2 Componentes

Los bloques de construcción de Angular.

2.1 Generación de components.

2.1.1 EL CLI y el Angular.json

```
{
  "schematics": {
    "@schematics/angular:component": {
      "changeDetection": "OnPush",
      "flat": true,
      "inlineTemplate": true,
      "inlineStyle": true,
      "skipTests": true
    }
  }
}
```

```
ng g c core/header
ng g c core/footer
```

2.2 Anatomía de un componente: plantillas y lógica.

> ***Contexto*** *explícito en el imports de cada componente standalone.*

```
@Component({
  selector: "lab-root",
  standalone: true,
  imports: [HeaderComponent, FooterComponent],
  template: `
    <lab-header />
    <lab-footer />
  `,
  styles: [],
})
export class AppComponent {}
```

2.2.1 Plantillas para las vistas

```
npm start
```

```
<lab-header />
<p>Angular works!</p>
<router-outlet></router-outlet>
<lab-footer />

<header>
  <nav>
    <a href="">Activity Bookings</a>
  </nav>
</header>

<footer>
  <nav>
    <a [href]="" target="_blank">© 2024 Alberto Basalo</a>
    <button>Accept Cookies</button>
  </nav>
</footer>
```

2.2.2 Propiedades y métodos

```
<header>
  <nav>
    <a href="">{{ title }}</a>
  </nav>
</header>
```

```
export class HeaderComponent {
  title = "Activity Bookings";
}
```

```
export class FooterComponent {
  author = {
    name: "Alberto Basalo",
    homepage: "<https://albertobasalo.dev>",
  };

  year = new Date().getFullYear();

  onAcceptClick() {
    console.log("Cookies accepted!");
  }
}
```

```
<footer>
  <nav>
    <a [href]="author.homepage">© {{ year }} {{ author.name }}</a>
    <button (click)="onAcceptClick()">Accept Cookies</button>
  </nav>
</footer>
```

2.3 Presentación de datos

2.3.1 Datos y transformación

```
ng g c bookings/bookings
```

Carpeta `Domain` para los modelos

```
@Component({
  selector: "lab-bookings",
  standalone: true,
  changeDetection: ChangeDetectionStrategy.OnPush,
  styles: ``,
  imports: [CurrencyPipe, DatePipe, UpperCasePipe],
  template: ``,
})
export class BookingsComponent {
  activity: Activity = {
    name: "Paddle surf",
    location: "Lake Leman at Lausanne",
    price: 100,
    date: new Date(2025, 7, 15),
    minParticipants: 4,
    maxParticipants: 10,
    status: "published",
    id: 1,
    slug: "paddle-surf",
    duration: 2,
    userId: 1,
  };
  currentParticipants = 3;
}
```

```
<article>
  <header>
    <h2>{{ activity.name }}</h2>
    <div [class]="activity.status">
      <span>{{ activity.location }}</span>
```

```
        <span>{{ activity.price | currency }}</span>
        <span>{{ activity.date | date }}</span>
        <span>{{ activity.status | uppercase }}</span>
      </div>
    </header>
    <main>
      <p>Participants: {{ currentParticipants }}</p>
    </main>
    <footer>
      <button>Book now!</button>
      <button>Cancel</button>
    </footer>
  </article>
```

2.3.2 Custom pipes

```
  "@schematics/angular:pipe": {
      "skipTests": true
  },
```

```
ng g pipe bookings/activityTitle
```

```
  @Pipe({
    name: "activityTitle",
  })
  export class ActivityTitlePipe implements PipeTransform {
    transform(activity: Activity, ...args: unknown[]): string {
      return `${activity.name} at ${activity.location}`;
    }
  }
```

```
  <header>
    <h2>{{ activity | activityTitle }}</h2>
    <div [class]="activity.status">
      <span>{{ activity.price | currency }}</span>
      <span>{{ activity.date | date }}</span>
      <span>{{ activity.status | uppercase }}</span>
    </div>
  </header>
```

2.3.3 Estilos

```
npm install @picocss/pico
```

```
"styles": ["node_modules/@picocss/pico/css/pico.min.css",
"src/styles.css"],
```

```
.draft {
  color: violet;
  font-style: italic;
}
.published {
  color: limegreen;
}
.confirmed {
  color: green;
}
.sold-out {
  color: green;
  font-style: italic;
}
.done {
  color: orange;
  font-style: italic;
}
.cancelled {
  color: red;
  font-style: italic;
}
```