

Ficha alumnado

Creación de la tabla en la BD

Nos dirigimos a mySQL y creamos la tabla en la base de datos:

```
CREATE TABLE sgi_alumnos (
  id VARCHAR(20) PRIMARY KEY,
  nombre VARCHAR(50) NOT NULL,
  apellidos VARCHAR(50) NOT NULL,
  fecha_nacimiento DATE NOT NULL,
  linkedin VARCHAR(100),
  nivel_ingles ENUM('', 'A1', 'A2', 'B1', 'B2', 'C1', 'C2'),
  minusvalia TINYINT CHECK (minusvalia BETWEEN 0 AND 100),
  otra_formacion VARCHAR(100),
  id_unidad_centro INT(11),
  CONSTRAINT fk_unidad_centro
    FOREIGN KEY (id_unidad_centro)
    REFERENCES sgi_unidades_centro(id_unidad_centro)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

```
MariaDB [app_radfpd]> show columns from sgi_alumnos;
```

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
nombre	varchar(50)	NO		NULL	
apellidos	varchar(50)	NO		NULL	
fecha_nacimiento	date	NO		NULL	
linkedin	varchar(100)	YES		NULL	
nivel_ingles	enum('', 'A1', 'A2', 'B1', 'B2', 'C1', 'C2')	YES			
minusvalia	tinyint(4)	YES		NULL	
otra_formacion	varchar(100)	YES		NULL	
id_unidad_centro	int(11)	YES	MUL	NULL	

9 rows in set (0.000 sec)

Como podemos observar, se ha creado correctamente.

Código

Como hay cantidades ingentes de código, no las voy a poner en el documento, adjunto un zip y comento los cambios relevantes.

Bueno primero creamos el modulo/componente datos-unidad-centro en /src/app/unidades-centro/, el código está adjunto. Lo incluimos en los imports de unidades-centro.module.ts de la siguiente forma:

```
@NgModule({
  declarations: [UnidadesCentroComponent, AddUnidadCentroComponent, DeleteUnidadCentroComponent,
    EditUnidadCentroComponent],
  imports: [
    CommonModule,
    UnidadesCentroRoutingModule,
    CrudMaterialModule,
    DatosUnidadCentroModule
  ]
})
export class UnidadesCentroModule { }
```

Continuamos modificando el unidades-centro.component.html, para añadir el botón del listado y comentamos el botón de edición.

```
<td mat-cell *matCellDef="let unidadCentro">
  <!-- Botón añadido para el listado -->
  <button mat-icon-button (click)="datosUnidadCentro(unidadCentro)">
    <mat-icon>
      assignment
    </mat-icon>
  </button>
  <!-- <button mat-icon-button (click)="editUnidadCentro(unidadCentro)">
    <mat-icon>
      mode_edit
    </mat-icon>
  </button-->
  <button mat-icon-button (click)="deleteUnidadCentro(unidadCentro)">
    <mat-icon>
      delete
    </mat-icon>
  </button>
</td>
```

Ahora, en unidades-centro.component.ts, añadimos dos async que utilizaremos más tarde dentro del nuevo módulo datos-unidad-centro:

```
async datosUnidadCentro(unidadCentro: UnidadCentro) {
  const UNIDAD_CENTRO = unidadCentro;
  const CICLOS = await this.getCiclos();

  if (UNIDAD_CENTRO) {
    const dialogRef = this.dialog.open(DatosUnidadCentroComponent, {
      width: '70em',
      maxWidth: '70em',
      scrollStrategy: this.overlay.scrollStrategies.noop(),
      disableClose: true,
      data: {
        unidadCentro: UNIDAD_CENTRO,
        ciclos: CICLOS,
      }
    });

    const RESULT = await dialogRef.afterClosed().toPromise();
  }
}
```

```

async getCiclos() {
  const RESPONSE = await this.servicioCiclos.getAllCiclos().toPromise();
  if (RESPONSE.ok) {
    console.log("OK");
    return RESPONSE.data as Ciclo[];
  }
}

```

Luego creamos también la interfaz datos-editar-unidad-centro en la carpeta /src/app/shared/interfaces/, y modificamos el servicio unidad-centro.service.ts, en /src/app/services/, para que incluya los siguientes setters:

```

setUnidadCentro(unidadCentro: UnidadCentro) {
  this.unidadCentro = unidadCentro;
}

setDatosBasicosUnidadCentro(formUnidadCentro: any) {
  this.unidadCentro.id_unidad_centro = formUnidadCentro.id_unidad_centro;
  this.unidadCentro.id_ciclo = formUnidadCentro.id_ciclo;
  this.unidadCentro.unidad_centro = formUnidadCentro.unidad_centro;
  this.unidadCentro.observaciones = formUnidadCentro.observaciones;
}

```

Dentro del modulo datos-unidad-centro, creamos el módulo datos-basicos-unidad-centro, que mostrará la edición de la unidad centro (la primera pestaña de la ficha). También está adjunto el código.

Ahora, creamos dentro de datos-unidad-centro el módulo alumnos-unidad-centro, el cual mostrará la lista de alumnos de la unidad centro que tengamos seleccionada. Se podrá acceder a él mediante el sidenav, será la segunda pestaña. También tenemos que crearle la interfaz de alumno, el service y, en el backend, debemos crear el php para interactuar con la base de datos. No se nos puede olvidar el pipe para la edad.

Resultado

Adjunto un vídeo de la ficha implementada y funcional, así como todo el código del módulo unidades-centro, las interfaces, los servicios y el php del backend.