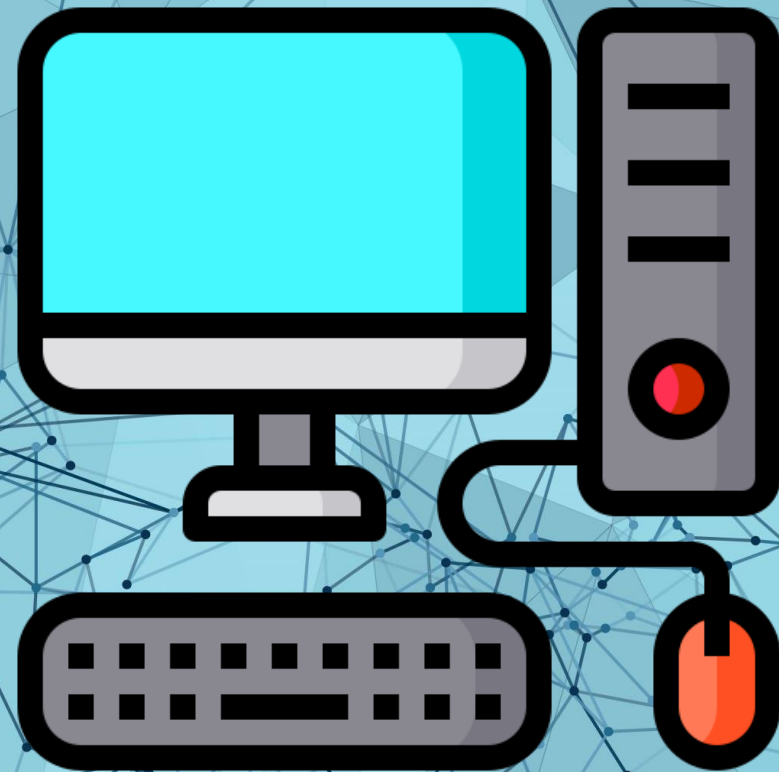


Frameworks Svelte y Spring.

Equipo NO. 1

- Keb Sebastián Fernández Guerrero
- Ricardo Rodríguez Escalera

Grupo 2TM3



¿Qué es un Framework?

Son conjuntos de herramientas, bibliotecas, y mejores prácticas diseñados para facilitar el desarrollo de software y aplicaciones. Estas estructuras proporcionan un conjunto predefinido de funcionalidades y abstracciones que permiten a los desarrolladores crear aplicaciones de manera más eficiente y consistente, ya que no tienen que construir todo desde cero.



Abstracción

Patrones
de diseño

Características

Componentes
reutilizables

Productividad

Mantenimiento





SVELTE

Svelte es un marco de JavaScript moderno para crear interfaces de usuario (UI). Es similar a otros marcos de UI como React, Vue.js y Angular, pero tiene algunas características distintivas que lo distinguen.



CARACTERÍSTICAS

- Es conocido por su simplicidad, rendimiento y facilidad de aprendizaje.
- Es una excelente opción para crear aplicaciones web modernas, especialmente si desea optimizar el rendimiento y reducir el tamaño de su paquete de aplicaciones.

COMPONENTES / DESCRIPCIÓN

- Enfoque basado en compilador
- Sin DOM virtual
- Sintaxis declarativa
- Declaraciones reactivas
- CSS con alcance
- Manejo de eventos
- Tiendas
- Animaciones
- Representación del lado del servidor (SSR)
- Tamaño de paquete pequeño
- Ecosistema en crecimiento
- SvelteKit oficial



 App.jsx App.jsx\  Navbar

```
import { useState } from "react";

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <button onClick={() => setCount(c => c + 1)}>
      count is {count}
    </button>
  );
}

export default Counter;
```

 App.svelte App.svelte\  script

```
<script>
  let count = 0;
</script>
💡
<button on:click={() => count++}>
  count is {count}
</button>
|
```

App.jsx App.jsx\ Counter

```
function Counter() {  
  const [count, setCount] = useState(0);  
  
  useEffect(() => {  
    document.title = `count is ${count}`;  
    }, [count]);  
}
```

App.svelte 2 App.svelte\...

```
<script>  
  let count;  
  
  $: document.title = `count is ${count}`;  
</script>
```

App.jsx App.jsx\ Counter

```
function Counter() {  
  const [count, setCount] = useState(0)  
  
  return <>  
    {count > 1000 ? (  
      <p>big</p>  
    ) : count > 500 ? (  
      <p>medium</p>  
    ) : (  
      <p>small</p>  
    )}  
  </>  
}
```

App.svelte App.svelte\...

```
<script>  
  let count = 0;  
</script>  
  
{#if count > 1000}  
  <p>big</p>  
{:else if count > 500}  
  <p>medium</p>  
{:else}  
  <p>small</p>  
</if>
```


App.jsx App.jsx\ Counter

```
function Counter() {  
  const items = [  
    { id: 1, name: "foo" },  
    { id: 2, name: "bar" },  
    { id: 3, name: "baz" },  
  ]  
  
  return <>  
    {items.map((item) => (  
      <div key={item.id}>{item.name}</div>  
    ))}  
  </>;  
}
```

App.svelte App.svelte\...

```
<script>  
  const items = [  
    { id: 1, name: "foo" },  
    { id: 2, name: "bar" },  
    { id: 3, name: "baz" },  
  ]  
</script>  
  
{#each items as item (item.id)}  
  <p>{item.name}</p>  
{/each}
```



spring
Framework

Spring es un marco de código abierto popular y ampliamente utilizado para crear aplicaciones Java de nivel empresarial. Proporciona una infraestructura integral para desarrollar aplicaciones seguras, de alto rendimiento y escalables.



CARACTERISTICAS

- Proporciona una infraestructura integral para desarrollar aplicaciones seguras, de alto rendimiento y escalables.
- Se adopta ampliamente en el ecosistema Java empresarial debido a su amplio conjunto de funciones, soporte comunitario y flexibilidad.
- Es adecuado para una amplia gama de aplicaciones, desde pequeños servicios web hasta sistemas de misión crítica a gran escala.

COMPONENTES

- Contenedor principal
- Acceso/integración de datos
- Spring MVC:
- Spring Boot
- Spring Security
- Spring Cloud
- Spring Data
- Spring Batch
- Spring Web Services
- Spring Testing



```
1 package com.universidad
2
3 public class EstudianteBean{
4     private String nombre;
5     private String apellido;
6     private String dni;
7
8     public String getNombre(){
9         return nombre;
10    }
11
12    public void setNombre(String nombre){
13        this.nombre = nombre;
14    }
15
16    public String getApellido(){
17        return apellido;
18    }
19
20    public void setApellido(String apellido){
21        this.apellido = apellido;
22    }
23
24    public String getDni(){
25        return dni;
26    }
27
28    public void setDni(String dni){
29        this.dni = dni;
30    }
31
32    public void mostrarEstudianteActual(){
33        System.out.println(nombre+" "+apellido+", es el estudiante
34        actual");
35    }
36 }
```

```
1 &lt;!DOCTYPE beans PUBLIC "-//PRING//DTD// BEAN
2 2.0//ES"
3 "http://www.springframework.org/dtd/spring-beans-
4 2.0.dtd"&gt;
5 &lt;beans&gt;
6   &lt;bean id="estudiante"
7   class="com.universidad.EstudianteBean"&gt;
8     &lt;property name="nombre" value="Juan"&gt;
9     &lt;property name="apellido"
value="Perez"&gt;
      &lt;property name="dni"
value="95.445.123"&gt;
      &lt;/bean&gt;
&lt;/beans&gt;
```

```
1 package com.universidad
2
3 import org.springframework.beans.factory.BeanFactory;
4 import
5     org.springframework.beans.factory.xml.XmlBeanFactory;
6 import org.springframework.core.io.ClassPathResource;
7 import org.springframework.core.io.Resource;
8
9 public class UniversidadLogic{
10
11     public static void main(String[] args){
12         Resource recurso = new
13         ClassPathResource("springconfig.xml");
14         BeanFactory factory = new
15         XmlBeanFactory(recurso);
16
17         EstudianteBean estudianteBean =
18         (EstudianteBean)factory.getBean("estudiante");
19
20         estudianteBean.mostrarEstudianteActual();
21     }
22 }
```

Gracias por su
Atención.

