

POOLSTORE

Supermercado virtual de productos de piscina con códigos QR

Trabajo Final de Carrera - TFC

Alumno : Roberto Carlo Carbajal Vásquez

Ingeniería Técnica en Informática de Gestión

Consultor : Marc Domingo Prieto

Enero 2013

A Camucha mi esposa, y mis 3 hijos, Paulo, Sergi y Robert, a quienes les debo todo en esta vida.

A mis padres Consuelo y Américo y mis hermanos, por estar siempre a mi lado.

Sois la fuerza que tira de mi.

Índice

1. Introducción.

- 1.1. Descripción del proyecto.**
- 1.2. Objetivos.**
- 1.3. Planificación.**
 - 1.3.1. Detalle de actividades.**
 - 1.3.2. Temporización del proyecto.**

2. Viabilidad del proyecto.

- 2.1. Técnica.**
- 2.2. Operativa.**

3. Análisis del sistema.

- 3.1. Plataforma de desarrollo.**
- 3.2. Diagrama de clases.**
 - 3.2.1. Package de la aplicación.**
 - 3.2.2. Diagrama de clases del paquete PoolStore.**
 - 3.2.3. Diagrama de clases del paquete Captura.**
 - 3.2.4. Diagrama de clases del paquete Carrito.**
 - 3.2.5. Diagrama de clases del paquete Pago.**
- 3.3. Diagrama de casos de uso.**
 - 3.3.1. Caso de uso de usuario no autenticado.**
 - 3.3.2. Caso de uso de usuario autenticado.**
- 3.4. Diagrama de estados.**

4. Diseño.

- 4.1. Centrado en el usuario.**
 - 4.1.1. Perfiles de usuario.**
 - 4.1.2. Contextos de uso.**
 - 4.1.3. Análisis de tareas.**
 - 4.1.4. Escenarios de uso.**
- 4.2. Prototipo de interfaces.**

4.2.1. Sketches.

4.2.2. Prototipo horizontal de alta fidelidad.

4.3. Test de usuario.

4.3.1. Pack de preguntas a los usuarios que realizarán el test.

4.3.2. Tareas que los usuarios han de realizar.

4.3.3. Preguntas referentes a las tareas.

5. Desarrollo e Implementación.

5.1. Definición de estructura de datos.

5.2. Desarrollo de interfaces.

5.3. Programación de módulos.

5.4. Implementación y puesta en marcha.

6. Conclusiones.

7. Bibliografía.

1. Introducción.

1.1. Descripción del proyecto.

Este proyecto nace de la necesidad de innovar en el sector del comercio electrónico en la multinacional para la que trabajo, llamada Fluidra www.fluidra.com, la cual se dedica a la fabricación y comercialización de productos del sector de la piscina, tanto de piscina pública (polideportivos, clubes, gimnasios, etc.) así como piscina residencial.

El proyecto denominado “**Supermercado virtual de productos de piscina con códigos QR en Android**” pretende llevar a las distintas ferias del sector así como a los diferentes puntos de venta que tiene esta empresa a nivel mundial, una serie de paneles fotográficos, comúnmente llamados lineales, con las fotografías de los diferentes productos con los precios y asociados a cada uno de ellos su código QR. También se podrá distribuir los códigos QR a través de los diversos catálogos que se distribuyen en los canales de la red comercial.

A partir de la captura del código QR por la aplicación desarrollada en Android, se podrá acceder al producto, en el cual se podrá seleccionar la cantidad a comprar y añadirla al carrito de compra.

Una vez se hayan seleccionado todos los productos que se desea comprar, se podrá proceder al proceso de pago con un sistema bancario o paypal, el cual permite realizar la operación con la máxima seguridad en el proceso de compra.

Después de que se haya realizado el pago correctamente, el sistema transferirá este pedido a los sistemas de la empresa para realizar su entrega al domicilio del cliente en los plazos previstos.

1.2. Objetivos.

Los objetivos que se persiguen con este proyecto, cumplen una doble función, por un lado, la académica y por otro, la transferencia de conocimiento universidad – empresa, ya que esta aplicación tiene un carácter eminentemente práctico y de rápida integración en el funcionamiento diario de la empresa para la cual se desarrolla.

En el apartado académico, permite poner en práctica los diferentes conocimientos adquiridos durante el estudio de la ingeniería técnica en informática de gestión, como son :

-
- Conocimientos de ingeniería del software.
 - Conocimientos de lenguajes de programación.
 - Conocimientos de gestión de bases de datos.
 - Conocimientos de redes y telecomunicaciones.
 - Conocimientos de administración de servidores de internet, como servidores web, servidores dns, servidores ftp, etc.
 - Conocimientos de diseño centrado en el usuario.
 - Conocimientos de edición web.
 - Conocimientos de programación de dispositivos móviles.
 - Conocimientos de edición y redacción de informes, etc.

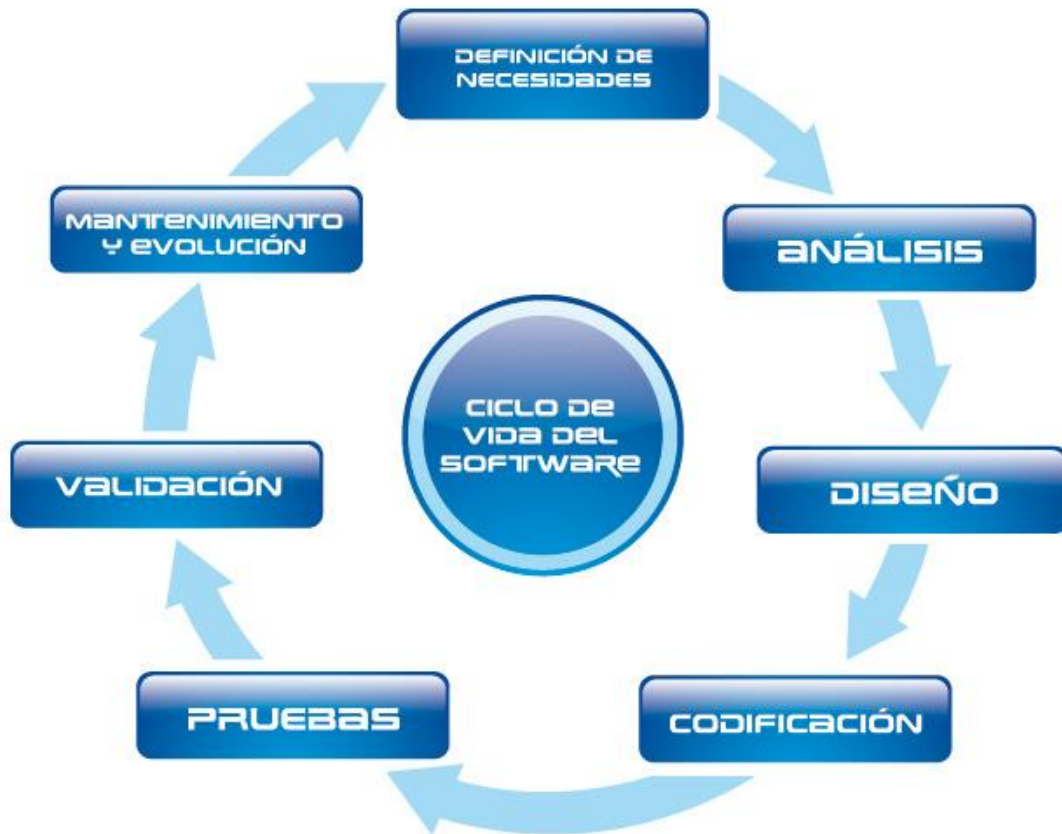
En el apartado de transferencia de conocimiento universidad-empresa, se cumple una de las premisas básicas de la interrelación de estas 2 instituciones, cumpliendo así el objetivo de su razón de ser, que es la de poder formar profesionales que lleven el know-how tecnológico de la universidad a la empresa y se pongan en marcha proyectos que las empresas necesitan.

1.3. Planificación.

En esta fase se lleva a cabo, la planificación de actividades en función de las PAC's propuestas para el desarrollo del proyecto, así como la temporización detallada con sus hitos, para llevar a término el trabajo.

1.3.1. Detalle de actividades.

Durante el proceso del proyecto de TFC se han realizado 3 actividades o PAC's que han permitido avanzar paso a paso cada una de las etapas, interaccionando en todo momento en un ciclo de desarrollo de software, tal como se indica en el diagrama siguiente:



Siguiendo este planteamiento, se han ido trabajando las siguientes PAC's :

PAC1 : Plan de Trabajo

Donde se prepara una descripción detallada de los diferentes apartados de los que está compuesto el proyecto, como son :

- Descripción del proyecto.
- Objetivos.
- Funcionalidades.
- Calendarios.
- Viabilidad.
- Análisis, Diseño e Implementación.
- Riesgos asociados.

PAC2 : Diseño centrado en el usuario

En esta entrega se trabaja la parte de diseño desde un punto de vista de usuario, analizando los perfiles de usuarios, tipos de usos, y contextos de trabajo, realizando una entrega que consta de estos puntos :

- Usuarios y contextos de uso.
- Diseño conceptual.

-
- Prototipo.
 - Evaluación.

PAC3 : Implementación

En la tercera PAC, se realiza la programación de la aplicación y el testing de la misma, detallándolo así :

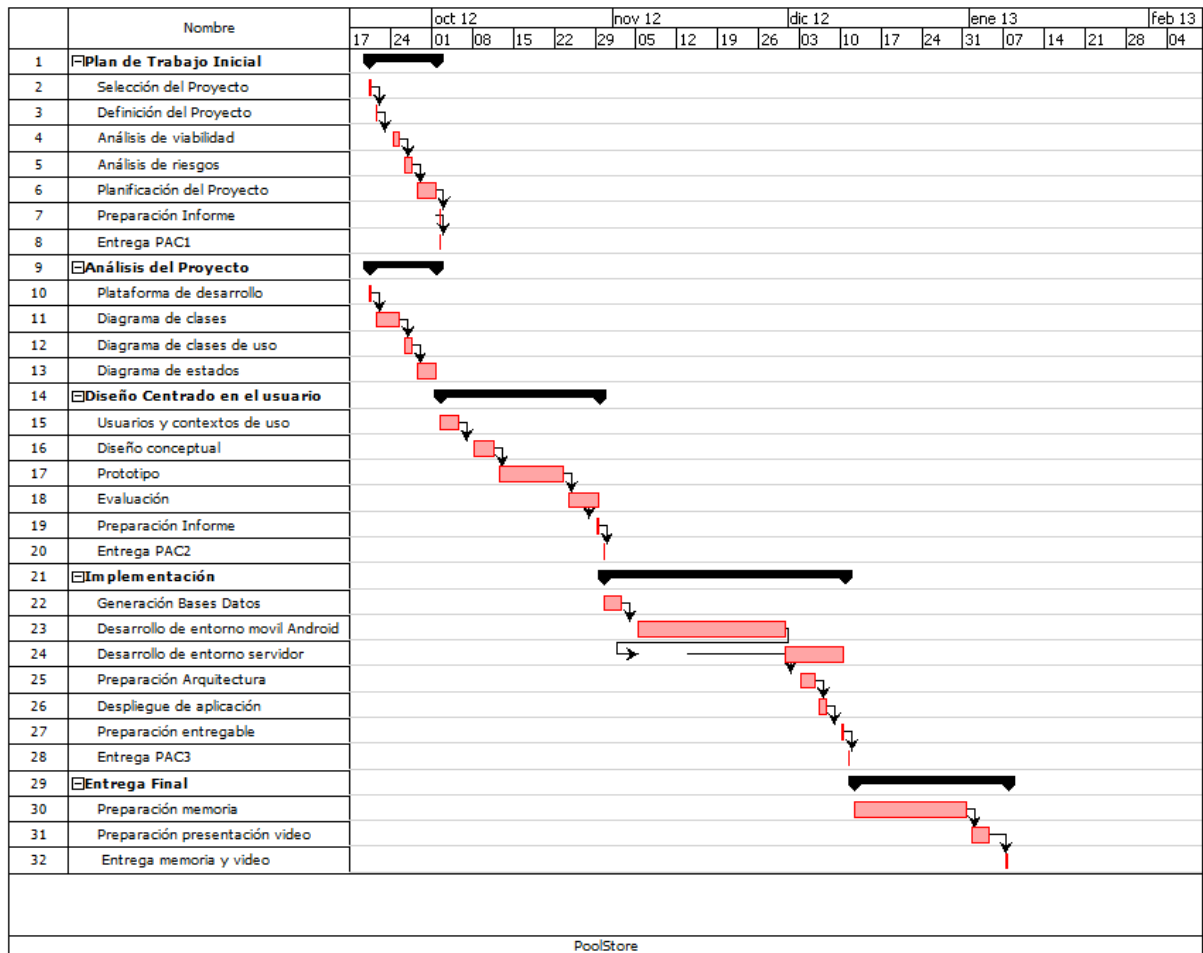
- Generación de la base de datos.
- Programación de la aplicación.
- Implementación de la aplicación, tanto de la parte móvil como la de servidor.
- Documentación de la arquitectura utilizada.

1.3.2. Temporización del proyecto.

A continuación se muestra el calendario de trabajo para llevar a cabo la realización de este proyecto:

Tarea	Duración	Inicio	Fin
Plan de Trabajo Inicial	8 días	20/09/2012	01/10/2012
Selección del Proyecto	1 día	20/09/2012	20/09/2012
Definición del Proyecto	1 día	21/09/2012	21/09/2012
Análisis de viabilidad	2 días	22/09/2012	25/09/2012
Análisis de riesgos	2 días	26/09/2012	27/09/2012
Planificación del Proyecto	2 días	28/09/2012	01/10/2012
Preparación Informe	1 día	30/09/2012	01/10/2012
Entrega PAC1	1 día	01/10/2012	01/10/2012
Análisis del Proyecto	8 días	20/09/2012	01/10/2012
Plataforma de desarrollo	1 día	20/09/2012	20/09/2012
Diagrama de clases	3 días	21/09/2012	25/09/2012
Diagrama de clases de uso	2 días	26/09/2012	27/09/2012
Diagrama de estados	2 días	28/09/2012	01/10/2012
Diseño Centrado en el usuario	20 días	02/10/2012	29/10/2012
Usuarios y contextos de uso	4 días	02/10/2012	05/10/2012
Diseño conceptual	4 días	06/10/2012	11/10/2012
Prototipo	8 días	12/10/2012	23/10/2012
Evaluación	4 días	23/10/2012	26/10/2012
Preparación Informe	1 día	27/10/2012	29/10/2012
Entrega PAC2	1 día	29/10/2012	29/10/2012
Implementación	30 días	30/10/2012	10/12/2012
Generación Bases Datos	4 días	30/10/2012	02/11/2012
Desarrollo de entorno móvil Android	20 días	02/11/2012	29/11/2012
Desarrollo de entorno servidor	7 días	05/11/2012	13/11/2012
Preparación Arquitectura	3 días	01/12/2012	05/12/2012
Despliegue de aplicación	2 días	06/12/2012	07/12/2012
Preparación entregable	1 día	08/12/2012	10/12/2012
Entrega PAC3	1 día	10/12/2012	10/12/2012
Entrega Final	19 días	12/12/2012	07/01/2013
Preparación memoria	14 días	12/12/2012	31/12/2012
Preparación presentación video	4 días	01/01/2013	04/01/2013
Entrega memoria y video	1 día	07/01/2013	07/01/2013

Diagrama de Gantt



2. Viabilidad del proyecto.

2.1. Técnica.

Este proyecto desde un punto de vista técnico es viable, al poder utilizar bases de datos, entornos de programación y plataformas de producción utilizando herramientas del mercado, tanto opensource como de pago.

Partiendo de la formación adquirida, como bases de datos tenemos diferentes opciones como puede ser mysql, sqlserver, postgresql u oracle.

En el apartado de desarrollo podemos utilizar netbeans o eclipse.

A nivel de servidores web, podemos utilizar apache como estándar de facto.

En cuanto a dispositivos móviles existe una variedad importante de ellos, basados en Android, a partir de la versión 2.3 Gingerbread.

Por la parte de capturas de códigos QR, si es necesario basa el uso de esta aplicación la utilización de la cámara de fotos que hace las funciones de captura de las imágenes de códigos QR.

2.2. Operativa.

Desde un punto de vista operativo, es necesario contar con la colaboración de los departamentos de la empresa, con el fin de recoger la información necesaria para subir los datos al sistema, así como las fotografías y lineales necesarios para lanzar este producto al mercado.

En el aspecto de dedicación, es un proyecto medianamente complejo, pero asequible a lo largo de un semestre, que permitirá desarrollar la aplicación, testearla y ponerla en producción.

En relación al entorno de desarrollo, testing e implementación se ha optado por desarrollar en local y publicarlo a través de un servicio de hosting gratuito comlu.com que nos permite utilizar las diferentes herramientas necesarias para el testing del proyecto.

A nivel operativo, cuando entre en producción, se podrá desplegar directamente a través del play store de google.

3. Análisis del sistema.

3.1. Plataforma de desarrollo.

Después de realizar el análisis de las diferentes opciones, se ha optado por un entorno de desarrollo opensource, utilizando las siguientes herramientas:

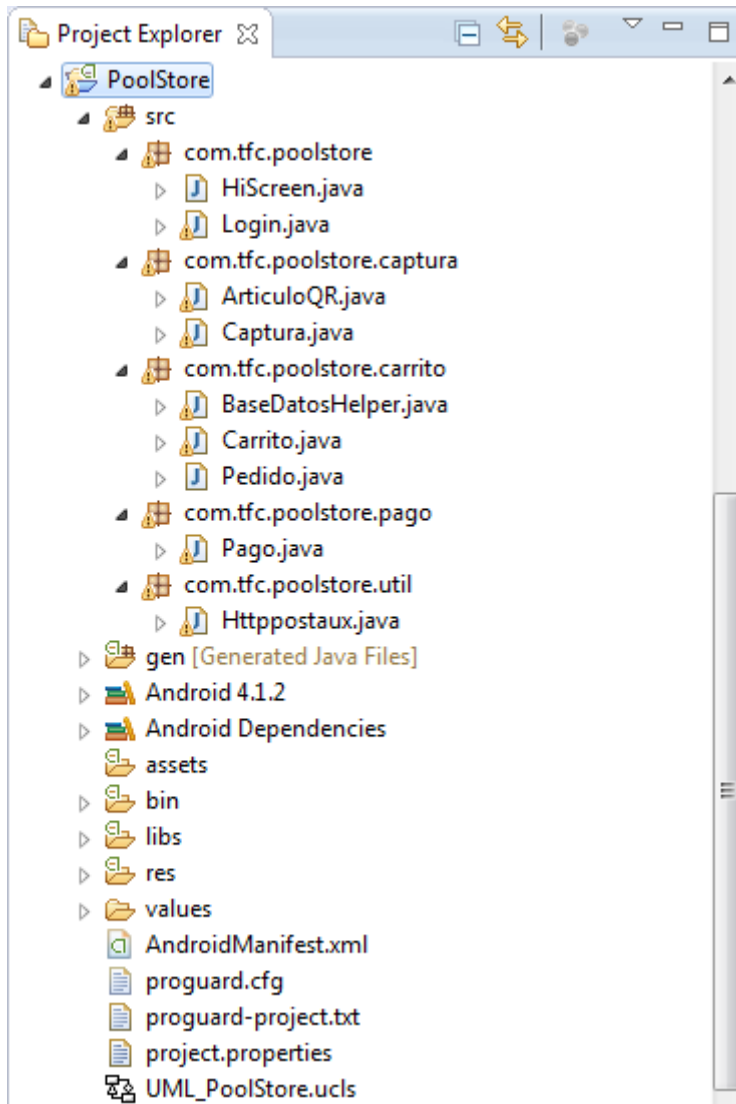
- Base de Datos : mysql
- Entorno Desarrollo Android : Eclipse + plugin Android.
- Editor HTML : CoffeeCup HTML editor
- Servidor Web : Apache.
- S.O : Linux
- Editor PHP : Eclipse + plugin PHP.
-



3.2. Diagrama de clases.

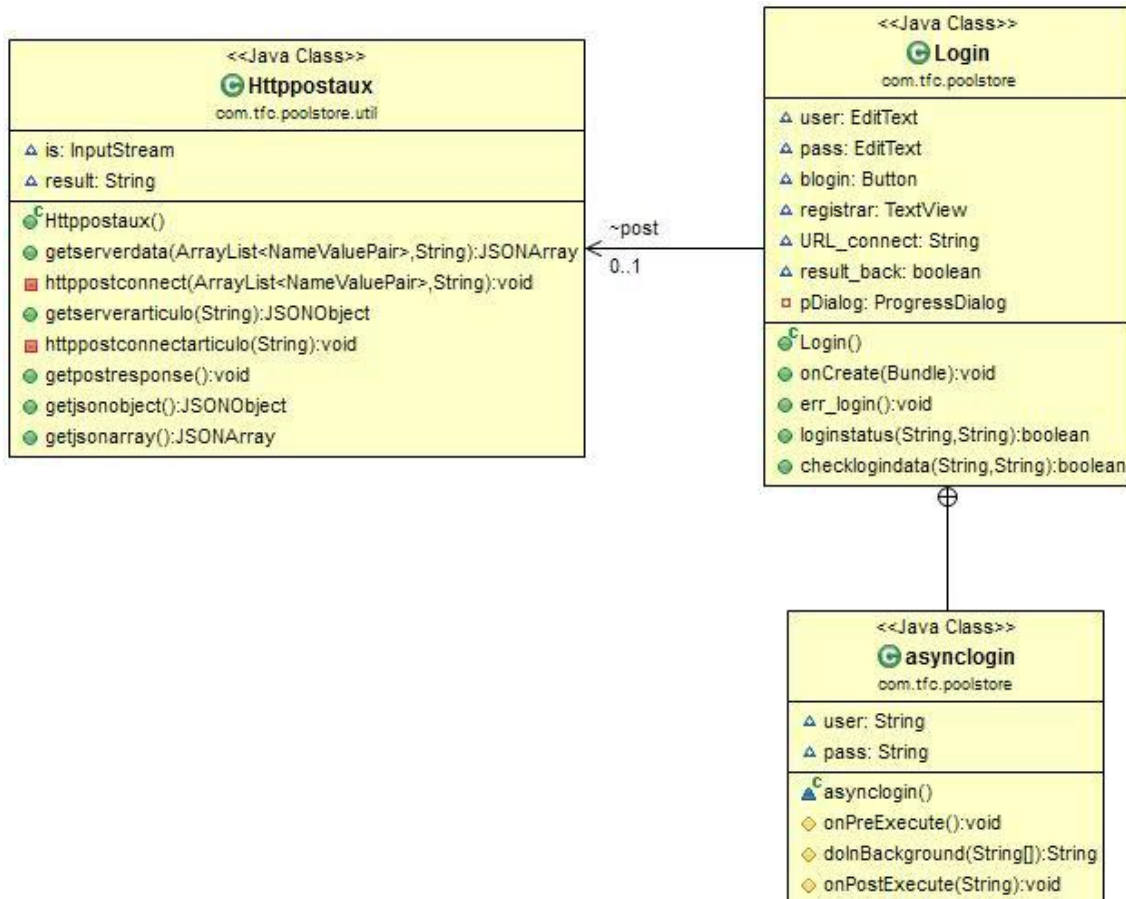
Partiendo de los diferentes paquetes que se han creado para la implementación del proyecto, mostramos a continuación, los diferentes diagramas de clases que se han utilizado para este proyecto.

3.2.1. Package de la aplicación :



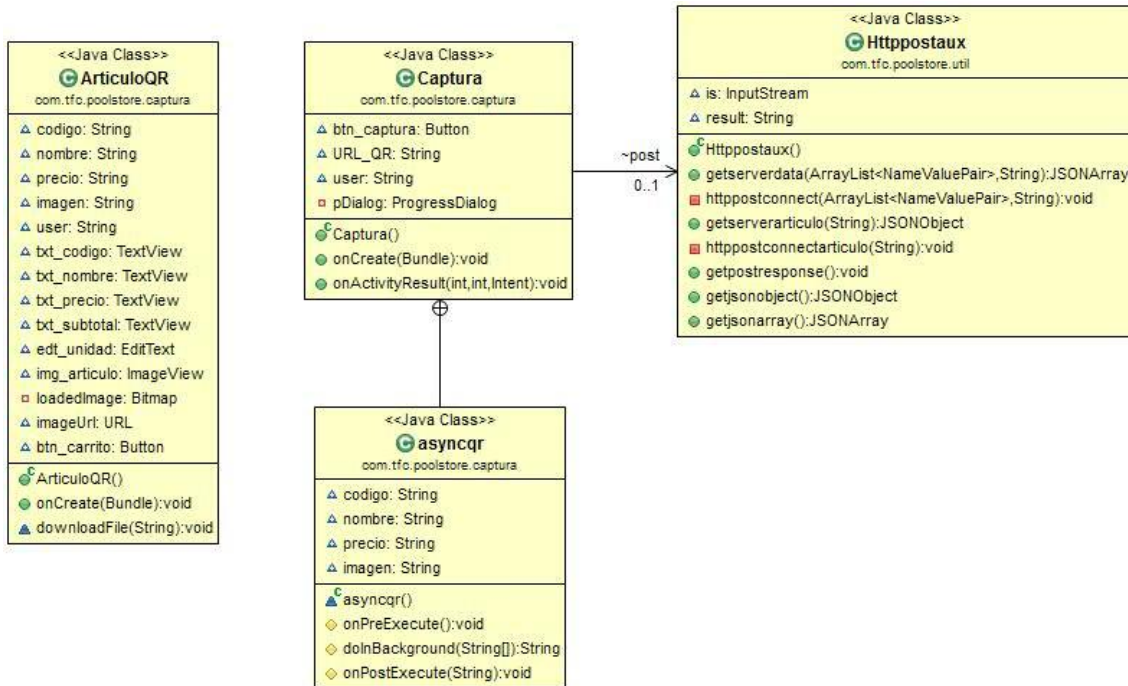
3.2.2. Diagrama de clases del paquete PoolStore:

Que controla el acceso a la aplicación, utilizando asynclogin para la transferencia de las comunicaciones en background, apoyándose en la clase Httppostaux, utilizando objetos JSON.



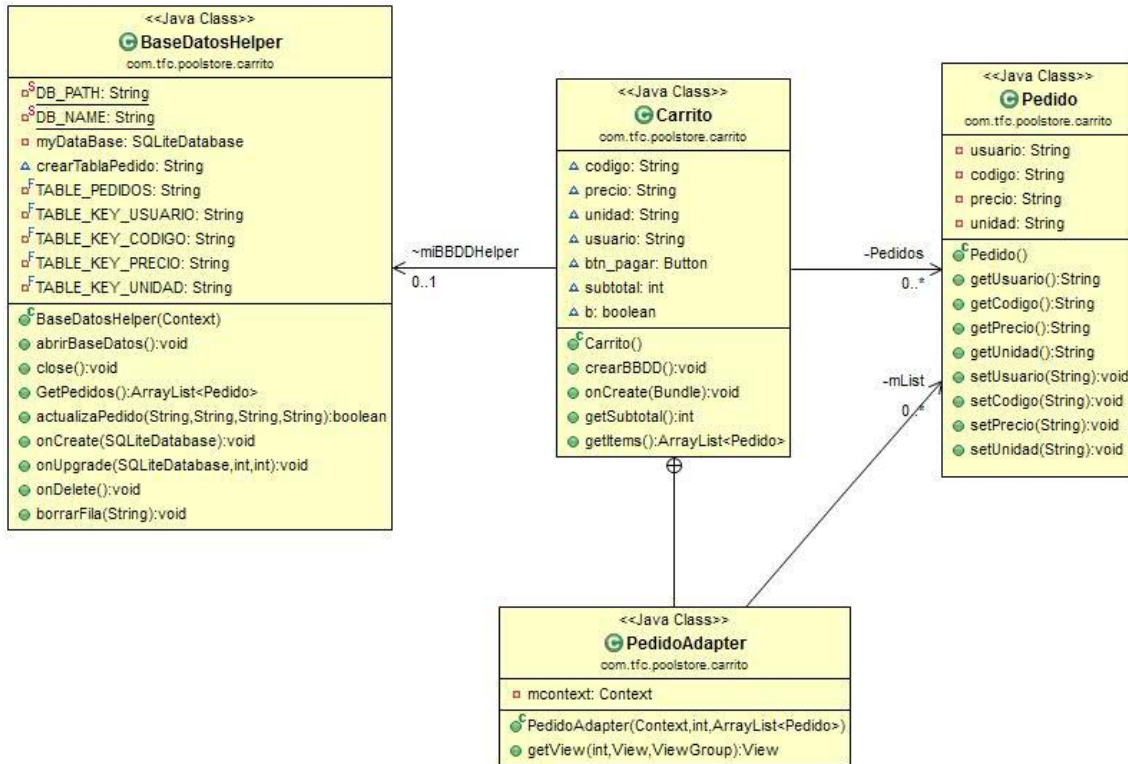
3.2.3. Diagrama de clases del paquete Captura :

Que controla el proceso de captura del código QR, y realiza la comunicación utilizando Httppostaux, de un articuloQR, y comunicándose a través de asyncqr, en background.



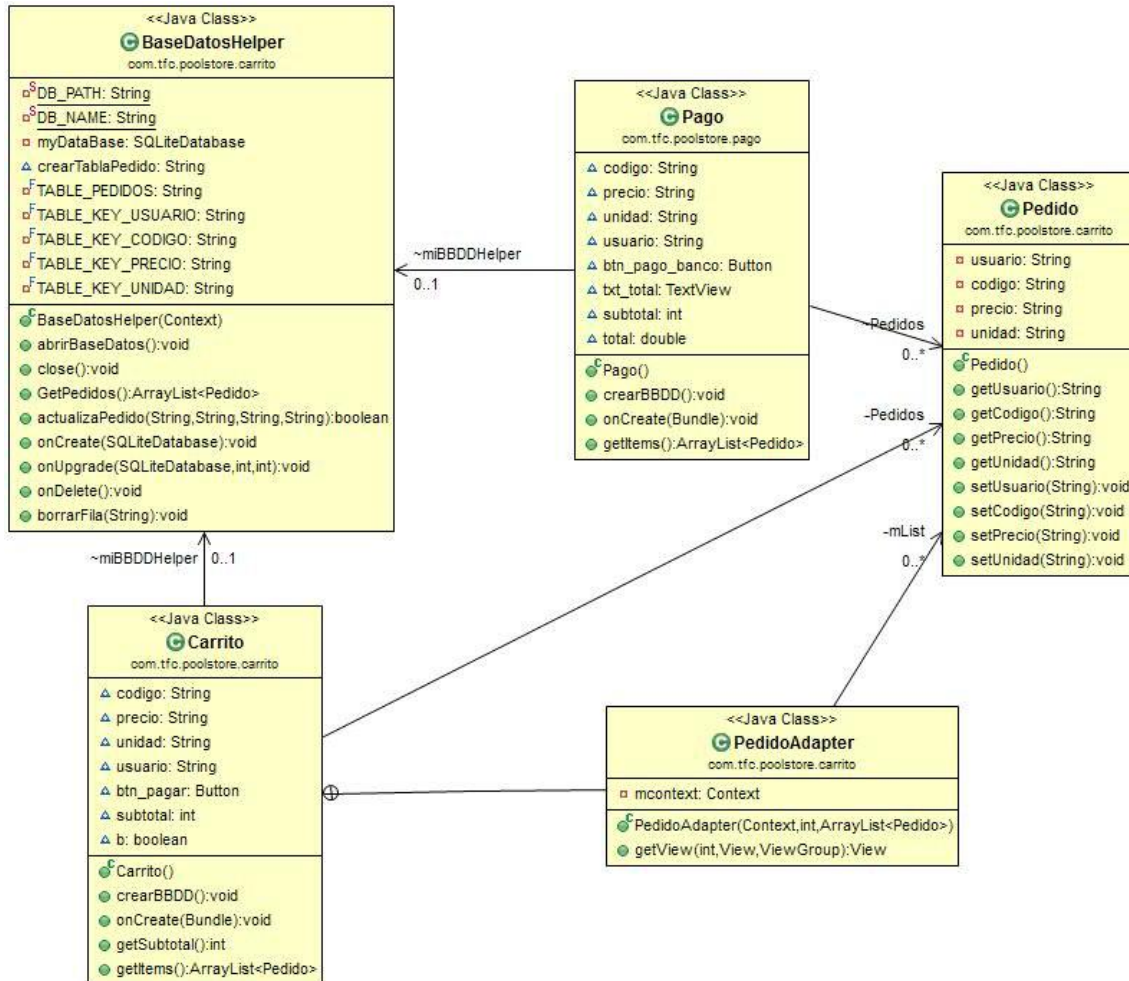
3.2.4. Diagrama de clases del paquete Carrito :

Que se encarga de añadir los pedidos al carrito de compra, después de haber realizado la captura, utilizando la clase Carrito, la clase Pedido, la clase BaseDatosHelper, para acceso a la base de datos SQLite del dispositivo móvil.



3.2.5. Diagrama de clases del paquete Pago :

Que se encarga de realizar de controlar el proceso de pago, y a partir de aquí enviar la información del carrito a la base de datos que se encuentra en la web. Aquí se muestra la clase pago y su interrelación con el resto de paquetes.



3.3. Diagrama de casos de uso.

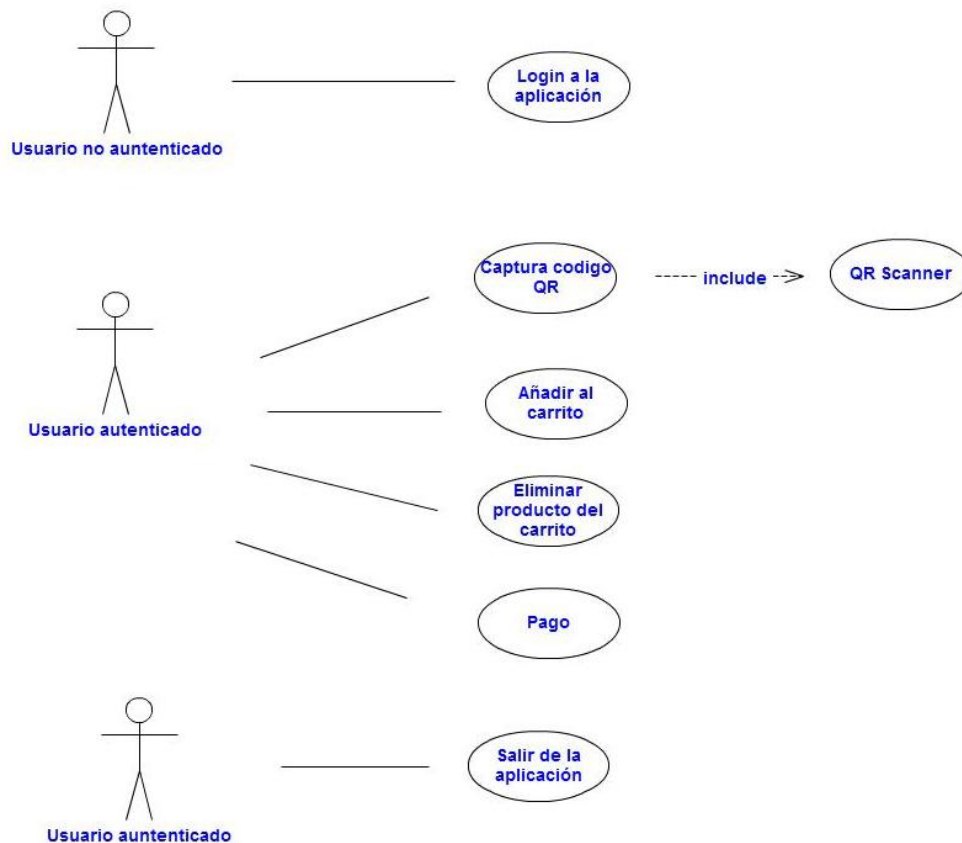
En este apartado, vemos los diferentes casos de uso que puede tener un usuario con la aplicación de PoolStore:

3.3.1. Caso de Uso de usuario no autenticado:

- Login a la aplicación.

3.3.2. Caso de Uso de usuario autenticado :

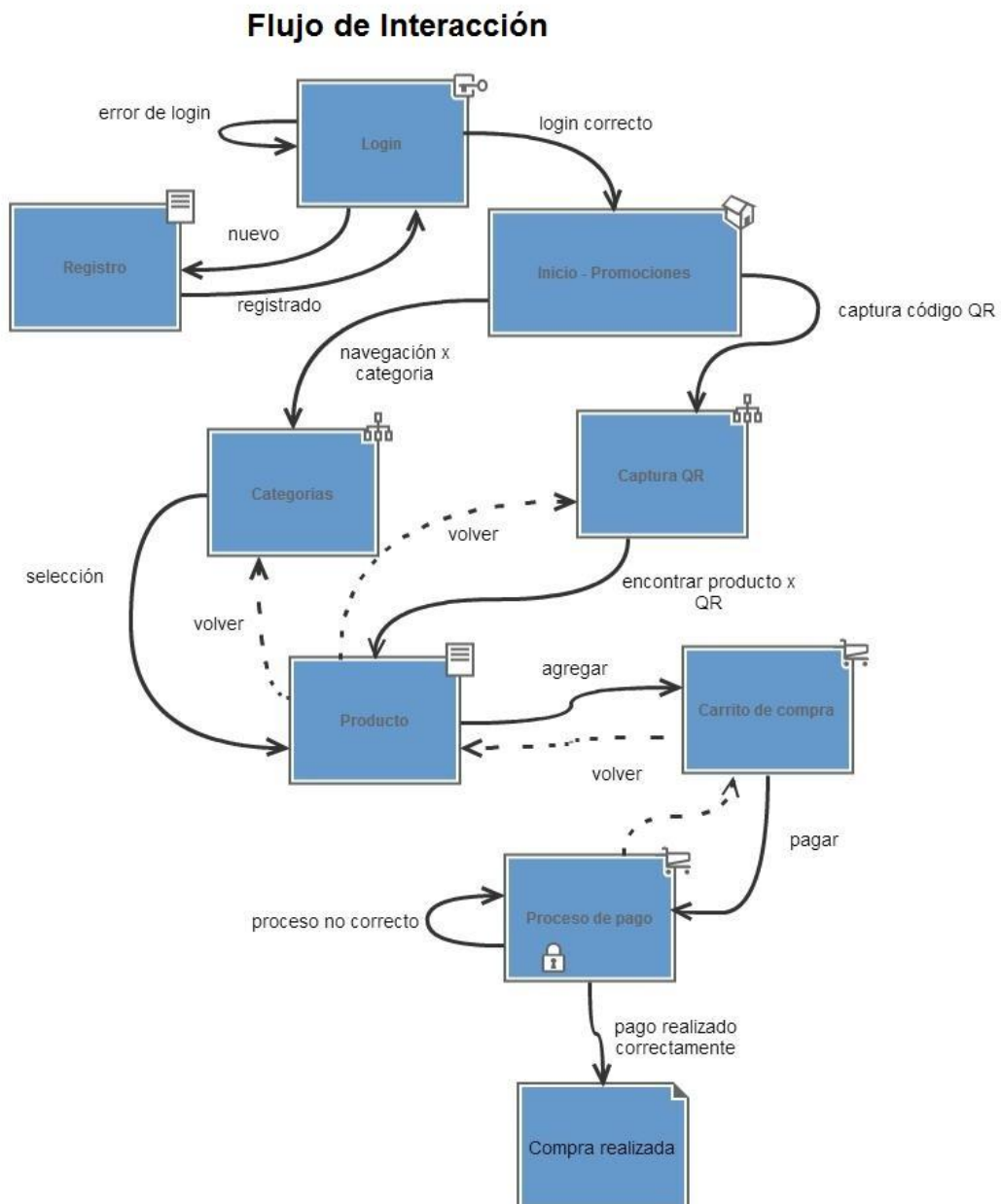
- Captura de código QR.
- Añadir producto al carrito.
- Eliminar producto del carrito.
- Realizar pago.
- Salir de la aplicación.



3.4. Diagrama de estados.

Flujos de Interacción.

En este punto, se describe los diferentes flujos o estados por los que pasa la aplicación en su proceso de interacción con el usuario :



4. Diseño.

4.1. *Centrado en el usuario.*

4.1.1. Perfiles de usuario.

De la metodología de la observación e investigación conceptual en campo, podemos deducir que trabajamos con 2 perfiles de edad definidos :

- De 25 a 45 años
- De 45 a 65 años.

También relacionado con estos grupos, podemos describir 2 grupos tecnológicos :

- Poco uso de tecnología móvil – en la utilización de aplicaciones-
- Uso normalizado de las aplicaciones móviles.

De las encuestas realizadas por las áreas de marketing de la empresa, llegamos a los siguientes datos:

- Un 70% se encuentran en el grupo de edad de 45 a 65 años y el otro 30% se encuentra en el grupo de 24 a 45 años.
- Asociado al dato anterior, podemos decir que el 70% de los perfiles de usuario no está habituado a trabajar con aplicaciones móviles.

Este hecho se desprende de la observación que el perfil profesional en el sector de la piscina viene del sector de la construcción, con muchos años de experiencia en el sector y una media de edad alta que es una barrera para integrarse en la tecnología móvil de forma rápida.

También es cierto que el 30% restante son más jóvenes y habituados al uso de tecnología y aplicaciones móviles. Estos usuarios no tienen problemas para integrarse rápidamente con aplicaciones móviles.

Este 30% es un porcentaje bajo de usuarios pero que va creciendo rápidamente por el tema demográfico en esta franja de edad y que será la generación que viene asumiendo cada vez más, puestos de tomas de decisión de compra, el cual es nuestro usuario clave, ya que esta aplicación está dirigida al profesional del sector y no al público en general.

4.1.2. Contextos de uso.

En los contextos de uso analizados, encontramos diversos contextos, entre otros más genéricos como puede ser las que aporta la movilidad y la ubicuidad, también tenemos los siguientes contextos de uso:

- Cuando los especialistas se encuentran a pie de obra y pueden realizar sus pedidos directamente al sistema.
- Cuando los profesionales se encuentran en las diferentes ferias sectoriales que se desarrollan y donde se hace mucha publicidad de productos con promociones especiales. Desde aquí se puede capturar los códigos QR para realizar el pedido.
- Cuando los profesionales se encuentran en sus oficinas, revisando los productos necesarios para sus proyectos.
- Cuando los profesionales visitan las delegaciones, donde pueden comprar directamente capturando los códigos QR en los paneles.

4.1.3. Análisis de tareas.

Entre las diferentes tareas que los usuarios pueden realizar, se encuentran los siguientes :

- Registro en el sistema.
- Login al sistema.
- Navegación por los menús de categorías.
- Búsqueda de productos.
- Captura de códigos QR.
- Añadir cantidades al carrito.
- Disminuir cantidades del carrito.

- Eliminar productos del carrito.
- Rellenar orden de pago.
- Realizar pago.

4.1.4. Escenarios de uso.

Se definen diversos escenarios de uso, entre los cuales destacamos el entorno de ferias y congresos del sector a nivel mundial, donde se lanzan ofertas importantes en paneles publicitarios con códigos QR. En estos puntos es desde un punto de vista de marketing el que puede atraer a los posibles usuarios-clientes.

Otro de los escenarios, sería las diferentes delegaciones del grupo distribuidas por el mundo, que permiten mostrar todo el abanico de productos que se deseen, sin necesidad de tenerlo físicamente en la delegación.

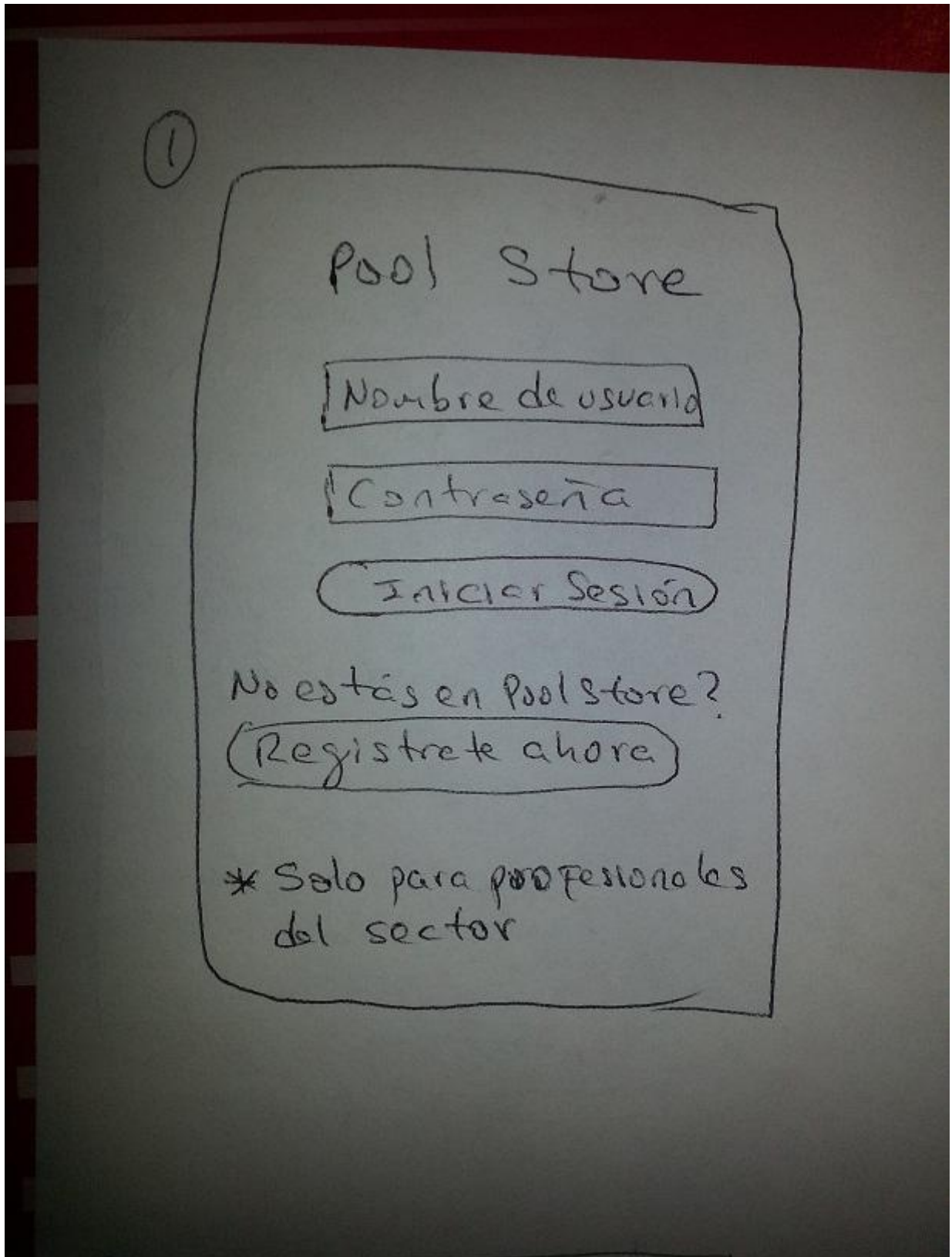
Un tercer escenario, es el entorno laboral del especialista de la piscina, tanto a pie de obra, donde de primera mano es capaz de realizar pedidos a través de la plataforma o desde la propia oficina o domicilio, donde puede navegar y encontrar aquello que busca.

4.2. *Prototipo de interfaces.*

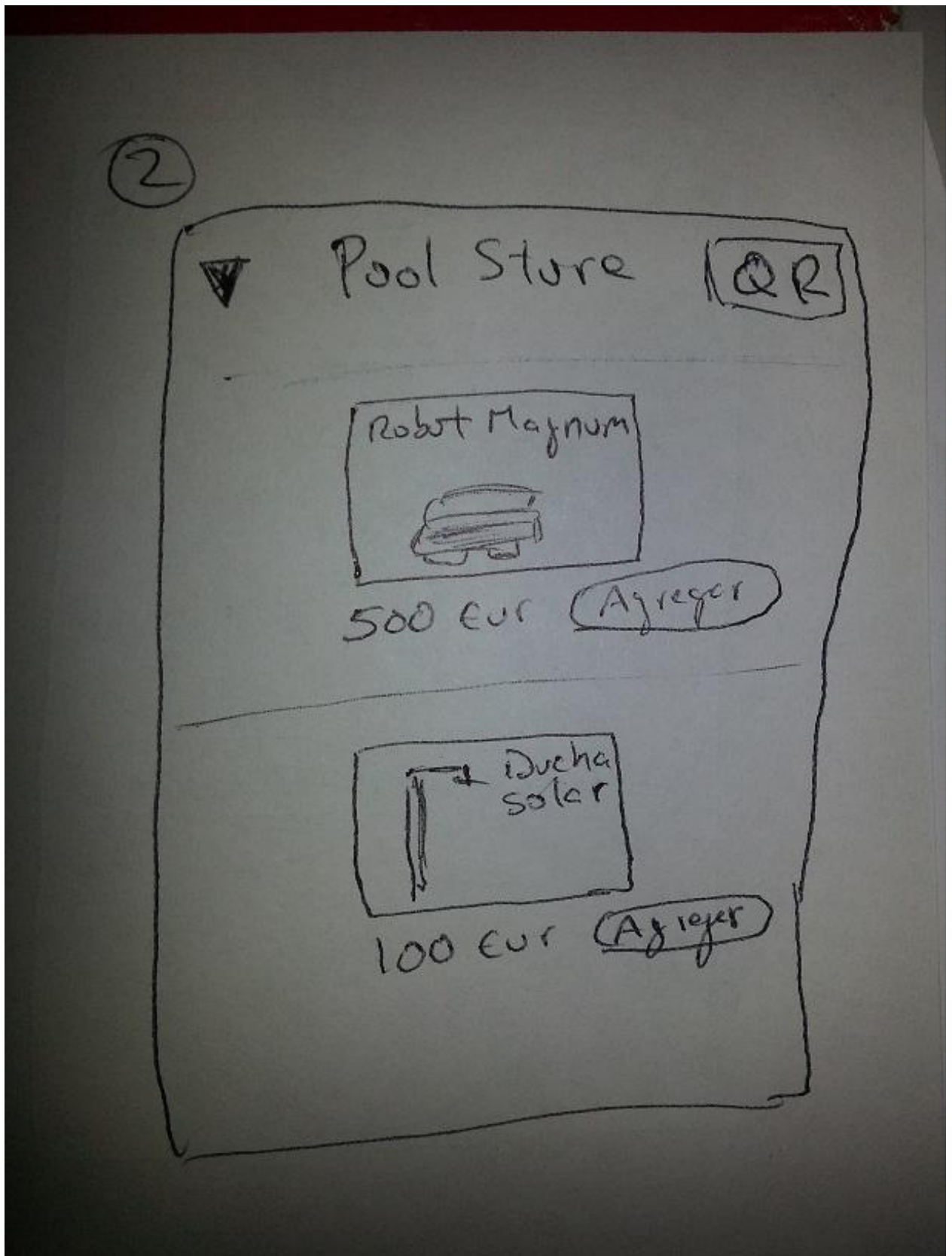
En este apartado aportamos los primeros sketches, así como un prototipado horizontal de alta fidelidad:

4.2.1. Sketches.

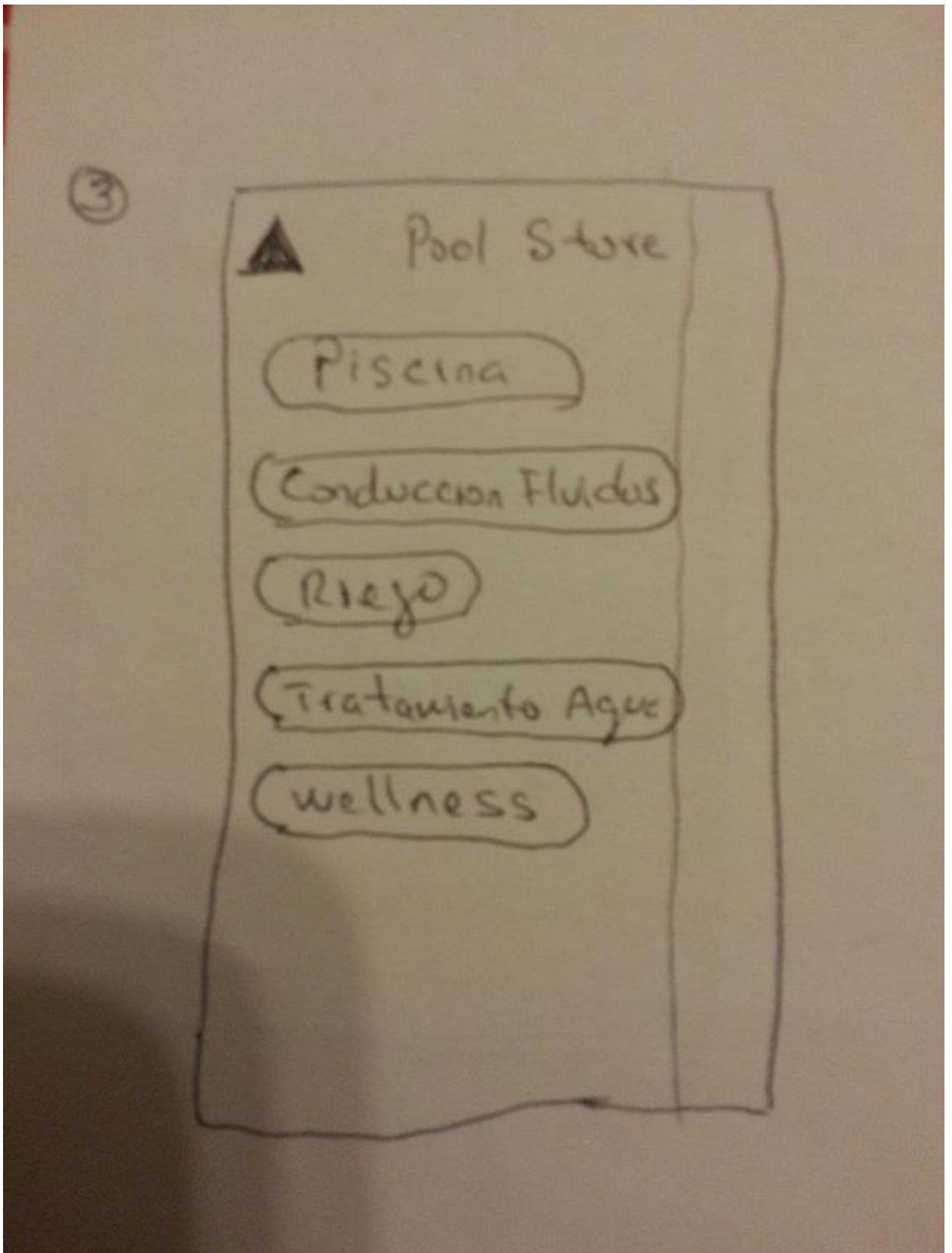
En este apartado se puede visualizar las diversas pantallas trabajadas, tanto por la parte de códigos QR, como por el menú de categorías, así como las promociones en la portada.

Sketch 1 : Pantalla login

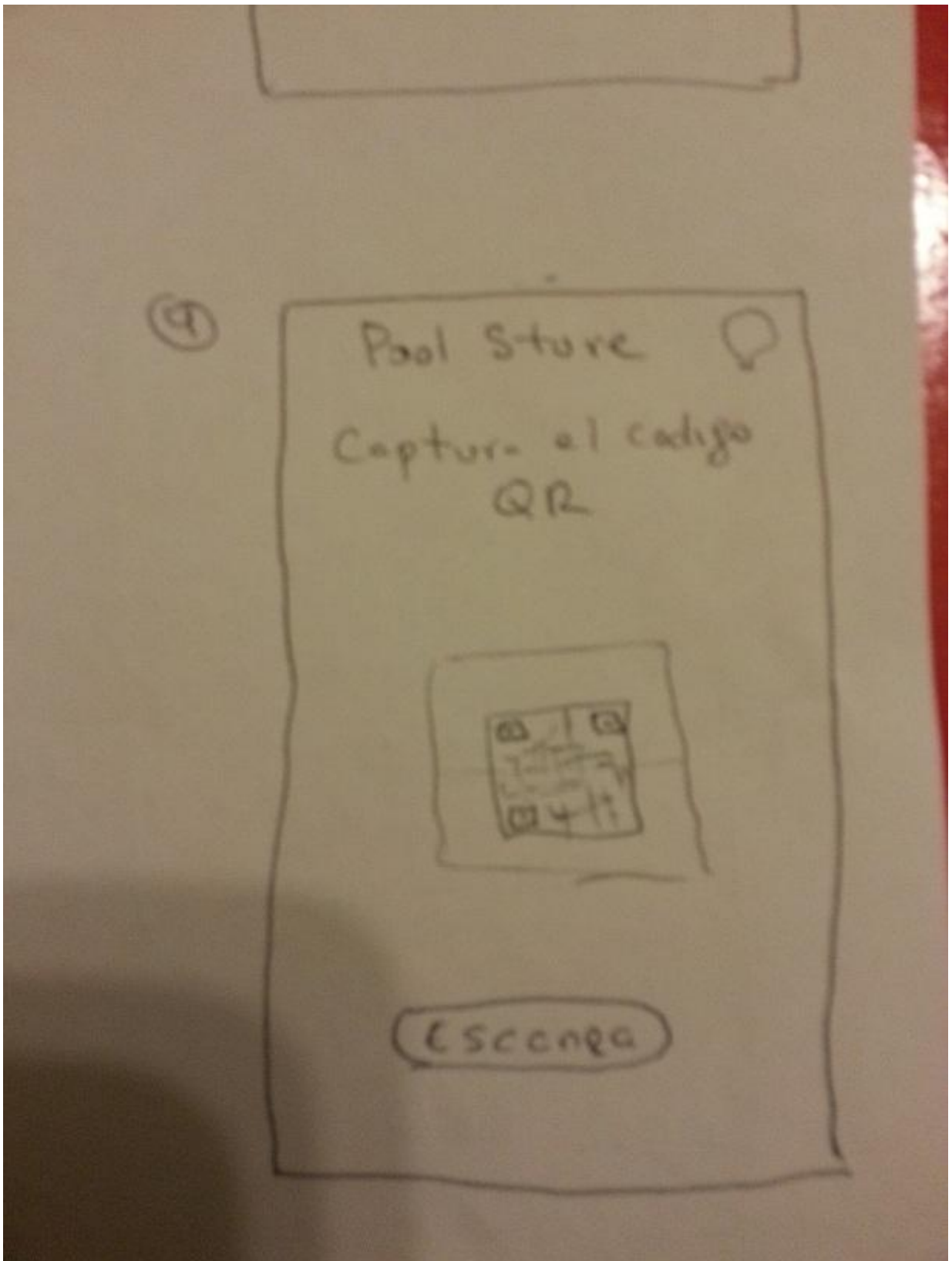
Sketch 2 : Pantalla después login – Promociones

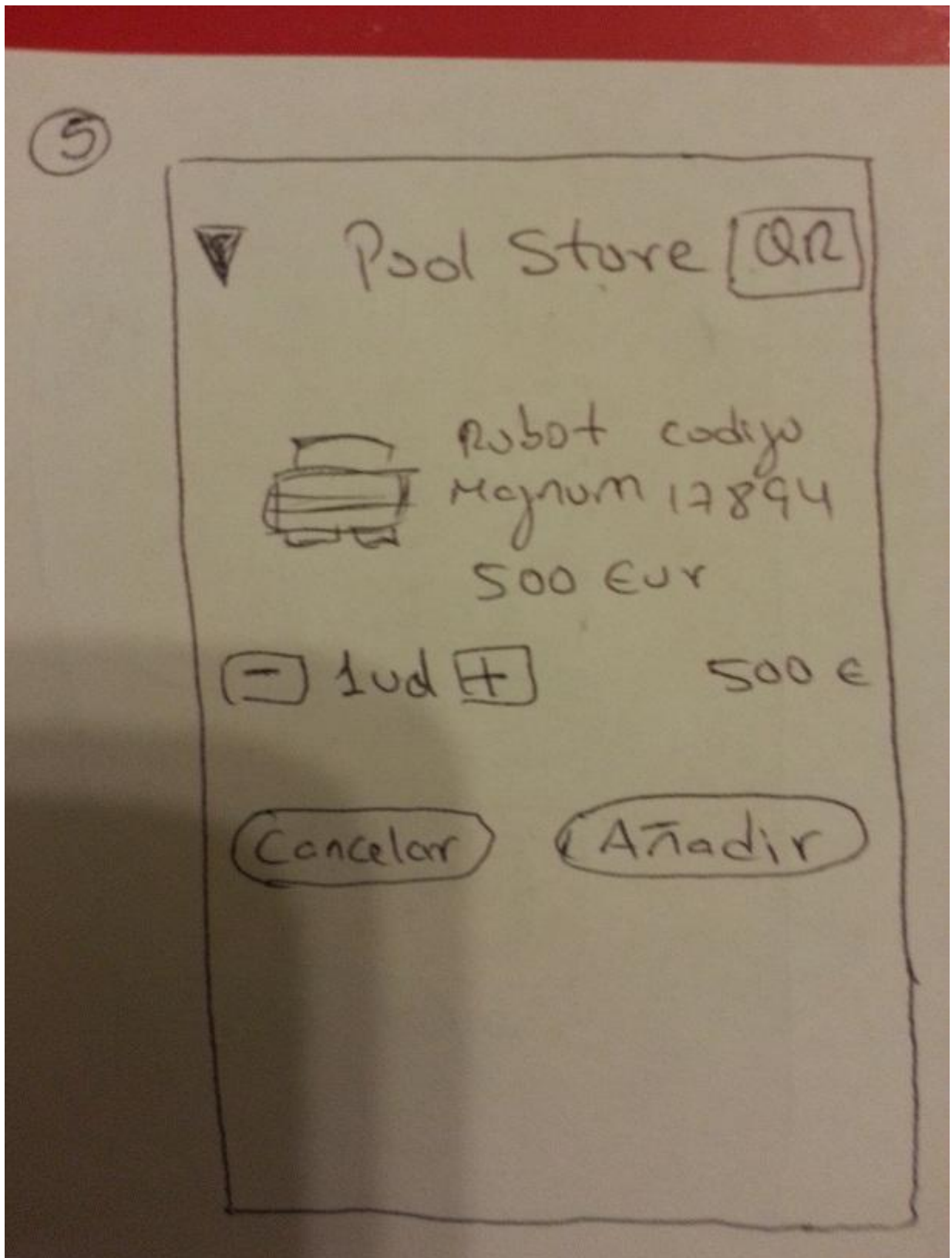


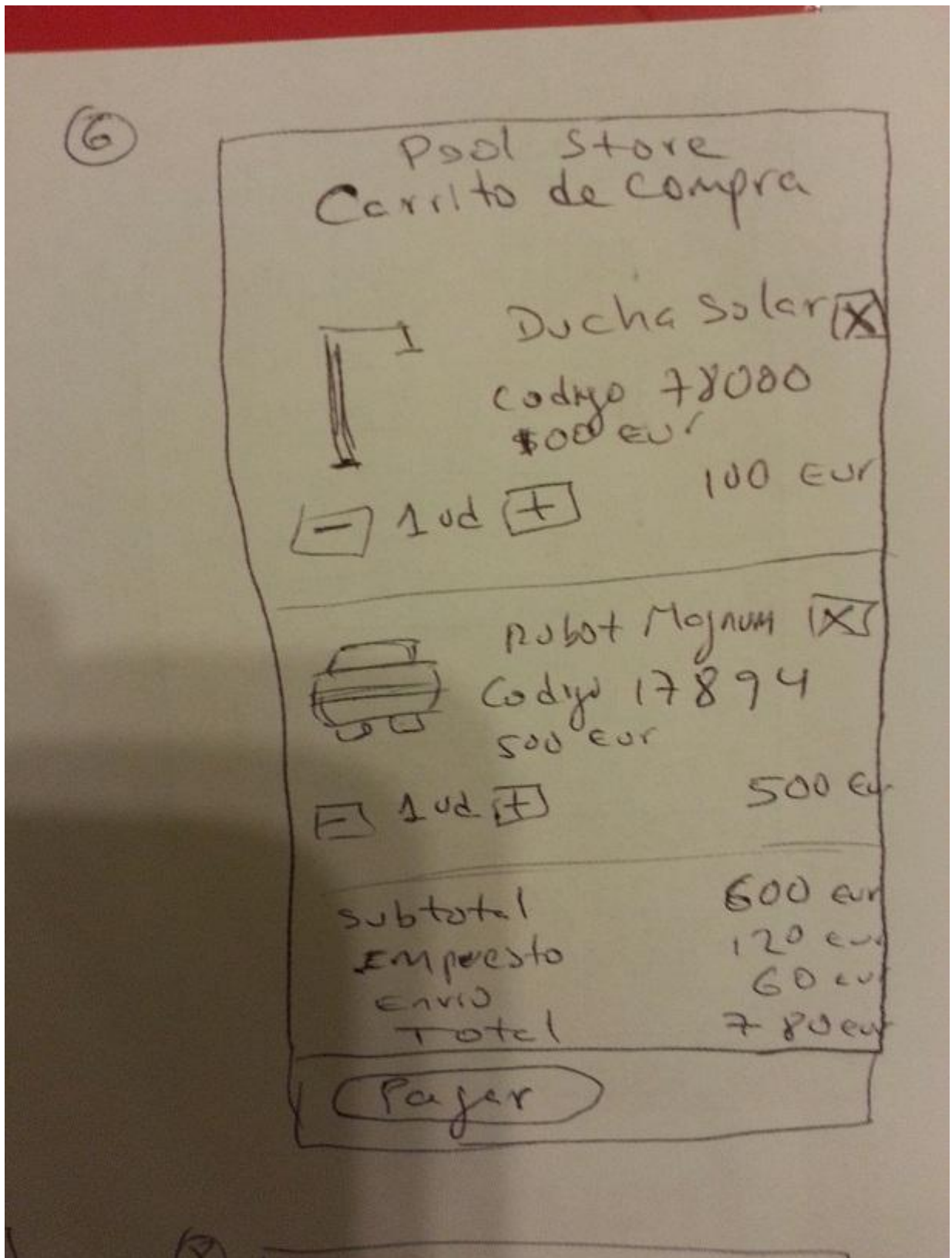
Sketch 3 : Pantalla de Menu Categorias



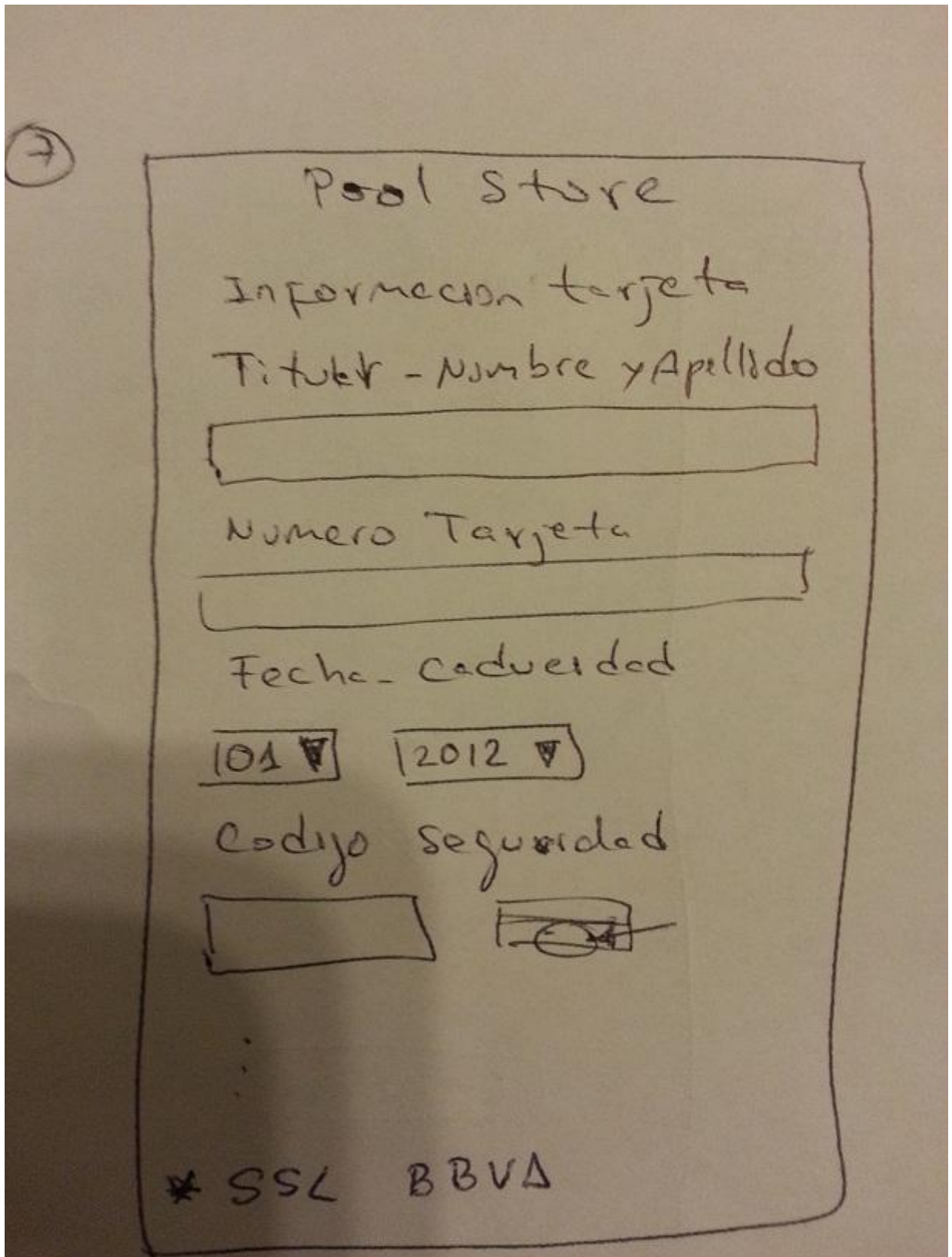
Sketch 4 : Pantalla Captura de códigos QR.



Sketch 5 : Pantalla agregar a carrito.

Sketch 6 : Pantalla de carrito de compra.

Sketch 7: Pantalla Datos Pago (1 pantalla de 2 – hacia abajo)



Sketch 8 : Pantalla Datos Pago (2 pantalla de 2 – hacia abajo)

⑧

Dirección Facturación

Igual que envío

Dirección

ciudad

C.P.

País

España ▼

Subtotal	600 EUR
Impuestos	120 EUR
Envío	60 EUR
Total	780 EUR

Completar Pago

4.2.2. Prototipo horizontal de alta fidelidad.

Basándonos en los pasos previos de análisis de usuarios, contextos de uso, diseño conceptual y sketches, llegamos a este prototipo con Balsamiq Mockup, donde se aprecia de forma horizontal el funcionamiento de la aplicación en sus diferentes pantallas.

Pantalla 1 : Login



Pantalla 2 : Opción captura de código QR de producto.



Pantalla 3 : Agregar producto capturado por QR al carrito

Pantalla 4 : Carrito de compra

Pantalla 5 : Proceso de pago con tarjeta. 1 de 2.



Pantalla 6 : Pantalla de pago con tarjeta. 2 de 2.



4.3. Test de usuario.

4.3.1. Pack de preguntas a los usuarios que realizarán el test.

En este apartado se definen las preguntas que pueden ayudar a conocer las opiniones de los usuarios respecto al DCU de la aplicación.

- Le parece intuitiva la pantalla de acceso ?

-
- Como recargaría productos nuevos ?
 - Sabe lo que es un código QR ?
 - La información del producto es adecuada o necesita más información ?
 - Las pestañas que presenta la aplicación son entendibles ?
 - Le es fácil navegar a través del menú ?
 - La forma de pago le da seguridad ?
 - Echa en falta algún elemento para facilitar el movimiento por las pantallas ?
 - Cree que son muchas pantallas para comprar un producto ?
 - La velocidad de acceso y navegación la encuentra correcta ?

4.3.2. Tareas que los usuarios han de realizar.

En este punto se indicará una serie de tareas que los usuarios deberían poder realizar de forma intuitiva.

- Hacer logín.
- Añadir un producto al carrito.
- Capturar un código QR.
- Aumentar / disminuir la cantidad de productos en el carrito.
- Eliminar un producto del carrito.
- Encontrar un producto.
- Navegar por los menus.
- Llegar al último punto del proceso de compra.
- Rellenar los campos solicitados.
- Encender el flash para capturar el código QR.
- Volver hacia atrás en cualquier menú.
- Salir de la aplicación.

4.3.3. Preguntas referentes a las tareas.

Hacer logín :

- Es intuitivo ?
- Es sencillo ?

Añadir un producto al carrito :

-
- Es un proceso sencillo ?
 - Echa en falta algún elemento en el proceso ?

Capturar un código QR :

- Ha llegado fácil a esta pantalla ?
- Que espera que la aplicación realice con este proceso ?
- Sabe para que sirve el icono de la bombilla ?

Aumentar / disminuir la cantidad de productos en el carrito :

- Es intuitivo el aumentar o disminuir una cantidad ?

Eliminar un producto del carrito.

- Le ha sido fácil ejecutar esta tarea ?

Encontrar un producto.

- Le ha sido fácil encontrar un producto ?

Navegar por los menus.

- Que le parece la forma de navegar por los menus ?

Llegar al último punto del proceso de compra.

- Cree que son pocos o muchos pasos para llegar al último paso del proceso de compra ?

Rellenar los campos solicitados.

- Son muchos campos o los necesarios ?
- Le parece un paso obligatorio pero imprescindible?
- Sabe que los datos de su tarjeta no se almacenan y pasan al banco, luego se elimina ?

Encender el flash para capturar el código QR.

-
- Cree que es necesario ?
 - Lo ha probado con poca iluminación ?

Volver hacia atrás en cualquier menú.

- Es intuitivo volver hacia atrás ?

Salir de la aplicación.

- Encuentra fácil y rápido salir de la aplicación ?

5. Desarrollo e Implementación.

5.1. Definición de estructura de datos.

A nivel de estructura de datos se ha trabajado con 2 tipos de estructura de datos:

- SQLite en el móvil Android, para la persistencia en el dispositivo.

Y

- Mysql en el servidor web, para la consulta de productos y precios, así como para el envío de los pedidos y el login del usuario.

En el código adjunto, se puede apreciar la definición de la sentencia SQL de creación y su método de creación en el dispositivo móvil:

Código de declaración de sentencia SQL para creación de la BBDD de persistencia en el dispositivo móvil sobre SQLite:

```
public class BaseDatosHelper extends SQLiteOpenHelper {
```

```
    private static String DB_PATH =  
    "/data/data/com.tfc.poolstore/databases/";  
    private static String DB_NAME = "poolstore";  
    private SQLiteDatabase myDataBase;
```

```
        String crearTablaPedido = "create table if not exists Pedido  
(codigo text primay key, usuario text not null, precio text not null, unidad text not  
null);";  
        .....
```

Código de creación de bases de datos en SQLite :

```
public void onCreate(SQLiteDatabase db) {  
  
    try {  
  
        db.execSQL(crearTablaPedido);  
  
    }  
    catch (SQLException e){  
        e.printStackTrace();  
    }  
}
```

Código de creación de tablas en mysql:

Articulo:

```
CREATE TABLE `articulo` (  
  `codigo` varchar(10),  
  `nombre` varchar(50),  
  `precio` float NOT NULL,  
  `imagen` varchar(50) NOT NULL,  
  `qr` varchar(50) NOT NULL,  
  PRIMARY KEY (`codigo`),  
  UNIQUE KEY `nombre` (`nombre`,`imagen`)  
) ENGINE=MyISAM;
```

Cliente:

```
CREATE TABLE `cliente` (  
  `usuario` varchar(15) NOT NULL,  
  `password` varchar(15) NOT NULL,  
  PRIMARY KEY (`usuario`)
```

```
) ENGINE=MyISAM DEFAULT;
```

Pedido:

```
CREATE TABLE `pedido` (  
  `id_pedido` int(11) NOT NULL AUTO_INCREMENT,  
  `usuario` varchar(50) NOT NULL,  
  `codigo` varchar(50) NOT NULL,  
  `precio` float NOT NULL,  
  `cantidad` int(11) NOT NULL,  
  `fecha_pedido` datetime NOT NULL,  
  PRIMARY KEY (`id_pedido`)  
) ENGINE=MyISAM AUTO_INCREMENT=103 ;
```

5.2. Desarrollo de interfaces.

Tomando como base, lo desarrollado, en la fase de diseño, es decir los sketches y los prototipos, se han realizado las diferentes interfaces de usuario en los diferentes layouts utilizados, como son:

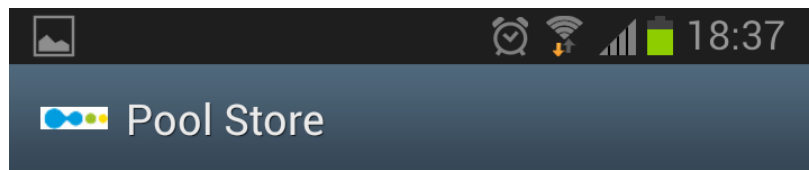
- Main.xml
- Captura.xml
- Carrito.xml
- Lista.xml
- Lista_item.xml
- Articulo_qr.xml
- Pago.xml

A continuación se muestran las interfaces finales en el dispositivo:

Icono de la aplicación en el móvil:



Login a la aplicación:



Nombre de usuario

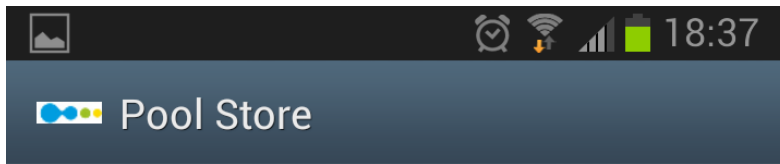
Contraseña

Iniciar sesión

Acceso solo para profesionales

Registrarse!

Botón de captura de códigos QR:



Captura el código QR

Pantalla de escaneo de códigos QR:



Pantalla de artículo escaneado:

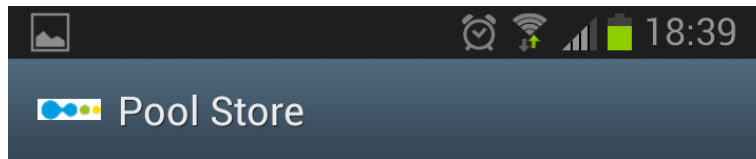
Código : 36596
Nombre : Filtro Atlas
Precio : 400 eur/unidad



Cantidad : 1

Añadir al carrito

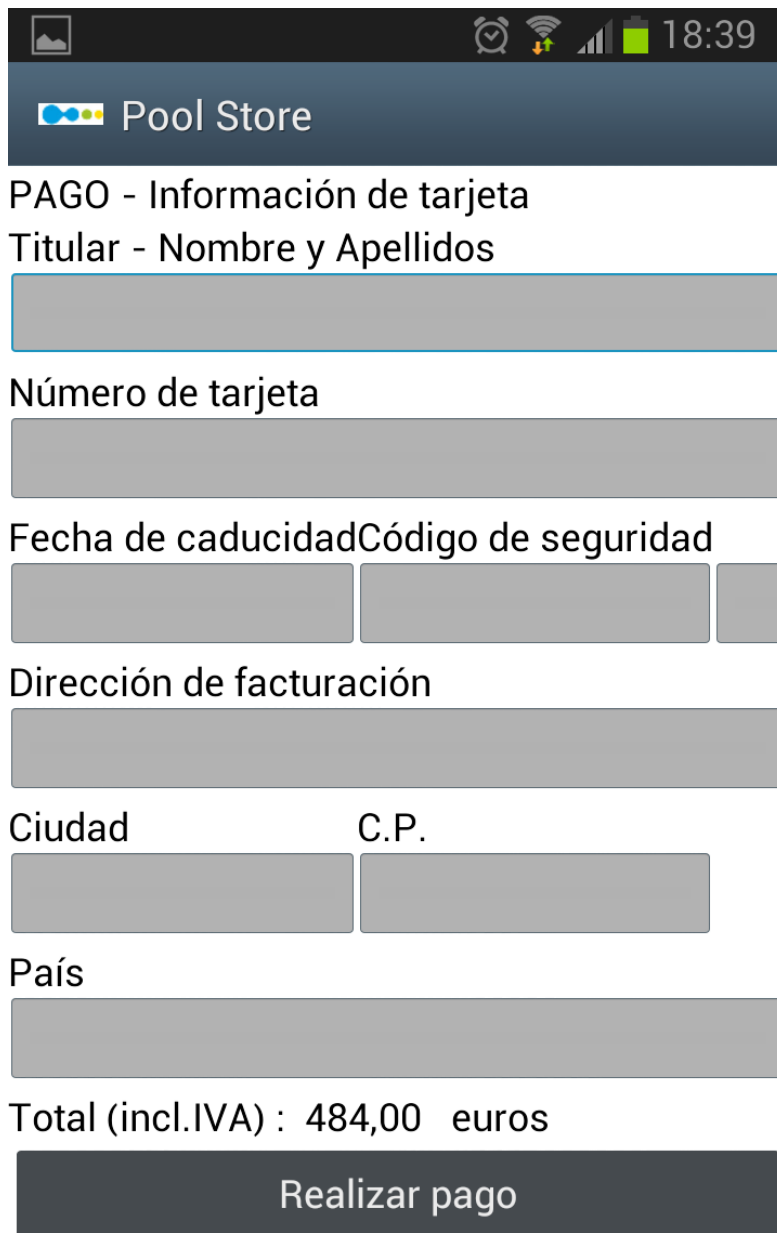
Pantalla de carrito:



Código:
36596
Cantidad pedida
1
Precio/unidad
400

--

Subtotal a pagar :400

Pantalla de pago:

Pool Store

PAGO - Información de tarjeta

Titular - Nombre y Apellidos

Número de tarjeta

Fecha de caducidad Código de seguridad

Dirección de facturación

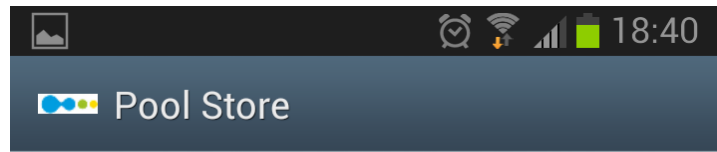
Ciudad C.P.

País

Total (incl.IVA) : 484,00 euros

Realizar pago

Pantalla de mensaje de pedido realizado con éxito:



Captura el código QR

Su pedido se ha realizado con éxito

5.3. Programación de módulos.

Este proyecto está basado en Android, el cual se ha desarrollado con Eclipse y utilizando el plugin de Android.

Se han desarrollado en 5 paquetes:

- **Com.tfc.poolstore:** Paquete principal de inicio de la aplicación y gestión de la autenticación.
- **Com.tfc.poolstore.captura:** Paquete donde se agrupa el proceso de captura de los artículos.
- **Com.tfc.poolstore.carrito:** Paquete donde se agrupa el proceso de gestión del carrito de compra, añadir productos, eliminar productos.
- **Com.tfc.poolstore.pago:** Paquete donde se agrupa el proceso de pago del pedido.
- **Com.tfc.poolstore.util:** Paquete donde está la clase de utilidades para las comunicaciones entre dispositivos móviles y el servidor de la aplicación.

En cada uno de ellos, se ha distribuido las ocho clases de las que consta esta aplicación, con el fin de organizarlos correctamente en función de las características.

Las clases realizadas han sido las siguientes, de las cuales mostramos los métodos más importantes:

- **Login.java** : Clase que controla la autenticación a la aplicación.

Método que comprueba el login:

// Se envía por JSON usuario y password a validar

```
public boolean loginstatus(String username ,String password ) {  
    int logstatus=-1;  
  
    ArrayList<NameValuePair> postparameters2send= new  
    ArrayList<NameValuePair>();  
  
    postparameters2send.add(new  
    BasicNameValuePair("usuario",username));
```

```

        postparameters2send.add(new
BasicNameValuePair("password",password));

        JSONArray jdata=post.getServerdata(postparameters2send,
URL_connect);

        SystemClock.sleep(950);

        if (jdata!=null && jdata.length() > 0){

            JSONObject json_data;
            try {
                json_data = jdata.getJSONObject(0);
                logstatus=json_data.getInt("logstatus");

            } catch (JSONException e) {

                e.printStackTrace();

            }

            if (logstatus==0){

                return false;
            }
            else{

                return true;
            }

        }else{

            return false;
        }

    }

```

- **ArticuloQR.java** : Clase para artículos capturados.

Mostramos 2 métodos, el que carga el artículo capturado por QR y el que envía el artículo al carrito:

```

public void onCreate(Bundle savedInstanceState) {

// Se carga el artículo recuperado con el código QR

    super.onCreate(savedInstanceState);
    setContentView(R.layout.articulo_qr);

```

```

txt_codigo = (TextView) findViewById(R.id.txt_codigo);
txt_nombre = (TextView) findViewById(R.id.txt_nombre);
txt_precio = (TextView) findViewById(R.id.txt_precio);
edt_unidad = (EditText) findViewById(R.id.edt_unidad);
txt_subtotal = (TextView) findViewById(R.id.txt_subtotal);

```

// Se envia al carrito el articulo

```

btn_carrito = (Button) findViewById(R.id.btn_carrito);
btn_carrito.setOnClickListener(new OnClickListener()
{
    public void onClick(View v)
    {
        Intent intent = new Intent(ArticuloQR.this, Carrito.class);
        intent.putExtra("codigo",txt_codigo.getText().toString());
        intent.putExtra("precio",txt_precio.getText().toString());
        intent.putExtra("unidad",edt_unidad.getText().toString());
        intent.putExtra("user", user);

        startActivity(intent);
    }
});

```

- **Captura.java** : Clase que se encarga de las llamadas a la aplicación de escaneo de códigos QR.

//Boton de llamada al servicio de escaneo de QR en una aplicacion externa Barcode Scanner

```

btn_captura = (Button) findViewById(R.id.btn_captura);
btn_captura.setOnClickListener(new OnClickListener()
{
    public void onClick(View v)
    {
        Intent intent = new
Intent("com.google.zxing.client.android.SCAN");

intent.putExtra("com.google.zxing.client.android.SCAN.SCAN_MODE",
"QR_CODE_MODE");
        startActivityForResult(intent, 0);
    }
});
}

```

//Se recibe el resultado de la lectura

```
public void onActivityResult(int requestCode, int resultCode, Intent
intent) {

    if (requestCode == 0) {
        if (resultCode == RESULT_OK) {
            String contents = intent.getStringExtra("SCAN_RESULT");

            URL_QR = contents;

            new asyncqr().execute(URL_QR);

            SystemClock.sleep(950);

        } else if (resultCode == RESULT_CANCELED) {

        }
    }
}
```

- **BaseDatosHelper.java:** Clase que gestiona la conexión con la base de datos.

```
public boolean actualizaPedido(String usuario, String codigo, String precio,
String unidad)
{

    String consulta = "";
    String consulta2 = "";
    String where = "";
    ContentValues cv = new ContentValues();
    cv.put(TABLE_KEY_USUARIO, usuario);
    cv.put(TABLE_KEY_CODIGO, codigo);
    cv.put(TABLE_KEY_PRECIO, precio);
    cv.put(TABLE_KEY_UNIDAD, unidad);
```

```
long rowid = -1;

try {

    consulta = "SELECT codigo FROM Pedido WHERE
codigo="+codigo+"";

    Cursor seleccion = myDataBase.rawQuery(consulta,null);

    if (seleccion.moveToFirst()) {

        where = "codigo=" + codigo;
        rowid = myDataBase.update(TABLE_PEDIDOS,cv, where, null);
    }

    else {

        rowid= myDataBase.insert(TABLE_PEDIDOS, null, cv);
    }

}

catch (SQLException e){
    e.printStackTrace();
}

return rowid >= 0;
}
```

- **Carrito.java** : Clase que gestiona el funcionamiento del carrito, añadir o eliminar artículos.

Método que carga el listado de productos:

```

public View getView(final int position, View convertView, ViewGroup parent) {

    View v = convertView;

    if (v == null) {

        LayoutInflater vi = (LayoutInflater)
getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);
        v = vi.inflate(R.layout.lista\_item, null);

    }

    final Pedido listItem = mList.get(position);

    if (listItem != null) {

        ((TextView)
v.findViewById(R.id.txt\_codigo)).setText(listItem.getCodigo());
        ((TextView)
v.findViewById(R.id.txt\_unidad)).setText(listItem.getUnidad());
        ((TextView)
v.findViewById(R.id.txt\_precio)).setText(listItem.getPrecio());
        ((Button)
v.findViewById(R.id.btn\_borrar)).setText("Eliminar");

        // Boton de borrado de linea de pedido

        ((Button)
v.findViewById(R.id.btn\_borrar)).setOnClickListener(new OnClickListener()
        {

            public void onClick(View v)
            {

                miBBDDHelper.abrirBaseDatos();
                miBBDDHelper.borrarFila(mList.get(position).getCodigo());
                miBBDDHelper.close();
                mList.remove(position);

                notifyDataSetChanged();
            }
        }
    }
}

```

```

    }
  });
}

return v;
}
}

```

- **Pedido.java** : Bean de objetos pedido, setters y getters.
- **Pago.java** : Clase que gestiona el pago y envía pedido al servidor después del pago.

//Simula el pago al banco y si es OK, guarda el pedido en la BBDD en el servidor remoto

```

public void onClick(View v) {
    HttpClient httpclient = new
DefaultHttpClient();
    HttpPost httppost = new
HttpPost("http://poolstore.comlu.com/pedido.php");

    try {

        for (int i=0; i < Pedidos.size();i++) {

            List<NameValuePair>
nameValuePairs = new ArrayList<NameValuePair>(4);
            nameValuePairs.add(new
BasicNameValuePair("usuario", Pedidos.get(i).getUsuario()));
            nameValuePairs.add(new
BasicNameValuePair("codigo", Pedidos.get(i).getCodigo());
            nameValuePairs.add(new
BasicNameValuePair("precio", Pedidos.get(i).getPrecio());
            nameValuePairs.add(new
BasicNameValuePair("unidad", Pedidos.get(i).getUnidad());

```

```

        httpPost.setEntity(new
        UriEncodedFormEntity(nameValuePairs));

        HttpResponse response =
        httpClient.execute(httpPost);

    }

    Toast.makeText(getApplicationContext(), "Su pedido se ha realizado con éxito",
    Toast.LENGTH_LONG).show();

        miBDDHelper.abrirBaseDatos();

        miBDDHelper.onDelete();

        miBDDHelper.close();

        SystemClock.sleep(900);

        usuario =
        Pedidos.get(0).getUsuario();

        Intent in=new Intent(Pago.this,
        Captura.class);
        usuario);

        in.putExtra("user",
        startActivity(in);

        finish();

```

- **HttpPostaux.java** : Clase que gestiona la conexión entre dispositivo y servidor por httpPost, transfiriendo los objetos JSON.

```

private void httpPostconnect(ArrayList<NameValuePair> parametros, String
urlwebserver){

    //
    try{
        HttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(urlwebserver);

        httpPost.setEntity(new UriEncodedFormEntity(parametros));

```

```
    HttpResponse response = httpClient.execute(httpPost);
    HttpEntity entity = response.getEntity();
    is = entity.getContent();

} catch (Exception e) {

}

}
```

5.4. Implementación y puesta en marcha.

En esta etapa del proyecto, se lleva a producción la aplicación desarrollada, en un entorno de producción, o bien por hosting o desde la misma empresa para la que se implementa la aplicación.

En el hosting interno o externo se ha de realizar lo siguiente:

- Crear la base de datos en mysql.
- Desplegar los html y php de la aplicación.
- Configurar los dns para el dominio a utilizar.
- Configurar Apache para esta nueva aplicación y dominio.

En relación a la aplicación móvil, se ha de desplegar bien a través de una web de descargas de aplicaciones android (.apk) o a través del Play Store de Google, que sería la opción más recomendada.

Asimismo se activará la pasarela de pago con el banco/paypal.

Y finalmente se enlazará con el entorno de la empresa, desde donde se gestionará el pedido para su entrega al domicilio del cliente.

Se realizan las pruebas en real y se forma al personal que gestionará la aplicación y todo el proceso de compra.

6. Conclusiones.

Este proyecto se ha llevado a cabo siguiendo la gestión de proyectos e ingeniería del software aprendida durante la carrera, permitiendo seguir paso a paso cada una de sus etapas.

Por otro lado, ha permitido utilizar diferentes conceptos tecnológicos actuales como son, el uso de bases de datos Mysql, el uso de bases de datos en dispositivos móviles SQLite, la programación a un nivel avanzado de Android, utilizando Eclipse y con objetos JSON.

Además ha permitido conocer la gestión de servidores Linux, utilizando Apache para el despliegue del entorno web, html y php, así como también el manejo de dispositivos móviles de última generación como es el Samsung Galaxy S3 y su integración con aplicaciones de escaneo de códigos QR y su gestión.

En el apartado de integración universidad-empresa, se ha conseguido uno de los objetivos que se buscan entre estas 2 instituciones que es la de compartir conocimiento y realizar la transmisión del mismo en ambas direcciones.

Y esto se ha conseguido ya que este proyecto se pondrá en marcha durante el presente año, en producción para ser utilizado por la empresa en la que trabajo actualmente.


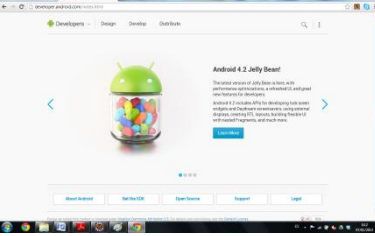


A nivel personal, ha sido una experiencia motivadora, por haber podido poner en práctica los conocimientos adquiridos, haber tenido la oportunidad de investigar nuevas tecnologías móviles, así como por haber conseguido fomentar y formar parte en el proceso de la relación entre universidad y empresa.

7. Bibliografía.

En este proyecto se han utilizado diversas fuentes, tanto a nivel de la universidad, con material de asignaturas diversas, así como páginas de internet y libros sobre las tecnologías utilizadas.

Entre las diversas fuentes consultadas, podemos resaltar las siguientes:

	<p>Curso Android. Desarrollo de aplicaciones móviles.</p> <p>Adrian Catalan</p> <p>Guías Maestros del Web 2011</p>
	<p>El Gran Libro de Android</p> <p>Jesús Tomás Girones</p> <p>Marcombo 2011</p>
	<p>Desarrollo de aplicaciones para Android</p> <p>Joan Ribas</p> <p>Anaya 2011</p>
	<p>Androideity</p> <p>Aurora Rodríguez</p> <p>www.androideity.com</p>

	<p>Code Project</p> <p>Community of Software development and Design developers</p> <p>www.codeproject.com</p>
	<p>Android Developers</p> <p>Android Developers Community</p> <p>http://developer.android.com</p>
	<p>El androide libre</p> <p>Comunidad Android</p> <p>www.elandroidelibre.com</p>
	<p>Android App Builder</p> <p>TechRepublic</p> <p>http://www.techrepublic.com/blog/app-builder</p>