

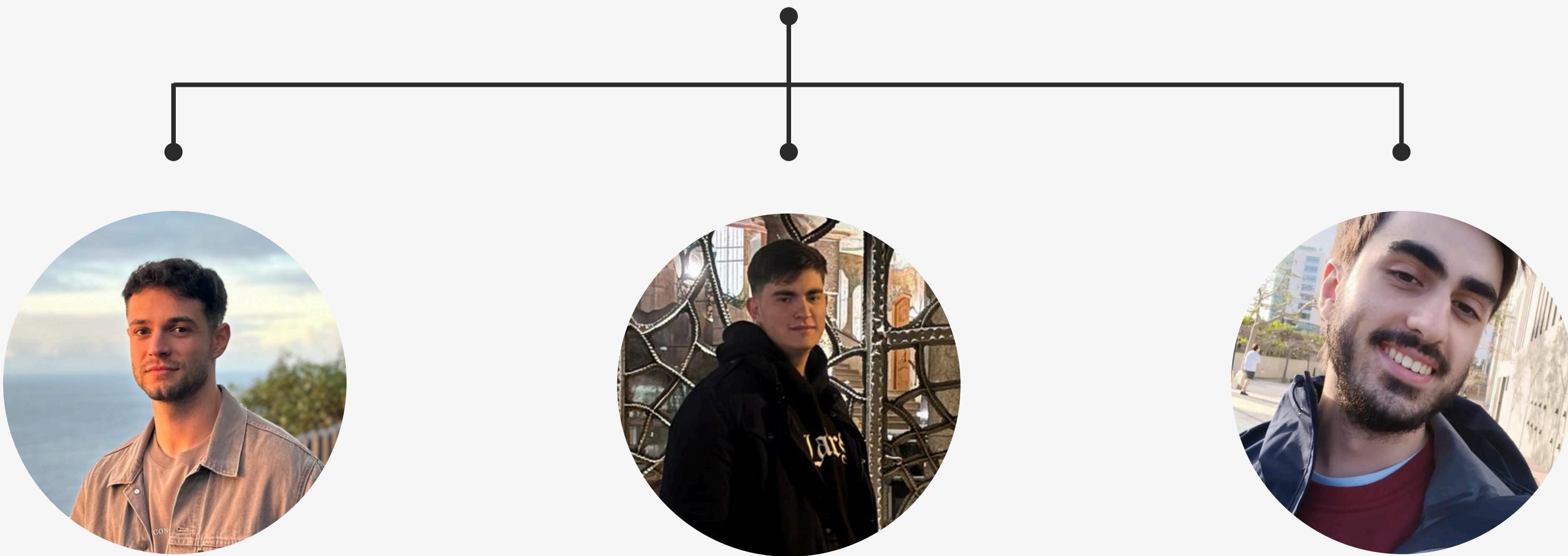
ANGULAR

Web application framework





AUTHORS



**Marcos Llinares
Montes**

alu0100972443@ull.edu.es

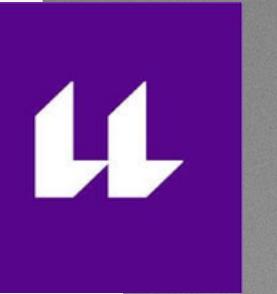
**Sergio Nicolás
Seguí**

alu0101389936@ull.edu.es

**Lucas de Vera
Gil**

alu0101550145@ull.edu.es

TABLE OF CONTENTS



| | |
|-----------------------------------|----|
| ■ Introduction to Angular | 04 |
| ■ Commands to start in angular 19 | 09 |
| ■ MVVM model | 12 |
| ■ Angular components | 14 |
| ■ Directives in angular | 21 |
| ■ Signals and types | 25 |





INTRODUCTION **TO ANGULAR**



INTRODUCTION TO ANGULAR

- Open-source web development framework maintained by Google.
- Ideal for dynamic web applications and Single Page Applications (SPAs).
- Component-based architecture, TypeScript, and integrated tools (Angular CLI).
- Focus on scalability, maintainability, and high performance.





THE BIRTH OF ANGULARJS

- Launched in 2010, based on MVC and two-way data binding.
- Revolutionized web development by adding interactivity to static pages.
- Limitations: performance and scalability issues for complex apps.

Build modern web apps with
ANGULAR



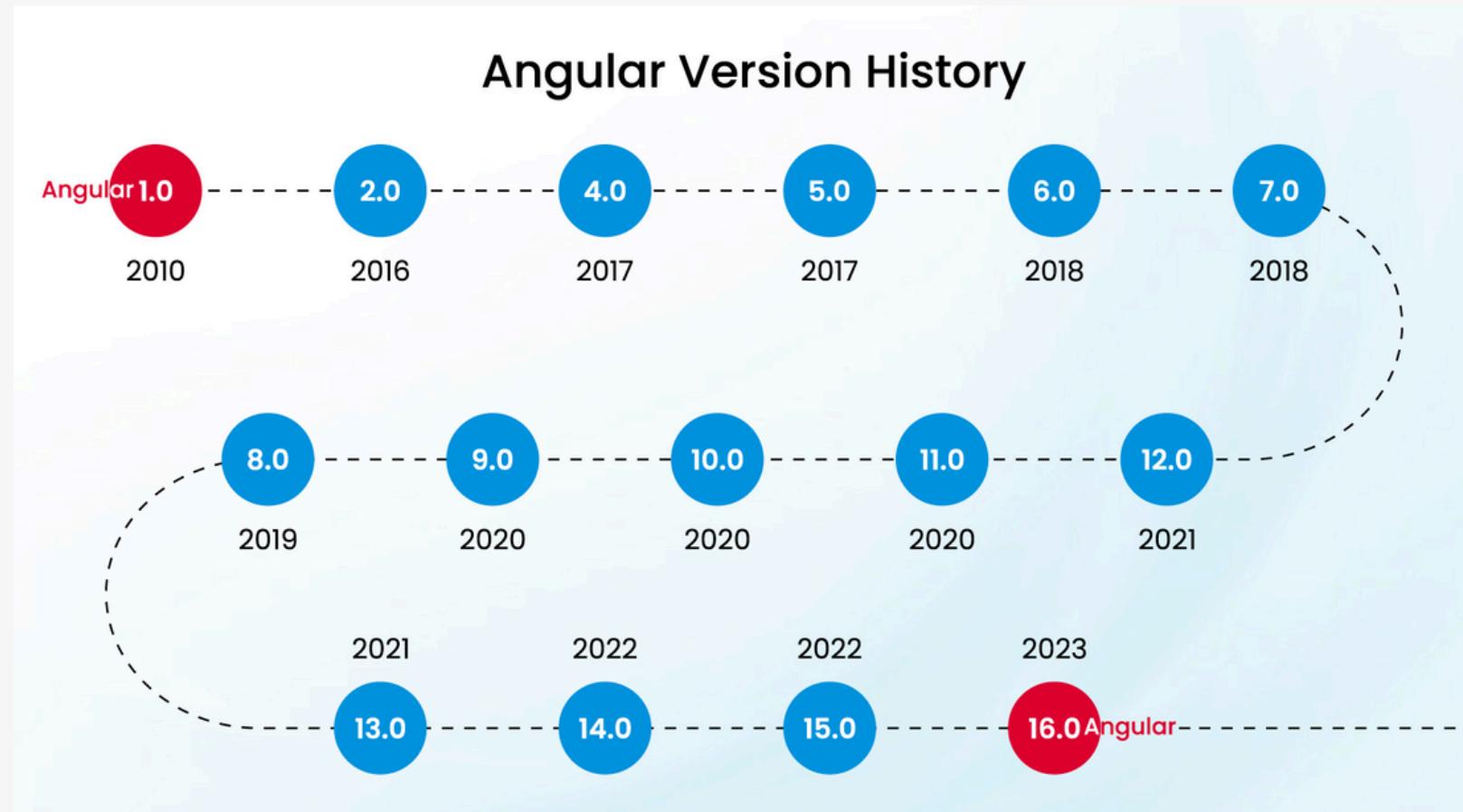
REINVENTING WITH ANGULAR 2 AND BEYOND

- **Angular 2 (2016): complete rewrite, no backward compatibility with AngularJS.**
- **Component-based architecture, TypeScript, and improved performance.**
- **Semiannual updates for continuous improvements and stability.**
- **Key milestones: AOT compilation, Ivy renderer, standalone components.**



ANGULAR 19 – THE PRESENT AND FUTURE

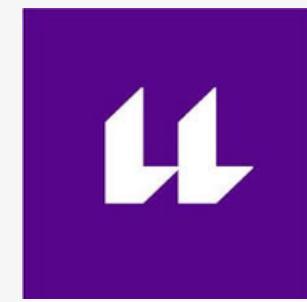
- Signals: reactive system for state management.
- Modern directives: @if, @for, @defer for intuitive DOM control.
- Standalone components: simplified project structure.
- Optimizations: efficient change detection, lazy loading, TypeScript 5 support.
- Built for modern, productive web development.





COMMANDS TO START IN ANGULAR 19

COMMANDS TO START IN ANGULAR 19



To install angular 19:

```
npm install -g @angular/cli@19.2.3
```

To create or firts proyect:

```
ng new <name-of-the-proyект>
```

COMMANDS TO START IN ANGULAR 19



To create a component:

`ng generate component <name-of-the-component>`

To create a service:

`ng generate service <name-of-the-service>`

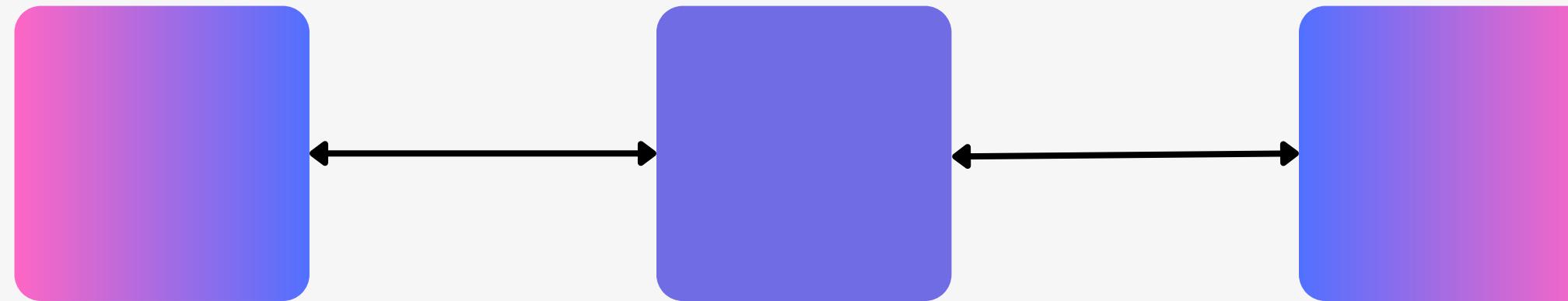
To start the local-web:

`ng serve`

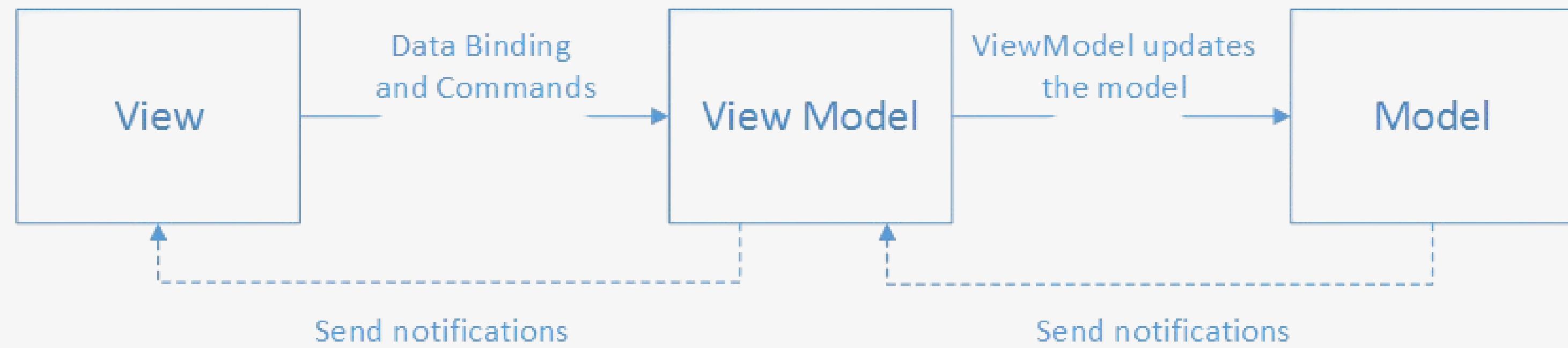


MVVM VS MVC

software architecture patterns



USAGE OF MVVM



Code example

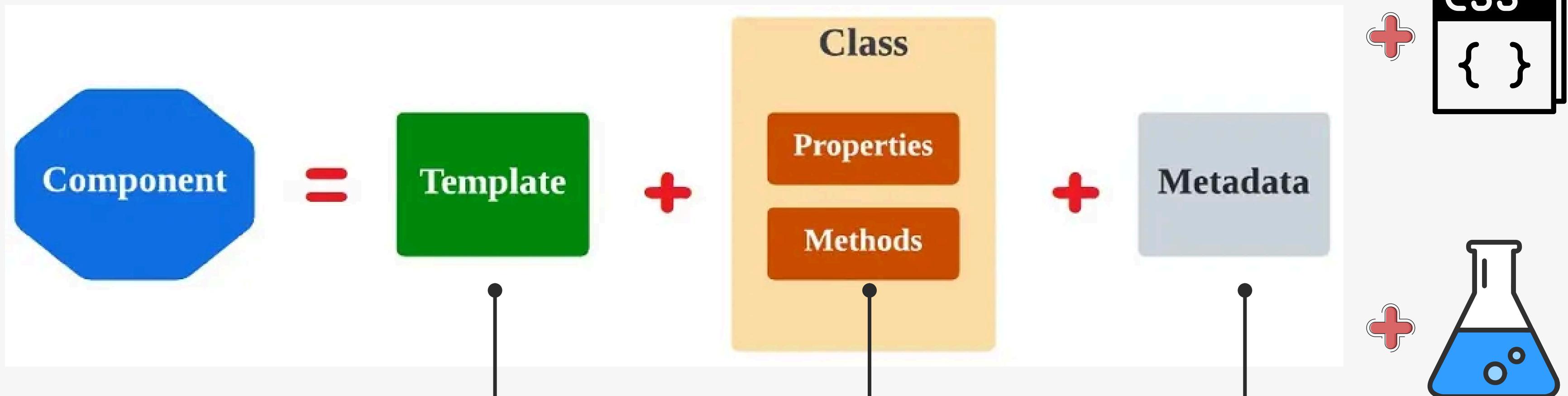




ANGULAR COMPONENTS

- login
- login.component.html
- login.component.scss
- login.component.spec.ts
- login.component.ts

WHAT IS AN ANGULAR COMPONENT?



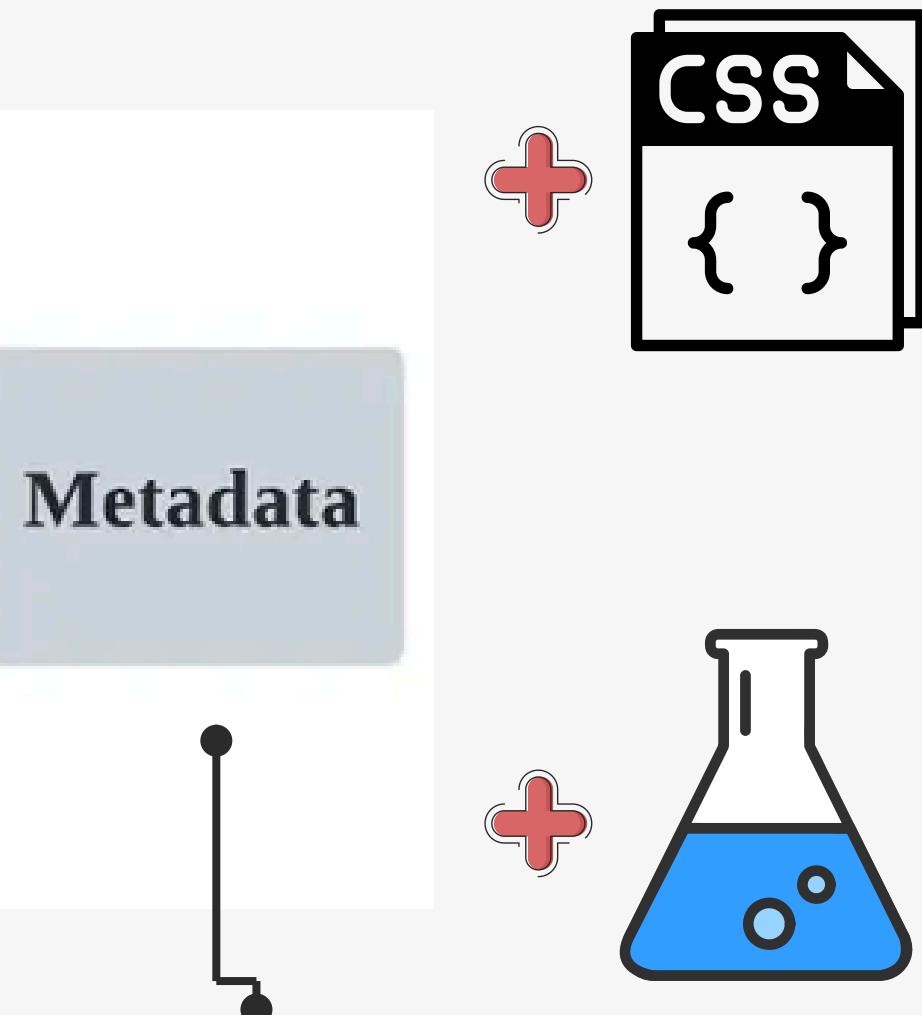
TEMPLATE (HTML)

- what the user will see on screen
- forms, buttons, text fields
- "view" in the MVVM pattern



CLASS (TYPESCRIPT)

- class and logic
 - properties and methods
- implements decorator
- "ViewModel" connects View with Model



METADATA

(@COMPONENT DECORATOR)

- configures selector, template, styles and necessary imports.

TEMPLATE

HTML



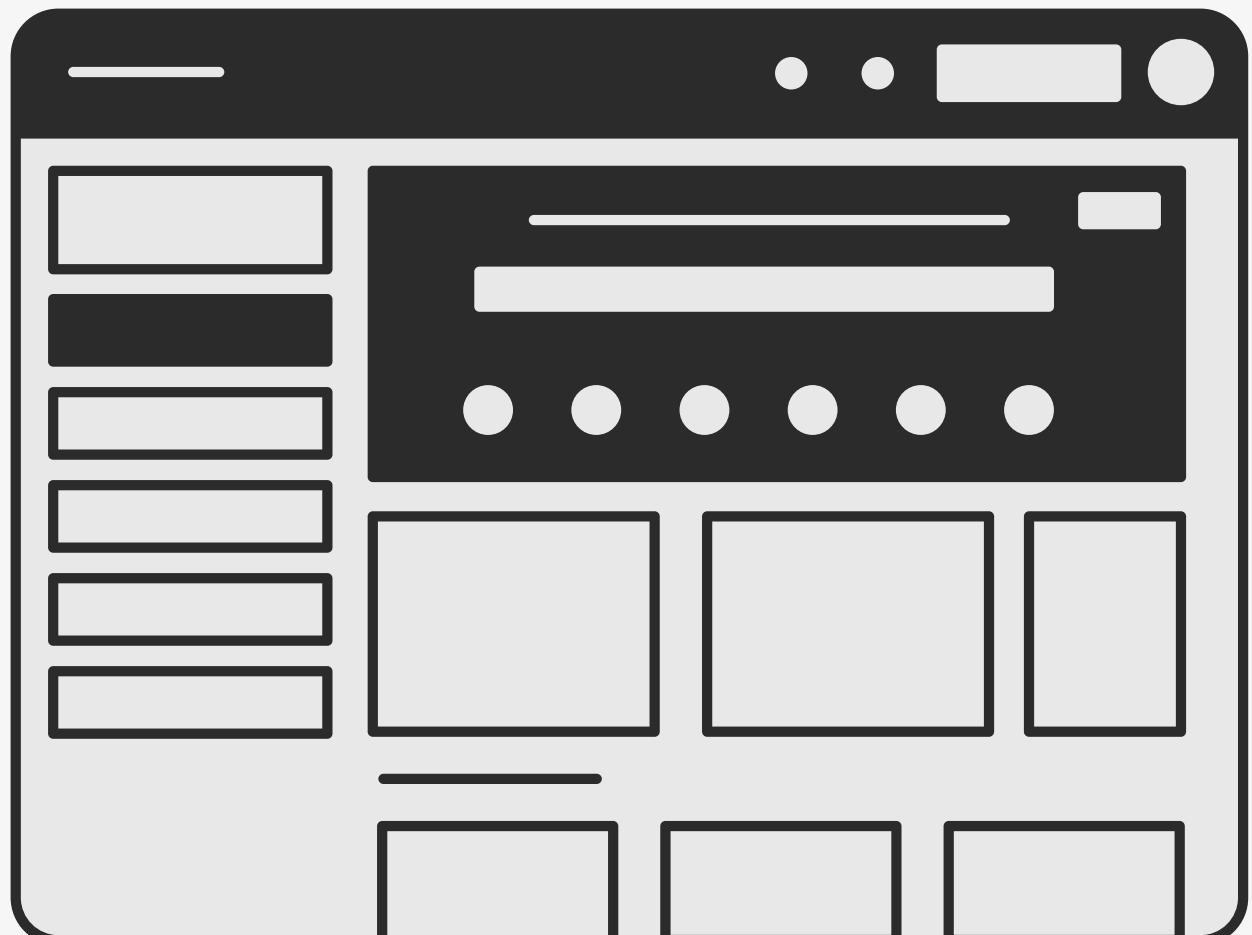
■ “VIEW” IN MVC

■ NOT CONTAIN BUSINESS LOGIC

only presents information
and capture user
interactions

■ CONNECTS WITH “VIEWMODEL”

It connects through data
and events links



HTML



- **[(ngModel)]="username"**
- **(click)="login()"**
- **{{ errorMessage}}**

```
1 <div class="login-container">
2   <h1 class="login-title">Iniciar Sesión</h1>
3   <div class="login-cube box">
4     <div class="field">
5       <label class="label">Usuario</label>
6       <div class="control">
7         <input class="input" type="text" [(ngModel)]="username" placeholder="Usuario">
8       </div>
9     </div>
10    <div class="field">
11      <label class="label">Contraseña</label>
12      <div class="control">
13        <input class="input" type="password" [(ngModel)]="password" placeholder="Contraseña">
14      </div>
15    </div>
16    <div class="field">
17      <div class="control">
18        <button class="button is-primary" (click)="login()>Iniciar Sesión</button>
19      </div>
20    </div>
21    @if (errorMessage) {
22      <p class="has-text-danger">{{ errorMessage }}</p>
23    }
24  </div>
25 </div>
```

Usuario

Contraseña

Iniciar Sesión

Usuario o contraseña incorrectos

COMPONENT CLASS



TS

IMPORTS

bring necessary external functionalities

DECORATOR

Set the component

CLASS

manage logic of login

- properties
- constructor
- login method

```
1 import { Component } from '@angular/core';
1 import { AuthService } from '../../../../../services/auth.service';
2 import { Router } from '@angular/router';
3 import { FormsModule } from '@angular/forms';
4
5 @Component({
6   selector: 'app-login',
7   standalone: true,
8   imports: [FormsModule], // Importa FormsModule aquí
9   templateUrl: './login.component.html',
10  styleUrls: ['./login.component.scss']
11 })
12 export class LoginComponent {
13   username: string = '';
14   password: string = '';
15   errorMessage: string = '';
16
17   constructor(private authService: AuthService, private router: Router) {}
18
19   login(): void {
20     this.authService.login(this.username, this.password).subscribe({
21       next: (isValid) => {
22         if (isValid) {
23           this.router.navigate(['/home']);
24         } else {
25           this.errorMessage = 'Usuario o contraseña incorrectos';
26         }
27       },
28       error: () => {
29         this.errorMessage = 'Error al iniciar sesión';
30       }
31     });
32   }
33 }
```

DECORATOR

TRANSFORM CLASSES INTO
ANGULAR COMPONENTS

DEFINE HOW IS INTEGRATED
INTO THE APPLICATION

CONNECTION BETWEEN LOGIC (TS)
AND PRESENTATION (HTML/CSS)

```
@Component({
  selector: 'app-login',
  standalone: true,
  imports: [FormsModule],
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
```

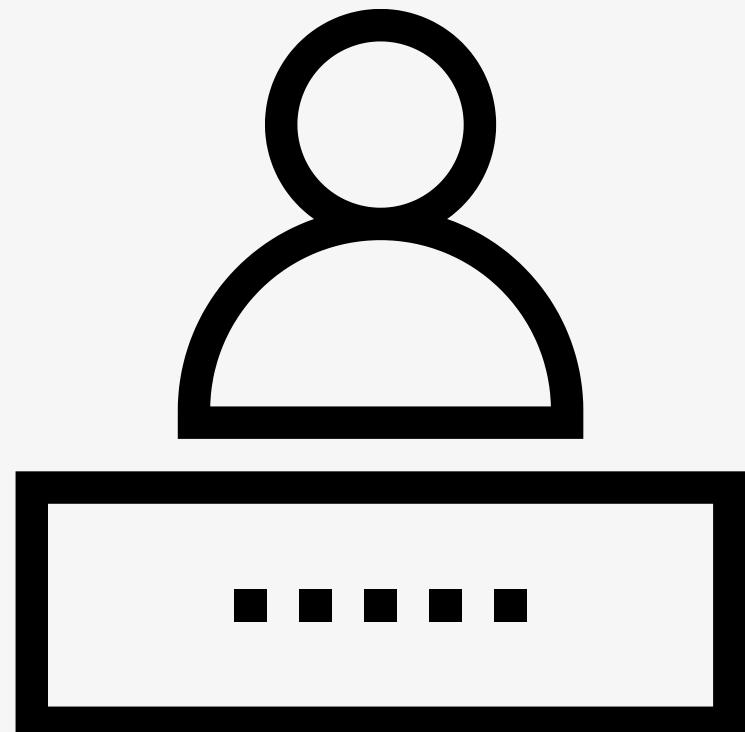
- **@Component** – defines metadata
- **selector** – custom HTML tag
- **standalone** – independent component

- **imports** – required modules
- **templateUrl** – HTML file
- **styleUrls** – CSS styles file



DEMO: LOGIN HOME

- Component-to-Service Interaction: injection of AuthService in LoginComponent
- Use of ngModel for two-way binding and how component properties drive the view



Usuario

Contraseña

Iniciar Sesión

[Code Example: Login Home](#)

Usuario

Usuario

Contraseña

Contraseña

Iniciar Sesión

- 1 INITIAL LOGIN COMPONENT LOAD
- 2 USER INTERACTION WITH THE FORM
- 3 FORM SENDING
- 4 RESPONSE PROCESSING
- 5 TRANSITION TO HOME PAGE

NEW DIRECTIVES IN ANGULAR 19





DIRECTIVE IF

```
@if (a > b) {  
    {{a}} is greater than {{b}}  
}  
@else if (b > a) {  
    {{a}} is less than {{b}}  
}  
@else {  
    {{a}} is equal to {{b}}  
}
```

Insertar enlace al código

DIRECTIVE FOR



```
@for (item of items; track item.name) {  
  <li>{{ item.name }}</li>  
}  
@empty {  
  <li>There are no items.</li>  
}
```

Insertar enlace al código

DIRECTIVE DEFER



```
@defer {  
    <large-component />  
}  
@loading {  
      
}  
@placeholder {  
    <p>Placeholder content</p>  
}
```

Insertar enlace al código

ANGULAR SIGNALS





INTRODUCTION TO SIGNALS IN ANGULAR 19

- Reactive primitives for state management.
- Introduced in Angular 16, fully integrated in Angular 19.
- Simplify change detection and reduce RxJS dependency.
- Core feature for modern, performant applications.





TYPES OF SIGNALS AND THEIR USAGE

- **signal:** Reactive value with read/write capabilities.
- **computed:** Derived value that auto-updates with dependencies.
- **effect:** Side effects triggered by signal changes.
- Declarative and intuitive API for state management.



ADVANTAGES OF SIGNALS IN ANGULAR 19

- Fine-grained reactivity for efficient change detection.
- Zone-less compatibility for better performance.
- Simplified state management compared to RxJS.
- Seamless integration with standalone components.



DEMO: SHOPPING CART WITH SIGNALS

- Standalone components with reusable Product component.
- signal for product quantities, computed for total price.



Shopping Cart

| | |
|---------------------|-------------|
| Awesome Book (\$10) | Quantity: 0 |
| Add | Remove |
| Cool Gadget (\$25) | Quantity: 0 |
| Add | Remove |
| Fancy Mug (\$8) | Quantity: 0 |
| Add | Remove |

Total Price: \$0
Your cart is empty.

[Code Example: Shopping Cart with Signals](#)



SIGNALS IN MODERN ANGULAR DEVELOPMENT

- Foundation for reactive, high-performance apps.
- Enable zone-less architectures and standalone components.
- Reduce complexity in state management.
- Bright future with evolving Angular ecosystem.

CODE EXAMPLE

POKER



Code Example: Poker Game





REFERENCES

[HTTPS://ANGULAR.DEV](https://angular.dev)

[HTTPS://ANGULAR.DEV/GUIDE/DIRECTIVES](https://angular.dev/guide/directives)

[HTTPS://DOCS.ANGULAR.LAT/GUIDE/ARCHITECTURE-COMPONENTS](https://docs.angular.lat/guide/architecture-components)

[HTTPS://V17.ANGULAR.IO/GUIDE/ARCHITECTURE-SERVICES](https://v17.angular.io/guide/architecture-services)

[HTTPS://MATERIAL.ANGULAR.DEV/](https://material.angular.dev/)

[HTTPS://BLOG.ANGULAR.DEV](https://blog.angular.dev)

[HTTPS://LEARN.MICROSOFT.COM/EN-US/DOTNET/ARCHITECTURE/MAUI/MVVM](https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm)

[HTTPS://ANGULAR.DEV/CORE/SIGNALS](https://angular.dev/core/signals)



THANKS!

DO YOU HAVE ANY QUESTIONS?

Marcos Llinares Montes
(alu0100972443@ull.edu.es)

Sergio Nicolás Seguí
(alu0101389936@ull.edu.es)

Lucas de Vera Gil
(alu0101550145@ull.edu.es)