Procedimientos para la capa de presentación

*(documento preeliminar)*

*Mayo 2018*

|  |
| --- |
| 1. Instalación del ambiente de desarrollo. |
| Angular |
| **EQUIPO WINDOWS**  **Instalación de node.js y npm.**  **npm**: administrador de paquetes para JS, podemos encontrar bibliotecas como bootstrap, react y angular.   * Ir al sitio <https://nodejs.org/en/download/> y seguir las indicaciones. * Verificar la instalación en una ventana de comandos:   > node -v  > npm -v  **Instalación de Visual Studio Code.**  Editor de código fuente, ligero, potente disponible para Windows, MacOS y Linux, con soporte integrado para TS y Node.js y con una terminal embebida.   * Ir al sitio <https://code.visualstudio.com> y seguir las indicaciones.   **Instalación de la CLI (Command Line Interface) de angular.**  Herramienta para la ejecución de comandos para la creación y prueba de proyectos angular.   * En una ventana de comandos ejecutar:   > npm install -g @angular/cli  **Si aún no tiene git instalado, instalar git.**  Ir al sitio <https://git-scm.com/downloads> y seguir las indicaciones.  **EQUIPO MAC**  *No se ha probado, por ahora el curso se dará con equipos Windows.*  *May 28, 18. I[[1]](#footnote-1). LGF.* |
| Android |
| **EQUIPO WINDOWS**  **Instalación de Android Studio.**  IDE oficial para el desarrollo de aplicaciones para Android.   * Ir al sitio <https://developer.android.com/studio/#downloads> y seguir las indicaciones.   *Ya no es necesario instalar de manera individual Android SDK o Google Android emulator.*  **Si aún no tiene git instalado, instalar git.**  Ir al sitio <https://git-scm.com/downloads> y seguir las indicaciones.  **EQUIPO MAC**  *No se ha probado, por ahora el curso se dará con equipos Windows.*  *May 28, 18. I. LGF.* |

|  |
| --- |
| iOS |
| **EQUIPO MAC**  **Instalación de XCode.**  *Pendiente documentar.*  **Si aún no tiene git instalado, instalar git.**  Ir al sitio <https://git-scm.com/downloads> y seguir las indicaciones.  **EQUIPO WINDOWS CON UNA MAQUINA VIRTUAL MAC**  *Por probar, validar la pertinencia y documentar, por ahora se planea dar el curso en equipo MAC.*  *May 28, 18. I. LGF.* |
| Xamarin.Android |
| **EQUIPO WINDOWS**  **Instalación de Visual Studio Community 2017.**  Xamarin está incluido en todas las ediciones de Visual Studio sin costo, no requiere licencia adicional.   * Ir al sitio <https://www.visualstudio.com/vs/> y descargar Visual Studio Community 2017. * Instalar Visual Studio agregando “Mobile development with .NET”. * Verificar la instalación dando click a Help en el menú.   *Se instala Android SDK API level 27, JDK8 (requerido a partir de API level 24, JDK9 no es soportado) y Google Android emulator.*  **Agregar Xamarin a Visual Studio, en caso de ya tener instalado Visual Studio.**  Si ya tiene instalado Visual Studio y no realizó el paso anterior.   * Ejecutar Visual Studio Installer, dar click en Modify, select “Mobile development with .NET” y click Modify.   **Si aún no tiene git instalado, instalar git.**  Ir al sitio <https://git-scm.com/downloads> y seguir las indicaciones.  **EQUIPO MAC**  *No se ha probado, por ahora el curso se dará con equipos Windows.*  *May 28, 18. I. LGF.* |
| Xamarin.iOS |
| **EQUIPO WINDOWS**  **Instalación de Visual Studio Community 2017.**  *Mismas indicaciones que Xamarin.Android.*  **Agregar Xamarin a Visual Studio, en caso de ya tener instalado Visual Studio.**  *Mismas indicaciones que Xamarin.Android.*  **Importante: Se requiere una conexión activa a una MAC para compilar y ejecutar la aplicación.**  **Si aún no tiene git instalado, instalar git.**  Ir al sitio <https://git-scm.com/downloads> y seguir las indicaciones.  **EQUIPO MAC**  **Instalación de Visual Studio Community 2017.**  *Pendiente.*  **Agregar Xamarin a Visual Studio, en caso de ya tener instalado Visual Studio.**  *Pendiente.* |
| Xamarin.Forms |
| **EQUIPO WINDOWS**  **Instalación de Visual Studio Community 2017.**  *Mismas indicaciones que Xamarin.Android.*  **Agregar Xamarin a Visual Studio, en caso de ya tener instalado Visual Studio.**  *Mismas indicaciones que Xamarin.Android.*  **Importante: Se requiere una conexión activa a una MAC para compilar y ejecutar la aplicación para iOS.**  **Si aún no tiene git instalado, instalar git.**  Ir al sitio <https://git-scm.com/downloads> y seguir las indicaciones.  **EQUIPO MAC**  *Pendiente probar el caso de Xamarin.Forms para iOS.*  *May 28, 18. I. LGF.* |

|  |
| --- |
| 2. Creación y prueba de un proyecto tipo “Hola Mundo”. |
| Angular |
| **Creación del proyecto.**  Abrir el IDE:  Visual Studio Code.  Abrir la carpeta donde se desea crear el proyecto.   * [Ctrl – k Ctrl – o] o menú: File y Open Folder…   Abrir la CLI.   * [Ctrl – ñ] o menú: Terminal Integrado[[2]](#footnote-2)…   Asegurarse de estar en la carpeta correcta y teclear:  > ng new <name-app> (*it may take a while*)  **Prueba del proyecto.**  Iniciar la aplicación:   * En la CLI, cambiarse a la carpeta del proyecto que acaba de crear:   > cd <name-app>   * Teclear:   > ng serve --open (ng serve –o, ng serve)[[3]](#footnote-3)  Abrir la aplicación:   * En un navegador, teclear en la barra de direcciones:   <http://localhost:4200/>  Detener la aplicación:   * En la CLI:   [Ctrl – c]  **Nota: el inicio del proyecto por omisión es:**   * app.component.ts * app.component.html   *May 28, 18. I. LGF.* |
| Android |
| **Creación del proyecto.**  Abrir el IDE:   * Android Studio.   Da click a menu: File, New, New project…, para ejecutar el Create Android Project Wizard...   * Teclea el nombre del proyecto <name-app>. * Navegar a la carpeta donde se creará la carpeta del proyecto y completar con el nombre del proyecto. * Deja los valores por omisión en la pantalla Target Android Devices. * Selecciona una Empty Activity en la pantalla Add an Activity to Mobile. * Deja los valores por omisión en la pantalla Configure Activity.   **Prueba de la app en el emulador.**   * En el IDE, dar click al ícono Run (triangulo verde), o Shift F10. * Si es la primera vez, puede crear un nuevo dispositivo virtual. * Seleccionar el dispositivo (*me ha funcionado bien: Pixel\_2\_API\_27*).   **Prueba de la app en un dispositivo real.**  Conecta el dispositivo mediante cable USB.  Si es necesario instala el driver USB apropiado:   * Descargar el driver: * <https://developer.android.com/studio/run/oem-usb#Drivers> * Descomprimirlo. * Abrir desde el panel de control el Device Manager y localizar Portable Devices. * Dar click-derecho al device and select Update Driver Software. * Selecciona la carpeta que acabas de descomprimir.   Habilita USB debugging en tu dispositivo:   * Open the Settings app. * Scroll and select About phone. * Tap seven-times Build number. * Return to the previous screen to find Developer options. * Open Developer options and enable USB debugging.   Instala y ejecuta la app.   * En Android Studio dar click al ícono Run (triangulo verde), [Shift + F10]. * Selecciona tu dispositivo y click OK.   **Nota: el inicio del proyecto por omisión es:**   * MainActivity.java * activity\_main.xml   *May 28, 18. I. LGF.* |
| iOS |
| **Baje la app de Luis a la MAC. Funciona OK.** |
| Xamarin.Android |
| **Creación del proyecto.**  Abrir el IDE:  Visual Studio 2017 Community.  Da click a menu: File, New, Project…, para ejecutar el Create Android Project Wizard...   * Seleccionar Visual C#> Android> Aplicación de Android (Xamarin) * Teclea el nombre del proyecto <name-app>. * Navegar a la carpeta donde se creará la carpeta del proyecto. * Teclea el nombre de la solución. * Check a “Crear directorio para la solución” y click a Aceptar. * Seleccionar *Empty Activity.*   **Prueba de la app en el emulador.**  Ejecutar el emulador:   * Abrir Android Device Manager.   Menú: Tools> Android> Administrador de dispositivos Android.   * Seleccionar Android\_Accelerated\_x86\_Oreo y dar click Start. * Esperar a que esté listo.   Seleccionar el emulador y dar click al combo/ícono Run, [Ctrl F5].  **Nota: el inicio del proyecto por omisión es:**   * MainActivity.cs * activity\_main.axml   *May 28, 18. I. LGF.* |
| Xamarin.iOS |
| **Baje la app de Ale a Windows, requiero una conexión a MAC para probarlo.** |
| Xamarin.Forms |
| **Creación del proyecto.**  Abrir el IDE:  Visual Studio 2017 Community.  Da click a menu: File, New, Project…, para ejecutar el Create Android Project Wizard...   * Seleccionar Visual C#> Cross-Platfform> Aplicación de Android (Xamarin.Forms) * Teclea el nombre del proyecto <AppName>.   **Importante: note que a diferencia de los ejercicios anteriores, da problemas al incluir un guion medio en el nombre de la aplicación.**   * Navegar a la carpeta donde se creará la carpeta del proyecto. * Teclea el nombre de la solución. * Check a “Crear directorio para la solución” y click a Aceptar. * Seleccionar *Aplicación en blanco*, marcar las plataformas Android y iOS. * Seleccionar la estrategia .NET Standard y dar click a OK.   **Prueba de la app en el emulador.**  Probar la aplicación para Android.   * Dar click derecho a la solución y dar click a Restaurar paquetes de NuGet (quizá no restaure nada). * Dar click derecho a la solución y dar click a Administración de la configuración…, uncheck la compilación del proyecto para iOS. * Dar click derecho al proyecto Android y dar click a Establecer como proyecto de inicio. * Dar click al botón de Run (puede tardar un poco abrir el emulador).   Probar la aplicación para iOS.   * Pendiente, se requiere acceso a una MAC.   **Nota: el inicio del proyecto por omisión es:**   * App.xaml, App.xaml.cs de proyecto principal. * MainPage.xaml, MainPage.xaml.cs del proyecto principal. * MainActivity.cs, para proyecto de Android. * Main.cs (Application class), para el proyecto iOS (pendiente validar).   *May 28, 18. I. LGF.* |

|  |
| --- |
| 3. Diseño de una vista (layout) |
| Angular |
| **Utilizar bootstrap para el acomodo de las vistas y otros elementos html.** |
| Android |
| **Verificar/Agregar soporte para ConstraintLayout.**  Verificar que el archivo **build.gradle** tenga la siguiente línea en el bloque de **dependencies** {  Implementation ‘com.android.support,constraint:constraint-layout:1.1.0’  **Utilizar el Layout Editor para el diseño.**   * Abrir el layout (archivo xml) y seleccionar la vista de diseño. * Agregar los elementos desde **Palette Windos** al *blueprint*. * Agregar los *constraint* mediante el menú contextual o la **Attributes Window**. * Identificar cada elemento, incluir en el nombre el tipo de elemento y una parte significativa. * Definir los recursos *string* que se utilizarán para los atributos de texto. * Establecer el *text attribute* utilizando los recursos string que definió. * Establecer otras propiedades como tamaño de letra, etc. * Verificar/modificar que haga sentido el archivo *xml* generado. * Ejecutar la app para ver que todo salió OK.   **Elementos (Widgets).**  Etiqueta.  <TextView/> Botón.  <Button/>  …  **Propiedades de los Widgets.**  android:id="@+id/TextViewName" android:text="@string/TextToShow"  android:textSize="24sp"  android:textAlignment="textStart" android:textAllCaps="false"  …  **Propiedades del Layout.**  android:layout\_width="wrap\_content" android:layout\_height="0dp" android:layout\_marginTop="24dp"  android:layout\_marginBottom="24dp" android:layout\_marginEnd="24dp"  …  **Propiedades del ConstraintLayout.**  *Utilizar el Layout Editor.*  *May 28, 18. I. LGF.* |
| iOS |
| **Storyboards?** |
| Xamarin.Android |
| **Agregar soporte para ConstraintLayout:**  (no funcionó correctamente, usar RelativeLayout).  Desde el administrador de paquetes:   * Menú: Herramientas > Administración de paquetes NuGet. * En el combo Origen del paquete seleccionar nuget.org * Buscar el paquete Xamarin.Android.Support.Constraint.Layout e Instalar.   Desde la Consola de Administrador de Paquetes:  PM> Install-Package Xamarin.Android.Support.Constraint.Layout -Version 1.1.0  **Utilizando RelativeLayout:**  Abrir el layout (archivo axml), note que el layout activity\_main.axml por omisión utiliza un RelativeLayout  Abrir la vista de diseño para agregar el RelativeLayout si es necesario.  Abrir la vista de origen, editar el código del archivo axml para agregar los elementos y sus parámetros.  Ver el resultado en la vista de diseño.  Consultar los parámetros de android:layout en:  <https://developer.android.com/reference/android/widget/RelativeLayout.LayoutParams>  Ejecutar la app para ver que todo salió OK.  **Elementos (Widgets).**  Etiqueta.  <TextView/> Botón.  <Button/>  …  **Propiedades de los Widgets.**  android:id="@+id/TextViewName" android:text="@string/TextToShow"  android:textSize="24sp"  android:textAlignment="textStart" android:textAllCaps="false"  …  **Propiedades del Layout.**  android:layout\_width="wrap\_content" android:layout\_height="0dp" android:layout\_marginTop="24dp"  android:layout\_marginBottom="24dp" android:layout\_marginEnd="24dp"  android:layout\_centerHorizontal="true"  android:layout\_below="@id/TextViewSummary"  android:layout\_alignStart="@id/TextViewTaskToDoCount"  android:layout\_alignParentBottom="true"  android:layout\_alignParentEnd="true"  *May 28, 18. I. LGF.* |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
| **Utilizando StackLayout:**  <StackLayout> </StackLayout>  Abrir el archivo MainPage.xaml, editar el código xaml para agregar los elementos y sus propiedades.  Para ver un previo de la página utilizar Xamarin.Forms Previer  Menú: View > Other Windows > Xamarin.Forms Previewer  **Controles (Views).**  <Label/>  <Button/>  Text="Summary"  BackgroundColor="Red"  Font="20"  HorizontalOptions="Center"  VerticalOptions="CenterAndExpand" |
|  |

|  |
| --- |
| 4. Control de versiones con git (crear repositorio y push inicial) |
| Angular |
|  |
| Android |
| **Crear el repositorio.**   * Dar click a menú: VCD > Import into Version Control > Create Git Repository… * Seleccionar la carpeta del proyecto.   Android ejecuta por ti el comando **git init** para crear el repositorio, es decir, agrega la carpeta .git y un archivo **HEAD** inicial para la **master branch**.  **Hacer un push inicial.**  Hacer un commit inicial.   * En la Version Control Window [Alt + 9], dar click a commit para agregar todos los archivos y hacer un primer commit. * Agregar el comentario “Initial Commit” y firmar.   Crear el repositorio remoto.   * Si es necesario crear una cuenta de github. * Crear un New Empty repository without readme file en github.   Hacer el push al repositorio remoto.   * Dar click a menu: VCD > Git > Push… * Indicar el nombre del remote y la URL. * Dar click a Push.   Android ejecuta por ti los siguientes comandos:   * git remote add origin https://github.com/LilianaGF/task-app.git * git push -u origin master |
| iOS |
|  |
| Xamarin.Android |
| **Crear el repositorio.**  Al crear la solución, se puede marcar la casilla Create new Git repository.  Para una solución donde no se creó el repositorio al inicio, dar click al botón Publish en la esquina inferior derecha de Visual Studio, esto creará un Git repository en la carpeta de la solución.  **Hacer un push inicial.**  Abrir la ventana Team Explorer.  Menú: Ver > Team Explorer  Para commmit, dar click a Changes.  Para push, dar click a Sync,  En Synchonization dar click a Push ( en español aparece como Confirmaciones de salida-Insertar) |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

|  |
| --- |
| 5. Modelo. |
| Angular |
| ***Creación de una carpeta para mantener todos los archivos del modelo***  ***Abrir el explorador de soluciones***  ***Ctrl+Shift\*E***  ***Dar click-derecho a la carpeta ./src/app***  ***Seleccionar New Folder***  ***El nombre debe ser model.***  Creación de una clase del modelo  En el explorador de soluciones dar click-derecho a la carpeta model  Seleccionar New File  El contenido del archivo debe ser como el siguiente: |
| Android |
|  |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

|  |
| --- |
| 6. Navegación entre pantallas. |
| Angular |
|  |
| Android |
|  |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

|  |
| --- |
| 7. Lógica de la capa de presentación. |
| Angular |
|  |
| Android |
|  |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

|  |
| --- |
| 8. Creación de servicios. |
| Angular |
|  |
| Android |
|  |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

|  |
| --- |
| 9. Acceso a datos. |
| Angular |
|  |
| Android |
|  |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

|  |
| --- |
| 10. Escritura y modificación de datos. |
| Angular |
|  |
| Android |
|  |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

|  |
| --- |
| 11. |
| Angular |
|  |
| Android |
|  |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

Android Studio:

Si al ejecutar la aplicación se obtiene el mensaje:

Installation failed with message Invalid File: C:\AndroidStudioProjects\taskapp\app\build\intermediates\split-apk\debug\slices\slice\_2.apk. It is possible that this issue is resolved by uninstalling an existing version of the apk if it is present, and then re-installing.

Menu: Build > Clean Project

Menú: Build > Build APK(s)

Run again.

Angular:

Al Teclear:

> ng serve --open (ng serve –o, ng serve)[[4]](#footnote-4)

*Troubleshooting.*

*Error message: node\_modules appears empty.*

*Why: not in the app directory.*

*Solution: cd to the app directory.*

1. *I, para indicar documentación inicial, utilizar otras notas para modificaciones posteriores.* [↑](#footnote-ref-1)
2. Pendiente checar nombre que aparece en inglés. [↑](#footnote-ref-2)
3. angular CLI calls tsc under the hood, bundles are injected at runtime. [↑](#footnote-ref-3)
4. angular CLI calls tsc under the hood, bundles are injected at runtime. [↑](#footnote-ref-4)