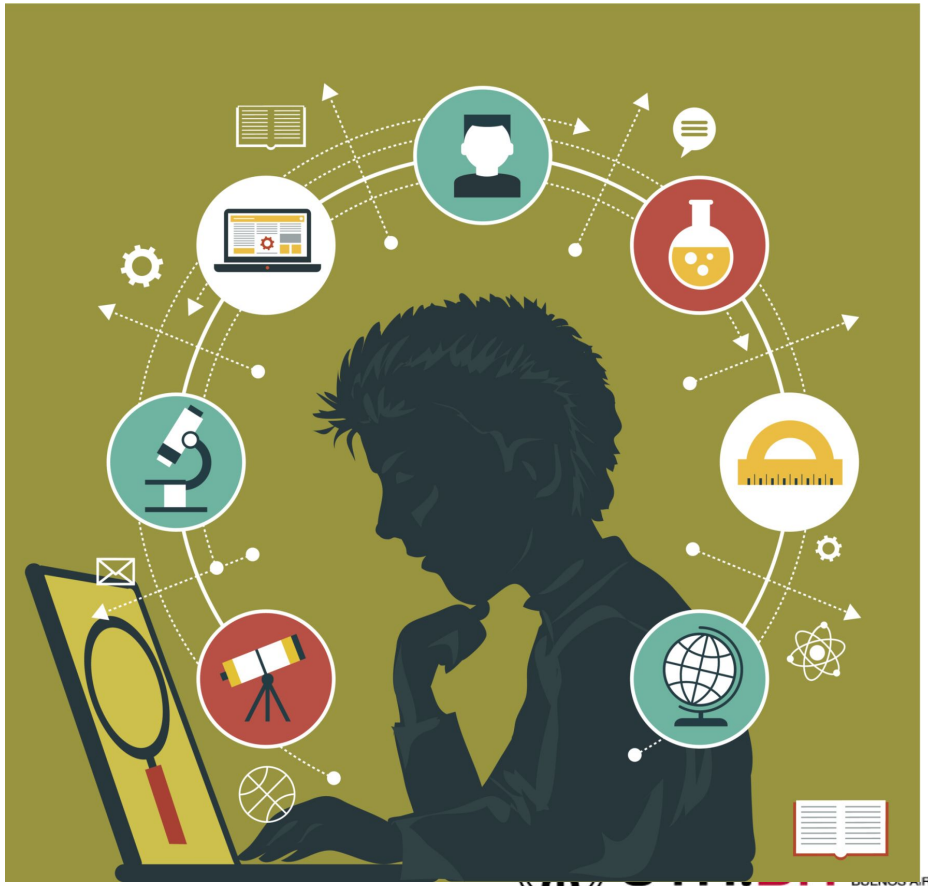
UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Preguntas?

Lo que vamos a ver hoy!

- Servicios
- Validación de formularios



Angular

Servicios

Servicios

Los servicios son una forma de compartir información entre clases que no se conocen.

Cuando estamos creando nuestros componentes muchas veces debemos compartir información entre ellos, este es el caso del uso de los Servicios.

En las buenas prácticas de Angular se recomienda liberar a los componentes de cualquier lógica no relacionada con la vista. Hay que mover toda esa lógica a un servicio.

Servicios

Para crear un servicio

```
$ ng g service nombre-servicio
```

Genera el esqueleto de la clase nombre-servicioService en **src/app/nombre-servicio.service.ts**.

Al consumo de los servicios inyectables se le conoce como DEPENDENCIA.

Si queremos que el servicio esté dentro de un módulo, agregamos la ruta:

```
$ ng generate service facturacion/clientes
```

Servicios

Cómo usar un Servicio?

Angular funciona a través de inyección de dependencias. Esto significa que se puede pasar una referencia a una instancia en diferentes componentes y permitirá utilizarla en diferentes partes de la App.

Servicios

Cómo disponer de un Servicio?

1. Agregar la declaración del servicio a un módulo

Para poder usar un servicio hay que agregarlo a un módulo. Así se puede usar en cualquier componente que pertenezca a ese módulo.

La declaración del servicio, a diferencia de los componentes, debe ser hecha manualmente.

```
@NgModule({
  imports: [
    CommonModule
  ],
  declarations: [ListadoClientesComponent],
  providers: [ClientesService]
})
```

Servicios

Cómo disponer de un servicio en un componente?

2. Hacer el **import** para que se conozca la clase service creada

```
import { ClientesService } from './clientes.service';
```

Servicios

Cómo disponer de un servicio en un componente?

3. Declaración en el componente, del servicio que vamos a usar

Se realiza en el constructor del componente donde vamos a usarlo, luego de hacer el import.

```
export class ListadoClientesComponent {
  constructor(public clientesService: ClientesService) { }
}
```

Servicios

Cómo disponer de un servicio en un componente?

3. Declaración del servicio que vamos a usar

```
export class ListadoClientesComponent {
  constructor(public clientesService: ClientesService) { }
}
```

=

```
export class ListadoClientesComponent {
  clientesService: ClientesService;
  constructor(clientesService: ClientesService) {
    this.clientesService = clientesService;
  }
}
```

Servicios

Cómo disponer de un servicio en un componente?

4. Usando el servicio dentro de la clase del componente

Accedemos como a cualquier otra propiedad, mediante la variable *this.*

```
export class ListadoClientesComponent implements OnInit {

  constructor(public clientesService: ClientesService) { }

  ngOnInit() {
    console.log(this.clientesService);
  }

}
```

Servicios

Cómo disponer de un servicio en un componente?

5. Usando el servicio en el template del componente

Como el servicio está en una propiedad del componente podemos acceder por el nombre

```
<p>
  URL De acceso: {{clientesService.accesoFacturacion}}
</p>
```

Validación de formularios

Validación de formularios

Vamos a trabajar con Formularios Reactivos

Para usarlos tenemos que declararlos en @NgModule

```
import { ReactiveFormsModule } from '@angular/forms';

@NgModule({
  imports: [
    ...,
    ReactiveFormsModule
  ],
  declarations: [...],
  bootstrap: [...]
})
export class AppModule {}
```

Validación de formularios

Definiciones

FormControl

Es un objeto que se usa en los formularios para tener el control sobre su valor y su estado en el formulario

```
const ctrl = new FormControl('some value');
console.log(ctrl.value);     // 'some value'
```

Validación de formularios

Definiciones

FormGroup

Es un conjunto de FormControls. El estado de FormGroup depende del estado de todos sus objetos, por lo tanto si uno de FormControls es inválido, el grupo entero será inválido.

```
const form = new FormGroup({
  first: new FormControl('Nancy', Validators.minLength(2)),
  last: new FormControl('Drew'),
});
```

Validación de formularios

Definiciones

Utilizando FormGroup y FormControl en el HTML

1. Crear en el componente el formGroup
2. Creamos dentro del FormGroup, los FormControls necesarios
3. En el HTML

```
<form novalidate [formGroup]="group">
  Name: <input type="text" formControlName="name">
  Location: <input type="text" formControlName="location">
</form>
```

Validación de formularios

Validadores

Para poder validar lo primero es importar en el componente los validadores

```
import { FormControl, FormGroup, Validators } from '@angular/forms';
```

Validación de formularios

Validadores

Para validar la información de los campos del formulario, podemos usar las funciones que nos ofrece Angular o implementar las nuestras propias:

```
ngOnInit() {
  this.user = new FormGroup({
    name: new FormControl('', [Validators.required, Validators.minLength(2)]),
      password: new FormControl('', Validators.required),
      passwordRepeat: new FormControl('', Validators.required)
  });
}
```

Validación de formularios

Validadores

También se pueden desactivar botones si el formulario no es válido

```
<form novalidate (ngSubmit)="onSubmit()" [formGroup]="user">
  ...
  <button type="submit" [disabled]="user.invalid">Sign up</button>
</form>
```

PRÁCTICA

PRÁCTICA ANGULAR

Ejercicio 1

Agregarle a la calculadora de la clase pasada servicios y validaciones.

En cuanto a los servicios, enviar a un servicio los cálculos.

En cuanto a validaciones: que los datos ingresados sean del tipo que deben ser. Desactivar el botón hasta que todas las validaciones sean correctas.

FIN!!!!