codenotch

Ionic FrameWork

Tema 1 | Ionic

codenotch

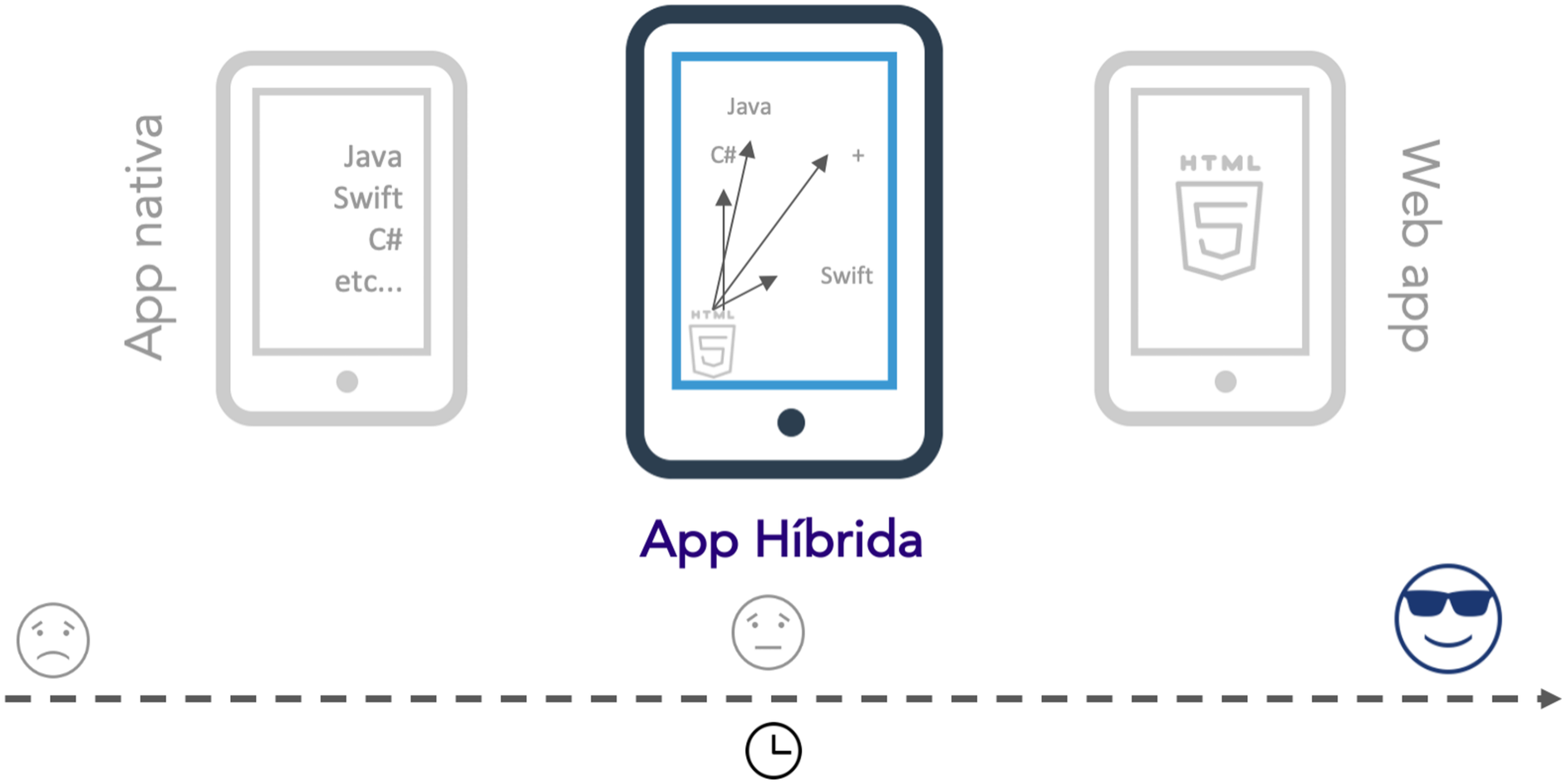
Índice

1. Aplicaciones Híbridas
2. Ionic
3. Creación y Estructura de páginas en Ionic
4. El Grid de Ionic
5. Librerias Nativas
6. Retos

codenotch

DEFINICIÓN

Aplicaciones Híbridas



codenotch

IONIC

Instalación de Ionic

Instalamos el cliente de ionic

- `npm i -g @ionic/cli`

Creamos una nueva aplicación a través del cliente

- ionic start NOMBRE_APLICACION TIPO_APLICACION

Ejemplo: `ionic start codenotch-ionic blank`

Tips

- Elegir Angular como Framework
- No, a instalar Capacitor
- No, a crear cuenta de Ionic

Referencias:
https://ionicframework.com/getting-started#cli

codenotch

IONIC

Configuración de Ionic

```
>_ ionic start MiProyecto-{INICIALES} blank
```

- `ionic` — Comandos de Ionic
- `start` — Crea un nuevo proyecto
- `MiProyecto-{INICIALES}` — Nombre del proyecto
- `blank` — Nombre del template a utilizar

Referencias:

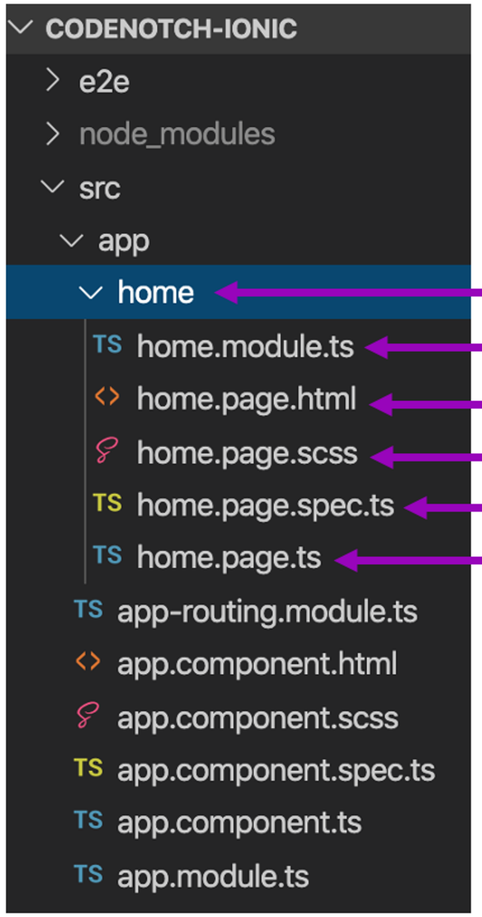
- https://ionicframework.com/docs/intro/installation/

codenotch

IONIC

Estructura de un Proyecto

Cada página en Ionic es un modulo de Angular



Directorio de la página principal de la aplicación (módulo)

Fichero del modulo

Vista

Diseño

Test

Controlador

codenotch

IONIC

Generar una Página en Ionic

```
>_ ionic generate page home
```

ionic	generate	page	home
Comandos de Ionic	Genera algo nuevo	Genera una nueva page	Nombre

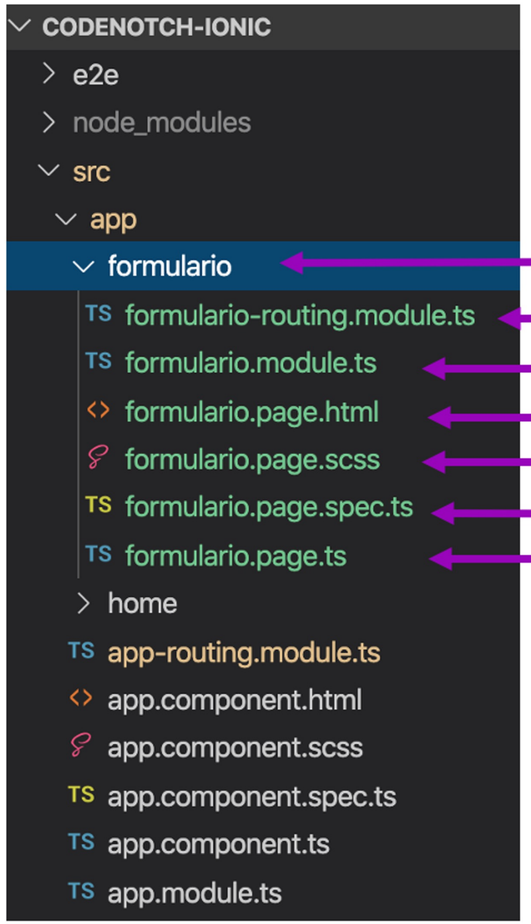
Referencias:

- https://ionicframework.com/docs/cli/generate/

codenotch

IONIC

Estructura de una Página



La nueva pagina en Ionic es otro modulo de Angular

Directorio de la página principal de la aplicación (módulo)

Fichero del routing del módulo

Fichero del modulo

Vista

Diseño

Test

Controlador

codenotch

IONIC

Generar otros Elementos Ionic

```
>_ ionic generate class models/name
```

- `ionic`: Comando de Ionic
- `generate`: Genera algo nuevo
- `class`: Genera una nueva clase
- `models/name`: Nombre

```
>_ ionic generate service shared/name
```

- `service`: Genera un nuevo servicio
- `shared/name`: Nombre

```
>_ ionic generate component name
```

- `component`: Genera un nuevo componente
- `name`: Nombre

Referencias:

- https://ionicframework.com/docs/cli

IONIC

Ejecución y Despliegue

Ejecución

```
>_    ionic serve
```

Despliegue

```
>_    ionic cordova platform add android
                                 ios
```

Referencias:

- https://ionicframework.com/docs/intro/installation/

codenotch

IONIC

Ionic Template

```
<ion-header>
```

Es la cabecera, se escribirán cosas como el titulo, iconos y navegación

```
</ion-header>
```

```
<ion-content>
```

Aquí escribiremos la estructura de nuestra página

```
</ion-content>
```

El Grid

	1	2	3	4	5	6	7	8	9	10	11	12	
<ion-row>	ion-col	ion-col	ion-col	ion-col	ion-col	ion-col	ion-col	ion-col	ion-col	ion-col	ion-col	ion-col	</ion-row>
<ion-row>	ion-col						ion-col						</ion-row>
<ion-row>	ion-col col-2												</ion-row>

codenotch

IONIC

Etiquetas Importantes

- `<ion-checkbox>` Elemento clicable que permite seleccionar opciones con “verdadero” o “falso”
- `<ion-icon>` Caja para mostrar iconos
- `<ion-input>` Caja de texto para obtener información del usuario
- `<ion-label>` Etiqueta de texto que usualmente acompaña a otros elementos
- `<ion-button>` Un botón clicable que desencadena acciones
- `<ion-item>` Una agrupación de elementos que se comportan como uno solo. Pueden estar dentro de listas, cols, etc

Referencias:
https://ionicframework.com/docs/components/

codenotch

IONIC

Etiquetas HTML5

- `<div>` Un contenedor que funciona como un bloque, el resto de la página se organiza por arriba o por debajo de este elemento .
- `<span>` Un contenedor entre líneas
- `<hr>` Dibuja una línea horizontal
- `<br>` Salto de línea
- `<p>` Lo que esté dentro se considera un párrafo
- `<strong>` Marca un texto como negrita

Referencias:
https://hostingfacts.com/html-cheat-sheet/

codenotch

Librerías Nativas

codenotch

DEFINICIÓN

Librerías Nativas en Ionic

Las librerías nativas de ionic te permiten acceder a los componentes nativos de tu dispositivo móvil. (Cámara, Localización, Almacenamiento, Etc).

Cada librería tiene sus instrucciones propias

Catálogo de librerías en Ionic:
- https://ionicframework.com/docs/native

Almacenamiento:
- https://ionicframework.com/docs/native/native-storage

Cámara:
- https://ionicframework.com/docs/native/camera

- Localización:
- https://ionicframework.com/docs/native/geolocation

codenotch

LIBRERÍA DE GEOLOCALIZACIÓN

Paso 1 : Instalar la librería

- `ionic cordova plugin add cordova-plugin-geolocation`

- `npm install @ionic-native/geolocation`

Tip: Aceptar las respuestas por defecto (letra mayúscula) en todas las preguntas

codenotch

LIBRERÍA DE GEOLOCALIZACIÓN

Paso 2 : Añadir módulo Geolocation

```
import { RouteReuseStrategy } from '@angular/router';

import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';

import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
import { Geolocation } from '@ionic-native/geolocation/ngx';

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy },
    Geolocation
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}
```

app.module.ts

codenotch

LIBRERÍA DE GEOLOCALIZACIÓN

Paso 3 : Invocar a la librería en el componente

```
import { Component } from '@angular/core';
import { Geolocation } from '@ionic-native/geolocation/ngx';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage
{
  public latitudPromesa;
  public longitudPromesa;
  public latitudObservable;
  public longitudObservable;

  constructor(private geolocation: Geolocation)
  {
    this.latitudPromesa = 0;
    this.longitudPromesa = 0;
    this.latitudObservable = 0;
    this.longitudObservable = 0;
  }
```

home.page.ts

codenotch

LIBRERÍA DE GEOLOCALIZACIÓN

Paso 3 (Cont)

```
obtenerLocalizacionPromesa():void
{
  this.geolocation.getCurrentPosition().then((resp) =>
  {
    this.latitudPromesa = resp.coords.latitude
    this.longitudPromesa =  resp.coords.longitude
   }).catch((error) =>
   {
     console.log('Error getting location', error);
   });
}

obtenerLocalizacionObservable():void
{
  let watch = this.geolocation.watchPosition();
  watch.subscribe((data:Geoposition) =>
  {
    this.latitudObservable = data.coords.latitude
    this.longitudObservable = data.coords.longitude
  });
}
}
```

home.page.ts

codenotch

Glosario

APLICACIONES HÍBRIDAS

IONIC

GRID DE IONIC

LIBRERÍAS NATIVAS

codenotch

#Retos

codenotch

RETOS

Reto 1

- Crear un nuevo proyecto en Ionic.

- Crear dos nuevas paginas: **formulario** y **vista**:
  - La página **formulario** va a tener un formulario simple con los cinco campos del **libro** sin incluir el **id_usuario** y los tres botones **añadir**, **modificar** y **borrar**. Más un botón que volverá a la pagina principal.
  - La página **vista** tendrá un único campo para indicar el **id** del **libro** a mostrar, si no se especifica el **id** se mostrarán todos los libros. Tendrá dos botones, mostrar y volver a la página principal.

- La **página principal** contendrá dos botones (**formulario** que irá a la pagina de formulario y **vista** que irá a la pagina de vista)

- Crear un modelo **libro**.

- Crear un **servicio libro** que se conecte con vuestra API REST.

codenotch

RETOS

Reto 2

- Crear un nuevo proyecto en **Ionic**.
- Añadirle la **librería** de **Geolocalización** tanto para las promesas como para los observables
- Añadirle **dos botones** a la pagina **home** uno que muestre la geolocalización con **promesas** y otro que muestre la geolocalización con **observables**.

codenotch