



|  |   |   |   |
|--|---|---|---|
| <br>CIPF VIRGEN DE GRACIA | <b>Desarrollo de Interfaces.</b><br><b>Resultado de aprendizaje: RA2</b>                  |   | <br><b>Página 1 de 3</b> |
|  | <b>PRÁCTICA EVALUABLE</b><br><b>Criterios de evaluación:</b><br><b>2.1, 2.2, 2.3, 2.4</b> | <b>Fecha entrega</b><br><b>25.02.2024</b> |   |

Realizaréis en grupo, **SIGUIENDO LA METODOLOGÍA SCRUM**, la siguiente aplicación en .NET MAUI con acceso a base de datos SQLITE.

La llamaréis **SG\_MAUI<VuestroNombreDelGrupo>**.

La apariencia general de la aplicación se deja a elección del alumno, si bien deben seguir las siguientes indicaciones:

- La aplicación tendrá navegación entre varias páginas.
- La página inicial debe ser una página de “login” que paso a explicar.

### **PÁGINA LOGIN**

Consistirá en una página que tendrá un contenedor de tipo “*VerticalStackLayout*” en la cual habrá que mostrar una imagen (elegida por el alumno), dos controles de tipo “*Entry*” para insertar los campos “login” y “contraseña” (este último campo debe mostrar asteriscos al escribir) y dos botones: “Aceptar” y “Limpiar”.

- El botón “Limpiar” limpiará el contenido de los dos controles de tipo “*Entry*”.
- El botón “Aceptar” solamente estará activo cuando el usuario haya escrito contenido (que no sea únicamente espacios en blanco) en los dos controles de tipo “*Entry*”.

El funcionamiento de “Aceptar” distinguirá dos situaciones:

- Si en la tabla de la base de datos que sirve para almacenar los “usuarios” de la aplicación no existe ningún registro (es decir, con la aplicación recién instalada) se va a considerar siempre que la validación ha sido correcta, por lo que:
  - Se insertará un nuevo registro en esa tabla de la base de datos con la información que se conoce de ese usuario: login y contraseña.
  - Posteriormente, se muestra la página principal de la aplicación, quitando la “página de login” de la pila de navegación para que no se vuelva a mostrar.
- Si en la tabla de la base de datos que sirve para almacenar los “usuarios” de la aplicación sí tiene registros se debe consultar sobre la tabla a ver si alguno de los registros de la aplicación coincide con ese login y esa contraseña que ha metido el usuario, por lo que:
  - Si no se encuentra ningún registro que coincida, se mostrará un mensaje de aviso al usuario y se limpiarán los campos de la página de login.
  - Si encuentra algún registro que coincida, se muestra la página principal de la aplicación, quitando la “página de login” de la pila de navegación para que no se vuelva a mostrar.

## **PÁGINA PRINCIPAL**

Será una página con un diseño de *"HorizontalStackLayout"* en la que se muestren 2 controles *"ImageButton"* (las imágenes a mostrar en cada uno se dejan a elección de los alumnos) que sirvan para:

- Gestión Usuarios
- Salir

Esta página principal no mostrará la barra de navegación.

Si el usuario pulsa en la opción de *"SALIR"* se le preguntará al usuario si está seguro, y si es así se cerrará totalmente la aplicación.

Si el usuario pulsa en el primer botón nos llevará a una nueva página, que se deberá apilar en la pila de navegación, y que paso a explicar a continuación.

## **PÁGINA GESTIÓN USUARIOS**

Esta página servirá para realizar la gestión de los usuarios. Se deben tener en cuenta las siguientes consideraciones:

- Desde esta barra se podrá volver a la Página principal, pero no debe mostrar la barra de navegación.
- El resto de la página se divide en dos zonas: superior e inferior.
- En la zona superior estarán los controles para guardar los datos de los usuarios (login, contraseña, email y otro más a elección de los alumnos), así como dos botones: Guardar y Borrar.
- El funcionamiento de esos botones (que se debe implementar con comandos)
  - *"Guardar"*: para guardar en la BD los cambios con la información que haya en los controles
  - *"Borrar"*: para eliminar el elemento que haya sido seleccionado.
- Cada vez que se produzca un cambio de información (nuevo usuario, modificación de datos de un usuario o eliminación de un usuario) se debe reflejar automáticamente en la lista de la zona inferior. También debe mostrar un mensaje de aviso al usuario cuando se haga alguna de esas operaciones.
- La lista distinguirá si hay elementos o no, entonces:
  - Si no hay elementos, se mostrará una imagen específica (elegida por el alumno) y un texto explicando que no hay usuarios en la BD.
  - Si hay elementos que mostrar, se hará en una *"CollectionView"* en forma de listado lineal de orientación *"horizontal"*, con un espaciado entre elementos de varios píxeles (a elegir por el alumno).
    - Cada elemento de la lista debe mostrarse dentro de un *Frame* que contenga dentro únicamente los campos *"login"* y *"contraseña"*.
- Si se selecciona un elemento de la lista se debe cargar en los controles de la zona superior la información de ese usuario. Solo se puede seleccionar un único elemento a la vez.

**IMPORTANTE A TENER EN CUENTA PARA LA APLICACIÓN:**

- **REALIZAR TODO EL PROYECTO SIGUIENDO LA METODOLOGÍA SCRUM, CON UN PROYECTO Y RESPOSITORIO GIT PRIVADO, TRABAJANDO CON EL FLUJO GITFLOW UTILIZANDO GITKRAKEN. POSTERIORMENTE SE PUEDE COMPROBAR CÓMO SE HA TRABAJADO**
- **El proyecto lo debe crear el *Scrum Master*** designado por el profesor, y agregar como colaboradores tanto al resto de compañeros del *Team* como al propio profesor (**jagarcias1978**).
- El proyecto debe tener como mínimo 2 sprints. Se *sugiere*:
  - 1º sprint: Página Login y página Principal. 20 y 21 de febrero.
  - 2º sprint: Página de Gestión de Usuarios. Del 22 al 25 de febrero.
- **CADA TRABAJO DEBE SER ORIGINAL:** se prestará especial atención a que no haya código copiado entre alumnos, ya que todos los trabajos implicados en la copia tendrán una **calificación de 0**.
- Sigue los patrones y la estructura de clases que se han seguido para el ejemplo que hemos realizado en clase: MVVM, Comandos, Repository, Converter, Singleton...
- Todos los botones funcionarán a través de Comandos.