ANGULAR:  
 **¿Cuál es la diferencia entre componentes y directivas en Angular?**

* **Respuesta:** Ambas funcionan de una manera similar solo que un componente de angular es mas completo y es mas sencillo de usar

 **¿Cómo funciona el ciclo de vida de un componente en Angular?**

* **Respuesta:** El ciclo de vida de un componente de angular empieza cuando inicializamos lo necesario para que el mismo pueda ejecutarse, para ellos antes de la versión 17 de angular hacíamos uso de un archivo .module donde importábamos todas las dependencias y en el componente las llamábamos en el constructor para realizar una inyección de dependencias, dicho esto lo que sigue es lo que pasa durante el uso del componente así que para eso revisamos cada cambio que hace el usuario por medio del método de ciclo de vida onchanges donde revisa si hay algún cambio dentro del componente y así sigue hasta que decidamos terminar el ciclo de vida, ya para finalizar el ciclo de vida usamos el método ondestroy donde este limpia todo lo que se este usando dentro del componente y finaliza su ciclo de vida

 **¿Qué es el RxJS y cómo se utiliza en Angular?**

* **Respuesta:** El RxJS es una librería que implementa algo conocido como Observables, esto se utiliza principalmente en la integración entre componentes y servicios dentro de angular, su implementación comienza con un subscribe o suscripción donde podemos hacer uso del servicio como nosotros queramos, esta librería abarca un sinfín de métodos como seria mapear el servicio y sus valores entrantes antes de ser asignados a una variable fuera del Observable y podemos realizar un debugeo mediante tap donde podemos observar que le está llegando al observable

 **¿Qué es un Guard en Angular y cómo se implementa?**

* **Respuesta:** Los Guard como su nombre lo indica son métodos que nos permiten asegurar en todo momento que el usuario final solo tenga acceso a lo que tenga permitido nada mas mediante su rol, ósea, que el Guard es una serie de métodos que permiten precargar una ruta con antelación y determina si puede acceder a la misma

 **¿Cómo funciona el enrutamiento en Angular?**

* **Respuesta:** El enrutamiento en angular permite definir unas rutas establecidas al proyecto de angular con el propósito de acceder a ciertos componentes y si se requiere pasar valores para que el componente al cual se esta accediendo y necesite de estos valores ya lo tenga disponible en caso de hacer una consulta a un servicio **Prueba Técnica**

**Proyecto:** **Explorador de Personajes de Marvel**

**Descripción:** El objetivo de esta prueba es crear una aplicación en Angular que consuma la API de Marvel para permitir a los usuarios explorar y buscar información sobre personajes de Marvel, cómics, y eventos.

**Requisitos del Proyecto:**

1. **Página de Inicio:**
   * Una vista inicial que muestre una lista de personajes populares de Marvel. Cada personaje debe mostrar su nombre, imagen, y un enlace para ver más detalles.
   * Agregar un formulario que permita crear un nuevo personaje, este formulario permitirá al usuario ingresar el nombre del personaje y la URL de su imagen, y luego agregar este personaje a la lista existente.

Material:  
<https://material.angular.io/components/categories>  
  
Repositorio:  
<https://github.com/arleyxD/PruebaAngular>  
  
Se debe entregar el proyecto en Repositorio de Git personal en publico.  
  
RESPUTA Ruta Git:  
  
  
JAVA SPRING BOOT:

**1.Pregunta:** ¿Cuál de las siguientes opciones es la forma correcta de definir un controlador REST en Spring Boot?

**Opciones:**

A)

java

Copiar código

@RestController

public class MyController {

@GetMapping("/hello")

public String sayHello() {

return "Hello, World!";

}

}

B)

java

Copiar código

@Controller

public class MyController {

@RequestMapping("/hello")

public String sayHello() {

return "Hello, World!";

}

}

C)

java

Copiar código

@RestController

public class MyController {

@RequestMapping("/hello")

public String sayHello() {

return "Hello, World!";

}

}

D)

java

Copiar código

@Controller

public class MyController {

@GetMapping("/hello")

public String sayHello() {

return "Hello, World!";

}

}

**2.Pregunta:** ¿Cuál es el archivo correcto en Spring Boot para configurar las propiedades de conexión a la base de datos cuando se usa Spring Data JPA?

**Opciones:**

A) application.properties

B) application.yml

C) db.properties

D) database-config.yml

**3.Pregunta:** ¿Cuál de las siguientes anotaciones es utilizada para definir la clave primaria de una entidad en JPA?

**Opciones:**

A) @Column

B) @Id

C) @GeneratedValue

D) @PrimaryKey

**4.Pregunta:** ¿Cómo se realiza la inyección de dependencias en Spring Boot?

**Opciones:**

A)

java

Copiar código

public class MyService {

private MyRepository myRepository = new MyRepository();

}

B)

java

Copiar código

public class MyService {

@Autowired

private MyRepository myRepository;

}

C)

java

Copiar código

public class MyService {

private MyRepository myRepository;

public MyService(MyRepository myRepository) {

this.myRepository = myRepository;

}

}

D)

java

Copiar código

public class MyService {

@Inject

private MyRepository myRepository;

}

**5.Pregunta:** ¿Cuál de las siguientes es la forma correcta de manejar excepciones en un controlador REST en Spring Boot?

**Opciones:**

A)

java

Copiar código

@RestController

public class MyController {

@ExceptionHandler(Exception.class)

public String handleException() {

return "An error occurred!";

}

}

B)

java

Copiar código

@RestController

public class MyController {

@ExceptionHandler(Exception.class)

public ResponseEntity<String> handleException() {

return new ResponseEntity<>("An error occurred!", HttpStatus.INTERNAL\_SERVER\_ERROR);

}

}

C)

java

Copiar código

@RestController

public class MyController {

@RequestMapping("/handleException")

public String handleException() {

return "An error occurred!";

}

}

D)

java

Copiar código

@RestController

public class MyController {

@ExceptionHandler(Exception.class)

public void handleException() {

// Handle the exception without returning any response

}

}

Tabla de respuestas :

|  |  |
| --- | --- |
| Pregunta 1 | A |
| Pregunta 2 | A |
| Pregunta 3 | B |
| Pregunta 4 | B |
| Pregunta 5 | B |