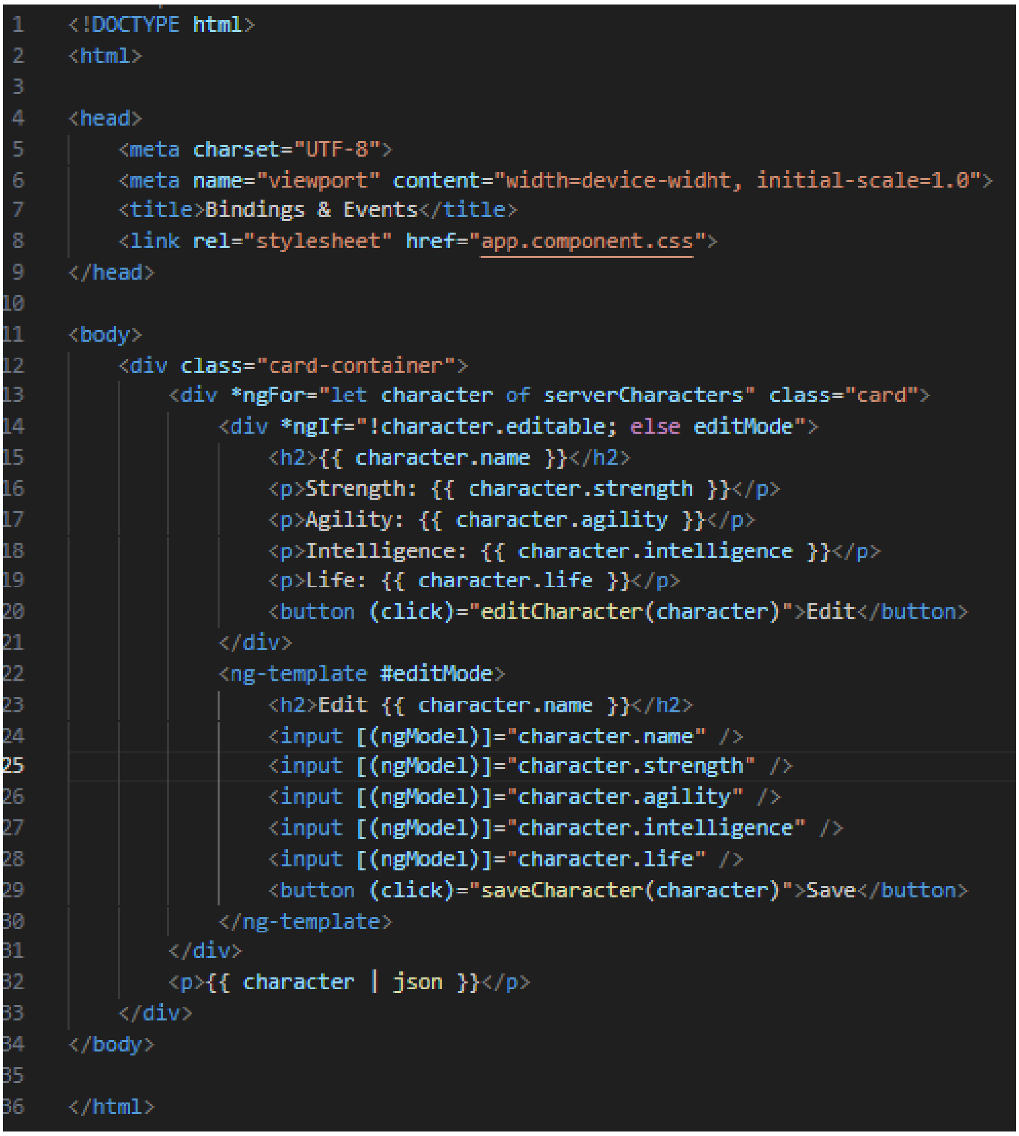
Edgar Chiquillo

Bindings & Events

Requisitos:

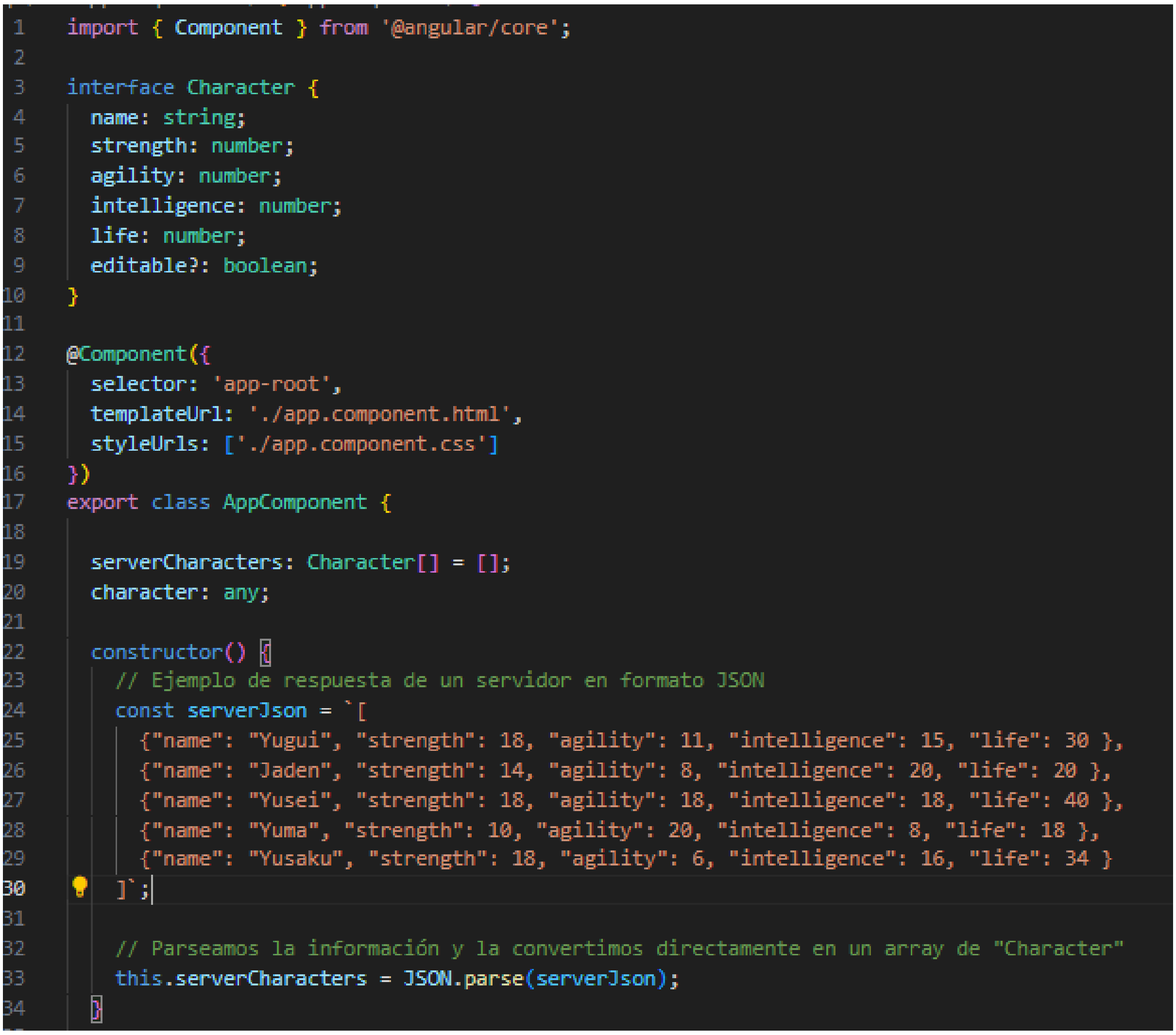
1. Mostrar todos los objetos del array "serverCharacters" en el html.
2. Por defecto no se puede editar ningún objeto del array.
3. Al dar al botón "Edit" te permite editar todos los campos del objeto y automáticamente cambia el nombre a "Save" del botón.
4. Además al dar al botón "Edit" se mostrará en formato texto el objeto correspondiente.
5. Al dar al botón "Save" se guardan los cambios en la variable pertinente.
6. Además al dar al botón "Save" se mostrará en formato texto el objeto correspondiente con las modificaciones hechas.
7. Tenéis que usar los conceptos siguientes: Events & Two Way Binding.

Código HTML

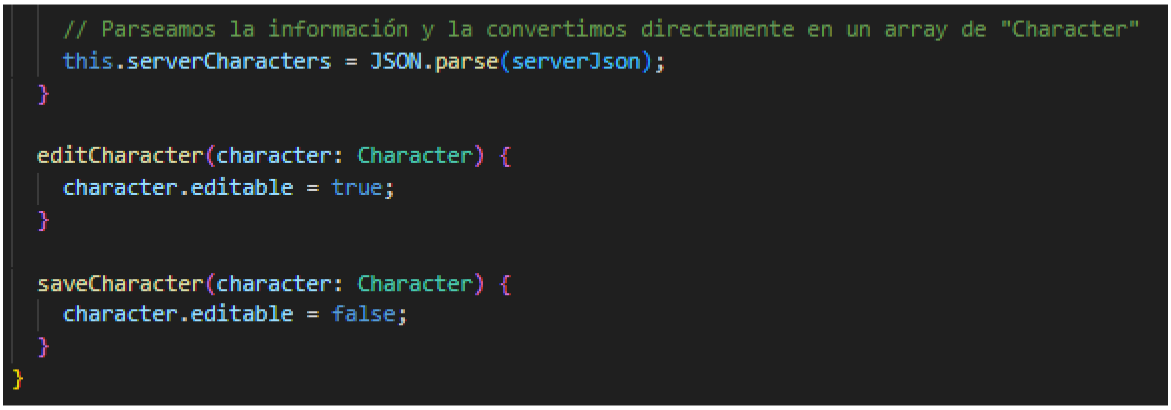


```html
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-widht, initial-scale=1.0">
    <title>Bindings & Events</title>
    <link rel="stylesheet" href="app.component.css">
</head>

<body>
    <div class="card-container">
        <div *ngFor="let character of serverCharacters" class="card">
            <div *ngIf="!character.editable; else editMode">
                <h2>{{ character.name }}</h2>
                <p>Strength: {{ character.strength }}</p>
                <p>Agility: {{ character.agility }}</p>
                <p>Intelligence: {{ character.intelligence }}</p>
                <p>Life: {{ character.life }}</p>
                <button (click)="editCharacter(character)">Edit</button>
            </div>
            <ng-template #editMode>
                <h2>Edit {{ character.name }}</h2>
                <input [(ngModel)]="character.name" />
                <input [(ngModel)]="character.strength" />
                <input [(ngModel)]="character.agility" />
                <input [(ngModel)]="character.intelligence" />
                <input [(ngModel)]="character.life" />
                <button (click)="saveCharacter(character)">Save</button>
            </ng-template>
        </div>
        <p>{{ character | json }}</p>
    </div>
</body>

</html>
```

Código TS

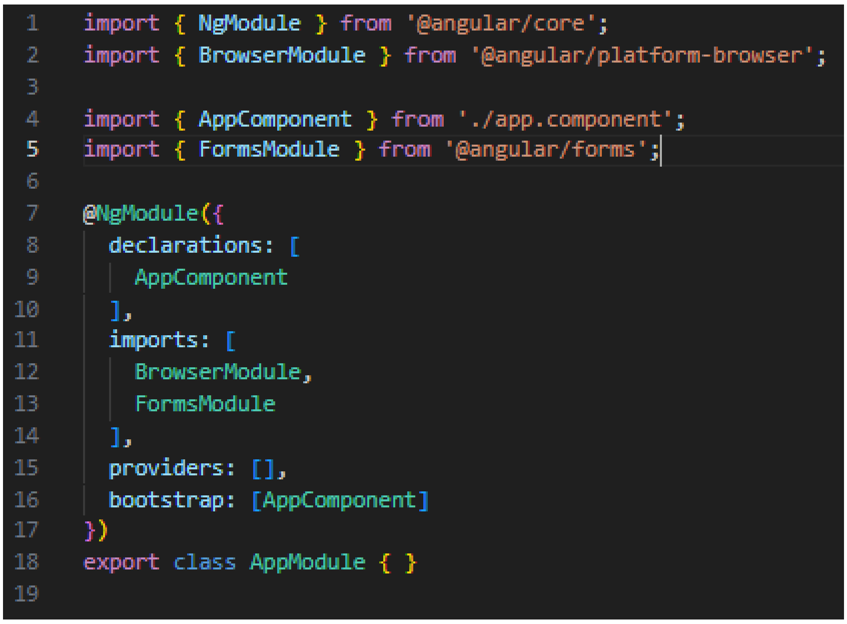


```typescript
import { Component } from '@angular/core';

interface Character {
  name: string;
  strength: number;
  agility: number;
  intelligence: number;
  life: number;
  editable?: boolean;
}

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {

  serverCharacters: Character[] = [];
  character: any;

  constructor() {
    // Ejemplo de respuesta de un servidor en formato JSON
    const serverJson = `[
      {"name": "Yugui", "strength": 18, "agility": 11, "intelligence": 15, "life": 30 },
      {"name": "Jaden", "strength": 14, "agility": 8, "intelligence": 20, "life": 20 },
      {"name": "Yusei", "strength": 18, "agility": 18, "intelligence": 18, "life": 40 },
      {"name": "Yuma", "strength": 10, "agility": 20, "intelligence": 8, "life": 18 },
      {"name": "Yusaku", "strength": 18, "agility": 6, "intelligence": 16, "life": 34 }
    ]`;

    // Parseamos la información y la convertimos directamente en un array de "Character"
    this.serverCharacters = JSON.parse(serverJson);
  }
```



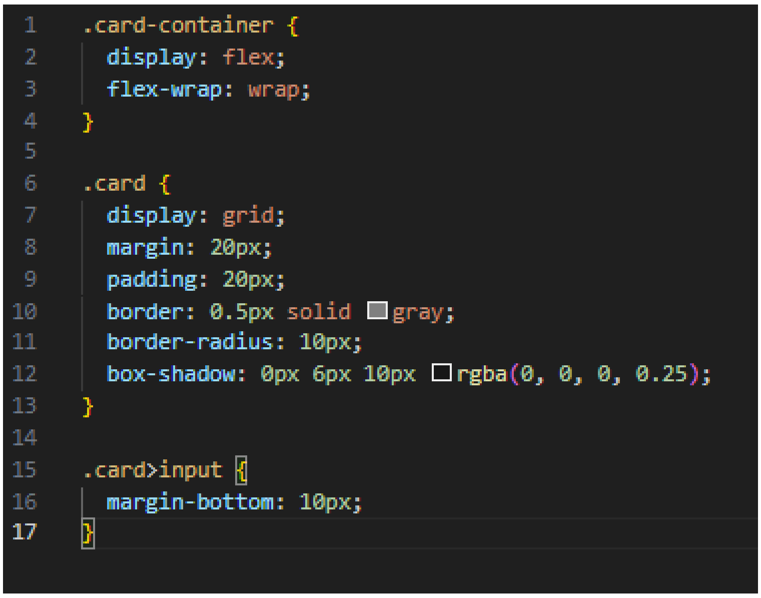
```typescript
    // Parseamos la información y la convertimos directamente en un array de "Character"
    this.serverCharacters = JSON.parse(serverJson);
  }

  editCharacter(character: Character) {
    character.editable = true;
  }

  saveCharacter(character: Character) {
    character.editable = false;
  }
}
```

Código Module TS



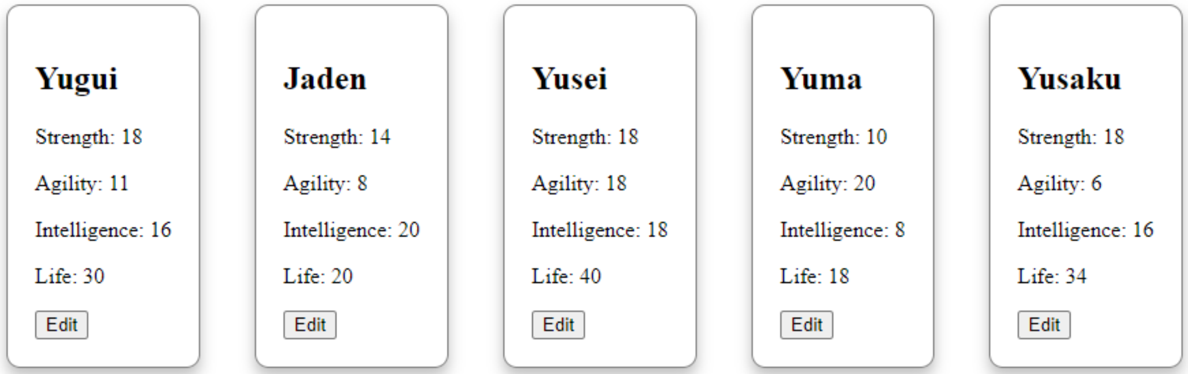
```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

CSS



```css
.card-container {
  display: flex;
  flex-wrap: wrap;
}

.card {
  display: grid;
  margin: 20px;
  padding: 20px;
  border: 0.5px solid gray;
  border-radius: 10px;
  box-shadow: 0px 6px 10px rgba(0, 0, 0, 0.25);
}

.card>input {
  margin-bottom: 10px;
}
```

Resultado



Yugui

Strength: 18

Agility: 11

Intelligence: 16

Life: 30

Jaden

Strength: 14

Agility: 8

Intelligence: 20

Life: 20

Yusei

Strength: 18

Agility: 18

Intelligence: 18

Life: 40

Yuma

Strength: 10

Agility: 20

Intelligence: 8

Life: 18

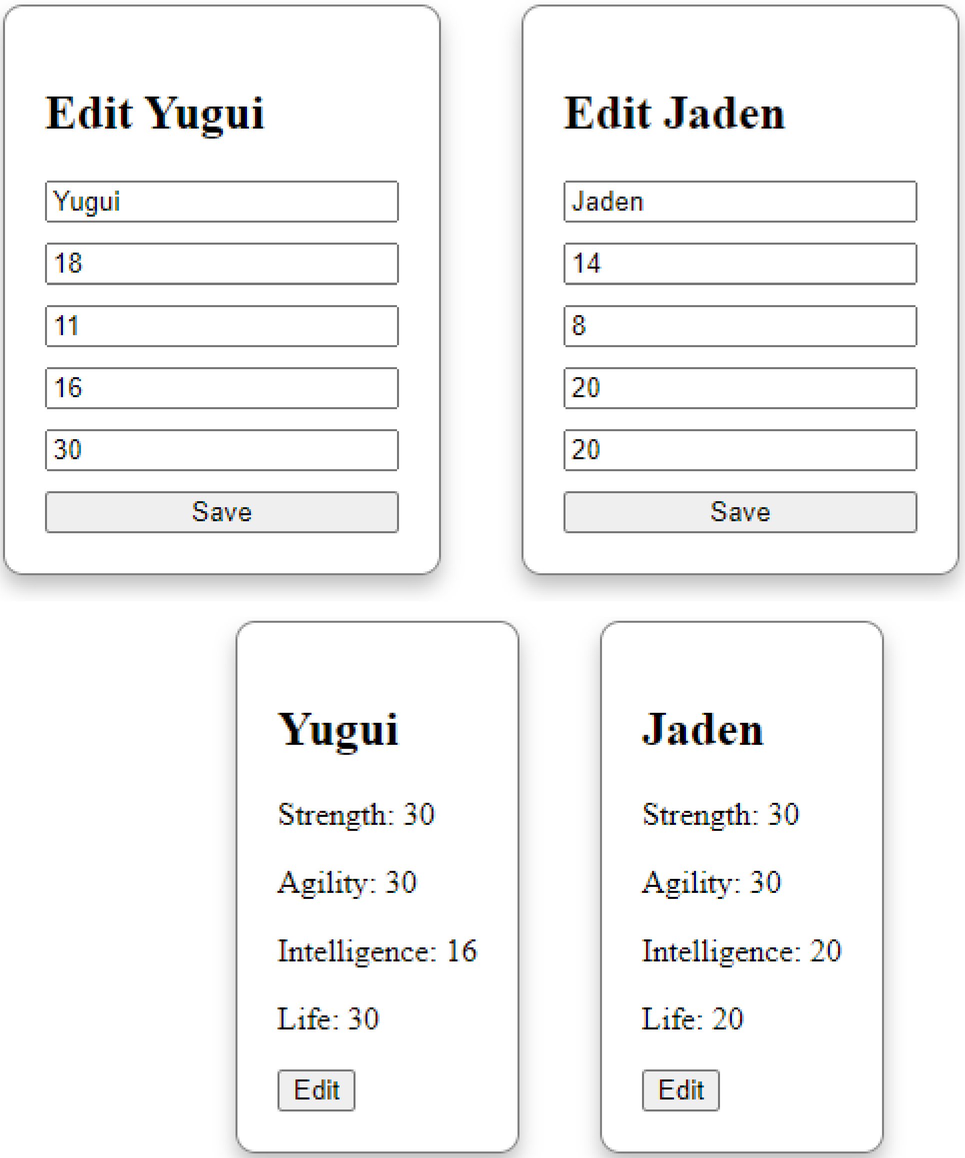
Yusaku

Strength: 18

Agility: 6

Intelligence: 16

Life: 34



Edit Yugui

Yugui

18

11

16

30

Edit Jaden

Jaden

14

8

20

20

Yugui

Strength: 30

Agility: 30

Intelligence: 16

Life: 30

Jaden

Strength: 30

Agility: 30

Intelligence: 20

Life: 20