

# Desafío técnico

## Objetivo:

El principal objetivo es construir una API REST con las operaciones CRUD necesarias para gestionar el recurso Persona.

Se divide en 3 niveles de complejidad para que puedas demostrar tu experiencia y conocimientos.

Se evaluará por cada nivel completo, pudiéndose entregar el desafío en forma parcial incluyendo solo nivel 1 o solo nivel 1 y 2, o los 3 niveles completos.

## Consigna:

- No puede haber personas repetidas (las personas se identifican por Tipo de documento, número de documento, país).
- Las personas deben tener al menos un dato de contacto.
- No pueden crearse personas menores de 18 años.
- Pueden tener cualquier nacionalidad.
- Las entidades complementarias a Personas deben estar pre-cargadas al momento de usar la API.

## Desafíos:

### Nivel 1:

- Desarrollo que cumpla con la consigna inicial, que pueda descargarse y ejecutarse localmente.
- Documentación para configurar el entorno local.

### Nivel 2:

- Extender la API para que permita relacionar personas.
- Se debe crear el endpoint `/personas/:id1/padre/:id2`
  - Debe soportar el método POST cuyo resultado es indicar que `":id1"` es padre de `":id2"`.
- Se debe crear el endpoint `/relaciones/:id1/:id2`
  - Debe soportar el método GET que retorne la relación que existe entre ambas personas.
- Las relaciones posibles para retornar son: {"HERMAN@", "PRIM@", "TI@"}

### Nivel 3:

- API REST publicada en algún servicio free de cloud computing (Google App Engine, Heroku, Amazon AWS, Azure, etc)
- Agregar el recurso `/stats` que devuelva cifras totalizadoras de cantidad de porcentaje de personas por país.

- Ejemplo:

```
[{
  "country": "Argentina",
  "percentage": "82.75"
}, {
  "country": "Brasil",
  "percentage": "17.25"
}]
```

- Desarrollar Test de performance considerando que la API pueda recibir fluctuaciones de tráfico (Entre 1 y 500 request por segundo).

### Entregar:

- Documentación de desarrollo, debes incluir toda la documentación que creas necesaria, para que el código entregado se ejecute sin errores.
- Código Fuente accesible en GitHub, Bitbucket o GitLab, donde se registren desde el commit inicial hasta el último de la versión productiva. Opcionalmente puede dejar en un Branch los features sin terminar.
- Además del código fuente, cualquier pieza adicional que consideres aporta a la calidad del software construido.
- URL de la API (Nivel 3) y credenciales en caso de ser necesario, para poder invocar la API desde Internet.

### Tecnología:

- Lenguaje: JAVA 11+
- Frameworks: Spring Boot, JPA, etc.
- Base de datos: libre elección.