¡Felicitaciones!

Si estás leyendo esto, es porque llegaste a una estapa muy importante de nuestro proceso de selección.
Te invitamos a desarrollar nuestro Challenge Técnico para la posición de **Sr Full Stack.**

¿Por qué esta etapa es importante?
Porque nos ayuda a realizar la próxima etapa (entrevista técnica) con mayor objetividad, pero principalmente nos aporta información muy valiosa sobre tus hard skills.

¿Cuánto tiempo tengo para realizar el Challenge?
Tienes 5 días corridos para realizarlo. Está pensado para invertir una hora al día, considerando que también tienes otras responsabilidades laborales como personales.

¿Qué sucede si no reaalizo el Challenge?
Lamentablemente, no podremos continuar con el proceso, ya que se trata de una instancia de las más importantes y definitorias.

Mucho éxito!

Challenge:
N5 company requests a Web API for registering user permissions, to carry out this task it is necessary to comply with the following steps:

- Create a **Permissions** table with the following fields:

Show All Filter Sort

Permisos

Name	Data Type	Extra	Field Description
Id	Integer	Auto-increment	Unique ID
NombreEmpleado	Text	Not null	Employee Forename
ApellidoEmpleado	Text	Not null	Employee Surname
TipoPermiso	Integer	Not null	Permission Type
FechaPermiso	Date	Not null	Permission granted on Date

- Create a PermissionTypes table with the following fields:

Show All Filter Sort

TipoPermisos

Name	Data Type	Extra	Field Description
Id	Integer	Auto-increment	Unique ID
Descripcion	Text	Not null	Permission description

- Create relationship between Permission and PermissionType.
- Create a Web API using ASP .NET Core and persist data on SQL Server.
- Make use of EntityFramework.
- The Web API must have 3 services “Request Permission”, “Modify Permission” and “Get Permissions”. Every service should persist a permission registry in an elasticsearch index, the register inserted in elasticsearch must contains the same structure of database table “permission”.
- Create apache kafka in local environment and create new topic where persist every operation a message with the next dto structure:
  -Id: random Guid
  -Name operation: “modify”, “request” or “get”.
- Making use of repository pattern and Unit of Work and CQRS pattern(Desired). Bear in mind that is required to stick to a proper service architecture so that creating different layers and dependency injection is a must-have.
- Create Unit Testing and Integration Testing to call the three of the services.
- Build an app in ReactJS and use Axios to connect to the backend

- Create the forms to consume the Web API.
- For the visual components, the candidate must use those provided by the Material-UI library. The project will already have the customized Theme installed to facilitate similarities with the proposed design.
- Use good practices as much as possible on the backend and frontend.
- Prepare the solutions to be containerized in a docker image.
- Upload exercise to some repository (github, gitlab, etc).