

Actividad:

codificación de módulos del software Stand alone, web y móvil de acuerdo al proyecto a desarrollar

GA7-220501096-AA3-EV01

Aprendiz:

Wilmer Jair Espinosa Silva

CC: 1.095.910.391

Instructor:

ISRAEL ARBONA GUERRERO

Servicio Nacional de aprendizaje-SENA

Curso: TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

Ficha: 2455285

Con base en la selección del proyecto a desarrollar móvil o web realice la codificación del módulo del proyecto aplicando alguno de los framework vistos en el componente formativo “Frameworks para construcción de aplicaciones con JAVA.”.

Elementos para tener en cuenta:

- Para la codificación del módulo debe tener en cuenta los artefactos del ciclo del software realizados con anterioridad: diagrama de clases, diagramas de casos de uso, historias de usuario, diseños, prototipos, Informe técnico de plan de trabajo para construcción de software con tecnologías seleccionadas etc.
- El código debe contener comentarios
- El código debe cumplir con estándares de codificación
- Se debe crear el proyecto utilizando herramientas de versionamiento.

RTA:

Hay varios frameworks que se pueden utilizar para construir un módulo en Java. Algunos de los más populares incluyen:

1. Spring: es un framework de aplicaciones Java completo que proporciona una gran cantidad de herramientas para desarrollar aplicaciones web, incluido un módulo de seguridad, gestión de transacciones, integración con bases de datos y más.
2. Hibernate: es un framework de mapeo objeto-relacional que permite a los desarrolladores trabajar con bases de datos de una manera más sencilla y eficiente.
3. Struts: es un framework de aplicaciones web que proporciona un marco para desarrollar aplicaciones web en Java.

Para comenzar a codificar el módulo, primero debe familiarizarse con el framework que desea utilizar y leer la documentación correspondiente. A continuación, puede comenzar a escribir código para implementar las funcionalidades deseadas utilizando las herramientas y las API proporcionadas por el framework.

Es importante destacar que cada framework tiene su propia sintaxis y convenciones, por lo que es importante asegurarse de seguir las recomendaciones y buenas prácticas específicas para ese framework.

- Código utilizando el framework Spring Boot:

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class PortfolioWebApplication {
    public static void main(String[] args) {
        SpringApplication.run(PortfolioWebApplication.class, args);
    }
}
```

- Código para un controlador en la aplicación:

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class DeveloperController {
    private List<Developer> developers = new ArrayList<>();

    public DeveloperController() {
        developers.add(new Developer("John Doe", "Full Stack"));
        developers.add(new Developer("Jane Doe", "Mobile"));
    }

    @GetMapping("/developers")
    @ResponseBody
    public List<Developer> getDevelopers() {
        return developers;
    }
}
```

Este código define una clase **DeveloperPortfolioApplication** que actúa como punto de entrada para la aplicación y está anotada con **@SpringBootApplication**. También se define una clase **DeveloperController** que proporciona un controlador REST que devuelve una lista de desarrolladores. Este es solo un ejemplo básico, y en una aplicación real, probablemente habría muchas más clases, métodos y lógica adicional.