

# Prueba Técnica - Desarrollador Fullstack

## Prueba Teórica

1. Describa la diferencia entre `app.use()` y `app.get()` en Express.
2. Describa qué es y cómo funciona un sistema de ruteo en Express.
3. Describa qué son los CORS y cómo se pueden configurar en una aplicación Express.
4. Describa el ciclo de vida de un componente en angular.
5. Describa qué es y cómo funciona un sistema de ruteo en Angular
6. Describa el patrón Observable y su uso en Angular
7. Especifique la división de capas en Express y Angular y agregue una descripción por cada una de estas.

## Prueba Practica

### Instrucciones

Luego de recibir la prueba debes enviar tus resultados en máximo 24 horas.

La prueba debe ejecutarse usando: Node.js, Express, Mongo DB y Angular codificado con el lenguaje TypeScript.

¡Buena suerte!

### BackEnd

El BackEnd debe tener los controladores: gatos e imágenes. Se debe crear un servicio por cada controlador y los modelos necesarios para el intercambio de datos a través de REST, además de implementar pruebas unitarias y buenas prácticas de desarrollo (SOLID, Clean Architecture, entre otros)

Los controladores deben conectarse a la API pública <https://thecatapi.com/>

.Puede usar la key o generar una propia

```
live_JBT0Ah0Nt12iyI2lpjQVLDWjcLk0GQwf4zI9wBMfmfejKmcC31mOJp4yJz5TsOUP
```

### Controlador Gatos:

El controlador debe tener 3 acciones:

- GET /breeds :

Debe retornar una lista de las razas de gatos .

- GET /breeds/:breed\_id

Debe retornar únicamente la raza de gato solicitada

- GET /breeds/search:

Debe retornar la información asociada a la consulta realizada de acuerdo a los parametros.

### Controlador Imagenes:

El controlador debe tener 1 accion:

- GET /imagesbybreedid :

Debe retornar las imágenes asociadas a una raza especifica de gato.

### Controlador Usuarios:

El controlador debe tener 2 acción:

- GET /login :

Debe permitir verificar en una colección de Mongo la existencia de un usuario y validar su contraseña y retorna la información del usuario .

- GET /Register :

Debe permitir registrar en una colección de Mongo un usuario con los datos básicos de registro.

### FrontEnd

El FrontEnd en Angular debe cumplir con los siguientes requerimientos, además de implementar pruebas unitarias, diseño responsivo y buenas prácticas de desarrollo (SOLID, Clean Architecture):

### Vista 1

- Debe tener una lista desplegable con los nombres de las razas disponibles para seleccionar.
- Luego de seleccionar una raza en la lista desplegable debe cargar un item dentro del front que me permita visualizar las imagen en un carrusel y la información relevante de la raza de gato seleccionada.
- Una vista adicional con una tabla donde pueda visualizar los datos más relevantes de las razas de gatos.

### Vista 2

- Debe agregar un filtro de búsqueda tipo texto que al ingresar información y dar clic en un botón de búsqueda filtra la tabla con las coincidencias del texto ingresado.

### Vista 3

- Login debe existir una vista permita ingresar le usuario y la contraseña y consumiendo la api del backend realizado me permita identificar al usuario

### Vista 4

- Registro debe existir una vista diligenciar los datos de registro básicos del usuario.

### Vista 5

- Vista protegida por un guardia que muestra la información del usuario logueado.

### Opcional

Cree un dockerfile para los proyectos anteriormente desarrollados que permitan contenerizarlos

## Entregables

- Para la prueba teórica debe adjuntar al correo un archivo con las respuestas de las preguntas.
- Para la prueba teórica se debe crear un repositorio en github para cada uno de los proyectos y compartir la URL a través de correo electrónico.