

**TECNOLOGICO NACIONAL DE MEXICO**

**INTITUTO TECNOLÓGICO DE OAXACA**

**Materia:**

Programación web

**Tema:**

Ejercicio Material Design - Repositorio Git

**Carrera:**

Ingeniería en sistemas Computacionales

**Docente:**

Martínez Nieto Adelina

**Alumno:**

Hernández Velasco Ricardo

**Grupo:** 7V2

Oaxaca de Juárez, Oaxaca a 16 de julio de 2024

## Tabla de contenido



### Contenido

Tabla de contenido.....	2
Ejercicios.....	3
AutoComplete.....	3
Badge-position.....	4
Button .....	6
Basic button-toggles.....	8
Card.....	9
CheckBox .....	11
Chips .....	12
Bibliografía .....	15

## Ejercicios

### AutoComplete.

Para realizar el siguiente ejercicio primero se ingreso a la pagina de Material Desing de angular y se presiono un clic en el botón del primer componente

Simple autocomplete  

HTML TS CSS

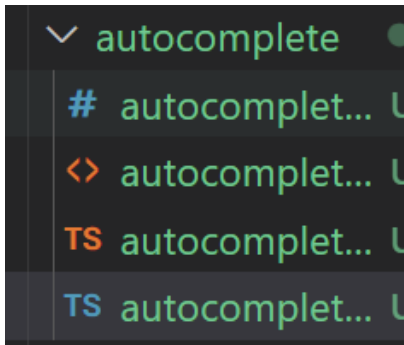
```
<form class="example-form">
  <mat-form-field class="example-full-width">
    <mat-label>Number</mat-label>
    <input type="text"
      placeholder="Pick one"
      aria-label="Number"
      matInput
      [formControl]="myControl"
      [matAutocomplete]="auto">
    <mat-autocomplete #auto="matAutocomplete">
      @for (option of options; track option) {
        <mat-option [value]="option">{{option}}</mat-option>
      }
    </mat-autocomplete>
  </mat-form-field>
</form>
```

Number

Luego se creo un componente en visual studio code con el siguiente comando

```
ricar@LAPTOP-E8E9TKT7 MINGW64
$ ng g c autocomplete
```

Posteriormente se crea una serire de archivos para poder ingresar el TS, HTML y CSS.



Donde en ts agregamos las importaciones y los métodos que aparezcan en el componente y lo mismo se hace en html y css.

Posteriormente se agrega el selector en el html del archivo principal.

```
<app-autocomplete></app-autocomplete>
```

Para posteriormente importar el método en el TS del archivo principal

```
import { AutocompleteComponent } from './autocomplete/autocomplete.component';  
import { RouterOutlet } from '@angular/router';
```

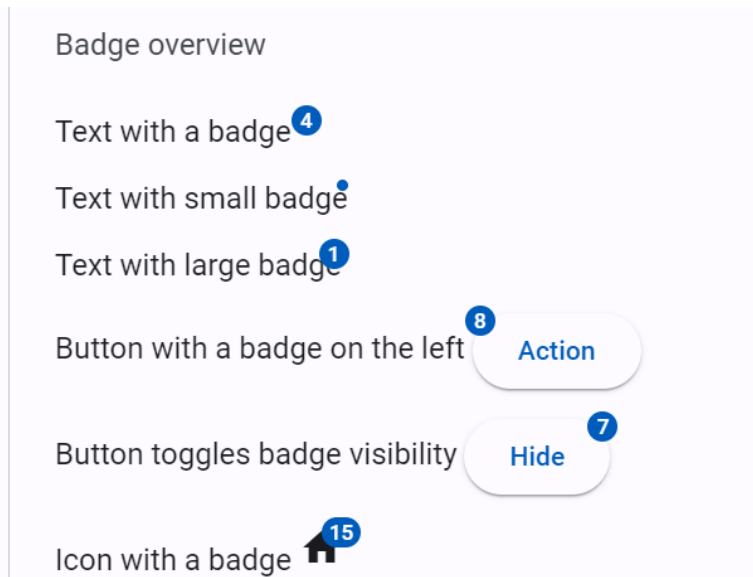
```
imports: [RouterOutlet,  
  Autocomplit2Component,  
  AutocompleteComponent,
```

Posteriormente se ingresa el siguiente comando “**ng serve**” para visualizar el componente de manera grafica

Number

## Badge-position

Para este componente se hace lo mismo se ingresa en la pagina de Material Desing y se selecciona el componente a copiar.



Se selecciona el componente y se copian las importaciones y los métodos del componente

```
import {Component} from '@angular/core';
import {MatIconModule} from '@angular/material/icon';
import {MatButtonModule} from '@angular/material/button';
import {MatBadgeModule} from '@angular/material/badge';

/**
 * @title Badge overview
 */
@Component({
  selector: 'badge-overview-example',
  templateUrl: 'badge-overview-example.html',
  styleUrls: ['badge-overview-example.css'],
  standalone: true,
  imports: [MatBadgeModule, MatButtonModule, MatIconModule],
})
export class BadgeOverviewExample {
  hidden = false;

  toggleBadgeVisibility() {
    this.hidden = !this.hidden;
  }
}
```

Posteriormente se copian los archivos HTML y CSS en nuestro proyecto para posteriormente copiar el selector del archivo y pegarlo en el html del proyecto principal.

```
<app-autocomplit2></app-autocomplit2>
<app-autocomplete></app-autocomplete>
<app-badge-position></app-badge-position>
```

Al igual que se copia el método del archivo y ese se pega en el archivo TS principal del proyecto y se realiza las importaciones.

```
import { BadgePositionComponent } from './badge-position/badge-position.component';
```

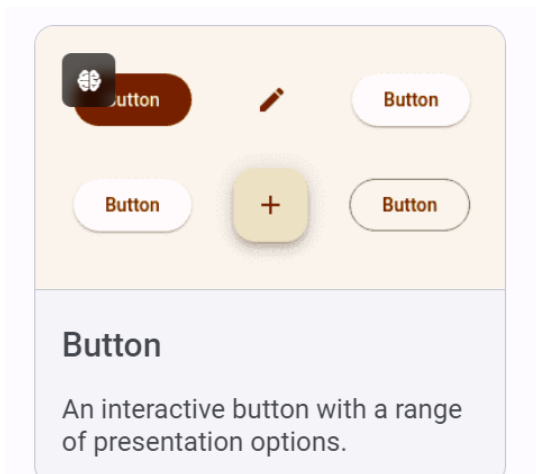
```
imports: [RouterOutlet,
  Autocomplit2Component,
  AutocompleteComponent,
  BadgePositionComponent,
```

Y posteriormente visualizamos el resultado en la pantalla



## Button

Para el boton hacemos exactamente lo mismo que en los anteriores ejercicios, nos dirigimos al componente para hacer un clic sobre el mismo y y posteriormente copiar las importaciones del TS, HTML, Y CSS en caso que lo tenga.



```
import {Component} from '@angular/core';
import {MatIconModule} from '@angular/material/icon';
import {MatDividerModule} from '@angular/material/divider';
import {MatButtonModule} from '@angular/material/button';

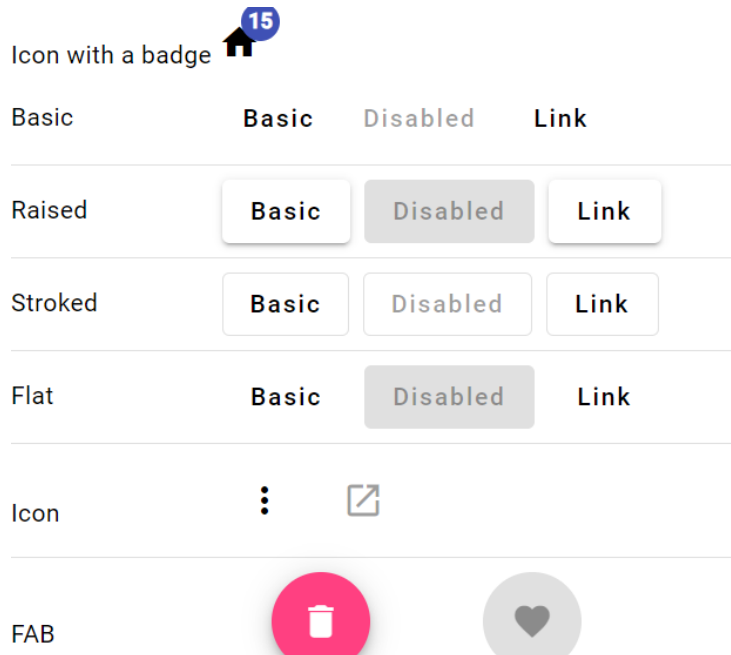
/**
 * @title Basic buttons
 */
@Component({
  selector: 'button-overview-example',
  templateUrl: 'button-overview-example.html',
  styleUrls: ['button-overview-example.css'],
  standalone: true,
  imports: [MatButtonModule, MatDividerModule, MatIconModule],
})
export class ButtonOverviewExample {}
```

Posteriormente nos dirigimos al html principal y pegamos el selector del TS que acabamos de ingresar y posteriormente importamos el método en el TS de la clase principal.

```
import { ButtonComponent } from './button/button.component';
```

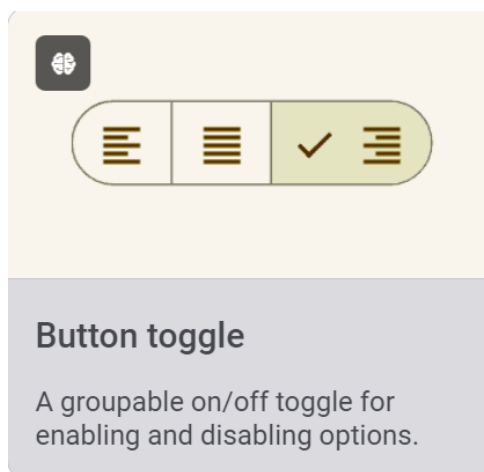
```
imports: [RouterOutlet,
  Autocomplit2Component,
  AutocompleteComponent,
  BadgePositionComponent,
  ButtonComponent,
  SelectionComponent,
```

Para finalizar visualizamos el componente en la pantalla



## Basic button-toggles

Con este componente se hace exactamente lo mismo que el anterior ingresamos a la pagina y presionamos un clic sobre el componente para ingresar a ello y copiamos los imports del TS así como los archivos HTML y CSS.





```

import {Component} from '@angular/core';
import {MatButtonModule} from '@angular/material/button-toggle';

/**
 * @title Basic button-toggles
 */
@Component({
  selector: 'button-toggle-overview-example',
  templateUrl: 'button-toggle-overview-example.html',
  standalone: true,
  imports: [MatButtonModule],
})
export class ButtonToggleOverviewExample {}

```

Posteriormente los agregamos a nuestro proyecto y copiamos el selector y la clase para ingresarlos a nuestro html principal y a nuestro TS.

```

8 import { ButtonTogglesComponent } from './button-toggles/button-toggles.component';

```

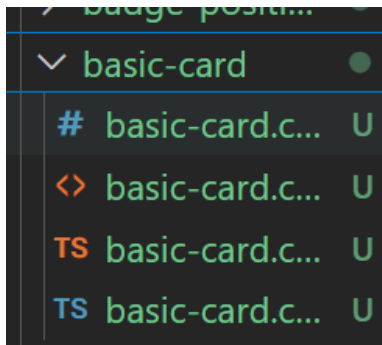
```

ButtonTogglesComponent,

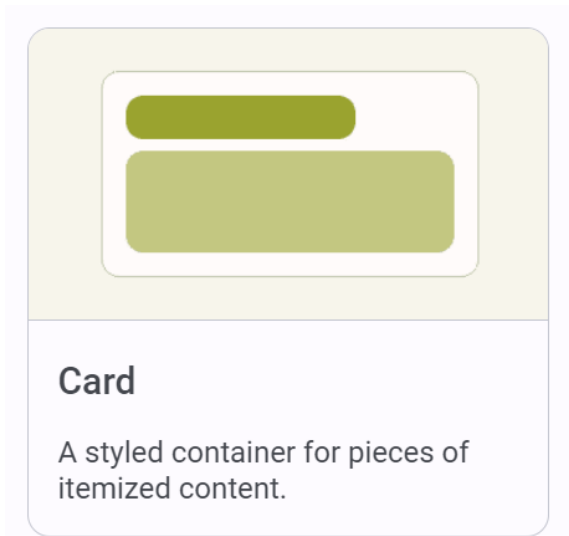
```

## Card

Para el siguiente componente creamos el proyecto con “ng g c Card”



una vez creado los archivos no dirigimos a la pagina para ingresar al componente y copiar los archivos TS , html y css del componete y pegarlo a nuestro proyecto



Se copia los componentes a nuestro proyecto

```
Basic cards

HTML TS

import {ChangeDetectionStrategy, Component} from '@angular/core';
import {MatCardModule} from '@angular/material/card';

/**
 * @title Basic cards
 */
@Component({
  selector: 'card-overview-example',
  templateUrl: 'card-overview-example.html',
  standalone: true,
  imports: [MatCardModule],
  changeDetection: ChangeDetectionStrategy.OnPush,
})
export class CardOverviewExample {}
```

```
Basic cards

HTML TS

<mat-card appearance="outlined">
  <mat-card-content>Simple card</mat-card-content>
</mat-card>

Simple card
```

Y posteriormente se copia el selector y se agrega al HTML principal, así como la Clase a nuestro archivo TS.

```
import { BasicCardComponent } from './basic-card/basic-card.component';
```

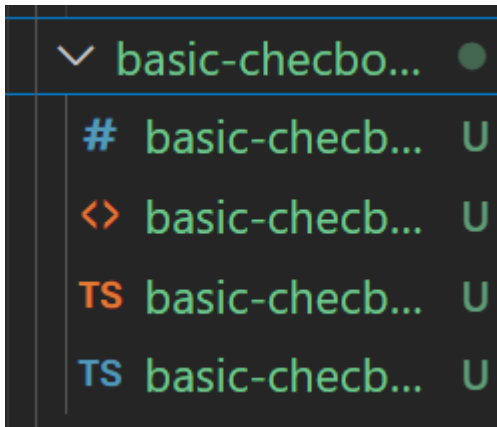
```
BasicCardComponent,
```

Y para finalizar se visualiza en la pantalla

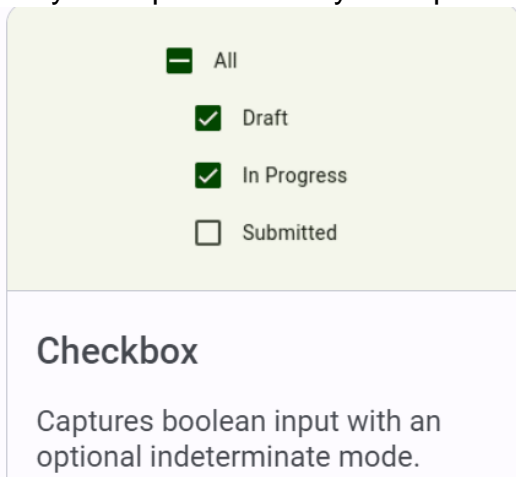
Simple card

## CheckBox

Se crea el proyecto ingresando el la siguiente palabra “ng g c checkbox” y posteriormente se abre la carpeta ts del archivo.



Posteriormente ingresamos al componente en la pagina y copiamos los imports de TS y se copia el HTML y CSS para agregarlo a nuestro proyecto



Basic checkboxes

HTMLTSCSS

```
<section class="example-section">
  <mat-checkbox class="example-margin">Check me!</mat-checkbox>
  <mat-checkbox class="example-margin" [disabled]="true">Disabled</mat-checkbox>
</section>

<section class="example-section">
  <span class="example-list-section">
    <mat-checkbox
      class="example-margin"
      [checked]="task().completed"
      [indeterminate]="partiallyComplete()"
      (change)="update($event.checked)"
    >
      {{task().name}}
    </mat-checkbox>
  </span>
</span class="example-list-section">
```

Como siguiente paso se copia el selector y la clase y se pega en el archivo principal para posteriormente visualizarlo en pantalla.

☐ Check me! ☐ Disabled

☐ Parent task

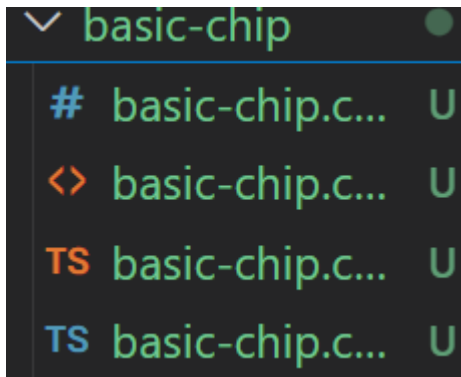
☐ Child task 1

☐ Child task 2

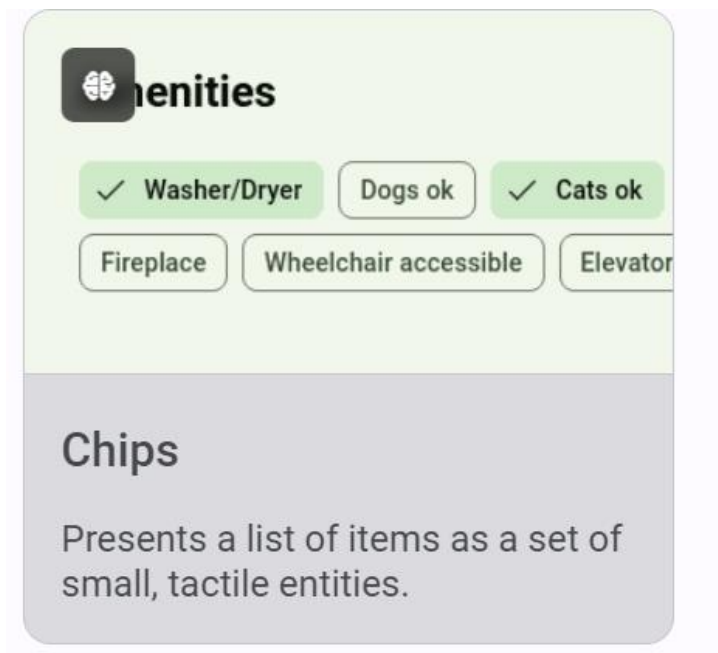
☐ Child task 3

## Chips

Se crea el componente en angular ingresando el siguiente comando “ng g c chips” para que se cree los archivos.



Posteriormente se ingresa a la pagina de angular materias para seleccionar el componente



Y ingresar al código del componente para posteriormente copiar el TS y los imports para pegarlos en nuestro proyecto, así también se copia el html y el css

## Basic chips

HTML

TS

```
<mat-chip-set aria-label="Fish selection">
  <mat-chip>One fish</mat-chip>
  <mat-chip>Two fish</mat-chip>
  <mat-chip>Three fish</mat-chip>
  <mat-chip disabled>Four fish</mat-chip>
</mat-chip-set>
```

One fish

Two fish

Three fish

Four fish

Posteriormente se copian el selector de la clase para pegarlo al html principal y también se copian los y se copia el nombre de la clase para pegarlo al TS principal.

```
import { FormFieldComponent } from './form-field/form-field.component';
import { GridListComponent } from './grid-list/grid-list.component';
import { IconComponent } from './icon/icon.component';
import { InputComponent } from './input/input.component';
import { ListComponent } from './list/list.component';
import { MenuComponent } from './menu/menu.component';
import { PaginatorComponent } from './paginator/paginator.component';
```

```
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet,
    Autocomplit2Component,
    AutocompleteComponent,
    BadgePositionComponent,
    ButtonComponent,
    SelectionComponent,
    ButtonTogglesComponent,
    BasicCardComponent,
    BasicCheckboxesComponent,
```

```
<app-badge-position></app-badge-position>
<app-button></app-button>
<app-selection></app-selection>
<app-button-toggles></app-button-toggles>
<app-basic-card></app-basic-card>
<app-basic-checkboxes></app-basic-checkboxes>
<app-basic-chip></app-basic-chip>
<app-daterickes></app-daterickes>
```

Para finalizar se visualiza el proyecto de manera grafica en la pantalla.

One fish

Two fish

Three fish

Four fish

## **Bibliografía**

<https://material.angular.io/components/chips/overview>