ReciclAPP: recicla y gana

Cristóbal Zaragoza Linares
Master universitario en Desarrollo de Aplicaciones para dispositivos móviles

Responsable del Curso: Carles Garrigues

Consultor: Roger Montserrat Ribes

Fecha de Entrega: Octubre 2017 – Enero 2018

Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-CompartirIgual
3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*ReciclAPP: recicla y gana*
Nombre del autor:	*Cristóbal Zaragoza Linares*
Nombre del consultor:	*Roger Montserrat Ribes*
Fecha de entrega (mm/aaaa):	*01/2018*
Titulación:	Master universitario en Desarrollo de Aplicaciones para dispositivos móviles

Resumen del Trabajo (máximo 250 palabras):

ReciclAPP es una aplicación para smartphones que nos permite saber en que contenedor (vidrio, plástico, orgánico, etc) o en que local de recogida una persona puede reciclar un residuo en concreto. Simplemente, mediante un buscador sencillo, el usuario teclea el nombre del residuo en cuestión y el sistema te proporciona el tipo de contenedor al cuál debería ser correctamente reciclado.

Mediante la geolocalización del Smartphone, nuestra APP es capaz de mostrarte los locales de recogida más cercanos e informarte de su horario de apertura.

La propuesta inicial de implantación, la cuál trabajaremos, es la ciudad de Barcelona.

Además, puesto que cada vez es más importante mantener el ecosistema en el que vivimos limpio, y concienciar de ello a todos, ayudamos a que el usuario consiga además otros beneficios. Es por eso que ReciclAPP te permite ganar puntos cívicos que puedes obtener y canjear por regalos.

Abstract (in English, 250 words or less):

RecyclAPP is an application for smartphones that allows us to know in which container (glass, plastic, organic, etc.) or in which waste collection center can can you recycle an object in concrete. Just using a simple search engine, the user types the name of the waste in question and the system gives you the type of container to which it should be properly recycled.

Through the geolocation of the Smartphone, our APP is able to show you the nearest collection places and inform you of their opening hours.

The initial proposal of implantation, which we will work, is the city of Barcelona.

In addition, since it is increasingly important to maintain the ecosystem in which we live clean, and to make everyone aware of it, we help the user to obtain other benefits as well. That's why ReciclAPP allows you to earn civic points that you can get and redeem for gifts.

Paraules clau (entre 4 i 8):

Reciclar, envases, contenedores, Barcelona, smart city

Índice

1. Introducción....1
1.1 Contexto y justificación del Trabajo....1
1.2 Objetivos del Trabajo....3
1.3 Enfoque y método seguido....4
1.4 Planificación del Trabajo....5
1.5 Breve sumario de productos obtenidos....9
1.6 Breve descripción de los otros capítulos de la memoria....10

2. Usuarios y contextos de Uso....11
2.1 Contextos y Perfiles de Usuario....11
2.2 Fichas de Usuario....12

3. Diseño Conceptual....14

4. Prototipado....15

5. Evaluación....28

6. Diseño Técnico....29
6.1. Definición de los casos de uso....29
6.2. Diseño de la arquitectura....38

7. Desarrollo....41
7.1. Estructura....41
7.2. Plugins nativos Cordova....43
7.3. Otras herramientas de desarrollo y consideraciones....44
7.4. Pantallas más importantes....45
7.5. Funcionalidades incompletas....47
7.6. Decisiones tomadas durante el desarrollo....48
7.7. Backend y API....49
7.8. Revisión de la planificación....51
7.9. Pruebas....52

9. Glosario....55

10. Bibliografía....56

11. Anexos....57
11.1. Compilación del código fuente....57
11.2. Recursos externos – Ejemplos de códigos QR....60
11.3. Manual de Usuario....61

Lista de figuras

Ilustración 1. Logotipo de ReciclAPP-BriziklAPP 1
Ilustración 2. Diagrama de Gantt. PEC2: Diseño 6
Ilustración 3. Diagrama de Gantt, PEC3: Desarrollo 7
Ilustración 4. Diagrama de Gantt, Entrega Final 8
Ilustración 5: Foto Ficha Persona 1 12
Ilustración 6: Foto Ficha Persona 2 13
Ilustración 7: Árbol de Navegación 14
Ilustración 8: Prototipado, Splash 15
Ilustración 9: Prototipo, Inicio 16
Ilustración 10: Prototipado, Login 17
Ilustración 11: Prototipado, Registro 18
Ilustración 12: Prototipado, Mi Cuenta 19
Ilustración 13: Prototipado, Cambiar Ciudad 20
Ilustración 14: Prototipado, Reciclar / Buscar 21
Ilustración 15: Prototipo, Contenedor 22
Ilustración 16: Prototipado, Residuo especial y Mapa 23
Ilustración 17: Prototipado, Residuo especial y Detalles 24
Ilustración 18: Prototipado, Ganar Puntos 25
Ilustración 19: Prototipado, Mis Puntos 26
Ilustración 20: Prototipado, Los que más reciclan 27
Ilustración 21: Caso de Uso 1 29
Ilustración 22: Caso de Uso 2 30
Ilustración 23: Caso de Uso 3 31
Ilustración 24: Caso de Uso 4 32
Ilustración 25: Caso de Uso 5 33
Ilustración 26: Caso de Uso 6 34
Ilustración 27: Caso de Uso 7 35
Ilustración 28: Caso de Uso 8 36
Ilustración 29: Caso de Uso 9 37
Ilustración 30: Estructura Ionic+Angular, https://ajgallego.gitbooks.io/ 38
Ilustración 31: UML Diseño BDD 39
Ilustración 32: Árbol de estructura Ionic compilada 41
Ilustración 33: Ejemplo de estructura de Componente Angular 41
Ilustración 34: Providers y fichero de configuración 42
Ilustración 35: Alerta de confirmación de permisos, ej: hacer fotos y videos 43
Ilustración 36: P. Dashboard 45
Ilustración 37: P. Top Stats 45
Ilustración 38: P. Búsqueda 45
Ilustración 39: P. Resultado 45
Ilustración 40: P. Mapa 45
Ilustración 41: P. Local 45
Ilustración 42: P. Login 46
Ilustración 43: P. Scan QR 46
Ilustración 44: P. Mi Perfil 46
Ilustración 45: P. Puntos 46
Ilustración 46: P. Ayuda 46

Ilustración 47: P. Logout 46
Ilustración 48: Backend - Listado de Residuos 49
Ilustración 49: Backend - Form Edición de Punto 49
Ilustración 50: API JSON Contenedores 50
Ilustración 51: Uso de npm install desde terminal 57
Ilustración 52: Uso de ionic serve desde terminal 58
Ilustración 53: Aplicación desde navegador (Chrome) 58
Ilustración 54: QR equivalente a 10 puntos 60
Ilustración 55: QR equivalente a 20 puntos 60
Ilustración 57: Manual, Acceder con tu cuenta 61
Ilustración 56: Manual, Inicio 61
Ilustración 58: Manual, Opciones de Navegación 62
Ilustración 59: Manual, Ejemplo de búsqueda 62
Ilustración 60: Manual, Botón Usuario 62
Ilustración 63: Manual, Acceder al Top10 63
Ilustración 61: Manual, Perfil 63
Ilustración 62: Manual, Mis Puntos 63

1. Introducción

1.1 Contexto y justificación del Trabajo

Normalmente nos encontramos con que no sabemos exactamente dónde debemos reciclar un objeto o residuo en concreto, y para los que queremos vivir en un mundo más sostenible nos puede suponer un quebradero de cabeza.

Además, hoy en día todavía existen errores comunes a la hora de reciclar. Por ejemplo, un vaso de cristal NO debe ir al contenedor de vidrio (verde), si no al de residuos (gris).

Es por eso que con esta APP, podrás buscar el tipo de residuo que deseas reciclar y el sistema será capaz de indicarte el tipo de contenedor adecuado para su reciclaje. También, cuando nos encontramos con casos de residuos especiales (aceite, pilas, electrodomésticos…) nuestra APP también te indicará aquellos puntos de recogida de residuos más cercanos a tu posición.

La propuesta del TFM solamente afecta a la ciudad en dónde vivo, Barcelona. Esta ciudad consta con una población considerable de habitantes (y por tanto, usuarios finales) y con una amplia red de contenedores y locales de recogida de residuos (Punt Verd[1]), tanto móviles como fijos (al menos uno de cada tipo por barrio / zona)

En el desarrollo del TFM, he buscado otras APPs similares en el mercado y he encontrado ReciclAPP-Briziklapp, cuya implantación sólo atañe a la comunidad foral de Navarra. Este producto informa además de cómo se recicla el residuo una vez depositado y recogido (http://www.ganasa.es/areas/calidad-ambiental/app_movil1.aspx)

Ilustración 1. Logotipo de ReciclAPP-BriziklAPP

También, desde la propia web del Ajuntament de Barcelona, existe un selector de residuo y la indicación posterior del contenedor correspondiente ofrecido por

[1] Punt Verd: sirven para deshacernos de los residuos que no podemos tirar a los contenedores que encontramos en la calle. Utilizando los puntos verdes contribuimos a mejorar el proceso de reciclaje y ayudamos a preservar el medio ambiente.

el sistema (http://ajuntament.barcelona.cat/ecologiaurbana/ca/serveis/la-ciutat-funciona/manteniment-de-l-espai-public/gestio-de-neteja-i-residus/cercador-de-residus)

A partir de mi idea inicial y de la combinación de ideas ya existentes, intento aportar un punto de vista para “fidelizar” el reciclaje, de manera que, mediante un QR, que se encontrará en cada local de recogida, el usuario podrá escanearlo y conseguir puntos cívicos CIVIS[2]. Estos puntos podrán ser gastados en aquellos productos, acciones y / o servicios que se encuentran en la página de CIVI CLUB.

El resultado final es poder obtener un producto, el cual, mediante un Smartphone y conexión a la red, una persona pueda saber dónde reciclar un residuo en cuestión en 1 minuto y además, en el caso de que el residuo necesite depositarse en un Punt Verd o similar, puede ganar puntos y así fomentar el reciclaje en su vida y en la de su comunidad de amigos, familia, etc.

[2] CIVIS: Moneda virtual impulsada por CIVI CLUB (https://www.civiclub.org/). Se puede utilizar para conseguir regalos: cheques descuentos, donaciones de libros y comida, etc

1.2 Objetivos del Trabajo

1.2.1. Fases e hitos

- Planificación correcta y realista del alcance del TFM
- Definición de usuarios y ámbitos de uso
- Diseño de los caso de uso
- Diseño de las pantallas (prototipado)
- Diseño de la arquitectura
- Desarrollo e implantación
- Realización de pruebas

1.2.2. Objetivos Funcionalidades del producto (ordenados por prioridad)

1. Búsqueda de un residuo en concreto
2. Informar del contenedor o local de recogida asociado al residuo buscado. Si fuera el caso, informar de la dirección exacta y del horario de los puntos de recogida má
s cercanos
3. Registro y login de un usuario
4. Conseguir puntos cívicos mediante la lectura de un código QR una vez se ha reciclado en cualquiera de los locales de recogida
5. Informar de estadísticas de la APP: Quien tiene más puntos, Que es lo que la gente más busca / recicla
6. Conseguir regalos según los puntos obtenidos

1.2.3. Objetivos NO funcionales del producto (ordenados por prioridad)

1. Mejorar el medio ambiente
2. Fomentar el “juego” del reciclaje entre la comunidad de usuarios
3. Obtener una experiencia de usuario óptima
4. Mostrar los resultados de una manera rápida y eficaz (tiempo de respuesta)

1.3 Enfoque y método seguido

Se recurrirá a una metodología AGILE, principalmente por:

- Los tiempos necesarios para la implantación son algo justos.
- Pueden existir varios cambios, debido a que el producto se debe desarrollar desde 0.
- Permite cierta flexibilidad de cara a añadir nuevas funcionalidades (o quitar) dado a que se realizarán pruebas según se terminen los módulos planteados del producto.
- Llegado a un caso extremo, el producto puede carecer de alguna funcionalidad avanzada pero la funcionalidad principal puede no verse afectada.

El producto final será una aplicación híbrida multiplataforma: Encaminada a todo tipo de usuarios (Android, iOS, gente mayor, gente joven, etc).

NO requiere de un alto nivel de uso nativo, puesto que la geolocalización hoy en día es soportada por la totalidad de los navegadores HTML5 y de fácil implementación. También existen librerías compatibles para poder leer códigos QR.

El desarrollo será bajo el framework de Ionic, de esta manera los lenguajes utilizados serán: HTML5, CSS (sass) y Typescript. La combinación de estas herramientas garantizan una gestión de la aplicación rápida y escalable.

1.4 Planificación del Trabajo

1.4.1. Recursos utilizados para el desarrollo:

- SublimeText: para el desarrollo de aplicaciones con Angular2 (HTML5, CSS, TypeScript)
- Portátil MacBook Air 13’’ 1,3GH Intel Core i5: desarrollo y documentación de todo el TFM
- Smartphone Samsung Galaxy J7: pruebas de ejecución de la APP
- Ionic Dev: app necesaria para acceso a pruebas en Smartphone
- Servidor Backend: API y CMS instalado en servidor ofrecido por la empresa en la cual trabajo, Dommia Design Studio.

1.4.3. Total de horas por hitos

- PEC2, Diseño: 44 horas
- PEC3, Desarrollo: 136 horas
- Entrega Final: 30 horas
- Total: 210 horas

- 1.4.2. Hito: PEC2, Diseño

  La fecha de inicio propuesta es el 12 de Octubre, y la de finalización el 27 de Octubre (debido a un viaje al extranjero, no se realizarán tareas entre el 28 de Octubre y el 5 de Noviembre).

  Las horas estimadas para cada tarea serían:

  1- Usuarios y contexto de Uso: 8 horas (4 en 2 días laborables y 4 en 1 día festivo)

  2- Diseño Conceptual: 12 horas (8 en 4 días laborables y 4 en 1 día festivo)

  3- Prototipado: 16 horas (8 en 4 días laborables y 8 en 2 días festivos)

  4- Diseño Técnico: 6 horas (6 en 2 días laborables)

  5- Evaluación y Entrega: 2 horas (2 en 1 día laboral)

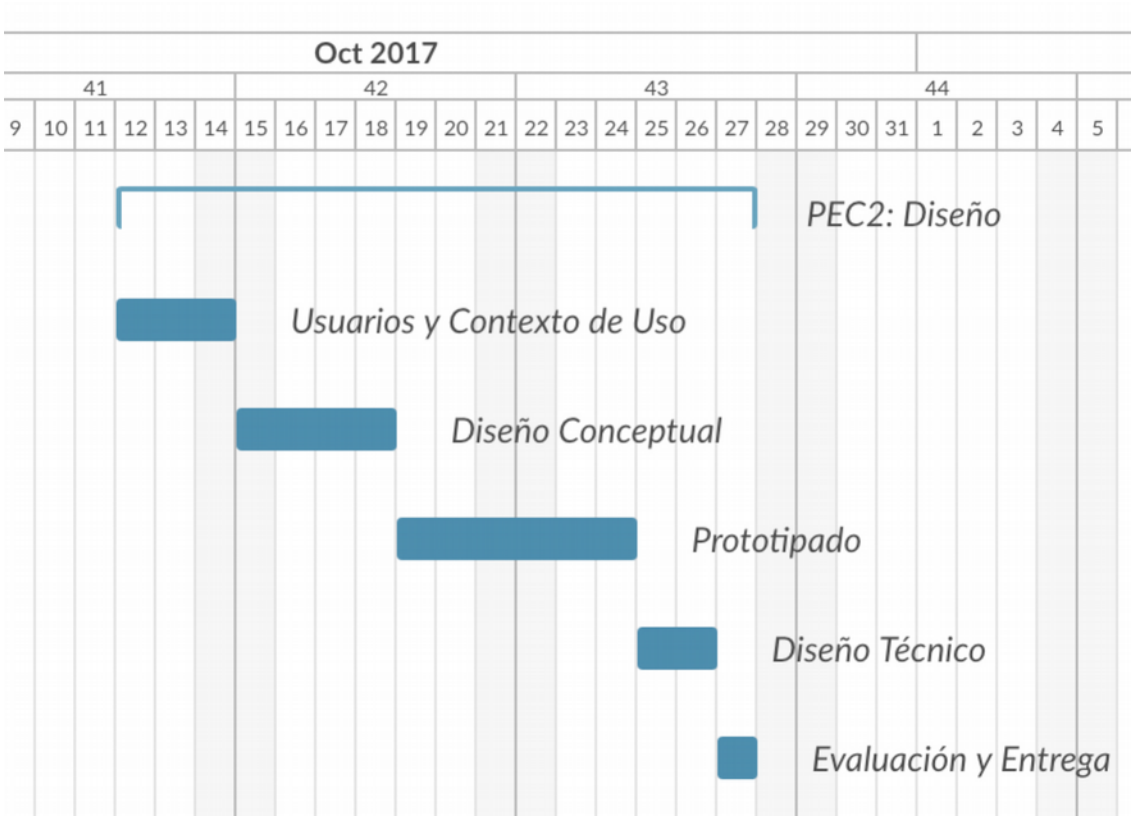

Ilustración 2. Diagrama de Gantt. PEC2: Diseño

1.4.3. Hito: PEC3, Desarrollo

La fecha de inicio propuesta es el 6 de Noviembre, y la de finalización el 13 de Diciembre (debido a un viaje al extranjero, no se realizarán tareas entre el 28 de Octubre y el 5 de Noviembre).

Las horas estimadas para cada tarea serían:

1- Arquitectura BDD: 4 horas (4 en 2 días laborables)

2- Configuración CMS: 4 horas (4 en 2 días laborables –CMS de desarrollo propio creado en 2016 de fácil instalación: DOMMIA-)

3- Desarrollo Frontend API: 104 horas (36 en 18 días laborables y 68 en 10 días festivos)

4- Desarrollo API (incluye inserción de datos): 18 horas (2 en 1 día laborable y 16 en 2 días festivos)

5- Pruebas: 2 horas (2 en 1 día laboral)

6- Documentación y Entrega: 4 horas (4 en 2 día laborables)

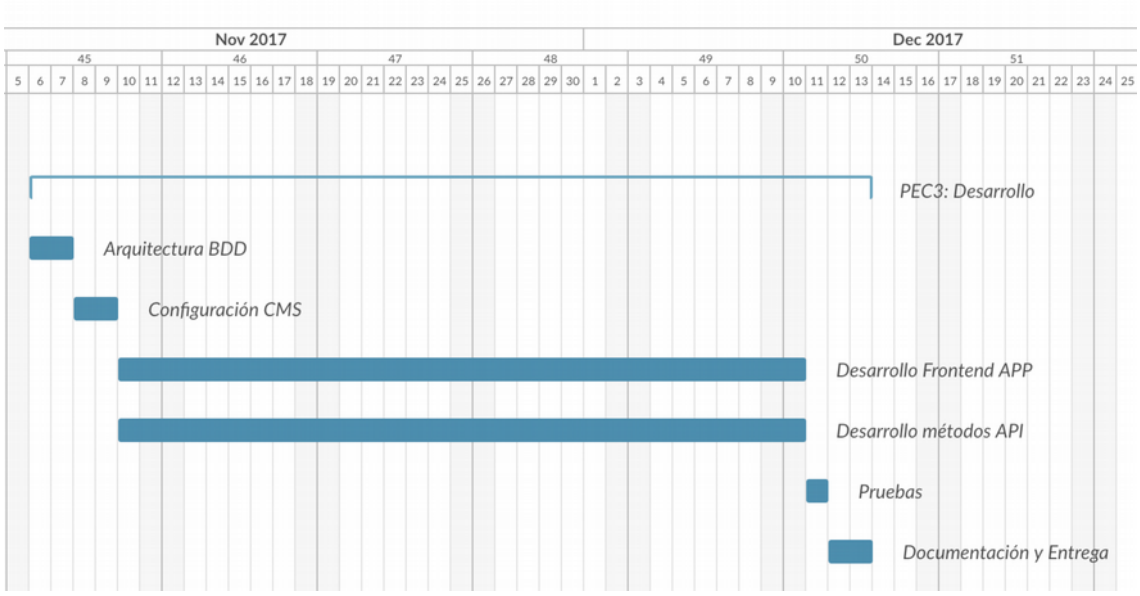

Ilustración 3. Diagrama de Gantt, PEC3: Desarrollo

1.4.4. Hito: Entrega Final

La fecha de inicio propuesta es el 14 de Diciembre, y la de finalización el 3 de Enero (debido a las fiestas de Navidad, no se realizarán tareas entre el 23 de Diciembre y el 25 de Diciembre).

Las horas estimadas para cada tarea serían:

1- Posibles Mejoras: 4 horas (4 en 2 días laborables y 8 en 2 días festivos)

2- Presentación: 12 horas (8 en 8 días laborables y 4 en 1 día festivo)

3- Video: 8 horas (2 en 1 día laboral y 6 en 2 días festivos)

4- Memoria y Entrega: 6 horas (6 en 3 días laborables)

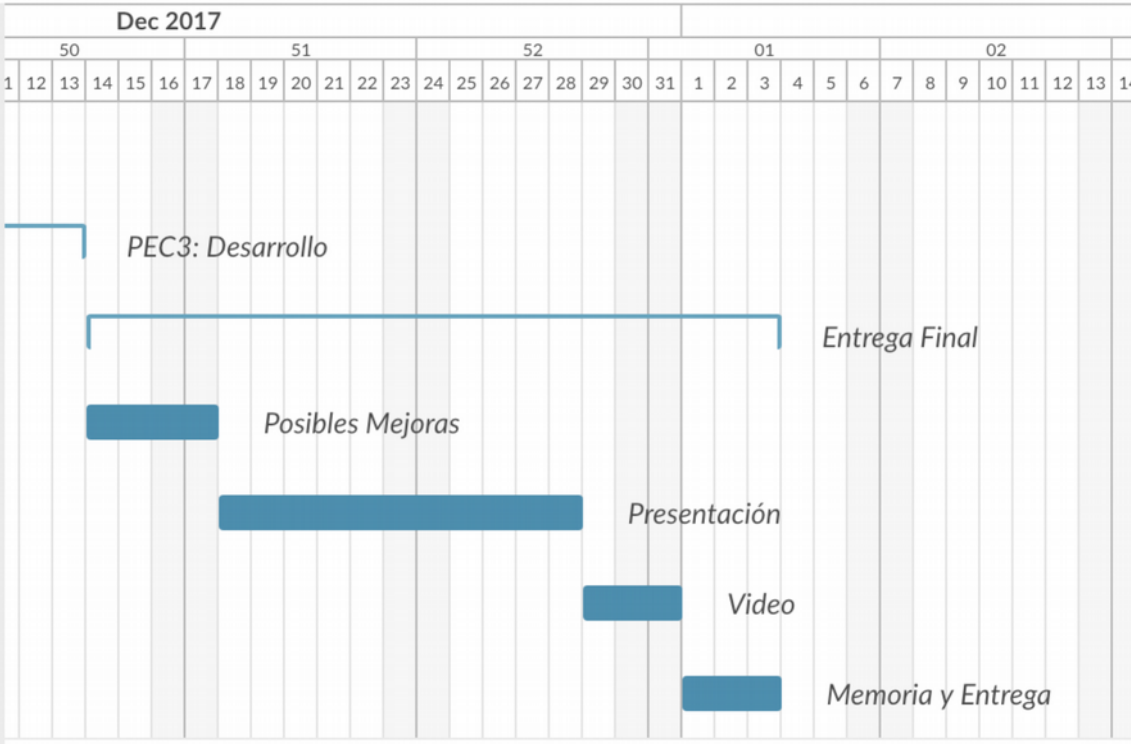

Ilustración 4. Diagrama de Gantt, Entrega Final

1.5 Breve sumario de productos obtenidos

En este punto, todavía no hemos podido desarrollar un producto entregable, no obstante el producto final constará de:

1- Memoria de trabajo
2- Presentación PDF
3- Presentación Video
4- Código fuente
5- Enlace a repositorio en GitLab
6- Aplicación fichero “compilado” en binario .apk
7- Manual de compilación / despliegue del código fuente del proyecto

1.6 Breve descripción de los otros capítulos de la memoria

El resto de capítulos, que coinciden con cada entrega son:

- Cap. 2. Diseño: Diseño desde un punto de vista del Usuario (casos de uso, prototipado, etc) y también técnico (entidades UML, BDD, etc)

- Cap. 3. Desarrollo: Creación del producto final utilizando las distintas técnicas aprendidas, en nuestro caso, WebApp desarrollada con Angular2.

- Cap. 4. Entrega final: Conclusiones, glosario, anexos, memoria del trabajo, presentación escrita y en captura de vídeo. Nos da una visión global de todo el proyecto.

2. Usuarios y contextos de Uso

2.1 Contextos y Perfiles de Usuario

ReciclAPP es una aplicación móvil desarrollada para todo tipo de personas, ya que el ámbito que ocupa es el del reciclaje convencional (residuos que se generan en el hogar o en una oficina) y no requiere un alto nivel técnico en el uso de APPs. Abastece cualquier edad y sexo y condición social. Añade además la posibilidad de fomentar el reciclaje mediante la adquisición de puntos cívicos, previo a un registro.

Por tanto, tendremos dos tipos de usuarios:

A) Perfil de usuario registrado

Uso muy habitual. Sin problemas para poder desplazarse hasta un local de reciclaje. Vive en ciudades con mayor número de contenedores y locales de recogida de residuos. Menores de 50 años. Se desenvuelven mejor en el uso de APPs a diario.

B) Perfil de usuario no registrado

Uso menos frecuente de nuestra APP y de cualquier APP en general. Menos predisposición a acudir a un centro a varios metros o kilómetros de su hogar u oficina. Mayores de 50 años.

Ambos perfiles de usuario tienen acceso a las siguientes funcionalidades:

- Buscar un residuo para poder reciclarlo
- Escoger ciudad
- Ver estadísticas de lo más reciclado y quién es el que más recicla

Sólo los usuarios que estén registrados (además de las funcionalidades anteriores pueden):

- Acudir a un punto dónde escanear un código QR y ganar puntos
- Cambiar puntos por regalos, vales de descuento, etc
- Modificar sus datos personales (perfil de usuario)

NOTA: Existe otro actor que puede intervenir en nuestra APP y es el operario de un local de reciclaje. Éste será el que muestre el código QR al usuario para poder escanearlo, pero no necesita hacer uso de nuestro producto, ya que estos QRs serán suministrados tras un hipotético acuerdo comercial entre la gestora de estos locales y ReciclAPP (supuesto que no se contempla en este TFM pero en el cual si que se simularán unos códigos a modo de pruebas sin que estos sean facilitados en ningún local).

2.2 Fichas de Usuario

Ficha 1

Ilustración 5: Foto Ficha Persona 1

Nombre: José Sirvent Navarro

Sexo: Hombre

Edad: 38 años

Reside en: Barcelona

Ocupación: CEO consultoría

Jornada Laboral: Autónomo (10 horas aprox. al día)

Estado civil: Casado

Experiencia uso APPs: Alto, experto

Idiomas: Castellano, Catalán, Inglés

Descripción de la persona:

José tiene una pequeña empresa que se dedica a la consultoría. Tiene 3 trabajadores a su cargo. Comparte con ellos una oficina a unos 35 kilómetros de su casa. Cada mañana se traslada en tren hasta su puesto de trabajo. Utiliza una APP para compartir ciclomotores, y las utiliza alguna vez para reuniones y comidas. Anteriormente trabajaba en una agencia de diseño de interiores. Entre sus hobbies se encuentran los videojuegos y viajar con su mujer al menos 4 veces al año.

Descripción del escenario:

En la oficina se consumen muchos folios, cartuchos de tinta, material de oficina diverso, cápsulas de café, botellas de agua, dispositivos periféricos defectuosos, etc. Una vez por semana, los jueves por la mañana, le dedica un tiempo a organizar estos residuos generados y a limpiar la oficina. A escasos 150 metros dispone de un “Punt Verd”, el cuál suele frecuentar cada jueves llevando todos los residuos que no pueden depositarse en ningún contenedor convencional.

Ficha 2

Ilustración 6: Foto Ficha Persona 2

Nombre: Iván Lorenzo García

Sexo: Hombre

Edad: 34 años

Reside en: Madrid

Ocupación: Programador Web

Jornada Laboral: 40 h / semana

Estado civil: Soltero

Experiencia uso APPs: Alto, experto, más de 50 apps instaladas en su dispositivo móvil

Idiomas: Castellano

Descripción de la persona:

Iván trabaja en una pequeña agencia de diseño gráfico y web como programador. Cada día se desplaza a pie hasta la oficina donde realiza una jornada laboral desde las 9 hasta las 18h. Dispone de pequeños descansos para almorzar y merendar y de 1 hora para comer. Al finalizar su jornada laboral juega a baloncesto 2 días por semana. El resto de días suele realizar la compra del hogar, limpieza, preparar la comida y cena, etc. Los fines de semana suele pasarlo con amigos o con la familia que viven a unos 450 km en tren.

Descripción del escenario:

Durante el día a día, Iván cocina en su casa y genera varios residuos, tales como, latas de atún, botellas de agua, servilletas, restos de comida, huesos de fruta, envases de porexpan, bolsas de plástico, aceite de cocina, etc. Tiene varios cubos en dónde separa los residuos generados. Cada 2 días más o menos los vacía en sus respectivos contenedores. A veces, duda en qué contenedor se debe depositar determinados residuos.

3. Diseño Conceptual

Mediante el siguiente Mapa de Navegación, podemos resumir de una forma clara los flujos por los cuales cualquier usuario pasa hasta llegar a la acción deseada.

Desde una pantalla inicial, podemos acceder todas las diferentes pantallas posibles. Podemos ver claramente los caminos de los usuarios registrados y los que no, para poder llegar a ellas.

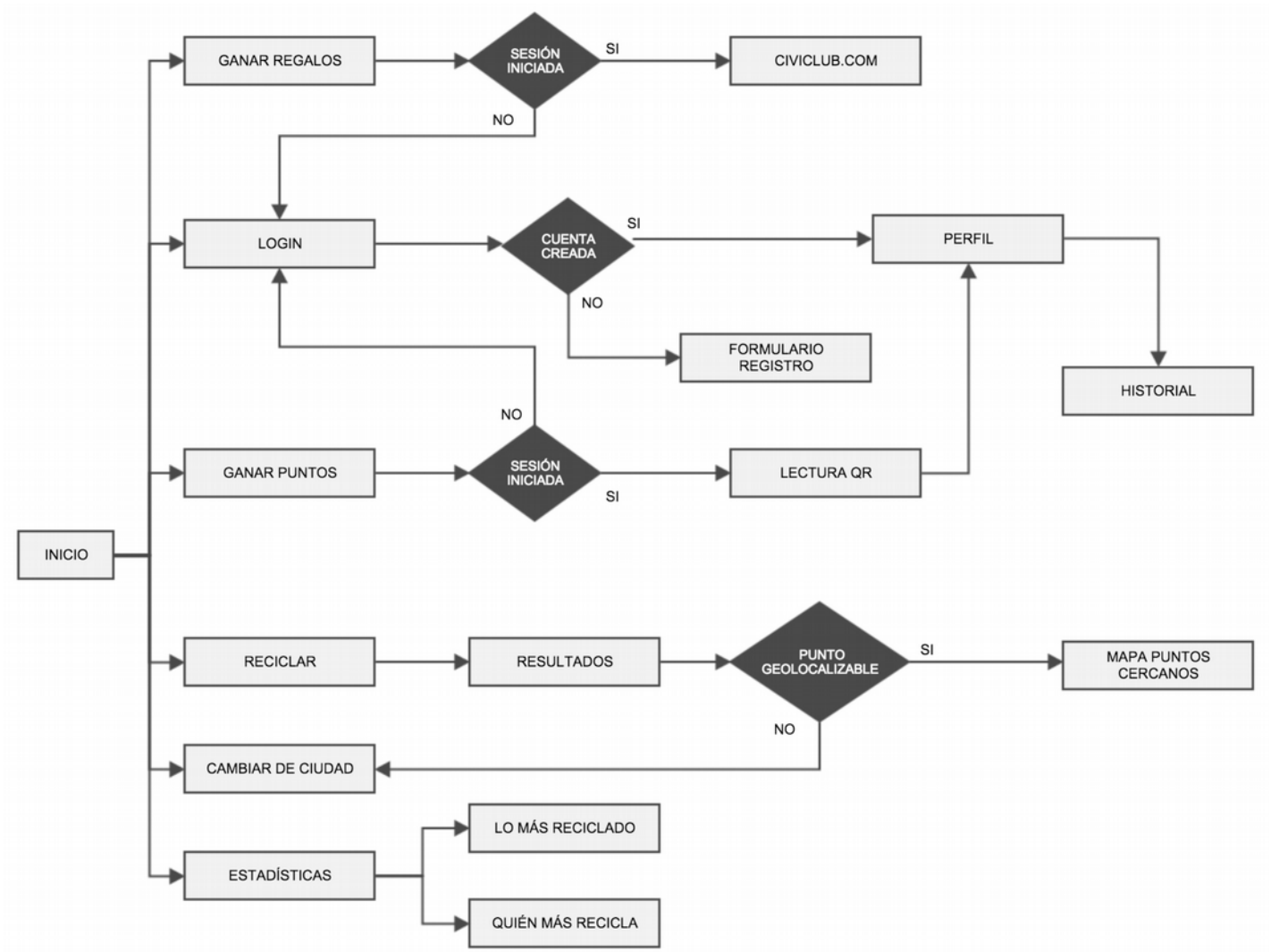

Ilustración 7: Árbol de Navegación

4. Prototipado

4.1. Pantalla de Splash

Es la pantalla inicial. Se le pide al usuario que acceda mediante registro o si lo prefiere lo puede hacer más tarde. Esta pantalla no volverá a aparecer.

Ilustración 8: Prototipado, Splash

4.2. Pantalla de Inicio

Desde aquí podemos acceder a las principales pantallas. Se muestra el total de puntos acumulados y un resumen de las estadísticas actuales (quién es el que más recicla y qué es lo que los usuarios más buscan).

Ilustración 9: Prototipo, Inicio

4.3. Pantalla de Login

Formulario de login básico mediante dirección de email. Opción de navegar a un formulario de registro, en el caso de no poseer una cuenta, y al proceso de recuperación de contraseña si se ha olvidado de ella.

Ilustración 10: Prototipado, Login

4.4. Pantalla de Registro

Formulario de registro de usuario. Se pide un nombre, una dirección de email y una contraseña como campos obligatorios. Como campos optativos tenemos: la foto de perfil, la ciudad dónde reside y los apellidos. Se debe aceptar la política de privacidad de la APP para poder finalizar el proceso.

Ilustración 11: Prototipado, Registro

4.5. Pantalla de Mi cuenta

Aparece un resumen del perfil del usuario con la sesión iniciada en el dispositivo. Se muestra un listado con las funcionalidades restantes que no aparecen en la pantalla de inicio (cerrar sesión, acceso a civiclub, conseguir regalos y editar el perfil).

Ilustración 12: Prototipado, Mi Cuenta

4.6. Pantalla de Cambiar Ciudad

Se muestra un desplegable con las distintas ciudades del sistema (en nuestro TFM contemplaremos sólo el caso de Barcelona). Puede darse el caso de que el usuario no escoja ninguna ciudad. Esta pantalla está accesible siempre desde la TabBar inferior.

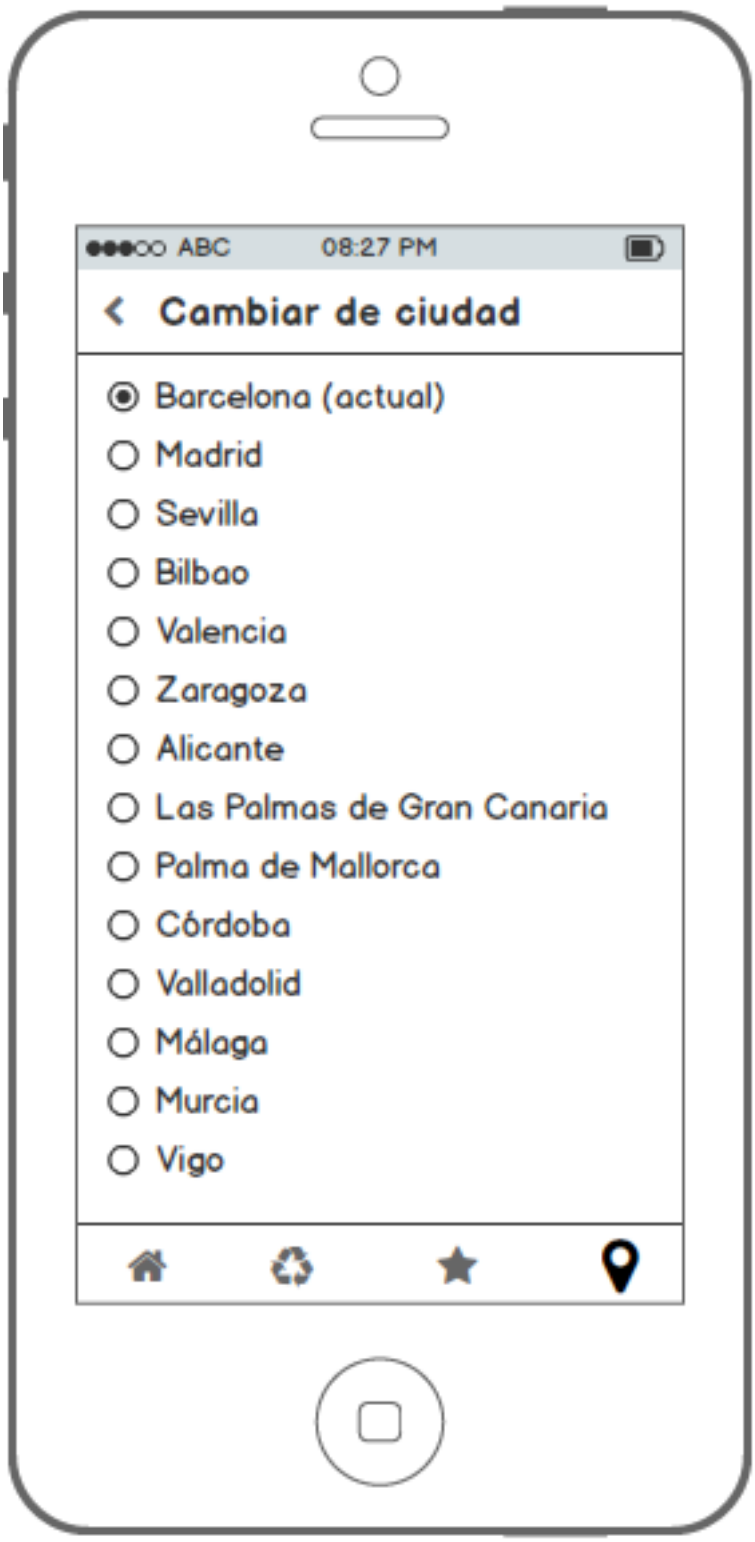

Ilustración 13: Prototipado, Cambiar Ciudad

4.7. Pantalla de Reciclar / Buscar residuo

Desde teclado, el usuario escribe el nombre del residuo que desea reciclar. Los resultados se muestran a modo de lista autocompletada con los elementos coincidentes mientras se teclea la palabra en concreto. Para seleccionar el residuo deseado, el usuario debe hacer “tap” sobre el elemento de la lista.

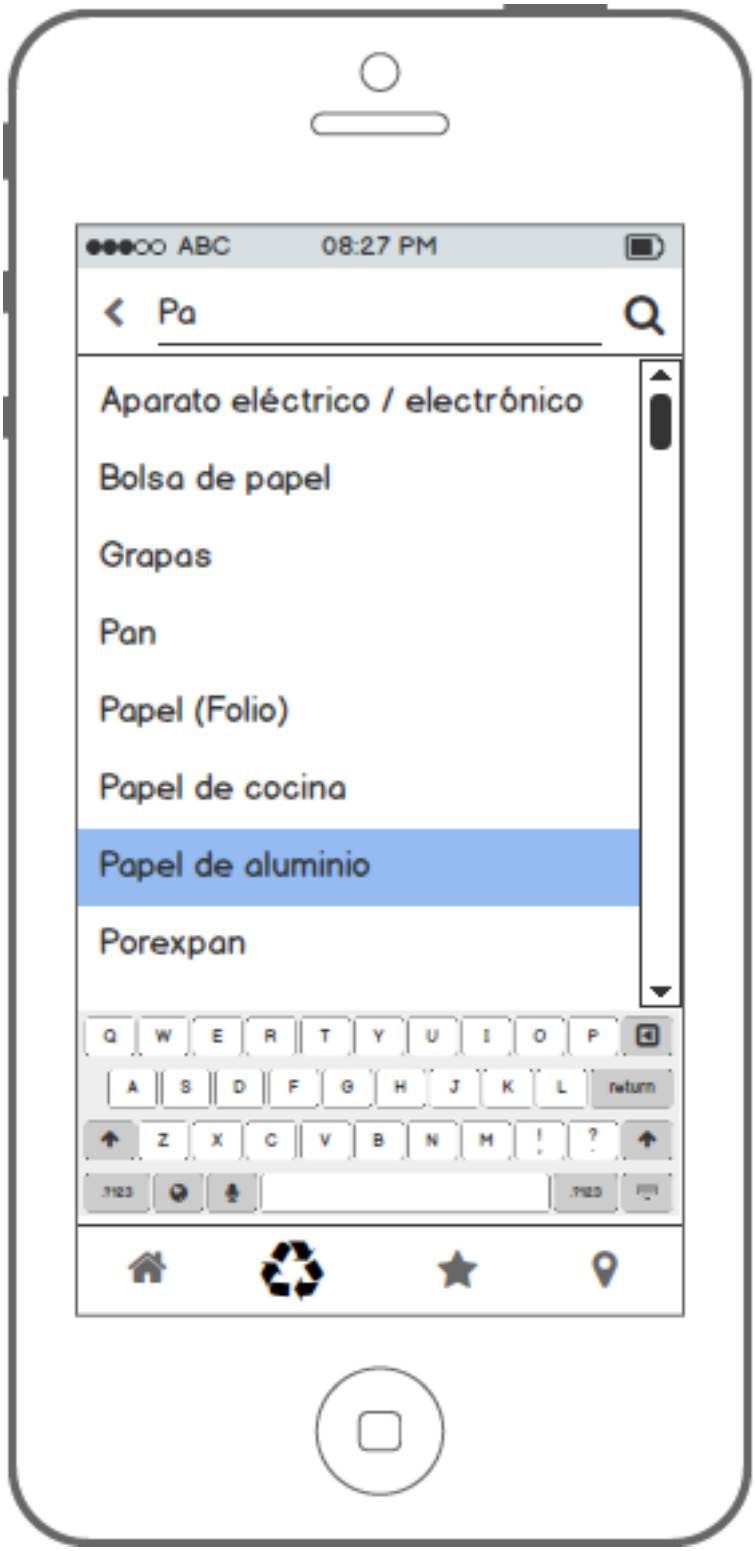

Ilustración 14: Prototipado, Reciclar / Buscar

4.8. Pantalla de Residuo y contenedor

Una vez seleccionado el residuo en cuestión, aparece una pantalla modal con la información del contenedor en dónde se debe depositar.

Ilustración 15: Prototipo, Contenedor

4.9. Pantalla de Residuo y mapa de local

Si el residuo en cuestión tiene un local específico asociado, se mostrará un mapa con los puntos en esa ciudad en dónde el usuario puede depositar el objeto a reciclar.

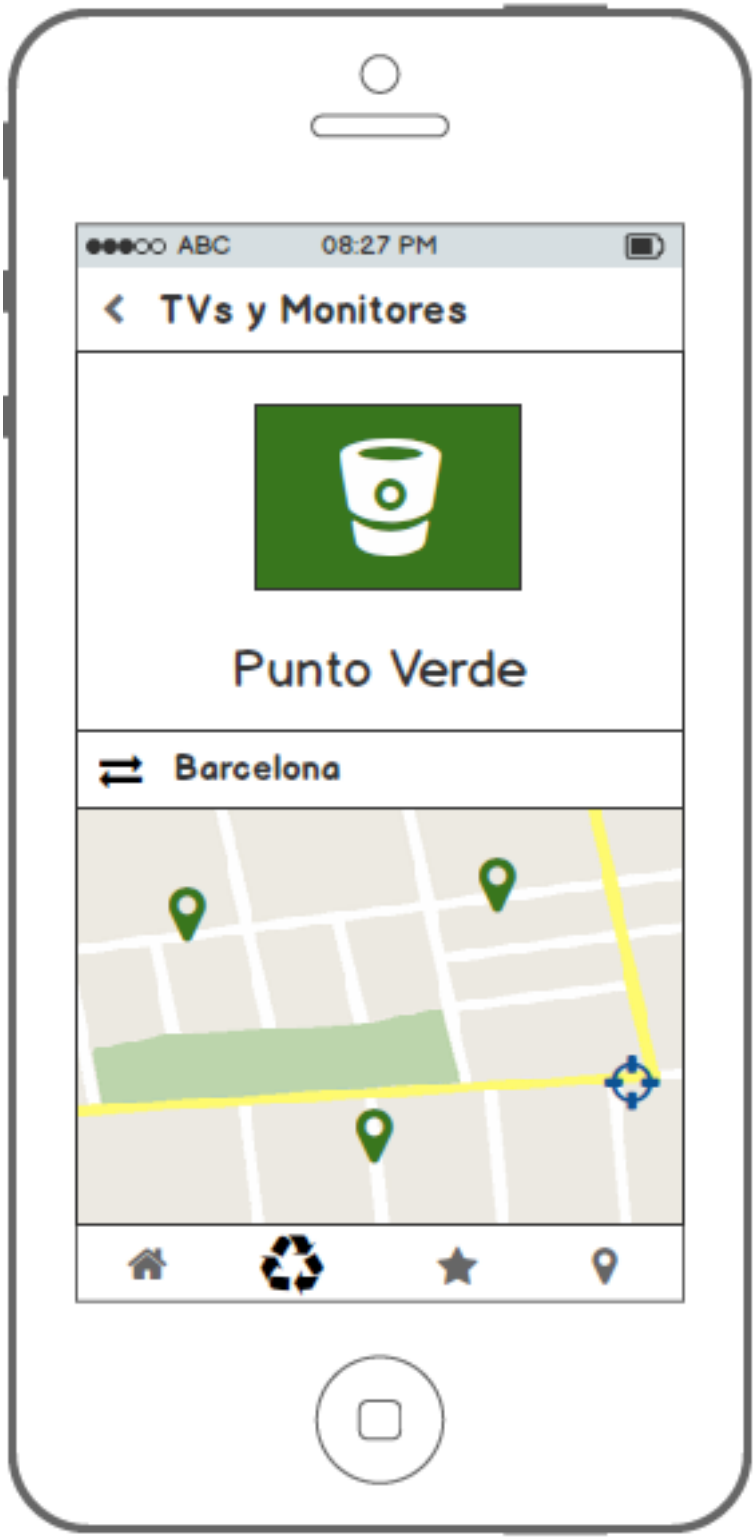

Ilustración 16: Prototipado, Residuo especial y Mapa

4.10. Pantalla de Residuo y mapa de local (detalles)

Si el usuario hace tap en cualquiera de los puntos del mapa, aparece un popover con la información detallada del local: Dirección, horario de apertura y, si procede, puntos que se pueden conseguir reciclando el residuo en cuestión.

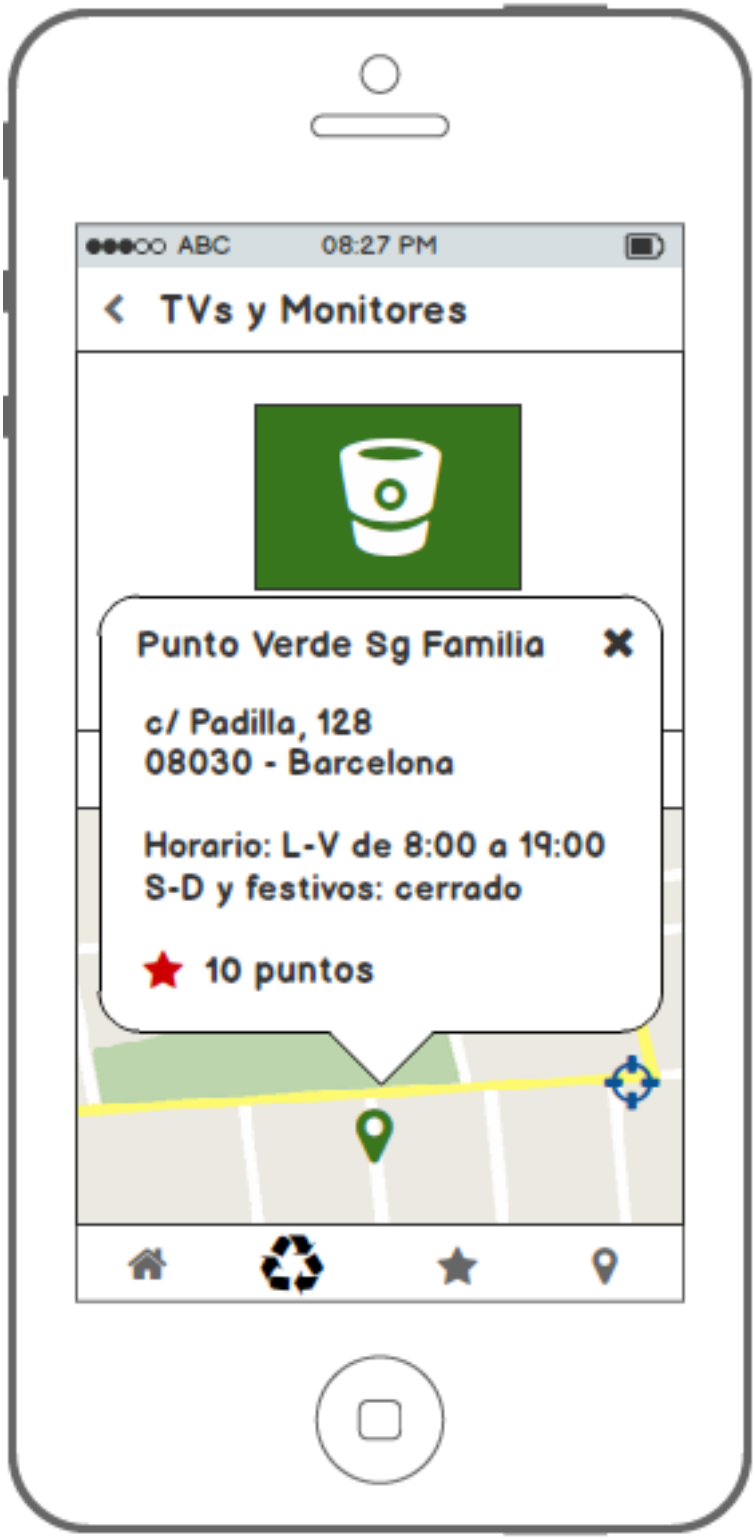

Ilustración 17: Prototipado, Residuo especial y Detalles

4.11. Pantalla de Ganar puntos

Una vez que el usuario esté en uno de los locales especiales de reciclaje, el operario le mostrará un código QR correspondiente al residuo o residuos reciclados. Desde la APP tendremos una pantalla para leer este código.

Ilustración 18: Prototipado, Ganar Puntos

4.12. Pantalla de Mis puntos / Historial

Listado con todos los movimientos que el usuario ha realizado respecto al saldo de puntos acumulados. Se muestran tanto los positivos como los negativos, ordenados por fecha descendente.

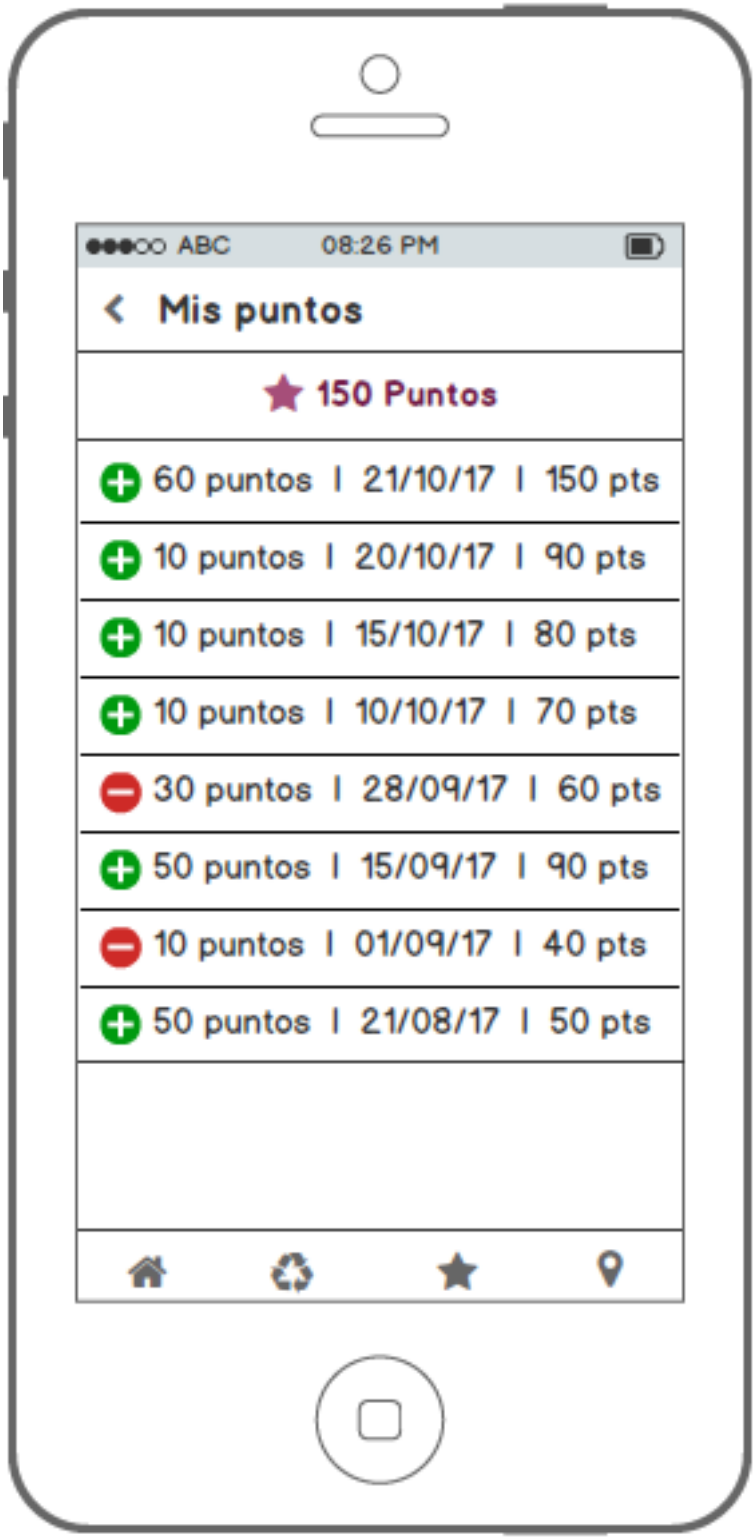

Ilustración 19: Prototipado, Mis Puntos

4.13. Pantalla de Los que más reciclan / Estadísticas

Listado con los usuarios activos que han acumulado más puntos mediante el uso de nuestra APP. Se destacan los 3 primeros y se muestra el Top10. También se resalta la posición del usuario actual.

Ilustración 20: Prototipado, Los que más reciclan

5. Evaluación

Para mejorar la usabilidad del prototipo, se han entregado las pantallas anteriores a un grupo de 4 personas, y se les ha propuesto que intenten seguir los casos de uso anteriores y a continuación han rellenado un pequeño test.

Pregunta: ¿Le ha sido fácil completar cada caso de uso?
Respuesta: Muy fácil | Fácil | Normal | Difícil | Muy difícil

Pregunta: ¿Cree que el tiempo necesario para cada caso es suficiente?
Respuesta: Resolución rápida | Resolución media | Resolución lenta

Pregunta: ¿La aplicación cumple con su objetivo inicial?
Respuesta: Sí, bastante | Sí | No | Absolutamente nada

Pregunta: ¿Echa algún elemento en falta?
Respuesta: Texto libre

Con las respuestas recogidas, se plantean posibles cambios para aplicar en el momento del desarrollo.

Otras pruebas

En el momento de la finalización del desarrollo, se realizaran las mismas preguntas a un grupo más amplio, pero esta vez la evaluación se realizará sobre el producto final (versión beta) y no sobre los prototipos.

Además de esto, se aplicarán tests de rendimiento: cargas del servidor, control de consumo de batería, velocidad de descarga de datos, legibilidad en el uso de tamaño y colores de textos y comparaciones de uso con la competencia.

6. Diseño Técnico

6.1. Definición de los casos de uso

Caso de Uso 1	Buscar un residuo para reciclar (sin local de reciclaje)
Actor / Actores	Cualquier usuario
Precondiciones	Ninguna
Flujo	1. Abrir la aplicación 1. Desde la pantalla de inicio, acceder al icono de “Reciclar” o “¿Qué vas a reciclar?” 2. Escribir el residuo que se desea buscar 3. Seleccionar el resultado coincidente de la lista 4. Leer la información mostrada (tipo de contenedor)
Postcondiciones	Se registra la acción de esta búsqueda como estadística para mostrar en la secci ón de "Lo más buscado"

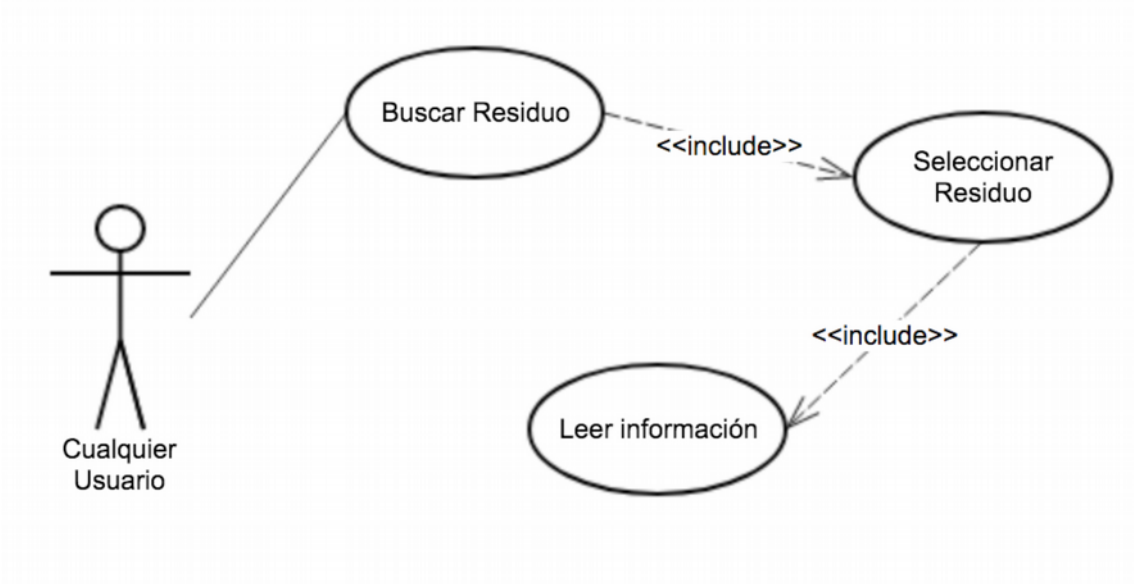

Ilustración 21: Caso de Uso 1

Caso de Uso 2	Buscar un residuo para reciclar (con local específico)
Actor / Actores	Cualquier usuario
Precondiciones	Ninguna
Flujo	2. Abrir la aplicación 3. Desde la pantalla de inicio, acceder al icono de “Reciclar” o “¿Qué vas a reciclar?” 4. Escribir el residuo que se desea buscar 5. Seleccionar el resultado coincidente de la lista 6. En el mapa resultante, hacer tap sobre el punto deseado (local de residuos) 7. Leer la información mostrada (dirección, horarios, etc)
Postcondiciones	Se registra la acción de esta búsqueda como estadística para mostrar en la secci ón de "Lo más buscado"

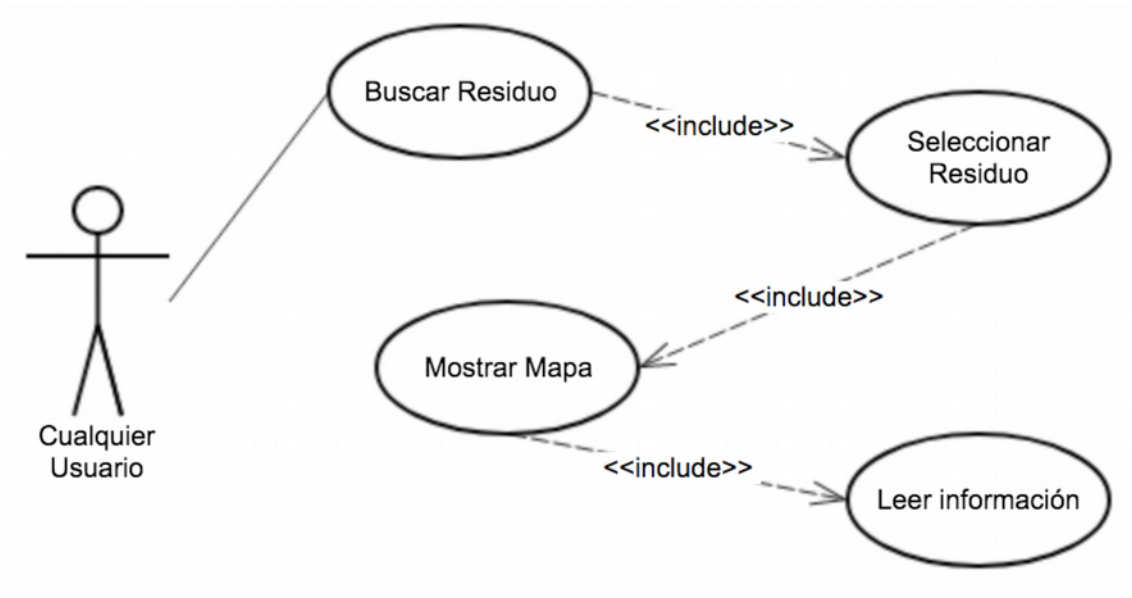

Ilustración 22: Caso de Uso 2

Caso de Uso 3	Conseguir puntos
Actor / Actores	Usuario registrado
Precondiciones	Haber iniciado sesión con su cuenta de usuario
Flujo	1. Abrir la aplicación 2. Desde la pantalla de inicio, acceder al icono de “Ganar Puntos” 3. Escanear el código QR con la cámara
Postcondiciones	Se almacenan los puntos obtenidos en la cuenta del usuario y se le avisa al usuario del resumen de sus movimientos

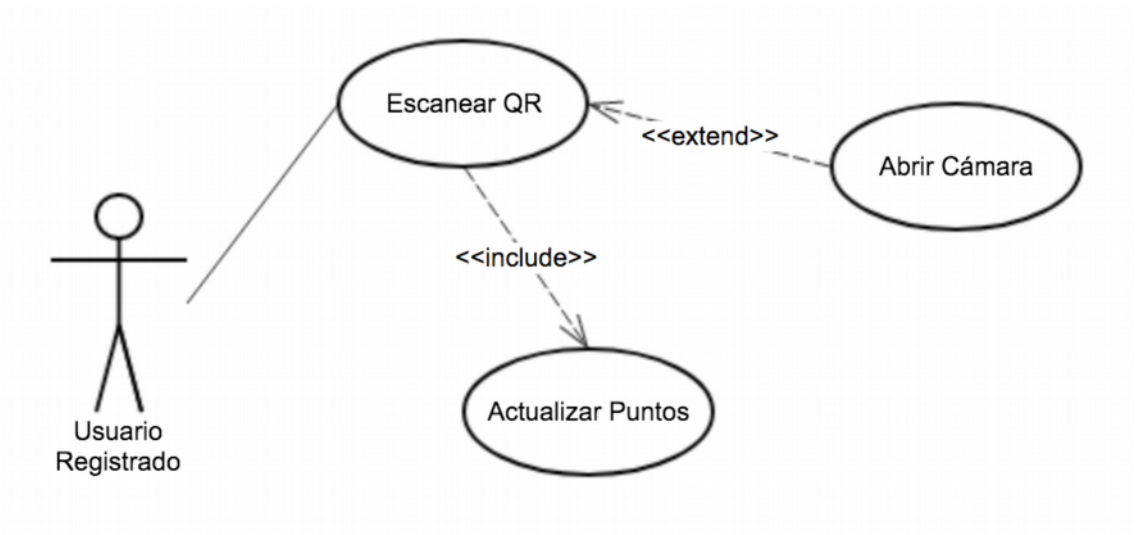

Ilustración 23: Caso de Uso 3

Caso de Uso 4	Registro de usuario
Actor / Actores	Usuario no registrado
Precondiciones	No tener una cuenta en uso y sin sesión iniciada
Flujo	1. Abrir la aplicación 2. Desde la pantalla de inicio, acceder al icono de “Usuario” 3. Desde la pantalla de login, acceder al formulario de registro 4. Rellenar los datos obligatorios del formulario (nombre, email, password y aceptar las condiciones de uso) 5. Hacer tap en el botón de “Crear cuenta”
Postcondiciones	Se comprueba que no exista la cuenta creada. En caso satisfactorio, se inicia la sesión en el dispositivo. En caso contratrio se notifica al usuario del error.

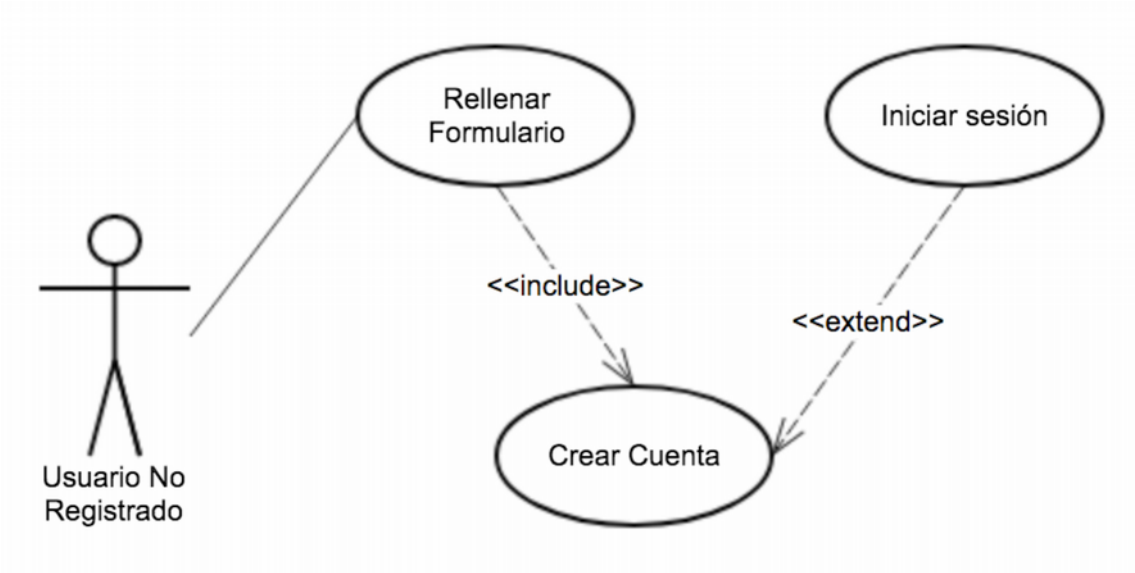

Ilustración 24: Caso de Uso 4

Caso de Uso 5	Login
Actor / Actores	Usuario registrado
Precondiciones	Tener una cuenta ya creada y tener la sesión finalizada
Flujo	1. Abrir la aplicación 2. Desde la pantalla de inicio, acceder al icono de “Usuario” 3. Rellenar los datos obligatorios del formulario (email, password) 4. Hacer tap en el botón de “Entrar cuenta”
Postcondiciones	Si la combinación de email y contraseña existe, la sesión se inicia. En caso contrario se notifica al usuario del error.

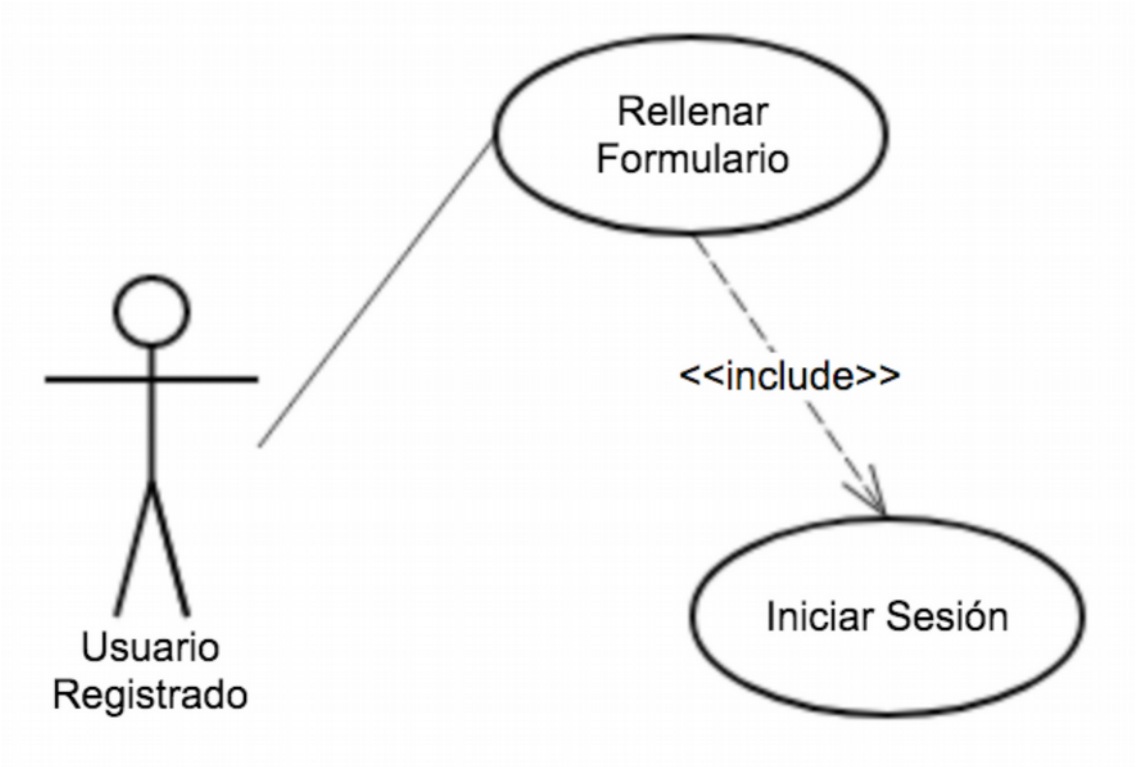

Ilustración 25: Caso de Uso 5

Caso de Uso 6	Cerrar sesión
Actor / Actores	Usuario registrado
Precondiciones	Tener la sesión iniciada
Flujo	1. Abrir la aplicación 2. Desde la pantalla de inicio, acceder al icono de “Usuario” 3. Hacer tap en la opción de “Cerrar sesión”
Postcondiciones	Se finaliza la sesión el dispositivo y se deriva al usuario a la página de Splash inicial

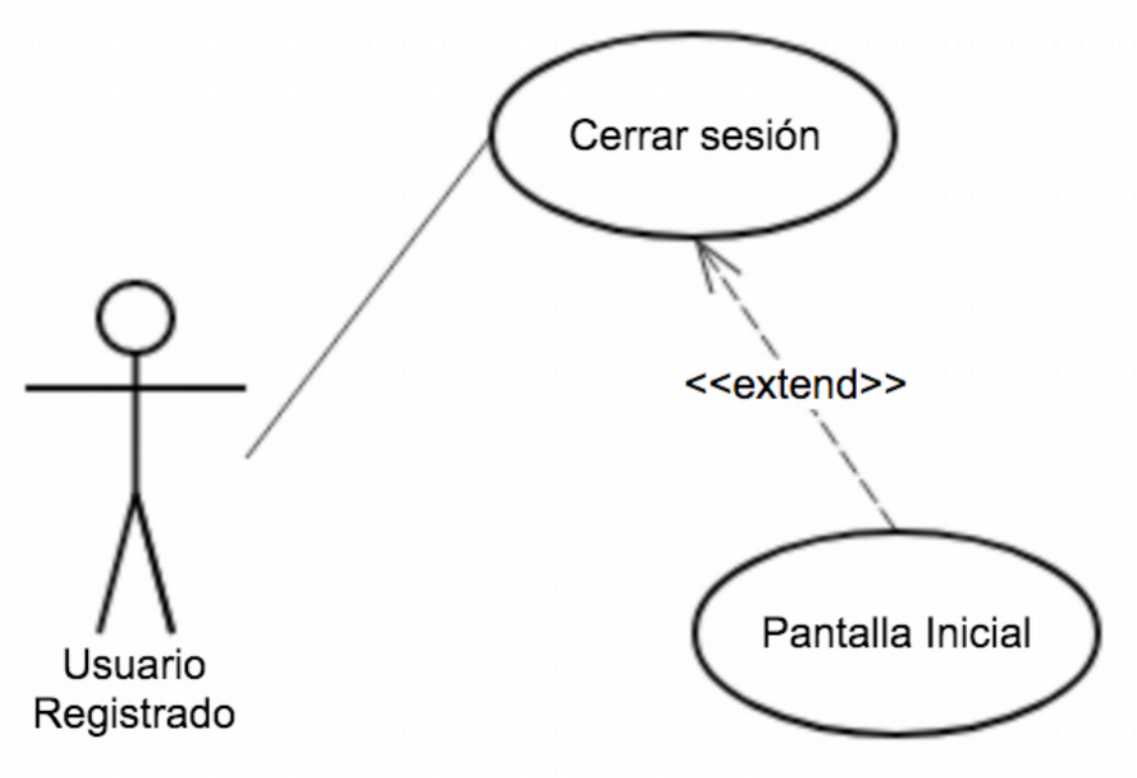

Ilustración 26: Caso de Uso 6

Caso de Uso 7	Modificar datos del perfil
Actor / Actores	Usuario registrado
Precondiciones	Tener la sesión iniciada
Flujo	1. Abrir la aplicación 2. Desde la pantalla de inicio, acceder al icono de “Usuario” 3. Hacer tap en la opción de “Editar perfil” 4. Modificar los campos que se deseen (foto, nombre, apellidos) 5. Hacer tap en el botón de “Actualizar”
Postcondiciones	Los datos se actualizan según los cambios realizados

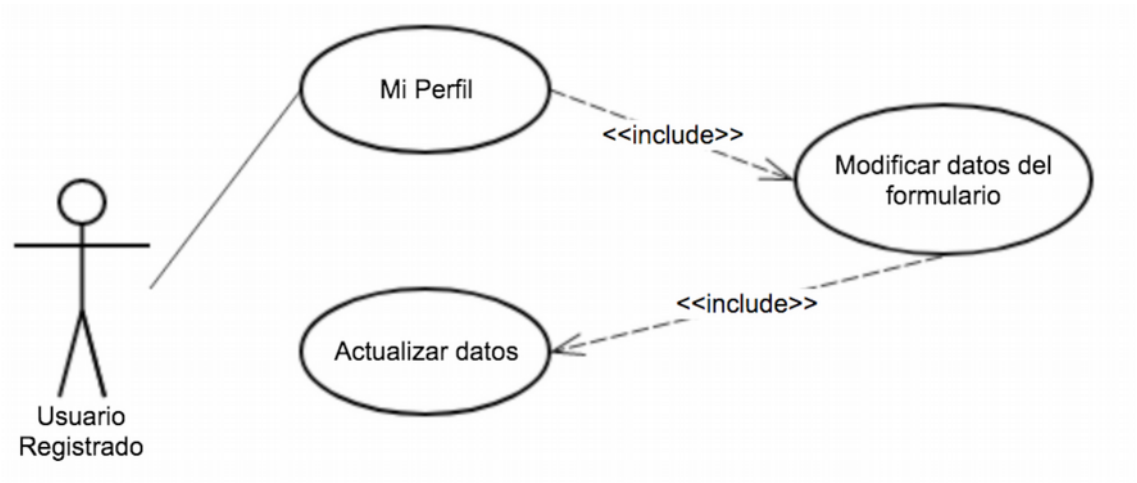

Ilustración 27: Caso de Uso 7

Caso de Uso 8	Cambiar de ciudad
Actor / Actores	Cualquier usuario
Precondiciones	Ninguna
Flujo	1. Abrir la aplicación 2. Desde la pantalla de inicio, acceder al icono de “Cambiar Ciudad” 3. Seleccionar de la lista la ciudad deseada
Postcondiciones	Se almacena esta preferencia en el dispositivo y en la cuenta del usuario en caso de tener la sesión iniciada

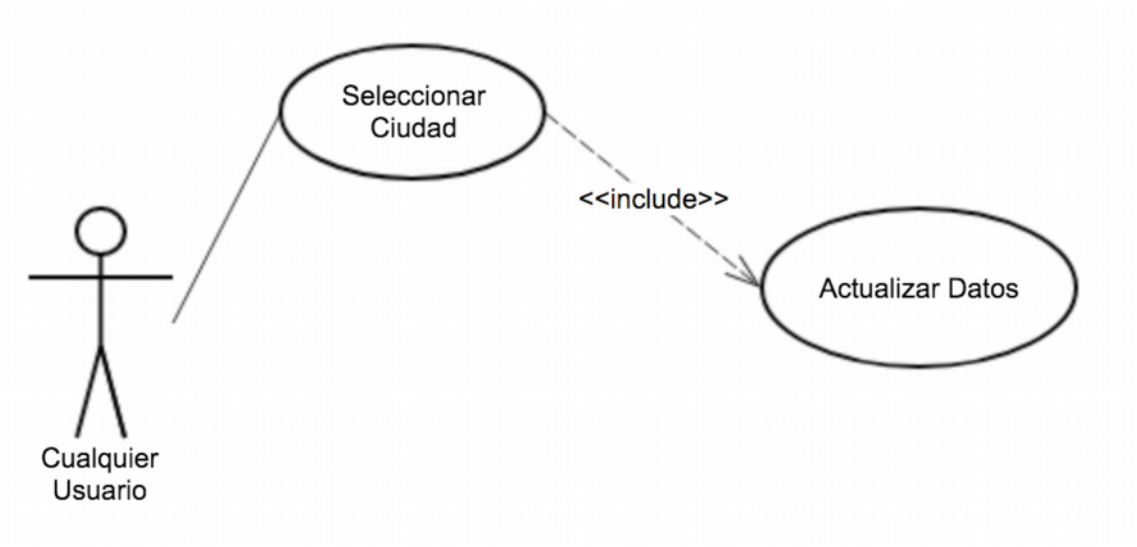

Ilustración 28: Caso de Uso 8

Caso de Uso 9	Conseguir regalos
Actor / Actores	Usuario registrado
Precondiciones	Tener la sesión iniciada y tener puntos acumulados
Flujo	1. Abrir la aplicación 2. Desde la pantalla de inicio, acceder al icono de “Usuario” 3. Seleccionar de la lista la opción de “Conseguir puntos” 4. Escoger el regalo deseado de la lista ofrecida
Postcondiciones	Se abrirá una pantalla externa, perteneciente a civiclub.com en dónde el usuario podrá escoger el regalo deseado según los puntos que posee. En caso de confirmar la adquisici ón, se actualizará el saldo de puntos del usuario.

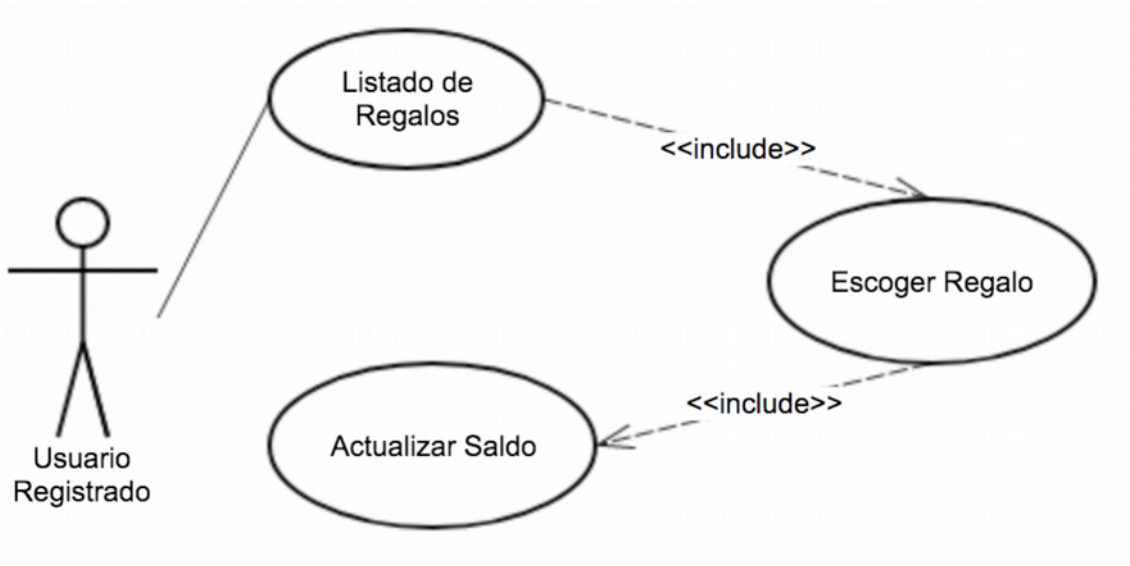

Ilustración 29: Caso de Uso 9

6.2. Diseño de la arquitectura

Para la implementación de la aplicación utilizaremos el framework de desarrollo para aplicaciones móviles hibridas **IONIC 2**. La mayor ventaja de este framework es que permite construir aplicaciones tanto para IOS, Android y Windows Phone con el mismo código de desarrollo.

Ionic 2 está basado en el framework **Angular 2 para la programación** y **Apache Cordova para la compilación del código**. El package obtenido estará listo para publicar en las diferentes tiendas de aplicaciones para cada plataforma.

6.2.1. Diagrama Estructura del sistema

Ionic, y por consiguiente Angular, se basan en el patrón Vista-Controlador. Este patrón permite crear vistas por cada sección de interfaz (y subvistas). Estas vistas están asociadas a un controlador, llamado Componen, que es el que se encarga de conectarse con los datos y definir la lógica de negocio para cada caso.

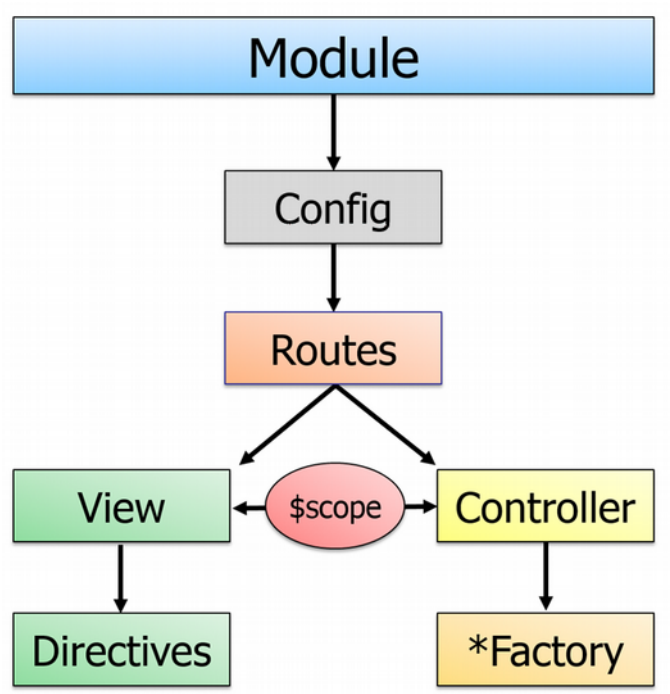

Ilustración 30: Estructura Ionic+Angular, https://ajgallego.gitbooks.io/

Por tanto, vemos que el Modelo de datos, queda fuera de este esquema. La base de datos queda almacenada en un webservice propio que servirá datos en formato JSON y recibirá acciones desde los Componentes.

6.2.2. Diagrama UML del diseño de la base de datos

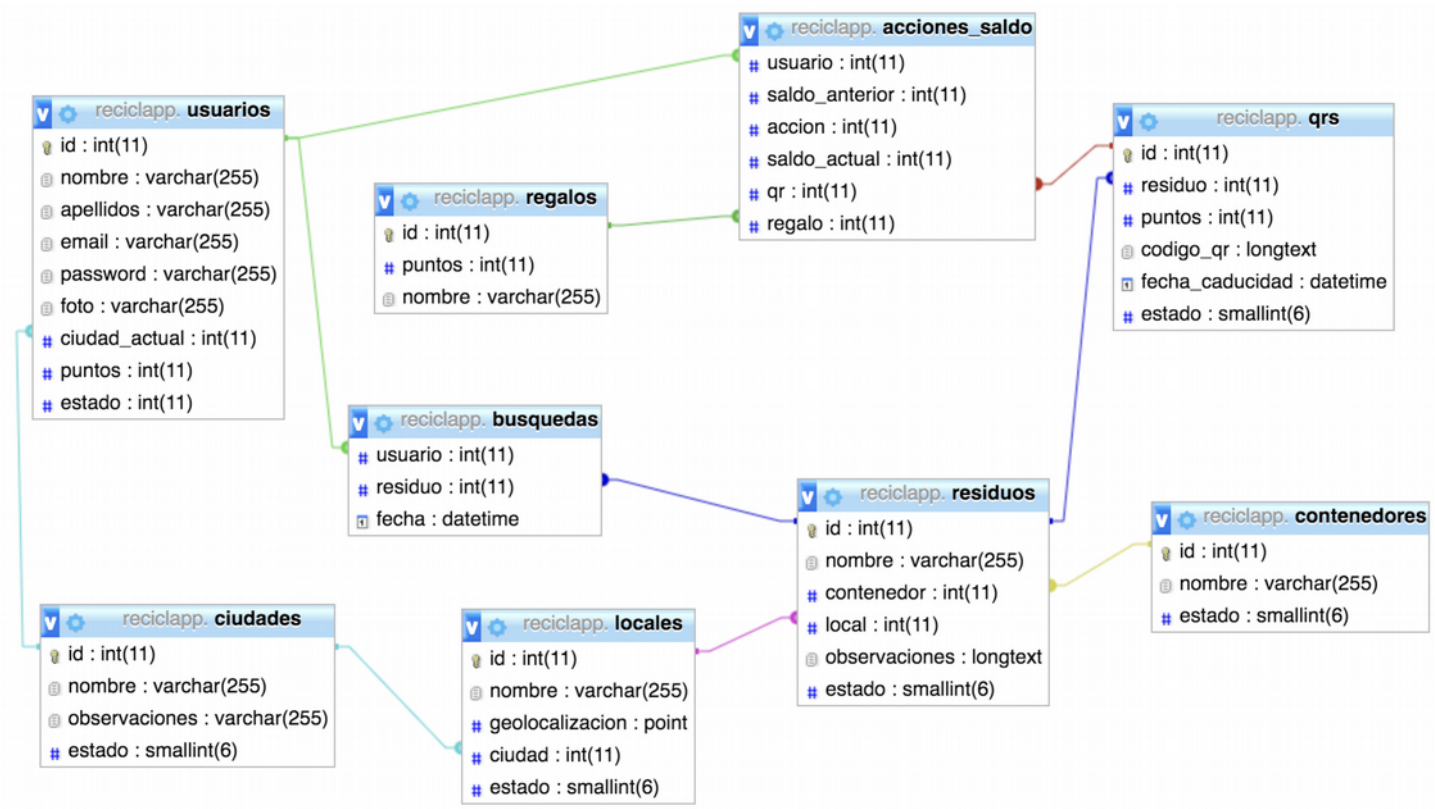

Ilustración 31: UML Diseño BDD

6.2.3. Llamadas al API (webservice)

doLogin()
Comprueba credenciales y lleva a cabo el login

checkUser()
Comprueba que el usuario no exista antes de registrarlo

registerUser()
Inserta un nuevo usuario a partir de la información recibida

setUser()
Actualiza los datos del usuario a partir de la información recibida

changeCity()
Cambia la ciudad actual del usuario

searchByTerm()
Realiza una búsqueda sobre todos los residuos según el término introducido

showResidue()
Muestra la información del residuo seleccionado

showMap()
Muestra el mapa con los locales del tipo seleccionado

selectLocalMap()
Muestra la información adicional de un punto en concreto

readQR()
Comprueba la URL del QR y acumula puntos en la cuenta del usuario

showQR()
Muestra el código QR a partir de la combinación de residuo y puntos

exchangePoints()
Modifica el saldo de un usuario

7. Desarrollo

7.1. Estructura

La estructura seguida es la típica que se utiliza utilizando Ionic 3 y Angular 2.

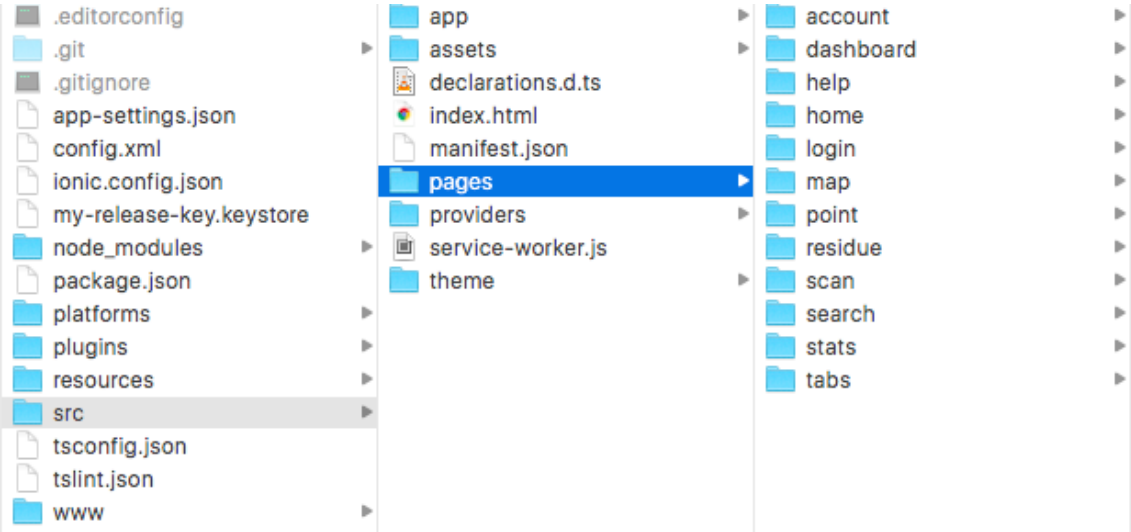

Ilustración 32: Árbol de estructura Ionic compilada

Las diferentes pantallas de nuestra APP se agrupan en carpetas formadas por 1 fichero .html, 1 fichero .scss y 1 fichero .ts. De esta manera tenemos un componente por cada pantalla.

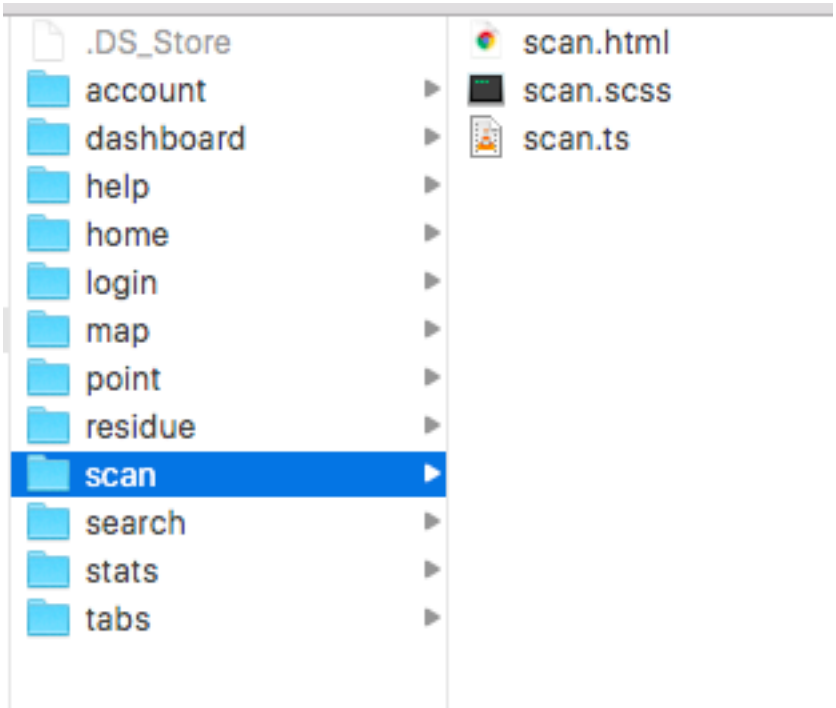

Ilustración 33: Ejemplo de estructura de Componente Angular

Para las funcionalidades comunes, el uso de constantes, llamadas a APIs … utilizamos 2 providers y 1 fichero de configuración:

- ApiService -> provider que se encarga de todas las llamadas al API
- UserService -> provider que se encarga de utilizar la información del usuario en la sesión
- Config -> fichero con la definición de constantes y variables globales del proyecto, algunas de ellas por ejemplo, claves de almacenamiento en el Storage

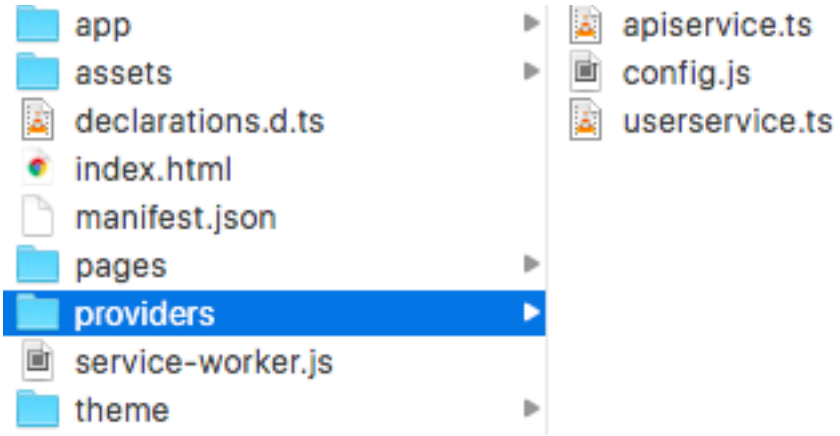

Ilustración 34: Providers y fichero de configuración

7.2. Plugins nativos Cordova

Para poder escanear un código QR se ha utilizado la librería QRScanner

Para poder obtener la posición del dispositivo se ha utilizado la librería GeoLocation

A partir de la versión 6.0 de Android, las aplicación piden autorización a permisos en tiempo de ejecución y no en la instalación, por lo que el usuario será informado antes de utilizar los 2 plugins anteriores.

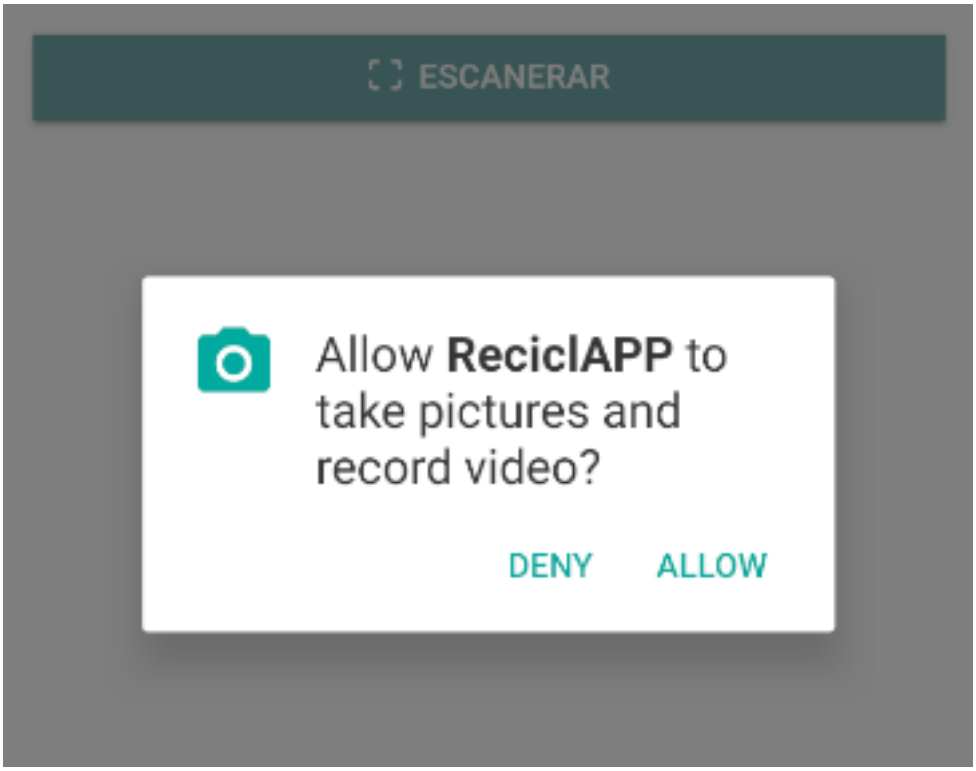

Ilustración 35: Alerta de confirmación de permisos, ej: hacer fotos y videos

7.3. Otras herramientas de desarrollo y consideraciones

Para el almacenamiento interno se ha utilizado Storage. De esta manera la sesión iniciada de un usuario se mantiene persistente en la APP

Se han modificado los estilos por defecto (theme) para modificar el aspecto de los botones, tabs, enlaces, colores primarios y secundarios, etc

Librería de terceros

Se ha utilizado la librería de terceros GoogleMaps para mostrar el mapa con los distintos locales para la recogida de residuos especiales.

Repositorio utilizado

El código fuente completo está accesible de manera pública (temporalmente) en un repositorio GIT de GitLab:

https://gitlab.com/czaragozal/reciclapp/

** En el Anexo final, aparecen varios ejemplos de códigos QR para su escaneo y testeo*

*** Si se desean realizar pruebas con usuarios ya existentes en el sistema, a continuación muestro los credenciales de un par de ellos:*

- Usuario 1
  Email: test@uoc.edu
  Pass: 123123

- Usuario 2
  Email: john@doe.com
  Pass: 123123

7.4. Pantallas más importantes

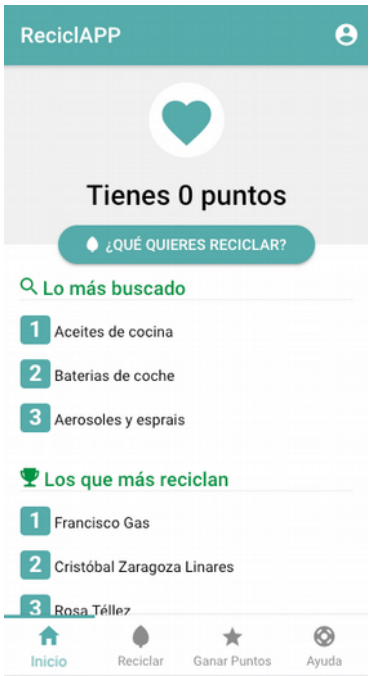

Ilustración 36: P. Dashboard

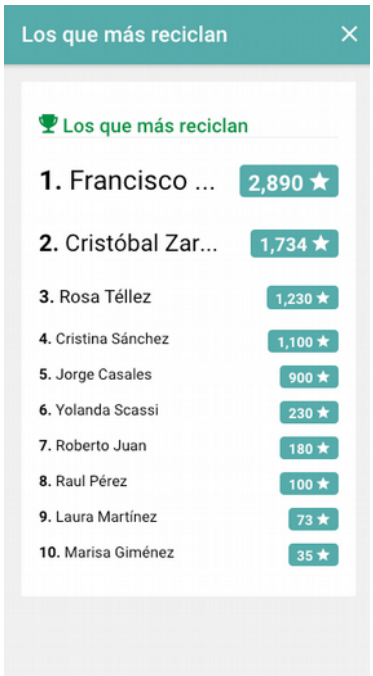

Ilustración 37: P. Top Stats

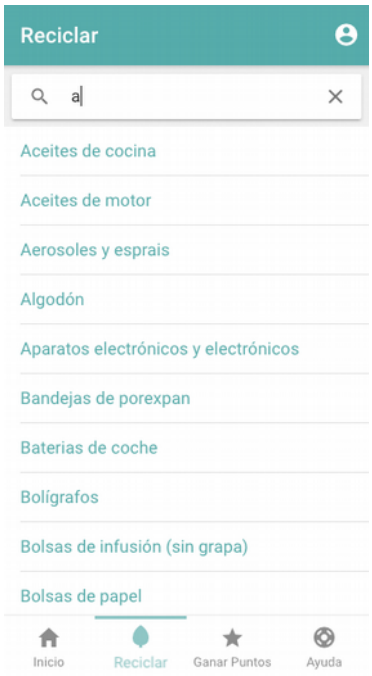

Ilustración 38: P. Búsqueda

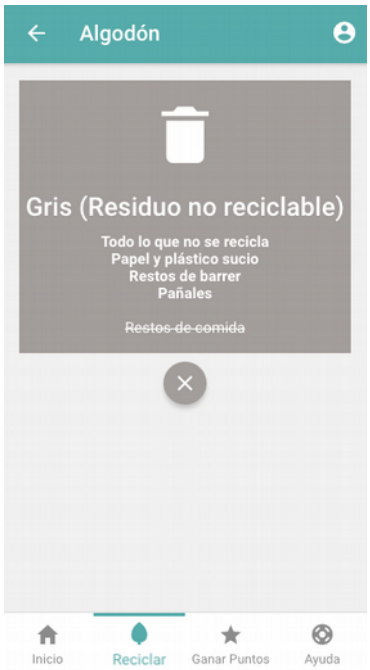

Ilustración 39: P. Resultado

Ilustración 40: P. Mapa

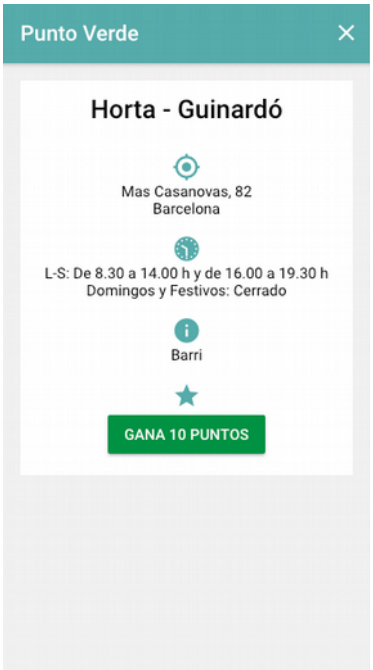

Ilustración 41: P. Local

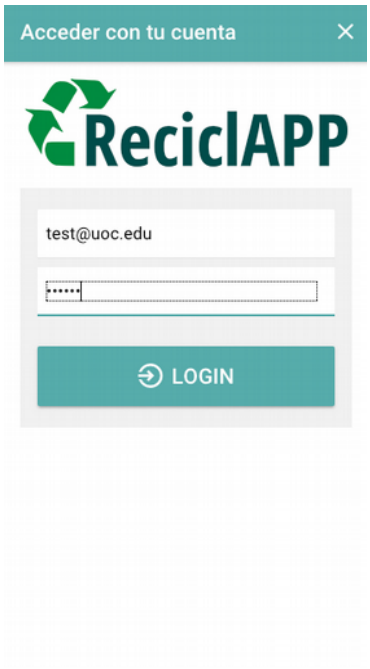

Ilustración 42: P. Login

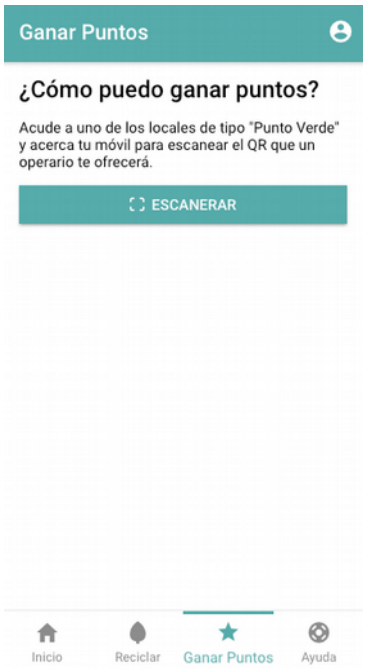

Ilustración 43: P. Scan QR

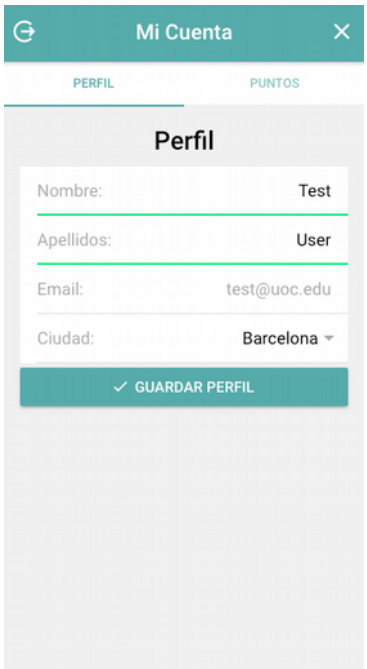

Ilustración 44: P. Mi Perfil

Ilustración 45: P. Puntos

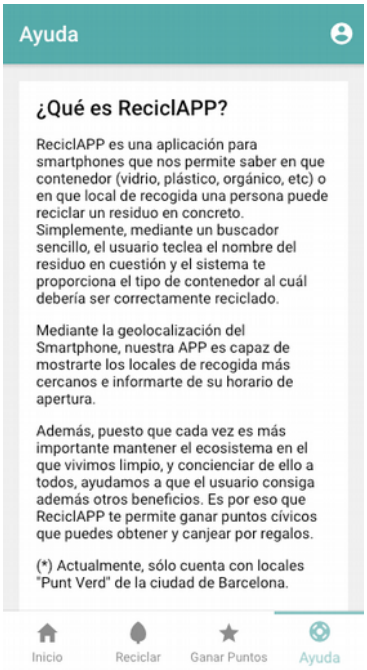

Ilustración 46: P. Ayuda

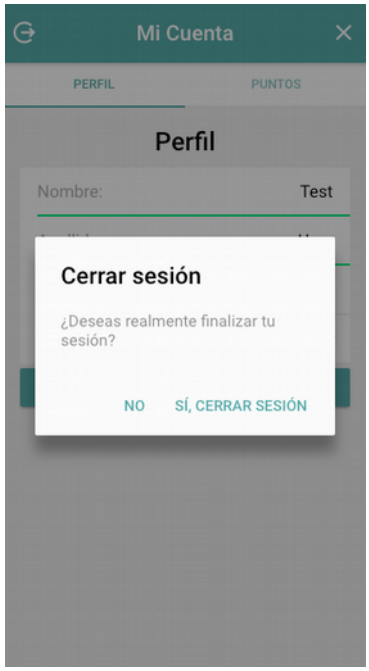

Ilustración 47: P. Logout

7.5. Funcionalidades incompletas

Durante el desarrollo han existido varios imprevistos y desviaciones de horas de desarrollo que han hecho que una funcionalidad esté incompleta, en concreto la de Registro de Usuario.

Para la PEC3, la funcionalidad de Registro de Usuario (y recuperación de contraseña), quedó pendiente y finalmente se completó en la PEC4.

Futuro

1. Generar el fichero binario para iOS

2. Proponer el registro de usuario mediante Deeplinking o Inserción de código de 6 dígitos enviado vía SMS

3. Añadir links para compartir las acciones a las redes sociales más populares

4. Mejorar el rendimiento de la carga de datos: caché de los JSON, mapa …

5. Gestión de LOGS desde el CMS

7.6. Decisiones tomadas durante el desarrollo

Cambio en el planteamiento

El planteamiento inicial proponía:

1. Desarrollar el API como primera fase y alimentar la Base de datos
2. Continuar con la implementación del Frontend sin funcionalidades (sólo el diseño)
3. Terminar con la lógica interna de toda la APP.

Por miedo a posibles incompatibilidades con las funcionalidades nativas de Ubicación y Lectura de QR, modifiqué el flujo existente por este otro:

1. Configurar la APP con dos pantallas que implementaran las funcionalidades de lectura de un código QR y otra para mostrar un mapa con diferentes puntos y con la localización actual del dispositivo.
2. En segundo lugar se modificaron los estilos para aplicar en todas las pantallas.
3. Insertar datos de prueba en la APP para poder mostrar
4. Posteriormente se crearon las funciones de tipo GET (devolución / select de datos) del API
5. Creación de pantalla a pantalla revisando el API si fuera necesario (funcionalidades POST que modifican datos)
6. Insertar datos reales en gran cantidad en la base de datos

Así pues, dejé para el final el registro del usuario, ya que todas las funcionalidades restantes podían “convivir” sin este formulario, pero teniendo que dar por supuesto que el usuario se habría registrado por otra vía.

Backend Completo

Invertí más horas de las necesarias en tener un producto bastante completo y similar a otros backends de terceros como Backendless o Firebase.De esta forma, la inserción y modificación de datos resultó ser más rápida y eficaz y con una mejor síntesis a la hora de mostrar los datos actuales, unidos en un CMS responsivo de acceso mediante cualquier navegador.

Generación del APK

Otros de los puntos en los cuales invertí más horas de las esperadas fue en la creación del código en binario (apk). Inicialmente pensaba que simplemente bastaría con poder visualizar la APP mediante el visor IonicView y tras una consulta con mi profesor realicé los pasos para poder generar el fichero firmado para su correcta instalación.

7.7. Backend y API

Para la conexión con la base de datos se ha utilizado un Backend desarrollado por la empresa en la que trabajo: Dommia Design Studio (https://www.dommia.com)

La base de datos esta configurada bajo un motor MySQL y la programación del Backend utiliza el lenguaje PHP.

Acceso privado vía web: https://cristobal.dommia.es/gestor

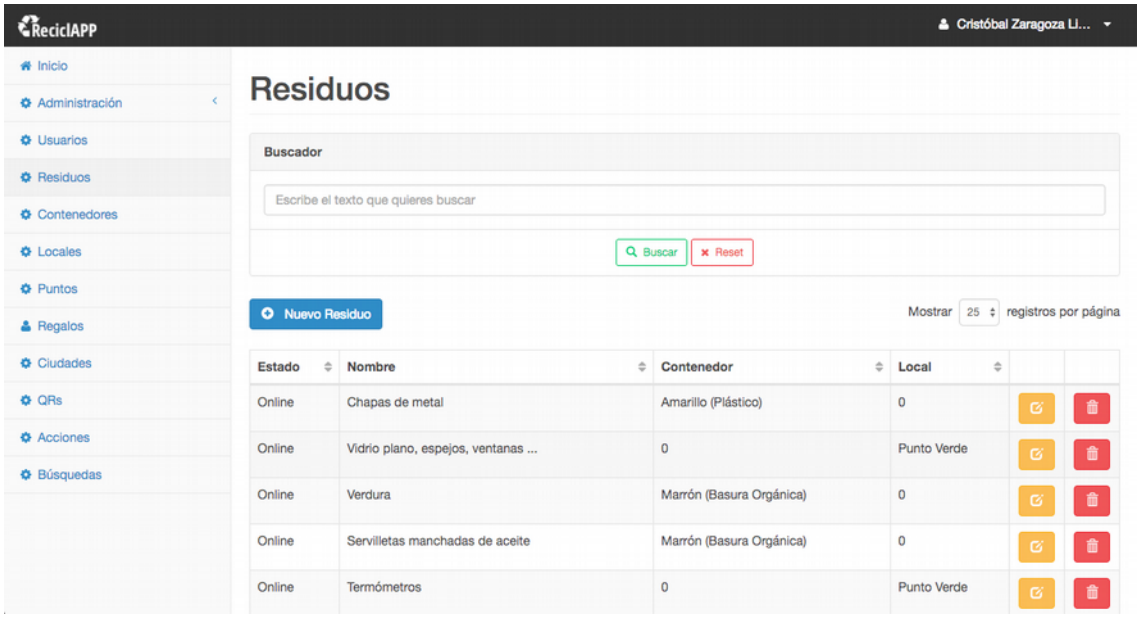

Ilustración 48: Backend - Listado de Residuos

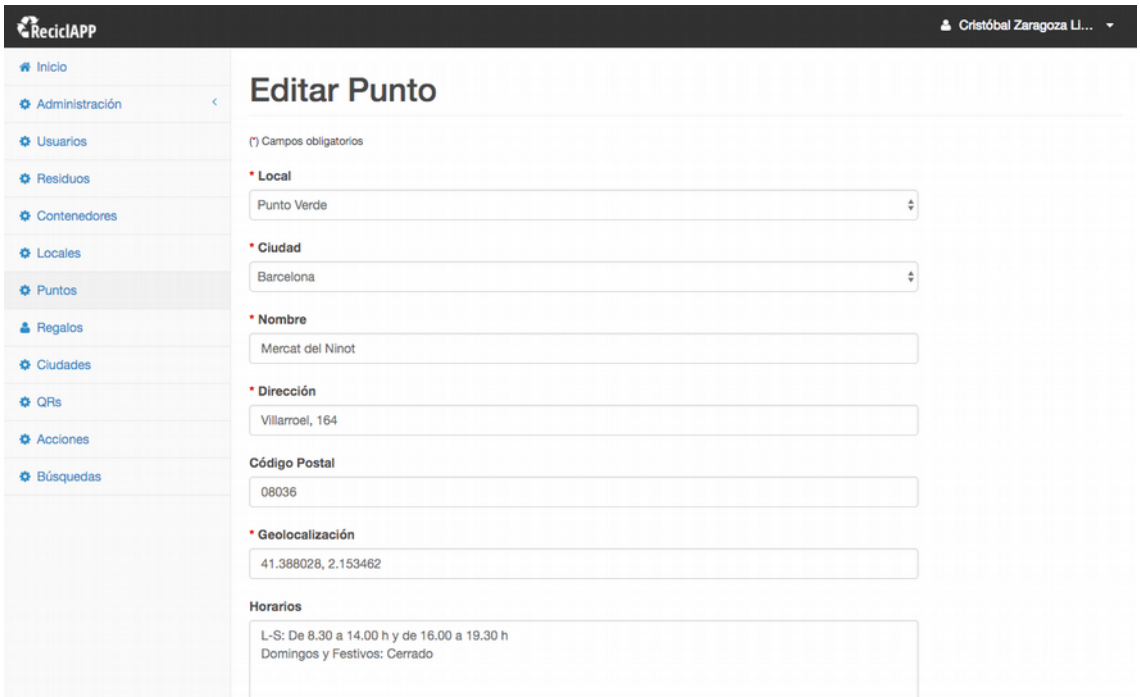

Ilustración 49: Backend - Form Edición de Punto

Para el API, se ha optado una comunicación GET para la devolución de ítems y una comunicación POST para las acciones de modificación, creación y eliminación.

Este recurso se puede encontrar bajo la siguiente dirección y mediante el protocolo seguro SSL: https://cristobal.dommia.es/api/rest-api.php/

Todos estos métodos se encuentran en la clase ApiService: provider dentro de nuestra APP y accesible en todo el entorno.

Las llamadas al API siempre se realizan de manera asíncrona. El usuario es informado de ello mediante un “loader” en forma de “spinner”, de tal manera que la navegación queda bloqueada hasta la respuesta del servidor. La mayoría de estas llamadas (las que son GET) se realizan en la primera ejecución, de manera que el acceso a la pantalla inicial puede ser un poco más lento pero esto beneficia a la velocidad en el resto de pantallas.

La respuesta es rápida y siempre se retorna un fichero JSON con 3 valores:

- status: 1 en caso de éxito, 0 en caso de error
- data: array con los resultados obtenidos en caso de éxito
- error: código de error si procede

Ejemplo de llamada GET que devuelve todos los tipos de contenedores:

https://cristobal.dommia.es/api/rest-api.php/contenedores

{"status":1,"data":[{"Id":"1","nombre":"Amarillo (Pl\u00elstico)","descripcion":"Envases de pl\u00elstico<br \/>Latas<br \/>Brics","prohibido":"Juguetes<br \/>CD y DVD","color":"yellow","estado":"1"},{"Id":"2","nombre":"Verde (Vidrio)","descripcion":"Botellas de vidrio<br \/>Botes de vidrio","prohibido":"Chapas y tapones de metal<br \/>Bombillas y fluorescentes<br \/>Restos de cer\u00elmica","color":"green","estado":"1"},{"Id":"3","nombre":"Azul (Papel)","descripcion":"Papel<br \/>Revistas<br \/>Peri\u00f3dicos<br \/>Cart\u00f3n plegado","prohibido":"Brics<br \/>Papel de aluminio","color":"blue","estado":"1"},{"Id":"4","nombre":"Marr\u00f3n (Basura Org\u00elnica)","descripcion":"Restos de comida<br \/>Tap\u00f3n de corcho<br \/>Restos de plantas y hierbas","prohibido":"Pa\u00flales<br \/>Restos de madera","color":"brown","estado":"1"},{"Id":"5","nombre":"Gris (Residuo no reciclable)","descripcion":"Todo lo que no se recicla<br \/>Papel y pl\u00elstico sucio<br \/>Restos de barrer<br \/>Pa\u00flales","prohibido":"Restos de comida","color":"grey","estado":"1"}]}

Ilustración 50: API JSON Contenedores

7.8. Revisión de la planificación

Durante el desarrollo, se han producido varios imprevistos y es por ello que la PEC3 se ha entregado con 4 días de desviación.

Para garantizar una completa entrega de la PEC4, se han modificado los siguientes procesos respecto al planteamiento original.

Completar la funcionalidad de Registro + Recuperar Contraseña
Entre el 18 y el 22 de Diciembre y entre el 27 y el 28 de Diciembre
14 horas

Finalizar la Memoria del Trabajo
El 29 de Diciembre
2 horas

Presentación: diseño, guión y estructura
El 30 de Diciembre
8 horas

Grabación del video y entrega final
El 31 de Diciembre
7 horas

7.9. Pruebas

7.9.1. Producto obtenido

El fichero binario es un apk firmado para la instalación en dispositivos Android, compatible con una versión m
ínima del API 16.

El fichero ocupa unos 6,10 MB de tamaño y necesita unos 13,20 MB de espacio disponible para su correcta instalación.

7.9.2. Pruebas unitarias

En los diversos dispositivos testados, se han realizado las siguientes pruebas.

Lote 1
Abrir la aplicación
Acceder sin cuenta de usuario
Buscar un residuo sin tratamiento especial, pe: “bolígrafos”
Abrir la pantalla de ayuda

Lote 2
Abrir la aplicación
Acceder sin cuenta de usuario
Buscar un residuo con tratamiento especial, pe: “pilas”
Modificar la ciudad actual por Barcelona

Lote 3
Abrir la aplicación
Acceder mediante login (usuario: test@uoc.edu)
Abrir pantalla de “Lista con los usuarios que más reciclan”
Abrir pantalla de “Conseguir puntos”
Escanear QR
Comprobar puntos obtenidos en el dashboard
Acceder a Mi Cuenta
Modificar el apellido
Acceder a Puntos y comprobar últimos movimientos
Logout

Lote 4
Abrir la aplicación
Acceder sin cuenta de usuario
Buscar un residuo con tratamiento especial, pe: “pilas”
Acceder mediante login (usuario: test@uoc.edu)
Tap sobre el punto más cercano
Acceder a Mi Cuenta
Acceder a Puntos

Acceder a “Conseguir Regalos”

7.9.3. Instalación en varios dispositivos y testeo

Se ha instalado nuestra APP en diferentes ámbitos y se han realizado las pruebas unitarias anteriores con el siguiente resultado:

Dispositivos Móviles Android

Samsung Galaxy J5 (2016) – Android 6.0.1 – Instalación y pruebas con éxito (con conexión 4G)

Samsung Galaxy A3 (2016) – Android 7.0 – Instalación y pruebas con éxito (con conexión WiFi)

Pixel XL – Android 8.0 – Instalación y pruebas con éxito (con conexión 3G)

Tablet Nexus 10 – Android 5.1 – Instalación correcta. Ejecución sin éxito (con conexión 3G)

Visor IonicView (APP: https://ionicframework.com/products/view)

Instalación correcta. Ejecución correcta excepto en el uso de plugins nativos tales como el uso de la cámara para poder escanear un código QR.

Navegador (vía http)

Google Chrome 62.0.3202.94 - Ejecución correcta excepto en el uso de plugins nativos tales como el uso de la cámara para poder escanear un código QR.

8. Conclusiones

Después del trabajo realizado entre Octubre y Diciembre de 2017, pasamos a enumerar las conclusiones extraídas:

1. La planificación es muy importante. Se debe cumplir y sobretodo se debe controlar una posible desviación ante posibles imprevistos.

2. Tener un documento con los diseños y casos de uso definidos ayuda a la fluidez del desarrollo.

3. El trabajo lo ha realizado una única persona. Como ventaja: conocimiento total sobre el producto, desde el diseño hasta el nivel más bajo de implementación. Desventaja: visión no global del producto y única persona que puede tomas decisiones con la mayor responsabilidad posible.

4. Para una visión global y comprobación del éxito de nuestra APP, se necesita realizar un lanzamiento y que un número considerado de usuarios testeen previamente el producto en varios dispositivos.

5. A nivel técnico, es mejor realizar como primer punto la navegación entre pantallas y aplicar estilos para luego desarrollar pantalla a pantalla la lógica de negocio y compilar el resultado para su prueba en dispositivo.

6. Sin poder dedicar 8 horas diarias al desarrollo, 2 meses se quedan cortos para obtener un producto muy pulido (control de logs, velocidad de carga, control de errores…)

7. Al finalizar el desarrollo, el hecho de ver completado el trabajo y probar su correcto funcionamiento, confirman la satisfacción del programador ante “el producto que ha creado”.

9. Glosario

API:
Interfaz de programación de aplicaciones. Métodos utilizados por las bibliotecas de programación a modo de comunicación entre 2 softwares (webs, apps, etc) En nuestro caso el recurso https://cristobal.dommia.es/ sirve los diferentes métodos que desde nuestra APP se ejecutan (mostrar residuos, contenedores, login de usuario, etc)

Frontend:
Capa de programación dedicada a la presentación y separada de la capa de acceso a datos.

Backend:
Capa de programación dedicada al acceso a datos y separada de la capa de presentación.

Loader:
Controlador de Ionic que se utiliza para detener el flujo de navegación mientras se realiza una acción asíncrona en background. A la imagen que suele acompañar en forma de cuadro de diálogo se le llama “Spinner”

Provider:
Clase Javascript utilizada en Angular para encapsular una serie de llamadas a métodos en común (a modo de servicio) y con la finalidad de poder ser reutilizable en todos los componentes de la APP.

10. Bibliografía

Web: https://www.civiclub.org/
8 de Octubre de 2017

Web: http://www.ganasa.es/areas/calidad-ambiental/app_movil1.aspx
8 de Octubre de 2017

Web: http://ajuntament.barcelona.cat/ecologiaurbana/ca/serveis/la-ciutat-funciona/manteniment-de-l-espai-public/gestio-de-neteja-i-residus/cercador-de-residus
8 de Octubre de 2017

Web: https://ajgallego.gitbooks.io/ionic/content/arquitectura.html
21 de Octubre de 2017

Web: http://www.cogersa.es/metaspace/portal/14498/20195
11 de Noviembre de 2017

Web: https://ionicframework.com/docs/
11 de Noviembre de 2017

11. Anexos

11.1. Compilación del código fuente

11.1.1. Paso previo

Necesitamos previamente tener bajo un entorno UNIX, un servidor web Apache instalado y configurado.

11.1.2. Instalación de Node JS y NPM

Necesario para poder instalar las librerías de código abierto. Disponible en la web: https://nodejs.org/es/download/

11.1.3. Instalación de Ionic CLI

Para poder ejecutar nuestra aplicación en el navegador, instalaremos la última versión de Ionic tecleando desde una terminal: npm install -g ionic@latest

11.1.4. Desplegar el código

Copiar el código fuente de nuestra APP en una carpeta vacía dentro de htdocs

11.1.5. Actualizar módulos y plugins a partir del package.json

Desde una terminal ejecutamos: npm install

```
sh-3.2# cd /Applications/XAMPP/htdocs/proyectos-angular/copia/
sh-3.2# npm install
npm WARN deprecated babili@0.1.2: babili has been renamed to babel-minify. Please update to babel-minify
npm WARN deprecated babel-preset-babili@0.1.4: babili has been renamed to babel-minify. Please update to babel-preset-minify
npm WARN deprecated object-keys@0.2.0: Please update to the latest object-keys
npm WARN prefer global node-gyp@3.6.2 should be installed with -g

> fsevents@1.1.3 install /Applications/XAMPP/xamppfiles/htdocs/proyectos-angular/copia/node_modules/fsevents
> node install

[fsevents] Success: "/Applications/XAMPP/xamppfiles/htdocs/proyectos-angular/copia/node_modules/fsevents/lib/binding/Release/node-v48-da
rwin-x64/fse.node" already installed
Pass --update-binary to reinstall or --build-from-source to recompile

> node-sass@4.5.3 install /Applications/XAMPP/xamppfiles/htdocs/proyectos-angular/copia/node_modules/node-sass
> node scripts/install.js

Downloading binary from https://github.com/sass/node-sass/releases/download/v4.5.3/darwin-x64-48_binding.node
Download complete[▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇▇] 0 :
Binary saved to /Applications/XAMPP/xamppfiles/htdocs/proyectos-angular/copia/node_modules/node-sass/vendor/darwin-x64-48/binding.node

> node-sass@4.5.3 postinstall /Applications/XAMPP/xamppfiles/htdocs/proyectos-angular/copia/node_modules/node-sass
> node scripts/build.js

Binary found at /Applications/XAMPP/xamppfiles/htdocs/proyectos-angular/copia/node_modules/node-sass/vendor/darwin-x64-48/binding.node
Testing binary
Binary is fine
reciclapp@0.0.0 /Applications/XAMPP/xamppfiles/htdocs/proyectos-angular/copia
├── @angular/common@4.0.2
├── @angular/compiler@4.0.2
├── @angular/compiler-cli@4.0.2
│   ├── @angular/tsc-wrapped@4.0.2
│   │   └── tsickle@0.21.6
│   ├── minimist@1.2.0
│   └── reflect-metadata@0.1.10
├── @angular/core@4.0.2
├── @angular/forms@4.0.2
├── @angular/http@4.0.2
├── @angular/platform-browser@4.0.2
├── @angular/platform-browser-dynamic@4.0.2
├── @ionic-native/app-preferences@4.5.2
```

Ilustración 51: Uso de npm install desde terminal

11.1.6. Probar y ejecutar nuestra APP

Desde una terminal ejecutamos: ionic serve

```
sh-3.2# sudo ionic serve
Starting app-scripts server: --address 0.0.0.0 --port 8100 --livereload-port 35729 --dev-logger-port 53703 --nobrowser -
Ctrl+C to cancel
[18:33:50]  watch started ...
[18:33:50]  build dev started ...
[18:33:50]  clean started ...
[18:33:50]  clean finished in 2 ms
[18:33:50]  copy started ...
[18:33:50]  transpile started ...
[18:33:54]  transpile finished in 3.90 s
[18:33:54]  preprocess started ...
[18:33:54]  deeplinks started ...
[18:33:54]  deeplinks finished in 84 ms
[18:33:54]  preprocess finished in 85 ms
[18:33:54]  webpack started ...
[18:33:54]  copy finished in 4.25 s
[18:34:06]  webpack finished in 12.40 s
[18:34:06]  sass started ...
[18:34:09]  sass finished in 2.55 s
[18:34:09]  postprocess started ...
[18:34:09]  postprocess finished in 8 ms
[18:34:09]  lint started ...
[18:34:09]  build dev finished in 19.01 s
[18:34:09]  watch ready in 19.11 s
[18:34:09]  dev server running: http://localhost:8100/

[OK] Development server running!
     Local: http://localhost:8100
     External: http://192.168.1.33:8100
     DevApp: ReciclAPP@8100 on MacBook-Air-de-Cristobal.local
```

Ilustración 52: Uso de ionic serve desde terminal

11.1.7. Testeo desde navegador

Desde un navegador podemos ejecutar nuestra aplicación accediendo a la siguiente dirección URL http://localhost:8100

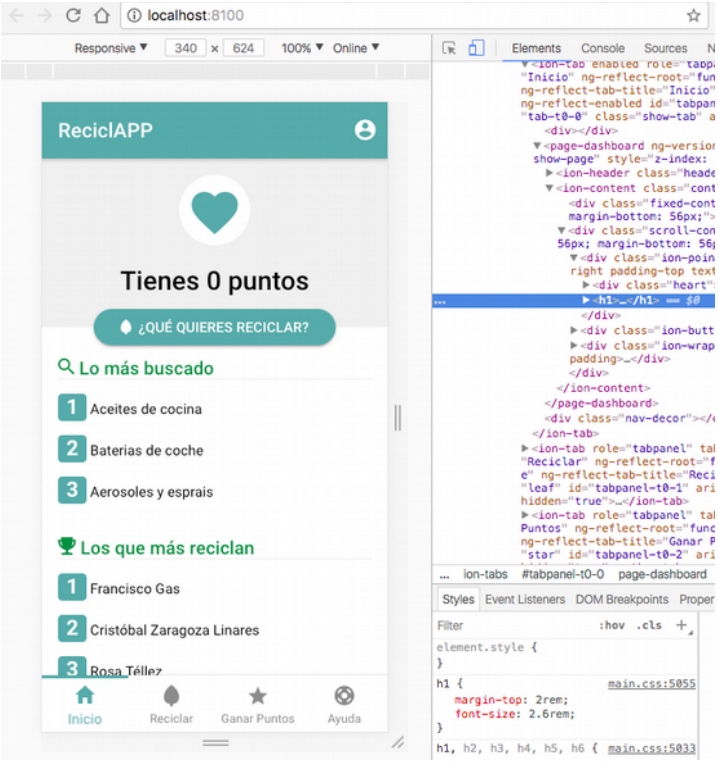

Ilustración 53: Aplicación desde navegador (Chrome)

11.1.8. Consideraciones

Es posible que se necesite ejecutar los anteriores comandos desde una terminal con una sesión iniciada previamente del usuario root mediante la sentencia “sudo”

11.1.9. Instalación en dispositivo Android

Simplemente ejecutar el fichero .apk entregado en el dispositivo y seguir los pasos que ofrece el Sistema Operativo para completar el proceso.

11.2. Recursos externos – Ejemplos de códigos QR

Ilustración 54: QR equivalente a 10 puntos

Ilustración 55: QR equivalente a 20 puntos

11.3. Manual de Usuario

¿Necesito estar registrado?

En el primer inicio de nuestra APP se da la opción de acceder como Invitado o de acceder mediante una cuenta de usuario.

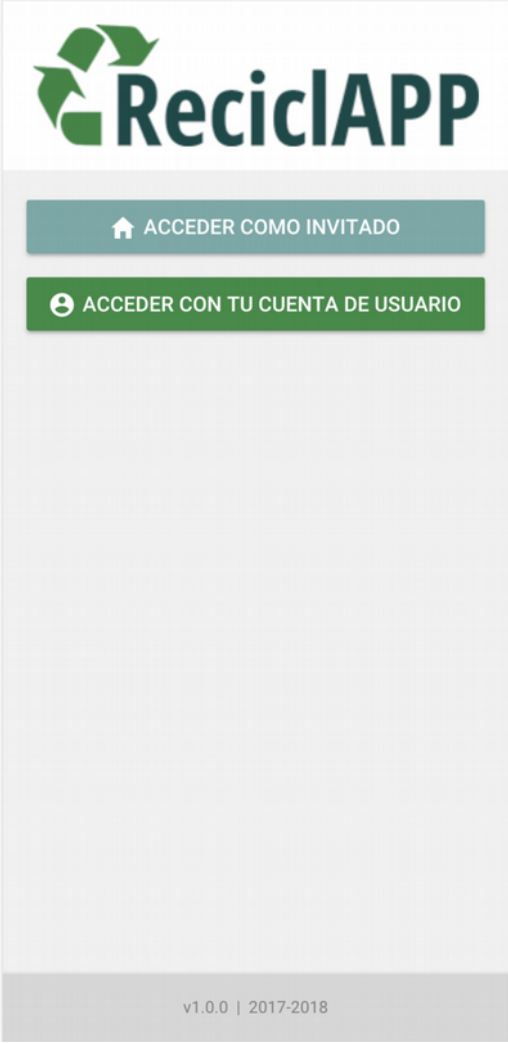

Ilustración 56: Manual, Inicio

¿Cómo me registro?

Accedemos a la opción de “Acceder con tu cuenta de Usuario” y luego accedemos a la opción “¿No tienes cuenta? Regístrate aquí”.

Deberás rellenar los campos con tus datos reales. El sistema comprueba que el email no exista previamente en la base de datos y en caso contrario recibirás un email con un enlace para que puedas validar tu cuenta.

Si todo ha ido correctamente, ya podrás acceder mediante el formulario de Login el email y contraseña que has facilitado.

He olvidado mi contraseña

En ese caso, debajo del formulario de Login aparece un botón con el texto “¿Has olvidado tu contraseña?” de manera que accediendo e introduciendo tu correo electrónico, el sistema te enviará un email con un enlace para poder restablecer tu nueva contraseña.

Ilustración 57: Manual, Acceder con tu cuenta

He accedido, ¿cómo encuentro un residuo y su contenedor?

Tanto si se ha accedido como invitado o como usuario registrado la búsqueda de un residuo o contenedor se realiza desde la opción de Navegación "Reciclar"

Ilustración 58: Manual, Opciones de Navegación

Desde esta pantalla puedes introducir el nombre del residuo en cuestión y el sistema te proporciona una lista con resultados coincidentes. Al seleccionar el resultado deseado, se te informará del contenedor correspondiente.

Ilustración 59: Manual, Ejemplo de búsqueda

¿Cómo puedo conseguir puntos cívicos?

Desde la opción de 'Ganar Puntos' podrás escanear el código QR (que te entregará el operario) del local al que has acudido para reciclar tus residuos.

Esta opción sólo esta disponible para usuarios registrados.

En toda la APP, existe un botón en la Toolbar del Header que nos abre el formulario de Login en caso de no tener iniciada la sesión o el formulario del Perfil en caso de estar ya registrados.

Ilustración 60: Manual, Botón Usuario

¿Cómo puedo cambiar los datos del Perfil?

Una vez iniciada la sesión, accediendo al Botón Usuario (ver Ilustración 60) aparecerá un formulario con los datos actuales para poder modificarlos.

Ilustración 61: Manual, Perfil

¿Cómo puedo finalizar mi sesión?

Una vez iniciada la sesión, accediendo al Botón Usuario (ver Ilustración 60) y posteriormente al botón de Logout situado en la parte izquierda de la Toolbar del Header.

¿Cómo puedo saber cuantos puntos tengo?

Una vez iniciada la sesión, accediendo al Botón Usuario (ver Ilustración 60) y posteriormente a la opción de Puntos, aparecerá un listado con los movimientos de puntos realizados y la cantidad de puntos acumulados y disponibles actualmente.

¿Cómo puedo cambiar mis puntos por regalos?

Una vez iniciada la sesión, accediendo al Botón Usuario (ver Ilustración 60) y posteriormente a la opción de Puntos, aparecerá un botón que te derivará a la plataforma CiviClub.

Ilustración 62: Manual, Mis Puntos

¿Cómo puedo saber mi posición en el ranking de las personas que más reciclan?

Desde la pantalla de Inicio, en la parte posterior aparece un botón “Lista completa” con el Top10 y tu posición en el ranking (basada en los puntos acumulados).

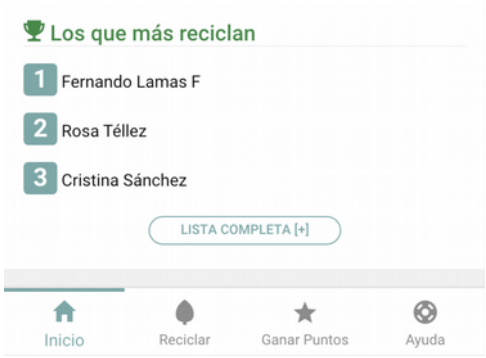

Ilustración 63: Manual, Acceder al Top10