

# Backend NodeJS.

## Descripción de la prueba:

El objetivo de esta prueba, es desarrollar una API utilizando NodeJS que permita la consulta de los *duacoders* existentes en una base de datos.

## Objetivos principales:

La API debe cubrir, al menos, los siguientes requisitos

- El acceso a la API debe estar restringido mediante algún tipo de autenticación con el objetivo de que no se pueda utilizar la API libremente desde internet. Puedes elegir la opción que tú consideres más apropiada: jwt, auth basic, OAuth... o cualquier otra que te parezca razonable.
- Se debe diseñar una base de datos para almacenar *duacoders*. ¿Y qué es un *duacoder*? Un *duacoder* es una persona que trabaja en duacode, y, está definida por, al menos, los siguientes campos:
  - Un NIF (obligatorio)
  - Un nombre
  - Una biografía
  - Un departamento asociado
  - Un puesto dentro del departamento
  - Una lista de *skills*
  - Una foto
  - Un campo que indique si le gusta la tortilla con cebolla o sin cebolla (obligatorio)
  - Una fecha de nacimiento (opcional)
- La API, debe proporcionar un CRUD completo sobre los elementos *duacoder*, proporcionando al menos los siguientes endpoints:
  - Crear *duacoders*
  - Editar *duacoders*
  - Listar *duacoders*, incluyendo paginación y filtros
  - Obtener el detalle de un *duacoder*
  - Eliminar *duacoders*
- Se valorarán positivamente endpoints adicionales, como pueden ser:
  - Generar un excel a partir de una lista de *duacoders* (que podrá estar filtrada).
  - Generar un PDF con una ficha que muestre todos los datos de un *duacoders* en concreto.
  - Cualquier otro endpoint adicional que se considere de interés, será valorado muy positivamente.
- Debe incluir un control de excepciones, para que los clientes que usen la API reciban códigos de error HTTP correctos y los errores no lleguen al cliente.

- Debe incluir un control de logs de errores, en algún directorio persistente, para poder llevar a cabo auditorías y revisión en caso de errores.
- Debe incluir la generación de documentación, mediante swagger, OpenAPI o similar.
- Se valorará positivamente cualquier funcionalidad extra, y se deja a criterio de la persona que realiza la prueba completar los requisitos indicados anteriormente de una forma diferente a lo planteado en este documento.

### **Requisitos técnicos:**

La prueba debe realizarse usando:

- NodeJS, valorable con el framework NestJS
- Base de datos relacional, idealmente MariaDB/Mysql
- El código debe entregarse en un repositorio accesible por duacode (GitHub, Gitlab, Bitbucket...etc) así como las instrucciones para instalación y ejecución del mismo.