**Data Bindings**

Enlaces de datos, propiedades y funciones de los componentes

Permiten trabajar de manera interactiva con los datos, entre usuario y sistemas, muchos cambios se aprecian en tiempo real por el usuario, esto debido al digest loop de angular, en cuanto se detecta un cambio se interpreta y visualiza.

**One way** : de archivo de datos(ts) a html, de html a archivo de datos

typescript -> html

html -> typescript

Interpolación, es enviar data desde el typescript al html, lo hemos venidos haciendo, revisar ejemplo de *capítulo 3\_angularcli/componentes.* Se utiliza sintaxis *moustache*

{{ objeto }}

Cuando el componente tiene una clase ya hablamos de **modelo**

***Al crear nuestro proyecto , adentro de la carpeta app creamos la carpeta modelos, esta contendrá nuestras clases o modelos***

***App/modelos/sucursales.modelo.ts***

export class Sucursal {

public id: number;

public nombre: string;

public direccion: string;

public ciudad: string;

constructor (id: number, nombre: string, direccion: string, ciudad: string){

this.id = id;

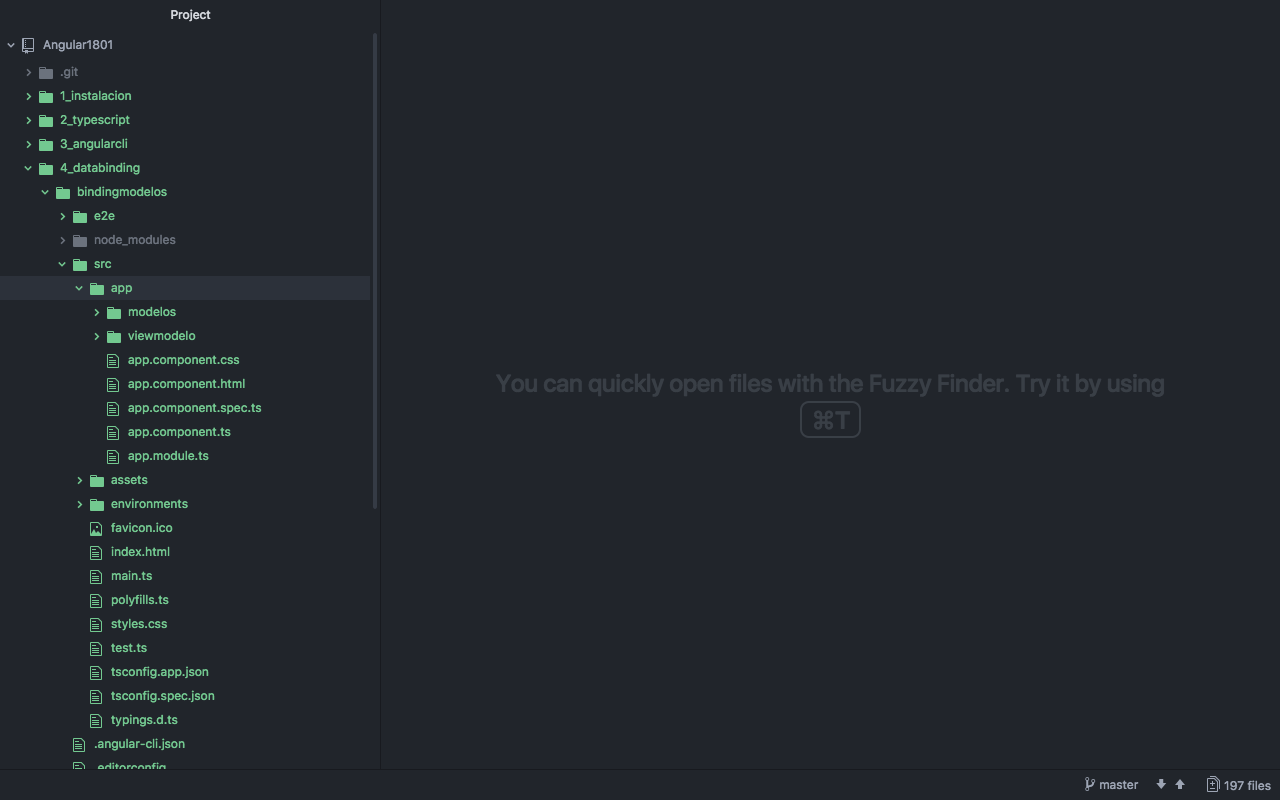
this.nombre = nombre;

this.direccion = direccion;

this.ciudad = ciudad;

} }

Estructura



Creamos un modelo

ng g c viewmodelo –spec false

En la carpeta viewmodelo/viewmodelo.component.html

<div class="container">

<h4>Información de Sucursal </h4>

<hr>

<h5>id: {{ sucursal1.id }}</h5>

<hr>

<h5>Nombre: {{ sucursal1.nombre }}</h5>

<hr>

<h5>Dirección: {{ sucursal1.direccion }}</h5> <hr>

<h5>Ciudad: {{ sucursal1.ciudad }}</h5>

<hr> </div>

viewmodelo.component.ts

import { Component, OnInit } from '@angular/core';

import { Sucursal } from '../modelos/sucursales.model';

@Component({

selector: 'app-viewmodelo',

templateUrl: './viewmodelo.component.html',

styleUrls: ['./viewmodelo.component.css']

})

export class ViewmodeloComponent implements OnInit {

sucursal1 = new Sucursal (1, 'Las Condes', 'Apoquindo 3452', 'Santiago');

constructor() { }

ngOnInit() {

}

}

En el archivo app.components.html

<!--The content below is only a placeholder and can be replaced.-->

<div style="text-align:center">

<h1>

{{ title }} Pizzas Tío Juan

</h1>

</div>

<app-viewmodelo></app-viewmodelo>

Archivo app.modules.ts

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

import { NgModule } from '@angular/core';

import { AppComponent } from './app.component';

import { ViewmodeloComponent } from './viewmodelo/viewmodelo.component';

@NgModule({

declarations: [

AppComponent,

ViewmodeloComponent

],

imports: [

BrowserModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

**Data binding event**

**ng new ejemploevento**

**ng g c enventos**

**app/eventos/eventos.component.ts**

**import { Component, OnInit } from '@angular/core';**

**@Component({**

**selector: 'app-enventos',**

**templateUrl: './enventos.component.html',**

**styleUrls: ['./enventos.component.css']**

**})**

**export class EnventosComponent implements OnInit {**

**texto = 'Originalmente el texto se carga así';**

**modTexto() {**

**this.texto = 'Al pulsar el botón el texto se muestra así';**

**}**

**constructor() { }**

**ngOnInit() {**

**}**

**}**

**enventos.components.html**

**<button class="btn btn-success" (click)="modTexto()">Modificar Texto</button>**

**<h3> {{ texto }} </h3>**

**app.components.html**

**he content below is only a placeholder and can be replaced.-->**

**<div style="text-align:center">**

**<h1>**

**Welcome to {{ title }}!**

**</h1>**

**<div class="container" >**

**<h3>Ejemplo de Event Binding</h3>**

**<hr> <app-enventos></app-enventos>**

**</div>**

**app.modules.ts**

**import { BrowserModule } from '@angular/platform-browser';**

**import { NgModule } from '@angular/core';**

**import { AppComponent } from './app.component';**

**import { EnventosComponent } from './enventos/enventos.component';**

**@NgModule({**

**declarations: [**

**AppComponent,**

**EnventosComponent**

**],**

**imports: [**

**BrowserModule**

**],**

**providers: [],**

**bootstrap: [AppComponent]**

**})**

**export class AppModule { }**

**Integración**