**EXCEPCIONES TYPESCRIPT**

Para crear una excepción en Typescript, se utiliza la palabra clave **throw**, seguida de una cadena que contenga un mensaje de error o una instancia al objeto **Error**.

errorsOnThree(input: number) {

if (input === 3) {

throw new Error('Three is not allowed');

}

return input;

}

Se recomienda crear una excepción utilizando una clase que implemente la interfaz de Error.

class ApplicationError implements Error {

public name = 'ApplicationError';

constructor(public message: string) {

}

toString() {

return this.name + ': ' + this.message;

}

}

errorsOnThreeClass(input: number) {

if (input === 3) {

throw new ApplicationError('Three is not allowed');

}

return input;

}

*Try-catch*

try {

var result = this.errorsOnThreeClass(3);

}catch (err) {

if (err instanceof ApplicationError) {

throw err;

}

console.log('Error caught, no action taken');

}

**SERVICIOS**

export class Service {

private url:String = "http://localhost:8081/";

constructor(private http:Http) { }

public getDato():Observable<Usuarios>{

return this.http.get(this.url+"datos")

.map(respuesta => respuesta.json())

.catch(this.handleErrorObservable);

}

private handleErrorObservable(error: Response | any){

  console.error(error.message || ‘Error en el servidor’);

  return Observable.throw(error.message || ‘Error en el servidor’);

}

}

**COMPONENTES**

export class ClienteComponent implements OnInit {

public respuesta;

constructor(private service : Service) { }

ngOnInit() {

this.getDato();

}

public getDato(){

this.service.getDato().subscribe(

data=>{this.respuesta = data;},

error => {console.log("ERROR al cargar los datos");}

);

}

}

<https://scotch.io/tutorials/angular-2-http-requests-with-observables>

https://github.com/dye784/collection/blob/master/public/Pro%20TypeScript.pdf