Presentation layer

Procedures for mobile development

*(preliminary document)*

*May 2018*

Contents

[1. Instalación del ambiente de desarrollo. 3](#_Toc515783211)

[2. Creación y prueba de un proyecto tipo “Hola Mundo”. 6](#_Toc515783212)

[3. Diseño de una vista (layout) 10](#_Toc515783213)

[4. Control de versiones con git (crear repositorio y push inicial) 14](#_Toc515783214)

[5. Modelo. 17](#_Toc515783215)

[6. Navegación entre pantallas. 18](#_Toc515783216)

[7. Lógica de la capa de presentación. 19](#_Toc515783217)

[8. Creación de servicios. 20](#_Toc515783218)

[9. Acceso a datos. 21](#_Toc515783219)

[10. Escritura y modificación de datos. 22](#_Toc515783220)

[11. Solución de problemas 23](#_Toc515783221)

|  |
| --- |
| 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, 2018. 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**  **Instalación de Android Studio.**  *No se ha probado, por ahora el curso se dará con equipos Windows.*  *May 28, 2018. LGF.* |

|  |
| --- |
| iOS |
| **EQUIPO MAC**  **Instalación de XCode.**  Abrir la App Store, buscar **Xcode**, dar click en obtener.  También se pueder ir al siguiente sitio y seguir las indicaciones.  Es posible que se requiera crear un Appli  **Si aún no tiene git instalado, instalar git.**  Ir al sitio <https://git-scm.com/downloads> y seguir las indicaciones.  **EQUIPO WINDOWS**  *Con una máquina virtual, por ahora se planea dar el curso en equipo MAC.*  *May 28, 2018. 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, 2018. LGF.* |
| Xamarin.iOS |
| **EQUIPO MAC**  **Instalación de XCode.**  **Instalación de Visual Studio Community 2017 para MAC.**  *Para trabajar desarrollo para iOS se requiere tener instalado previamente xCode.*  Desde el equipo MAC Ir al sitio <https://www.visualstudio.com/>, descargar Community 2017 para Mac.  Dar doble-click al archivo con extensión dmg descargado.  Dar doble-click al Installer y seguir las indicaciones.  **Si aún no tiene git instalado, instalar git.**  **https://visualstudio.github.com/index.html**  **EQUIPO WINDOWS**  *\*Requiere una conexión remota a una MAC.*  *\*Se probó, es muy lento, por ahora el curso se dará con la MAC.*  **Instalación de Visual Studio Community 2017.**  **Agregar Xamarin a Visual Studio, en caso de ya tener instalado Visual Studio.**  **Crear una conexión a un equipo MAC que tenga instalado xCode y VisualStudio for MAC**  Habilitar *remote login* en la MAC   * On the Mac, open System Preferences and go to the **Sharing (Compartir)** pane. * Check **Remote Login (Sesión remota)** in the **Service** list. * Permitir a todos los usuarios   **Si aún no tiene git instalado, instalar git.**  *May 28, 2018. LGF.* |
| Xamarin.Forms |
| **EQUIPO WINDOWS**  *Mismas indicaciones que Xamarin.iOS*  **EQUIPO MAC**  *No se ha probado, por ahora el curso se dará con equipos Windows.*  *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[[1]](#footnote-1)…   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)[[2]](#footnote-2)  Abrir la aplicación:   * En un navegador, teclear en la barra de direcciones:   <http://localhost:4200/>  Detener la aplicación:   * En la CLI:   [Ctrl – c]  *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.   *May 28, 18. I. LGF.* |
| iOS |
| **Creación del proyecto.**  Abrir el IDE:   * xCode.   Dar click a File> New> Project:   * Select iOS. * Select Single View App. * Teclear el nombre de la app. * Selecciona el lenguaje Swift. * (Por ahora no incluiré pruebas). * Seleccionar Create Git repository on my Mac.   **Prueba de la app en el emulador.**  Dar click al ícono run. |
| 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].  *May 28, 18. I. LGF.* |
| Xamarin.iOS |
| **Creación del proyecto.**  Abrir el IDE:  Visual Studio 2017 Community para MAC.  Da click a menu: Archivo> Nueva solución   * Seleccionar iOS> Aplicación, Aplicación de una vista y dar click a Siguiente. * 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”. * Usar Git para el control de versiones y click a Crear.   **Prueba de la app en el emulador.**  Dar click al ícono Run.  *May 28, 18. I. LGF.* |
| 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. * Seleccionar el dispositivo y dar click al botón de Run (puede tardar un poco abrir el emulador).   Probar la aplicación para iOS.   * Establecer comunicación con una MAC, como se especifica arriba. * En esta ocasión check la compilación para iOS. * Dar click derecho al proyecto iOS y dar click a Establecer como proyecto de inicio. * Seleccionar el dispositivo y dar click al botón de Run, alternatively, choose Product > Run or press Command-R (tarda mucho).   *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 *(Con el Layout Editor y modificando el archivo xml).* |
| **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 *(Con el* [*Interface Builder*](https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/GlossaryDefinitions.html#//apple_ref/doc/uid/TP40015214-CH12-SW47)*, se genera código).* |
| **Utilizar el** [**Interface Builder**](https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/GlossaryDefinitions.html#//apple_ref/doc/uid/TP40015214-CH12-SW47) **(visual interface editor) para el diseño del storyboard.**  *\*El* ***storyboard*** *contiene una* [***scene***](https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/GlossaryDefinitions.html#//apple_ref/doc/uid/TP40015214-CH12-SW62)*, which represents a screen of content in your app.*   * Habilitar mostrar el marco de los elementos en la escena.   Choose Editor > Canvas, and make sure Show Bounds Rectangles is selected.  *\*This setting causes Interface Builder to draw a blue bounding box around all the views in the canvas.*   * Mostrar la librería de objetos:   View > Utilities > Show Object Library.  Agregar elementos a la escena:   * Buscar los elementos en el filtro del Object Library   *Por ejemplo escribir Text View.*   * Arrastrar los elementos a la scene. * Modificar su tamaño y posición. * Utilizar el **Atrributes inspector** para modificar atributos: Text, Color, Font, Placeholder, ….   View> Utilities> Show Attributes Inspector (cuarto ícono en la ventana de la derecho).  Utilizar los botones de la parte inferior del *canvas* para:   * Agrupar elementos (Embed in Stack). * Agregar constraints y alineamiento.   **Revise el Outline View**  Muestra como organizan los elementos que se han agregado a la scene.  El primer ícono en la parte inferior del canvas oculta y muestra el outline view.  **Elementos (views objects: UIView)**  Text View  Button  **Propiedades de los elementos.**  Establecer en el Attributes inspector:  Title (double-click)  Text (double-click)  **Alineamiento y constraints.**  Establecer en el Canvas (íconos inferiores).  *May 28, 18. I. LGF.* |
| Xamarin.Android *(Editando el archivo axml y ver resultado en la vista de diseño)*. |
| **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 *(De forma gráfica, se genera código).* |
| Abrir el storyboard (Main.storyboard).  Agregar los elementos.  Establecer propiedades y constraints. |
| Xamarin.Forms *(Editando el archivo xaml).* |
| **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/>  …  **Propiedades de los Controles.**  x:Name="LabelSummary"  Text="Summary"  Font="30"Text="Summary"  **Propiedades del Layout.**  RelativeLayout.XConstraint=  "{ConstraintExpression Type=RelativeToParent,  Property=X, Factor=1, Constant=120}"  RelativeLayout.YConstraint=  "{ConstraintExpression Type=RelativeToParent,  Property=Y, Factor=1, Constant=80 }" />  RelativeLayout.XConstraint=  "{ConstraintExpression Type=RelativeToView, ElementName=LabelSummary,  Property=X, Factor=1, Constant=-10}"  RelativeLayout.YConstraint=  "{ConstraintExpression Type=RelativeToView, ElementName=LabelSummary,  Property=Y, Factor=1, Constant=100}" /> |

|  |
| --- |
| 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   **Clonar un proyecto remoto a un repositorio local.**  *El siguiente procedimiento es independiente de la tecnología, aplica a todos los casos.*   * Abrir una bash terminal y ubicarse en la carpeta donde se clonará el proyecto. * Ejecutar el comando   git clone <https://github.com/>...  *May 28, 18. I. LGF.* |
| iOS |
| **Crear el repositorio.**   * Al crear el proyecto, Seleccionar Create Git repository on my Mac.   **Hacer un push inicial.**  Crear el repositorio remoto.   * Si es necesario crear una cuenta de github. * Crear un New Empty repository without readme file en github.   Agregar la cuenta de github en Xcode   * Xcode> Preferences…Accounts. * Click al ícno + para gregar una cuenta.   Agregar el remote:   * Abrir el Souce Control Navigator. * Click ícono junto al del Project Navigator (segundo ícono). * Control-click a la raíz del árbol que se muestra. * Seleccionar Add Existing Remote… * Configurar la dirección del repositorio creado.   Hacer el commit y el push.   * Ir al menú Source Control > Commit y ahcer el commit dejando un mensaje. * Ir al menú Source Control> Push, dar click a push.   **Clonar un proyecto remoto a un repositorio local.**  *Ver arriba.*  *May 28, 18. I. LGF.* |
| 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.**  Crear el repositorio remoto.   * Si es necesario crear una cuenta de github. * Crear un New Empty repository without readme file en github.   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)  **Clonar un proyecto remoto a un repositorio local.**  *Ver arriba.*  *May 28, 18. I. LGF.* |
| Xamarin.iOS |
| **Crear el repositorio.**  Al crear la solución.  **Hacer un push inicial.**  Crear el repositorio remoto:   * Crear un New Empty repository without readme file en github.   Desde Visual Studio Community para Mac:   * Ir al menú: Control de versiones> Revisar la solución y confirmar. * Dar click a confirmar y agregar un mensaje. * Hacer el push desde una terminal, ubicarse en la carpeta del proyecto y ejecutar los siguientes comandos:   git remote add origin <https://github.com/LilianaGF/xamarin_ios>  git push origin master  **Clonar un proyecto remoto a un repositorio local.**  *Ver arriba.* |
| Xamarin.Forms |
| *Mismas indicaciones que Xamarin.Android.* |

|  |
| --- |
| 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. Solución de problemas |
| Angular |
| **Se presenta al teclear:**  > ng serve --open (ng serve –o, ng serve)  **Se manifiesta:**  Error message: node\_modules appears empty.  **Explicación:**  Not in the app directory.  **Solution:**  cd to the app directory. |
| Android |
| **Se presenta al teclear:**  **Se manifiesta:**  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.  **Solution:**  Menu: Build > Clean Project  Menú: Build > Build APK(s)  Run again. |
| iOS |
|  |
| Xamarin.Android |
|  |
| Xamarin.iOS |
|  |
| Xamarin.Forms |
|  |

1. Pendiente checar nombre que aparece en inglés. [↑](#footnote-ref-1)
2. angular CLI calls tsc under the hood, bundles are injected at runtime. [↑](#footnote-ref-2)