

Escuela Rural Challenge

Resumen de la problemática

En una escuela rural se desea automatizar la gestión de asistencia diaria de los alumnos.

Para esto te proponemos pensar y diseñar una solución de arquitectura de software que cumpla con dicho propósito.

Contexto

Diariamente se realiza el relevamiento de asistencia de alumnos. Este relevamiento lo puede realizar tanto un **docente** como un **preceptor**.

Los datos de asistencia de los alumnos son persistidos en una base de datos diferenciando la siguiente información:

- Fecha
- Se especifica si es día lluvioso
- Lista de alumnos presentes
- Lista de alumnos ausentes (con y sin justificación de ausencia)

La **justificación de ausencia** consta de un texto predefinido que justifica la ausencia (ej. "enfermedad", "trámite personal", "consulta médica", "día lluvioso", etc.).

Al tratarse de una escuela rural, y dada las condiciones de los caminos no asfaltados, el hecho de que se trate de un **día lluvioso** justifica automáticamente la ausencia de todos los alumnos ausentes.

Además, una vez tomada la asistencia de alumnos se procede a generar un **reporte estadístico** con la siguiente información:

- Fecha
- Cantidad total de alumnos del curso
- Cantidad de alumnos presentes

- Cantidad de alumnos ausentes (con y sin justificación de ausencia)

Dicho reporte puede ser guardado en la base de datos local o bien en un folder como archivo json.

Sobre cada alumno se desea contar con la siguiente información personal:

- DNI (dato único, representa además el "username")
- Contraseña de ingreso (dato sensible)
- Apellido
- Nombre
- Nacionalidad
- Email

Constraints:

*Un **alumno** puede:*

- Consultar y Modificar sus datos personales
- Consultar sus asistencias

*Un **docente** puede:*

- Consultar los datos personales de los alumnos
- Consultar las asistencias del curso
- Tomar la asistencia del curso
- Editar una asistencia del curso
- Borrar una asistencia del curso

*Un **preceptor** puede:*

- Consultar y Modificar los datos personales de los alumnos
- Consultar las asistencias del curso
- Tomar la asistencia del curso
- Editar una asistencia del curso
- Borrar una asistencia del curso
- Dar de alta un nuevo alumno
- Deshabilitar a un alumno

Otras constraints:

- Un alumno **deshabilitado** no se considera para la asistencia del curso, y además queda inhabilitado para visualizar y/o modificar sus datos personales.

- Al tratarse de una escuela rural, existe únicamente un curso (aula) donde cursan todos los alumnos.

Desarrollo

Al momento de tomar la asistencia del curso, se debe determinar si un día es lluvioso o no, a partir de la información del estado del tiempo que retorne una Weather API externa. Recomendamos -> <https://open-meteo.com> 🌤️

En caso de que tu Weather API falle luego de N reintentos (por indisponibilidad, timeout, etc), se debe adoptar una alternativa que permita obtener un pronóstico del tiempo para ese día, por ejemplo, buscando en una db local, en otro servicio, etc.

El *lenguaje de programación* que recomendamos para este desarrollo es **Python**, ayudándote del uso de cualquier framework web, idealmente **FastAPI** o **Flask**.

Terminando, los aspectos técnicos que más quisieramos que tengas en cuenta para este desarrollo son:

- **Clean code (SOLID), Clean Architecture**
- **Documentación de API**
- **Unit Tests**

Importante: agregar un **README** que tenga info del stack, cómo hacer un deploy de cada componente del ecosistema (dbs, apis, services, etc) y cualquier otra info que aporte valor.

Bonus points

- Securizar endpoints
- Utilizar patrones de diseño, arquitectura, ect.
- Diagrama de Arquitectura, Diagrama Domain-model
- Uso de Containers (idealmente Docker)
- Proponer un esquema CI/CD