resumen de asistencia (cantidad de faltas, gráfico por materia).
- Implementar componentes de tarjetas para cada sección de información.

Vista de Materias

- Lista de materias en las que está inscrito el estudiante.
- Para cada materia mostrar: nombre, código y ubicación.
- Implementar filtrado por nombre o código.

- Recordatorio visual de que solo se pueden inscribir máximo 2 materias.

Vista de Asistencias

- Tabla de inasistencias del estudiante.
- Mostrar: fecha, materia y motivo.
- Agrupar por materia.
- Incluir contador de faltas por materia.
- Usar el endpoint de resumen para mostrar estadísticas.

Navegación

- Implementar menú de navegación entre vistas.
- Mostrar en todo momento el nombre del estudiante activos.
- Incluir opción de cerrar sesión (eliminar API key y redirigir a login).

Flujo de la Aplicación

1. Usuario ingresa la API key en la página de login
2. Al autenticarse correctamente, se muestra lista de estudiantes
3. Usuario selecciona un estudiante para ver su información
4. Se redirige al dashboard con resumen de información
5. Usuario puede navegar libremente entre las vistas de materias y asistencias
6. Al cerrar sesión, se elimina la API key y se redirige a login

Ejemplo Visual

En la carpeta `media/ejemplo1.gif` encontrará un ejemplo conceptual de cómo podría verse la aplicación. Este ejemplo es solo una guía, no es necesario replicarlo exactamente. Lo importante es implementar las funcionalidades y componentes requeridos.

Consideraciones Adicionales

- Se proporcionan 150 llamadas a la API para completar la prueba. Si se agotan, se descontarán 5 puntos por reiniciar el contador.
- Puede utilizar el endpoint `/docs` para explorar la API completa.
- Enfóquese en la funcionalidad y en la estructura de componentes más que en el diseño elaborado.

- Asegúrese de manejar correctamente las llamadas asíncronas y mostrar estados de carga.
- Implemente alertas o notificaciones para errores y operaciones exitosas.

API KEY

Revisa tu correo para encontrar tu respectiva API key.

Link para Backend

El link para consumir el backend es:

<http://pc2-matricula-alb-2123051620.us-east-1.elb.amazonaws.com/>

Puedes revisar la documentación de la API en Swagger en: </docs>

1

JSON

```
[
  {
    "id": 0,
    "first_name": "string",
    "last_name": "string",
    "email": "string",
    "dni": "string",
    "phone": "string",
    "birth_date": "2025-06-25",
    "admission_date": "2025-06-25",
    "profile_photo_key": "string"
  }
]
```

JSON

```
{
  "id": 0,
  "first_name": "string",
  "last_name": "string",
  "email": "string",
```

```
"dni": "string",  
"phone": "string",  
"birth_date": "2025-06-25",  
"admission_date": "2025-06-25",  
"profile_photo_key": "string"  
}
```

JSON

```
[  
  {  
    "id": 0,  
    "name": "string",  
    "code": "string",  
    "description": "string",  
    "location": "string",  
    "max_students": 0  
  }  
]
```

JSON

```
[  
  {  
    "id": 0,  
    "student_id": 0,  
    "subject_id": 0,  
    "enrollment_date": "2025-06-25"  
  }  
]
```

Penultimo

JSON

```
[  
  {  
    "id": 0,
```

```
    "student_id": 0,  
    "subject_id": 0,  
    "date": "2025-06-25",  
    "reason": "string",  
    "comment": "string"  
  }  
]
```

ultimo

```
JSON  
{  
  "student_id": 0,  
  "student_name": "string",  
  "total_absences": 0,  
  "absences_by_subject": [  
    {  
      "subject_id": 0,  
      "subject_name": "string",  
      "subject_code": "string",  
      "total_absences": 0,  
      "absence_dates": [  
        "2025-06-25"  
      ],  
      "reasons": [  
        "string",  
        null  
      ]  
    }  
  ]  
}
```

Por si acaso:

| | | | | |
|------|---|-------------------------------|-----|---|
| GET | /students/ | List Students | 🔒 | ▼ |
| POST | /students/ | Create Student | 🔒 | ▼ |
| GET | /subjects/ | List Subjects | 🔒 | ▼ |
| POST | /subjects/ | Create Subject | 🔒 | ▼ |
| POST | /enrollments/ | Create Enrollment | 🔒 | ▼ |
| GET | /enrollments/ | List Enrollments | 🔒 | ▼ |
| POST | /absences/ | Create Absence | 🔒 | ▼ |
| GET | /absences/ | List Absences | 🔒 | ▼ |
| GET | /students/{student_id}/ | Get Student Profile | 🔒 | ▼ |
| GET | /students/{student_id}/absence_summary/ | Get Student Absence Summary | 🔒 | ▼ |
| GET | /subjects/occupancy/ | Get Subjects Occupancy | 🔒 | ▼ |
| GET | /enrollments/check_availability/ | Check Enrollment Availability | 🔒 | ▼ |
| GET | /teachers/ | List Teachers | 🔒 | ▼ |
| POST | /teachers/ | Create Teacher | 🔒 | ▼ |
| GET | /teachers/{teacher_id}/ | Get Teacher | 🔒 | ▼ |
| GET | /teachers/{teacher_id}/subjects/ | Get Teacher Subjects | 🔒 | ▼ |
| POST | /grades/ | Create Grade | 🔒 | ▼ |
| GET | /grades/ | List Grades | 🔒 | ▼ |
| GET | /students/{student_id}/grades/ | Get Student Grades | 🔒 | ▼ |
| GET | / | Health Check | | ▼ |
| GET | /health | Detailed Health Check | | ▼ |
| POST | /reset_api_key/ | Reset Api Key | 📱 🔒 | ▼ |