



**TECNOLOGICO
DE ESTUDIOS
SUPERIORES DE
ECATEPEC**

**División de ingeniería en
sistemas computacionales**

Integrantes:

Albor Ramos Jovani

Rojas Jiménez Alejandro

Sánchez Martínez Diego Alan

BD para dispositivos móviles

Griselda Cortes Barrera

Proyecto final(Manual Frontend)

Manual Frontend pasos a seguir

Antes de comenzar debemos instalar los siguientes comandos para que todo funcione correctamente:

```
npm install @angular/common @angular/router rxjs
```

```
npm install @ionic/angular
```

```
npm install --save-dev @angular-devkit/build-angular
```

```
npm install @angular/forms
```

```
npm install @angular/http
```

```
npm install --save @angular/core
```

```
npm install --save @ionic/storage-angular
```

ya una vez hecho esto comenzamos.

Primero creamos un proyecto de ionic como ya habíamos hecho anteriormente mediante el comando **ionic start** ,ya que este todo listo procederemos a importar los módulos a utilizar en el archivo **app.module.ts**

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
import { HttpClientModule } from '@angular/common/http';
import { FormsModule } from '@angular/forms';
import { Router } from '@angular/router';

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

Lo siguiente será crear un servicio llamado eventos que al crearlo nos mostrará dos archivos ts donde el único que usaremos será el de **eventos.service.ts** en este archivo haremos los imports necesarios de lo que instalamos anteriormente para después mediante las rutas mencionadas en el backend que nos muestran los datos en formato JSON almacenaremos en una nueva variable del front esa información, ya una vez hecho esto crearemos métodos que permitan el rescate de información de esas nuevas variables que creamos y poder hacer uso de esta en cualquier módulo de nuestro proyecto.

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root',
})
export class eventosService {
  private apiUrlConferencias = 'http://localhost:3000/conferencias';
  private apiUrlTalleres = 'http://localhost:3000/talleres';
  private apiUrlConcursos = 'http://localhost:3000/concursos';

  constructor(private http: HttpClient) {}

  // Obtener conferencias
  Tabnine | Edit | Test | Explain | Document | Ask
  getConferencias(): Observable<any> {
    return this.http.get<any[]>(this.apiUrlConferencias);
  }

  // Obtener eventos
  Tabnine | Edit | Test | Explain | Document | Ask
  getTalleres(): Observable<any> {
    return this.http.get<any[]>(this.apiUrlTalleres);
  }

  // Obtener ponentes
  Tabnine | Edit | Test | Explain | Document | Ask
  getConcursos(): Observable<any> {
    return this.http.get<any[]>(this.apiUrlConcursos);
  }
}
```

Pasaremos a el archivo **home.page.ts** donde haremos sus respectivos imports para después colocar las variables que almacenaran los arreglos y que haremos uso de ellas después lo siguiente dentro de este archivo es obtener los datos de los métodos creados en el servicio para almacenarlos como habíamos dicho en las nuevas variables, en este caso como haremos uso de una barra de búsqueda colocamos nuevas variables donde iría esa misma información pero que se haría uso de ella en otro método el cual será el filtrado de búsqueda ¿pero cómo funciona este filtrado? Bueno este método funciona tomando en cuenta lo que el usuario escribe en la barra de búsqueda y para poder hacer el filtrado se necesitan condiciones que permitan identificar lo que se busca en nuestro caso que la palabra se encuentre en un título, en la descripción o mediante un id, de esta manera si se cumpla aunque sea una sola condición mostrara las tarjetas este proceso se repitió dos veces mas una por cada tabla que tenemos y para finalizar haremos uso del método que nos permitirá dirigirnos a ver los detalles del evento a otra ruta que explicaremos más adelante.

```
import { Component } from '@angular/core';
import { eventosService } from '../eventos.service';
import { HttpClient } from '@angular/common/http';
import { AlertController } from '@ionic/angular';
import { Router } from '@angular/router';
@Component({
  selector: 'app-home',
  templateUrl: './home.page.html',
  styleUrls: ['./home.page.scss'],
})
export class HomePage {
  conferencias: any[] = [];
  talleres: any[] = [];
  concursos: any[] = [];
  filteredConferencias: any[] = [];
  filteredTalleres: any[] = [];
  filteredConcursos: any[] = [];
  searchQuery: string = '';

  constructor(
    private eventosService: eventosService,
    private http: HttpClient,
    private alertController: AlertController,
    private router: Router
  ) {}
```

```

ngOnInit() {
  // Obtener conferencias
  this.eventosService.getConferencias().subscribe(
    (data) => {
      console.log('Conferencias:', data); // Verifica los datos
      this.conferencias = data;
      this.filteredConferencias = data;
    },
    (error) => {
      console.error('Error al obtener las conferencias', error);
    }
  );

  // Obtener talleres
  this.eventosService.getTalleres().subscribe(
    (data) => {
      this.talleres = data;
      this.filteredTalleres = data;
    },
    (error) => {
      console.error('Error al obtener los talleres', error);
    }
  );
}

```

```

// Obtener concursos
this.eventosService.getConcursos().subscribe(
  (data) => {
    this.concursos = data;
    this.filteredConcursos = data;
  },
  (error) => {
    console.error('Error al obtener los concursos', error);
  }
);
}

// Función para manejar el filtrado
Tabnine | Edit | Test | Explain | Document | Ask
onSearch() {
  console.log('Buscando con el término:', this.searchQuery);
  const query = this.searchQuery.toLowerCase();

  // Filtrar las conferencias
  this.filteredConferencias = this.conferencias.filter(conferencia =>
    conferencia && conferencia.titulo && conferencia.titulo.toLowerCase().includes(query) ||
    conferencia && conferencia.descripcion && conferencia.descripcion.toLowerCase().includes(query) ||
    (conferencia && conferencia.idcon && conferencia.idcon.toString().includes(query)) // Cambiado a idcon
  );
}

```

```

// Filtrar los talleres
this.filteredTalleres = this.talleres.filter(taller =>
  taller && taller.titulo && taller.titulo.toLowerCase().includes(query) ||
  taller && taller.descripcion && taller.descripcion.toLowerCase().includes(query) ||
  (taller && taller.idtar && taller.idtar.toString().includes(query)) // Cambiado a idtar
);

// Filtrar los concursos
this.filteredConcursos = this.concursos.filter(concurso =>
  concurso && concurso.titulo && concurso.titulo.toLowerCase().includes(query) ||
  concurso && concurso.descripcion && concurso.descripcion.toLowerCase().includes(query) ||
  (concurso && concurso.idconc && concurso.idconc.toString().includes(query)) // Cambiado a idconc
);
}
Tabnine | Edit | Test | Explain | Document | Ask
goToEventDetails(event: any) {
  this.router.navigate(['/event-details'], {
    state: { event: event } // Pasando el objeto del evento completo
  });
}
}

```

Ahora pasaremos al archivo **home.page.html** donde realizaremos la interpolación usando las variables creadas anteriormente en el ts colocando esta información en ion-cards que muestren la imagen, título, fecha y descripción de cada evento y que estas mismas tarjetas sirvan como un botón que permita ver los detalles de cada evento más extensamente se hizo uso de una ion-card para cada tipo de evento.

```

<ion-header>
  <ion-toolbar>
    <ion-title>Eventos disponibles</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <!-- Barra de búsqueda -->
  <ion-searchbar [(ngModel)]="searchQuery" (ionInput)="onSearch()" debounce="500" placeholder="Buscar por título o palabra clave"></ion-searchbar>

  <!-- Mostrar Conferencias -->
  <ion-card *ngFor="let conferencia of filteredConferencias" (click)="goToEventDetails(conferencia)">
    <img [src]="conferencia.imgcon">
    <ion-card-header>
      <ion-card-title>{{ conferencia.titulo }}</ion-card-title>
      <ion-card-subtitle>{{ conferencia.fecha }}</ion-card-subtitle>
    </ion-card-header>
    <ion-card-content>
      <p>{{ conferencia.descripcion }}</p>
    </ion-card-content>
  </ion-card>

  <!-- Mostrar Talleres -->
  <ion-card *ngFor="let taller of filteredTalleres" (click)="goToEventDetails(taller)">
    <img [src]="taller.imgtar">
    <ion-card-header>
      <ion-card-title>{{ taller.titulo }}</ion-card-title>
      <ion-card-subtitle>{{ taller.fecha }}</ion-card-subtitle>
    </ion-card-header>
    <ion-card-content>
      <p>{{ taller.descripcion }}</p>
    </ion-card-content>
  </ion-card>

```

```

<!-- Mostrar Concursos -->
<ion-card *ngFor="let concurso of filteredConcursos" (click)="goToEventDetails(concurso)">
  <img [src]="concurso.imgconc">
  <ion-card-header>
    <ion-card-title>{{ concurso.titulo }}</ion-card-title>
    <ion-card-subtitle>{{ concurso.fecha }}</ion-card-subtitle>
  </ion-card-header>
  <ion-card-content>
    <p>{{ concurso.descripcion }}</p>
  </ion-card-content>
</ion-card>
</ion-content>

```

Para mostrar un mejor diseño en el archivo **home.page.scss** colocamos los siguientes parámetros:

```

ion-header{
  --background: white !important; /* Cambia el color de fondo del header */
  text-align: center; /* Centra el texto en el header */
  width: auto;
  background-color: white !important; /* Cambia el color de fondo del header */
}

#container {
  text-align: center;

  position: absolute;
  left: 0;
  right: 0;
  top: 50%;
  transform: translateY(-50%);
}

#container strong {
  font-size: 20px;
  line-height: 26px;
}

#container p {
  font-size: 16px;
  line-height: 22px;

  color: #8c8c8c;

  margin: 0;
}

#container a {
  text-decoration: none;
}

.inicio{
  background-color: white;
  background: white;
}

```

```

ion-content {
  --background: #f0f0f0; /* Cambia el color de fondo */
}

ion-card-img {
  width: 10%; /* Asegúrate de que la imagen ocupe el espacio disponible */
  height: 10%;
}

ion-card{
  margin: 10px; /* Ajusta el espacio entre las tarjetas */
  background-color: white; /* Cambia el color de fondo de las tarjetas */
  border-radius: 10px; /* Ajusta el radio de curvatura de las tarjetas */
  width:350px;
  height: 350px;
  font-family: Arial, Helvetica, sans-serif;
}

ion-searchbar{
  margin: 10px; /* Ajusta el espacio entre el header y el searchbar */
  width:auto;
  --background:white;
  color: black;
}

```

Con esto concluiríamos la parte de la página principal, así como de la búsqueda lo siguiente serán los detalles de cada evento de manera mas extensa haremos esto a través de la creación de una página llamada **event-details** creada a través del comando

ionic generate page evento-details al crear esta nueva página nos dará 6 archivos de los cuales solo usaremos 3 que serán **event-details.page.ts**, **event-details.page.html** y **event-details.page.scss**.

antes de ir a esos archivos iremos al archivo **app-routing.module.ts** donde colocaremos la ruta para acceder a esta nueva página:


```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
const routes: Routes = [
  {
    path: 'home',
    loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
  },
  {
    path: '',
    redirectTo: 'home',
    pathMatch: 'full'
  },
  {
    path: 'event-details',
    loadChildren: () => import('./event-details/event-details.module').then( m => m.EventDetailsPageModule)
  },
];

@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Ya una vez hecho esto pasaremos al archivo ts de la pagina donde colocaremos una variable que almacene la información obtenida a través de la navegación ya que haremos uso de ella después.

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

@Component({
  selector: 'app-event-details',
  templateUrl: './event-details.page.html',
  styleUrls: ['./event-details.page.scss'],
})
export class EventDetailsPage {
  event: any;

  constructor(private router: ActivatedRoute) {}

  ngOnInit() {
    // Obtener el evento pasado a través de la navegación
    this.event = history.state.event;
  }
}
```

Ahora pasamos al archivo html donde haremos uso de esa variable para la interpolación y la creación de la ion-card

```

<ion-header>
  <ion-toolbar>
    <ion-title>Detalles sobre el evento</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <ion-card>
    <img [src]="event.imgcon || event.imgtar || event.imgconc">
    <ion-card-header>
      <ion-card-title>{{ event.titulo }}</ion-card-title>
      <ion-card-subtitle>{{ event.fecha }}</ion-card-subtitle>
    </ion-card-header>
    <ion-card-content>
      <p>{{ event.descripcion }}</p>
    </ion-card-content>
  </ion-card>
</ion-content>

```

Y por ultimo el diseño que colocamos en el archivo scss.

```

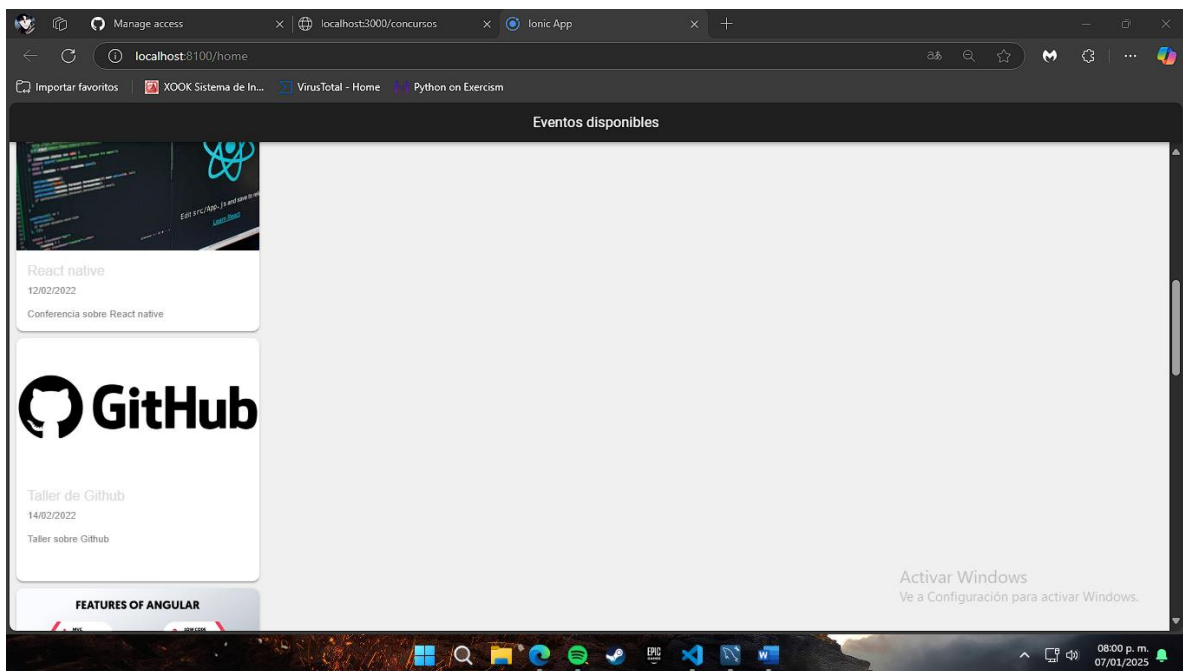
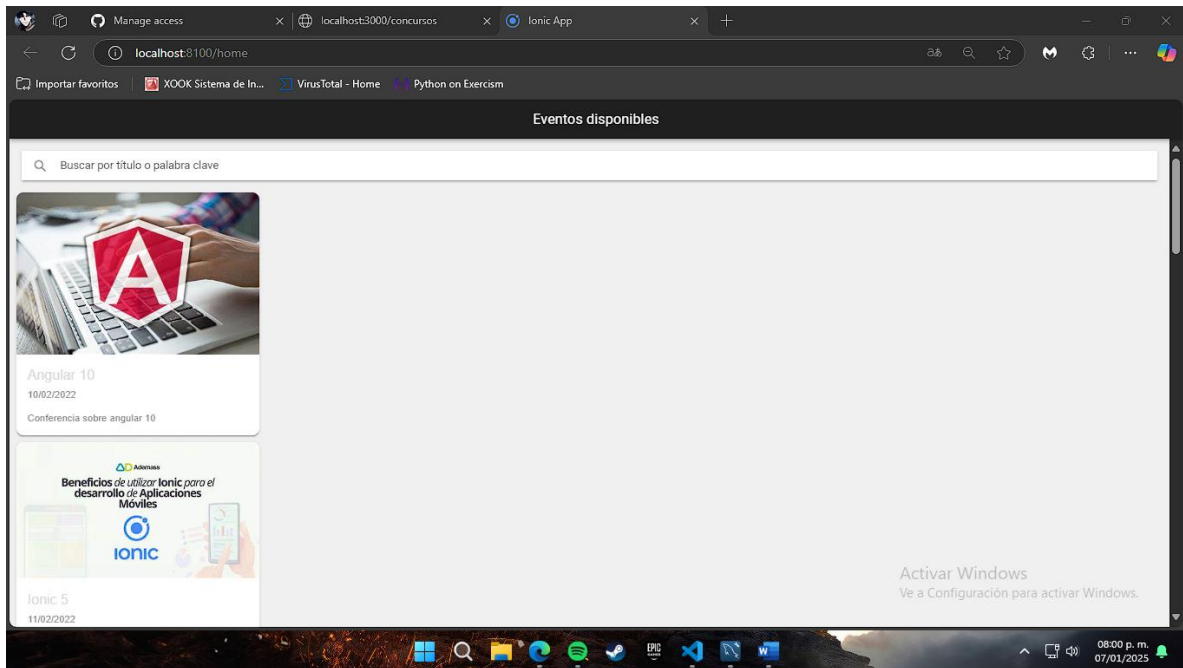
ion-card-img {
  width: 10%; /* Asegúrate de que la imagen ocupe el espacio disponible */
  height: 10%;
}
ion-card{
  margin: 10px; /* Ajusta el espacio entre las tarjetas */
  background-color: white; /* Cambia el color de fondo de las tarjetas */
  border-radius: 10px; /* Ajusta el radio de curvatura de las tarjetas */
  width: 350px;
  height: 350px;
  font-family: Arial, Helvetica, sans-serif;
}

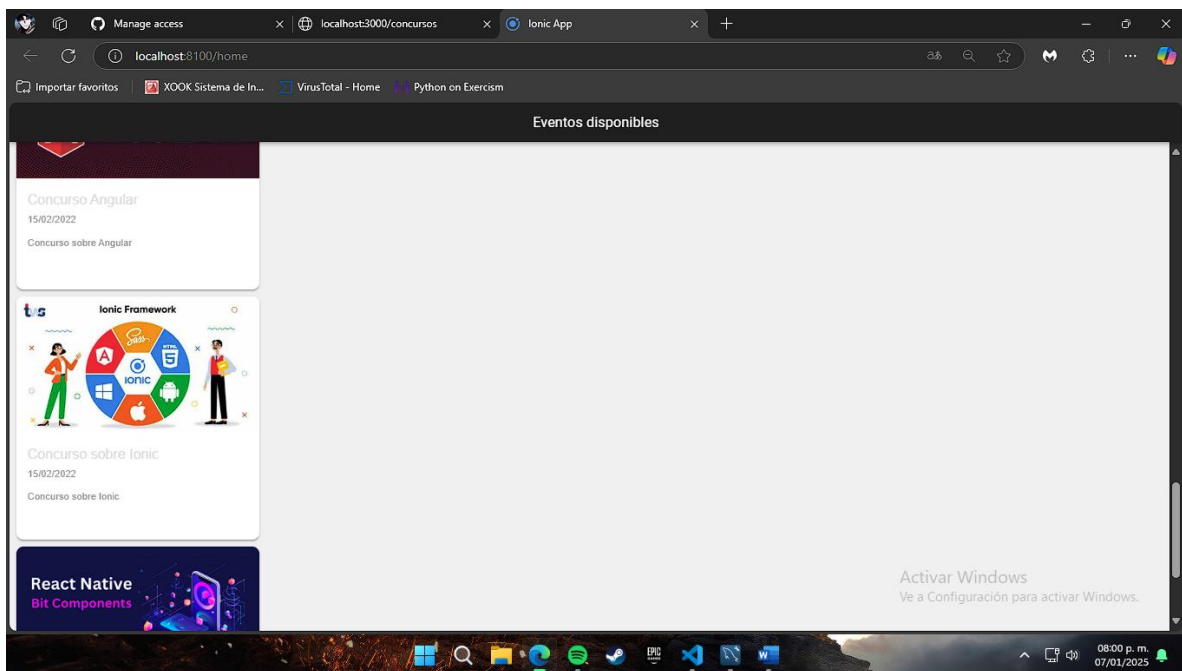
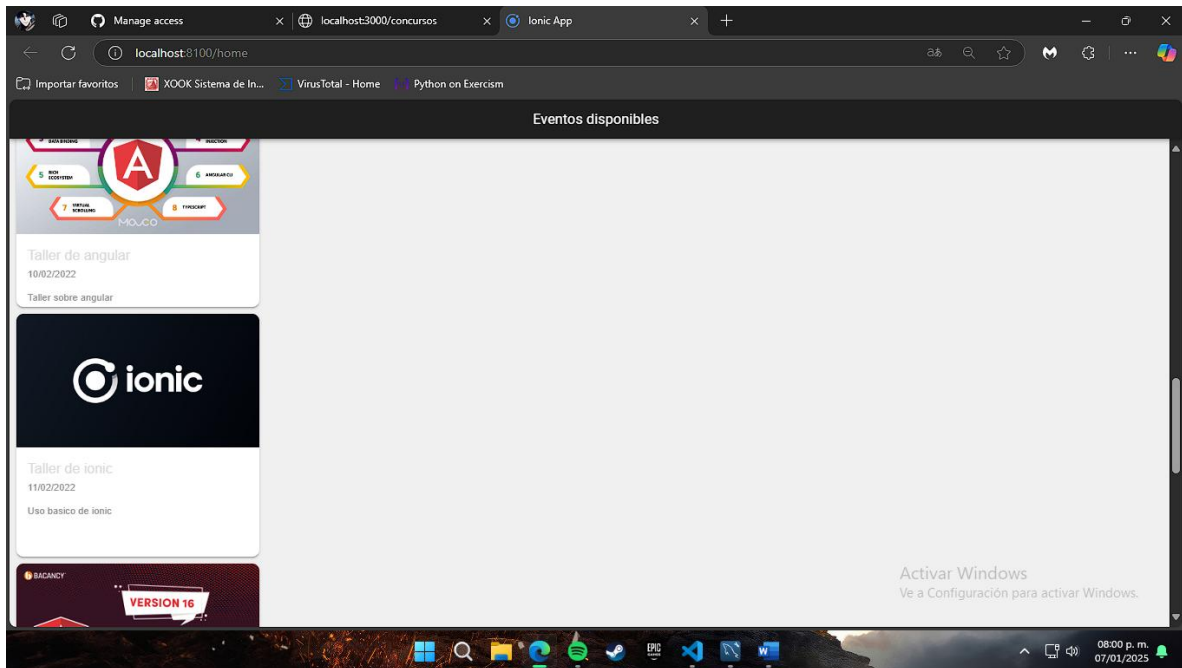
ion-content {
  --background: #f0f0f0; /* Cambia el color de fondo */
}
ion-header{
  --background: white !important; /* Cambia el color de fondo del header */
  text-align: center; /* Centra el texto en el header */
  width: auto;
  background-color: white !important; /* Cambia el color de fondo del header */
}

```

Y con esto finalizaríamos el frontend de nuestro proyecto a continuación se mostrará el resultado final de como se ve el frontend para esto usaremos el comando ionic serve.

Página principal





Uso de la barra de búsqueda

Angular



Angular 10

10/02/2022

Conferencia sobre angular 10



Taller de angular

10/02/2022

Taller sobre angular

BACANCY

VERSION 10

ionic



Ionic 5

11/02/2022

Conferencia sobre ionic 5



Taller de ionic

11/02/2022

Uso basico de ionic

ts

Ionic Framework

react



React native

12/02/2022

Conferencia sobre React native



React Native
Bit Components

Concurso de React native

15/02/2022

Concurso sobre React native

Detalles eventos

localhost:3000/concursos

localhost:8100/event-details


Importar favoritos

XOOK Sistema de In...

VirusTotal - Home

Python on Exercism

Detalles sobre el evento



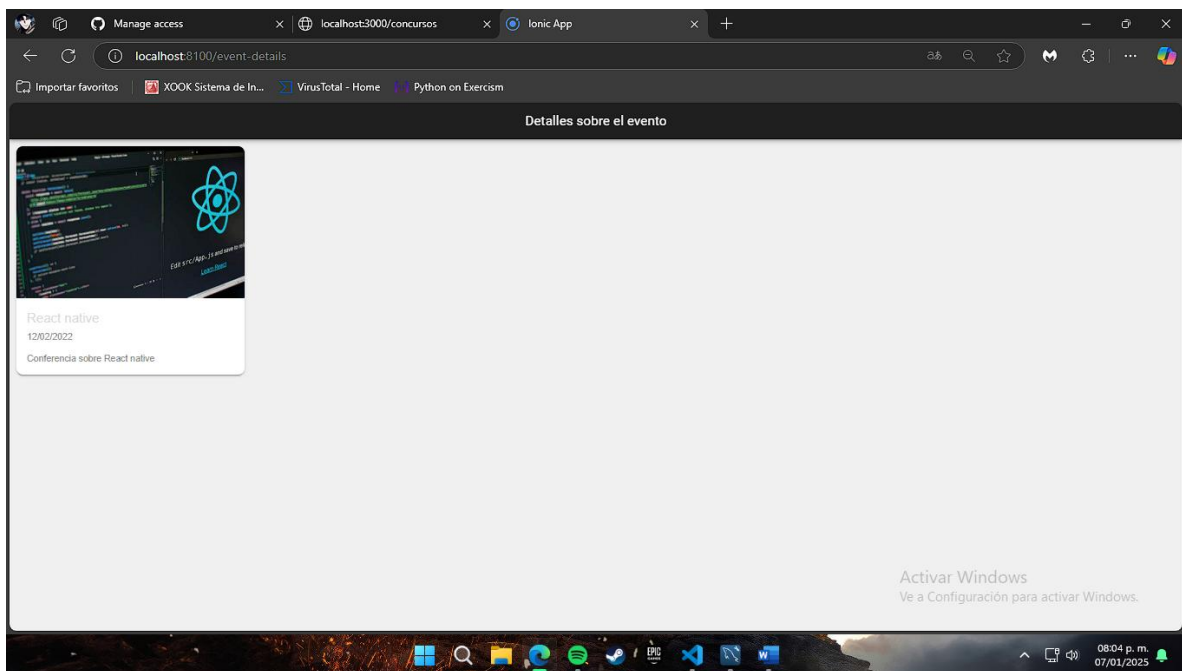
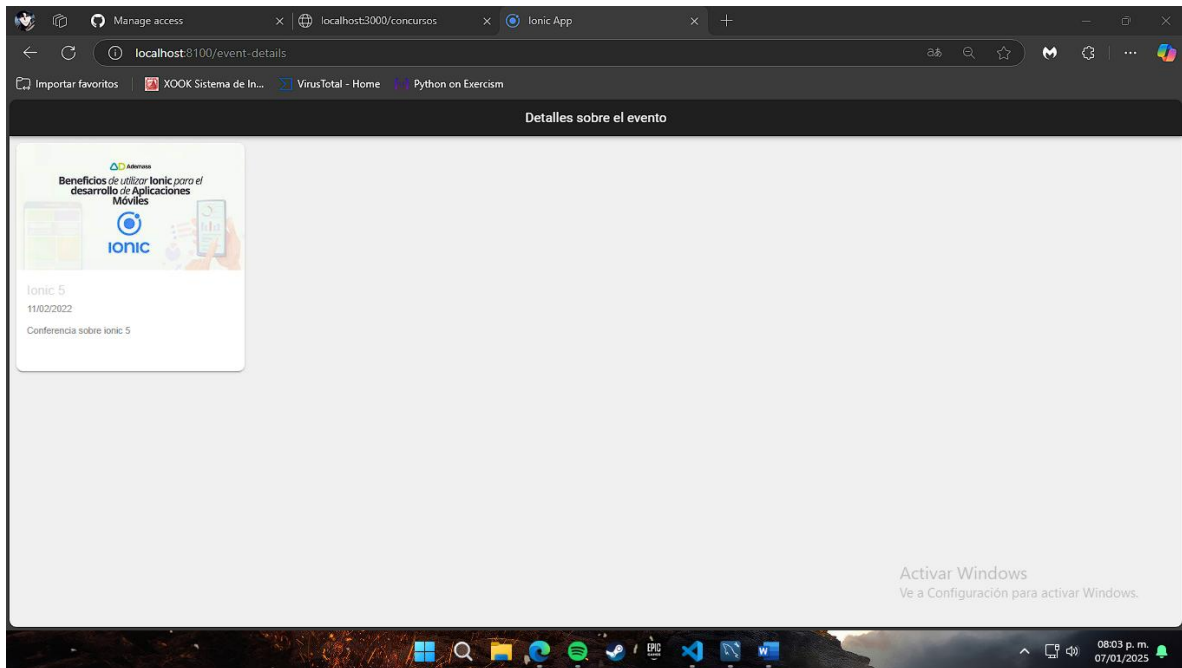
Angular 10

10/02/2022

Conferencia sobre angular 10

Activar Windows
Ve a Configuración para activar Windows.

08:03 p. m.
07/01/2025



Y con esto finalizaría el frontend y todo el proyecto.