



	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2025/05/03</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

## INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Programacion Web 2				
TÍTULO DE LA PRÁCTICA:	<i>Git de Sesiones Angular</i>				
NÚMERO DE PRÁCTICA:		AÑO LECTIVO:	2025	NRO. SEMESTRE:	1
FECHA DE PRESENTACIÓN	3/07/2025	Repositorio	<a href="https://github.com/JesusFSP/Curso-PWeb2.git">https://github.com/JesusFSP/Curso-PWeb2.git</a>		
INTEGRANTE (s): Silva Pino Jesus Francisco				NOTA:	
DOCENTE(s): CARLO JOSE LUIS CORRALES DELGADO					

SOLUCIÓN Y RESULTADOS
<p><b>I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS</b></p> <h3>1. Configuración Inicial del Proyecto</h3> <p><b>Pasos realizados:</b></p> <ol style="list-style-type: none"> <li>1. Creación del proyecto Angular usando Angular CLI.</li> <li>2. Configuración de la estructura de archivos principal.</li> <li>3. Establecimiento de las dependencias básicas.</li> </ol> <p><b>Funcionalidad implementada:</b></p> <ul style="list-style-type: none"> <li>• Base estructural para una aplicación Angular moderna.</li> <li>• Configuración inicial de rutas y módulos.</li> <li>• Preparación del entorno de desarrollo.</li> </ul>

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y</b>  <b>SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<p align="center"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2025/05/03</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p align="right"><b>Página:</b> 2</p>

## 2. Implementación del Componente Principal

### Archivos modificados:

- app.ts (Lógica principal)
- app.html (Template principal)
- app.css (Estilos principales)



### Funcionalidades implementadas:

// Componente principal que gestiona:  
// - Datos del usuario (nombre, email, página web)  
// - Lista de hobbies  
// - Lista de usuarios  
// - Funciones para manipular datos

```
export class App {
  title = 'my-dream-app';
  name: string;
  email: string;
  webpage: string;
  hobbies: string[];
  users: string[];

  constructor() {
    this.name = "Usuario Ejemplo";
    this.email = "usuario@example.com";
    this.webpage = "https://www.example.com";
    this.hobbies = ["Deportes", "Programación", "Música"];
    this.users = ['usuario1', 'usuario2', 'usuario3'];
  }

  // Métodos para gestión de datos
  newHobby(hobby: any) {
    this.hobbies.push(hobby.value);
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2025/05/03</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p><b>Página:</b> 3</p>

```

hobby.value = "";
}

deleteUser(user: string) {
  this.users = this.users.filter(u => u !== user);
}

addUser(newUser: any) {
  this.users.push(newUser.value);
  newUser.value = "";
  newUser.focus();
}
}

```

### 3. Creación del Componente HelloWorld

#### Archivos creados:

- hello-world.ts (Componente)
- hello-world.html (Template)
- hello-world.css (Estilos)

#### Funcionalidades implementadas:

```

// Componente secundario que:
// - Recibe datos del componente padre
// - Implementa interacción básica
export class HelloWorldComponent {
  @Input() nameUser: string = "";

  sayHello(nameUser: string) {
    alert(`Hola ${nameUser}`);
  }
}

```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2025/05/03</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 4</p>

## 4. Integración de Componentes

### Pasos realizados:

1. Importación del componente secundario en el principal.
2. Configuración del sistema de templates.
3. Implementación de data binding entre componentes.

### Implementación:

```
<!-- En app.html -->
<div *ngFor="let user of users">
  <app-hello-world [nameUser]="user"></app-hello-world>
</div>
```

## 5. Configuración de Rutas y Módulos

### Archivos configurados:

- app.config.ts (Configuración principal)
- main.ts (Punto de entrada)


### Estructura implementada:

```
// Configuración básica de la aplicación
export const appConfig: ApplicationConfig = {
  providers: [
    provideRouter([]) // Sistema de rutas
  ]
};
```

## 6. Implementación de Funcionalidades Adicionales

### Características implementadas:

1. Sistema de gestión de hobbies:
  - Adición dinámica de nuevos hobbies.
  - Visualización condicional.
2. Sistema de gestión de usuarios:
  - Adición de nuevos usuarios.

	<p align="center"><b>UNIVERSIDAD NACIONAL DE SAN AGUSTIN</b>  <b>FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y</b>  <b>SERVICIOS</b>  <b>ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</b></p>	
<p align="center"><b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p><b>Aprobación:</b> 2025/05/03</p>	<p><b>Código:</b> GUIA-PRLE-001</p>	<p align="right"><b>Página:</b> 5</p>

- Eliminación de usuarios existentes.
  - Data binding bidireccional.
3. Integración de componentes:
- Comunicación padre-hijo.
  - Paso de parámetros.
  - Manejo de eventos.

## 7. Estructura Final del Proyecto

src/app/

```

├─ app.html (Template principal)
├─ app.ts (Componente principal)
├─ app.css (Estilos globales)
├─ app.config.ts (Configuración)
├─ hello-world/
|   ├─ hello-world.html (Template)
|   ├─ hello-world.ts (Componente)
|   └─ hello-world.css (Estilos)
└─ ...

```

## 8. Funcionalidades Clave Implementadas

### 1. Componente Principal:

- Gestión centralizada del estado de la aplicación.
- Comunicación con componentes hijos.
- Manejo de formularios.


### 2. Componente HelloWorld:

- Recepción de datos desde componente padre.
- Emisión de eventos.
- Presentación de información.

### 3. Sistema de Templates:

- Directivas estructurales (\*ngFor, \*ngIf).
- Data binding (interpolación, property binding).




- Manejo de eventos.



# Hello, proyecto-angular

Congratulations! Your app is running. 🎉

[Explore the Docs](#)[Learn with Tutorials](#)[Prompt and best practices for AI](#)[CLI Docs](#)[Angular Language Service](#)[Angular DevTools](#)



## my-dream-app

Jesus Silva

jsilva@unsa.edu.pe

hobbies:

- Fútbol
- Programación
- Netflix

[Visita mi sitio Web](#)

### Desde app.component

• Jesus

Borrar Usuario

• joe

Borrar Usuario

• cameron

Borrar Usuario

• john

Borrar Usuario

•

hello-world works! Jesus

•

<b>II. CONCLUSIONES</b>

<b>RETROALIMENTACIÓN GENERAL</b>

<b>REFERENCIAS Y BIBLIOGRAFÍA</b>