

ACTIVIDAD 2

INTRODUCCIÓN IONIC

EJERCICIO 1: GAMEHUB

Vamos a crear una pequeña aplicación de móvil en la que trabajaremos estos conceptos:

- Páginas generadas con ionic generate page
- Rutas lazy con loadComponent
- Subrutas con children + ion-router-outlet
- Navegación usando:
 - routerLink
 - Router.navigate()
 - NavController.navigateForward() y pop() / navigateBack()
- Paso de:
 - Parámetro :id
 - Objeto completo por state
- TabBar
- Menú lateral
- Modal
- AlertController + alerta inline

PÁGINAS DEL PROYECTO:

- Vamos a crear estas páginas:
- dashboard (contenedora)
- dashboard /juegos (listado principal)
- dashboard /tienda (categorías)
- dashboard /perfil (padre)
- juego-detalle/:id
- Subpáginas Perfil:
 - dashboard /perfil/configuracion
 - dashboard /perfil/estadisticas

TABBAR PRINCIPAL:

Crear 3 pestañas:

- Juegos
- Tienda
- Perfil

MENÚ LATERAL:

Crear un menú lateral con los siguientes enlaces:

- Juegos
- Tienda
- Perfil
- “Acerca de GameHub” → abre MODAL

PÁGINA "JUEGOS" (LISTADO)

En juegos.page.ts crea un array con un mínimo de 5 juegos con esta estructura de juego:

```
{  
  id: number,  
  titulo: string,  
  plataforma: 'PC' | 'PS5' | 'Xbox' | 'Switch',  
  genero: string,  
  precio: number,  
  popular: boolean  
}
```

1. En la vista:

Mostrar cada juego con ion-card o ion-item.

Cada juego debe tener:

- Botón **Ver detalle**→ Navega a /juego-detalle/:id
- Botón **Vista rápida**→ Abre un **modal** pasando el objeto completo

PÁGINA "JUEGO DETALLE"

En el archivo ts:

1. Leer parámetro id con ActivatedRoute
2. Buscar el juego en el array
3. Mostrar:
 - Título
 - Plataforma
 - Género
 - Precio
 - Estado popular

Añadir en esta página un botón “Volver”

PÁGINA "TIENDA" (ROUTER + STATE)

Mostrar botones por plataforma:

- PC
- PS5
- Xbox
- Switch

Al pulsar sobre cada botón, navegar a dashboard/juegos pasando State: {plataforma: 'plataforma'}, siendo 'plataforma' PC, PS5, Xbox, Switch.

En juegos si llega state:plataforma: filtrar los juegos de esa plataforma automáticamente.

PERFIL CON SUBRUTAS

Perfil debe contener:

- <ion-router-outlet>
- Botones
 - Configuración
 - Estadísticas
- Por defecto debe redirigir a /dashboard/perfil/configuración

NAVCONTROLLER

En detalle implementar:

Botón “Ir tienda” usando navCtrl.navigateForward('/dashboard/tienda');

En configuración implementar:

Botón “Volver a perfil” usando navCtrl.pop();

VISTA RÁPIDA MODAL

Generar una componente game-preview dentro de una carpeta modals que debe:

Debe:

- Recibir objeto juego por Input
- Mostrar:
 - Título
 - Plataforma
 - Género
 - Precio
- Botón cerrar modal

ALERTAS

En configuración crear dos botones:

- “Activar notificaciones” que use AlertController y muestre una alerta con estos datos:
 - Header: Preferencias
 - Message: Notificaciones de lanzamientos activadas
 - Botón: OK
- “Normas de uso” que use una alerta inline con los siguientes datos:
 - Header: GameHub
 - SubHeader: Aviso
 - Message: No compartas tu cuenta con terceros.
 - Botón: Entendido

CONFIGURACIÓN

En la subpágina configuración implementar un **formulario reactivo** para editar los datos del perfil del usuario.

- Crear un FormGroup con los siguientes campos:

Campo	Tipo	Validaciones
nombre	text	required, minlength(3)
email	email	required, email
edad	number	min(13), max(99)
notificaciones	checkbox	(sin validación)

- Vista (HTML):

En la página configuración mostrar:

- ion-input para Nombre, Email y Edad
- ion-toggle para Notificaciones
- Botón Guardar perfil

El botón debe estar deshabilitado si el formulario es inválido.

Al pulsar guardar mostrar en un alert los datos del objeto completo.

Mostrar mensaje de error en los campos nombre, email y edad.