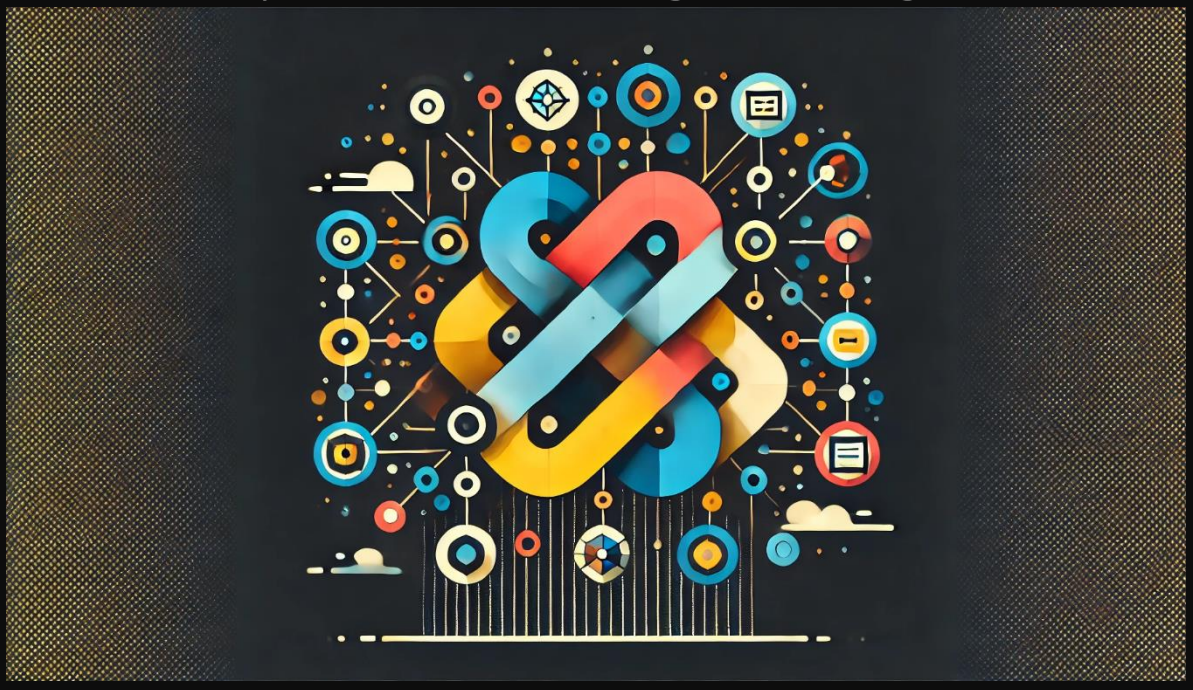
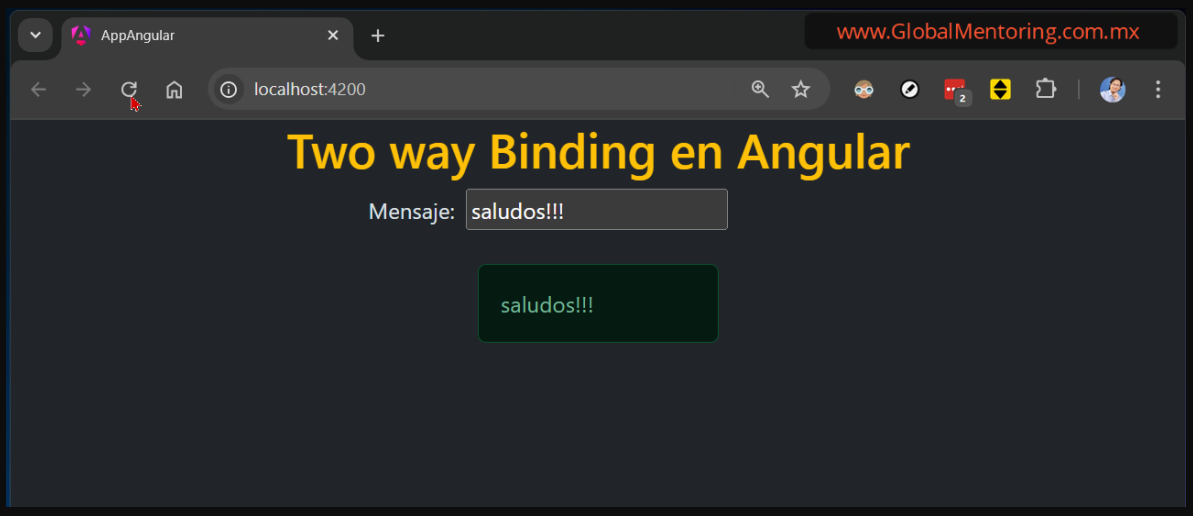
Two Way Data Binding en Angular



Two way Data Binding en Angular

Two-way data binding en Angular permite que los datos entre el componente y la vista estén sincronizados automáticamente. Esto significa que, si el valor en la vista cambia, el valor en el componente también se actualiza, y viceversa.



Sintaxis Básica:

La sintaxis para **two-way data binding** en Angular utiliza la combinación de **property binding** y **event binding** a través de la directiva especial `[(ngModel)]`

```
<input [(ngModel)]="variable" />
```

- **`[(ngModel)]`**: Esta directiva realiza el enlace bidireccional entre la propiedad del componente y el elemento HTML.

Ejemplo en Angular:

Vamos a crear un componente donde capturaremos un mensaje y lo replicaremos en otra sección de la página y mostrar el mensaje proporcionado en tiempo real.

1. Creación del Componentes `SaludarComponent`:

ng g c saludar --skip-tests

2. Componente (`saludar.component.ts`):

```
import { Component } from '@angular/core';
import { FormsModule } from '@angular/forms';

@Component({
  selector: 'app-saludar',
  standalone: true,
  imports: [FormsModule],
  templateUrl: './saludar.component.html',
  styleUrl: './saludar.component.css'
})
export class SaludarComponent {
  saludo = 'Saludo inicial';
}
```

Importante: Importar `FormsModule` en el Componente Standalone:

En un componente standalone, necesitas importar explícitamente el `FormsModule` en la configuración del componente para poder usar `ngModel` según se muestró.

3. Plantilla (`saludar.component.html`):

```
<div class="d-flex justify-content-center">
    <div class="w-50">
        <div class="mb-3">
            <label for="inputTexto" class="mx-2">Mensaje: </label>
            <input id="inputTexto" type="text" [(ngModel)]="saludo"
            autocomplete="off"/>
        </div>
```

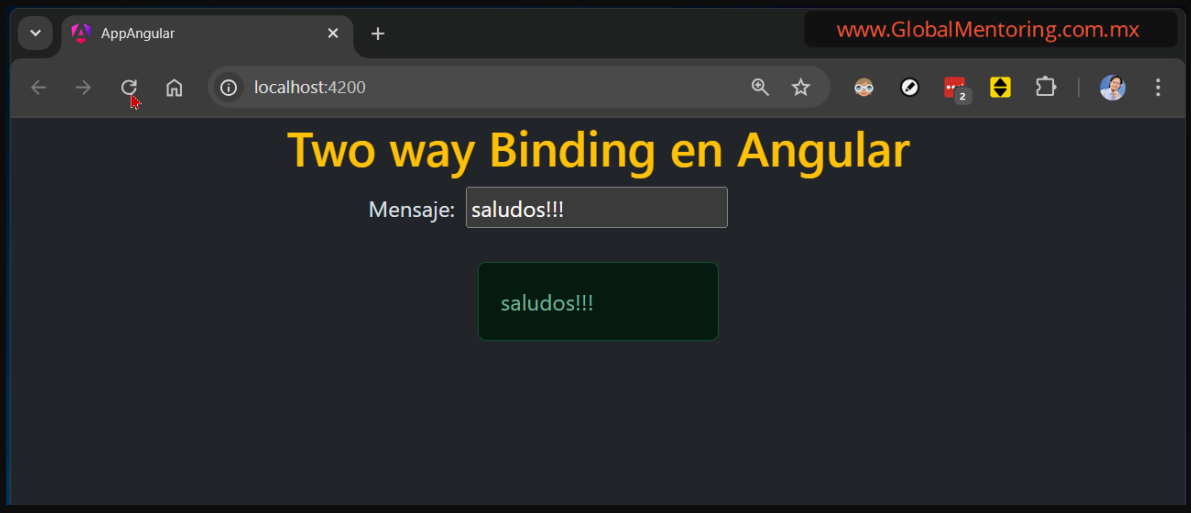
```
        <div class="mb-3">
            <div class="alert alert-success mt-4 mx-auto w-50">
                {{ saludo }}
            </div>
        </div>
    </div>
</div>
```

Explicación del Ejemplo:

1. **`[(ngModel)]="saludo"`**: Esta directiva enlaza bidireccionalmente el valor del campo de entrada (`input`) con la propiedad `saludo` del componente. Cada vez que el usuario escribe algo en el campo de entrada, la propiedad `saludo` se actualiza automáticamente. Del mismo modo, si `saludo` cambia en el componente, el valor del campo de entrada se actualizará.
2. **`{{ saludo }}`**: Interpolación que muestra el valor actual de `nombre` en la vista.
3. **Bootstrap**: Se usa para estilizar el formulario y el texto.

Resultado Esperado:

Al ejecutar la aplicación, verás un campo de entrada de texto. A medida que escribes en el campo, el texto que escribes se mostrará instantáneamente debajo del campo con el mensaje proporcionado. Esto muestra cómo Angular sincroniza automáticamente los datos entre el componente y la vista.



Notas Importantes:

- **`ngModel`**: Si no se está usando standalone componentes, entonces debes asegurarte de importar `FormsModule` en el módulo correspondiente, por ejemplo, en `app.module.ts`. Sin embargo, si estas usando standalone components es se debe omitir.

```
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    // Tus componentes aquí
  ],
  imports: [
    BrowserModule,
    FormsModule // Asegúrate de importar FormsModule aquí
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Este ejemplo ilustra cómo usar **two-way data binding** en Angular para mantener sincronizados los datos entre la vista y el componente de forma sencilla y efectiva.

Saludos!

Ing. Ubaldo Acosta

Fundador de GlobalMentoring.com.mx