**Ejercicio #1 – App de Xamarin.Forms**

1. Crear una nueva app (**CDIS**) de Xamarin.Forms: New, Mobile App (Xamarin.Forms) 🡪 Blank App, Android y iOS.
2. Comentar el código:
   1. Los tres proyectos que se generan.
   2. Las referencias de Android y iOS (a Forms y al proyecto .NET Standard) Revisar que todas las referencias de Android se carguen.
   3. Las referencias del proyecto de .NET Standard.
   4. La clase **App**: El constructor -> **MainPage**, el **XAML** de MainPage (sólo un label).
   5. Las instancias de la clase **App** en **MainActivity** (Android) y **AppDelegate** (iOS).
3. Correr la aplicación (startup en Android).
4. Continuar con presentación.

**Ejercicio #2 – Crear la pantalla Phoneword (API nativa)**

1. En el proyecto .NET Standard, crear carpeta llamada **Phoneword**, dentro de la carpeta crear la clase **PhonewordTranslator** (copiar el código de la clase de la carpeta **Recursos**). Comentar el código.
2. Crear la UI basada en código:
   1. Borrar **MainPage.xaml**.
   2. Crear una nueva Content Page (C#), **PhonewordPage.cs**, y agregar el código de la UI: tres controles como atributos y el código del constructor …
   3. Crear el eventode **translateButton**: crear variable global **translatedNumber**, agregar propiedad **Clicked** del botón en el constructor y agregar el código del evento …
3. Crear el evento de **callButton**: agregar propiedad **Clicked** del botón en el constructor y agregar el código del evento(sin el cuerpo de **DisplayAlert**) …
4. Correr la app; modificar codebehind de **App** para que **MainPage** ahora sea **PhonewordPage** (se muestra la alerta).
5. Revisar la presentación (Código específico a plataformas (C#)).
6. Crear la interfaz **IDialer** …
7. Crear clases en cada plataforma para implementar la interfaz:
   1. Android y iOS: clase **PhoneDialer**, implementar **IDialer**.
   2. De la carpeta **Recursos**, copiar el código correspondiente a cada clase …
8. Registrar el servicio **Dependency** en cada plataforma (agregar **using** **Xamarin.Forms** y **using** proyecto de plataforma):

using Phoneword.iOS;

// This same attribute definition needs to go in each platform-specific file

[assembly: Dependency(typeof(PhoneDialer))]

1. Agregar permisos para Android:
   1. **Properties**, **Android Manifest**, Required permissions: **CALL\_PHONE**.
   2. También se deben solicitar permisos en tiempo de ejecución:
      1. Instalar el plugin de **NuGet**: **Plugin.Permissions**, en todos los proyectos.
      2. En **MainActivity**:
         1. Agregar using **Plugin.Permissions y Plugin.CurrentActivity**.
         2. En **OnCreate**, inicializar **CrossCurrentActivity**.
         3. En **OnRequestPermissionsResult** agregar el método de **Permissions** (comentar la línea de **Xamarin.Essentials**).
      3. En **PhoneDialer** de Android agregar **usings (Permissions, Permissions.Abstractions y Android.Telephony**.
2. Utilizar la implementación de **IDialer** en el código compartido en el cuerpo de **DisplayAlert**…
3. Correr la app (se realiza la llamada).
4. Continuar con la presentación.

**Ejercicio #3 – Modificar la pantalla Phoneword para utilizar Xamarin.Essentials**

1. Se requiere el permiso de Android: **CALL\_PHONE** (ya lo tenemos).
2. En **MainActivity**:
   1. En **OnCreate**, quitar la línea que inicializa **CrossCurrentActivity**.
   2. En **OnRequestPermissionsResult** quitar el método de **Permissions** y descomentar el de **Essentials**.
3. En **PhonewordPage**:
   1. Agregar using **Xamarin.Essentials**.
   2. En el código de la alerta, ejecutar **PhoneDialer.Open**; agregar las excepciones correspondientes.
4. Correr la app (se abre la app de llamadas).
5. Continuar con la presentación.

**Ejercicio #4 – Definir la UI de Phoneword con XAML**

1. Renombrar **PhonewordPage** a **OldPhonewordPage**.
2. Crear una ContentPage (XAML) llamada **PhonewordPage**; crear sus cuatro controles dentro de un **StackLayout** en XAML y generar el código codebehind:
   1. Asignar el **Padding** en **ContentPage** y el **Spacing** en **StackLayout**.
   2. Asignar los eventos de los botones desde XAML y en el constructor en codebehind.
3. Correr la app (se realiza la llamada).
4. Continuar con la presentación (Código específico a las plataformas (XAML)).

**Ejercicio #5 – Agregar código específico a las plataformas (XAML)**

1. Quitar el atributo **Padding** de **ContentPage**.
2. Agregar en el constructor del codebehind las condiciones de **Padding** para cada plataforma (iOS, Android).
3. Correr la app (se muestra el padding correcto).
4. Comentar el código del constructor; en XAML, agregar la etiqueta **OnPlatform** dentro de **ContentPage.Padding**, con los atributos correspondientes.
5. Correr la app (se muestra el padding correcto).