

Desarrollo Móvil con Xamarin

3 – Navegación y mapas

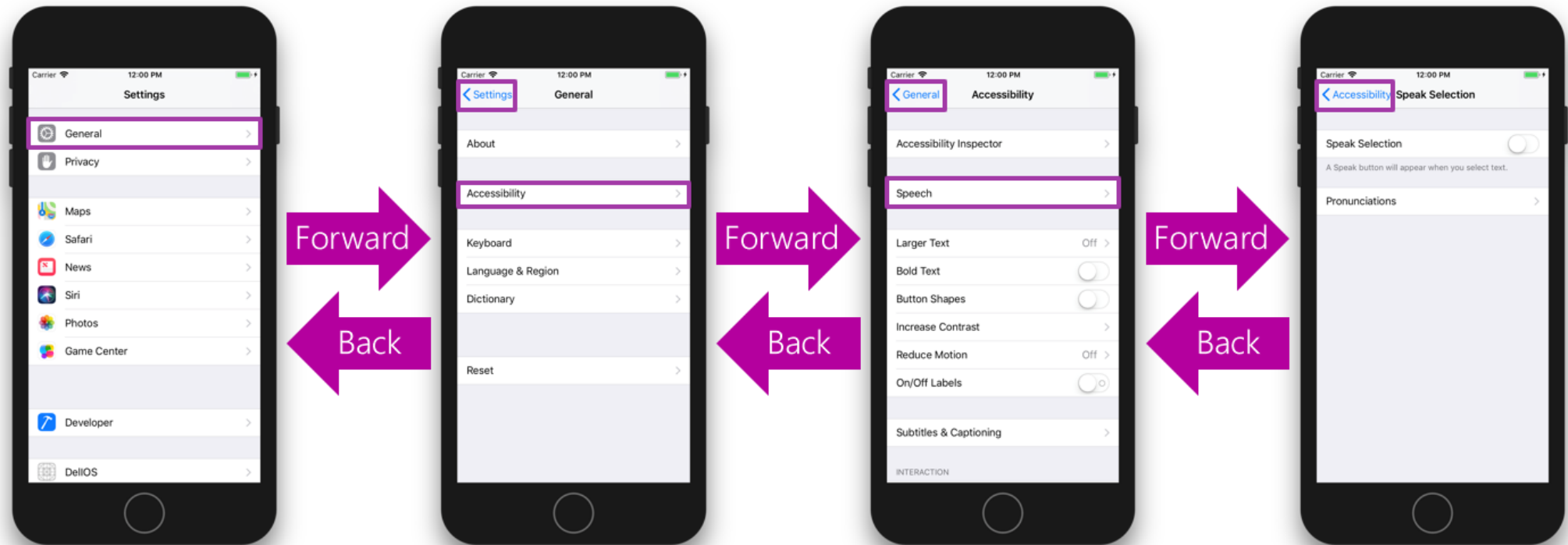
Contenido

- Introducción
- Stack navigation
- Tab navigation
- Master-detail
 - Drawer navigation
- API nativa del mapa

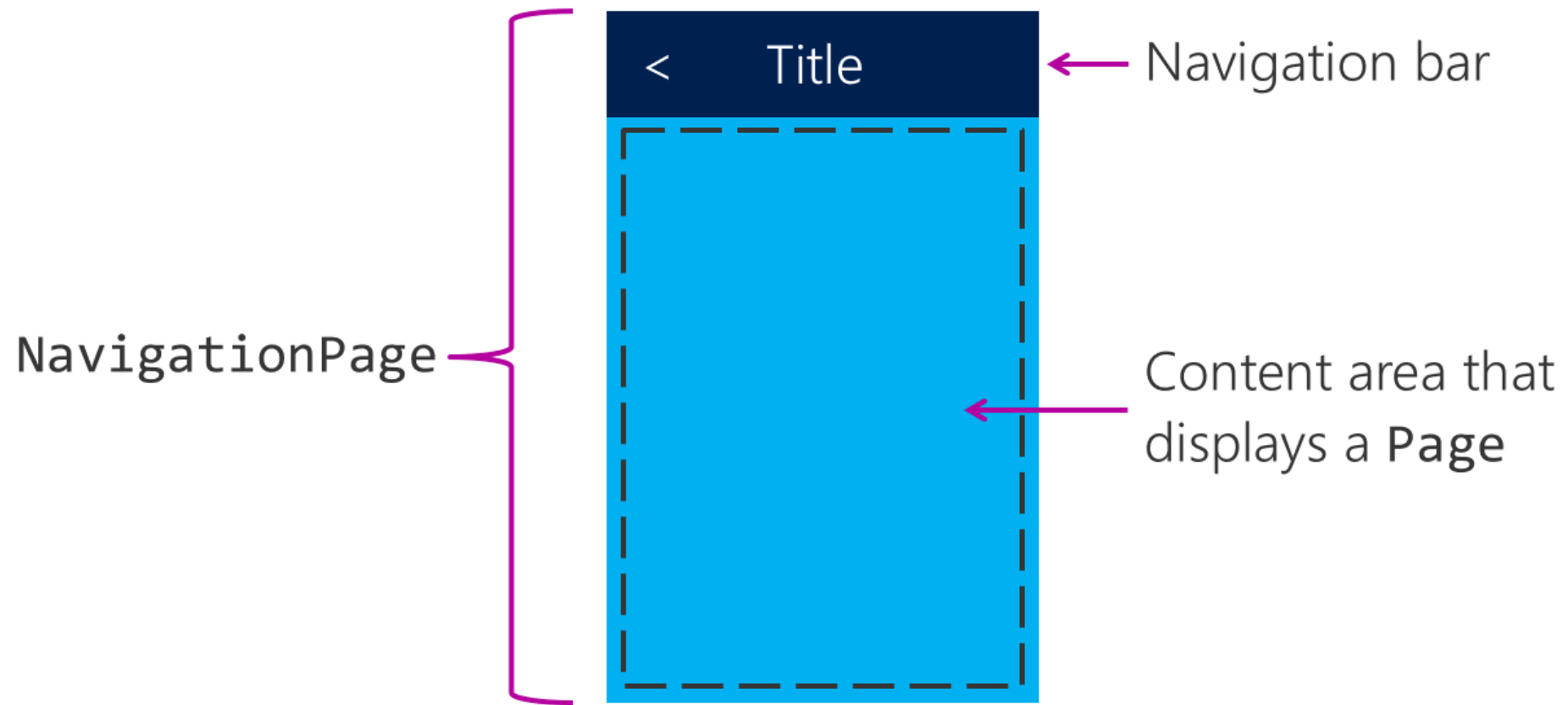
Introducción

- Un aspecto muy importante en la arquitectura de una aplicación es la forma en que el usuario navegará entre las pantallas.
- ¿Los usuarios navegarán hacia adelante y hacia atrás en una secuencia de pantallas? ¿La aplicación tiene una pantalla de inicio o hay múltiples pantallas con un mismo nivel de importancia?
- La elección que se haga para la navegación debe ser consistente con el contenido de la aplicación.

Stack navigation

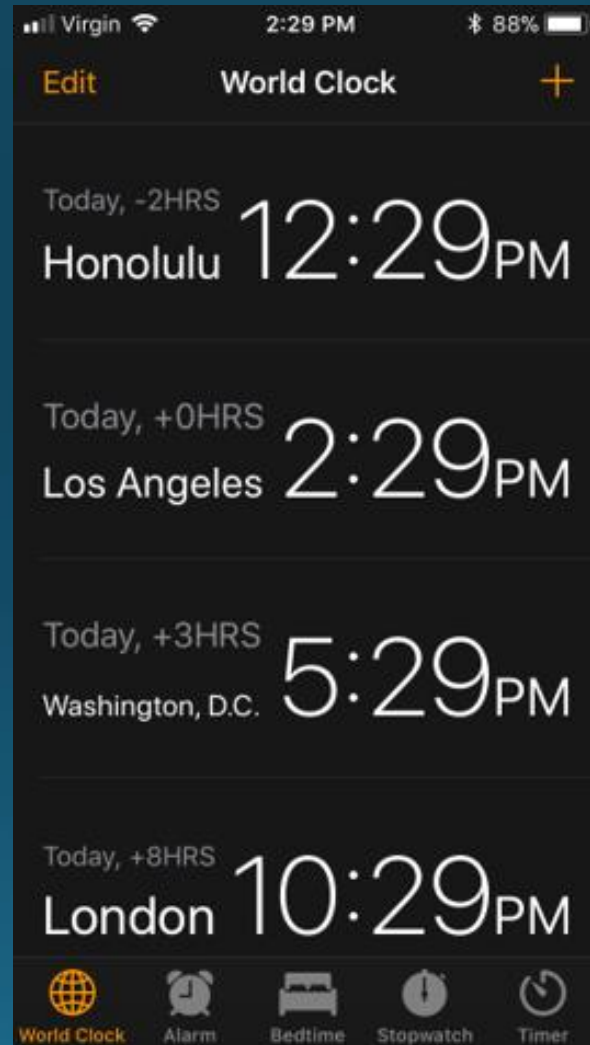


- Este patrón de navegación le permite al usuario moverse hacia adelante a nuevas pantallas y hacia atrás a pantallas previas.
- Las plataformas (Android, iOS) proveen alguna forma de botón para regresar a pantallas previas.
- En Xamarin.Forms, la navegación stack se provee mediante una **NavigationPage**. La pantalla provee una barra de navegación y un área de contenido para la pantalla actual.



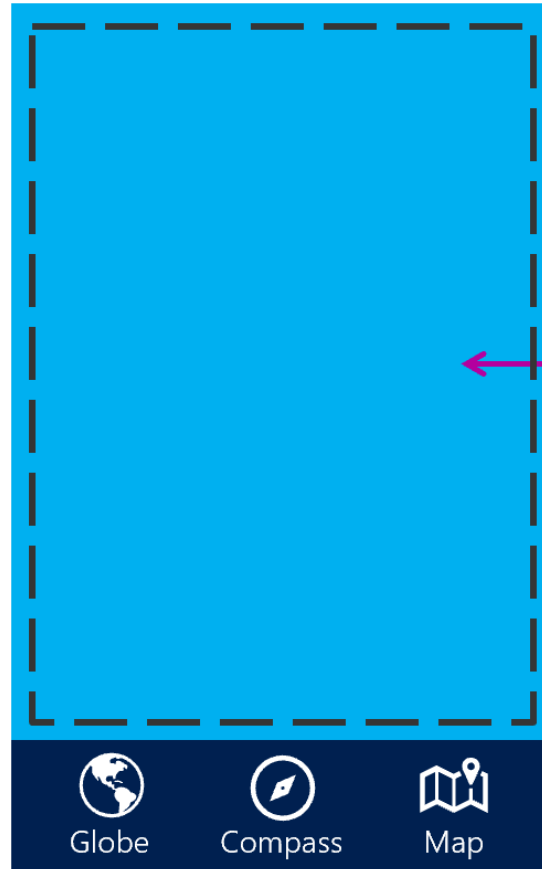
- Ejercicio #1 – Implementar stack navigation

Tab navigation



- En este patrón de navegación, se muestra una fila de pestañas de manera permanente en la parte superior o inferior de la pantalla.
- Cada pestaña representa una sección o una pantalla específica de la aplicación. Esta navegación es ideal cuando se tienen pantallas que el usuario utilizará frecuentemente, y que tienen un mismo nivel de importancia.
- **TabbedPage** implementa la navegación tab en Xamarin.Forms.

TabbedPage

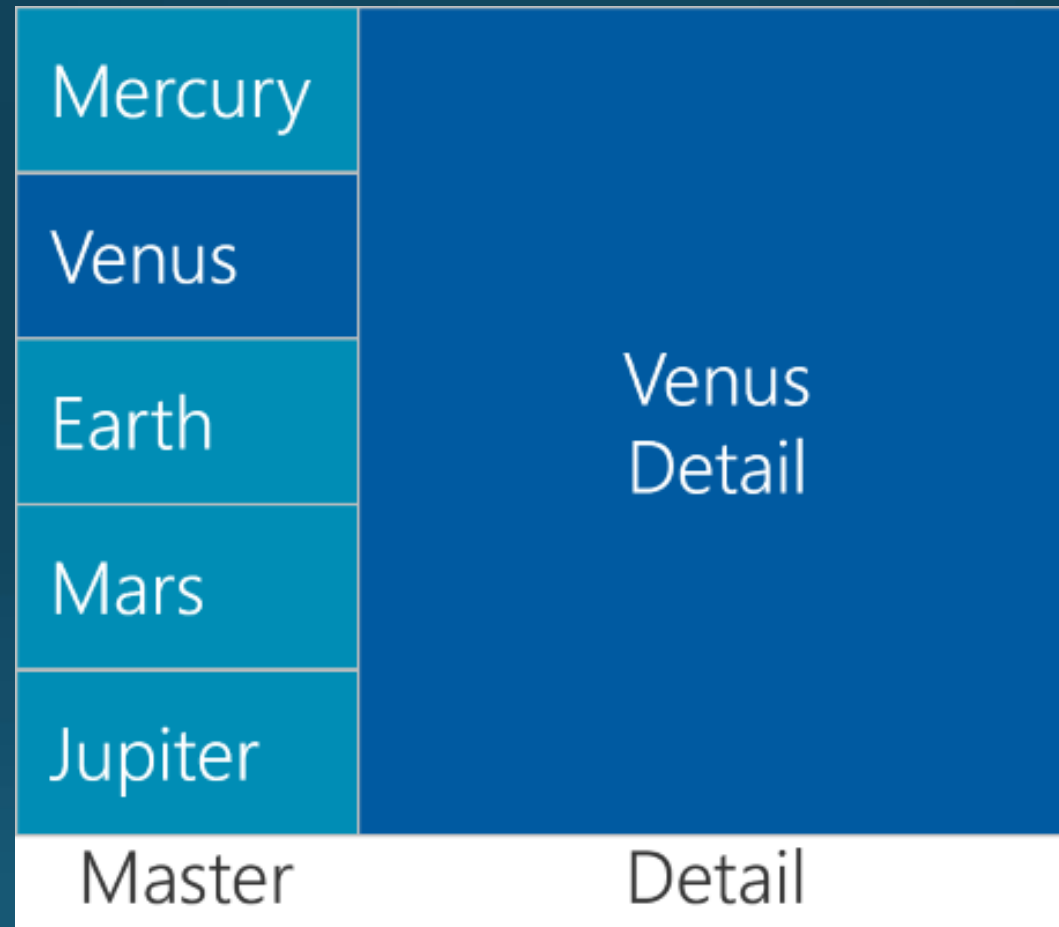


Content area that displays a **Page**

Tabs

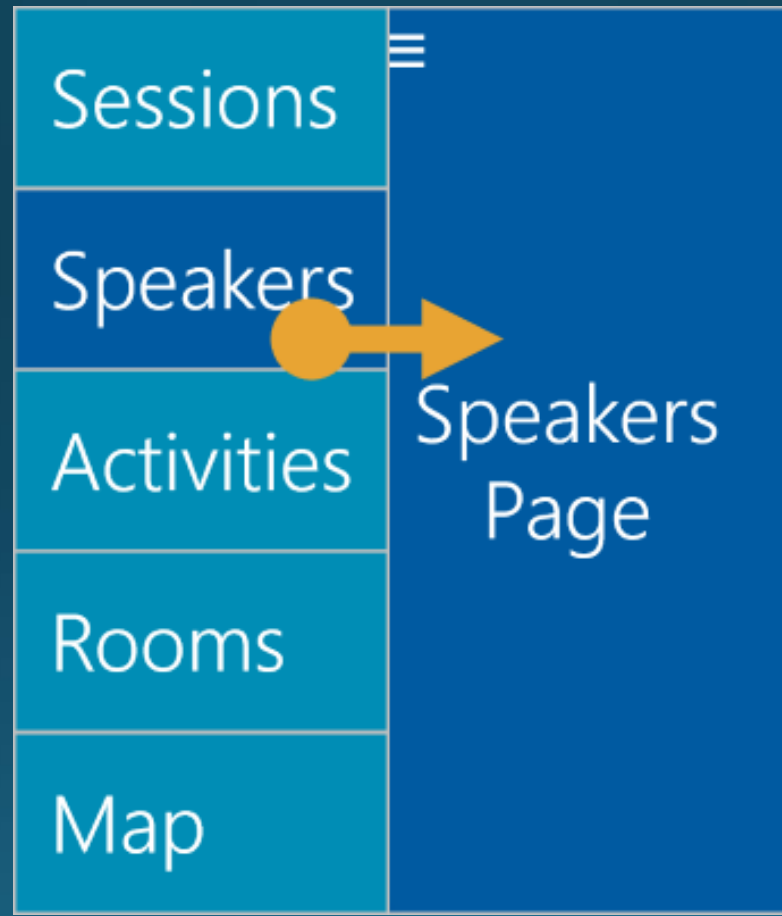
- Ejercicio #2 – Implementar tab navigation

Master-detail



- La navegación master-detail tiene dos componentes:
 - **Master:** Una sección que muestra una lista de elementos.
 - **Detail:** Una sección que muestra información detallada de cada elemento.
- **MasterDetailPage** es el tipo de página de Xamarin.Forms que provee una pantalla master y una detail, y que coordina la sincronización entre ellas.

Drawer navigation



- Este patrón de navegación utiliza un componente de UI que está fuera de la pantalla, llamado **drawer**.
- El **drawer** contiene un menú de pantallas. Típicamente, este componente se muestra al dar clic a un botón (comúnmente llamado “hamburguesa”).
- Cuando el usuario selecciona una pantalla, el **drawer** se vuelve a ocultar y la pantalla se muestra al usuario.

- **MasterDetailPage** se utiliza tanto para la navegación master-detail como para la navegación drawer.
- El valor de la propiedad **MasterBehavior** determina cómo se muestra la pantalla master:
 - **split**: Master siempre es visible.
 - **popover**: Master se muestra en un drawer (se oculta y se muestra).

- Ejercicio #3 – Crear drawer menu

- Ejercicio #4 – Implementar drawer navigation

API nativa del mapa

- Ejercicio #5 – Incluir un mapa en la app
 - Obtener una llave de la API de Google Maps:
<https://docs.microsoft.com/es-es/xamarin/android/platform/maps-and-location/maps/obtaining-a-google-maps-api-key?tabs=windows>

- Ejercicio #6 – Acceder a una aplicación de mapas