

Automatización de la gestión de la piscina de la UCM mediante Pista Virtual

Trabajo de Fin de Grado en Grado en Ingeniería del Software,
Grado en Ingeniería Informática de la Facultad de Informática,
Universidad Complutense de Madrid



Autores:

Carlos Martín Gutiérrez.

Patricia Ortiz Zamora.

Director: Adrián Riesco

Índice

Resumen.....	4
Summary	5
Agradecimientos	6
Acknowledgements.....	7
1. Introducción	8
1.1. Motivación	8
1.2. Objetivos	8
1.3. Plan de trabajo	9
1.4. Estructura de la memoria.....	9
2. Introduction	10
2.1. Motivation.....	10
2.2. Objectives	11
2.3. Workplan.....	11
2.4. Memory structure	12
3. Preliminares	12
3.1. Pista Virtual	12
3.2. Tecnologías.....	13
3.2.1. .NET	13
3.2.2. Xamarin	13
3.3. Visual Studio.....	14
3.4. Licencias	14
4. Requisitos.....	15
4.1. Requisitos Iniciales	15
4.2. Requisitos Posteriores.....	17
4. Implementación	19
4.1. Diseño.....	19
4.2. Explicación de la implementación.....	20
5. Manual de Usuario	27
5.1. Aplicación Gestor	28
5.2. Aplicación usuario	30
6. Contribución de los alumnos.....	35
6.1. Carlos.....	35
6.2. Patricia.....	37
7. Conclusiones y trabajo futuro	38
7.1. Conclusiones.....	39

7.2. Trabajo futuro	40
8. Conclusions and Future work.....	40
8.1. Conclusions	41
8.2. Future Work	42
Bibliografía	43

Resumen

A pesar de la modernización de distintas áreas de las zonas deportivas de la Universidad Complutense de Madrid, no se ha modernizado la gestión de los accesos y reservas a las instalaciones. Sin embargo, en los últimos años se ha hecho un esfuerzo para que estas acciones puedan llevarse a cabo online a través de una aplicación móvil llamada Pista Virtual.

En este contexto, en este trabajo de fin de grado presentamos una extensión de Pista Virtual a la gestión de las piscinas.

La aplicación tiene varios módulos, nosotros hemos realizado la parte de la piscina de verano. En un principio iban a ser ambas piscinas, pero finalmente de la piscina de invierno no recibimos ningún requisito por lo que solo hemos implantado la piscina de verano.

El módulo piscina está implementado con las tecnologías .NET y Xamarin, nuevas para nosotros en un primer momento. A día de hoy entendemos por qué Pista Virtual decidió desde un primer momento desarrollar la aplicación en C#, ya que el desarrollo multiplataforma da muchas facilidades para los distintos sistemas operativos (Android, iOS) que hoy en día son mayoritariamente utilizados.

El resultado que hemos tenido de este Trabajo de Fin de Grado, es el módulo integrado en la aplicación. En él podemos tener dos roles, el de usuario o el de gestor. Para los usuarios la aplicación tiene una sección llamada *Piscina* en la que puedes comprar entradas para el acceso a la misma, la entrada se genera mediante un código BIDI que permite acceso al usuario para el uso y disfrute de la piscina. Para los gestores la aplicación tiene una sección llamada *Entradas* en la que seleccionan el día y les permiten escanear los códigos BIDI y así poder dar acceso a los distintos usuarios a la piscina.

Palabras clave: Pista Virtual, Piscina UCM, .NET, Xamarin, C#, Android, iOS, usuario, gestor, acceso, Código BIDI.

Summary

Despite the modernization of different areas of the sports areas of the Complutense University of Madrid, the management of access and booking to the facilities have not been modernized. However, in recent years an effort has been made so that these actions can be carried out online through a mobile application called Pista Virtual.

In this context, in this end-of-degree project we present an extension of Pista Virtual to the management of swimming pools.

The application has several modules, we have made part of the summer pool. At first they were going to be both pools, but at the end we did not receive any requirement about the Winter pool so we only implemented the summer pool.

The pool module is implemented with .NET and Xamarin technologies, new for us at first. Today we understand why Pista Virtual decided from the first moment to develop the application in C#, since the multiplatform development gives many facilities for the different operating systems (Android, iOS) that today are mostly used.

The result we have had from this Final Degree Project is the module integrated in the application. We can find two roles in it, the user and the manager. For the users, the application has a section called "Pool" where you can buy tickets to Access, the entry is generated by a BIDI code that allows users to enter the pool. For managers the application has a section called "Tickets" in which the users select the day and allow them to scan the BIDI codes and give access to the different users to the pool.

Keywords: Pista Virtual, Pool UCM, .NET, Xamarin, C #, Android, iOS, user, manager, access, BIDI Code.

Agradecimientos

En primer lugar queremos agradecer a nuestro tutor, Adrián Riesco, el haber aceptado nuestro TFG, el haber estado todo el curso con nosotros ayudando y tranquilizándonos en los momentos de mayor agobio. También queremos dar las gracias al equipo de Pista Virtual, que siempre nos han recibido con los brazos abiertos en sus oficinas, y han estado disponibles para ayudarnos cuando lo hemos necesitado, en especial queríamos agradecerse a Rafael, que ha sido el que más tiempo ha estado con nosotros ayudándonos y explicándonos las distintas tecnologías y resolviéndonos cada duda que teníamos. Nos sentimos verdaderamente realizados tras haber aprendido una tecnología nueva, que sin su ayuda no hubiera sido posible.

No nos queremos olvidar de nuestra familia y amigos que nos han estado apoyando y motivando a lo largo de todo el curso, siempre con una sonrisa y dándonos ánimos para conseguir sacar este proyecto adelante.

Acknowledgements

First of all we want to thank our tutor, Adrián Riesco, to accepting our TFG, for being the whole course with us helping us and reassuring us in the moments of greatest load of work. We also want to thank the Pista Virtual team, who have always welcomed us with open arms in their offices, and willing to help us when we needed it, especially we wanted to thank Rafael, who has been the guy who has helped us the most , explaining the different technologies and resolving every doubt we had. We feel truly fulfilled after having learned a new technology, which without their help would not have been possible.

We do not want to forget about our family and friends who have been supporting and motivating us throughout the course, always with a smile and giving us encouragement to get this project forward

1. Introducción

En esta sección de este Trabajo de Fin de Grado presentamos la motivación, los objetivos y el plan de trabajo. Por último, describimos la organización del resto de la memoria.

La gestión del acceso a las zonas deportivas en la UCM se quiere automatizar poco a poco. Esta automatización se está realizando mediante una aplicación móvil llamada Pista Virtual. En particular, en las piscinas, tienen una gestión manual que ralentiza el acceso a los usuarios y dificulta la gestión a los administradores. En este trabajo proponemos una extensión de Pista Virtual que consiste en la entrada a la piscina de verano mediante la aplicación.

El acceso a la piscina de verano de la Complutense que ha habido hasta ahora ha sido mediante la compra de abonos en papel o de entradas individuales en la puerta de acceso. Para poder pasar solo tenías que presentar un carnet universitario y pagar en la puerta o dar el bono. Además, el aforo era difícil de controlar ya que no había un registro de ventas diarias.

Introducir la piscina en Pista Virtual supone una ayuda para poder controlar el aforo, ya que la aplicación no dejará reservar más entradas de las permitidas. Impedir que con un mismo abono de la Complutense pasen dos o más personas, ya que un abono lo puedes utilizar una vez al día. Tener controlado el acceso de usuarios tanto que pertenecen a la Complutense como los que no y que los usuarios que accedan conozcan las normas para poder disfrutar de la piscina.

El módulo piscina en Pista Virtual ayuda a un avance tecnológico en el acceso a la piscina ya que con solo llevar el móvil puedes acceder y pagar en el momento.

1.1. Motivación

Ayudar a la Complutense a mejorar su sistema de acceso a las piscinas fue una de las principales motivaciones de este proyecto. Cuando conocimos a Pista Virtual y nos ofreció realizar este proyecto no pudimos decir que no ya que saber que nuestra aplicación va a utilizarse y va a estar implantada en el sistema de la UCM de carácter más o menos inmediato nos ha servido como motivación para hacer el proyecto y trabajar con más ganas.

Otro motivo por el cuál elegimos colaborar con ellos fue aprender un lenguaje nuevo como es .NET y Xamarin.Forms, ya que es algo que a lo largo de la carrera no hemos utilizado y gracias a que Pista Virtual está implementada con esta tecnología hemos podido aprenderla y formarnos en un ámbito que hasta ahora no habíamos conocido.

Introducir Pista Virtual en la universidad supone un avance tecnológico ya que tras años de colas y esperas para acceder a cualquier instalación deportiva, gracias a esta aplicación solo necesitas tenerla en tu móvil para poder reservar el acceso a cualquiera de ellas. Por esto mismo elegimos una colaboración con ellos para nuestro TFG.

1.2. Objetivos

El objetivo principal de este TFG es crear una herramienta que permita el acceso a las piscinas de la Complutense de una manera más cómoda para los usuarios y mejorar la gestión de la piscina.

Este proyecto tiene como objetivo la realización del módulo para la automatización de la piscina, es decir, el acceso a la piscina mediante la aplicación. El proyecto está dividido en dos áreas independientes, una para el trabajador de la UCM que consiste en la lectura de un código BIDI para permitir el acceso a la piscina, y otra área para los usuarios (empleados, estudiantes, alumni) que consiste en la compra de entradas o bonos para acceder a la piscina mediante la generación de un código BIDI.

La organización del trabajo se ha llevado a cabo como sigue:

- Investigación y aprendizaje sobre la tecnología y del lenguaje que íbamos a utilizar.
- Instalación del módulo de .NET en Visual Studio.
- Realizar las primeras pantallas y pruebas para comenzar a familiarizarnos con el lenguaje.
- Desarrollo de la aplicación previa a los requisitos.
- Introducción de los requisitos en nuestra aplicación.
- Pruebas de funcionamiento y arreglo de los errores.
- Adaptación de nuestras interfaces a las de Pista Virtual utilizando los estilos que nos dieron.
- Integración del Módulo Piscina en la aplicación de Pista Virtual.

1.3. Plan de trabajo

Durante los primeros meses nuestro principal trabajo fue familiarizarnos con el lenguaje y con la tecnología que hemos estado usando. Hicimos pequeñas interfaces que se conectaban las unas con las otras y con un funcionamiento mínimo.

Comenzamos realizando un pequeño menú como el que podemos ver en nuestro código, del que poco a poco fue surgiendo nuestro modulo. Fuimos añadiendo funcionalidad a medida que fuimos visitando a Pista Virtual y aprendiendo de los errores que íbamos cometiendo para no tenerlos más adelante.

A finales de marzo La Complutense nos dio sus requisitos, los añadimos en nuestra aplicación y comenzamos a ir a Pista Virtual con bastante más frecuencia para que nos indicaran como querían las interfaces y nos solucionaran alguna duda que íbamos teniendo respecto a los requisitos, ya que ellos han sido los que se han comunicado con la Complutense en todo momento.

La semana previa a Semana Santa, Pista Virtual nos dio sus interfaces y adaptamos nuestra aplicación a los diseños que ellos nos pidieron.

En Semana Santa hicimos la entrega del código al equipo de Pista Virtual para que hicieran la integración en la aplicación y la Complutense pudiera probar el código y los requisitos con tranquilidad de cara a la apertura de la piscina de verano.

Después de Semana Santa escribimos la memoria.

1.4. Estructura de la memoria.

El resto de la memoria se estructura de la siguiente forma:

- En la sección 2 se explican las tecnologías utilizadas y qué es Pista Virtual.
- En la sección 3 se detallan los requisitos que hemos seguido durante la realización del proyecto.
- En la sección 4 se expone la implementación del módulo y cómo está formado.
- En la sección 5 ofrecemos un manual de usuario.
- En la sección 6 hablamos de las contribuciones al proyecto de cada uno de nosotros.
- En la sección 7 se muestran las conclusiones del proyecto y el trabajo futuro.

El código de la aplicación está disponible en:

<https://github.com/patrioz/Modulo-PiscinaTFG/tree/master/ModuloPiscina>

2. Introduction

In this section of this Degree Thesis we explain the motivation, the objectives and the work plan. Finally, we present the organization of the rest of the memory.

The management of the sports areas in the UCM is being automated little by little. It is being done through a mobile application called Pista Virtual. In the pools, they have a manual management that slows down access to users and makes it difficult for administrators to manage. In this work we propose an extension of Pista Virtual consisting of the entrance to the summer pool through the application.

The access to the summer pool of the Complutense that has been used so far, it has been managed through the purchase of paper tickets or individual tickets at the gate. In order to pass you only had to present a university card and pay at the door or hand out the voucher. In addition, the capacity was difficult to control since there was no record of daily sales.

Introducing the pool in Pista Virtual helps to control the capacity, since the application will not book more tickets than allowed. It also prevents people from using the same pass, as a pass can be used only once a day. To control the access of users both belonging to the Complutense and those who do not and that the users who access are in knowledge the rules to enjoy the pool.

The Pista Virtual pool module helps a technological breakthrough in the access to the pool since you can access and pay at the moment by just carrying the mobile phone.

2.1. Motivation

Helping the Complutense to improve its access system to the swimming pools was one of the main motivations of this project. When we met Pista Virtual and offered to carry out this project we could not say no because knowing that our application will be used and will be implemented in the UCM system of more or less immediate nature has served us as a motivation to fulfill the project and work with more desire.

Another reason why we chose to collaborate with them was to learn a new language such as .NET and Xamarin.Forms, since it is something that we have not used throughout the degree and thanks to the fact that Pista Virtual is implemented with this technology, we have been able to learn it and train us in a field that until now we had not known.

2.2. Objectives

The main objective of this TFG is to create a tool that allows access to the swimming pools of the Complutense in a more comfortable way for users and improve the management of the pool.

The objective of this project is to create the module for the automation of the pool, that is to say, access to the pool through the application. The project is divided into two independent areas, one for the staff of the UCM that consists of reading a BIDI code to allow access to the pool, and another area for users (employees, students, alumni) consisting of the purchase of tickets or bonuses to access the pool by generating a BIDI code.

The organization of work has been carried out as follows:

- Research and learning about the technology and the language that we would use.
- Installation of the .NET module in Visual Studio.
- Make the first screens and tests to begin to familiarize ourselves with the language.
- Development of the application prior to the requirements.
- Introduction of the requirements in our application.
- Tests of operation and arrangement of errors.
- Adaptation of our interfaces to Pista Virtual using the visual styles they gave us.
- Integration of the Pool Module in the Pista Virtual application.

2.3. Workplan

Among the first months our main job was to familiarize ourselves with the language and with the technology that we have been using. We made small interfaces that were connected to each other and with minimal operability.

We started by making a small menu like the one we can see in our code, from which our module was gradually growing. We were adding functionality as we were visiting Pista Virtual and learning from the mistakes we were making so we will not make them again.

At the end of March, Complutense gave us their requirements and started to implement our application, and at this moment we started to go to Pista Virtual a lot more frequently so they could indicate us how they wanted the interfaces and solve any doubts that we were having regarding the requirements, since they have been the ones who have communicated with the Complutense at all times.

The week before Easter, Pista Virtual gave us the interfaces and we adapted our application to the designs they asked us for.

In Easter we sent Pista Virtual the code so they could do the integration in the application and the Complutense could try the code and the requisites with tranquility facing the opening of the summer pool.

After Easter we wrote the memory.

2.4. Memory structure

The rest of the memory is structured as follows:

- Section 2 explains the technologies used and what is Virtual Track
- Section 3 details the requirements that we have followed during the implementation of the project.
- Section 4 explains the implementation of the module and how it is formed.
- In Section 5 we present the user manual.
- In section 6 we discuss the contributions to each of our projects.
- Section 7 concludes and presents some issues of future work.

The code of the application is available in:

<https://github.com/patrioz/Modulo-PiscinaTFG/tree/master/ModuloPiscina>

3. Preliminares

En esta sección presentamos a Pista Virtual y las tecnologías utilizadas en el proyecto. En particular el desarrollo de la automatización de la gestión de las piscinas se ha realizado con .NET y Xamarin.Forms, que detallamos a continuación.

3.1. Pista Virtual

Pista Virtual es un equipo de desarrolladores que cuando estaban acabando la carrera se les ocurrió, como a nosotros, realizar como TFG una aplicación para la gestión polideportiva ya fuese de una universidad o de un grupo de centros deportivos. Una vez presentado su TFG decidieron seguir con él y dar un paso más, se lo ofrecieron a la Complutense y llevan dos años intentando implantar su aplicación en la universidad.

Nosotros tuvimos una idea muy similar a la de ellos, cuando se la propusimos a nuestro tutor nos recomendó hablar con el equipo de deportes, y fueron ellos los que nos llevaron hasta el equipo de Pista Virtual quienes sin ningún problema nos enseñaron su aplicación y nos ofrecieron realizar el módulo de la piscina para posteriormente poder implantarlo en su aplicación.

Desde un primer momento el equipo de Pista Virtual nos ofreció toda la ayuda que estaba en sus manos, nos ofrecieron ir a visitarles siempre que lo necesitásemos y un lugar donde poder trabajar, nos ayudaron a dar los primeros pasos en la aplicación explicándonos el funcionamiento de la plataforma y del lenguaje, nos enseñaron como trabajaban ellos y como ir haciéndonos a las nuevas tecnologías que hemos estado utilizando.

Sin conocerles mucho, desde la primera reunión que tuvimos con ellos hace ya más de un año nos transmitieron confianza y supimos que hacer el proyecto con ellos sería una buena idea, siempre que hemos tenido algún problema o hemos necesitado de su ayuda han estado a nuestra disposición sin importarles perder una tarde entera con nosotros explicándonos el funcionamiento de sus módulos o de ideas de cómo realizar el nuestro.

La piscina de verano va a ser el primer módulo que pongan en funcionamiento. Considerando que después de dos años de reuniones con la universidad para poder implantar la aplicación lo

primero que la Complutense haya decidido desplegar sea nuestro módulo nos motiva bastante, ya que es un módulo que ha salido adelante en un curso académico y que creemos que va a ser muy útil y cómodo tanto para los trabajadores como para los usuarios. Para los trabajadores porque con un simple sonido sabrán qué tipo de usuario está accediendo a la piscina y saber si la entrada es buena o mala. Y para los usuarios porque solo necesitan una aplicación en el teléfono para poder acceder a la piscina.

Con este módulo conseguimos mejorar el acceso que ha habido hasta ahora que era mediante la compra de abonos en papel o de entradas individuales en la puerta de acceso. Para poder pasar tenías que presentar un carnet universitario y pagar en la puerta o dar el bono.

3.2. Tecnologías

Como tecnologías utilizadas durante el desarrollo del proyecto has sido .NET y Xamarin.Forms, ya que es la tecnología en la que el equipo de Pista Virtual tiene implementada su aplicación.

3.2.1. .NET

.NET es una tecnología para el desarrollo de software multiplataforma, usada para que la aplicación tenga el mismo soporte ya sea para un sistema operativo u otro. Esto nos ha facilitado mucho las cosas, ya que lo que hemos tenido que programar para Android y para IOS ha sido mínimo.

Desde el punto de vista tecnológico, .NET desarrolla aplicaciones y sistemas que son independientes de la arquitectura física y del sistema operativo sobre el que se ejecuta, gracias a esto se pueden realizar en distintos ámbitos, en nuestro caso en una aplicación móvil.

Una de las principales ventajas de esta tecnología es el desarrollo multiplataforma que hace que la implementación se pueda utilizar en los distintos sistemas operativos de los teléfonos móviles. Gracias a este motivo apenas hemos tenido que programar en Android y en iOS. El lenguaje utilizado para el desarrollo del proyecto ha sido C#.

El entorno de trabajo que proporciona .NET es un motor de ejecución que controla las aplicaciones en ejecución y la biblioteca de clases.

Los componentes principales de .NET [1] son:

- El Entorno Común de Ejecución para Lenguajes (CLR): Entorno de ejecución proporcionado por .NET Framework que ejecuta el código y proporciona servicios que facilitan el proceso de desarrollo.
- La Biblioteca de Clases Base (BCL): La biblioteca de clases está orientada a objetos, lo que proporciona tipos que son fáciles de usar y reduce el tiempo asociado con el aprendizaje de nuevas características de .NET.
- Conjunto de lenguajes de programación: soporta ya más de veinte lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos. Algunos de los lenguajes desarrollados son: C# (Utilizado en el proyecto), Visual Basic, C++, J#, Perl Python, Fortan y Cobol.NET.

3.2.2. Xamarin

Xamarin es un entorno unificado, ya que ofrece un lenguaje, una biblioteca de clases y un entorno de ejecución que funciona en múltiples plataformas. En este sentido, Xamarin es único

porque combina la potencia de las plataformas nativas y agrega una serie de características propias muy eficaces [2]:

- Tiene funciones para invocar directamente a la biblioteca.
- Utiliza C# que incluye las mejoras de Objective-C y Java.
- Usa Visual Studio como entorno de desarrollo, que es compatible para Mac en Mac OS X y Visual Studio en Windows, nosotros hemos utilizado ambos programas para el desarrollo del proyecto.
- Ofrece compatibilidad multiplataforma móvil, ofrece una compatibilidad multiplataforma sofisticada con las tres principales plataformas móviles: iOS, Android y Windows Phone. Es posible escribir aplicaciones de modo que compartan hasta el 90% del código, como es en el caso del proyecto. La biblioteca Xamarin.Mobile ofrece una API unificada para tener acceso a los recursos comunes de las tres plataformas. Nosotros hemos desarrollado toda la aplicación en C# a excepción de los permisos de cámara que los hemos tenido que poner tanto en Android como en iOS.

El funcionamiento de Xamarin nos muestra dos productos, Xamarin.iOS y Xamarin.Android. Ambos se basan en una versión de código abierto del entorno de trabajo de .NET que tiene en cuenta todas las normas de esta tecnología

3.2.2.1. Xamarin.Forms

Representa todas las herramientas de interfaz de usuario multiplataforma para desarrolladores de .NET. Puedes compilar las aplicaciones en los distintos sistemas operativos, como ya hemos citado anteriormente, con C# en Visual Studio. Esta versión de código abierto puede ejecutarse en prácticamente todas las plataformas [3].

Las aplicaciones Xamarin usan un tiempo de ejecución que controla todo automáticamente. Cuando se compilan aplicaciones Xamarin, el resultado es un paquete de aplicación, ya sea un archivo para iOS o Android.

El proyecto entero ha sido desarrollado con Xamarin.Forms y con las bibliotecas que proporciona.

3.3. Visual Studio

La herramienta seleccionada para el desarrollo del proyecto ha sido Visual Studio, ya que el equipo de Pista Virtual es el programa que utiliza y en que Xamarin.Forms proporciona todas sus bibliotecas y con C# puedes ejecutar la aplicación en los distintos sistemas operativos para las aplicaciones móviles en sus simuladores.

No nos ha sido difícil familiarizarnos con él ya que en la carrera desde primer curso nos han hecho trabajar con esta herramienta.

3.4. Licencias

Las licencias de Software empleadas en este proyecto han sido las empleadas por Pista Virtual, utilizan una licencia con software propietario[4], es decir no existe forma de acceso al código fuente de la aplicación, y no se permite su libre modificación, adaptación o incluso lectura de terceros. En este software el autor, en nuestro caso, Pista Virtual, no transmite ninguno de los derechos, ha establecido las condiciones con nosotros de que el software puede ser utilizado,

limitando los derechos de copia, modificación, cesión o redistribución, y nos ha especificado que el producto pertenece al propietario, y nos ha concedido el privilegio de utilizarlo[5] .

4. Requisitos

A continuación, en esta sección vamos a explicar los requisitos que la Universidad Complutense de Madrid nos indicó para la realización de la aplicación.

4.1. Requisitos Iniciales

Al inicio del verano de 2018, junto con un asesor del Vicerrectorado de Deportes de la Universidad Complutense, nuestro tutor y con el equipo de Pista Virtual, nos acercamos a las instalaciones de la piscina de verano de la universidad para ver personalmente cómo funcionaba el sistema de entrada y tener en cuenta ciertos requisitos iniciales.

Tras reunirnos con el personal encargado, definimos los siguientes requisitos:

- **Aforo:** El aforo de la piscina está limitado a 1500 personas. Esto es un requisito importante ya que el sistema de conteo actual es manual, es decir, cada vez que una persona accede a las instalaciones el personal apunta en un papel el número de personas que han accedido.

Cabe decir que no se controlan las personas que salen de las instalaciones, simplemente se lleva un control visual aproximado del aforo en cierto momento, generando esto una incongruencia con respecto al conteo de entrada y un descontrol que puede generar grandes problemas.

- **Roles:** Los usuarios con acceso a las instalaciones deben ser mayores de 18 años y deben tener uno de los siguientes roles: Estudiantes UCM, Estudiante NO UCM, Alumni y Trabajador. Estos roles limitan el acceso a cualquier persona ajena a la Universidad Complutense y/o que no esté cursando estudios universitarios.
 - **Estudiantes UCM:** Cualquier estudiante de la Universidad Complutense con su correspondiente carné universitario en vigor y disponible para ser presentado ante el personal de las instalaciones.
 - **Estudiantes NO UCM:** Cualquier estudiante de universidades de España con su correspondiente carné universitario en vigor y disponible para ser presentado ante el personal de las instalaciones.
 - **Alumni:** Cualquier persona que se haya graduado por la Universidad Complutense y en capacidad de demostrarlo.
 - **Trabajador:** Cualquier empleado de la Universidad Complutense con su correspondiente acreditación y disponible para ser presentado ante el personal de las instalaciones.

Estos roles cobran especial importancia de cara al aforo y al tipo de público permitido, también en relación a la implementación del aplicativo siendo unos de los requisitos con mayor peso.

- **Tipo de Entrada:** Teniendo en cuenta los dos requisitos anteriores, especialmente la pertenencia a uno de los roles definidos, podemos definir el tipo de entrada a la piscina el cual solo se puede realizar de tres únicas maneras:
 - **Entrada Individual:** Entrada de un único uso y de validez limitada al día en que se adquiere. Este ticket es comprado en la misma entrada de las instalaciones siempre que el aforo no se vea superado y puedas acreditar tu pertenencia a uno de los roles definidos previamente. Una vez se abandonen las instalaciones, ese ticket no podrá volver a ser utilizado.
 - **Entrada Abono 10:** Abono de 10 entradas individuales y de validez limitada al fin de la temporada de verano. Este abono también es adquirido en la entrada de las instalaciones y quedará limitado tanto al aforo de la piscina como a la acreditación de uno de los roles definidos previamente excluyendo al rol de Alumno NO UCM el cual no tendrá acceso a este tipo de entrada. Una vez se abandonen las instalaciones se deberá gastar otro uso del bono.
 - **Entrada anual:** Tarjeta anual de acceso. Este tipo de entrada también está directamente relacionada con el aforo de las instalaciones, así como con el rol al que se pertenezca excluyendo al rol de Alumno NO UCM el cual no tendrá acceso a este tipo de entrada.
- **Precio:** El precio de entrada depende tanto del rol al que pertenezcas como del tipo de entrada que se adquiera. La siguiente tabla presenta todas las combinaciones **posibles de roles y tipos de entradas**.

	INDIVIDUAL	ABONO 10	ABONO ANUAL
NO UCM	5€	-	-
UCM	4€	25€	50€
ALUMNI	4€	25€	75€
PERSONAL	4€	25€	30€

Quedando los requisitos iniciales previos a la primera implementación definidos, podemos describir el método de entrada a la piscina:

- ➔ Encontrándose el usuario en la entrada de las instalaciones, deberá presentar su abono anual al personal allí presente, el cual le daría acceso inmediato siempre y cuando el aforo lo permita. Si no dispone de abono anual, deberá o bien comprar una entrada individual o un abono de 10 usos (si no tiene ya uno en su poder). Una vez realizada la adquisición (cumpliendo siempre los requisitos) el usuario podrá acceder a las instalaciones.

Como se puede observar, este sistema es bastante rudimentario y anticuado, lo que provoca que tenga carencias de todos los niveles como la inexactitud del aforo al no ser controlada la salida de las instalaciones o la posible formación de colas para poder acceder a la piscina. Incluso podría llegar a existir la suplantación de identidad a la hora de presentar la acreditación que

verifica la pertenencia a cualquiera de los roles debido a que la exigencia que se observó en estas situaciones no era de gran nivel.

4.2. Requisitos Posteriores

Una vez concluimos con la observación del proceso de entrada y conociendo los requisitos previos existentes, procedimos a contemplar los casos de uso y requisitos de la nueva implementación. Se definieron dos casos de uso: obtener ticket y acceder al recinto.

- **Obtener Ticket:** El presente caso de uso permite al usuario obtener un ticket de acceso a la piscina a través de la nueva aplicación.
La precondición a tener en cuenta en este primer escenario es que el usuario esté registrado en la aplicación de Pista Virtual.

Flujo principal

1. El usuario se valida en el sistema.
2. El sistema comprueba que tipo de descuento tiene el cliente en relación al rol al que pertenece.
 - 2.1. Si el cliente ya posee un tipo de entrada como un abono de 10 o un abono anual, le permite adquirir un ticket a precio 0 €.
 - 2.2. En caso de que el cliente no posea ningún tipo de abono, le permite la adquisición de un ticket sencillo aplicando la tarifa establecida en la tabla en relación a su rol.
3. Cuando el cliente vaya a realizar la adquisición de la entrada, se le ofrece la posibilidad de adquirir hasta 2 tickets extra para acompañantes.
 - 3.1. El usuario deberá indicar para cada entrada si los acompañantes están vinculados a la UCM o no, asociándose así la correspondiente tarifa a las entradas extra.
4. Se valida el número de tickets que el usuario está solicitando y posteriormente se procede a su adquisición y al pago en su caso.
5. El sistema contabiliza los tickets emitidos.

Flujos alternativos

- A. El sistema no encuentra el usuario o no tiene acceso a descuentos de ningún tipo.
 - I. Muestra un mensaje informado al usuario y le permite el acceso como usuario general.
- B. El sistema no tiene disponibilidad de tickets a la venta, mostrando en este caso un mensaje informando al usuario que no existen entradas para la fecha seleccionada.

Las **postcondiciones** establecidas son: la disposición de uno o varios tickets por parte del usuario que le permiten el acceso al recinto y la notificación de los tickets vendidos.

Cabe decir que la adquisición de un ticket es obligatoria. Siempre que se quiera acceder a la piscina. Estos tickets pueden tener un coste de 0 € si se posee un abono anual o un abono de 10 usos. El abono anual no es posible adquirirse a través de la aplicación (decisión del cliente) mientras que el abono de 10 usos sí lo es. En caso de no poseer ninguno de estos descuentos, siempre se podrá adquirir un ticket al precio establecido.

Al tratarse de una primera especificación de requisitos y definición de caso de uso, ambos flujos fueron modificados con ciertos detalles que mejoraron el proceso de adquisición de entradas. Estas modificaciones fueron consultadas y aprobadas por el cliente respetando siempre la idea final del producto.

En particular, la localización del tipo de descuento asociado al usuario se realizará como paso previo a la compra final de las entradas y una vez localizado (si lo hubiese), si este fuera un abono de 10, se le restaría un uso a dicho descuento eliminándolo finalmente si este se quedara sin usos. Se añadirá un *flujo alternativo* al punto A, limitando al usuario que ha accedido sin identificación a la compra única de tickets de acceso excluyéndolo de la compra de abonos de 10 usos. Se añadirá un caso de uso posterior, contemplado la compra de un abono de 10 usos.

- **Acceder al recinto:** El presente caso de uso describe el acceso del usuario a las instalaciones de la piscina.
Las precondiciones para este caso de uso son: La llegada del cliente a las instalaciones con el ticket de acceso previamente obtenido a través de la aplicación, el dispositivo de acceso/lectura está funcionando y no se permite el acceso si se ha llegado al aforo máximo.

Flujo principal

1. El usuario presenta un código QR generado en la compra del ticket frente al dispositivo de lectura.
2. El sistema comprueba la validez del ticket presentado.
 - 2.1. En caso de que el ticket esté caducado (la adquisición de dicho ticket es sólo válida para el mismo día de la compra) o sea inválido, muestra un mensaje visual y un sonido de error.
 - 2.2. En caso de que el ticket sea válido, si el usuario es personal NO vinculado a la UCM mostrará un mensaje de aceptación emitiendo un sonido diferente para que el administrador del lugar compruebe su validez.
 - 2.3. En caso de que el ticket sea válido, si el usuario es personal vinculado a la UCM mostrará un mensaje de aceptación.
3. El sistema aumenta en 1 la ocupación por cada ticket válido presentado.

Flujos alternativos

- A. El sistema no reconoce como válido el ticket presentado.
 - i. Muestra por pantalla un mensaje y emite un aviso sonoro informando del resultado erróneo, KO.

- B. El sistema ha alcanzado el límite máximo del aforo.
 - ii. Muestra por pantalla un mensaje informando del límite alcanzado emitiendo un aviso sonoro.

Las **postcondiciones** en este caso son: la entrada del usuario al recinto y la notificación del número de entradas que se producen.

Es importante tener en consideración que en un primer momento se planteó el sistema de acceso a través de la aplicación con un código QR y unos tornos que controlasen tanto la entrada como la salida de las personas, llevando así un conteo exacto del aforo. A posteriori, el cliente nos comunicó la imposibilidad de instalar los tornos. Esto supuso el cambio en los flujos principales de los dos casos de uso anteriores. El usuario ya no se encontrará con la imposibilidad de entrar habiendo comprado su ticket para ese día por un escenario de aforo máximo, ya que la compra de entradas se limitará a ese mismo día controlando el aforo de la piscina en el mismo momento de la compra. Por otro lado, al no poder controlar con los tornos la salida de los usuarios, este control de aforo quedará en manos del administrador de las instalaciones disminuyendo manualmente el foro actual en el sistema.

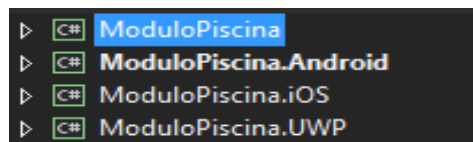
4. Implementación

En este apartado se va a explicar el diseño y estructura completa del aplicativo. En primer lugar, se explicará la estructura general, clases, subclases, así como los *plugins* añadidos para completar la funcionalidad.

Como hemos explicado en la sección 2 de esta memoria, se ha utilizado esta tecnología debido a la facilidad que plantea a la hora de implementar código multiplataforma. Esto nos permite desarrollar el código básicamente solo en un lenguaje combinado C# y Xaml para todas las plataformas existentes: IOS, Android y UWP (Universal Windows Platform). Cabe mencionar que en ciertos casos sí que es necesario especificar y desarrollar código para cada plataforma debido a las diferencias en ciertas características, como por ejemplo en las notificaciones o los permisos, como por ejemplo el de la cámara de fotos [6].

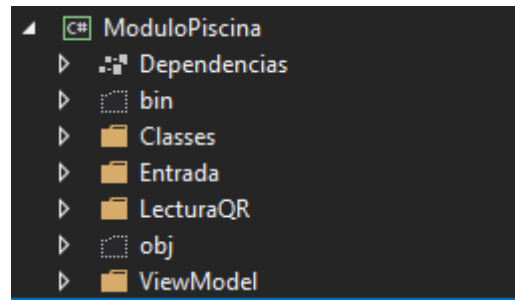
4.1. Diseño

La estructura que se genera cuando se crea un proyecto de Xamarin es la siguiente:



Como se puede observar se crean 4 proyectos. El proyecto principal y donde se va a desarrollar la mayor parte del código es en el primero, *ModuloPiscina*. Aquí es donde se implementa el código nativo, es decir, el código que va a ser útil para las 3 plataformas. También se crean otros 3 proyectos con propiedades específicas de cada plataforma como se ha comentado en la introducción de este apartado.

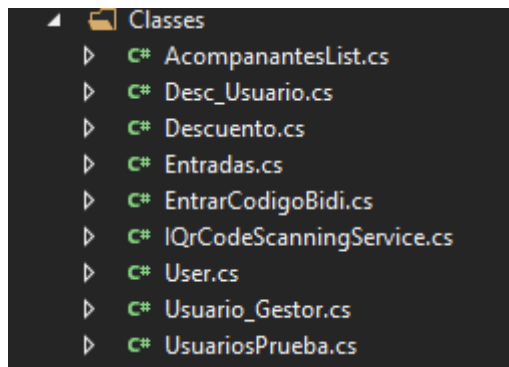
La estructura resultante de ModuloPiscina es la siguiente:



En esta estructura se observan varias carpetas para una mejor organización del proyecto. Por ejemplo, podemos diferenciar entre la parte de acceso a la piscina en la carpeta *Entrada* y la parte de control de acceso en *LecturaQR* o la parte de conexión del contenido visual y la lógica en *ViewModel* o incluso un conjunto de clases básicas con atributos de identificación. Todas ellas serán explicadas a continuación.

4.2. Explicación de la implementación.

En la carpeta de “*Classes*”, como comentamos anteriormente, encontramos clases simples implementadas en C# con atributos importantes y de los cuales se va a hacer uso en la parte visual.



Algunas clases son proporcionadas por el equipo de Pista Virtual para facilitar la integración en sus sistemas como por ejemplo la clase **Descuento**. Esta clase ya es usada en su aplicación para otros tipos de deportes y alquiler de pistas, por tanto nos pidieron que tuviésemos de base esta clase, la cual cuenta con atributos como *Id_Descuento*, *Nombre Id_Gestor*, *Cantidad*, *Fecha_Ini*, *Fecha_Fin*, etc., para la realización de nuestra aplicación.

En la parte relacionada con los usuarios encontramos la clases *User* en la cual se definen los atributos básicos a tener en cuenta en un usuario como el *Nombre*, *Apellidos*, *DNI...* y otros importantes como son:

```
public virtual List<Descuento> Descuentos { get; set; }  
public virtual List<Entradas> HistoricoEntradas { get; set; }
```

Nos encontramos ante dos listas de **Descuento** y **Entradas** las cuales van a permitir localizar, asignar y eliminar un descuento o una entrada al usuario correspondiente. Estas listas son inicializadas a través de la clase UsuariosPrueba, la cual es la encargada de inicializar los atributos básicos de la clase User y estas listas tan importantes de la siguiente manera

```
public class UsuariosPrueba
{
    public static User UsuarioEstudiante() {

        User usuario = new User
        {
            Id_Usuario = 1,
            Nombre = "carlos",
            Apellidos = "martin gutierrez",
            Email_Preferido = "camart12@ucm.es",
            Fecha_Nacimiento = new DateTime(1995, 8, 14),
            Sexo = true,
            DNI = "51229181T",
            Descuentos = new List<Descuento>(),
            Gestores = new List<Usuario_Gestor>(),
            HistoricoEntradas = new List<Entradas>()
        };
    }
}
```

```
usuario.Descuentos.Add(new Descuento {
    Id_Descuento = 2,
    Nombre = "Abono de 10 usos",
    Descripcion = "Este bono de 10 baños es para uso individual, no transferible. El uso de este abono esta limitado a usuarios de la UCM mayores de edad.",
    Precio = 25,
    Num_Usos = 8,
    Fecha_Inicio = DateTime.Now,
});
```

```
usuario.HistoricoEntradas.Add(new Entradas(2.ToString(), usuario.Id_Usuario, 2,
9, new DateTime(2009, 08, 25), new DateTime(2018, 08, 26), null ));

return usuario;
}
}
```

La clase **EntrarCódigoBidi** se encarga de generar a través del plugin instalado ZXingBarcodeImageView [7]el código QR para el servicio de escaner identifique una entrada. Se ha realizado de la siguiente manera:

```
barcode = new ZXingBarcodeImageView
{
    HorizontalOptions = LayoutOptions.FillAndExpand,
    VerticalOptions = LayoutOptions.FillAndExpand,
};
barcode.BarcodeFormat = ZXing.BarcodeFormat.QR_CODE;
barcode.BarcodeOptions.Width = 250;
barcode.BarcodeOptions.Height = 250;
barcode.BarcodeValue = id_usuario + " " + id_entrada + " " + id_descuento + " " +
num_usos + " " + fecha_adquisicion.ToShortDateString() + " " +
fecha_fin.ToShortDateString();
```

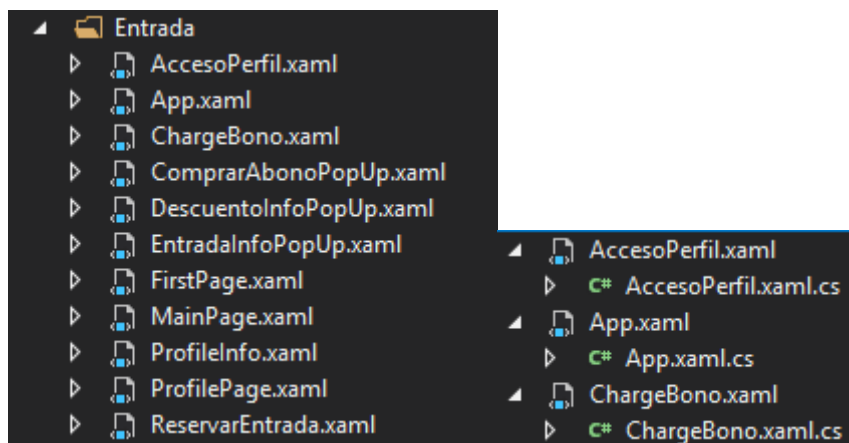
Como se puede observar en las 3 últimas líneas de este fragmento de código, a partir de varios atributos de la clase Entrada se identifica y se crea el código QR el cual podrá ser leído posteriormente y localizar así el ID del usuario al que pertenece, el propio ID de la entrada, etc. Además de la generación del código, muestra información de interés de la entrada.

La clase `IQRCodeScanningService` es simplemente una interfaz encargada de inicializar la cámara y el plugin encargado de la lectura de códigos QR.

```
public interface IQRCodeScanningService
{
    Task<string> ScanAsync();
}
```

Por último, podemos definir a las clases **AcompañantesList**, **Desc_Usuario** y **Usuario_Gestor** como clases de conexión tanto para el contenido visual con la lógica como entre otras clases. Por ejemplo `Usuario_Gestor` crea una relación entre un usuario (`Id_Usuario`) y el tipo de rol al que pertenece y `AcompañantesList` nos proporciona a través de un objeto de tipo `Entrada` una lista de precios para los vinculados a la UCM y los NO vinculados lo cual cobrará sentido en la parte visual.

Continuamos con la carpeta “Entrada” en la cual se definen las clases relacionadas con la parte del cliente a la hora de realizar la compra, la reserva o acceso a las instalaciones. En esta carpeta encontramos uno de los esquemas característicos de este lenguaje, la conexión entre la lógica en C# y la parte visual en Xaml, un lenguaje basado en HTML y XML con ciertos atributos que nos permiten esa conexión. En la Imagen 2 se observa el desglose de una página de contenido (C# + Xaml). El esquema resultante sería el siguiente:



Cada una de estas páginas de contenido representa una ventana en la aplicación móvil. A continuación, se explicarán una a una todas estas páginas, su funcionalidad principal y alguna referencia al código en sí. Se explicarán en orden de acceso a la aplicación.

En primer lugar, la página `App` es creada automáticamente en un proyecto de Xamarin. Es la página por defecto que se crea en el proyecto nativo para poder lanzar la aplicación. Como se ha comentado con anterioridad, tenemos la parte de contenido visual, y la parte lógica en C#. A continuación, se muestra la parte lógica `App.Xaml.Cs`:

```

namespace ModuloPiscina
{
    public partial class App : Application
    {
        public App()
        {
            InitializeComponent();
            MainPage = new NavigationPage(new MainPage());
        }
    }
}

```

En la constructora se observan dos líneas. La primera, `InitializeComponent()`, es una función interna que permite el lanzamiento del contenido visual de propia página la cual vamos a encontrar en el resto de páginas de contenido. La segunda, es la inicialización de la navegación entre páginas que nos permitirá navegar entre las diferentes partes de la aplicación.

A continuación, el flujo de navegación continuaría hasta `MainPage`. En esta página ya podemos decir que la parte visual cobra importancia, situando dos botones que van a dividir la parte de control de acceso y la parte de entrada, compra, etc. El código Xaml que encontramos en `MainPage.xaml` es el siguiente:

```

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:ModuloPiscina"
             x:Class="ModuloPiscina.MainPage"
             Title="PISCINAS UCM">

    <StackLayout >

        <Button ClassId="usuario" Clicked="OnButtonClicked" Style="{StaticResource
            botonStyle2}" Text="USUARIO" VerticalOptions="CenterAndExpand"/>

        <Button ClassId="trabajador" Clicked="OnButtonClicked"
            Style="{StaticResource botonStyle2}" Text="TRABAJADOR"
            VerticalOptions="CenterAndExpand"/>

    </StackLayout>

</ContentPage>

```

Como se puede observar, con la etiqueta `<Button>` se crean los dos botones que, con ciertos atributos de estilo `Style`, de control de acción `Clicked`, de identificación `ClassId`, o del texto a mostrar `Text`, quedan implementados. La funcionalidad de los botones se implementa en la parte lógica de la página `MainPage.xaml.cs` donde a través del valor otorgado al atributo `Clicked`, el método que realiza la funcionalidad deseada es llamado y ejecutado. Dicho método `OnButtonClicked` queda definido de la siguiente manera:

```

async void OnButtonClicked(Object sender, EventArgs e){

    Button btn = (Button)sender;

    switch (btn.ClassId) {

```

```

        case "usuario": await Navigation.PushAsync(new ProfilePage()); break;

        case "trabajador": await Navigation.PushAsync(new ScannerPage()); break;
    }
}

```

A través de los atributos de identificación podemos controlar qué botón ha sido accionado y, como consecuencia, su correspondiente acción a realizar. Para ambos botones la acción será redirigir a las siguientes páginas en el flujo de navegación.

Continuamos con **ProfilePage** donde se creará un listado de objetos de tipo `User` con diferentes descuentos y entradas que nos permitirá acceder al perfil seleccionado. Es importante recalcar que este tipo de acceso se ha llevado a cabo para la versión *offline* ya que una vez integrada en la aplicación de Pista Virtual, se realizará con los credenciales de la base de datos de sus sistemas y cuyos datos personales como de contenido no serán cargados de forma manual y estática.

Una vez se ha accedido con uno de los usuarios de la lista, nos situamos en la pantalla principal de la aplicación **FirstPage**, donde se tendrá la posibilidad de realizar las principales acciones y funcionalidades del aplicativo. En la parte visual de esta página, tenemos los siguientes botones:

```

<StackLayout Grid.Row="2">

    <Button ClassId="comprarEntrada" Style="{StaticResource botonStyle2}"
    Text="Comprar entrada" Clicked="NavigationBarButtons" />
    <Button ClassId="comprarBono" Style="{StaticResource botonStyle2}"
    Text="Comprar bono" Clicked="NavigationBarButtons" />
    <Button ClassId="misEventos" Style="{StaticResource botonStyle2}" Text="Mis
    Eventos" Clicked="NavigationBarButtons" />

</StackLayout>

```

En el primero de ellos navegaremos hasta la pantalla donde podremos reservar una entrada mediante la función `ReservarEntrada`. Previamente a la navegación a esta página, se realizan varias comprobaciones de cumplimiento de los requisitos previos especificados en la sección 3, como por ejemplo que el aforo no se haya superado o que el usuario ya haya realizado 3 reservas para el día actual, especificación que se concretó en reuniones posteriores. Todas las posibles opciones de error y denegación de acción por incumplimiento de requisitos están controladas por cuadros de diálogo que informan al usuario de lo ocurrido.

Una vez nos situamos en **ReservarEntrada**, nos encontramos con una de las páginas más complejas y con contenido del aplicativo. En esta página se ofrecerá al usuario reservar una entrada para el día actual, así como las entradas para los posibles acompañantes, con un máximo de 2. Esta funcionalidad se realizará a través de una lista de tickets donde el usuario podrá elegir la vinculación de sus acompañantes con la UCM debido al cambio de precio que implica. A su vez, se mostrará dinámicamente el precio de cada entrada y el precio total a pagar por el usuario. Como se indica anteriormente, las entradas que el usuario podrá reservar quedarán limitadas a un máximo de 3.

```

<ListView x:Name="acompanantesLista" SeparatorColor="Black">
    <ListView.ItemTemplate>
        <DataTemplate >
            <ViewCell Height="70">
                <ContentView>
                    <Grid>

```

```

<Grid.ColumnDefinitions >
  <ColumnDefinition Width="4.5*" />
  <ColumnDefinition Width="4.5*" />
  <ColumnDefinition Width="1*" />
</Grid.ColumnDefinitions>

  <Label Grid.Column="0" TextColor="Black"
        Text="{Binding Entrada.Id_entrada_String}"
        HorizontalOptions="Start" >
  </Label>
  <Picker Grid.Column="1" WidthRequest="70"
        SelectedItem="{Binding ValorSeleccionado}"
        ItemsSource="{Binding ValoresPicker}" TextColor="Red"
        Title="Pertenencia a la UCM" HorizontalOptions="Start" />

  <Label x:Name="cantidadIndividual" Grid.Column="2"
        Text="{Binding Cantidad}" />

</Grid>
</ContentView>
</ViewCell>
</DataTemplate>
</ListView.ItemTemplate>
</ListView>

```

Aquí se muestra parte de **ReservarEntrada.xaml**. La complejidad de esta clase no reside en la parte lógica, donde se suman cantidades, se crean objetos de tipo y se insertan datos en listas Descuentos o Entradas, sino en la conexión y vinculación (*Binding*) [8] de los datos de estas clases con la parte visual.

Como se observa en el código anterior, esta conexión se hace a través del atributo *Binding*, donde el valor que toma hace referencia al atributo de la clase que se está representado. Si se trata de una lista como en este caso *ListView*, no es necesario indicar la clase a la que pertenecen los valores. Si hablamos de otras representaciones como etiquetas o cuadros de texto, habría que indicar en la lógica en qué contexto de vinculación nos encontramos (*BindingContext HACER REFERENCIA otra vez*) aunque también existen casos donde se podría indicar directamente en la parte visual con atributos de etiqueta como en los *Picker* o *selectores* (resaltado en verde).

Una vez realizada la reserva, el flujo de navegación nos redireccionará a la lista de entradas del usuario identificado. Esto quiere decir que nos encontramos en la siguiente página de contenido de la aplicación, **AccesoPerfil**.

Nos encontramos ante una lista de objetos de tipo *Entrada*. Como se ha explicado anteriormente, es necesario vincular el contenido para ser mostrado en *AccesoPerfil.xaml* pero al tratarse de una lista no es necesario indicar en contexto, simplemente habría que indicar cuál es la fuente de datos para esa lista (identificada a través del *Id*) en la parte lógica *AccesoPerfil.xaml.cs*, normalmente en la constructora de la misma. En este caso la fuente de datos proviene de la lista de entradas del usuario.

```

todasEntradas.ItemsSource = user.HistoricoEntradas;

```

Volviendo al menú principal, tenemos la opción de comprar un abono. Actualmente, por petición del cliente, solo se puede realizar la compra de un abono de 10 baños. En un principio se contemplaban todas las posibilidades de compra (abono anual, piscina de invierno y verano) pero a lo largo del desarrollo las especificaciones fueron cambiando, siendo esta una de las

modificaciones. Hay que tener en cuenta que solo los usuarios identificados van a poder realizar la compra de estos abonos.

Esta funcionalidad es implementada en **ChargeBono**, donde se da al usuario la opción (a través de *Picker*) de elegir tanto el tipo de abono a comprar como del método de pago, eligiendo entre cargo en cuenta o tarjeta bancaria. Se decidió implementar este método de cara a que en el futuro se añadiesen el resto de abonos a las opciones. Con respecto al sistema de pago, cabe decir que el equipo de Pista Virtual nos ha proporcionado el TPV que utilizan en sus sistemas y a recomendación de nuestro tutor, lo aceptamos. En la aplicación *offline* no está implementado, simplemente mientras se cumplan los requisitos que necesita la aplicación para la adquisición de estos abonos se completa. Ocurre lo mismo en la reserva de entradas.

Antes proceder a la compra, se activa un *Pop-Up* el cual muestra al usuario la información relevante del abono que están a punto de comprar. Esto nos pareció un requisito a tener cuenta ya que en nuestra opinión, el usuario debe estar lo más informado posible del producto que está a punto de adquirir. Cabe decir que no era ninguna exigencia ni requisito por parte del cliente, pero nos pareció buena idea y al enseñárselo no se opusieron a esta funcionalidad.

Este *Pop-Up* lo encontramos en la clase **ComprarAbonoPopUp** donde a través de la vinculación de contenidos mostramos información de tipo Descuento.

----- ComprarAbonoPopUp.xaml.cs -----

```
BindingContext = new DescuentoViewModel(descuento);
```

.....

Esta vez no vamos a vincular datos sobre una lista, sino que los datos se van a mostrar en etiquetas de texto *Label* a las cuales hay que indicarles el origen de estos datos. En este caso se hace a través de una clase auxiliar *DescuentoViewModel* que nos permite, pasándole por parámetro el descuento indicado, obtener a través de *getters* los atributos de la clase Descuento. Esto es posible ya que los atributos de una clase generan automáticamente estos métodos de acceso

```
public int Id_Descuento { get; set; }
public byte Tipo { get; set; } [ Atributos de la clase Descuento ]
public string Nombre { get; set; }
```

----- DescuentoViewModel.cs -----

```
public string Fecha_Adquisicion {
public string Nombre {
get { return "Tipo de Descuento: " + descuento.Nombre; }
}
public string Usos {
get { return "Número de usos " + descuento.Num_Usos.ToString(); }
```

```
}
```

----- ComprarAbonoPopUp.xaml -----

```
<StackLayout>
  <Label TextColor="Black" Text="{Binding Nombre }" />
  <Label TextColor="Black" Text="{Binding Usos }" />
  <Label TextColor="Black" Text="{Binding Precio }" />
  <Label TextColor="Black" Text="{Binding Descripcion }" />
</StackLayout>
```

Terminamos explicando las clases **ProfileInfo** y **DescuentoInfoPopUp**, las cuales fueron descartas en último momento en las reuniones finales. La clase ProfileInfo lo que ofrece es la visualización de los datos del usuario como su nombre, apellidos... con diferentes etiquetas *Label* y haciendo la vinculación de datos con la clase User de la misma forma que la descrita anteriormente. Lo mismo ocurre para la clase DescuentoInfoPopUp, que existiendo una lista de abonos pertenecientes al usuario, muestra la información relativa a este como la fecha de adquisición, número de usos en caso de ser abono de 10 usos...

El motivo de descarte fue que ya existía una sección de perfil del usuario en la aplicación donde este módulo iba a ser integrado. Esta página esta implementada y sigue activa en la versión *offline*.

5. Manual de Usuario

En esta sección vamos a mostrar y a explicar las diferentes pantallas de la aplicación integrada. Como hemos explicado a lo largo de la memoria, el módulo Piscina ha sido desarrollado como parte de la aplicación final que el equipo de PistaVirtual ya poseía y también como proyecto de fin de grado.

Cabe decir que ciertas funcionalidades realizadas en la aplicación *offline* han sido suprimidas por PistaVirtual o bien por cambio de estructura de código o por una implementación ya existente en la aplicación general. También ciertos posicionamientos de elementos visuales, así como su presentación de colores, fuentes... también han podido ser modificados para el producto final por decisión del equipo.

Muy importante resaltar que el producto final, el cual en la parte *offline* está unificada en una única aplicación, consta de dos partes, de dos aplicaciones separadas. Por un lado la parte de gestión y acceso, a la cual nos referiremos como "Gestor", a la cual pertenece la lectura de los códigos QR, gestión de aforo o visualización de histórico de entradas. Por otro lado, la parte del usuario, a la cual nos referiremos como "Usuario", donde se puede realizar la compra y adquisición de entradas o abonos, la visualización del perfil o el histórico personal de acceso y compras.

5.1. Aplicación Gestor

El aplicativo de la gestión de PistaVirtual consiste en la gestión del aforo y la entrada a través del escaneo de los códigos generados en la parte del Usuario. Esta parte es sencilla y consta de pocas pantallas, obviamente siempre haciendo referencia al módulo de la piscina.

Este aplicativo también consta de una pantalla de autenticación, la cual lógicamente requiere de unos credenciales diferentes a los de la parte del Usuario. En este caso, dicho datos han sido proporcionados por PistaVirtual. Una vez dentro, nos encontramos con el siguiente menú donde se pueden observar las diferentes funcionalidades del Gestor.

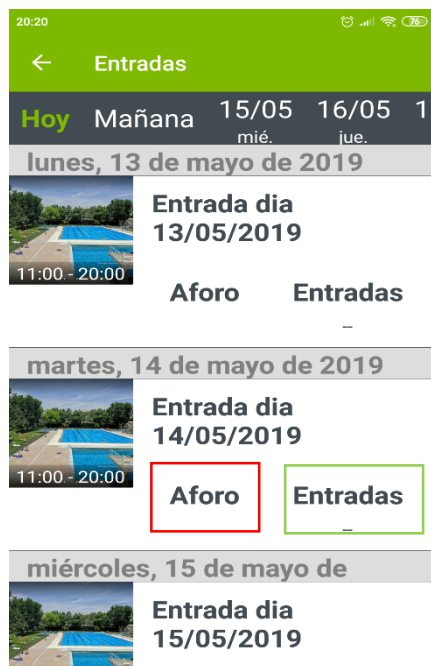
Nosotros nos vamos a centrar en el botón situado en la parte inferior derecha de la pantalla, “ENTRADAS” (cuadro rojo), donde se gestionan las entradas a todos los diferentes eventos que la aplicación ya gestionaba, incluyendo ahora las entradas a la piscina.



Como comentamos, en esta pantalla encontraremos filtrado por días todos los eventos que gestiona PistaVirtual. Actualmente solo están habilitados los eventos de la piscina por una cuestión de exigencia del cliente.

En esta visualización de los eventos de piscina, nos encontramos las horas de apertura, así como el aforo máximo que permite (cuadro rojo) y las entradas restantes disponibles (cuadro verde).

Actualmente la funcionalidad está deshabilitada. Sin embargo, cuando se active a su paso a producción, al pinchar en la opción de las entradas disponibles, el administrador o el encargado de las instalaciones podrá reducir o aumentar el número de entradas disponibles, siendo este el sistema de control de aforo (a parte de la reducción automática de este contador cuando se realiza una compra). Este sistema se llevó a cabo debido a la posibilidad de reserva de entradas para grupos especiales, actividades extraordinarias y, lo más importante, la cancelación de los tornos para el sistema de entrada.



Pinchando en el cuadro del día nos encontramos con la pantalla encargada de escanear los códigos y mostrar los correspondientes mensajes de aceptación o de error. En la parte superior de la pantalla se observa cómo nos indica el día para el cual estamos escaneando las entradas. Esto sirve de orientación ya que el requisito principal para que una entrada sea aceptada es el día al que pertenece. Justo debajo de la fecha podemos observar un rectángulo con tres celdas, una en la que se indica el aforo de la piscina, otra las entradas vendidas y una última con las entradas validadas del día indicado. Una vez escaneado el código QR, debajo de la pantalla te sale un recuadro con un listado de las entradas ya escaneadas que sirva para que el personal pueda comprobar fácilmente las entradas ya aceptadas, y si intentas pasar una entrada dos veces se ve cómo te dice que ese ticket no es válido porque ya ha sido usado.



Por último, en referencia al Gestor del módulo de la piscina nos encontramos con los dos posibles casos de lectura: Entrada válida o no válida. En ambos casos, observamos un mensaje que nos indica el resultado de la validación, indicándonos en más detalle el por qué en el posible caso de que la lectura haya resultado incorrecta. Una vez más de cara a facilitar el trabajo y la visualización del personal de las instalaciones.

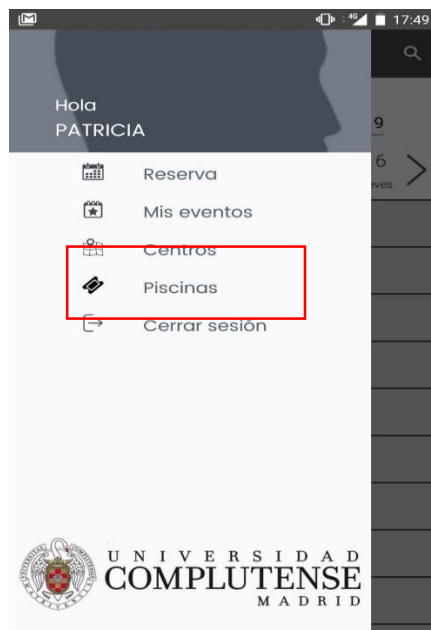


5.2. Aplicación usuario

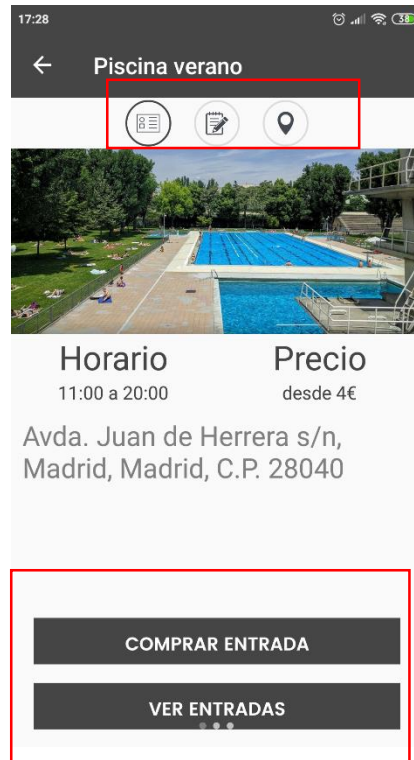
En primer lugar, nos encontramos con el acceso a la aplicación. Podemos realizar la autenticación de dos maneras, o bien con un usuario y contraseña previamente habiendo realizado un registro, o bien con los credenciales de tu cuenta de la UCM. Se observa también la posibilidad de recuperar tu contraseña siempre y cuando estés registrado.



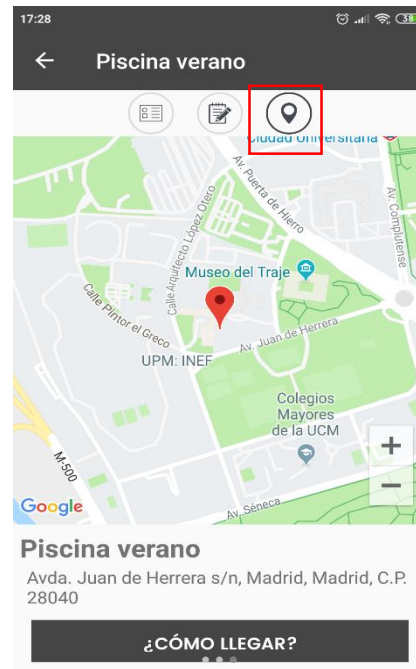
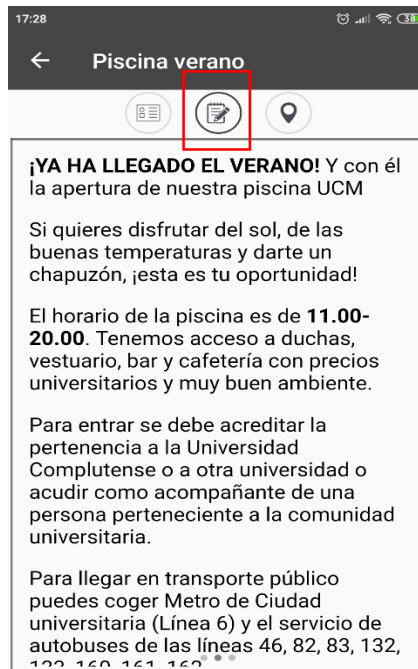
Una vez hemos accedido con los credenciales correspondientes nos encontramos ante la pantalla principal de la aplicación. Esta pantalla ya es parte de la aplicación previa existente en la que se ha integrado el módulo piscina, en la cual nos vamos a centrar. En la parte superior izquierda, nos encontramos con un menú que nos permite el acceso a la parte de las piscinas.



A continuación, nos situamos en la pantalla principal del módulo de las Piscinas de la UCM. Podemos observar una información muy básica la cual es ampliada deslizando la pantalla hacia la derecha (3 círculos), donde además de información extra nos encontramos con un mapa integrado de donde localizar las instalaciones. Observamos 2 botones al final de la pantalla donde podremos realizar la compra/reserva de una entrada o ver las entradas para el día de hoy.



Como comentamos, podemos encontrar deslizando hacia la derecha más información acerca de la piscina de verano y que los círculos encuadrados en la anterior imagen nos indican dónde estamos. En la segunda imagen el botón “¿CÓMO LLEGAR?” nos redireccionaría a Google Maps con la dirección de las instalaciones.



Volviendo la pantalla principal del módulo de la piscina, vamos a acceder a la parte de compra y de visualización de imágenes.

En primer lugar, pinchando en el botón de comprar entrada encontramos una página donde habrá que seleccionar, en función de la relación del usuario con la UCM, el número de entradas que deseamos comprar. En función del aforo del momento y del número máximo de entradas

por comprar permitidas para un usuario, nos aparecerá un mensaje de error informando de lo ocurrido o nos permitirá continuar con la compra. Se aprecia el precio total a pagar función de las selecciones, así como un desplegable indicando el método de pago (actualmente solo se permite el pago por tarjeta).

Una vez se pincha en “PAGAR” nos redirige al TPV virtual que PistaVrtual y la Universidad Complutense de Madrid tiene contratado, en este caso gestionado por redsys.

Comprar entrada

Tipo de entrada	Nº
Comunidad UCM	4,00 € 1
No vinculados UCM	5,00 € 0

Forma de pago: Tarjeta

Importe a pagar 4,00 €

PAGAR

Pago con tarjeta

4,00 €

Pagar con Tarjeta

Nº Tarjeta:

Caducidad: mm aa

Cód. Seguridad: ?

Cancelar Pagar

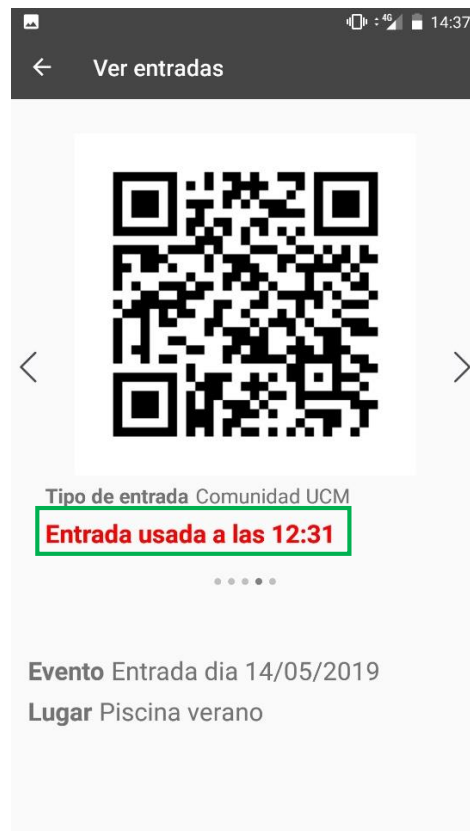
Datos de la operación

Importe:	4,00 €
Comercio:	RESERVAS DEPORTES UCM (SPAIN)
Terminal:	66622879-1
Pedido:	000000504080
Fecha:	13/05/2019 18:37

En segundo lugar, pinchando en VER ENTRADAS nos encontramos ante todas las entradas que el usuario posee para el día de hoy. Como se puede observar (cuadros rojos) en esta imagen, existen varias entradas a las cuales se accede deslizando hacia la derecha. También nos encontramos con información relevante (cuadros verdes) la cual puede ser útil para el personal de las instalaciones o el usuario a la hora de localizar e identificar información relacionada con la entrada a escanear. Las entradas caducarán si no han sido usadas y dejarán de ser válidas una vez se han escaneado. Cada código QR almacena un identificador del precio de entrada, tipo de entrada, identificador del usuario, nombre del comprador, precio del pago, identificador de entrada, nº entrada, todo ello encriptado para mejorar la protección de datos, por lo que el gestor lee, descrypta y valida.



Existe el caso en el que el usuario se encuentre en la pantalla de visualización de sus entradas y con código ya escaneado. En esta situación, la aplicación nos indicará a través de un mensaje, situado debajo del código QR, que la entrada ya ha sido utilizada junto con la hora del uso. En tal caso la aplicación del Gestor no dará por válido la lectura de la misma.



6. Contribución de los alumnos

En esta sección presentamos las contribuciones individuales de cada uno de los miembros del equipo.

6.1. Carlos

Cuando decidimos hacer este Trabajo de Fin de Grado lo primero que empecé a hacer fue comenzar a revisar documentación de .NET y de C#.

Tras la reunión de verano en la piscina, estuve mirando TPV y cómo conectar a los tornos la aplicación, ya que en un principio lo que íbamos a implantar eran unos tornos de acceso que la Complutense tenía comprados desde hacía bastante tiempo.

Una vez comenzado el curso, estuve documentándome más sobre la tecnología y las herramientas que íbamos a utilizar. Lo primero que hice fue actualizar el Visual Studio e instalarme los módulos necesarios para poder empezar a realizar la implementación.

Una vez estuvo todo instalado fuimos a una reunión con Pista Virtual, y con ellos decidimos que mejor que ellos nos dieran su aplicación, era empezar una desde el principio nosotros y así poco a poco ir aprendiendo desde los comienzos y no meternos de lleno en una aplicación que en ese momento no sabríamos ni por dónde empezar.

Para comenzar estuve mirando tutoriales de YouTube [9] y empezando a hacer pequeñas aplicaciones independientes al módulo piscina, que sirvieron para familiarizarme con el lenguaje y con la nueva tecnología que hemos utilizado.

Mientras íbamos realizando nuestras primeras ventanas, fuimos a la piscina de invierno a informarnos del horario, las clases y los precios. Durante una tarde estuvimos hablando con la persona al cargo y nos dio toda la información que necesitábamos, principalmente los precios, los roles de los usuarios, los horarios y el funcionamiento de la piscina de invierno.

Hasta prácticamente navidad estuvimos realizando aplicaciones por nuestra cuenta, que poco a poco se fueron adaptando a lo que íbamos a tener que entregar al equipo de Pista Virtual.

Cuando comenzamos a realizar una aplicación en conjunto, empezamos realizando un menú en el que estaban incluidos todas las opciones que podría tener nuestra aplicación.

De este menú comencé realizando el Perfil del usuario, que, aunque no esté incluido en Pista Virtual porque ellos ya tenían su propio perfil, hicimos uno para nuestra aplicación. El perfil, estaba formado por los datos personales de usuario y el tipo de abono o pase que tenía para la piscina.

Después del perfil estuve generando el código QR, ver cómo se hacía me llevó bastante tiempo, a pesar de estar siguiendo un tutorial, fuimos a una tutoría con Pista Virtual en la que nos ayudaron y nos enseñaron un GitHub que nos sirvió de ayuda. Para la generación del código QR tuvimos que instalarnos un paquete nuevo de Xamarin, que nos facilitó bastante las cosas. Una vez que el código QR se mostraba le añadimos más información para que fuese más seguro, aunque actualmente el equipo de Pista Virtual, ha añadido más de la que nosotros pusimos en su momento.

Una vez teníamos el código BIDI y el escáner, estuvimos uniéndolos en una aplicación, una vez unidos empezamos a hacer un poco la lógica de acceso y para ello nos hicimos nuestros propios caso de uso, y comencé a realizar la implementación de acceso a la piscina para los usuarios.

Cuando ya teníamos casi toda la aplicación acabada, la Complutense nos dio nuevos casos de uso. El primero de ellos solicitaba que el usuario pueda comprar una entrada para la piscina, el cual tenía unos pasos a seguir. Lo primero que tenía que comprobar era que el usuario estuviera en el sistema, y comprobar que tipo de descuento tenía asociado el usuario. En caso de que el usuario tuviera un abono el coste para sería de 0 euros, en caso contrario, el usuario accedería a la compra de un ticket para él y un máximo de dos tickets más para posibles acompañantes, pertenezcan a la Complutense o no.

Una vez terminados los casos de uso, adaptamos nuestra aplicación lo máximo posible a lo que Pista Virtual tenía, para ello el equipo nos proporcionó los colores y los distintos estilos que ellos tenían en su aplicación. Una vez tuvimos todos los botones y colores adaptados, pista virtual nos dio unos *mockups* de cómo tenían pensado que fuese la aplicación.

Una vez acabada la aplicación comenzamos a realizar la memoria. La dividimos entre los dos autores y la unimos poco a poco.

Lo primero que realice de la memoria fueron los requisitos, primero los que habíamos acordado tanto en la reunión del verano de 2018 en la piscina de verano, como los que nos explicaron en la piscina de invierno durante el primer cuatrimestre. A continuación expuse los requisitos que la Complutense nos detalló para el funcionamiento de la aplicación.

Después de acabar esta parte tuvimos una primera reunión con el tutor sobre cómo estructurar la memoria, ya que no sabíamos exactamente qué poner. En esa reunión establecimos un índice con las partes más importantes que debería llevar la memoria. Una vez establecido el índice hicimos un reparto equitativo y fuimos escribiendo poco a poco, como detalle a continuación.

Cuando terminé este apartado, me puse a redactar el código que habíamos ido implementando cada uno de nosotros, tanto la parte del diseño como la de la implementación.

Una vez nuestro código estuvo integrado en la aplicación de Pista Virtual, y tras haber estado haciendo diferentes pruebas, comencé a realizar el manual de usuario de la misma que podemos ver en la sección anterior, a falta de algunas imágenes que iban a irse actualizando a medida que hicieran nuevas versiones.

Continué escribiendo las conclusiones de este Trabajo de Fin de Grado.

Una vez terminada la memoria en castellano y validada pues nuestro tutor a medida que íbamos avanzando y corrigiendo nos pusimos con la traducción. Yo realicé la traducción de la introducción y de las conclusiones y trabajo futuro.

Por último, realizamos este apartado, en el cuál he redactado mis aportaciones al proyecto durante todo este curso.

6.2. Patricia

Una vez decidido el TFG y ya sabiendo las tecnologías que íbamos a utilizar para su implementación, lo primero que hice fue empezar a leer la documentación de Microsoft, comencé mirando que era .NET, para qué servía y cómo se utilizaba. A continuación estuve mirando qué era C# y qué patrones seguía.

Igual que mi compañero, tras la reunión con la piscina de verano y sabiendo que en un principio íbamos a utilizar los tornos de la piscina de invierno, estuve mirando distintas posibilidades de conexión de los tornos con la aplicación ya que la Complutense los tenía comprado desde hacía tiempo y los tiene en la piscina de invierno sin utilizar.

Lo primero que hice fue lo mismo que hizo Carlos, me actualicé el Visual Studio añadiendo los módulos necesarios para comenzar la implementación de la aplicación y poder empezar a programar el módulo piscina. Una vez tuvimos todo instalado y habíamos mirado suficiente documentación tuvimos una primera reunión con el equipo de Pista Virtual. Nos aconsejaron empezar haciendo nosotros pequeñas aplicaciones, uniendo unas pantallas a otras, por lo que eso fue lo que hicimos los dos. Igual que mi compañero comencé con el tutorial de YouTube [9], que nos sirvió para familiarizarnos y poco a poco ir avanzando en el lenguaje y cogiendo base de cara a las futuras implementaciones.

Para comenzar a hacer estas pantallas nos reunimos los dos e hicimos unos casos de uso a papel que posteriormente enseñamos al equipo de Pista Virtual, para que nos los validaran y vieran que les parecía la lógica que queríamos darle a la aplicación.

Cuando ya tuvimos los casos de uso establecidos, comenzamos a realizar las ventanas y pequeñas aplicaciones con una lógica que ya habíamos establecido previamente.

Mientras íbamos adquiriendo conocimientos de las herramientas nuevas e íbamos afianzando conocimientos, fuimos a la piscina de invierno para ver cómo era su funcionamiento. Estuvimos una tarde allí con la persona al cargo, quien nos explicó todo sobre ambas piscinas. Nos ofreció un documento en el que especificaba todos los roles de los usuarios y los distintos precios de ambas piscinas.

Una vez comenzado el segundo cuatrimestre, las visitas a Pista Virtual aumentaron. Con ellos decidimos comenzar haciendo un menú y poco a poco ir avanzando.

De este menú comencé realizando la información, que en ella estaban incluidas las dos piscinas, la de verano y la de invierno y una información general acerca de lo que son ambas piscinas. Cabe destacar que en la aplicación que enseñamos en el manual de usuario solo está añadida la información de la piscina de verano, ya que de la piscina de invierno todavía no hay casos de uso ni requisitos. Una vez acabada la información, mientras que mi compañero hacía el desarrollo del código QR, yo realicé una aplicación paralela que simula al gestor. Es decir, comencé a realizar el escáner de código QR que validaba el código BIDI que mi compañero había creado. Para ello también me ayudé de un tutorial y del GitHub que Pista Virtual nos había enseñado previamente en una de las reuniones con ellos y que nos había facilitado la instalación de un paquete de Xamarin para ayudarnos con estos apartados.

Una vez tuvimos tanto el escáner como el código BIDI unimos ambas aplicaciones y empezamos a realizar las primeras pruebas de lectura y a comenzar con la lógica de acceso a la piscina.

Una vez tuvimos la aplicación bastante avanzada, fue cuando la Complutense nos dio sus especificaciones, en concreto había dos, yo me encargué de la segunda de ellas. En ella hacía referencia al acceso al recinto de la piscina de verano. En este caso de uso especificaba que el acceso se hacía mediante un ticket (código BIDI), para ello leía el ticket y comprobaba si era válido. En caso de ser apto, muestra un mensaje de acceso permitido, con el rol del usuario y con un sonido predeterminado. En caso de que el ticket no sea apto para entrar muestra un mensaje de error y porque, como por ejemplo si ya se ha pasado previamente con esa entrada o si está caducada, esta lectura irá acompañada de un sonido distinto al del acceso correcto. Junto con el acceso de la persona se suma uno a las personas en la piscina para así no sobrepasar el aforo.

Como mi compañero ha comentado anteriormente, una vez tuvimos terminada la aplicación nos pusimos a realizar las interfaces que Pista Virtual nos había dado como guías para que a la hora de integrar la aplicación les fuera lo más fácil posible.

Una vez acabada la aplicación nos pusimos con la memoria. Nos dividimos las distintas secciones y las fuimos juntando poco a poco mientras nuestro tutor nos la iba validando.

Lo primero que realicé de la memoria fueron algunos apartados como la motivación, los objetivos, una pequeña introducción y parte de las tecnologías. Puesto que no sabíamos muy bien como estructurar la memoria, tuvimos una reunión con el tutor para que nos revisara lo poco que habíamos escrito y ver como estructurarla. Durante la reunión, como ha comentado mi compañero, hicimos un índice inicial sobre el que basarnos para ir empezando.

Lo primero que hice tras esta reunión, fue reestructurar el punto de la introducción y dejarlo como un capítulo entero. Para ello me base en el índice que habíamos estructurado.

Cuando terminé con la introducción, me puse a escribir los preliminares, en los que uní parte de las tecnologías que ya había escrito previamente, pero que nuestro tutor nos dijo que cambiáramos porque era muy técnico y era mejor explicarlo de manera diferente y con referencias a la documentación. Nos aconsejó que en este apartado explicáramos qué es Pista Virtual, ya que durante toda la memoria les mencionamos, pero en ningún momento decíamos quiénes son.

Una vez terminados estos puntos escribí el trabajo futuro del proyecto, abarcando posibles mejoras y futuras incorporaciones para la aplicación.

Una vez terminada gran parte de la memoria, escribí el resumen y los agradecimientos.

Cuando la memoria estuvo corregida, procedimos a la traducción, yo me encargué del resumen y de los agradecimientos.

Por último añadimos esta sección en la que yo he escrito mi contribución al proyecto, desde el primer día que nos pusimos con el Trabajo de Fin de Grado.

7. Conclusiones y trabajo futuro

Para terminar, exponemos a continuación ciertas conclusiones que abarcan tanto desde el punto de vista de la trabajar con una nueva tecnología, hasta las sensaciones de trabajar para un cliente tan importante como puede ser la Universidad Complutense de Madrid.

Nos centraremos también en las posibles aplicaciones que el módulo implementado puede aportar en las instalaciones de las piscinas.

7.1. Conclusiones

Como trabajo de fin de grado, podemos decir que ha sido una experiencia y un trabajo satisfactorio y gratificante ya que el hecho de aprender una nueva tecnología de desarrollo, como ha sido en este caso con .NET y Xamarin, siempre aporta nuevos conocimientos y experiencias que pueden ser muy útiles.

No hay que dejar de lado que el hecho de enfrentarse a un TFG de estas características y para un cliente muy exigente y a veces poco organizado, es un reto más al que enfrentarse ya que gran parte del tiempo empleado ha estado dedicado al estudio de la nueva tecnología y a la adaptación de esta al módulo a desarrollar. Cabe decir que gracias al equipo de Pista Virtual, este proceso ha sido más fácil de cumplir ya que ellos tenían un amplio conocimiento de estas herramientas de trabajo y nos han recibido siempre sin ningún problema. En ciertas ocasiones también ha sido un obstáculo el hecho de que ellos, ya expertos en las herramientas, tuviesen exigencias y especificaciones técnicas de cara a la integración del aplicativo en su aplicación en el futuro

A grandes rasgos, el principal problema que nos hemos encontrado en esta tecnología es la conexión del contenido situado en la parte lógica de la aplicación con la parte visual Xaml debido a la existencia de varios métodos de realización. Esto a priori puede ser una ventaja ya que no te limita el cómo hacerlo pero hay que tener en cuenta que dependiendo de la situación y el tipo de contenido que se quiera lanzar, puede haber restricciones y los métodos no siempre ser válidos.

A la hora del aprendizaje del lenguaje, sobre todo en la parte visual del *framework* nos hemos encontrado con una gran cantidad de etiquetas de atributo con infinidad de funcionalidades, lo cual es fantástico ya que esto te permite realizar una gran cantidad de cosas muy interesantes y diferentes, pero te limita si eres nuevo con la tecnología ya que te llegas a perder e incluso frustrar, de todo lo que tiene y a veces no encontrar lo que estás buscando. A su favor, diremos que la documentación que ofrece Microsoft es muy amplia y está bien redactada, aunque las traducciones al español dejan mucho que desear.

En segundo lugar, como hemos comentado en la introducción de esta sección, vamos a contar las experiencias y sensaciones de haber realizado el trabajo para la Universidad Complutense de Madrid y con PistaVirtual como intermediarios en la mayoría de las ocasiones.

En general, podemos decir que la experiencia ha sido buena y positiva para nosotros. Estamos contentos y satisfechos en poder decir que hemos realizado un trabajo de utilidad para la universidad la cual ha sido nuestra casa durante esta etapa de nuestra vida y ese fue el principal motivo por el que decidimos realizar este trabajo de fin de grado.

Con respecto al compromiso y seriedad de la organización encargada de los deportes en la universidad, sí que podemos decir que estamos un poco descontentos ya que en un principio y según nos hicieron saber, estaban totalmente comprometidos y con actitud positiva, pero a lo largo del año esas ganas parece que desaparecieron excepto para el momento final en el mes de mayo cuando las prisas aparecieron al ver que la temporada de verano empezaba y sabían poco de la aplicación.

Con respecto a los requisitos iniciales podemos decir que han sufrido, en nuestra opinión, demasiadas modificaciones a lo largo del desarrollo. Tan pronto el sistema de entrada iba a ser con tornos como de repente no querían un control de aforo. Esto no es un problema si se notifica con antelación, pero teniendo en cuenta que la mayoría de cambios han sido notificados en el último mes previo a la apertura de las instalaciones no se pueden satisfacer todas las especificaciones.

Esto supone que llegue la fecha de apertura y que la aplicación y el nuevo sistema de entrada puede que no estén listos según estos últimos requisitos. Por nuestra parte, y en referencia las primeras especificaciones y algunas posteriores (con un tiempo de reacción lógico a los plazos de entrega de este trabajo) han sido cumplidas. En general, y en nuestra opinión, la gestión por parte de las instituciones de la universidad no ha sido las adecuadas para el desarrollo de software.

Por parte de PistaVirtual, solo tenemos palabras de agradecimiento tanto por su siempre disponibilidad, compromiso o amabilidad, como por su implicación y ayuda que nos han prestado. La dinámica de trabajo ha sido muy completa, con varias reuniones al mes, comunicaciones continuas tanto para dudas técnicas como para información llegada desde la universidad.

7.2. Trabajo futuro

Para este proyecto se pueden plantear distintos trabajos futuros.

- Uno de los más inmediatos podría ser el acceso a la piscina de invierno mediante la aplicación, ya que como ha estado pasando en la de verano se accede mediante un bono o una tarjeta. Esta tarjeta lleva escrita en su parte trasera el acceso de días y horas a las que el usuario puede pasar. Insertar esta piscina en la aplicación supondría una mejora en el control de acceso y poder obtener toda la información necesaria de los distintos grupos de clases en una misma plataforma.
- Otra mejora sería el acceso mediante tornos, cuando comenzamos este trabajo nos dijeron que tenían unos tornos comprados desde hacía tiempo para el acceso a las piscinas, pero que nunca se habían llegado a utilizar, si finalmente se implanta la piscina de invierno podría accederse a ella mediante los tornos, ya que los tienen comprados pero en desuso. Igualmente estos tornos podrían utilizarse mediante el carnet universitario o mediante la tarjeta de acceso de los usuarios con un código de barras.
- Ampliaciones que la aplicación podría tener pueden ser la implantación de devoluciones, por si finalmente se cancela el plan de ir a la piscina.
- Poder reservar la entrada para el acceso a la piscina con varios días de antelación.
- Añadir distintos idiomas a la aplicación para los estudiantes Erasmus.

8. Conclusions and Future work

To conclude, we present below some conclusions that range from the point of view of working with a new technology, to the sensations of working for such an important client as the Complutense University of Madrid.

We will also focus on the possible applications that the implemented module can provide in the swimming pool facilities.

8.1. Conclusions

As Bachelor degree thesis, we can say that it has been a satisfying and rewarding experience since the fact of learning a new development technology, as in this case with .NET and Xamarin, always brings new knowledge and experiences that can be very useful.

We must not forget that the fact of facing a project of these characteristics, for a very demanding and sometimes not very organized client, is a challenge. A great part of the time spent has been devoted to the study of the new technology and the adaptation of this to the module under development. We must say that thanks to the Pista Virtual team, this process has been easier to fulfill since they had a wide knowledge of these work tools and they have always received us without any problem. On certain occasions, it has also been an obstacle the fact that they, already experts in the tools, had requirements and technical specifications for the integration of the application in the future that we might not be able to understand.

Broadly speaking, the main problem we have found in this technology is the connection of the content located in the logical part of the application with the visual part Xaml due to the existence of several methods of realization. This a priori can be an advantage because it does not limit you how to do it but you have to keep in mind that depending on the situation and the type of content you want to launch, there may be restrictions and the methods may not always be valid.

At the time of learning the language, especially in the visual part of the framework we have found a lot of attribute labels with many features, which is great since this allows you to do a lot of very interesting and different things, but it also limits you if you are new with technology because you get lost and even frustrated of how extensive it is and sometimes you cannot find what you are looking for. In its favor, we will say that the documentation offered by Microsoft is very broad and well written, although the translations into Spanish don't meet expectations.

Secondly, as we have said in the introduction of this section, we are going to tell you about the experiences and sensations of having done the work for the Complutense University of Madrid and with PistaVirtual as intermediaries in most of the occasions.

In general, we can say that the experience has been good and positive for us. We are happy and satisfied to say that we have done a useful job for the university which has been our home during this phase of our life and that was the main reason why we decided to do this final degree project.

In reference to the commitment and seriousness of the organization in charge of sports in the university, we can say that we are a little bit unhappy since at the beginning and as they let us know, they were totally committed and with positive attitude, but throughout the year those hopes seem to have disappeared except for the final moment in the month of May when the rush appeared because the summer season was about to begin and they knew little about the application.

With respect to the initial requirements we can say that they have suffered, in our opinion, too many modifications throughout the development. As soon as the entry system was going to be with lathes as suddenly they did not want a gauging control. This is not a problem if the developer is notified in advance, but considering that most changes have been notified in the last month, just before the opening of the facilities, it is impossible to implement it with the appropriate quality assurance.

This assumes that the opening date arrives and that the application and the new entry system may not be ready according to these latter requirements. For our part, and in reference the first specifications and some later ones, (with an adequate reaction time to the deadlines of this work) have been accomplished. In general and in our opinion, the management by the institutions of the university are not the ones expected for an adequate software development experience.

On the part of PistaVirtual, we only have words of thanks for their always availability, commitment, and kindness, as well as for their involvement and help that they have given us. The work dynamics has been very complete, with several meetings per month, continuous communications for both technical questions and information coming from the university.

8.2. Future Work

For this project different future works can be proposed.

- One of the most immediate could be the access to the winter pool through the application, since as it has been happening in the summer is accessed by a voucher or a card. This card has written on the back the access of days and hours to which the user can pass. Inserting this pool in the application would imply an improvement in the access control and be able to obtain all the necessary information of the different groups of classes in a same platform.
- Another improvement would be the access through lathes, when we began this work they told us that they had some lathes bought a long time ago for accessing the pools, but they had never been used, if finally the winter pool was implanted it could be accessed by them since they have bought them but they are in disuse. Likewise these lathes could be used through the university card or through the access card of users with a bar code.
- Extensions that the application could have could be the implementation of returns, in case the plan to go to the pool is finally canceled.
- To be able to book the entrance to the pool several days in advance.
- Add different languages to the application for Erasmus students.

Bibliografía

- [1] https://es.wikipedia.org/wiki/Microsoft_.NET
- [2] <https://docs.microsoft.com/es-es/xamarin/cross-platform/get-started/introduction-to-mobile-development>
- [3] <https://docs.microsoft.com/es-es/xamarin/get-started/>
- [4] https://es.wikipedia.org/wiki/Licencia_de_software#Licencia_de_software_de_c%C3%B3digo_abierto_robustas
- [5] <https://velneo.es/licencias-software-libre-propietario-otros/>
- [6] <https://docs.microsoft.com/es-es/windows/uwp/porting/desktop-to-uwp-migrate>
- [7] <https://github.com/Redth/ZXing.Net.Mobile/tree/master/Samples/Forms>
- [8] <https://docs.microsoft.com/es-es/dotnet/api/system.windows.data.binding?view=netframework-4.8>
- [9] <https://www.youtube.com/watch?v=5sWJIGkUPsA>