



# Aplicación básica de gestión de Usuarios para una academia que imparte clases de Inglés

Vitor Hugo Augusto Rodrigues

Claudio Peña Rodríguez

## Índice

<b>Introducción.....</b>	<b>3</b>
<b>Objetivos.....</b>	<b>3</b>
<b>Tecnologías y lenguajes .....</b>	<b>4</b>
<b>Metodología de trabajo y Entorno de trabajo .....</b>	<b>5</b>
<b>Modelo de datos .....</b>	<b>6</b>
<b>Vistas .....</b>	<b>6</b>
Vistas Administrador .....	6
Vistas Usuario .....	9
Vistas Comunes .....	13

## Introducción

Para desarrollar este proyecto, hemos optado por la idea de crear una aplicación multiplataforma para gestionar usuarios. En este caso, el ámbito de la aplicación estará destinado a la enseñanza, en particular, a una academia de inglés.

Para diseñar e implementar esta aplicación multiplataforma, nos apoyaremos con los puntos estudiados en la asignatura.

A continuación, resumiremos las fases más importantes de nuestro proyecto y las tecnologías empleadas en el desarrollo del mismo.

## Objetivos

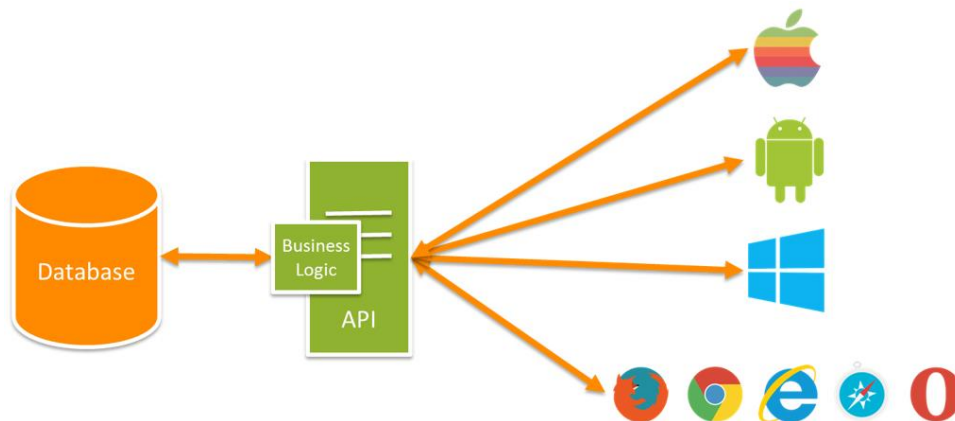
Entre los objetivos que queremos alcanzar con la realización de este proyecto, destacamos la creación de una aplicación dinámica donde tengamos operaciones contra una base de datos, además de conocer las nuevas tecnologías que existen para el desarrollo de aplicaciones multiplataforma.

A continuación, detallaremos los puntos que queremos alcanzar en nuestra aplicación:

- Registro de usuarios (*Create*).
- Actualización de datos de usuarios (*Update*).
- Login de Usuarios.
- Listado de cursos, noticias y cursos matriculados por usuario (*Get*).
- Matriculación en un curso (por parte de un usuario).
- Creación de Noticias (*Create*).
- Creación de Cursos (*Create*).
- Vista Usuario y vista Administrador.

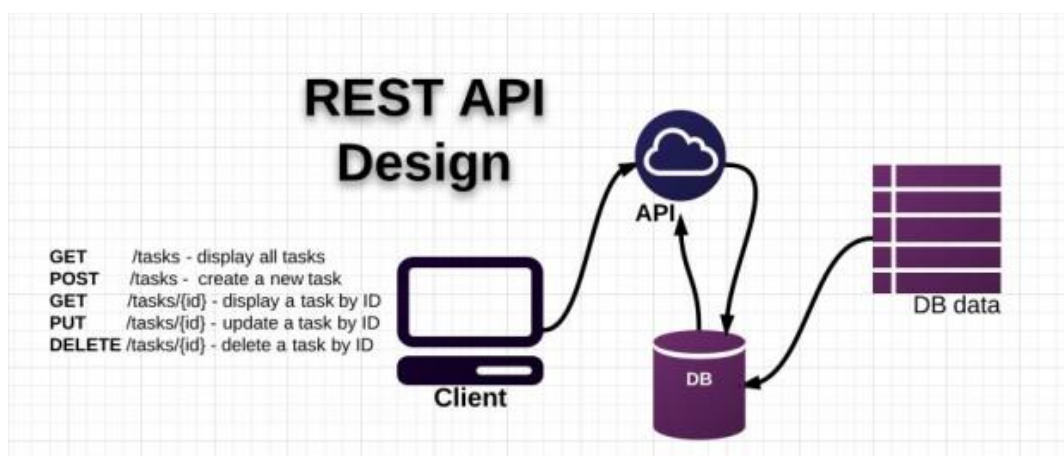
## Tecnologías y lenguajes

Anteriormente hemos mencionado que desarrollaremos una aplicación multiplataforma dinámica, o lo que es lo mismo, deberemos estar realizando constantemente llamadas a nuestra base de datos para refrescar el contenido de nuestra aplicación. Para ello, utilizaremos una API RESTful que servirá de intermediario entre la base de datos y los diferentes dispositivos, es decir, crearemos una simple API que realizará las peticiones a la base de datos.



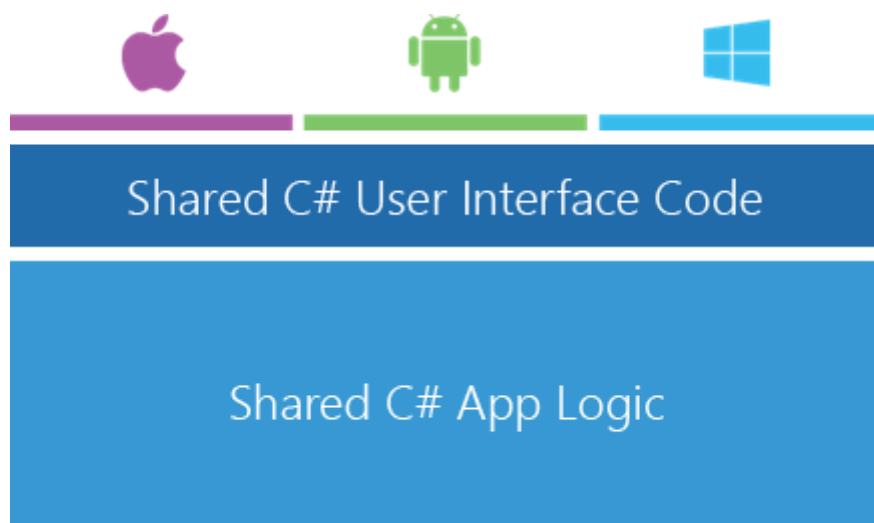
Usaremos el framework php *Laravel*, el cual nos permitirá crear la API de una forma sencilla. En ésta tendremos las operaciones básicas que se realizan sobre la base de datos:

- GET muestra todos los registros o un registro concreto.
- POST inserta un nuevo registro.
- PUT actualiza un registro.
- DELETE elimina un registro.



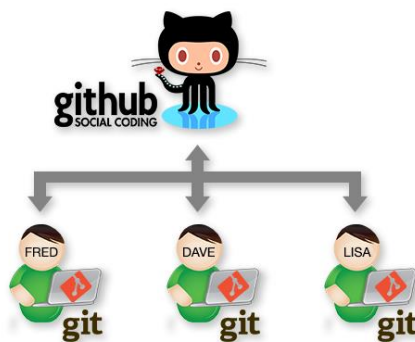
Una vez tengamos la comunicación entre la base de datos y la API, ya podremos pasar al desarrollo de nuestra aplicación.

Como hemos visto en clase, Visual Studio nos proporciona varias herramientas para llevar a cabo esta tarea. En nuestro caso, nos hemos decantado por utilizar *Xamarin.Forms*, ya que nos permite desarrollar código una vez y éste se adapta a las plataformas actuales: Android, UWP y iOS. *Xamarin* es una plataforma que nos permite crear aplicaciones nativas y utilizar un solo lenguaje de programación, C#. Además, *Xamarin.Forms* permite el prototipado rápido de aplicaciones que pueden evolucionar con el tiempo hacia aplicaciones complejas.



## Metodología de trabajo y Entorno de trabajo

La metodología de trabajo que seguiremos para desarrollar este proyecto se basará en la utilización de un sistema de versiones, en este caso, *Git*. Para la distribución de tareas, nos iremos dividiendo el trabajo en función de la disponibilidad de cada miembro del grupo.



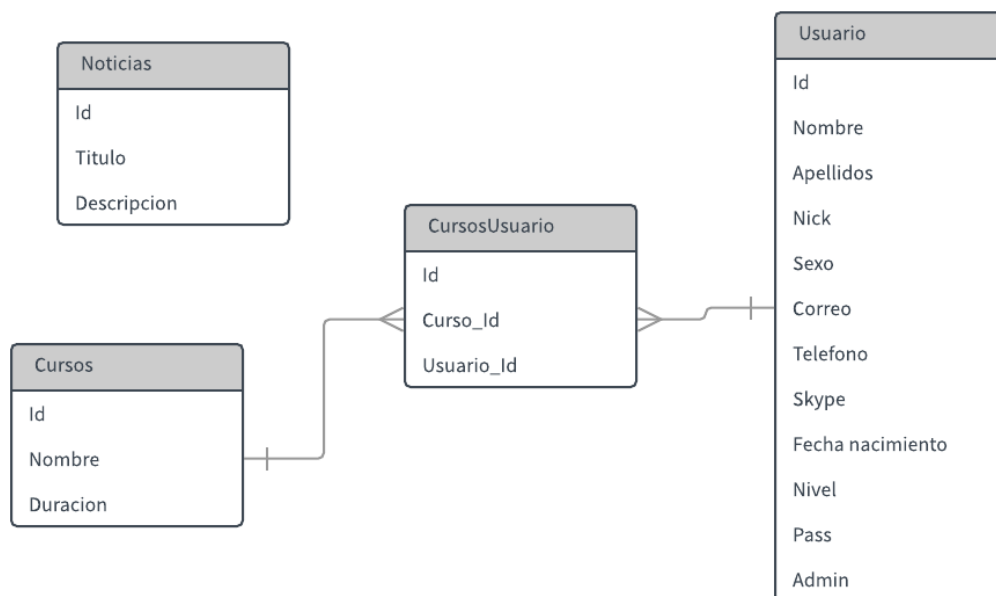
GitHub: <https://github.com/vhar95/XamarinApp>

En cuanto al entorno de trabajo, utilizaremos el entorno de desarrollo *Visual Studio 2017*, el cual ya incorpora la integración de versiones de *Git*, por lo que se agilizará nuestro trabajo.

Para la realización de pruebas de nuestra aplicación, utilizaremos un dispositivo móvil Android para las pruebas sobre Android, y dos PCs portátiles (el de cada miembro del grupo) con Sistema Operativo *Windows 10* para las pruebas sobre UWP. En el caso de IOS no hemos realizado ninguna prueba.

## Modelo de datos

Antes de entrar en la explicación de las vistas de nuestra aplicación, mostramos en la siguiente figura el modelo de datos que hemos utilizado para nuestra aplicación.

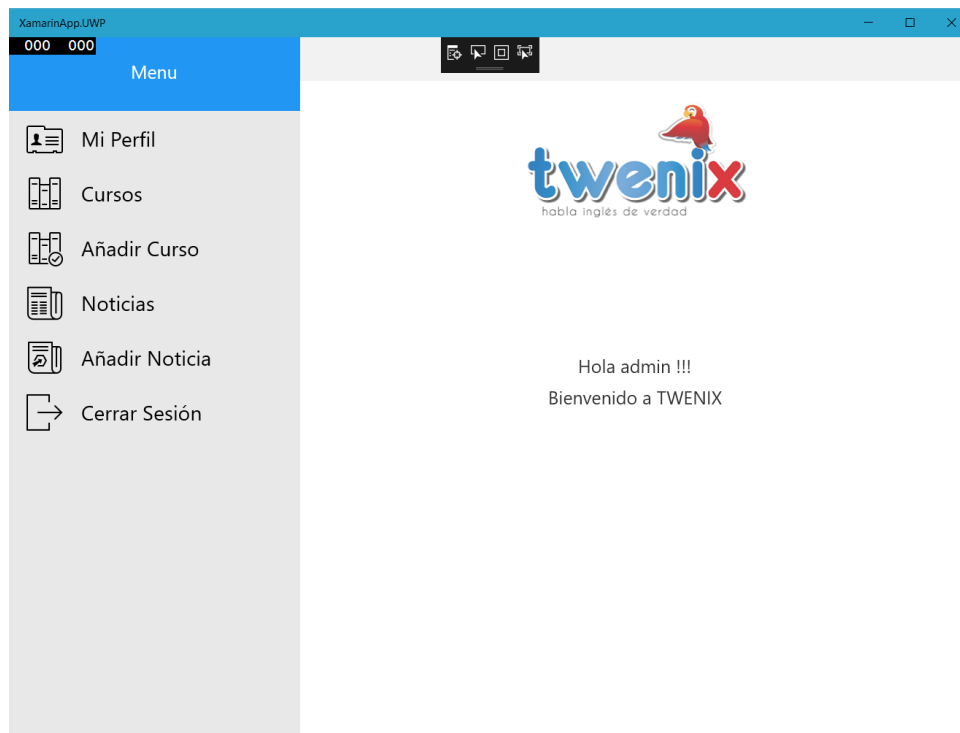


## Vistas

En este apartado definiremos cada una de las vistas que tendrá nuestra aplicación y explicaremos su funcionamiento. Para ello, dividiremos este punto en tres apartados: vistas administrador, vistas usuario y vistas comunes.

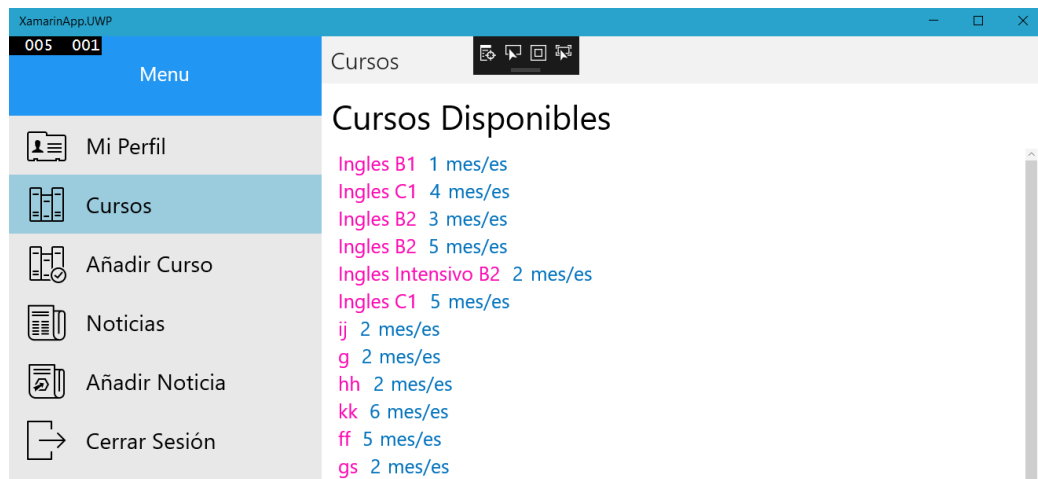
### Vistas Administrador

Tendremos una página principal para el administrador, la cual dispondrá de un menú lateral con diversas opciones.

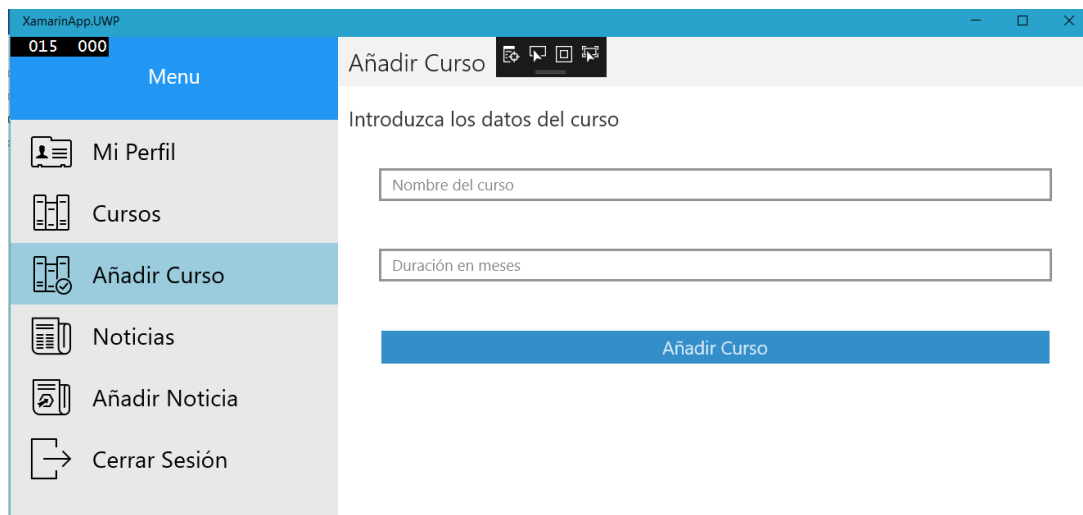


En la opción *Mi Perfil* podremos actualizar los datos del usuario cuyo rol es “administrador”.

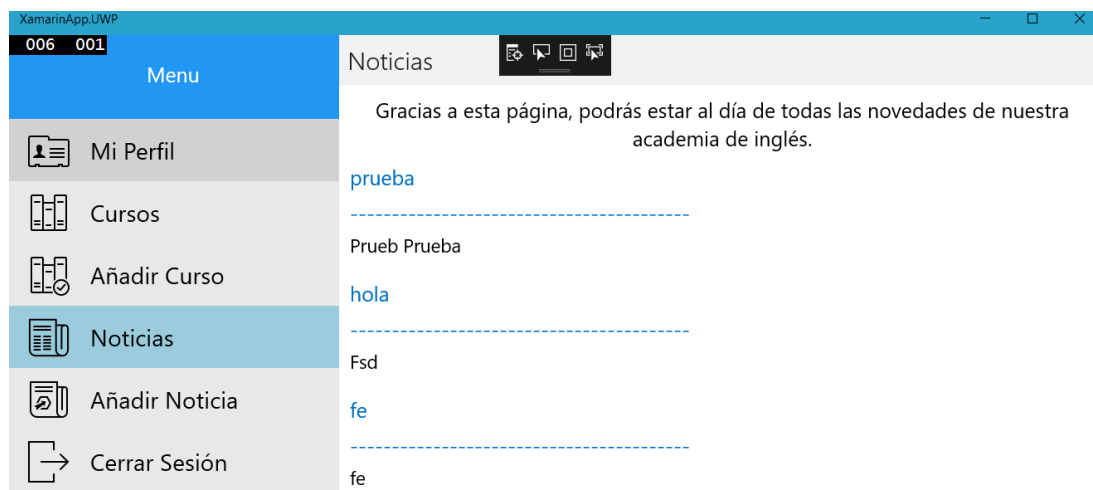
Tendremos también la opción de ver una lista de los cursos disponibles que imparte la academia de inglés.



En la opción *Añadir Curso* tendremos un formulario con el que podremos dar de alta un nuevo curso.



Además, la aplicación dispondrá de una lista con las noticias que el administrador vaya escribiendo.



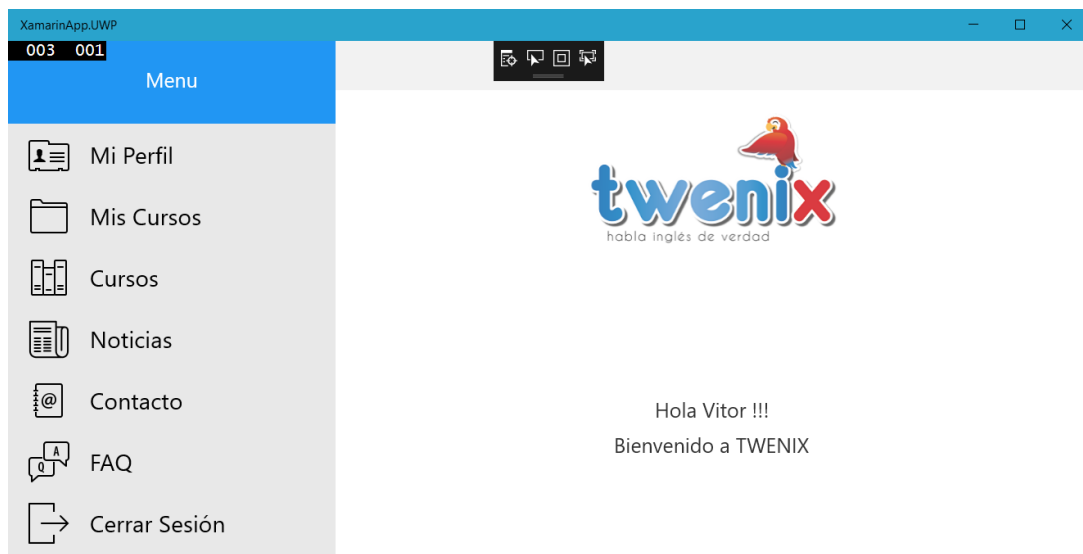


En la opción *Añadir Noticia* tendremos un formulario para dar de alta una nueva noticia.

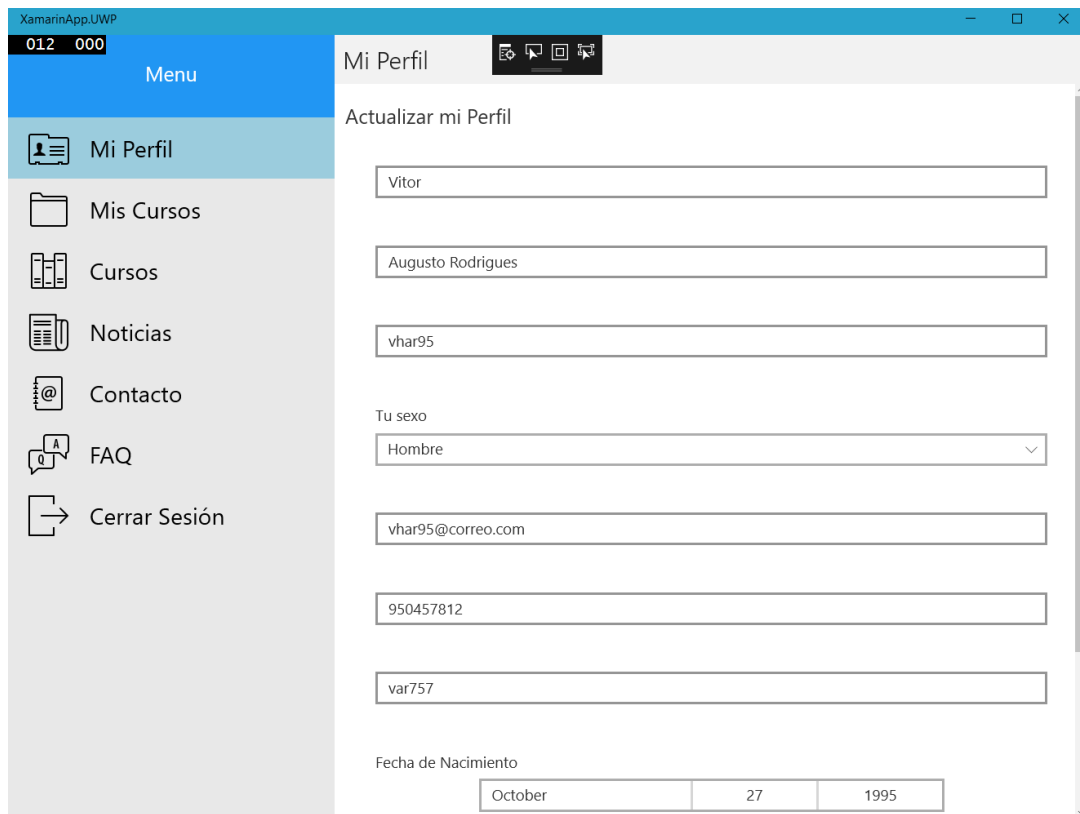
The screenshot shows the 'Añadir Noticia' (Add News) screen of the XamarinApp.UWP. On the left is a blue sidebar menu with the following options: 'Mi Perfil', 'Cursos', 'Añadir Curso', 'Noticias', 'Añadir Noticia' (highlighted), and 'Cerrar Sesión'. The main content area has a title bar 'Añadir Noticia' and a subtitle 'Introduzca los datos de la noticia'. It contains two input fields: 'Título de la noticia' and 'Descripción'. At the bottom right is a blue button labeled 'Añadir Noticia'.

## Vistas Usuario

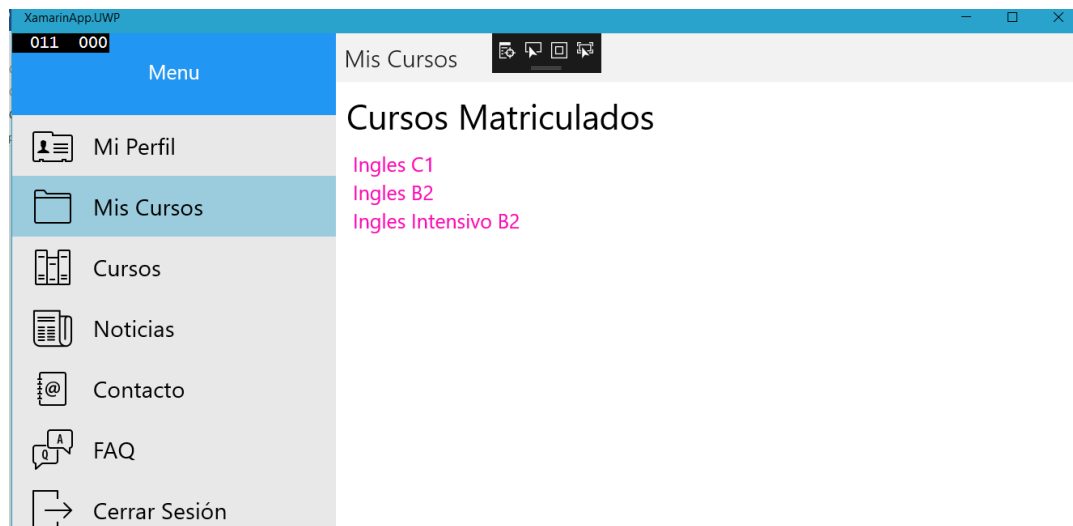
La página principal para el caso de un usuario normal (un alumno) muestra un menú lateral con más opciones.



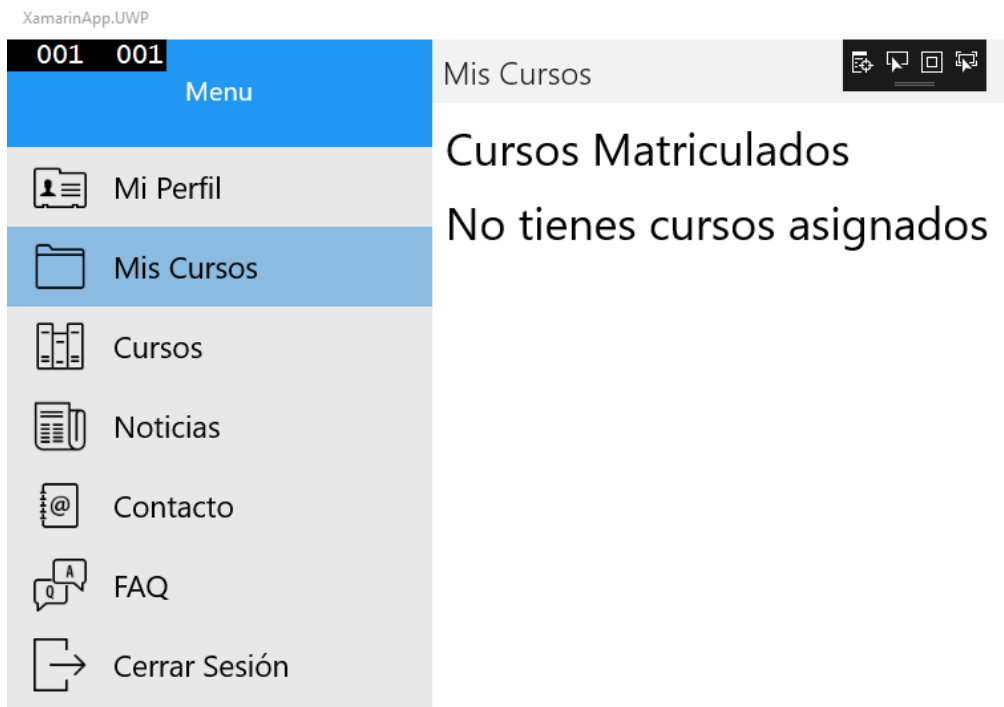
Al igual que en la vista de administrador, en la opción *Mi Perfil* podremos actualizar los datos de un usuario, en este caso, los de un alumno.



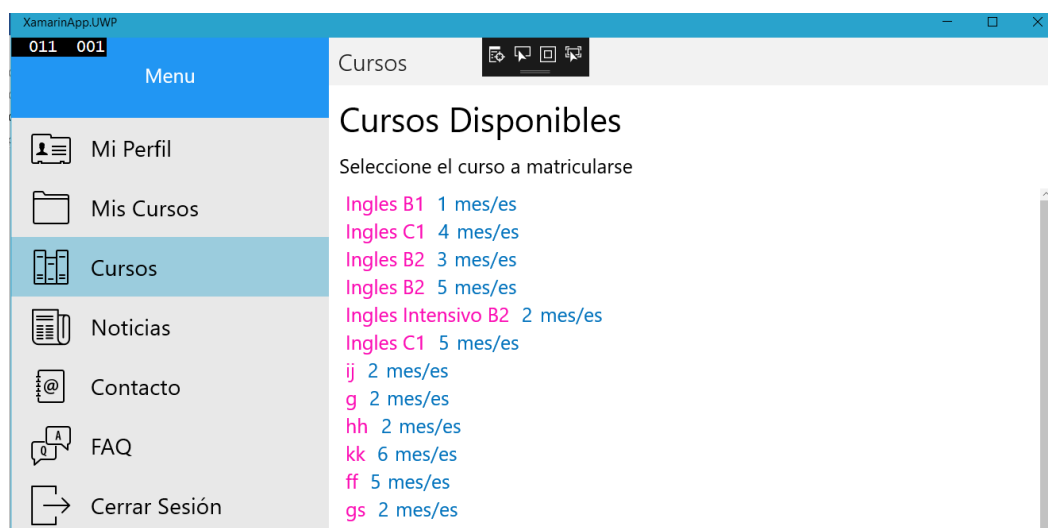
En *Mis Cursos* podemos ver los cursos en los que está matriculado un usuario.



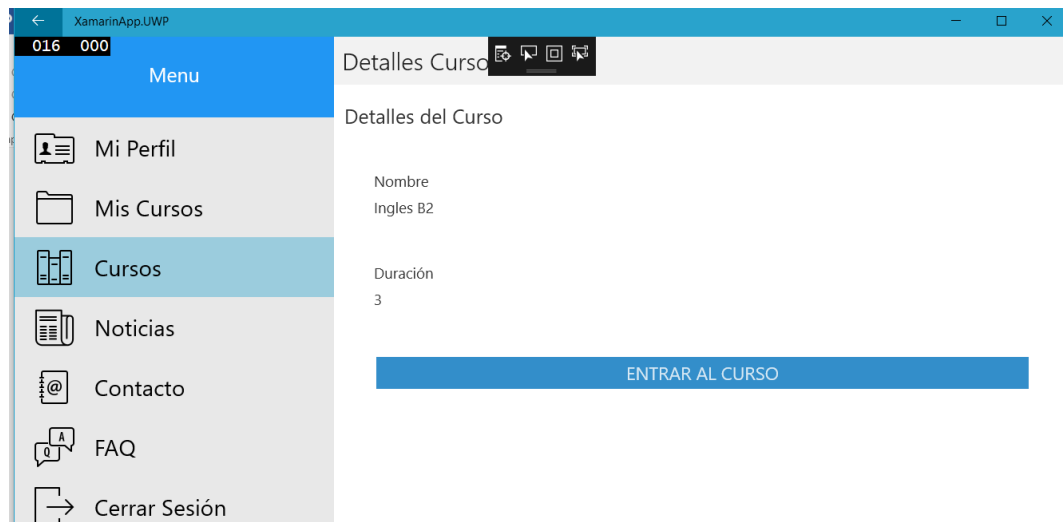
En caso de que el usuario no se haya matriculado todavía en ningún curso, aparecerá un mensaje indicando que dicho usuario no tiene cursos asignados.



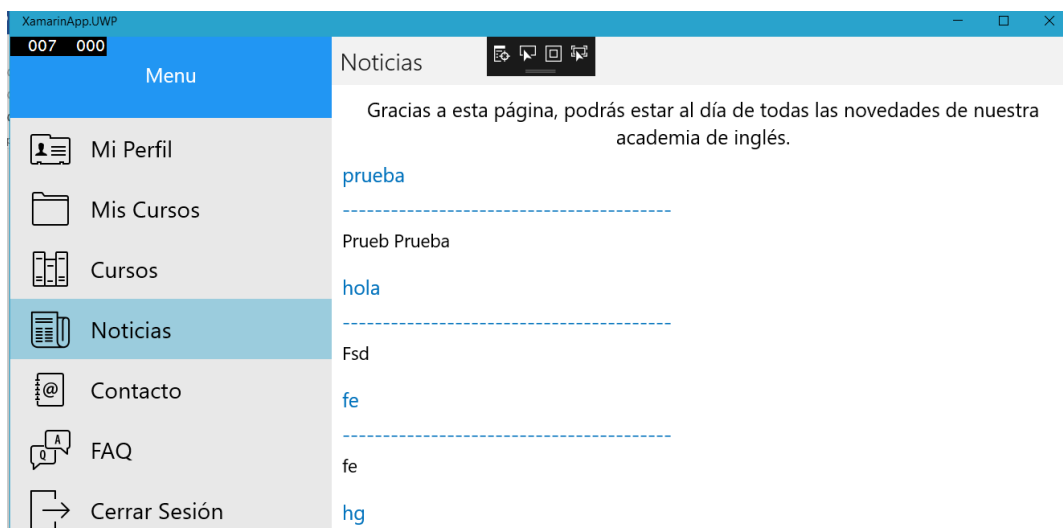
Si accedemos a *Cursos*, veremos una lista de todos los cursos disponibles que imparte la academia. Si seleccionamos un curso, podremos matricularnos en éste.



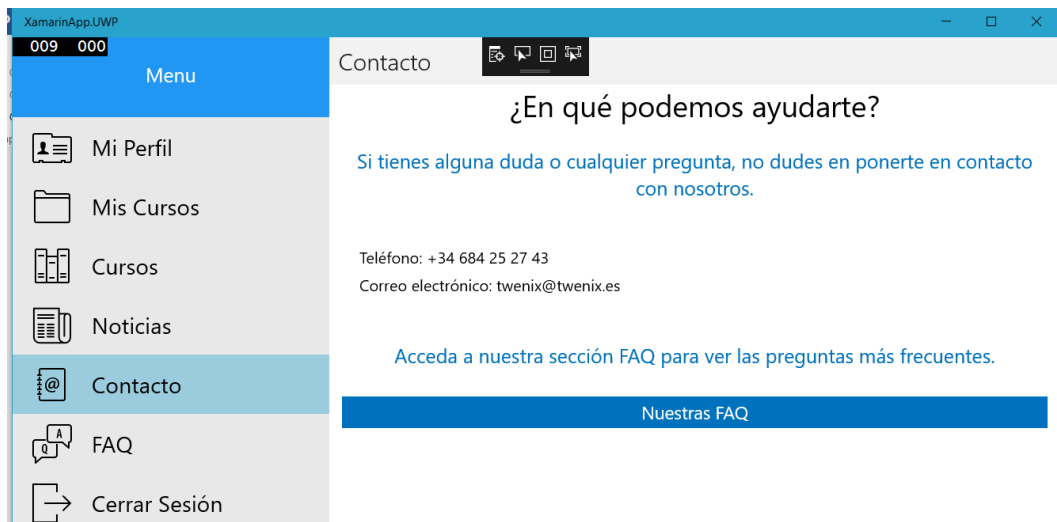
Al seleccionar un curso, se mostrarán los detalles del mismo y nos dará la opción de entrar a dicho curso.



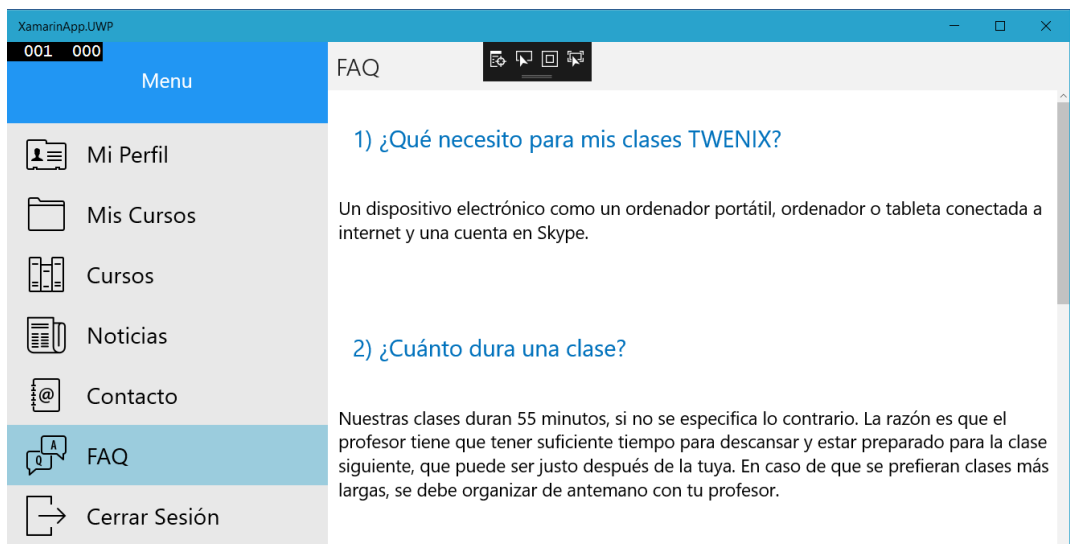
En la opción *Noticias*, veremos una lista con todas las noticias que haya escrito el administrador.



En el apartado *Contacto* dispondremos de los datos de contacto de la academia. Además de un enlace que nos llevará a la sección *FAQ* o de preguntas más frecuentes realizadas por los usuarios.

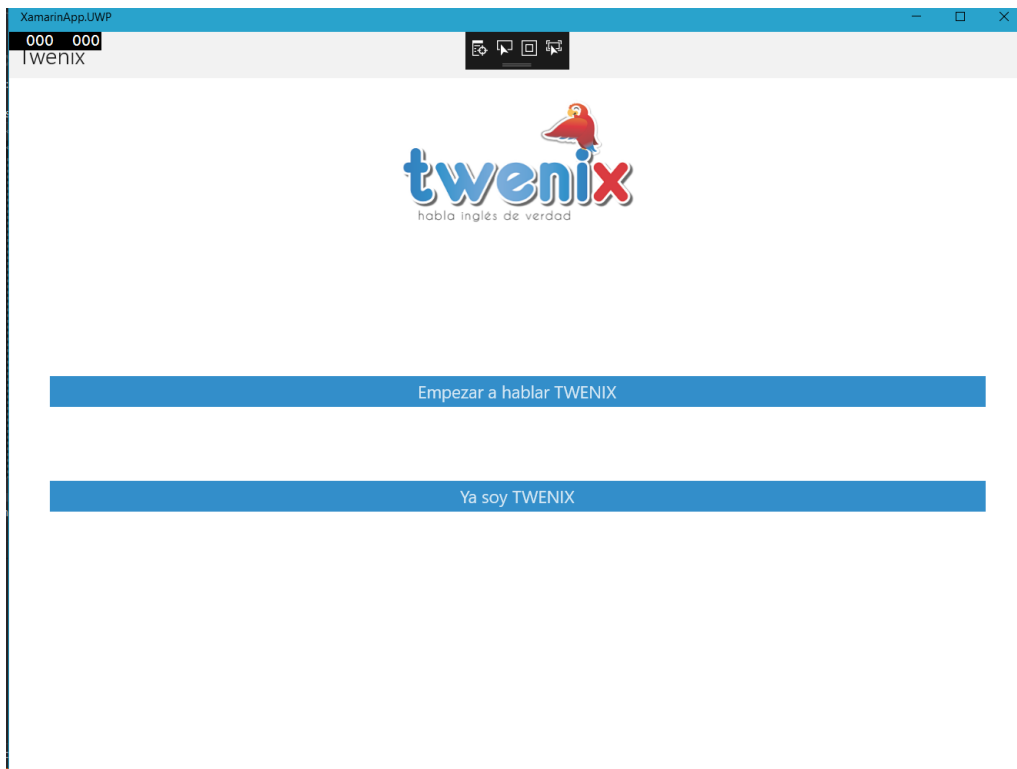


Y otra opción que nos llevará directamente a la propia sección *FAQ* que acabamos de nombrar.



## Vistas Comunes

En la vista principal de la aplicación tendremos dos opciones: una que nos permitirá dar de alta a un usuario en la aplicación (botón “Empezar a hablar TWENIX”), y otra que permitirá que el usuario (ya sea un usuario normal o un administrador) se loguee (botón “Ya soy TWENIX”).



La opción de registro consistirá en un formulario, con todos los campos validados, en el cual introduciremos los datos del nuevo usuario.

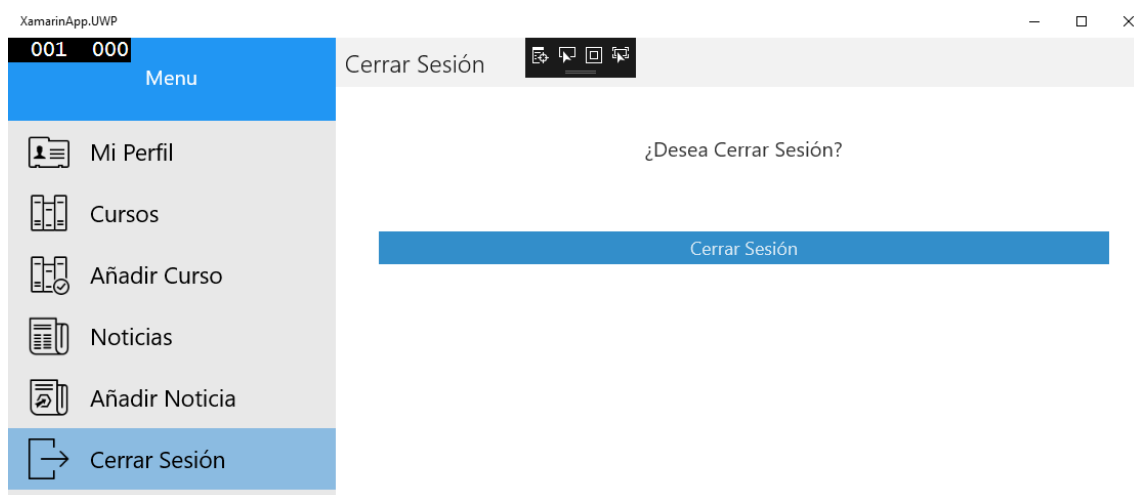
The screenshot shows the registration form in the TWENIX app. The title is 'Introduce tus datos'. The form contains several input fields: 'Tu nombre', 'Tus apellidos', 'Tu Nick de Usuario', 'Tu sexo' (a dropdown menu), 'Tu correo electrónico', 'Tu contraseña', 'Tu teléfono', and 'Tu nombre de Usuario de Skype'. A vertical scrollbar is visible on the right side of the form.

En la opción de logueo, deberemos introducir el correo electrónico y la contraseña para que se identifique el usuario, y así poder acceder a su página principal.



The screenshot shows the login interface of the XamarinApp.UWP application. At the top, there is a header bar with a back arrow, the text "XamarinApp.UWP", and a status bar showing "004 000" and "Iwenix". Below the header, the main content area has a title "Introduce tu correo y contraseña para que podamos identificarte". There are two input fields: "Tu correo" and "Tu contraseña". Below these fields is a blue button labeled "LOGIN".

Por último, tanto el usuario normal como el administrador tendrán una opción en el menú lateral para poder cerrar sesión.



The screenshot shows the application interface with a side menu open. The menu is titled "Menu" and contains several options: "Mi Perfil", "Cursos", "Añadir Curso", "Noticias", "Añadir Noticia", and "Cerrar Sesión". The "Cerrar Sesión" option is highlighted. To the right of the menu, there is a confirmation dialog titled "¿Desea Cerrar Sesión?" with a blue button labeled "Cerrar Sesión".

Eligiendo esta opción y pulsando el botón "Cerrar Sesión", la aplicación nos dirigirá a la vista principal de la misma.



Empezar a hablar TWENIX

Ya soy TWENIX