

BACKEND:

Los endpoints utilizados son:

POST/api/alumno: Este endpoint permite agregar un nuevo alumno a la base de datos. Recibe los datos del alumno (`firstname` , `lastname` , `grado` , `instituto`) desde el frontend y los guarda en la colección `alumnos` .

Adicionalmente se agrega el mail del usuario logueado para luego mostrar en la tabla aquellos alumnos guardados con un usuario específico.

```
_id: ObjectId('6731550266f0cd7d0e8b1ed1')
usuario : "nicolasnievas1@gmail.com"
firstname : "Florecia"
lastname : "Romano"
grado : "4"
instituto : "Inst. del Rosario"
```

```
_id: ObjectId('673156db66f0cd7d0e8b1ed2')
usuario : "nico_nievas@hotmail.com"
firstname : "Facundo"
lastname : "Rabbia"
grado : "3"
instituto : "Leibnitz"
```

POST/api/informes: Este endpoint recibe los datos de un alumno y características específicas a través del frontend.

- Busca los datos del alumno en la base de datos MongoDB.
- Genera un prompt detallado usando los datos del alumno y las características proporcionadas.
- Envía este prompt a la API de Gemini para obtener un informe generado automáticamente.
- Almacena el informe junto con los detalles del alumno en la colección `informes` de la base de datos.

- Finalmente, responde al frontend con el texto del informe.

Adicionalmente se agrega el mail del usuario logueado para luego mostrar en la tabla aquellos informes generados por un usuario específico.

```

_id: ObjectId('67314eec5d4c8b2e6eea7c6e')
usuario: "nicolasnievas1@gmail.com"
  alumno: Object
    area: "Arte"
    instituto: "Manuel Belgrano"
    caracteristica1: "Proactivo"
    caracteristica2: "Le gusta mucho trabajar en grupos"
    caracteristica3: "Resuelve inconvenientes complejos"
    caracteristica4: "inquieto"
    caracteristica5: "respetaba a los maestro"
    informeTexto: "## Informe de Arte - Antonella Peretti (Grado 3)

                **Responsabilidad:**..."

_id: ObjectId('673156f966f0cd7d0e8b1ed3')
usuario: "nico_nievas@hotmail.com"
  alumno: Object
    area: "Arte"
    instituto: "Leibnitz"
    caracteristica1: "Proactivo"
    caracteristica2: "Le gusta mucho trabajar en grupos"
    caracteristica3: "Resuelve inconvenientes complejos"

```

GET /api/alumnos: Recupera y envía una lista de todos los alumnos guardados en la colección `alumnos` de MongoDB para cada usuario específico. Esto permite que el frontend muestre la lista de alumnos disponibles.

GET /api/informes: Este endpoint envía una lista de todos los informes almacenados en la base de datos de cada usuario específico al frontend, lo que permite a los usuarios visualizar los informes generados por ellos mismos.

GET /api/informePDF/: Este endpoint genera un PDF a partir de un informe ya guardado en la base de datos.

- Busca el informe usando el `informeId` proporcionado en la URL.

- Si encuentra el informe, usa la función `generarInformePDF` para crear un PDF con los datos del informe y lo envía al usuario.
- Después de descargar el PDF, el archivo temporal se elimina del servidor para liberar espacio.

PUT /api/alumno/:id: Este endpoint permite editar los datos del alumno previamente guardado en la Base de Datos.

- Los campos editables son Nombre, Apellido, Institución y Grado

PUT /api/informe/:id: Este endpoint permite editar el informe previamente guardado en la Base de Datos.

- El único campo editable es el informe generado por la IA, en caso que el usuario desee modificar la información.

DELETE/api/informe/:id: Este endpoint permite ELIMINAR el informe guardado en la Base de Datos.

DELETE/api/alumno/:id: Este endpoint permite ELIMINAR el alumno guardado en la Base de Datos.