

TRABAJO FIN DE GRADO

“PYMNet”

**Desarrollo de una aplicación Android para la incursión de las PYMEs
en el mundo de las aplicaciones móviles.**



Autor: Alejandro Ortiz García

Director: María Victoria Bueno Delgado

Codirector: Juan Antonio López Soro

Septiembre 2016



Autor	Alejandro Ortiz García
E-mail del Autor	alex.ortiz.garcia@gmail.com
Director(es)	Mª Victoria Bueno Delgado
E-mail del Director	Mvictoria.bueno@upct.es
Codirector(es)	Juan Antonio López Soro (jalopezsoro@gmail.com)
Título del PFC	<i>PYMNet – Desarrollo de una aplicación Android para la incursión de las PYMES en el mundo de las aplicaciones móviles.</i>
Descriptores	PYMNet, PYME, Empresa, Android, App
Resumen	
<p>Implementar una aplicación en Android para dispositivos móviles con el fin de que las PYMES tengan acceso al mundo de las aplicaciones móviles de forma barata, cómoda y sencilla teniendo así una aplicación propia totalmente gestionable.</p>	
Titulación	GIT, Grado en Ingeniería Telemática
Intensificación	
Departamento	Tecnologías de la Información y las Comunicaciones
Fecha de Presentación	Septiembre 2016

ÍNDICE

Índice de figuras	7
1 Introducción.....	9
1.1 Antecedentes.....	9
1.2 Motivación y objetivos	10
1.3 Estructura del contenido	10
2 Solución técnica.....	12
2.1 Android	12
2.2 Android studio	14
2.3 Git y Bitbucket.....	16
2.4 Backendless	18
2.5 Librerías externas y layouts personalizados	19
3 Aplicación.....	21
3.1 Creación del proyecto en Android studio	21
3.2 Configuración Backendless	24
3.3 Bases de datos	28
3.4 Línea de Diseño	30
3.4.1 Logo.....	30
3.4.2 Colores.....	30
3.4.3 Layouts	31
3.4.4 Detalles	31
3.5 Actividades y clases java	32
3.5.1 AndroidManifest.xml	32
3.5.2 ActivityChooser.....	32
3.5.3 Login.....	32
3.5.4 Register.....	34
3.5.5 MainActivity.....	34
3.5.6 Tab_User	35
3.5.7 ImageViewer	37
3.5.8 Tab_Favoritos.....	37
3.5.9 Tab_empresa.....	38
3.5.10 Nueva_Empresa_1	39
3.5.11 Nueva_Empresa_2.....	40

3.5.12	Open_empresa	40
3.5.13	OpenUser	41
3.5.14	Tab_búsqueda.....	42
3.5.15	Tab_Settings	44
3.5.16	ConfigUser	45
3.5.17	Objetos (clases) de la base de datos.....	45
4	Conclusiones y Lineas futuras	47
4.1	Líneas futuras.....	48
	Bibliografía.....	49

ÍNDICE DE FIGURAS

Figure 1: Número de empresas en España.....	9
Figure 2: Versiones de Android.....	12
Figure 3: Arquitectura de Android.	14
Figure 4: Android Studio, página de bienvenida.....	15
Figure 5: Android Studio, Interfaz gráfica.	15
Figure 6: Logo de JetBrains. /// Logo de Gradle.	16
Figure 7: Logo de Git.	17
Figure 8: Logo de Bitbucket.....	17
Figure 9: Logo de Backendless	18
Figure 10: Identificación y clave secreta de Backendless.	19
Figure 11: Ejemplo de layout personalizado.	20
Figure 12: Pantalla de bienvenida de Android Studio.....	21
Figure 13: Creación de proyecto. Android Studio.....	22
Figure 14: Creación de proyecto. Plataforma y min SDK.....	23
Figure 15: Creación del proyecto. Añadir Actividades. Android Studio.	24
Figure 16: Backendless. Cabecera de la Web.....	24
Figure 17: Backendless. Nombre de la aplicación	25
Figure 18: Backendless. Administrar.	26
Figure 19: Backendless. Pestaña de Usuarios.	27
Figure 20: Esquema de la tabla Users	28
Figure 21: Esquema de la tabla Empresa.....	28
Figure 22: Esquema de la tabla UserFriends	28
Figure 23: Esquema de la tabla UserEmpresas.	29
Figure 24: Esquema de la tabla EventosEmpresas	29
Figure 25: Esquema de la tabla EventosUsuarios	29
Figure 26: Esquema de la tabla Comentarios.	29
Figure 27: Esquema de la tabla Catalogo	29
Figure 28: Esquema de la tabla Trabajadores.....	30
Figure 29: Dialog informado del proceso de Login.	31
Figure 30: PYMNet. Login.....	33
Figure 31: PYMNet. Registro.....	34
Figure 32: PYMNet. Distribución en Pestañas.....	35
Figure 33: PYMNet. Pestaña de Usuario.	36
Figure 34: PYMNet. Visor de Imágenes.	37
Figure 35: PYMNet. Pestaña de Favoritos	38
Figure 36: PYMNet. Pestaña de Empresas.....	39
Figure 37: PYMNet. Formulario de registro de empresa.	40
Figure 38: PYMNet. Perfil de Empresa EmpYtec.	41
Figure 39: PYMNet. Perfil de usuario. (Toast notificando que se ha seguido correctamente).	42
Figure 40: PYMNet. Pestaña de Búsqueda de Usuarios.	43
Figure 41: PYMNet. Pestaña de Búsqueda de Empresas.....	44
Figure 42: PYMNet. Pestaña de Settings.	45

1 INTRODUCCIÓN

Este proyecto se ha llevado a cabo en el marco del programa de becas “Talentum Startups” dentro de la cátedra Telefónica-UPCT.

El objetivo del citado programa es la creación de nuevas empresas de base tecnológica con ideas de negocio innovadoras, una de las ideas seleccionadas ha sido PYMNet.

1.1 ANTECEDENTES

Las PYMEs (pequeñas y medianas empresas) son empresas con menos de 250 trabajadores cuyo volumen de negocio no supera los 50 millones de euros anuales.

Según una publicación del Ministerio de Industria, Energía y Turismo (podemos encontrar dicha publicación [aquí](#)) el 99.88% del tejido empresarial está constituido por PYMEs. 3.110.522 de las 3.114.361 empresas españolas son PYMEs. Las PYMEs generan el 66% del empleo empresarial. A nivel mundial se estima que el 30% del PIB es generado por éstas. Ya sea ofreciendo bienes y servicios o demandando productos constituyen la columna vertebral de cualquier economía.

	Micro Sin asalariados *	Micro 1-9	Pequeñas 10-49	Medianas 50-249	PYME 0-249	Grandes 250 y más	Total
ESPAÑA	1.670.329	1.314.398	107.784	18.011	3.110.522	3.839	3.114.361
%	53,6	42,2	3,5	0,6	99,9	0,1	100
UE-28 %	92,4		6,4	1,0	99,8	0,2	100

Figure 1: Número de empresas en España.

La importancia de las PYMEs en la economía española y mundial es indudable. Es el motor de cualquier economía y un factor de crecimiento para países en desarrollo.

Por otro lado, el diario www.elperiodico.com afirma que el 71% de los españoles usan el Smartphone para acceder a Internet, las ventas de dispositivos móviles crecen año tras año.

1.2 MOTIVACIÓN Y OBJETIVOS

Sumando la importancia de las PYMEs en la economía española, lo poco conectadas a Internet que se encuentran dichas empresas y el crecimiento de los Smartphone surge la idea de PYMNet.

PYMNet es una aplicación Android cuya finalidad es la de permitir a las PYMEs tener una aplicación propia a un precio muy económico ofreciendo así la oportunidad de llegar a clientes potenciales que no puede llegar en su forma actual. Trata de conectar a estos clientes potenciales con dichas empresas.

Además de la incursión de las PYMEs en el mundo de las aplicaciones móviles, PYMNet fomenta el estado centralizado de las empresas por lo que los usuarios que busquen empresas solo tendrán que instalar una única aplicación para interactuar con todas las empresas.

Su funcionamiento es el siguiente: una empresa se registra en PYMNet creando así su propio perfil en la aplicación, añadiendo los servicios (como un servicio de catálogo de productos, un servicio de reservas y citas previas o un servicio de comentarios) y configurando a su antojo el perfil y dichos servicios. Los usuarios pueden seguir a dicha empresa y hacer uso de sus servicios o simplemente ponerse en contacto con la empresa. De esta forma se rompen las barreras geográficas que tienen las PYMEs en estos momentos.

Los objetivos de este proyecto es la realización de una aplicación Android que cumpla las especificaciones mínimas de lo que es PYMNet. Un portal centralizado para clientes y empresas que pretende fomentar la incursión de las PYMEs en el mundo de las aplicaciones móviles y crear un puente entre estas empresas y sus potenciales clientes por medio de los Smartphone.

1.3 ESTRUCTURA DEL CONTENIDO

La estructura de este proyecto está contenida en distintos capítulos:

1. Introducción: es el capítulo actual, ofrece el contexto por el cual surge PYMNet así como los objetivos y motivaciones que lo llevan a cabo.
2. Solución Técnica: en este capítulo se introducen las tecnologías utilizadas para el desarrollo de PYMNet.

3. **Aplicación:** en este capítulo se detallan las distintas decisiones de diseño y se explican de forma más avanzada las distintas partes de la aplicación, incluyendo el diseño visual y la funcionalidad.
4. **Líneas futuras:** en este capítulo nos centramos en el futuro de PYMNet como aplicación pero también como Startup.
5. **Conclusiones:** en este capítulo abarcamos las distintas conclusiones alcanzadas en la realización del proyecto.
6. **Bibliografía:** aquí encontramos todas las fuentes de información utilizadas para la redacción de esta memoria.

2 SOLUCIÓN TÉCNICA

2.1 ANDROID

Android es un sistema operativo basado en el núcleo Linux diseñado principalmente para dispositivos móviles con pantalla táctil (teléfonos inteligentes, tabletas u otros dispositivos como relojes, televisores o automóviles inteligentes).

Inicialmente fue desarrollado por Android Inc., Google compró dicha empresa en 2005 y en 2007 Android fue presentado junto la fundación del Open Handset Alliance (un consorcio de 84 compañías que se dedican a desarrollar estándares abiertos para dispositivos móviles).

Desde su lanzamiento oficial en 2007, Android ha visto numerosas actualizaciones que sirven para arreglar bugs o agregar nuevas funcionalidades. Estas actualizaciones vienen con un nombre en código: Android 1.6 “Donut”, Android 2.1 “Eclair”, Android 2.2 “Froyo”, Android 2.3 “Gingerbread”, Android 3.0 “Honeycomb”, Android 4.0 “Ice Cream Sandwich”, Android 4.1 “Jelly Bean”, Android 4.4 “KitKat”, Android 5.0 “Lollipop”, Android 6.0 “Marshmallow”.

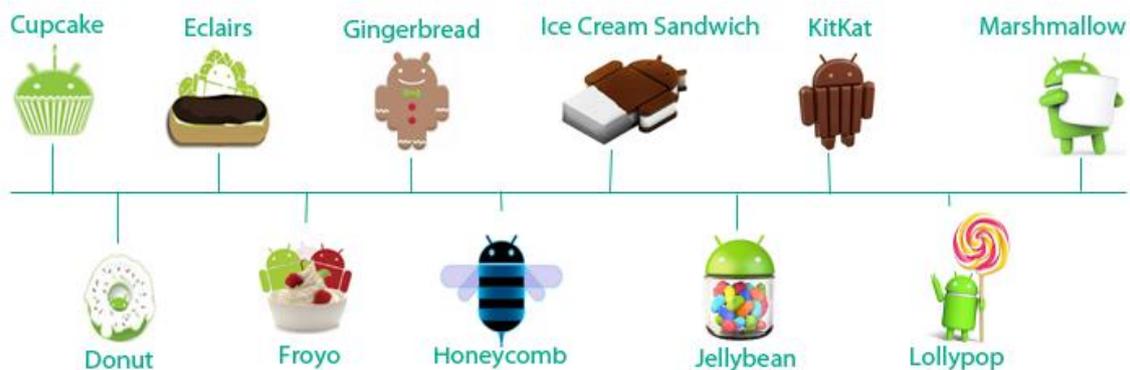


Figure 2: Versiones de Android.

Android es un sistema libre, es cierto que no todas las aplicaciones son libres pero cualquiera puede conseguir el código base de Android, hacer un “fork” y hacer su propia versión del sistema operativo.

Para hablar de la arquitectura de Android podemos distinguir entre cinco partes:

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecutaba hasta la versión 5.0 archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx". Desde la versión 5.0 utiliza el ART, que compila totalmente al momento de instalación de la aplicación.
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

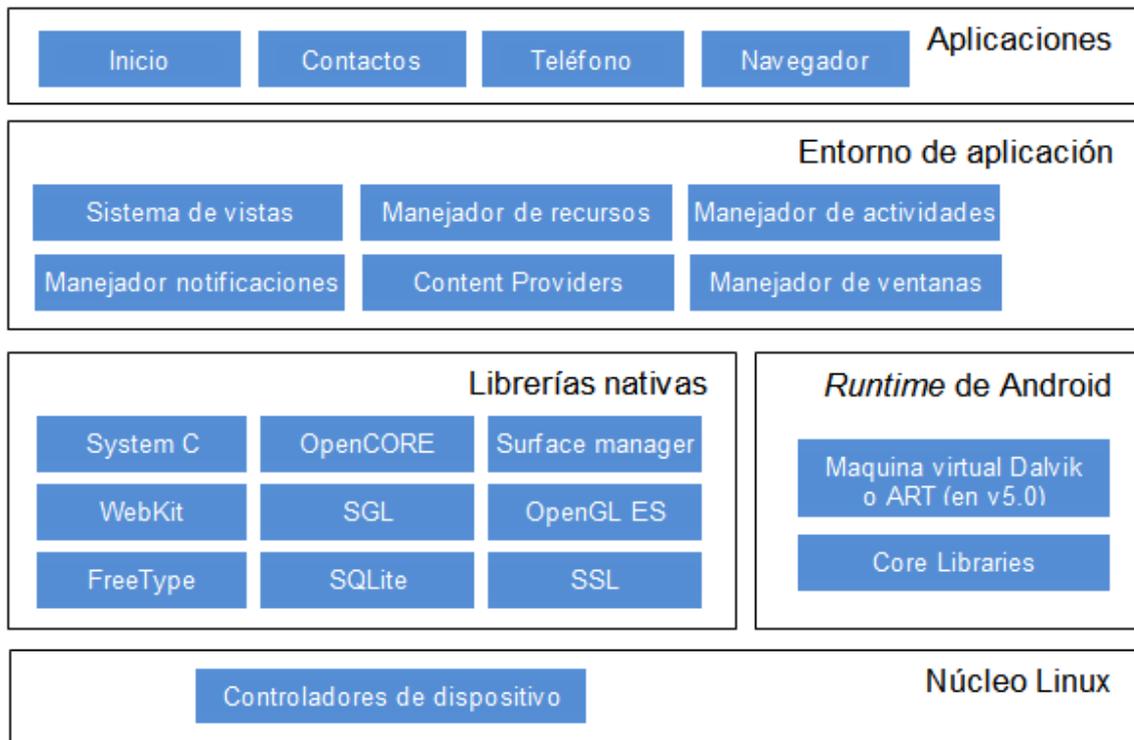


Figure 3: Arquitectura de Android.

Hoy en día, en España, la cuota de mercado de Android frente a sus competidores (iOS o Windows Phone) supera el 89% lo que lo convierte en el sistema operativo preferido para los españoles. Android sigue siendo el sistema operativo predominante en dispositivos móviles y por esta razón que se ha escogido esta plataforma para desarrollar la aplicación.

2.2 ANDROID STUDIO

Existen varios IDE para desarrollar aplicaciones Android, en nuestro caso hemos optado por la opción oficial que recomiendan los creadores de Android, Android Studio. Fue anunciado el 16 de mayo de 2013 reemplazando así a Eclipse como IDE oficial. Está basado en el software IntelliJ IDEA de JetBrains y es publicado de forma gratuita a través de la licencia Apache 2.0.

Nos ofrece renderización en tiempo real, consola de desarrollador con diversas pestañas con información relevante y de gran ayuda, soporte para construcción basada en Gradle, refactorización específica de Android y arreglos rápidos, herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas, plantillas para crear diseños comunes, soporte para programar aplicaciones para Android

2. Solución Técnica

Wear y mucho más. Una buena característica es el emulador, que te permite testear tu app en distintos dispositivos para ver su comportamiento.

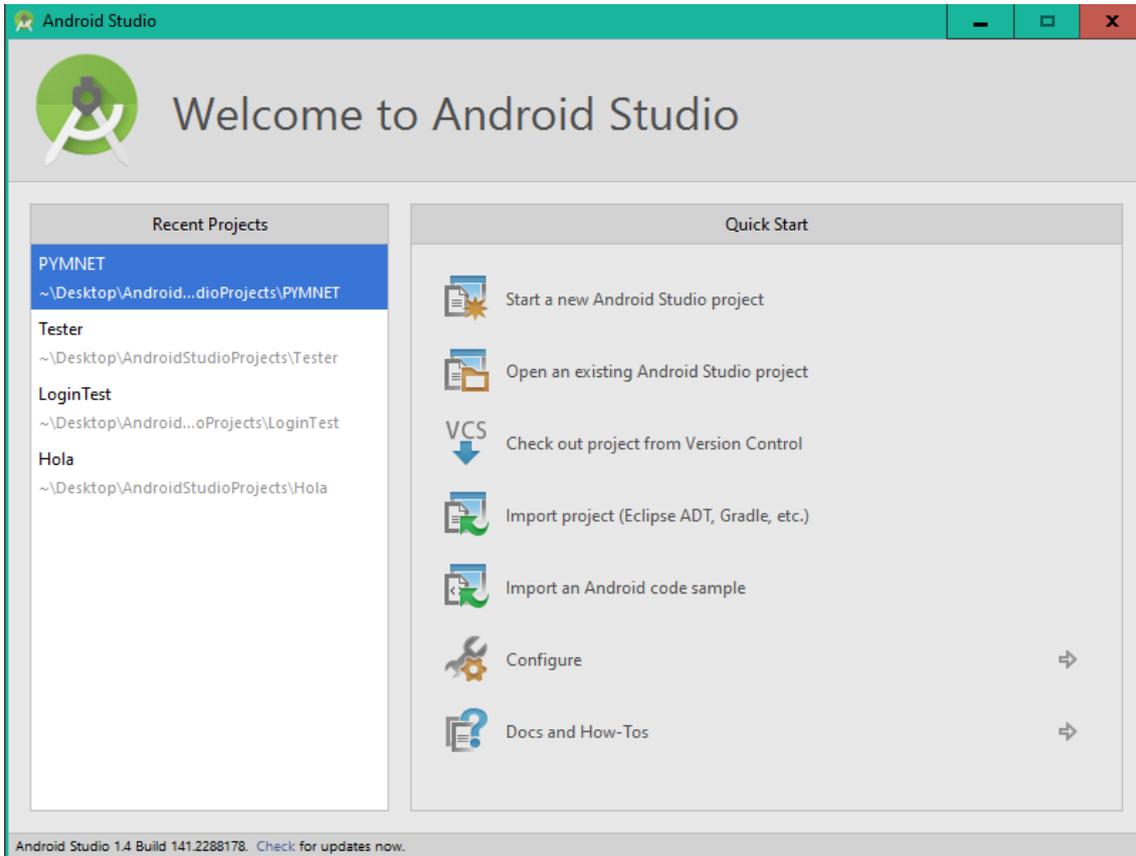


Figure 4: Android Studio, página de bienvenida.

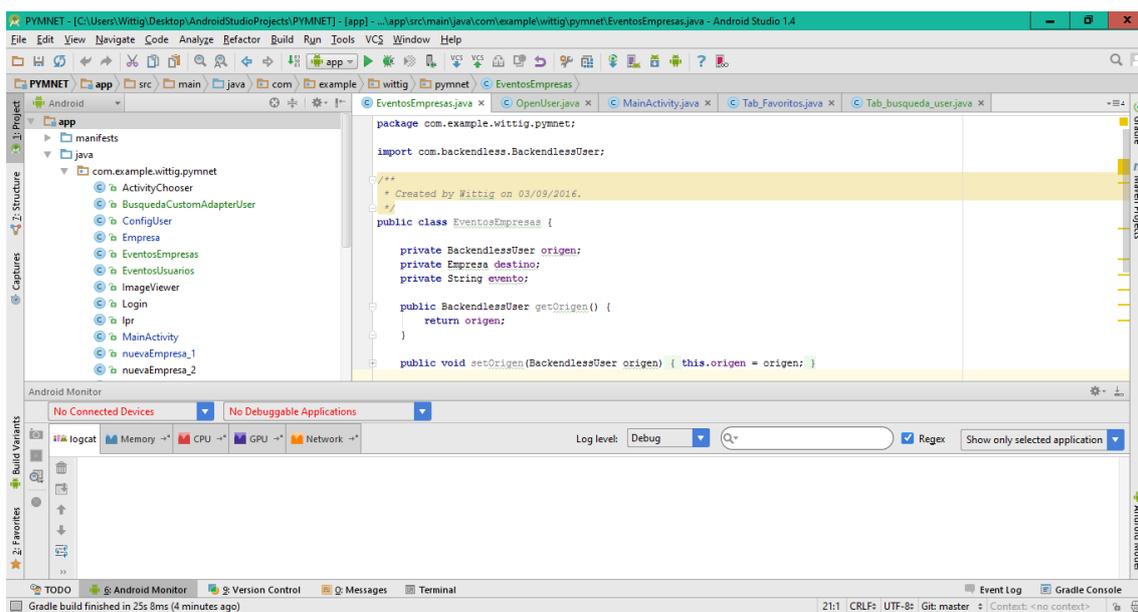


Figure 5: Android Studio, Interfaz gráfica.

JetBrains, anteriormente conocido como IntelliJ es una compañía de desarrollo de software centrados en la creación de herramientas de desarrollo, esta compañía ofrece una extensa familia de IDE para programar en lenguajes como Java, Ruby, Python, PHP, SQL, Objective-C, C++ o JavaScript.

Gradle es una herramienta de automatización de la construcción de nuestro código que bebe de las aportaciones que han realizado herramientas como ant y maven, pero intenta llevarlo todo un paso más allá soportando varios lenguajes de programación distintos, no solo Java.



Figure 6: Logo de JetBrains. /// Logo de Gradle.

Una de las facilidades que ofrece Android Studio es su fácil integración con sistemas de versiones como Git, CVS, Mercurial o Subversion. En nuestro caso se ha utilizado el control de versiones Git, combinado con un repositorio online alojado en www.bitbucket.org.

2.3 GIT Y BITBUCKET

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados sistemas de control de versiones o VCS (del inglés Version Control System). Estos sistemas facilitan la administración de las distintas versiones de cada producto

desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico). Ejemplos de este tipo de herramientas son entre otros: CVS, Subversion, SourceSafe, ClearCase, Darcs, Bazaar, Plastic SCM, Git, Mercurial, Perforce, Fossil SCM, Team Foundation Server.

El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente dando lugar a los sistemas de control de código fuente o SCM (siglas del inglés Source Code Management). Sin embargo, los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etc.

Se ha optado por utilizar el sistema de control de versiones Git ya que es el más extendido en la actualidad y ya me encontraba familiarizado con su funcionamiento. Git es un VCS diseñado por Linus Torvalds (creador de Linux) pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos. Hay proyectos de mucha relevancia que utilizan Git, como por ejemplo, el núcleo de Linux.



Figure 7: Logo de Git.

Con la integración del proyecto en el sistema de control de versiones Git, hemos conseguido una mejor organización, un sistema de versiones escalable y recuperable y una flexibilidad en el desarrollo aun mayor. Pero todos estos cambios se almacenan en local, en nuestra carpeta. Para una mayor seguridad, se ha optado por utilizar un repositorio online alojado en <http://www.bitbucket.org/>.



Figure 8: Logo de Bitbucket.

A simples rasgos, un repositorio online, es una copia de nuestra carpeta local en la nube, desde la consola de Git o desde el mismo Android Studio podemos subir nuestra carpeta con los cambios pertinentes a la nube. De esta forma tenemos la seguridad de que aunque

se estropeará el ordenador donde está el proyecto, podríamos recuperar el proyecto sin apenas pérdidas.

2.4 BACKENDLESS

Por la naturaleza de la aplicación es necesario diferenciar entre front-end y back-end. El front-end es la parte visual de la aplicación, es decir, la propia aplicación Android que los usuarios instalarán en sus dispositivos. Pero es necesario una parte de servidor que gestione todas las peticiones de los usuarios, esta parte es la denominada back-end.

En nuestro caso, el back-end debería controlar 3 pilares fundamentales en la aplicación:

- Gestión de usuarios: controlar el registro único de usuarios, el login, sus datos, etc.
- Almacenar información: almacenar información acerca de usuarios, empresas, y otros datos de interés para el beneficio de la aplicación y una mejor experiencia para el usuario.
- Almacenar archivos: como imágenes de perfil, etc.

La implementación de un servicio de back-end independiente y exclusivo para PYMNet sobrepasa la complejidad del proyecto por lo que se optó por investigar y usar la tecnología y usar un MBaaS (Mobile Backend as a Service).

Un MBaaS proporciona a los desarrolladores de aplicaciones una forma de vincular sus servicios con almacenamiento en la nube, además incorporan funcionalidades de gestión de usuarios, notificaciones, servicios de redes sociales, servicios de análisis de datos y servicios de geolocalización.

El servicio de MBaaS que se ha escogido para este proyecto es Backendless, un servicio que satisface todas las necesidades del proyecto, que es gratuito y escalable. Además de una extensa documentación para las distintas plataformas para las que está disponible, entre ellas, Android. [Aquí](#) se puede ver la documentación relacionada con Android.



Figure 9: Logo de Backendless

Para hacer uso de Backendless en nuestra aplicación tenemos que indicar a esta que se conecte con los servicios de Backendless, para ello, hacemos uso del siguiente código:

```
Backendless.initApp( Context, application-id, secret-key, version );
```

Los parámetros de application-id y secret-key son valores que obtenemos de nuestro panel de control (en la web de Backendless) y sirven para conectarnos a nuestra aplicación, son valores secretos para que terceros no puedan usar los servicios de nuestro backend.



Figure 10: Identificación y clave secreta de Backendless.

2.5 LIBRERÍAS EXTERNAS Y LAYOUTS PERSONALIZADOS

Además de las librerías base de Android y los diseños gráficos proporcionados, se han utilizado otras librerías de funciones que mejoraban la experiencia del usuario, la eficiencia optimizando los recursos del dispositivo, tanto en memoria como en uso de datos.

Algunas de estas librerías, las más interesantes de nombrar son las siguientes:

- Backendless: la API de Backendless nos permite comunicar nuestra aplicación Android con el servicio de backend en Backendless. Ofreciendo métodos para la gestión de usuarios, la gestión de las distintas bases de datos, etc. Podemos encontrar la API [aquí](#).
- Picasso: librería que permite descargar imágenes y adecuarlas en tamaño al recipiente donde serán cargadas, guarda automáticamente las imágenes en caché para no tener que descargarlas varias veces, es una librería muy simple, pero que facilita mucho el trabajo y permite optimizar el control de memoria y el uso de datos del dispositivo. Se puede ver más información [aquí](#). El código es abierto y está disponible en [github](#).
- Android-Shape-ImageView: es una librería que permite diseños personalizados de distintos elementos de Android como ImageView, ImageButton, etc. Gracias a esta librería se han utilizado ImageView con bordes redondeados o circulares

para ofrecer un diseño más estilizado a nuestra aplicación. Es una librería libre y su código está disponible en [github](#).

Además, para tener un diseño propio de la aplicación se han creado distintos diseños para distintos elementos de la aplicación, entre ellos, el más predominante es el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<shape
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:shape="rectangle" >
  <solid
    android:color="@color/fondo" >
  </solid>
  <stroke
    android:width="2dp"
    android:color="#000000" >
  </stroke>
  <corners
    android:radius="11dp" >
  </corners>
</shape>
```

Con esta porción de código se consigue un diseño con bordes redondeados de color negro, con un grosor de 2dp (densidad de píxeles, unidad utilizada en Android para ajustar los distintos elementos de la interfaz gráfica a los distintos tamaños de pantalla disponibles).

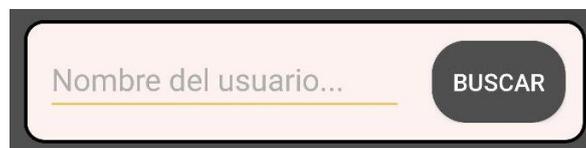


Figure 11: Ejemplo de layout personalizado.

3 APLICACIÓN

El primer paso antes de empezar con el desarrollo propiamente dicho tiene que ver con la preparación del proyecto en Android studio y las distintas tablas de la base de datos. Pese a que esta última tarea ha sido progresiva conforme la aplicación iba cogiendo forma.

3.1 CREACIÓN DEL PROYECTO EN ANDROID STUDIO

En primer lugar se llevó a cabo la creación del proyecto en Android Studio.

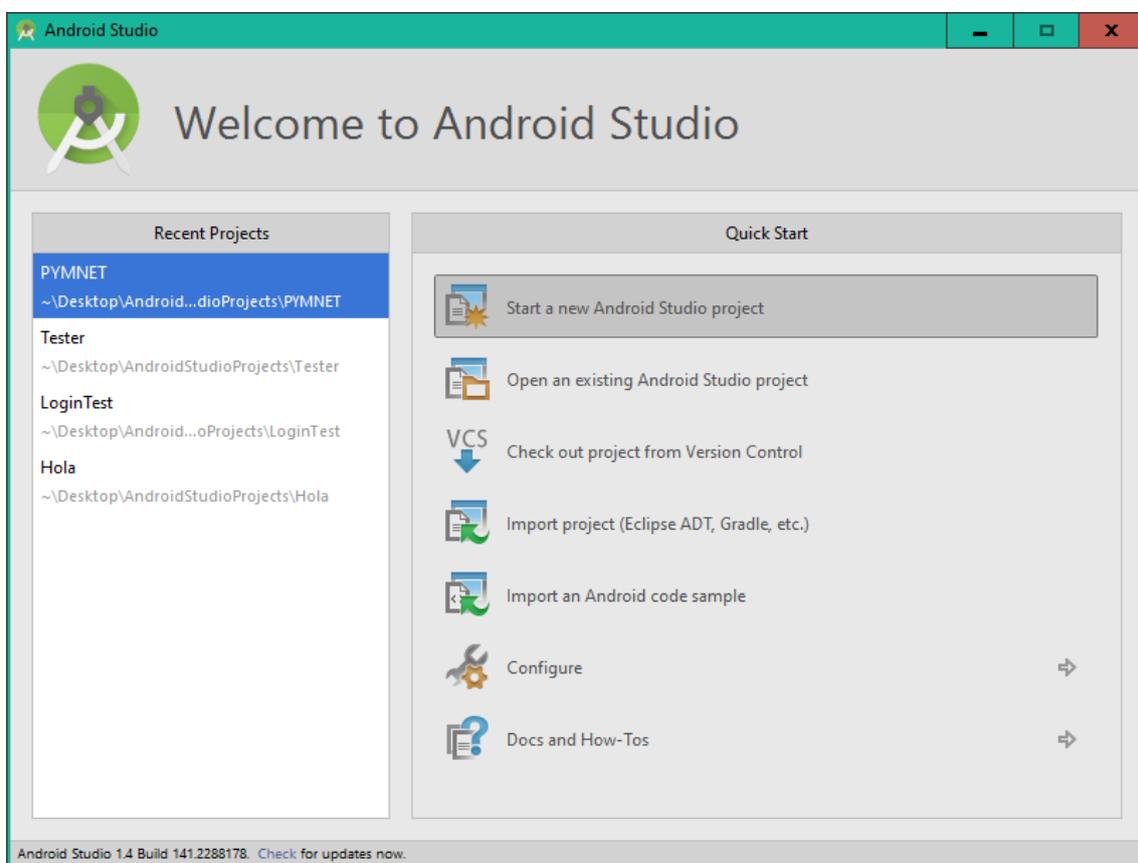


Figure 12: Pantalla de bienvenida de Android Studio

Para ello pinchamos en el botón “Start a new Android Studio Project” de la página de bienvenida de Android Studio. El siguiente paso es seleccionar el nombre de nuestro proyecto, el dominio de la compañía y el directorio donde guardaremos dicho proyecto:

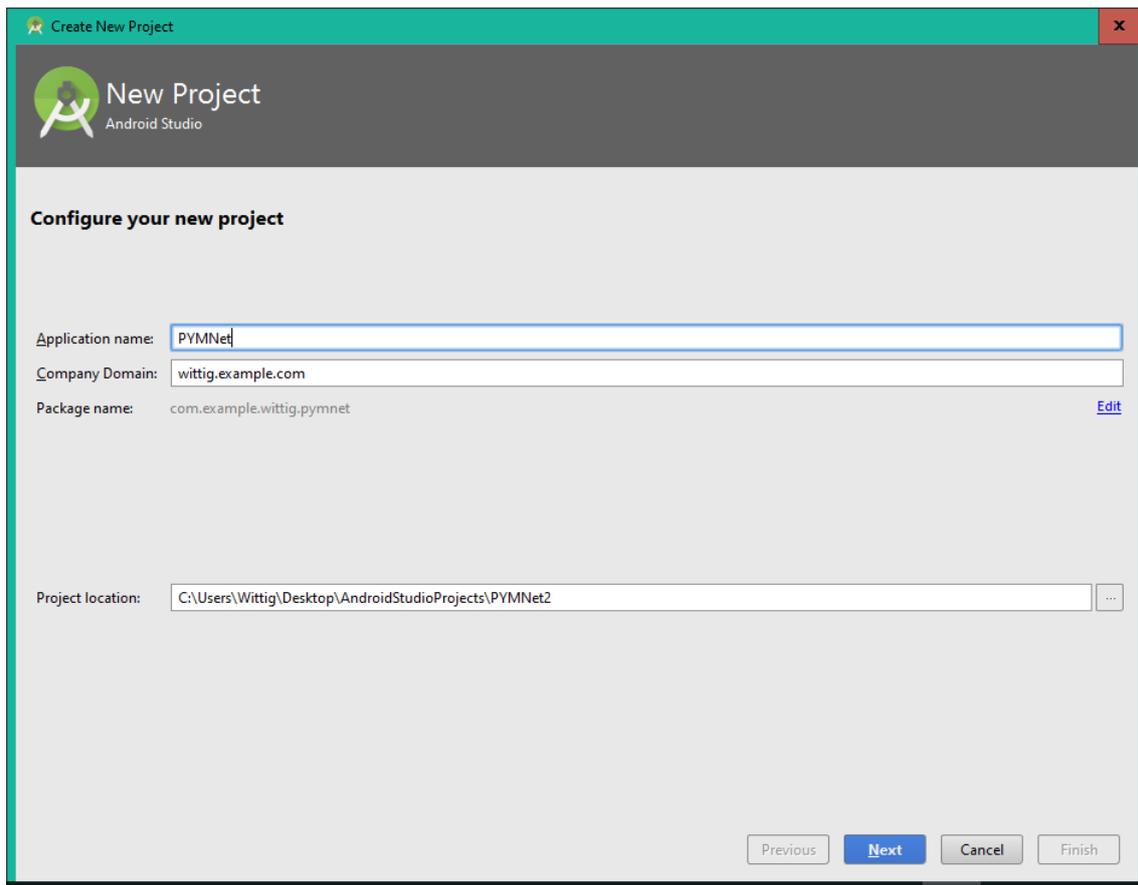


Figure 13: Creación de proyecto. Android Studio

En este punto hay que decidir a qué dispositivos Android vamos a orientar la aplicación, en este caso, a Smartphone o Tablet, seleccionamos el SDK mínimo para abarcar la mayor cantidad de dispositivos, el propio asistente de creación de proyectos de Android Studio nos indica el porcentaje de dispositivos que podrán utilizar nuestra aplicación si seleccionamos dicho SDK. Para este proyecto se ha optado por utilizar la API 15 (Android 4.0.3 IceCreamSandwich) ya que abarca el 94% de los dispositivos Android.

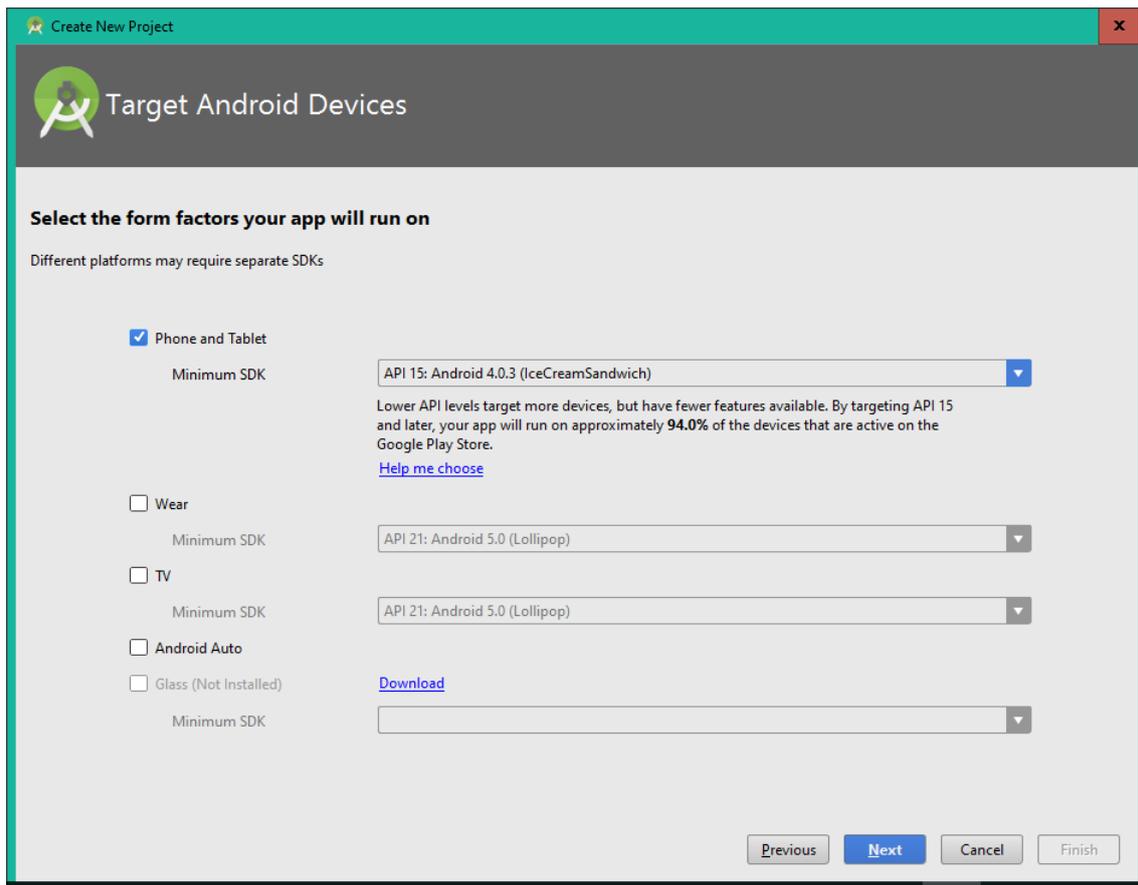


Figure 14: Creación de proyecto. Plataforma y min SDK.

A continuación, Android Studio nos permite añadir Actividades por defecto a nuestro proyecto, se ha decidido crear todas las actividades y sus layouts desde cero para tener un mayor control y conocimiento del funcionamiento de la aplicación y para tener un diseño propio y exclusivo. Por esta razón se ha seleccionado la opción “Add no Activity”.

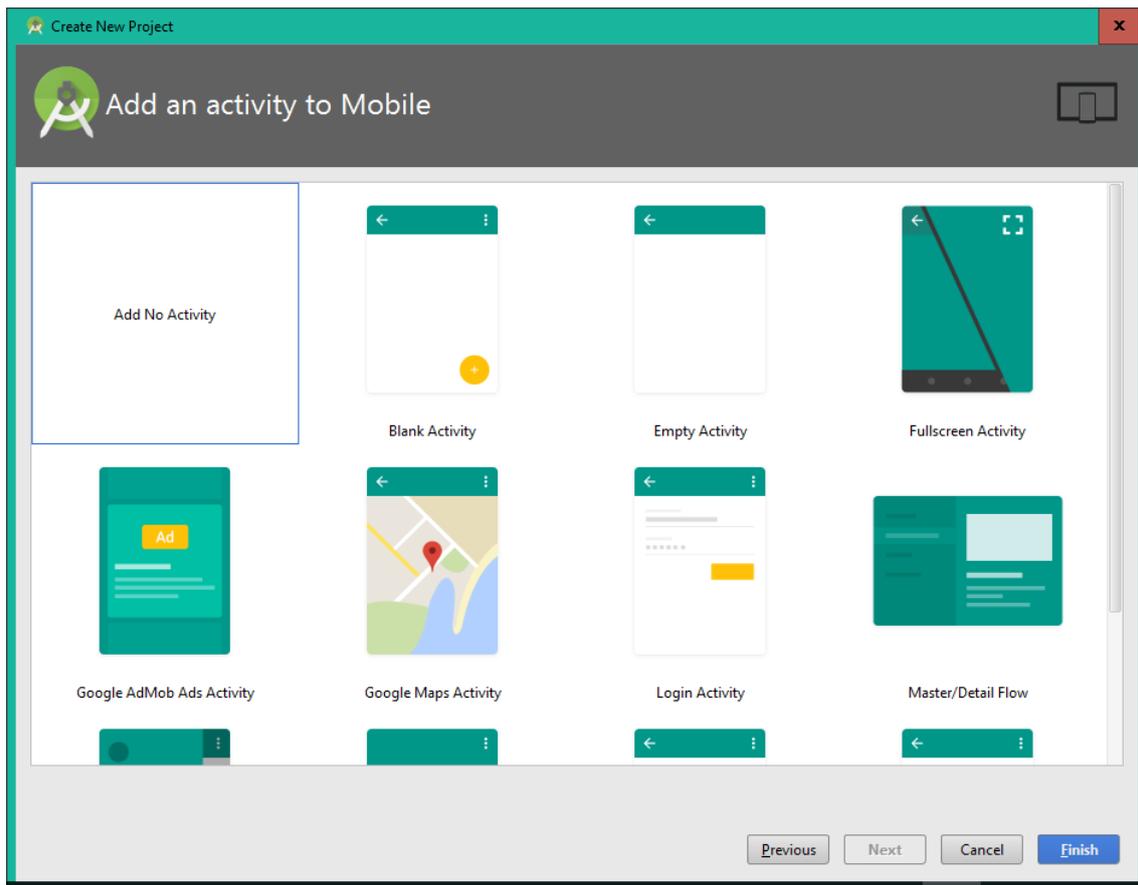


Figure 15: Creación del proyecto. Añadir Actividades. Android Studio.

Llegados a este punto, se fueron creando de forma progresiva las distintas Actividades que formar el proyecto, estas Actividades serán explicadas con más detalle en el apartado 3.3 de este mismo documento.

3.2 CONFIGURACIÓN BACKENDLESS

Una vez creado el proyecto se pasó a crear la aplicación en Backendless comenzando por las opciones más básicas y finalmente diseñando las distintas bases de datos y configurando las opciones más importantes para el buen uso del servicio de back-end.

Backendless es un servicio Freemium por lo que necesitas registrarte para utilizarlo, una vez estamos registrados podemos crear la aplicación haciendo click en el botón “Crear aplicación”.

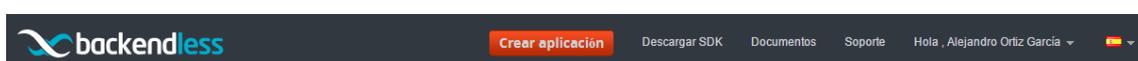


Figure 16: Backendless. Cabecera de la Web.

Al pulsar en el botón “Crear aplicación” tenemos que indicar el nombre de la aplicación en cuestión:



Figure 17: Backendless. Nombre de la aplicación

Nuestra aplicación es creada y entramos en el panel de control y configuración de dicha aplicación, dividido en varias pestañas:

En la pestaña de “Administrar” podemos ver los códigos secretos de la aplicación para cada plataforma (Android, iOS, etc) así como el identificador de la aplicación, podemos configurar el login con redes sociales como Facebook o Google, configurar las notificaciones push o configurar el servidor de correo electrónico entre otras. Podemos ver las distintas versiones de nuestra aplicación, analizar las llamadas a la API, los usuarios registrados, etc. Exportar o importar nuestra aplicación incluso configurar el sistema de facturación.



Figure 18: Backendless. Administrar.

En la pestaña “Usuarios” podemos configurar las propiedades de los usuarios registrados en la aplicación y activar la definición dinámica de propiedades de usuario. Esta opción nos permite crear distintas propiedades de forma dinámica, sin tener que configurarlas en la tabla, al encontrar una propiedad nueva se creará la columna automáticamente. Podemos configurar el registro de nuevos usuarios, configurando el registro en 2 pasos mediante un mensaje al correo electrónico, etc. Podemos configurar los distintos roles de los usuarios así como configurar los correos automáticos que mandará Backendless a los usuarios registrados para realizar acciones como recuperar la contraseña. Esta última es

una herramienta muy agradecida para cualquier aplicación que requiera de un login de usuarios.

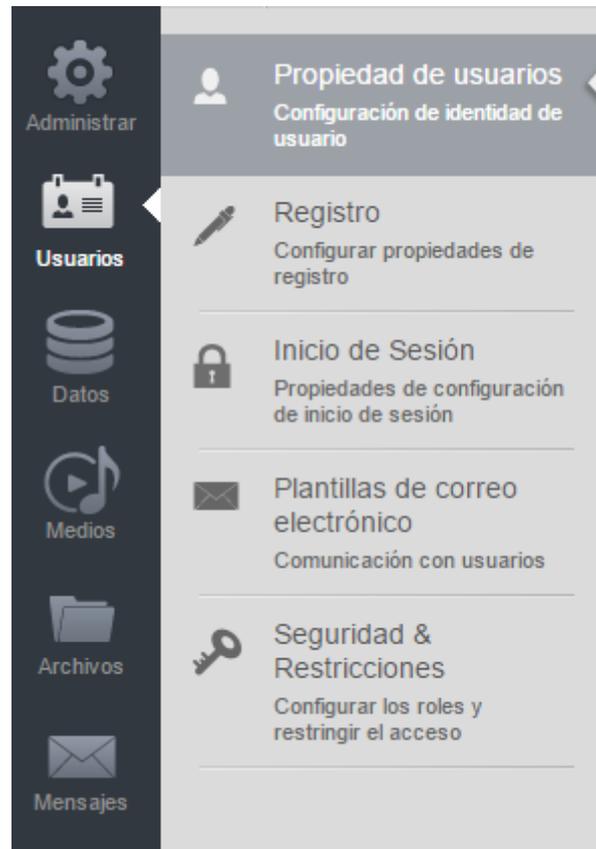


Figure 19: Backendless. Pestaña de Usuarios.

En la pestaña “Datos” podemos ver las distintas bases de datos creadas, crear nuevas tablas o configurar las tablas actuales, nos permite hacer búsquedas SQL y ordenar los datos a nuestro criterio.

En la pestaña “Archivos” encontramos el sistema de carpetas al que tenemos acceso, para subir imágenes o cualquier documento. La API de Backendless nos devuelve la URL de un archivo que subimos para poder guardarlo en la base de datos.

Las pestañas de “Medios”, “Mensajes”, “Geolocalización”, “Lógica de Negocio”, “Marketplace”, “Generar códigos” no han sido utilizadas en el desarrollo de este proyecto.

3.3 BASES DE DATOS

Las tablas creadas para este proyecto son las siguientes:

- Users: Esta tabla contiene la información de los usuarios registrados en nuestra aplicación. Esta tabla viene con la aplicación Backendless y por lo tanto tiene algunas propiedades predefinidas como userStatus, objectId, created, etc. Además de estas propiedades, se han diseñado otras como nombre, email, información de redes sociales, url de la imagen de perfil, etc. Estas propiedades forman la información de usuario.

Users											
email	name	aboutme	cita	facebook	googleplus	instagram	linkedin	pic	twitter	youtube	password

Figure 20: Esquema de la tabla Users

- Empresa: Esta tabla contiene la información de las empresas registradas en nuestra aplicación, algunas de las propiedades son nombre de la empresa, descripción, email, sector, teléfono o localización.

Empresa							
descripcion	email	name	pic	sector	telefono	web	localización

Figure 21: Esquema de la tabla Empresa.

- UserFriends: Esta tabla contiene una relación de usuario-usuarios, es decir, guarda la lista de usuarios que un usuario sigue. Gracias a los tipos de relación de Backendless, podemos establecer una relación entre estos campos y la tabla "Users". Cuando realizamos la petición SQL para que nos devuelva una entrada de esta tabla, nos devuelve la lista de usuarios amigos por lo que nos facilita la información para mostrar en nuestra aplicación.

UserFriends	
usuario	amigos

Figure 22: Esquema de la tabla UserFriends

- UserEmpresas: Esta tabla contiene la misma relación que la anterior pero entre usuario y empresas que sigue. El funcionamiento es similar.

UserEmpresas	
usuario	empresas

Figure 23: Esquema de la tabla UserEmpresas.

- EventosEmpresas: contiene la información de los eventos que requieren la interacción con las empresas.

EventosEmpresas		
origen	destino	evento

Figure 24: Esquema de la tabla EventosEmpresas

- EventosUsuarios: contiene la información de los eventos que requieren la interacción con los usuarios.

EventosUsuarios		
origen	destino	evento

Figure 25: Esquema de la tabla EventosUsuarios

- Comentarios: esta tabla contiene los comentarios que los usuarios dejan a las empresas utilizando el servicio de comentarios que explicaremos más adelante.

Comentarios			
usuario	empresa	comentario	fecha

Figure 26: Esquema de la tabla Comentarios.

- Catálogo: esta tabla contiene la información de los distintos productos que una empresa añade en el servicio de Catalogo.

Catalogo			
empresa	nombreProducto	descripcion	precio

Figure 27: Esquema de la tabla Catalogo

- Trabajadores: establece una relación entre empresa y los usuarios registrados que trabajan en ella. Es otro de los servicios añadidos.

Trabajadores	
empresa	trabajadores

Figure 28: Esquema de la tabla Trabajadores

3.4 LÍNEA DE DISEÑO

3.4.1 Logo

El logo de PYMNet es de autoría propia y consiste en una forma redondeada (como curiosidad, no llega a ser un círculo perfecto) con las letras P de pyme y T de Internet entrecruzados, dado que el objetivo de esta aplicación es, entre otras, el acercamiento entre las PYMEs e Internet, el nombre y el logo vinieron de la mano.

El logo es verde y blanco, y la explicación de esta decisión nace en los significados que transmiten dichos colores, el verde transmite esperanza, crecimiento, etc. El color blanco tranquilidad, seriedad.

3.4.2 Colores

Pese a que en un primer momento la línea de diseño tenía colores oscuros en los fondos y letras claras se descartó ya que cansaba la vista e invita a no pasar mucho tiempo navegando por la aplicación, por esta razón se realizó un diseño basado en colores claros con letras oscuras.

Los iconos de las distintas pestañas contrariaban bien el gris oscuro que parece enmarcar el resto de elementos, se ven claros y estéticamente son del gusto del autor. El color de fondo de los distintos elementos es un blanco rosado, no es un blanco total que resultaría dañino a la vista de cualquier usuario que no esté acostumbrado a tanto brillo. Es un color algo menos brillante, un toque muy sutil y característico de PYMNet.

De la misma forma, el color de las letras no es totalmente negro, es un gris oscuro, un poco más claro para los “hint” de algunos EditText.

3.4.3 Layouts

Las distintas partes de nuestra aplicación vienen definidas por elementos rectangulares con los bordes redondeados, el color de fondo blanco rosado mencionado anteriormente y los bordes de un grosor considerable de color negro.

El hecho de hacer los bordes redondeados añadía cierta simpatía al usuario, añadía delicadeza y simpleza al diseño.

Los bordes pronunciados de color negro es un guiño a un videojuego llamado “Borderlands” el cual ayudó a este autor a sobrepasar los momentos más duros del desarrollo de la aplicación.

3.4.4 Detalles

En la implementación de PYMNet se han cuidado toda serie de detalles como el uso de Dialog para informar al usuario de los procesos que se realizan con la base de datos o de Toast para informar de pequeñas acciones o comentarios informativos.

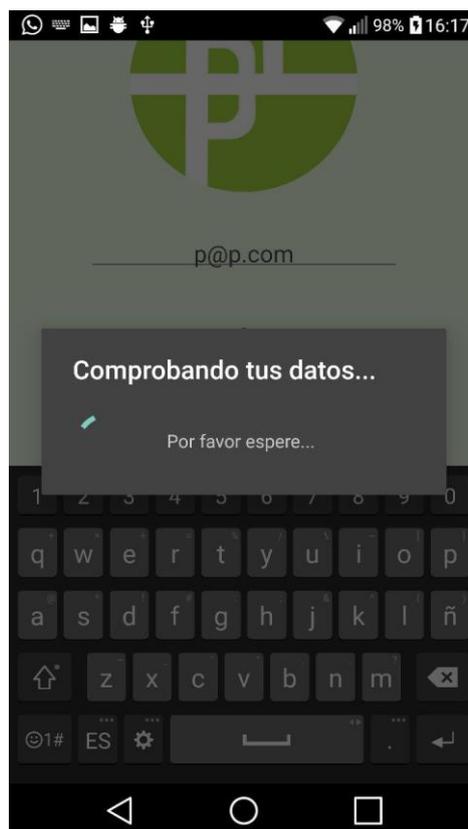


Figure 29: Dialog informado del proceso de Login.

3.5 ACTIVIDADES Y CLASES JAVA

3.5.1 AndroidManifest.xml

Cada aplicación Android debe tener un archivo AndroidManifest.xml (con ese mismo nombre). Es un archivo de configuración que contiene información acerca de la aplicación que el sistema tiene que conocer antes de poder ejecutar la aplicación.

Entre otras cosas, el AndroidManifest recoge el nombre de la aplicación, la versión de la misma, el icono de la aplicación, todas las actividades que componen la aplicación y los permisos de la aplicación.

En nuestro caso los permisos que requiere nuestra aplicación son los siguientes:

- Acceder al estado de la red y tener acceso a internet. Para realizar consultas a la base de datos y tener la interacción deseada es necesario internet.
- Poder leer y escribir en la memoria del dispositivo, esto nos servirá para subir imágenes y guardar algunas otras.
- Acceso a la localización del dispositivo. Esto está pensando para futuras actualizaciones donde la localización del usuario sirva a la hora de buscar empresas a su alrededor.

3.5.2 ActivityChooser

ActivityChooser es la actividad con la que se inicia nuestra aplicación, al hacer click en el icono de PYMNet, es esta actividad la que se ejecuta en primer lugar. Su funcionamiento es simple pero muy importante, ya que queremos conservar el login del usuario para evitar que tenga que introducir sus datos continuamente esta actividad nos conduce a la actividad de login si no se está logueado en ese momento o a la pantalla principal de nuestro usuario si estamos logueados.

Es en esta actividad donde se inicia la aplicación de Backendless para poder usar sus métodos y conectarnos a la base de datos en el resto de actividades de la aplicación.

3.5.3 Login

La funcionalidad básica de esta actividad es la de permitir al usuario iniciar sesión con sus datos de usuario (email y contraseña).

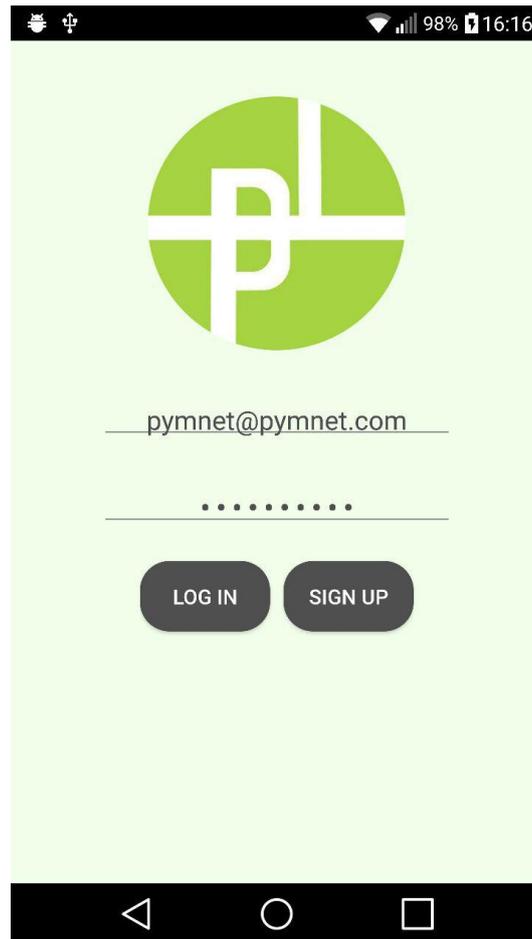


Figure 30: PYMNet. Login

Como se puede comprobar en la imagen, el diseño de esta actividad se basa en una imagen con el logo de PYMNet, dos EditText que nos permiten ingresar nuestros datos. Debajo del EditText de la contraseña se encuentra un TextView que nos informará de los problemas de nuestro ingreso y nos ayudará a recordar la contraseña en caso de que el usuario falle en sus intentos de iniciar sesión. Abajo tenemos dos botones, uno para iniciar sesión con los datos introducidos (botón de “LOG IN”) y otro, si es la primera vez que se abre la aplicación, para registrarnos en PYMNet (botón de “SIGN UP”). Como funcionalidad avanzada tenemos que los EditText tienen formato de email y de contraseña respectivamente, de este modo, el teclado en el campo de email tendrá un botón con el símbolo “@” y otro con “.com” que facilitan la experiencia del usuario. El campo de contraseña esconderá nuestra contraseña utilizando los típicos puntos. Para hacer el login hacemos uso de la librería de Backendless:

```
Backendless.UserService.login(usermail, password, new
BackendlessCallback<BackendlessUser>() {}, stayLoggedIn);
```

3.5.4 Register

La funcionalidad básica de esta actividad es la de mostrar un formulario de registro a los usuarios nuevos que quieran ingresar en PYMNet.

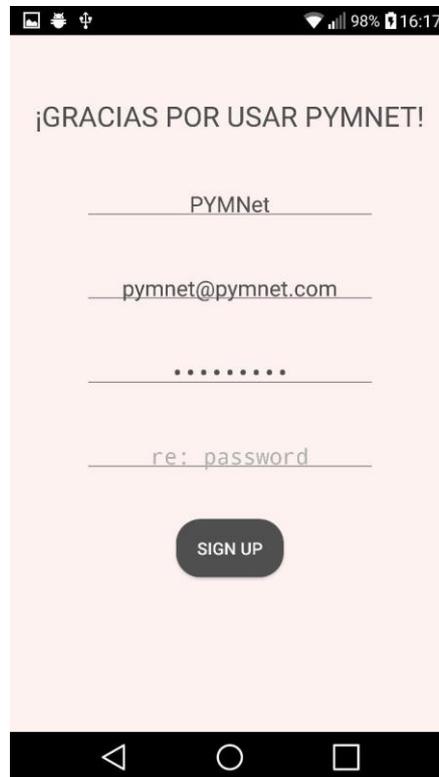


Figure 31: PYMNet. Registro.

Como se puede comprobar en la imagen, el formulario de registro consta de un TextView animando al usuario a utilizar PYMNet, cuatro EditText para el nombre del usuario, el email, la contraseña y repetir la contraseña, seguido del botón de “SIGN UP” para realizar el proceso y entrar en su perfil. El resto de datos los puede actualizar el usuario una vez ha ingresado en la aplicación.

De forma adicional hay un reconocimiento de los datos introducidos en el usuario que no permite poner email sin el formato de una dirección de correo electrónico o no permite contraseñas de menos de 8 caracteres, etc.

3.5.5 MainActivity

Esta actividad, como su nombre indica, es la actividad principal, donde se encuentra el grosso de la aplicación. En ella se ha implementado un diseño de pestañas para dividir la aplicación en cinco partes, todas ellas serán explicadas a continuación. Para implementar el diseño de pestañas se ha utilizado el elemento TabLayout y ViewPager, en el layout de

la actividad. En el código Java se han inicializado estos elementos así como las instancias de las distintas pestañas (Fragments), hemos creado un Adapter personalizado para agregar nuestras pestañas al TabLayout. El resultado lo podemos ver en la imagen, un diseño de cinco pestañas por las cuales podemos movernos simplemente arrastrando el dedo desde el lado que queramos movernos.



Figure 32: PYMNet. Distribución en Pestañas.

3.5.6 Tab_User

La primera de nuestras pestañas es la pestaña de usuario, donde encontramos la información de nuestro perfil de usuario.

Esta pestaña muestra información como nuestra imagen de perfil, nuestro nombre, una cita personalizable, información de nuestras redes sociales, un bloque donde podemos contar algo más acerca de nuestra persona o la actividad reciente realizada.



Figure 33: PymNet. Pestaña de Usuario.

Para poder definir la pestaña hemos extendido de la clase Fragment, de este modo, la activity es la misma pero vamos cambiando el Fragment que la define. Para la imagen de perfil usamos la ya nombrada librería Picasso que nos permite ajustar la imagen a su recipiente optimizando el uso de memoria del teléfono.

Cada una de las redes sociales activadas por el usuario lleva asociado una acción. Al pulsar en cualquiera de los iconos se iniciará la aplicación en cuestión, si está instalada en el dispositivo, o se iniciará mediante el navegador. En caso de Gmail, se podrá enviar un correo a dicha dirección, las redes sociales abrirán el perfil de usuario, etc. Las entradas del apartado “Mi Actividad” nos llevaran a los perfiles de los usuarios o empresas relacionados con el evento mediante las actividades de OpenUser y Open_empresa.

3.5.7 ImageViewer

Esta actividad nos permite ver las imágenes de perfil y otras imágenes en miniatura en pantalla completa, su funcionalidad es simple pero añade seriedad y una buena experiencia al usuario.

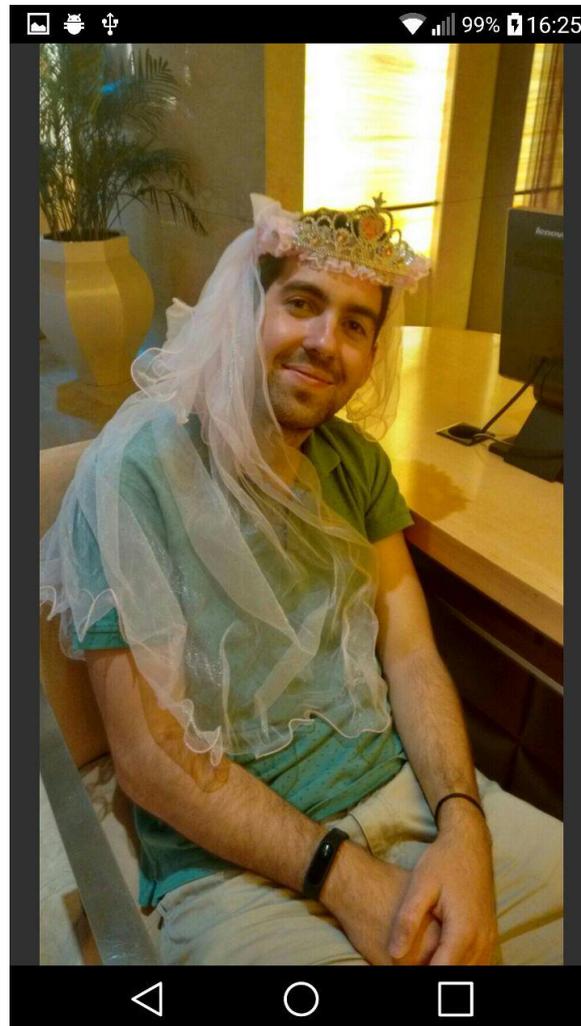


Figure 34: PYMNet. Visor de Imágenes.

3.5.8 Tab_Favoritos

La funcionalidad básica de la pestaña de favoritos es la de organizar los usuarios y las empresas que seguimos para tener un acceso rápido a ellos. Consta de tres partes un EditText para filtrar la búsqueda por nombre de usuarios o de empresas y dos partes iguales que contienen los usuarios y las empresas a las que seguimos.



Figure 35: PYMNet. Pestaña de Favoritos

Como se puede observar en la imagen, ambas listas son horizontales, por lo que se ha implementado un scroll horizontal en cada una para poder visualizar todos los usuarios o empresas. Al pulsar a un usuario o empresa de la lista se abrirá su perfil mediante las actividades `OpenUser` y `Open_empresa`.

Para ordenar a los usuarios y las empresas se han realizado sendas llamadas a la base de datos `UserEmpresas` y `UserFriends` que nos devuelve la lista de los usuarios y empresas que sigue el usuario. Para mostrarlos de una forma ordenada se ha utilizado un `ListView` con un `Adapter` personalizado que consta de la imagen de perfil del usuario/empresa y el nombre del mismo/a.

3.5.9 Tab_empresa

En esta pestaña tenemos acceso a las distintas empresas creadas por el usuario así como podremos crear una nueva. Se ha impulsado la idea de que un usuario pueda tener varias empresas asociadas por el rol que tienen algunas empresas de llevar la publicidad a otras.

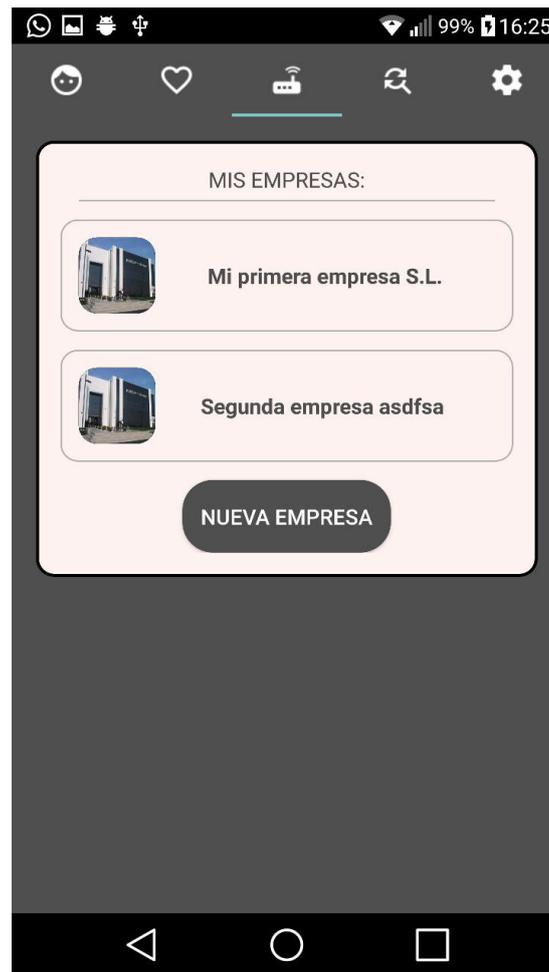


Figure 36: PYMNet. Pestaña de Empresas.

El rol de community master puede establecerse en PYMNet. Su diseño es sencillo, una lista de las empresas asignadas al usuario y un botón para crear una empresa nueva.

3.5.10 Nueva_Empresa_1

Al pulsar en el botón de crear empresa en la pestaña de “Tab_empresa” se inicia esta actividad que contiene un formulario con la información necesaria para crear la empresa así como un menú de selección de los distintos servicios ofertados por la aplicación en estos momentos. Los servicios disponibles actualmente son: Comentarios, Trabajadores y Catálogo.

Figure 37: PYMNet. Formulario de registro de empresa.

3.5.11 Nueva_Empresa_2

Al pulsar en el botón de siguiente de la actividad anterior, pasamos a esta actividad donde se configurarán los servicios seleccionados uno a uno, una vez hayamos finalizado este proceso, nuestra empresa estará correctamente registrada en PYMNet y será visible para todos. La configuración de los distintos servicios así como la información de la empresa es configurable en cualquier momento únicamente por el usuario que ha registrado dicha empresa.

3.5.12 Open_empresa

La actividad Open_empresa recibe como parámetro en el Bundle la empresa que tiene que mostrar, su función es la de mostrar la información de la empresa así como los servicios que esta implemente.



Figure 38: PYMNet. Perfil de Empresa EmpYtec.

En la imagen podemos observar la cabecera con el nombre de la empresa, información básica de contacto y uno de los servicios, el Catalogo.

3.5.13 OpenUser

Su función es la misma que la actividad anterior pero con el usuario en cuestión. Mostrando la información de perfil de usuario. Podremos seguir o dejar de seguir a un usuario y ver su actividad reciente.



Figure 39: PYMNet. Perfil de usuario. (Toast notificando que se ha seguido correctamente).

3.5.14 Tab_búsqueda

En esta pestaña nos centramos en la búsqueda de empresas y usuarios ya sea por nombre, por sector al que se dedica la empresa o por localización. Para hacer la búsqueda más amigable para el usuario se ha dividido la búsqueda en otras dos pestañas, de forma que podemos buscar usuarios o empresas de forma independiente y acogiéndonos al criterio seleccionado. La búsqueda se realiza con el nombre parcial, o el sector parcial. Es decir, buscando “Al” nos saldrán resultados como “Álex”, “Alicia”, “Álvaro”, etc.

- Tab_búsqueda_user: Es la pestaña de búsqueda de usuarios, como podemos observar en la imagen se basa en un EditText donde pondremos el nombre que deseamos buscar, los resultados se mostraran ordenados en un ListView que se autogenerará con los datos recibidos de la base de datos y se amoldarán con un adapter personalizado.

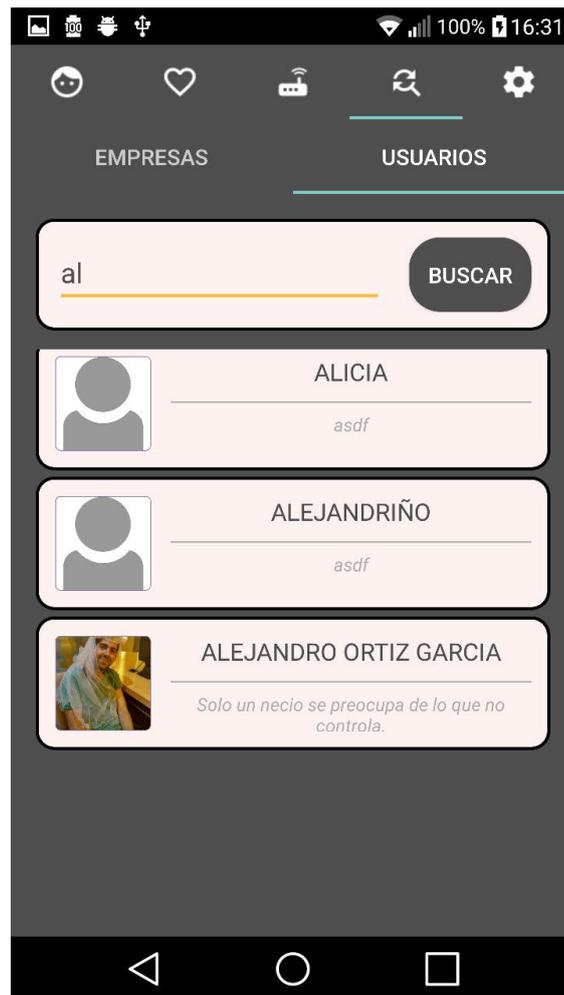


Figure 40: PymNet. Pestaña de Búsqueda de Usuarios.

- Tab_búsqueda_emp: Es la pestaña de búsqueda de empresas, podemos buscar empresas por sector (peluquería, carnicería, etc), por nombre o por localización. De la misma forma que los usuarios, los resultados se mostrarán ordenados en una lista.



Figure 41: PYMNet. Pestaña de Búsqueda de Empresas.

3.5.15 Tab_Settings

Esta pestaña está destinada a la configuración del usuario en la aplicación y en su perfil, de este modo tiene acceso a actualizar su información de usuario o de cerrar sesión de la aplicación.

Su diseño es simple, intuitivo, toda la aplicación se ha desarrollado teniendo en cuenta que sea fácil de utilizar, para todos los públicos.

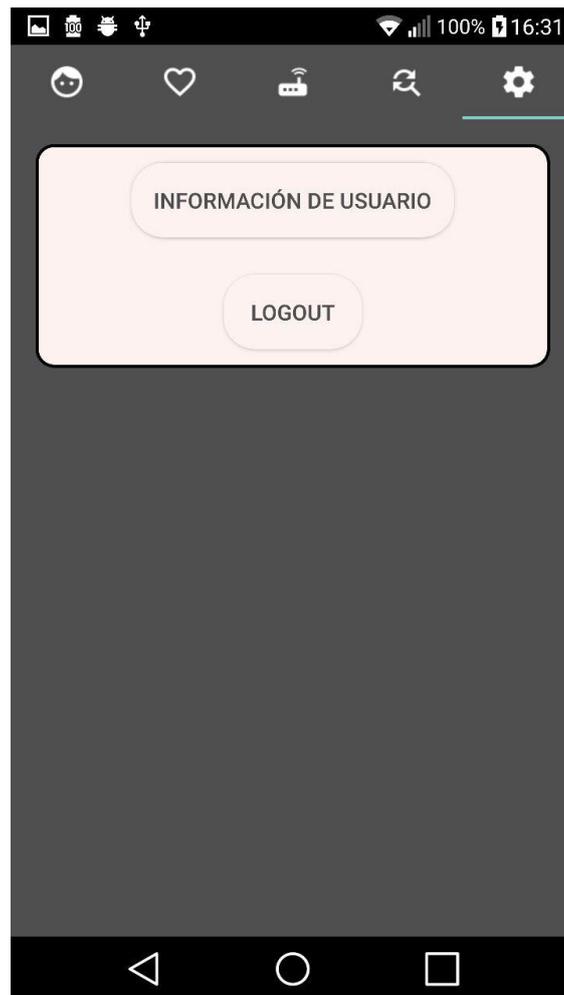


Figure 42: PymNet. Pestaña de Settings.

3.5.16 ConfigUser

En esta actividad podemos actualizar la información correspondiente a nuestro perfil, nombre, contraseña, redes sociales, imagen de perfil, etc. La gestión de la imagen de usuario se realiza subiendo la imagen a Backendless y guardando el enlace de dicha imagen en la correspondiente tabla de la base de datos Users, para evitar un colapso en cuanto a la subida de archivos, se limita a una imagen por usuario.

3.5.17 Objetos (clases) de la base de datos

La forma con la que Backendless trabaja con su base de datos a la hora de guardar entradas en las tablas es la de crear un objeto que contenga los mismos atributos que queramos guardar y guardamos una instancia serializada de dicho objeto, dicho de otra manera, para guardar una empresa, debe existir una clase Empresa y atribuirle las propiedades antes de realizar el método de `Backendless.Data.save()`.

Para la tabla Users ya existe la clase BackendlessUser pero es por esta razón por lo que para el resto de tablas creadas en la base de datos requerimos una clase para instanciar objetos de la misma. De este modo, tenemos las clases de Empresa, UserFriends, UserEmpresas, EventosEmpresas, EventosUsuarios, Comentarios, CatalogoItem, Trabajadores.

Estas clases constan de las propiedades necesarias y con métodos para establecer o recuperar el valor de cada una de estas propiedades.

En clases como UserFriends, las propiedades son:

- BackendlessUser usuario: es el usuario en cuestión.
- List<BackendlessUser> amigos: la lista de usuarios que sigue el usuario en cuestión.

Es por este motivo que para actualizar la lista debemos completarla por completo, por ejemplo, si queremos eliminar al amigo “B” de la secuencia “A”, “B”, “C”. Debemos crear un objeto UserFriends con la lista “A” y “C” para el usuario en cuestión y guardarla de nuevo sobrescribiendo la anterior entrada en la tabla.

4 CONCLUSIONES Y LINEAS FUTURAS

En este proyecto se ha implementado la aplicación “PYMNet”, una aplicación diseñada con el objetivo de fomentar la incursión de las PYMEs en el mundo de las aplicaciones móviles y dar el siguiente gran salto en cuanto a Internet se refiere (el primero fue el acceso a Web). Con esto se pretende crear un puente entre los potenciales clientes y las empresas registradas fomentando así la posibilidad de vínculos entre ambos de forma gratuita para los usuarios y muy económica para las empresas.

La metodología de trabajo ha sido la siguiente:

1. Estudio de viabilidad del proyecto: encuesta a varias empresas de diferentes sectores con el objetivo de comprobar la viabilidad comercial del proyecto y saber los pilares base que debe tener.
2. Empezar a pensar en la forma de la aplicación a rasgos muy básicos. Distinguir entre usuarios y empresas o dar la posibilidad de que los usuarios sean los que crean empresas, diseño básico y línea de diseño de la aplicación (colore, formas, etc).
3. Creación del proyecto en Android Studio. Primeras configuraciones del proyecto dando acceso a la aplicación a Internet o a la galería de imágenes, entre otras.
4. Creación de la aplicación en Backendless. Diseño de las distintas tablas de las bases de datos, definir las relaciones entre ambas y familiarización con la API de Backendless.
5. Desarrollo de la aplicación de forma escalable.

Para el desarrollo de la aplicación se han afianzado conocimientos estudiados en el grado y también se han adquirido nuevos conocimientos como:

- Diseño de aplicaciones Android.
- Sistemas de MBaaS como Backendless.
- Creación de distintos layouts.
- Uso de objetos propios de Android como Bundle o Intent.
- Fragments.
- Interfaz Parcelable.
- Ciclo de vida de las activities y fragments.

Es por todos los conocimientos afianzados y los nuevos conocimientos adquiridos que este proyecto ha sido un éxito a nivel personal, dándome la posibilidad de crear mi propia aplicación donde los aciertos, pero también los errores son míos aprendiendo en todos los aspectos del desarrollo de aplicaciones.

4.1 LÍNEAS FUTURAS

PYMNet se encuentra en un estado de desarrollo que podríamos calificar como Beta, en estos momentos es una aplicación que dispone de los mínimos que se tenían previstos. Un servicio de usuarios y empresas, con interacción entre ambos y un pack de 3 servicios básicos. Es por esta razón que su expansión puede ser grande en varias líneas:

- Desarrollar más servicios para aumentar la interacción entre los usuarios y las empresas, de este modo se llegaría a un público mayor ya que se ofrecerían más servicios que pueden interesar a ambas partes. Entre estos servicios podríamos nombrar un servicio de citas previas, un servicio de reservas o un chat.
- Notificaciones: diseñar un sistema de notificaciones push para mantener a usuarios y empresas informados de las novedades o los eventos que les incumben.
- Integración con redes sociales, pese a que existen pequeñas interacciones con redes sociales se puede extender mucho más haciendo posible el login con Facebook o Google, buscando amigos en redes sociales, etc.
- Desarrollo en otras plataformas móviles como iOS o Windows Phone, incluso una versión Web.

BIBLIOGRAFÍA

- <http://stackoverflow.com>
- <http://www.xatakamovil.com/futuro/el-mercado-de-los-Smartphone-en-la-actualidad-suben-las-exigencias-y-baja-la-demanda>
- <http://www.elperiodico.com/es/noticias/sociedad/los-espanoles-usan-smartphone-para-acceder-internet-4463760>
- http://cincodias.com/cincodias/2015/09/02/pyme/1441206221_324356.html
- http://www.ipyme.org/publicaciones/retrato_pyme_2015.pdf
- <http://andro4all.com/2016/09/android-arrasando-cuota-mercado-ios-crece>
- https://www.android.com/intl/es_es/history/#/marshmallow
- <https://es.wikipedia.org/wiki/Android>
- https://es.wikipedia.org/wiki/Open_Handset_Alliance
- <http://androidzone.org/2013/05/historia-de-android-la-evolucion-a-lo-largo-de-sus-versiones/>
- <http://www.xatakandroid.com/sistema-operativo/android-es-libre-que-no-os-enganen>
- <https://androidsubway.files.wordpress.com>
- <http://www.androidcurso.com/index.php/99>
- <https://developer.android.com/studio/index.html?hl=es-419>
- https://es.wikipedia.org/wiki/Android_Studio
- https://es.wikipedia.org/wiki/IntelliJ_IDEA
- <https://en.wikipedia.org/wiki/Gradle>
- <http://www.arquitecturajava.com/que-es-gradle/>
- <https://www.jetbrains.com/>
- <https://en.wikipedia.org/wiki/JetBrains>

- https://es.wikipedia.org/wiki/Control_de_versiones
- <https://es.wikipedia.org/wiki/Git>
- https://backendless.com/documentation/users/android/users_requirements.htm
- <http://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>
- https://es.wikipedia.org/wiki/Backend_as_a_service
- <http://square.github.io/picasso/>
- <https://github.com/square/picasso>
- <http://www.androidcurso.com/index.php/tutoriales-android/41-unidad-7-seguridad-y-posicionamiento/282-el-esquema-de-permisos-en-android>
- <https://developer.android.com/guide/topics/manifest/manifest-intro.html>
- <http://www.tuprogramacion.com/glosario/que-es-el-android-manifest/>