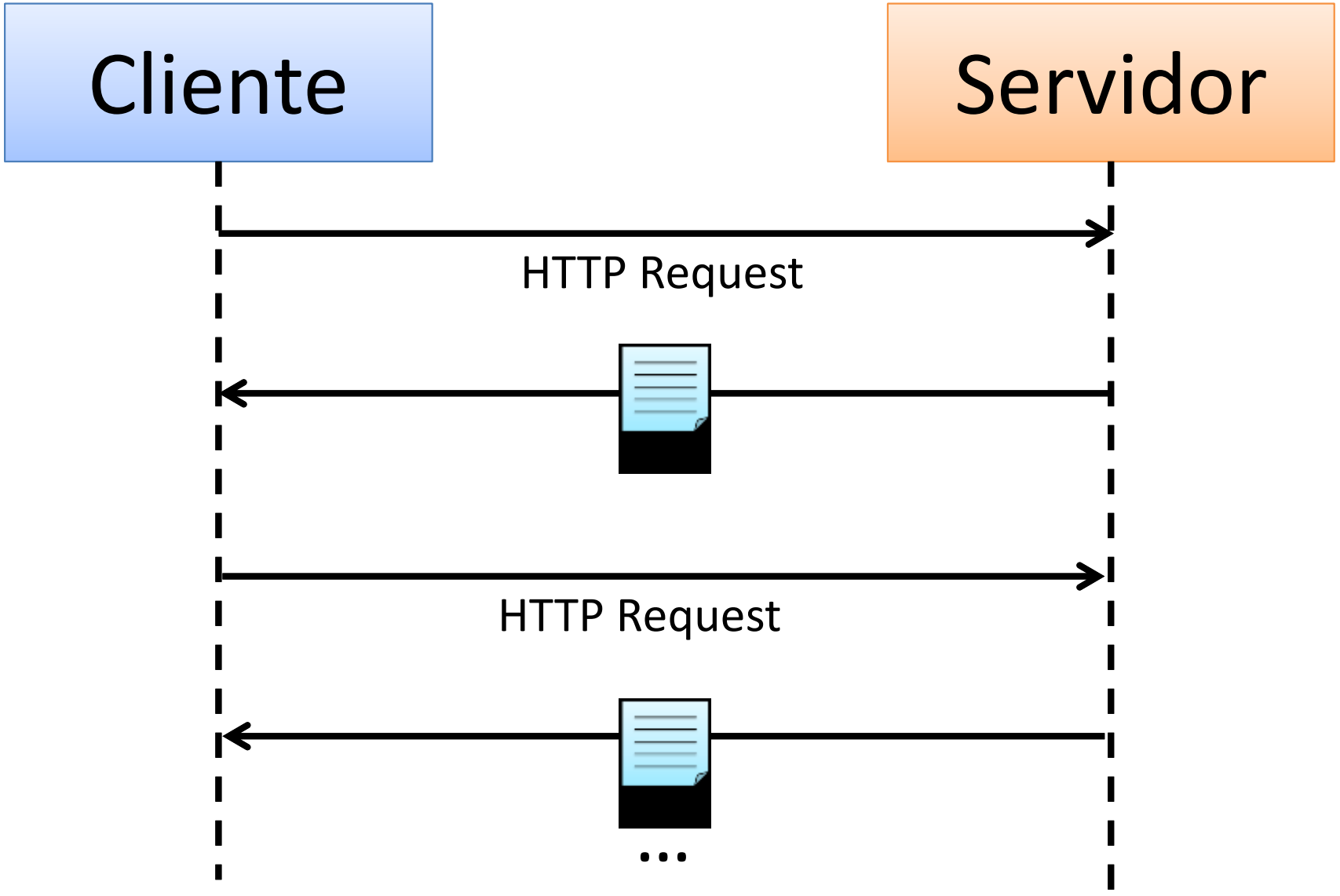
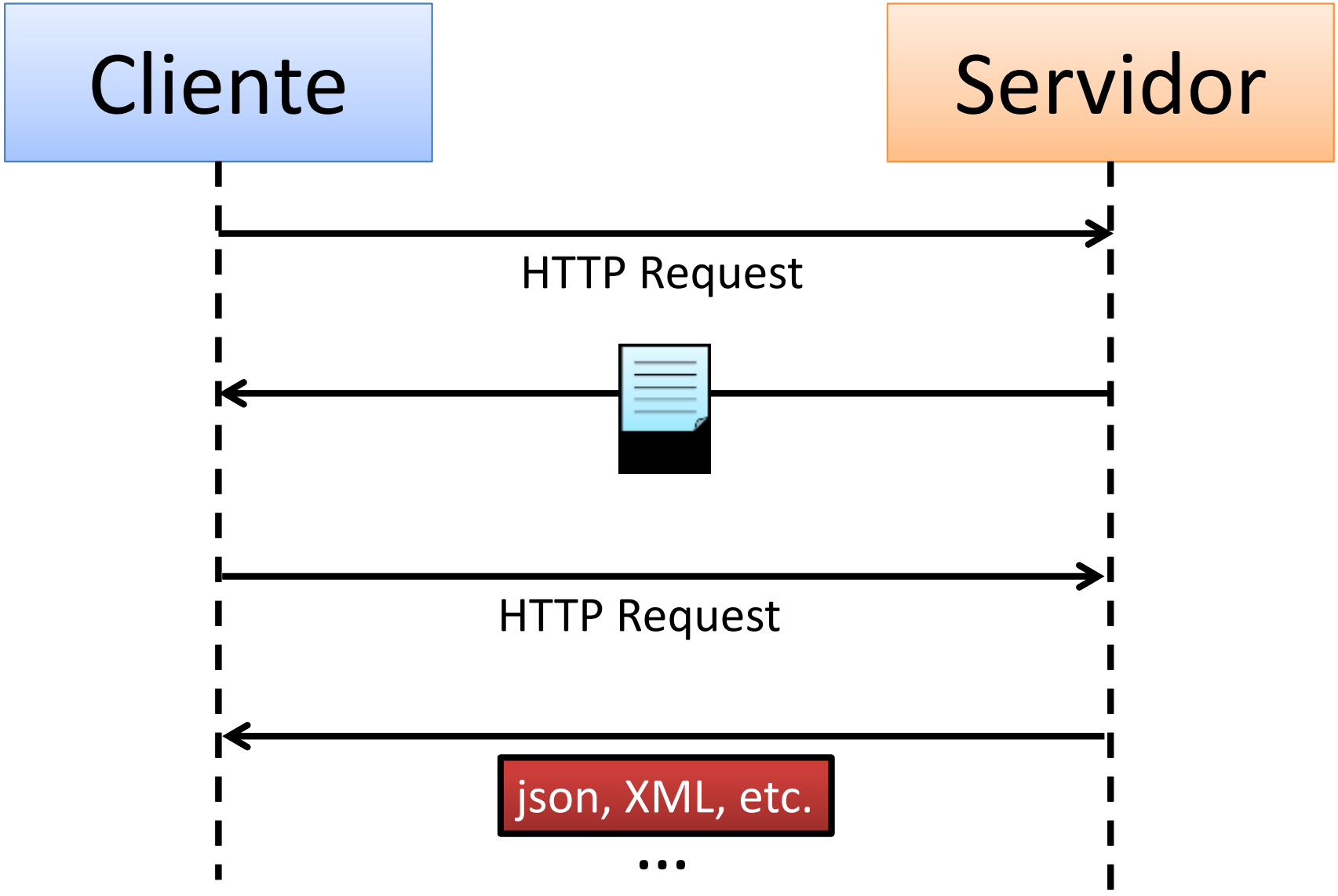
Single Page Applications
&
Angular
(Parte 2)

Jaime A. Pavlich-Mariscal

Multi-Page Applications (MPA)



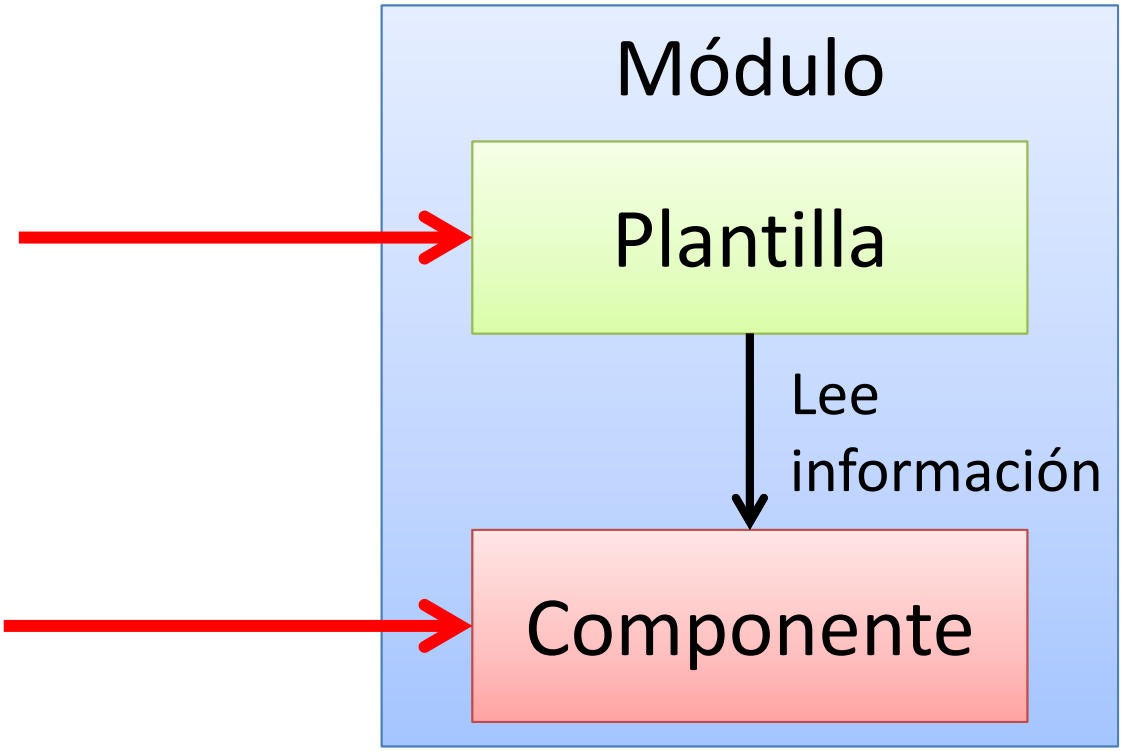
Single-Page Applications (SPA)



Principales conceptos de Angular

HTML
+
extensiones **Angular**

Typescript
+
librerías **Angular**

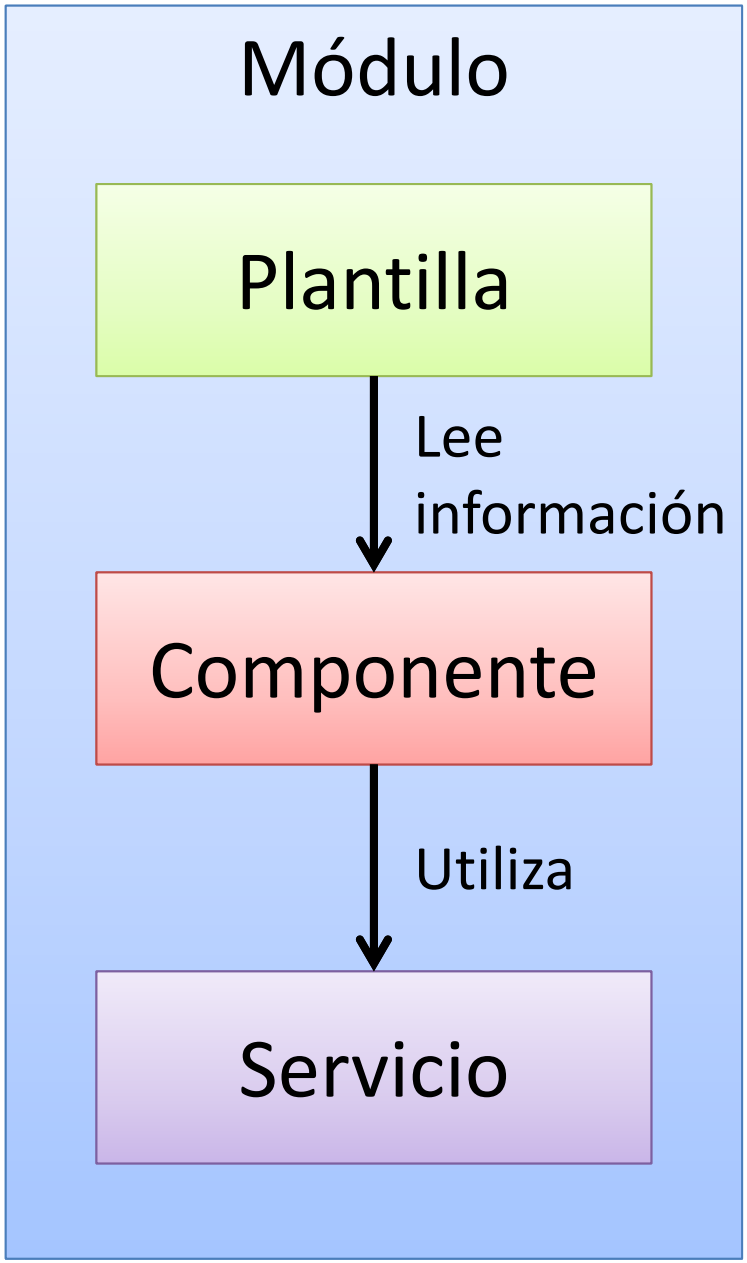


Características avanzadas de Angular

- Servicios e inyección de dependencias
- Enrutamiento
- Programación reactiva

Servicios e inyección de dependencias

Servicios


Módulo
Plantilla
Lee información
Componente
Utiliza
Servicio

Injección de dependencias en Angular

Programación “clásica” (sin inyección de dependencias)

```typescript
export interface Persistence {
    save(obj: any) : void
}

export class XMLPersistence implements Persistence {
    save(obj: any) {
        // Guarda obj en un archivo XML
    }
}

export class SQLPersistence implements Persistence {
    save(obj: any) {
        // Guarda obj en una base de datos SQL
    }
}
```

```typescript
export class DataProcessor {
    persistence : Persistence = new XMLPersistence();
    execute() {
        let d = new Data();
        this.persistence.save(d);
    }
}

let dp = new DataProcessor();
dp.execute();
```

Ventajas?
Desventajas?

Programación con inyección de dependencias

```
export interface Persistence {
    save(obj: any) : void
}

export class XMLPersistence implements Persistence {
    save(obj: any) {
        // Guarda obj en un archivo XML
    }
}

export class SQLPersistence implements Persistence {
    save(obj: any) {
        // Guarda obj en una base de datos SQL
    }
}
```

```
export class InjectableDataProcessor {
    constructor(public persistence: Persistence) {}
    execute() {
        let d = new Data();
        this.persistence.save(d);
    }
}

let idp = new InjectableDataProcessor(new XMLPersistence());
idp.execute();
```

Ventajas?
Desventajas?

Ejemplo

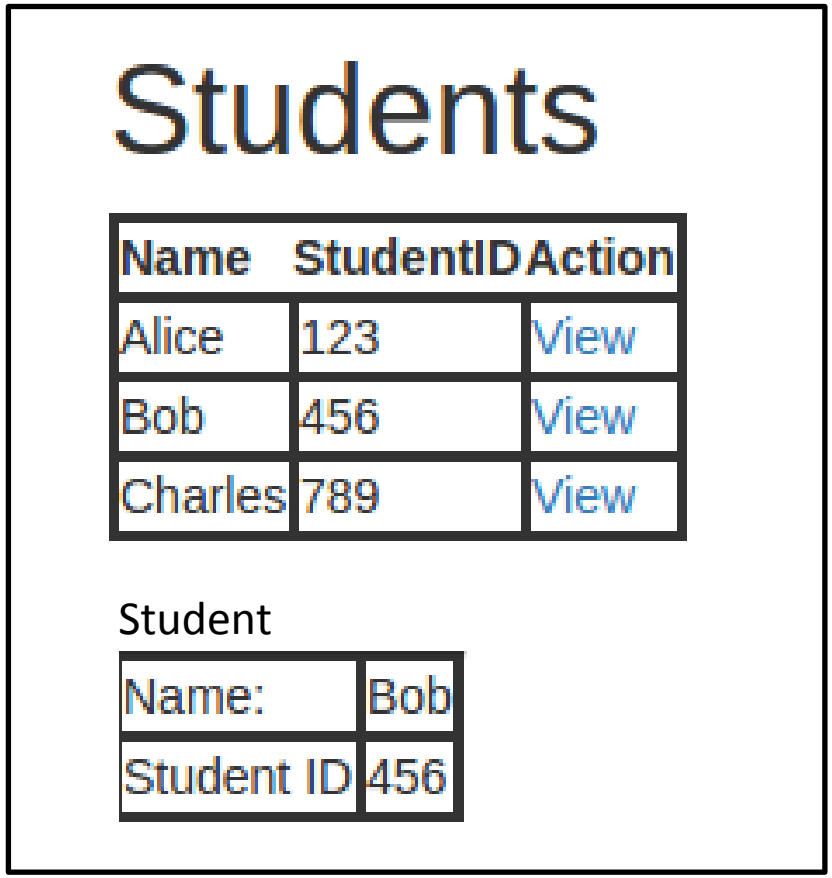
```
git clone https://bitbucket.org/jpavlich/pw.git
```

Ejercicio: modificar el listado de mensajes

- Crear una clase `Mensaje` que guarde cada publicación
- Crear un servicio `MensajeService` que administre los mensajes
  - El componente debe llamar a métodos del servicio para crear y recuperar el listado de mensajes

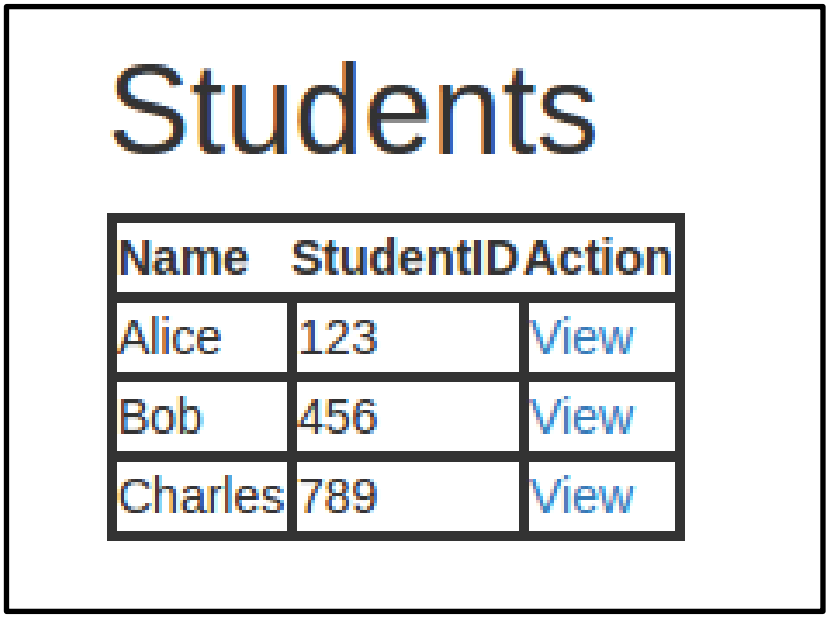
Comunicación entre componentes y Enrutamiento

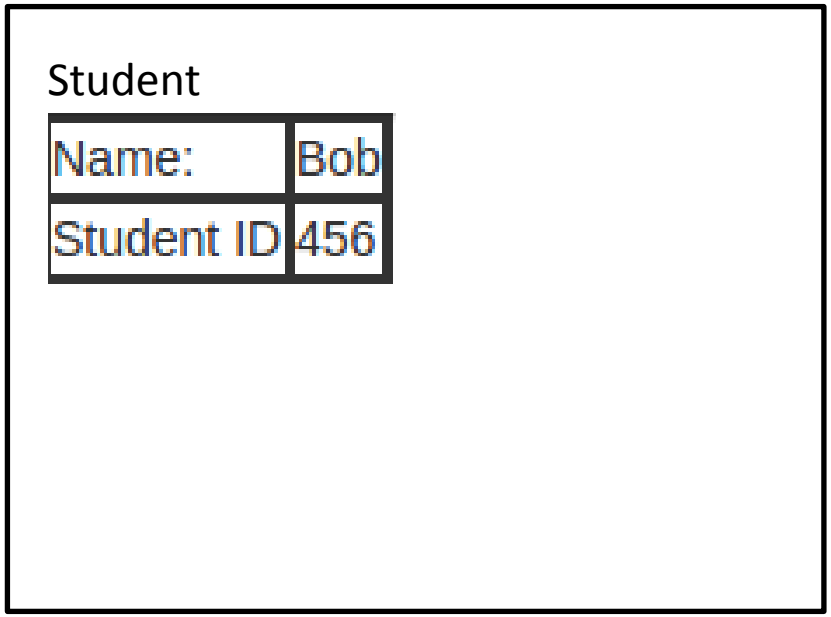
Comunicación entre componentes en una misma página



- Padre -> Hijo
- Hijo -> Padre

Mostrar components en diferentes páginas





- Enrutamiento

Utilizar módulo `@angular/router`

- Creación manual de un proyecto con rutas
  - Agregar anotación `<base>`
  - Utilizar etiqueta `<router-outlet>`
  - Crear e importar módulo de rutas
- Creación automática

```
ng new NOMBRE_PROYECTO --routing
```

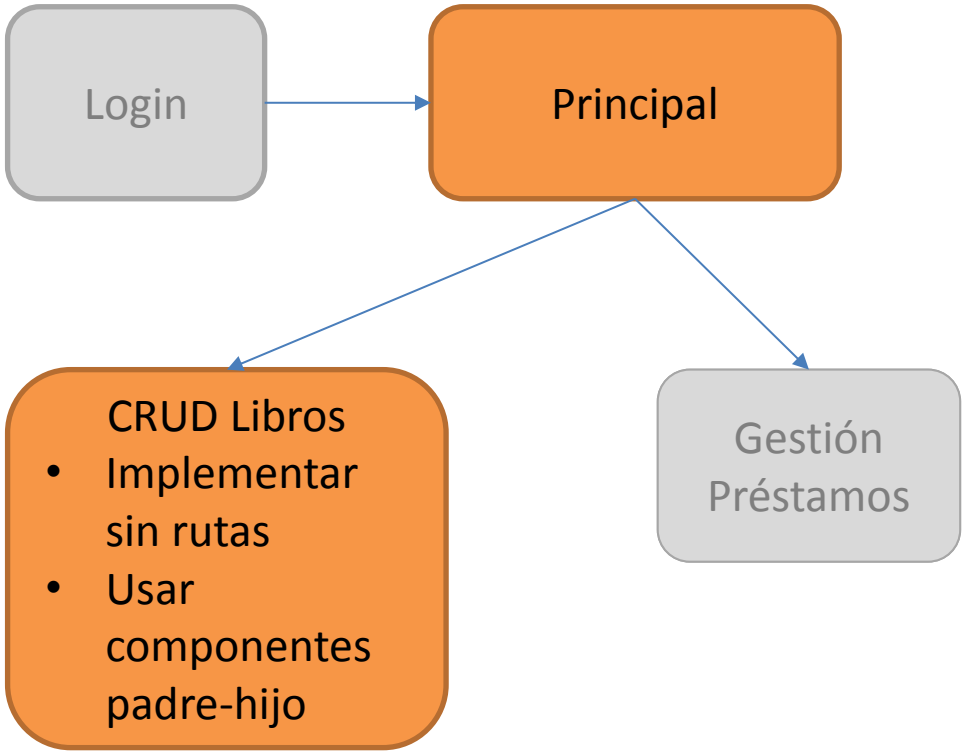
- Pasos posteriores
  - Definir rutas
  - Utilizar atributo `routerLink`
  - Implementar método `ngOnInit()`
  - Utilizar `*ngIf` para mostrar el componente

Ejercicio

- Proyecto semestral:
  - Implementar toda la funcionalidad a la que accede el primer rol del enunciado
  - Sin conexión a base de datos
- Tips:
  - Utilizar servicios “mockups”

Implementar el proyecto desde cero, usando Angular-CLI

No implementar modificando los proyectos de ejemplo!


Login
Principal
CRUD Libros
• Implementar sin rutas
• Usar componentes padre-hijo
Gestión Préstamos