

Sistema de control de menús de residencias universitarias: APP para dispositivos Android

Estudiante

Urko Lopez de Abechuco Ruiz de Mendarozqueta – urko@urko.eus

Máster Universitario en Ingeniería Informática

Desarrollo de Aplicaciones sobre Dispositivos Móvil

Profesores colaboradores

Jordi Ceballos Villach

Jordi Almirall López

Profesor responsable

Robert Clarisó Viladrosa

Fecha de entrega

10 de enero de 2018



Esta obra está sujeta a una licencia de Reconocimiento
– No Comercial – Sin Obra Derivada
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Hace algo más de dos años, terminaba de escribir la memoria del Trabajo Fin de Grado (TFG). Los agradecimientos de aquella memoria empezaban con algo parecido a:

“No cabe ninguna duda de que no podría estar aquí, escribiendo estas últimas líneas de la memoria, sin la ayuda de muchas personas que han pasado al lado mío durante mis, casi, dos patitos de edad.”

En aquel entonces, agradecía a muchas personas que habían estado a mi lado. Algunas de esas personas, por el motivo que haya sido, han salido o han tenido que salir de mi vida durante el transcurso de estos dos años. No es fácil afrontar la salida de ciertas personas, sobre todo cuando la situación no ha sido la más sencilla.

Aun así, hay veces que hay que seguir adelante, intentando mirar atrás lo justo y necesario, y contando con las personas que sabes que siempre van a estar ahí. Y en este momento, me gustaría volver a escribir algunas de las mismas líneas que escribí hace dos años:

“Sin duda alguna, tengo que empezar agradeciendo a mis “aitas”, Miren y Patxi, que sin ellos no podría ser la persona que soy. Asimismo, a mi hermana, Oihane, que, aunque no hemos podido tener la relación que me hubiese gustado, sé que siempre estás en los malos y buenos momentos.”

Creo que no hace falta añadir ni una sola palabra más, las propias líneas de hace dos años vuelven a significar lo mismo, y así seguirán. Mila esker guztiagatik.

Por otro lado, agradecer a las personas que han hecho posible este proyecto por parte de RESA, entre ellas Amaia Ruiz, Carlos Cano e Irupé Barroeta.

Por último, pero no menos importante, me gustaría agradecer a todas esas personas que habéis estado al lado mío, algunas de ellas ayudándome, otras haciéndome sonreír. Y bueno, también a esas personas que están por venir pero que poco a poco vas conociendo. No tenía pensado poner nombres, porque siempre se me olvidan, pero... A mi compi de cocina, que tantas charlas hemos *tenido que tener*: Kevin. A todas esas personas que habéis estado cuando lo he necesitado: Iratxe, Saioa, Piko... A todas esas personas, que, si alguna vez os tengo que dejar, marcareis un antes y un después, Luis, entre otros. Me dejo a mucha gente, pero como se suele decir, por problemas de espacio no puedo nombrar a todas.

Me gustaría terminar igual que hace dos años:

“Ya me habían dicho que la vida no era fácil, pero nadie me había avisado que en algunos momentos fuera a ser tan dura. Aquí se acaba la mejor etapa de mi vida, pero por primera vez, con unas pocas ganas de seguir luchando por hacer que mi vida mejore.”

Todo sigue adelante. Gracias una vez más.

Urko.

PD: *De vez en cuando, hay que dejar de pensar en los demás, y empezar a pensar en ti mismo, aunque sea sólo unos segundos.* Los procesos de cambio, sólo se dan contigo mismo, cambiando momentáneamente el entorno y dejándote respirar a ti sólo por un tiempo.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Sistema de control de menú de residencias universitarias: APP para dispositivos Android
Nombre del autor:	Urko Lopez de Abechuco Ruiz de Mendarozqueta
Nombre de los consultores:	Jordi Ceballos Villach Jordi Almirall López
Nombre del PRA:	Robert Clarisó Viladrosa
Fecha de entrega:	(10/2018)
Titulación:	Máster Universitario en Ingeniería Informática
Área del Trabajo Final:	Desarrollo de Aplicaciones sobre Dispositivos Móvil
Idioma del trabajo:	Castellano
Palabras clave:	Android – Residencias – Dispositivos móviles

Resumen del Trabajo:

En este proyecto, se ha propuesto la creación de una aplicación, con el objetivo de satisfacer algunas necesidades de los y las jóvenes universitarias alojadas en residencias con comedor.

Entre ellas, se encuentra la necesidad de saber en todo momento, y desde cualquier lugar, cuál va a ser el menú de la residencia en los próximos días y decidir si se quiere encargar la comida para llevar, por ejemplo, si se dispone de poco tiempo entre clases para comer. Hasta ahora, esta necesidad se cubría mediante el uso de tableros de información y redes sociales, lo cual complicaba la búsqueda rápida de información.

El resultado de este proyecto ha sido una aplicación para dispositivos Android, que recupera la información necesaria desde un back-end (cuál es el menú diario, opciones disponibles, información sobre alérgenos, etc.). Así mismo, permite encargar una comida para llevar, contratar comidas extras o, incluso, recargar el monedero virtual de la persona usuaria.

Para la realización del proyecto, se ha contado con la colaboración de RESA, empresa líder en residencias universitarias en el estado.

Abstract:

In this project, the creation of an application has been proposed, with the aim of satisfying some needs of university students housed in residences with a dining room.

Among them, is the need to know always, from anywhere, what will be the menu of the residence in the coming days and decide if you want to order food to take-away, for example, if you have little time between classes to eat. Until now, this need was

covered by information boards and social networks, which complicated the rapid search of the information.

The result of this project has been an application for Android devices, which retrieves the necessary information from a back-end (what is the daily menu, available options, information about allergens, etc.). Likewise, it allows you to order a meal to go, buy extra meals or even recharge the virtual wallet of the user.

For the realization of the project, we have had the collaboration of RESA, leading company in university residences in the state.

Laburpena:

Proiektu honetan, jantokidun erresidentzia batean dauden unibertsitarioen premia batzuk asetzeko asmoz, aplikazio baten sormena proposatu da.

Premia horien artean hauxek daude: erresidentzian hurrengo egunetan egongo den menua jakitea, eta beharrezkoa izango balitz, modu erraz eta azkar batean janari konkretu bat eramateko prestatzeko abisua ematea. Orain arte horrelako premiak iragarki-taulen eta sare sozialen bidez asetzen ziren, eta honek informazioaren bilaketa arina oztopatzen zuen.

Proiektu honen emaitza Android gailuentzako aplikazio bat izan da. Aplikazio honen bidez back-end-an dagoen informazioa berreskuratzen da (eguneroko menua, aukerak, alergenak...). Aldi berean, eramateko janaria eskatzea, janari estrak kontratatzea, eta erabiltzailearen diru-zorro birtuala kargatzeko aukera ere ematen du.

Proiektu hau egiteko RESA enpresaren laguntzarekin kontatu da, Estatuko unibertsitate erresidentzien artean enpresa liderra.

Índice

1	Introducción	12
1.1	Contexto y justificación	12
1.2	Objetivos del trabajo	13
1.3	Uso de las distintas versiones de Android en el mundo	13
1.4	Enfoque y método seguido	14
1.5	Planificación del trabajo	14
1.6	Breve resumen de productos obtenidos	15
2	Diseño centrado en el usuario	16
2.1	Primera fase – Usuarios y contexto de uso	16
2.1.1	Métodos de indagación	16
2.1.2	Perfiles de usuario identificados	21
2.2	Segunda fase – Diseño conceptual	24
2.2.1	Escenarios de uso	24
2.2.2	Flujos de interacción	26
2.2.3	Personas	26
2.3	Tercera fase – Prototipado	27
2.3.1	Sketches	27
2.3.2	Prototipo horizontal de alta fidelidad	29
2.4	Cuarta fase – Evaluación	35
2.4.1	Preguntas para obtener información	35
2.4.2	Tareas a realizar por los usuarios y usuarias y sus preguntas	36
3	Casos de uso	40
3.1	CU-001 Identificarse	40
3.2	CU-002 Cerrar sesión	40
3.3	CU-003 Ver menú	40
3.4	CU-004 Ver tabla alérgenos	41
3.5	CU-005 Encargar comida para llevar	41
3.6	CU-006 Cancelar comida para llevar	42
3.7	CU-007 Ver ofertas	42
3.8	CU-008 Valorar servicio	43
3.9	CU-009 Recargar monedero virtual	43
3.10	CU-010 Encargar comida extra	44
3.11	CU-011 Mostrar notificación	44

4	Implementación en Android	45
4.1	Herramientas utilizadas	45
4.2	Back-end	45
4.3	Estructura del proyecto	46
4.3.1	Activity y fragments	46
4.3.2	Dominio	47
4.3.3	Servicios	47
4.3.4	Otros archivos	47
4.4	Traducciones	47
4.5	Resultado	48
5	Control de versiones: Git	55
5.1	Características básicas de Git	55
5.1.1	Instalación de Git en Ubuntu	55
5.1.2	Primeros pasos con Git	55
5.2	Uso de Git en el desarrollo del software en el proyecto	59
5.2.1	Git en el entorno de desarrollo	59
5.2.2	Elección de soporte de Git en la nube (servicio escogido)	60
6	Gestión del proyecto	61
6.1	Diario	61
6.2	Gestión del alcance	61
6.3	Gestión del tiempo	61
6.4	Gestión de costes	62
7	Conclusiones	63
8	Próximos pasos	65
8.1	Conectar al back-end existente	65
8.2	Publicar la aplicación	65
8.3	Implementar la aplicación para otros Smartphone	65
8.4	Añadir nuevas funciones y mejorar las existentes	65
8.5	Traducciones	65
9	Glosario	66
10	Bibliografía	69
11	Anexos	70
11.1	Manual: Compilar la aplicación	70
11.2	Manual: Instalar la aplicación en un dispositivo físico	70
11.3	Manual: Uso de la aplicación	70
11.3.1	Identificación y recuperación de la contraseña	71

11.3.2	Funcionamiento	71
--------	----------------	----

Lista de figuras

<i>Ilustración 1- Planificación</i>	<i>15</i>
<i>Ilustración 2- Flujo de interacción.....</i>	<i>26</i>
<i>Ilustración 3- Sketches (identificarse, pantalla principal y menú)</i>	<i>28</i>
<i>Ilustración 4- Sketches (ver menú, alérgenos y monedero)</i>	<i>28</i>
<i>Ilustración 5- Sketches (ofertas y valorar)</i>	<i>29</i>
<i>Ilustración 6- Sketches (peticiones)</i>	<i>29</i>
<i>Ilustración 7- Prototipo horizontal: identificación.</i>	<i>30</i>
<i>Ilustración 8- Prototipo horizontal: bienvenida.</i>	<i>30</i>
<i>Ilustración 9- Prototipo horizontal: navegación.</i>	<i>31</i>
<i>Ilustración 10- Prototipo horizontal: menú por día.</i>	<i>31</i>
<i>Ilustración 11- Prototipo horizontal: alérgenos.</i>	<i>32</i>
<i>Ilustración 12- Prototipo horizontal: monedero.</i>	<i>32</i>
<i>Ilustración 13- Prototipo horizontal: ofertas.</i>	<i>33</i>
<i>Ilustración 14- Prototipo horizontal: valorar I.</i>	<i>33</i>
<i>Ilustración 15- Prototipo horizontal: valorar II.</i>	<i>34</i>
<i>Ilustración 16- Prototipo horizontal: petición I.</i>	<i>34</i>
<i>Ilustración 17- Prototipo horizontal: petición II.</i>	<i>35</i>
<i>Ilustración 18- Actividades y fragmentos de la aplicación.</i>	<i>46</i>
<i>Ilustración 19- Dominio de la aplicación.....</i>	<i>47</i>
<i>Ilustración 20- Servicios de la implementación</i>	<i>47</i>
<i>Ilustración 21- Pantalla identificación</i>	<i>48</i>
<i>Ilustración 22- Pantalla de identificación con error.....</i>	<i>49</i>
<i>Ilustración 23- Pantalla principal.....</i>	<i>49</i>
<i>Ilustración 24- Pantalla de navegación</i>	<i>50</i>
<i>Ilustración 25- Pantalla del menú diario.....</i>	<i>50</i>
<i>Ilustración 26- Pantalla de alérgenos</i>	<i>51</i>
<i>Ilustración 27- Pantalla de ofertas</i>	<i>51</i>
<i>Ilustración 28- Pantalla de peticiones.....</i>	<i>52</i>
<i>Ilustración 29- Pantalla del monedero</i>	<i>52</i>
<i>Ilustración 30- Pantalla de valoraciones.....</i>	<i>53</i>
<i>Ilustración 31- Pantalla de información de la residencia.....</i>	<i>53</i>
<i>Ilustración 32- Pantalla de ajustes</i>	<i>54</i>
<i>Ilustración 33- Android Studio IDE.....</i>	<i>70</i>

Lista de tablas

<i>Tabla 1- Uso de las diferentes versiones de Android</i>	<i>13</i>
<i>Tabla 2- Caso de uso: CU-001</i>	<i>40</i>
<i>Tabla 3- Caso de uso: CU-002</i>	<i>40</i>
<i>Tabla 4- Caso de uso: CU-003</i>	<i>40</i>
<i>Tabla 5- Caso de uso: CU-004</i>	<i>41</i>
<i>Tabla 6- Caso de uso: CU-005</i>	<i>41</i>
<i>Tabla 7- Caso de uso: CU-006</i>	<i>42</i>
<i>Tabla 8- Caso de uso: CU-007</i>	<i>42</i>
<i>Tabla 9- Caso de uso: CU-008</i>	<i>43</i>
<i>Tabla 10- Caso de uso: CU-009</i>	<i>43</i>
<i>Tabla 11- Caso de uso: CU-010</i>	<i>44</i>
<i>Tabla 12- Caso de uso: CU-011</i>	<i>44</i>
<i>Tabla 13- Gestión de costes (en horas)</i>	<i>62</i>

1 Introducción

El proyecto que se presenta en esta memoria, se ha realizado durante el transcurso del primer semestre del curso 2017-2018, del Máster Universitario en Ingeniería Informática. Éste Trabajo de Fin de Máster (TFM), se ha enmarcado en un proyecto más grande que se estaba realizando con la colaboración de RESA.

1.1 Contexto y justificación

RESA (1), empresa especializada en servicios de hospedaje para universitarios y universitarias, abarca más de 8.000 plazas de alojamiento en 33 residencias universitarias ubicadas en 19 ciudades distintas del territorio estatal. La mayoría de ellas disponen de servicio de comedor, con posibilidad de contratar desayuno, media pensión o pensión completa.

Los orígenes del proyecto se remontan al mes de febrero de 2017. Tras unas conversaciones con Carlos Cano (director comercial y marketing de RESA), Amaia Ruiz (directora de la Residencia RESA Manuel Agud Querol), e Irupé Barroeta (responsable del programa RESA HUB dentro del departamento comercial), se decidió empezar a construir una solución para varios problemas que se encontraban los y las residentes en el día a día:

- La información de los menús semanales no estaba accesible: no existía un sitio web o una aplicación donde se mostrase esta información de forma actualizada. La única opción que disponían los y las residentes para acceder al menú, era ir al tablón de anuncios del comedor, y en el mejor de los casos, mirarlo en la página de Facebook de la residencia (si ésta se actualizaba regularmente).
- Cuando un o una residente quería pedir “take-away” se tenía que personar en recepción, para elegir la comida “para llevar” y la hora de recogida. Esto implicaba una presencialidad que en muchos casos no era necesaria y que resultaba engorrosa.
- No existía una herramienta con la cual los y las residentes pudiesen dar su opinión sobre la comida y/o sobre el servicio dado. Por otro lado, tampoco había un buzón donde se pudiesen dejar sugerencias o quejas.
- Para contratar comidas extras o algún servicio adicional, se tenía que bajar a recepción en horario de oficina y hacer la contratación desde ese lugar.

Como se puede apreciar en los puntos anteriores, los y las residentes no tenían las facilidades necesarias para hacer diferentes operaciones del día a día. Por este motivo, el proyecto en el que se engloba este sub-proyecto, el TFM, consistía en “construir” una aplicación capaz de controlar diferentes aspectos del día a día de los comedores de las residencias universitarias, facilitando las tareas a los y las residentes y a las personas encargadas.

La solución aportada en el proyecto permite, entre otras cosas, informar de los menús diarios y de las ofertas a los y las residentes, la solicitud de take-away (encargar comida para llevar) para un día en concreto, el control de las personas que han utilizado

el comedor mediante el uso de la tecnología NFC (en las residencias que esté disponible, quedando fuera del alcance de este proyecto), la valoración por parte de los y las clientes/as finales tanto del servicio y como de la comida, la opción de contratar uno o más servicios extras mediante el uso de un monedero virtual, etc.

Este proyecto es muy amplio, por este motivo, se decidió, con la mediación de uno de los consultores, limitar el alcance del TFM al front-end de Android del proyecto global. A partir de ahora, esta memoria se centrará en ese sub-proyecto.

1.2 Objetivos del trabajo

Realizar una aplicación pensada para dispositivos Android 4.0 o superior, con conexión a Internet y personas residentes en alguna de las residencias RESA con comedor.

Funcionalidades que abarcará la aplicación:

- Autenticación con el usuario dado por parte de la residencia.
- Ver el calendario semanal/mensual con las comidas de cada día.
- Ver los alérgenos de cada comida
- Ver ofertas puntuales de la residencia.
- Pedir o cancelar: “Take-Away” con personalización de cada comida.
- Recibir notificaciones push desde recepción.
- Ver el saldo actual del monedero virtual y recargarlo (la recarga se hará sin pasarela de pago, “virtual”).
- Valorar el servicio y la comida cada vez que se haga uso del servicio.
- Consultar la información de la residencia y del usuario.

1.3 Uso de las distintas versiones de Android en el mundo

Según la página oficial de Android (2) el uso de cada versión / API de Android es el que se puede observar en la siguiente tabla:

Tabla 1- Uso de las diferentes versiones de Android

Versión	Nombre	API	Distribución
2.3.3 - 2.3.7	Gingerbread	10	0.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.5%
4.1.x		16	1.9%
4.2.x	Jelly Bean	17	2.9%
4.3		18	0.8%
4.4	KitKat	19	12.8%
5.0		21	5.7%
5.1	Lollipop	22	19.4%
6.0	Marshmallow	23	28.6%

7.0	Nougat	24	21.1%
7.1		25	5.2%
8.0	Oreo	26	0.5%
8.1		27	0.2%

Como se puede intuir, el haber utilizado como mínimo la API 15 (Android Ice Cream Sandwich, 4.03) permite que la aplicación sea instalada en, casi, cualquier dispositivo que tenga Android.

1.4 Enfoque y método seguido

Actualmente no se dispone de ningún producto que cumpla las mismas funciones que se buscan en este proyecto. Es verdad, que existen aplicaciones que permiten pagar con el móvil, o incluso hacer un pedido en un restaurante, pero dadas las particularidades del problema planteado, no es posible adaptar una aplicación a las características de este proyecto.

Además, dado que el proyecto depende de un back-end externo preparado para el problema en particular, el enfoque más correcto sería realizar una aplicación a medida para los casos de uso dado.

1.5 Planificación del trabajo

Recursos necesarios:

- Ordenador con Android Studio.
- Emulador Android y/o dispositivo físico con Android.
- Lista de funciones del back-end actual, o en caso de no estar disponible, un back-end con base de datos que podría suplir el anterior.

Tareas y organización de trabajo:

- Hacer el diseño de la aplicación teniendo en cuenta la metodología DCU:
 - Usuarios y usuarias que lo van a usar y su contexto de uso.
 - Diseño conceptual de la aplicación.
 - Prototipado.
 - Evaluación de los resultados.
- Especificar el diseño técnico:
 - Definición de casos de uso.
 - Diseñar la arquitectura de la aplicación.
- Construir la aplicación de Android en diferentes iteraciones:
 - Fase Alpha.
 - Fase Beta

- Prueba con Usuarios.
- Fase Final

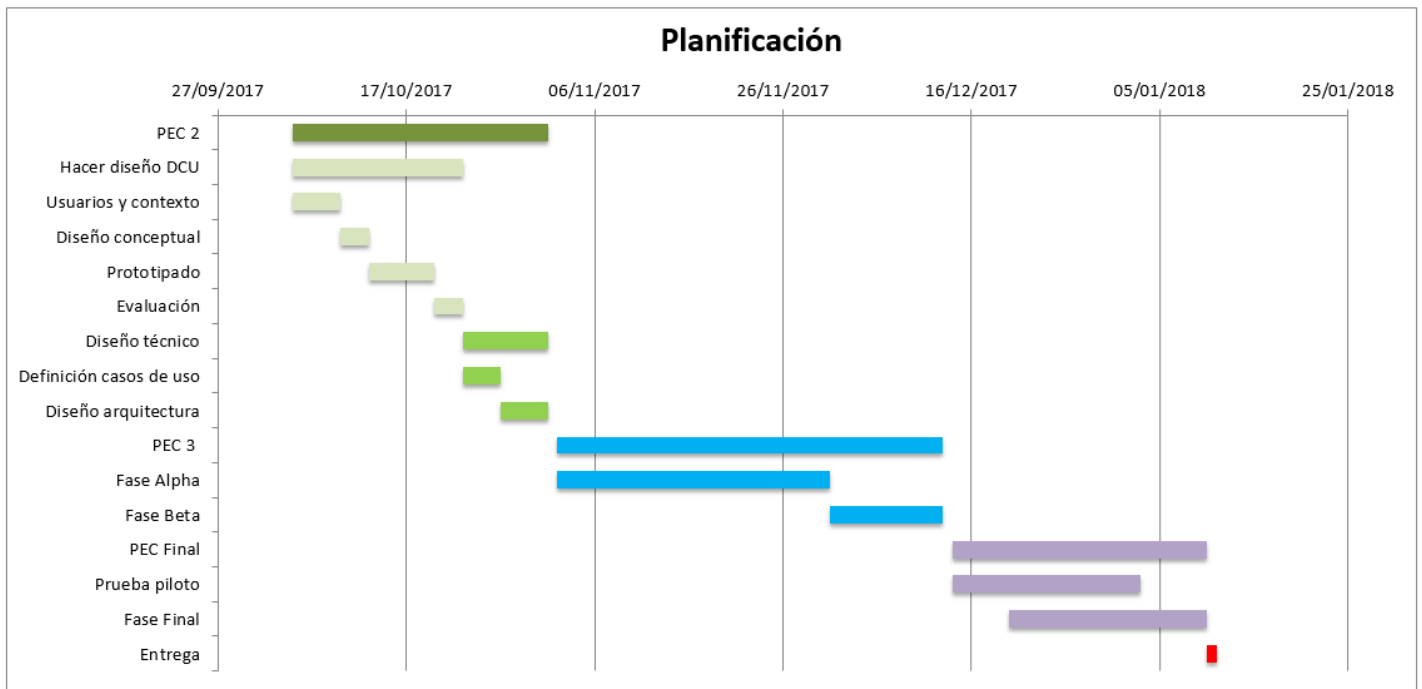


Ilustración 1- Planificación

1.6 Breve resumen de productos obtenidos

Se obtendrá un único producto, aparte del documento actual: aplicación funcional para dispositivos móviles Android, junto con su código fuente. Así mismo, se adjunta una presentación del proyecto, un manual de usuario y un manual de instalación.

El back-end y el resto de funcionalidades, quedan fuera del alcance del proyecto.

2 Diseño centrado en el usuario

En este capítulo se describe el proceso que se ha seguido para obtener un diseño mediante el uso de diferentes técnicas del DCU (diseño centrado en el usuario). Para ello, y tal y como recomienda el DCU, se han seguido cuatro fases (3) que se explican seguidamente:

2.1 Primera fase – Usuarios y contexto de uso

A continuación, se muestran los diferentes métodos de indagación que se han utilizado, y, por último, los resultados obtenidos: perfiles de usuarios identificados.

2.1.1 Métodos de indagación

En la primera fase se han usado tres métodos de indagación: Shadowing, entrevistas y encuestas. Varios han sido los motivos para la elección de estos métodos:

- Poder entender, de una manera directa, el entorno de los y las residentes cuando están haciendo alguna actividad relacionado con el comedor.
- Recibir feedback completo de algunos usuarios y usuarias.
- Entender mejor los contextos de uso y las características de los perfiles de usuario o usuaria.

Estos tres métodos se han hecho de una forma líneal y ordenada, es decir, primero se ha empezado haciendo una observación de campo (shadowing), a continuación, se ha entrevistado a 5 personas y, por último, se ha realizado una encuesta vía Google Form al conjunto de residentes.

2.1.1.1 *Shadowing*

Durante dos turnos de comida se ha desplazado un observador al comedor y ha estado “infiltrado”, como un individuo más, de la forma más discreta posible. Ha hecho lo que realizaría cualquier residente: entrar al comedor, identificarse y comer. Así mismo, ha estado en algunos momentos clave: cuando una persona quería encargar una comida para llevar (take-away) y cuando una persona quería saber cuál iba a ser el menú diario/semanal (en el tablón).

En este caso, se ha preferido que el observador no realizase ningún tipo de pregunta a las personas que se ha ido encontrando, ya que el objetivo era reunir la máxima información posible para poder definir de mejor manera las siguientes fases, es decir, se puede entender como un método que se ha utilizado con el objetivo de entender el contexto.

A continuación, se presentan algunos de los aspectos más interesantes que se han obtenido mediante la observación de campo:

- El menú diario se cuelga en un tablón exclusivo para ello. No hay ningún otro método (Redes sociales, envío de emails, pagina web, etc.)

- Si un o una residente quiere encargar take-away tiene que hacerlo con antelación y en presencia de la persona responsable. No hay forma de hacerlo por otro medio.
- El control del comedor se hace mediante listado.
- La mayoría de la gente es muy joven y dispone de algún dispositivo móvil.

2.1.1.2 Entrevista

En este método se ha hecho una entrevista personal a 5 personas, las cuales son usuarias de las instalaciones del comedor. La entrevista no tenía un guion marcado, sino que se esperaba que fuesen surgiendo ideas a medida que se iba entrevistando. Eso si, había algunos puntos importantes sobre los que recoger información:

- Como sabe cada persona encuestada el menú que va a tener al día siguiente o en la próxima semana.
- Cuál es la dinámica de uso del comedor según el punto de vista de los y las encuestados.
- Si encargan la comida para llevar o no. En caso negativo, los motivos de este hecho.
- Si valoran el servicio y como lo hacen en caso afirmativo.

Para realizar la entrevista se han seleccionado cinco personas aleatoriamente: tres chicas y dos chicos. Dicha entrevista, se ha realizado en la propia residencia, en una sala que se ha destinado momentáneamente para dicho uso. Cada una de ellas no ha llegado a superar los 15 minutos, siendo la media alrededor de 10. Hay que tener en cuenta, que al comienzo de la entrevista se ha explicado la motivación del proyecto, y a la vez, las condiciones de participación en la entrevista: anónima, sólo con apuntes, sin grabación de audio.

Los resultados más relevantes de las entrevistas han sido los siguientes:

- Ninguna de las personas entrevistadas suele valorar el servicio ya que no tienen una forma fácil de hacerlo: no hay un buzón de sugerencias y tampoco una plantilla donde rellenar un cuestionario de forma rápida y entregarlo. En caso de que quieran sugerir algo, se lo dicen de forma verbal a las personas encargadas del comedor.
- No suelen pedir comida para llevar porque en la práctica les supone un engorro: hay que avisar con mucha antelación y además de forma presencial, lo que supone planificar de antemano los siguientes días.
- Tres de las cinco personas entrevistadas han declarado no saber que van a comer en el día, por no tener una manera sencilla de consultarlo. Los cinco han reclamado el uso de una aplicación o web para este propósito.
- Dos personas han mencionado que, si tienes Media Pensión y quieres comer también, hay que avisar y pagar en recepción con antelación, lo que les ha supuesto en la mayoría de ocasiones tener que comer en el campus.

2.1.1.3 Encuesta

En el último método de indagación se ha realizado una encuesta a los residentes. Para ello, se ha preparado un Google Form (formularios de Google Drive) y se les ha pasado el enlace mediante email a todas las personas que residen en la residencia escogida.

Para el planteamiento, se ha usado toda la información que se ha recopilado mediante los métodos usados anteriormente, haciendo preguntas más enfocadas a la información que se quería averiguar. Así mismo, también se han hecho preguntas más genéricas, para poder tener diferentes perfiles de usuario.

A continuación, se muestra un listado de las preguntas con las posibles respuestas. En ellas mismas, se puede encontrar el resultado de los 50 residentes que han contestado y cuyas respuestas se han dado por válidas.

Encuesta

Datos personales:

1. Me considero:
 - a. Mujer (58%)
 - b. Hombre (42%)
2. Mi edad es:
 - a. Entre 17 y 19 años (88%)
 - b. Entre 20 y 23 años (12%)
 - c. Mayor de 24 años (0%)
3. Este es mi ... en la residencia.
 - a. Primer año (62%)
 - b. Segundo año (30%)
 - c. Tercer año o más (8%)
4. Dispongo de un dispositivo
 - a. Android (82%)
 - b. Iphone (18%)
 - c. Otro: _____ (0%)
5. Dispongo de un plan de datos en el dispositivo
 - a. Correcto (100%)
 - b. Incorrecto (0%)
 - c. No sé (0%)

Uso general del comedor:

6. El plan de comidas que tengo contratado es:
 - a. Media pensión (80%)

- b. Pensión completa (20%)
- 7. En caso de sólo tener media pensión y/o querer “invitar” a otra persona a cenar, el proceso es:
 - a. Fácil, me parece muy sencillo (0%)
 - b. Normal, sólo tengo que avisar (24%)
 - c. Difícil al tener que contratarlo con mucha antelación y en horario de oficina (46%)
 - d. Nunca he tenido que hacerlo (30%)
- 8. Me gustaría tener una aplicación para poder hacer el anterior proceso de una manera mucho más sencilla, incluyendo la opción de tener un monedero virtual para hacer el pago:
 - a. Si (46%)
 - b. Si, pero sin monedero (14%)
 - c. No (12%)
 - d. Indiferente (28%)
- 9. El uso que le doy al comedor es el siguiente la mayoría de veces (marque las casillas necesarias; una, dos o tres):
 - a. Desayunar (100%)
 - b. Comer (70%)
 - c. Cenar (100%)
- 10. Normalmente, cuando voy al comedor a comer o a cenar:
 - a. Bajo solo (0%)
 - b. Bajo solo, pero me uno con los que estén en ese momento (10%)
 - c. Suelo quedar con amigos o amigas para ir (62%)
 - d. Siempre acudo con las mismas personas (28%)
- 11. El motivo de la respuesta anterior es:
 - a. Me gusta bajar siempre con las mismas personas (20%)
 - b. No me queda otra, ya que no tengo medios para quedar (un chat, aplicación para quedadas, etc.) (16%)
 - c. La verdad, me es indiferente (48%)
 - d. No me identifico con ninguna de las respuestas anteriores (16%)

Menú

- 12. Consultaría, si estuviese la opción disponible, el menú en los siguientes medios (marcar los necesarios)
 - a. Tablón de anuncios del comedor (56%)
 - b. Redes sociales (34%)
 - c. Web de la residencia (12%)
 - d. Aplicación para dispositivos móviles (82%)
 - e. No lo miro (24%)
 - f. No sé dónde lo miraría (6%)
- 13. Suelo saber el menú que va a haber antes de entrar al comedor
 - a. Siempre (18%)
 - b. Casi siempre (52%)
 - c. Casi nunca (16%)
 - d. Nunca (14%)

14. Me gustaría tener una aplicación con toda la información recogida sobre el menú diario
 - a. Si (78%)
 - b. Indiferente (22%)
 - c. No (0%)
15. Creo que el hecho de no tener la información del menú en el móvil es un hándicap para no saber lo que va a comer antes de entrar al comedor:
 - a. Si (56%)
 - b. No sé (24%)
 - c. No (20%)

Comida para llevar:

16. Suelo encargar la comida para llevar en las siguientes ocasiones:
 - a. Nunca (62%)
 - b. Entre 1 y 5 ocasiones al mes (10%)
 - c. Entre 5 y 10 ocasiones al mes (18%)
 - d. Entre 10 y 20 ocasiones al mes (10%)
 - e. Casi diariamente o diariamente (0%)
17. Me identifico más con la siguiente frase:
 - a. Encargar la comida para llevar es muy fácil y no tiene complicaciones (6%)
 - b. Aunque sea fácil encargar la comida para llevar, me gustaría que el proceso fuese más simple (8%)
 - c. El proceso para encargar la comida no es el adecuado y me hace perder mucho tiempo (12%)
 - d. No pido comida para llevar por el proceso de encargarla (38%)
 - e. No me identifico con ninguna de las anteriores (36%)
18. Me gustaría disponer de una aplicación para poder encargar la comida para llevar:
 - a. Si (76)
 - b. Indiferente (22%)
 - c. No (2%)

Valorar la comida y/o el servicio

19. Soy consciente de que existe algún medio escrito para valorar la comida y/o el servicio que se presta en el comedor:
 - a. Si (0%)
 - b. No (92%)
 - c. No me acuerdo. (8%)
20. He valorado alguna vez la comida y/o el servicio
 - a. Si (10%)
 - b. No (76%)
 - c. No me acuerdo. (14%)
21. En caso afirmativo, podría indicar el procedimiento brevemente:
 - a. _____
Hablar con el personal encargado del comedor (8%)

Dirección (2%)

22. Me gustaría tener un método más sencillo para dar una valoración, como una aplicación móvil donde me salga una alerta cada vez que uso el comedor:

- a. Si (68%)
- b. Indiferente (30%)
- c. No (2%)

Muchas gracias por participar en la encuesta.

2.1.1.4 Conclusiones

Como conclusiones finales de los métodos de indagación se podrían resaltar las siguientes cuestiones:

- Todos los y las residentes consultadas tienen un Smartphone, teniendo la mayoría de los dispositivos el sistema operativo Android. Así mismo, todos disponen de una tarifa de datos. Seguramente debido a que la edad se enmarca entre 17 y 23 años.
- El uso del comedor es mayoritario en las tres comidas, aunque en la del mediodía baja el uso. En caso de querer contratar alguna comida extra para ellos o para otra persona, el proceso no es sencillo para más de la mitad de las personas.
- Mas de la mitad de los y las residentes cree que el motivo de no saber el menú es no tener un medio adecuado para ello, como una aplicación para dispositivos móviles.
- La mayoría de los y las residentes creen que no hay un método escrito para valorar el servicio y/o la comida.
- Como curiosidad, a tener en cuenta, sólo los y las antiguas tienden a encargar la comida para llevar, así mismo, son los únicos que nunca bajan solos.
- La mayoría de residentes quieren una aplicación para poder consultar el menú diario y encargar la comida para llevar. Así mismo, les gustaría tener en esa misma aplicación la opción de valorar el servicio y contratar las comidas extras, junto a un monedo virtual para hacer los pagos.

2.1.2 Perfiles de usuario identificados

A continuación, se muestran los perfiles de usuarios identificados siguiendo los métodos de indagación expuestos en el apartado anterior:

2.1.2.1 Nuevos residentes

Son usuarios y usuarias recién llegados a la residencia (comúnmente llamados “novatos”), normalmente del primer año del grado, lo que conlleva, en la mayoría de los

casos, tener 17-18 años. Proviene de diferentes ciudades o pueblos, y se alojan en una de las residencias RESA de la ciudad elegida para estudiar.

Al ser el primer año fuera de casa, los intereses se centran en emanciparse y crear un nuevo grupo de amigos y amigas. Para ello, la residencia es tan vital como el estar integrado dentro de ella. En concreto, el comedor es fundamental para poder completar esa función satisfactoriamente.

En este caso, el uso de la tecnología móvil es muy fuerte, debido al alto uso de las redes sociales y mensajería, como WhatsApp. En la mayoría de casos suelen disponer de plan de datos en el smartphone.

Esta aplicación la usaran en cualquier momento del día en el que tengan un interés concreto, fundamentalmente ver cuál va a ser el menú de ese día o de días posteriores.

Tareas:

- Descargar la aplicación desde la tienda de aplicaciones.
- Identificarse con el usuario y contraseña dado por la residencia
- Ver el menú del día actual o de otro día.
- Inscribirse a quedadas para coincidir en el comedor.

Listado de características descubiertas:

- Opción de hacer una quedada para ir al comedor unas cuantas personas.

2.1.2.2 Renovados o residentes

Son usuarios o usuarias que llevan, por lo menos, un año alojados en esa residencia (comúnmente llamados “veteranos”). Ya han pasado el proceso de emanciparse, y en la mayoría de los casos, ya disponen de una “cuadrilla” con la que suelen bajar al comedor.

Al tener su cuadrilla, no “necesitan” hacer nuevas amistades, aunque suelen querer incorporar a nuevas personas al grupo al principio del curso. Para ello, el comedor es fundamental y sobre todo la idea de crear convocatorias. Ellos jugarían principalmente el rol de organizadores de esas quedadas.

En este caso, el uso de la tecnología móvil es muy amplio, debido al alto uso de las redes sociales y mensajería, como WhatsApp. En la mayoría de casos suelen disponer de plan de datos en el smartphone.

La aplicación la usaran en cualquier momento del día en el que tengan un interés concreto, fundamentalmente ver cuál va a ser el menú de ese día o de días posteriores. Esta vez, también sería fundamental el uso de la app una hora antes de ir al comedor para organizar las quedadas.

Tareas:

- Descargar la aplicación desde la tienda de aplicaciones.
- Identificarse con el usuario y contraseña dado por la residencia
- Ver el menú del día actual o de otro día.

- Poder organizar quedadas para coincidir en el comedor con nuevas y nuevos residentes.

Listado de características descubiertas:

- El perfil de organizador/a – coordinador/a de quedadas dentro de la app, como actividad de integración.

2.1.2.3 Residentes con posibilidad de solicitar comidas extras

Son personas que tienen contratado el régimen de media pensión, lo que les supone la posibilidad de coger comidas extras.

En este caso, mayormente van a ser residentes que lleven más de un año en la residencia, ya que las y los nuevos suelen contratar pensión completa.

Una vez más, la experiencia en el uso de la tecnología va a ser muy importante.

La aplicación la usaran mayormente entre las 11:00 y las 14:00, con el fin de ver el menú de ese día y contratar la comida extra.

Tareas:

- Descargar la aplicación desde la tienda de aplicaciones (si no está descargada).
- Identificarse con el usuario y contraseña dado por la residencia
- Ver el menú del día ese día y si tuviese interés, utilizar la opción de contratar esa comida directamente desde la aplicación.

Listado de características descubiertas:

- Opción de tener un monedero virtual con el que pagar la comida directamente desde el dispositivo móvil sin tener que acudir a recepción antes.

2.1.2.4 Residentes que cogen con frecuencia take-away

Son usuarios y usuarias que tienen contratado el régimen de pensión completa y tienen una distancia considerable hasta la universidad, con un horario de clases de mañana y tarde.

Una vez más, la experiencia en el uso de la tecnología va a ser muy extensa.

La aplicación la usarán mayormente el día anterior a la jornada que quieran reservar comida para llevar, en horario de tarde.

Tareas:

- Descargar la aplicación desde la tienda de aplicaciones.
- Identificarse con el usuario y contraseña dado por la residencia
- Ver el menú del día siguiente y encargar la comida para llevar.

Listado de características descubiertas:

- Poder cancelar la petición del take-away hasta una hora determinada.

2.1.2.5 Residentes con intolerancias, alergias u otras preferencias

Son residentes que tienen algún tipo de intolerancia, alergia o preferencias en la comida.

Una vez más, la experiencia en el uso de la tecnología va a ser muy intensa.

La aplicación la usarán mayormente en cualquier momento del día, pero sobre todo el día anterior al que quieren utilizar el comedor, para poder saber si tienen que hacer alguna petición especial.

Tareas:

- Descargar la aplicación desde la tienda de aplicaciones.
- Identificarse con el usuario y contraseña dado por la residencia
- Ver el menú del día siguiente y observar el cuadro de alérgenos.

Listado de características descubiertas:

- Tener un cuadro de alérgenos dentro de cada menú.

2.2 Segunda fase – Diseño conceptual

En esta fase se caracterizan los diferentes escenarios, se describe gráficamente los flujos de interacción, y, por último, se hace una descripción ficticia de personas que podrían llegar a ser usuarias de la aplicación.

2.2.1 Escenarios de uso

A continuación, se pueden observar una serie de escenarios que encajarían con los perfiles anteriormente nombrados.

2.2.1.1 Primer escenario

Un antiguo residente (perfil de usuario: renovador) está con un exresidente en una cafetería tomando algo y éste le invita a cenar un día en su casa. Así que quiere ver (objetivo) el menú semanal para poder decidir cuál es el día que menos le gusta la cena y acudir esa noche a casa del amigo.

Para ello (tareas), coge su móvil y abre la aplicación creada para este fin. Como es la primera vez que la va a abrir, antes tiene que ir a la tienda de su dispositivo móvil y descargarla. Una vez instalada y abierta, mete el usuario y contraseña, y a continuación, verá el menú de ese día. Irá cambiando de día, hasta encontrar la información que busca: cuál es el día con la peor cena según su criterio.

2.2.1.2 Segundo escenario

Una persona, recién llegada a la residencia (perfil de usuario: nuevo residente), acude por primera vez al comedor, y después de cenar, en su habitación, decide que quiere valorar el servicio (objetivo), ya que le ha gustado mucho la comida y el servicio. Aparte quiere hacer una sugerencia (objetivo).

Para ello (tareas), coge su móvil y se descarga la aplicación de RESA. Introduce el usuario y contraseña que le han dado a su llegada, y busca la opción de valorar por lo menús. En este caso, selecciona el día y la comida y mediante el uso de unas estrellas valora el servicio. Por último, introduce la sugerencia y pulsa enviar.

2.2.1.3 Tercer escenario

Una persona que tiene contratada media pensión (perfil de usuario: residente con posibilidad de coger comidas extras) está estudiando desde bien temprano. Son las 13:00 y se da cuenta que no quiere perder tiempo haciendo la comida, ya que va muy mal de tiempo. Por este motivo (objetivo), piensa que la mejor idea es comprar una comida extra en la propia residencia y comer en el comedor.

Para ello (tareas), abre la aplicación de su móvil y mira si tiene saldo en el monedero virtual. Como no tiene, decide recargar mediante tarjeta el monedero y comprar la comida extra. Antes de nada, comprueba que el menú que hay es de su agrado.

2.2.1.4 Cuarto escenario

Una chica con intolerancia a la proteína de leche (perfil de usuario: residentes con intolerancias o alergias) quiere comprobar el menú de la semana siguiente, para cerciorarse que no hay ningún alérgeno que no pueda consumir.

Para ello (tareas), abre la aplicación ya instalada en su smartphone y selecciona la opción necesaria para ver el menú de la siguiente semana. Va pasando comida tras comida, y comprueba en cada una de ellas la tabla de alérgenos.

2.2.1.5 Quinto escenario

Un grupo de amigos y amigas de la residencia está en clase, y el profesor les comunica que, una vez más, al día siguiente tienen una práctica obligatoria justo después de comer. Lo que conlleva, a que sólo dispongan de 45 minutos para comer. Por este motivo, y teniendo en cuenta que todos tienen pensión completa, deciden, como es habitual, (objetivo) encargar para el día siguiente la comida para llevar (perfil de usuario: residentes que cogen con frecuencia take-away) y que esté preparada para el desayuno.

Para ello (tareas), cada persona del grupo abre la aplicación, ve el menú del día siguiente en el apartado correspondiente y encarga la comida para llevar.

2.2.1.6 Sexto escenario

Uno de los amigos del quinto escenario (perfil de usuario: residentes que cogen con frecuencia take-away) se pone enfermo a la tarde, sobre las 18:30, y decide ir al médico. Éste le comunica que necesita dos días de reposo, lo que conlleva que no va a poder ir a las prácticas y decide cancelar el take-away del siguiente día (objetivo).

Para ello (tareas), vuelve a abrir la aplicación, va al apartado de peticiones y comprueba que está a tiempo de cancelar la petición. Le da al botón de cancelar y la aplicación le confirma que el día siguiente tendrá una comida normal.

2.2.1.7 Séptimo escenario

Un residente nuevo (perfil de usuario: nuevo residente) está en la universidad, concretamente en el descanso entre dos clases del grado. Desde la residencia, han tenido un problema y han tenido que cambiar la cena del día de hoy. Por este motivo, el residente ha recibido una notificación Push con el cambio de comida.

Para ello, cuando el residente ha visto la notificación, ha pulsado en ella y ha procedido a leer el aviso con el cambio de cena.

2.2.2 Flujos de interacción

En la siguiente imagen, se puede observar el flujo de interacción de la aplicación que se está modelando.

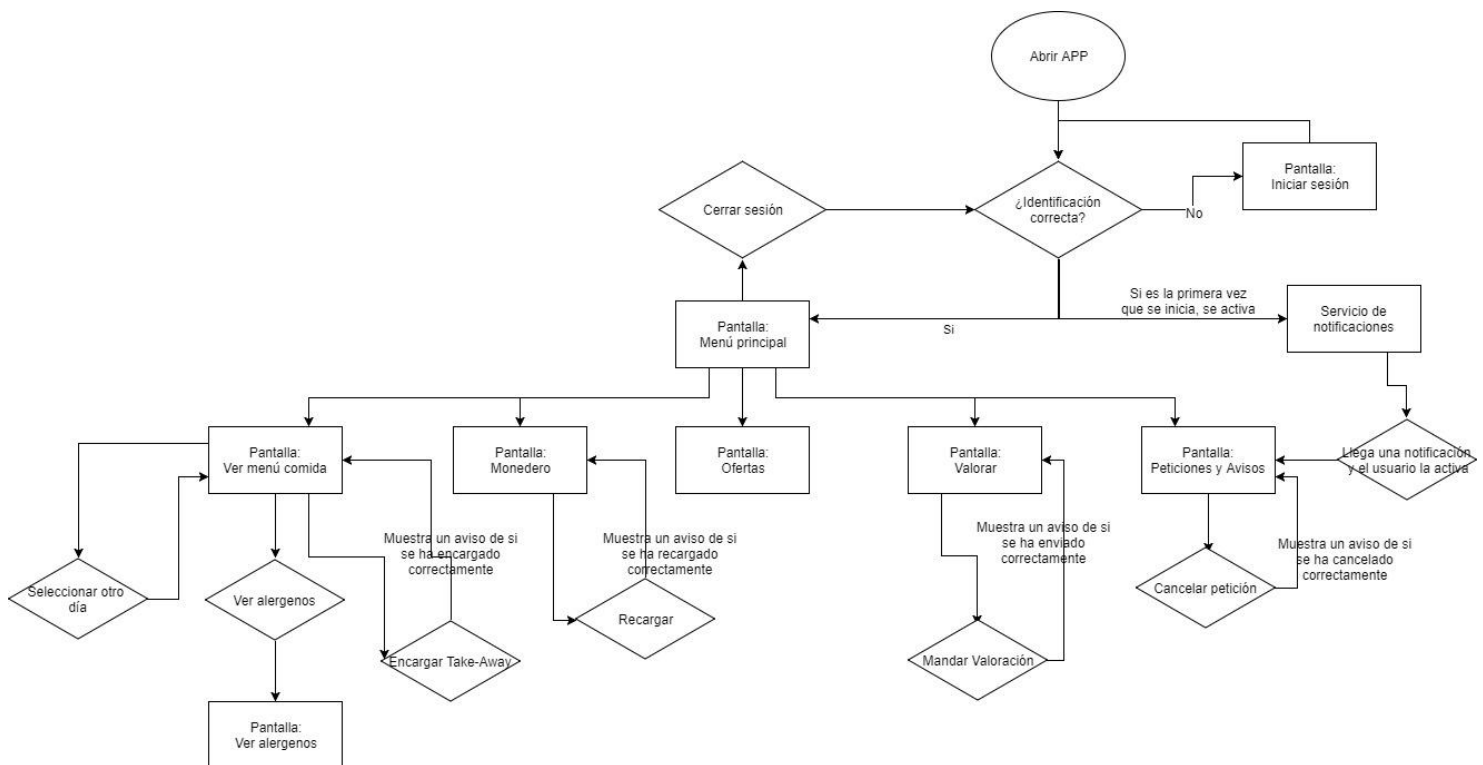


Ilustración 2- Flujo de interacción

2.2.3 Personas

A continuación, se muestran descripciones ficticias de personajes que podrían llegar a ser usuarios de la aplicación, basadas en los datos recopilados anteriormente:

2.2.3.1 Primera persona

Raúl es un chico de 18 años recién llegado a Barcelona. Es su primer año en la residencia y está empezando a estudiar el Grado en Ingeniería Informática. Ha hecho

nuevos amigos y amigas, tanto en la residencia, como en la universidad. Dispone de un Smartphone Android con tarifa plana de datos.

Los estudios le obligan a tener que acudir todos los días a clase, pero sólo durante la mañana. Por este motivo, podrá acudir todos los días a comer a la residencia que está a menos de 10 minutos del campus.

2.2.3.2 Segunda persona

Leire es una chica de 19 años que está estudiando periodismo. Es su segundo año en la universidad y en una de las residencias de Barcelona, por lo cual ya es veterana. Dispone de un móvil Android con una tarifa de Internet que le permite estar conectada en todo momento.

El campus está a más de media hora en metro. Además, tiene clases a la mañana y a la tarde, pero no todos los días. Por eso, desde el curso pasado tiene contratado media pensión en la residencia y de vez en cuando, si el tiempo se lo permite, contrata alguna comida extra.

2.2.3.3 Tercera persona

Esti, una amiga de Leire, también está estudiando periodismo en la universidad. Así mismo, es su segundo año en la residencia ya que está en segundo curso y tiene 19 años. Dispone de un móvil Samsung con Android y tarifa plana de datos.

Ella tiene contratada pensión completa, así que los días que tiene clase a la mañana y a la tarde, encarga la comida para llevar.

2.2.3.4 Cuarta persona

Santi lleva ya 3 años en la residencia. Es un chico de 21 años, veterano, que padece intolerancia a la proteína de la leche, lo que le trae dificultades a la hora de comer. Por este motivo y dado que en la residencia tienen mucho cuidado con las comidas, sigue en la residencia con pensión completa.

Está estudiando derecho y dispone de un dispositivo Android y tarifa de datos.

2.3 Tercera fase – Prototipado

En esta fase, se ha hecho el prototipado de la aplicación teniendo en cuenta los datos obtenidos en las fases anteriores, primero mediante la técnica de sketches y, a continuación, mediante un prototipo de alta fidelidad.

2.3.1 Sketches

En las siguientes imágenes se pueden ver los prototipos hechos a mano alzada (sketches). Cada pantalla representa un apartado de la aplicación.

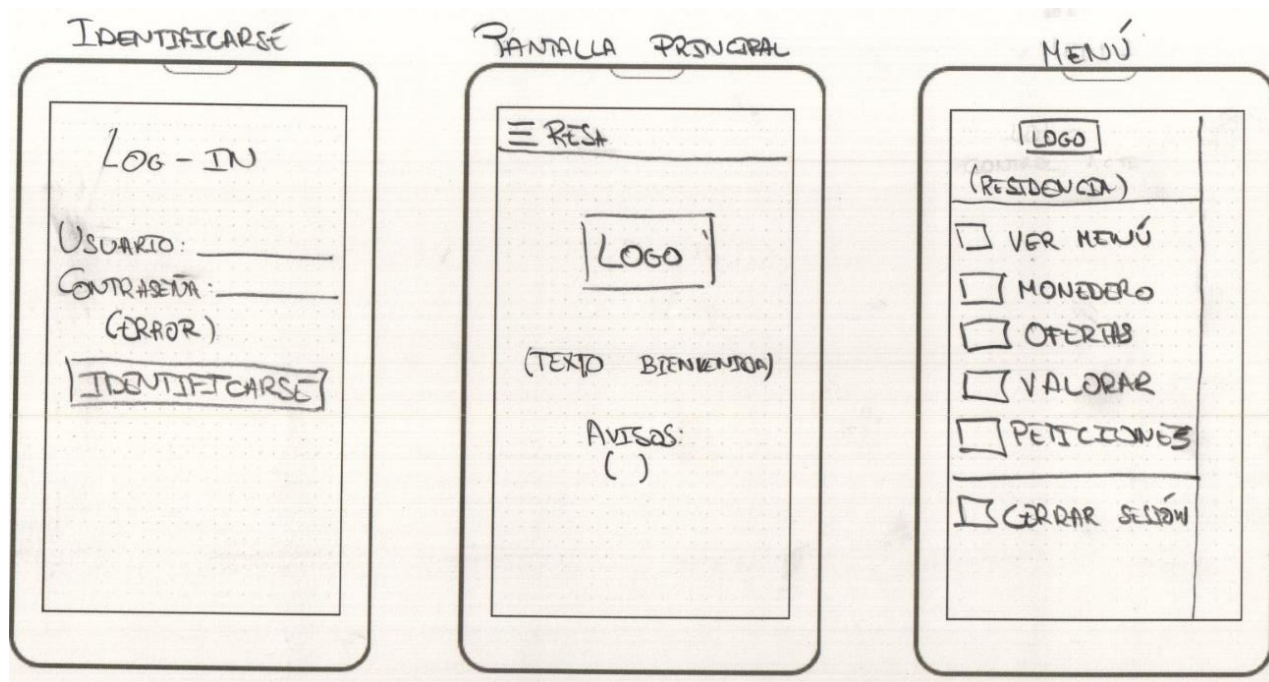


Ilustración 3- Sketches (identificarse, pantalla principal y menú)

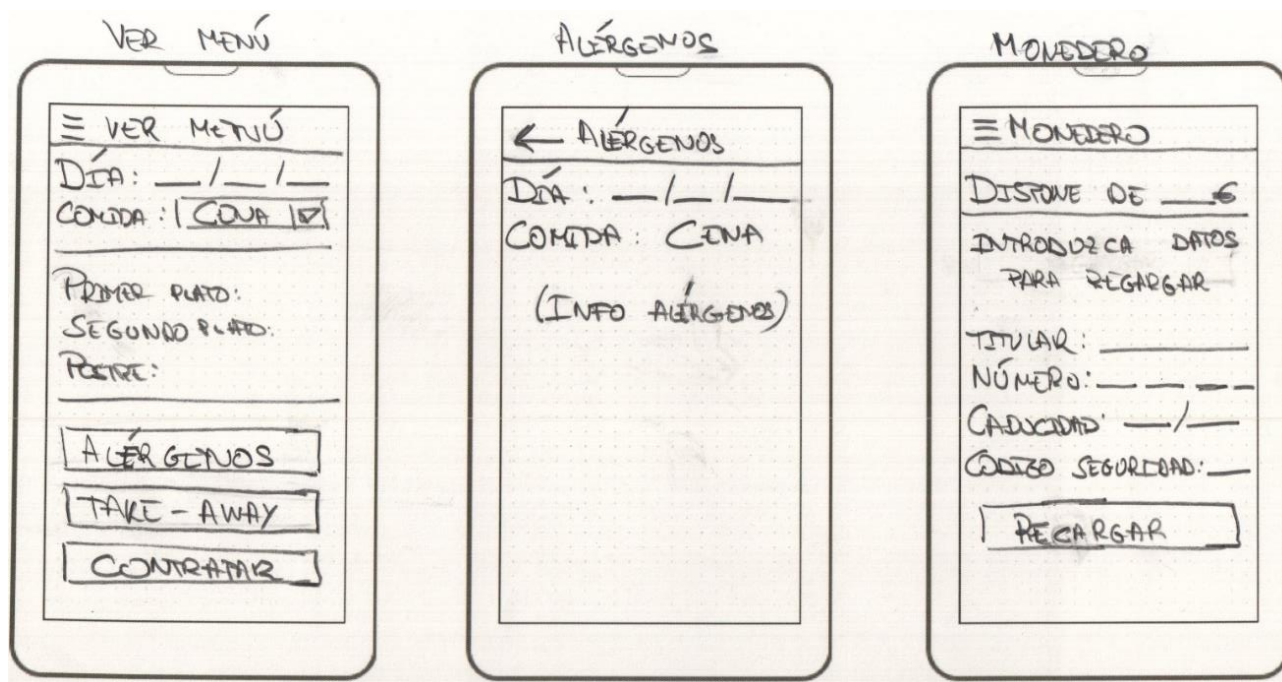


Ilustración 4- Sketches (ver menú, alérgicos y monedero)

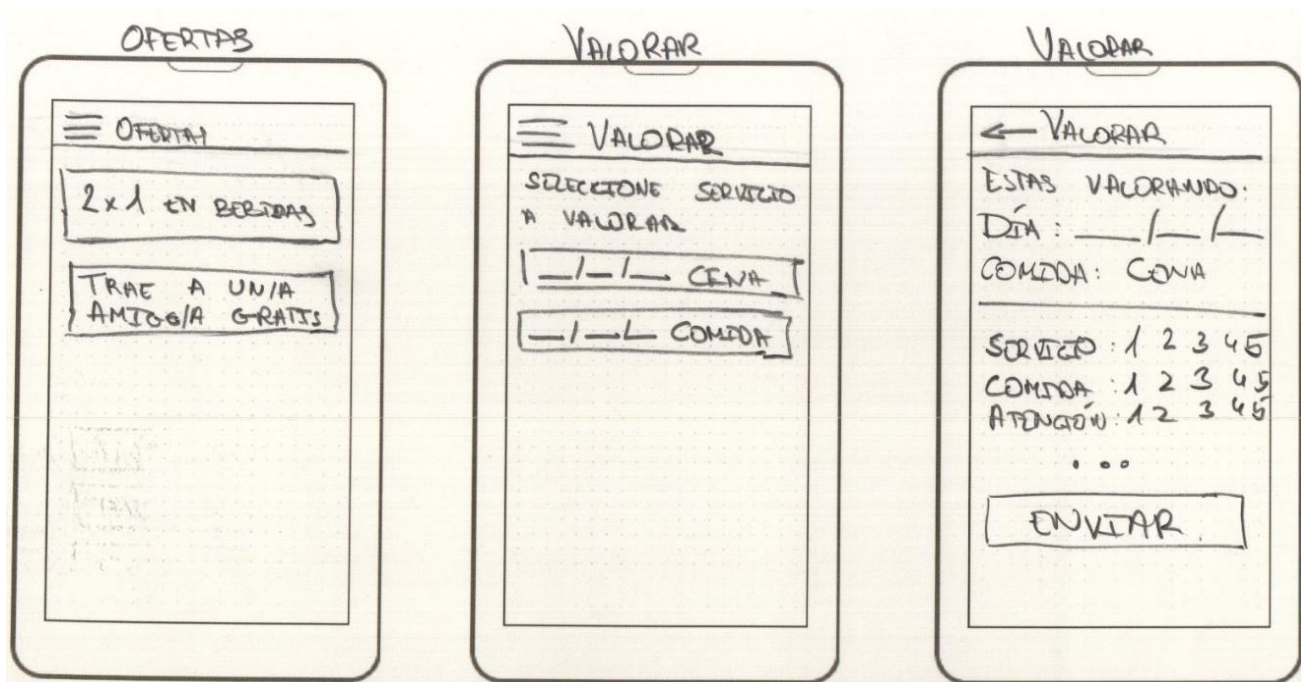


Ilustración 5- Sketches (ofertas y valorar)

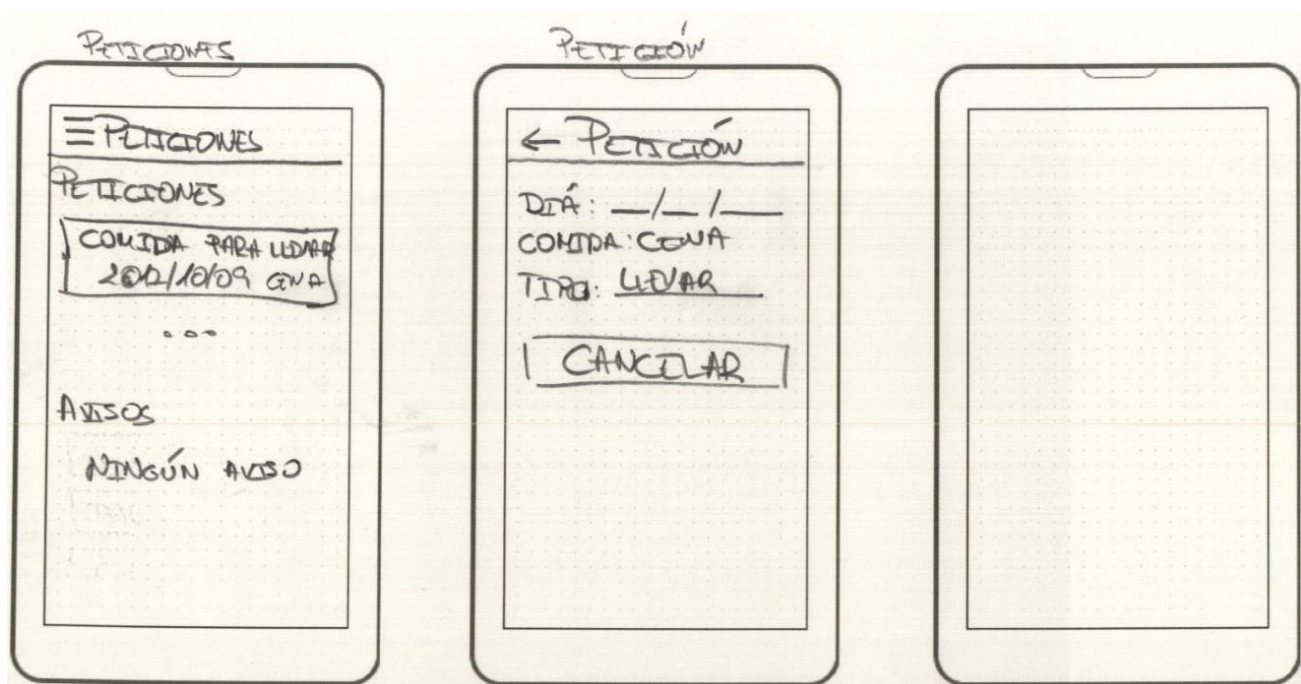


Ilustración 6- Sketches (peticiones)

2.3.2 Prototipo horizontal de alta fidelidad

Una vez hecho el prototipado a mano alzada, y después de validarlo, se ha realizado ese mismo prototipo otra vez, pero en este caso usando herramientas que permiten obtener un prototipo de alta fidelidad. A continuación, se muestran las diferentes pantallas de la aplicación.

2.3.2.1 Pantalla de identificación

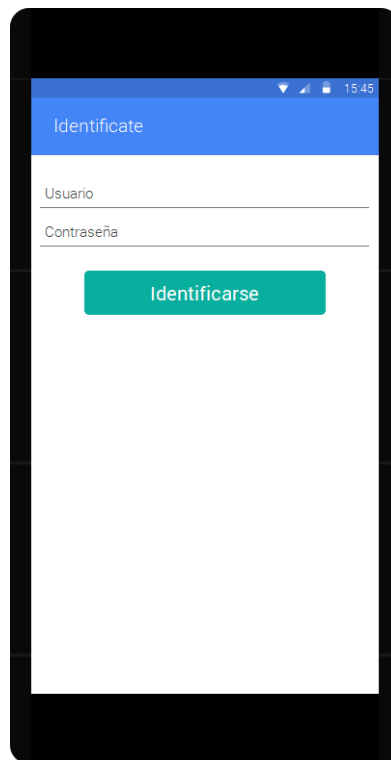


Ilustración 7- Prototipo horizontal: identificación.

2.3.2.2 Pantalla de bienvenida

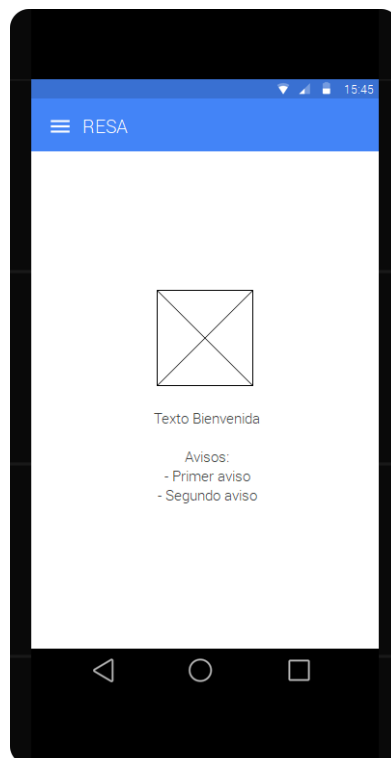


Ilustración 8- Prototipo horizontal: bienvenida.

2.3.2.3 Pantalla del menú de navegación

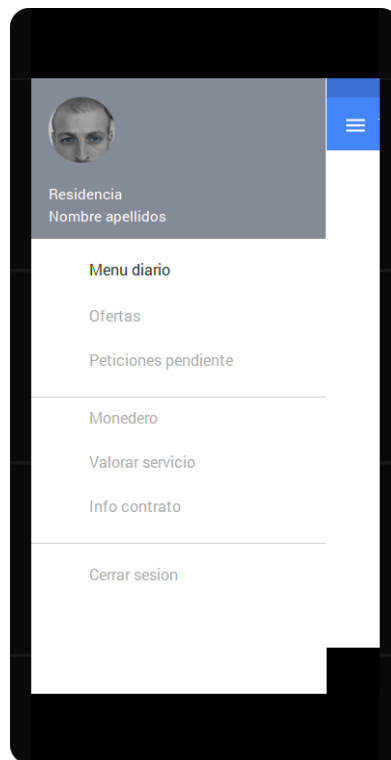


Ilustración 9- Prototipo horizontal: navegación.

2.3.2.4 Pantalla del menú de la comida por día

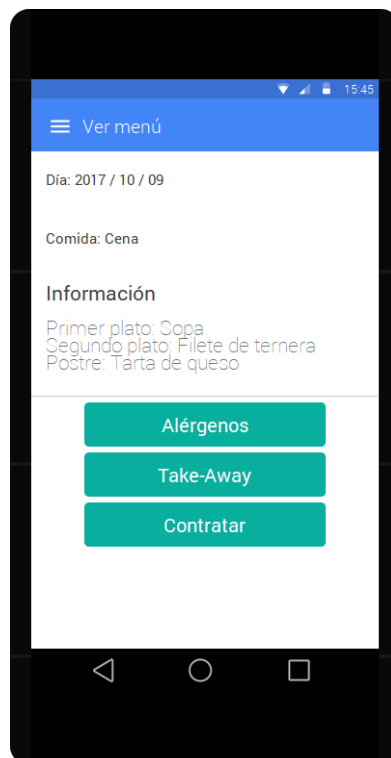


Ilustración 10- Prototipo horizontal: menú por día.

2.3.2.5 Pantalla de alérgenos

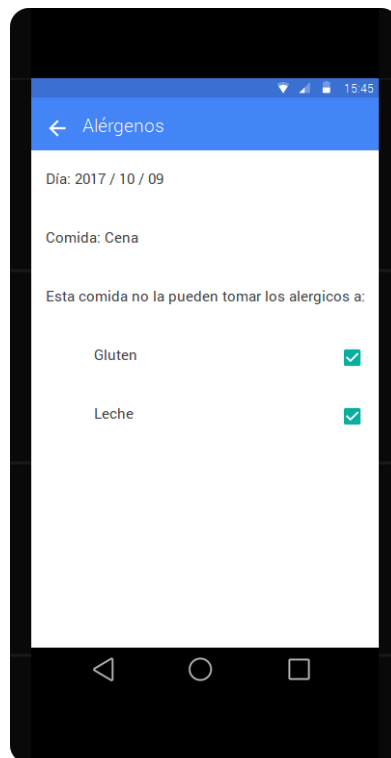


Ilustración 11- Prototipo horizontal: alérgenos.

2.3.2.6 Pantalla del monedero

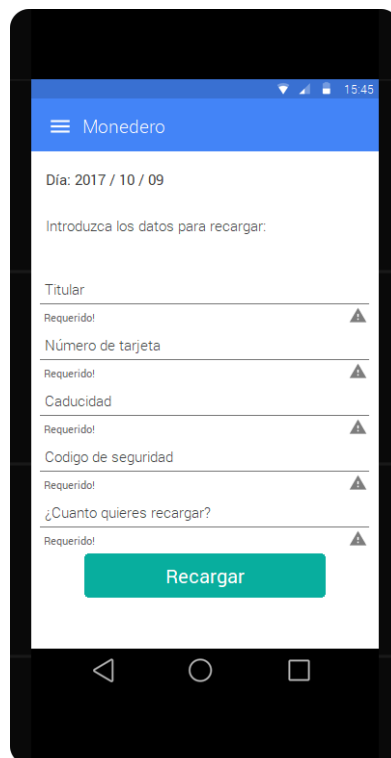


Ilustración 12- Prototipo horizontal: monedero.

2.3.2.7 Pantalla de ofertas

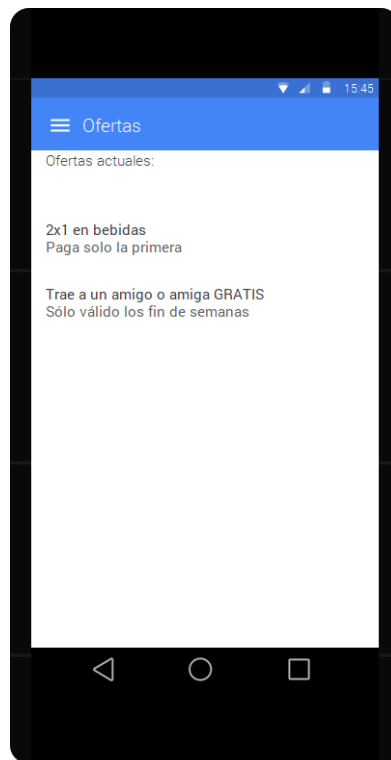


Ilustración 13- Prototipo horizontal: ofertas.

2.3.2.8 Pantallas de valoración



Ilustración 14- Prototipo horizontal: valorar I.

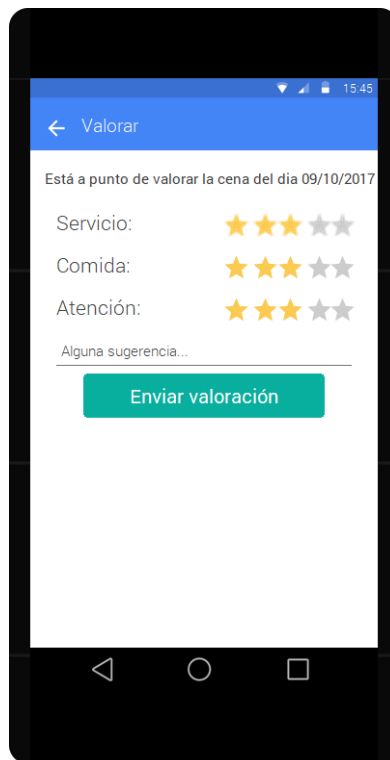


Ilustración 15- Prototipo horizontal: valorar II.

2.3.2.9 Pantallas de peticiones

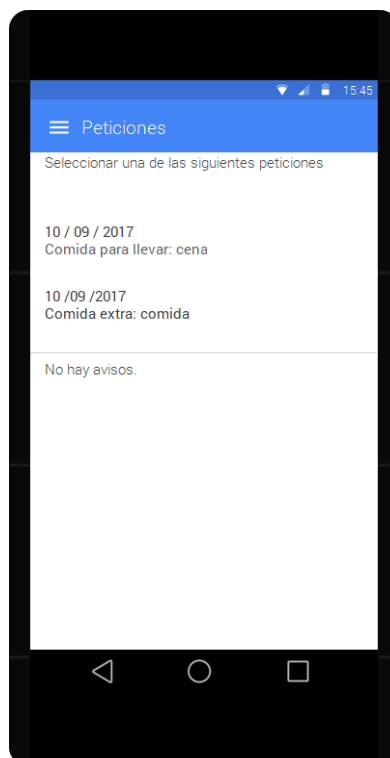


Ilustración 16- Prototipo horizontal: petición I.



Ilustración 17- Prototipo horizontal: petición II.

2.4 Cuarta fase – Evaluación

En esta fase, se ha ideado el método de evaluación para poder comprobar que el diseño obtenido es el adecuado.

2.4.1 Preguntas para obtener información

A continuación, se listan las preguntas necesarias para recopilar información sobre las personas que realizaran el test:

- (Apuntar si es hombre o mujer)
- ¿Qué edad tienes?
- ¿Qué dispositivo móvil utilizas normalmente? ¿Android o iPhone?
- ¿Dispones de un plan de datos?
- ¿Qué estás estudiando? ¿Qué curso?
- ¿Cuál es tu horario del grado?
- ¿Cuántos años llevas en la residencia?
- ¿Por qué elegiste alojarte este año en la residencia?
- ¿Qué tipo de pensión tienes contratada?
- ¿Sueles bajar solo/a o acompañado/a? Si bajas acompañado/a, ¿cómo quedas con los demás?

- ¿Has comprado alguna vez una comida extra o has invitado a otra persona al comedor?
- ¿Sueles consultar el menú antes de comer? ¿Cómo?
- ¿Tienes alguna alergia o intolerancia?
- ¿Has encargado alguna vez comida para llevar? ¿Con que frecuencia?
- ¿Has valorado el servicio alguna vez? ¿Cómo?

2.4.2 Tareas a realizar por los usuarios y usuarias y sus preguntas

2.4.2.1 Primera tarea

La persona que esté realizando el test tendrá que identificarse en la aplicación. Para ello, tendrá que introducir los siguientes datos:

- Erróneos:
 - Usuario: ejemplo@ejemplo.com
 - Contraseña: mal
- Correctos
 - Usuario: ejemplo@ejemplo.com
 - Contraseña: bien

Primero introducirá los erróneos con el objetivo de que le salga un error, y a continuación, los correctos para conseguir loguearse.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Te ha salido algún error? En caso afirmativo, ¿puedes explicarlo?
- ¿Te ha resultado fácil? ¿Has tenido alguna dificultad?
- Cuando has introducido los datos erróneos, ¿el mensaje de error se entendía?
- ¿Alguna sugerencia?

2.4.2.2 Segunda tarea

En este caso, primero hay que ver el menú de la comida del día, y, a continuación, cambiar a la cena del día siguiente. La aplicación ya está identificada.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Te ha resultado fácil? ¿Has tenido alguna dificultad?
- ¿Qué te ha parecido el diseño del menú?

- ¿Te ha resultado fácil cambiar de día y de comida?
- ¿Te ha salido algún error? En caso afirmativo, ¿puedes explicarlo?
- ¿Alguna sugerencia?

2.4.2.3 Tercera tarea

Hay que seleccionar la cena del próximo viernes y consultar la tabla de alérgenos en particular de esa comida. La aplicación ya está identificada.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Te ha resultado fácil? ¿Has tenido alguna dificultad?
- ¿Qué te ha parecido el diseño del menú?
- ¿Te ha resultado fácil cambiar de día y de comida?
- ¿Te ha salido algún error? En caso afirmativo, ¿puedes explicarlo?
- ¿Alguna sugerencia?

2.4.2.4 Cuarta tarea

Primero hay que seleccionar la comida del viernes. Una vez localizada, hay que encargar la comida para llevar (Take-Away). Cuando hayamos recibido el aviso de confirmación, hay que cancelar la petición. La aplicación ya está identificada.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Te ha resultado fácil encargar la comida para llevar? ¿Y, después, cancelarla? ¿Has tenido alguna dificultad en algún proceso?
- ¿Qué te ha parecido el proceso de encargar el take-away? ¿y el de cancelarlo?
- ¿Has estado seguro/a en todo momento de las acciones que estabas haciendo? En caso negativo, ¿podrías indicar los momentos con dudas?
- ¿Te ha salido algún error? En caso afirmativo, ¿puedes explicarlo?
- ¿Alguna sugerencia?

2.4.2.5 Quinta tarea

Primero hay que hacer una recarga en el monedero virtual y, a continuación, con el dinero virtual encargar la comida extra del lunes siguiente. La aplicación ya está identificada.

Para la recarga puedes usar los siguientes datos:

- Titular: Ejemplo Ejemplo

- Número de tarjeta: 5408 0000 0000 0001
- Fecha de caducidad: 10/2019
- Código de seguridad: 515

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Te ha resultado fácil hacer la recarga del dinero? ¿Te has encontrado con algún problema? ¿Te ha parecido seguro el proceso?
- En el caso de encargar la comida, ¿Qué te ha parecido el proceso?
- ¿Has estado seguro/a en todo momento de las acciones que estabas realizando? En caso negativo, ¿podrías indicar los momentos con dudas?
- ¿Te ha salido algún error? En caso afirmativo, ¿puedes explicarlo?
- ¿Alguna sugerencia?

2.4.2.6 Sexta tarea

Hay que hacer dos críticas sobre el servicio. La primera tiene que ser positiva y la segunda, negativa (indicando alguna sugerencia para mejorar, como, por ejemplo, que la comida tiene que estar más caliente y menos hecha). La aplicación ya está identificada.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Te ha resultado fácil hacer la valoración? ¿Te has encontrado con algún problema?
- ¿Te ha salido algún error? En caso afirmativo, ¿puedes explicarlo?
- ¿Alguna sugerencia?

2.4.2.7 Séptima tarea

Hay que encontrar las ofertas disponibles en el momento actual. La aplicación ya está identificada.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Te ha resultado útil la información de las ofertas?
- ¿Alguna sugerencia?

2.4.2.8 Octava notificación

En unos instantes se recibirá un aviso mediante notificación. La idea es que se abra y se lea. La aplicación ya está identificada.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Has experimentado algún error o dificultad?
- ¿Te ha parecido correcto el texto y la presentación?
- ¿Alguna sugerencia?

2.4.2.9 Novena notificación

La aplicación ya está identificada, y el objetivo de esta tarea es que el usuario o la usuaria cierre la sesión.

Preguntas referentes a la tarea:

- ¿Has conseguido correctamente el objetivo de la tarea?
- ¿Has experimentado algún error o dificultad?
- ¿Alguna sugerencia?

3 Casos de uso

A continuación, se describen los casos de uso que se han decidido desarrollar para después implementarlos en el proyecto.

3.1 CU-001 Identificarse

Tabla 2- Caso de uso: CU-001

Identificador	CU-001
Nombre	Identificarse
Prioridad	Alta
Descripción	El sistema tendrá que comprobar las credenciales y mostrar un error sin son incorrectas. En otro caso, pasará a la siguiente pantalla.
Actores	Usuario / Usuaría
Precondiciones	Instalada la app
Iniciado por	Usuario / Usuaría
Flujo	<ol style="list-style-type: none">1. El usuario o usuaria abre la app2. Introduce los datos de identificación3. La app los comprueba contra el back-end4. En caso de ser incorrectos, muestra error. Sino pasa a la siguiente pantalla
Postcondiciones	La app se queda identificada
Notas	El sistema puede tener un control del máximo de intentos de logueo por parte del servidor

3.2 CU-002 Cerrar sesión

Tabla 3- Caso de uso: CU-002

Identificador	CU-002
Nombre	Cerrar sesión
Prioridad	Media
Descripción	El sistema tendrá que acabar con la sesión una vez que la o el usuario lo solicite.
Actores	Usuario / Usuaría
Precondiciones	Aplicación instalada e identificada.
Iniciado por	Usuario / Usuaría
Flujo	<ol style="list-style-type: none">1. Abre la app2. Solicita “cerrar sesión”3. Se cierra la sesión
Postcondiciones	La app deja de estar identificada
Notas	-

3.3 CU-003 Ver menú

Tabla 4- Caso de uso: CU-003

Identificador	CU-003
Nombre	Ver menú

Prioridad	Alta
Descripción	El sistema deberá mostrar el menú para la hora seleccionada del día escogido
Actores	Usuario / Usuaría
Precondiciones	Aplicación instalada e identificada.
Iniciado por	Usuario / Usuaría
Flujo	<ol style="list-style-type: none"> 1. Abre la app 2. Selecciona “ver menú” 3. Escoge que comida y el día 4. Recupera la información desde el back-end 5. Muestra la información
Postcondiciones	El usuario o la usuaria sabe el menú de esa comida en concreto.
Notas	Si la información no está disponible, muestra un mensaje.

3.4 CU-004 Ver tabla alérgenos

Tabla 5- Caso de uso: CU-004

Identificador	CU-004
Nombre	Ver tabla alérgenos
Prioridad	Baja
Descripción	El sistema deberá mostrar la tabla de alérgenos sobre el menú para la hora seleccionada del día escogido
Actores	Usuario / Usuaría
Precondiciones	Aplicación instalada e identificada.
Iniciado por	Usuario / Usuaría
Flujo	<ol style="list-style-type: none"> 1. Abre la app 2. Selecciona “ver menú” 3. Escoge una de las tres comidas de un día en particular 4. Recupera la información desde el back-end 5. Muestra la información 6. Selecciona “ver tabla de alérgenos” 7. Muestra la información sobre alérgenos
Postcondiciones	El usuario o la usuaria sabe los alérgenos del menú de esa comida en concreto
Notas	Si la información no está disponible, muestra un mensaje.

3.5 CU-005 Encargar comida para llevar

Tabla 6- Caso de uso: CU-005

Identificador	CU-005
Nombre	Encargar comida para llevar
Prioridad	Media
Descripción	El sistema deberá encargar el take-away de una de las tres comidas de un día en particular.
Actores	Usuario / Usuaría
Precondiciones	Aplicación instalada e identificada. Tener contratada esa comida.
Iniciado por	Usuario / Usuaría

Flujo	<ol style="list-style-type: none"> 1. Abre la app 2. Selecciona “ver menú” 3. Escoge una de las tres comidas de un día en particular 4. Recupera la información desde el back-end 5. Muestra la información 6. Selecciona “Encargar take-away” 7. Solicita al back-end la petición 8. Muestra la respuesta
Postcondiciones	El usuario o la usuaria ha encargado una comida para llevar y se le ha notificado
Notas	Si no se ha podido encargar la comida para llevar, muestra un aviso con el error.

3.6 CU-006 Cancelar comida para llevar

Tabla 7- Caso de uso: CU-006

Identificador	CU-006
Nombre	Cancelar comida para llevar
Prioridad	Baja
Descripción	El sistema deberá cancelar la petición de take-away de una de las tres comidas de un día en particular.
Actores	Usuario / Usuaria
Precondiciones	Aplicación instalada e identificada. Tener contratada esa comida y haber pedido la comida para llevar.
Iniciado por	Usuario / Usuaria
Flujo	<ol style="list-style-type: none"> 1. Abre la app 2. Selecciona “ver peticiones” 3. Escoge la petición a cancelar 4. Selecciona “cancelar” 5. Hace la petición al back-end 6. Muestra la respuesta
Postcondiciones	El usuario o la usuaria ha cancelado la comida para llevar y se le ha notificado
Notas	Si no se ha podido cancelar la comida para llevar, muestra un aviso con el error.

3.7 CU-007 Ver ofertas

Tabla 8- Caso de uso: CU-007

Identificador	CU-007
Nombre	Ver ofertas
Prioridad	Baja
Descripción	El sistema deberá mostrar las ofertas disponibles en ese momento
Actores	Usuario / Usuaria
Precondiciones	Aplicación instalada e identificada.
Iniciado por	Usuario / Usuaria
Flujo	<ol style="list-style-type: none"> 1. Abre la app

	<ol style="list-style-type: none"> 2. Selecciona “ver ofertas” 3. Recupera la información desde el back-end 4. Muestra la información
Postcondiciones	El usuario o la usuaria sabe las ofertas disponibles en el momento de la petición.
Notas	-

3.8 CU-008 Valorar servicio

Tabla 9- Caso de uso: CU-008

Identificador	CU-008
Nombre	Valorar servicio
Prioridad	Media
Descripción	El sistema deberá mandar la valoración del servicio
Actores	Usuario / Usuaria
Precondiciones	Aplicación instalada e identificada. Haber realizado alguna comida en el comedor.
Iniciado por	Usuario / Usuaria
Flujo	<ol style="list-style-type: none"> 1. Abre la app 2. Selecciona “Valorar servicio” 3. Recupera el listado de comidas a valorar desde el back-end 4. Muestra la información 5. Selecciona la comida a valorar 6. El usuario o la usuaria valora el servicio 7. Manda la información al back-end 8. Muestra la respuesta
Postcondiciones	El usuario o la usuaria ha mandado la valoración sobre el servicio y la comida.
Notas	-

3.9 CU-009 Recargar monedero virtual

Tabla 10- Caso de uso: CU-009

Identificador	CU-009
Nombre	Recargar monedero virtual
Prioridad	Alta
Descripción	El sistema deberá recargar el monedero del usuario o usuaria
Actores	Usuario / Usuaria
Precondiciones	Aplicación instalada e identificada.
Iniciado por	Usuario / Usuaria
Flujo	<ol style="list-style-type: none"> 1. Abre la app 2. Selecciona “Monedero” 3. Selecciona recargar 4. Introduce la información sobre el pago 5. Manda la información al back-end y hace la solicitud de pago 6. Muestra la respuesta
Postcondiciones	El usuario o la usuaria ha incrementado el dinero en el monedero virtual.

Notas	Si no se ha podido hacer correctamente el pago, recibe un aviso.
--------------	------------------------------------------------------------------

3.10 CU-010 Encargar comida extra

Tabla 11- Caso de uso: CU-010

Identificador	CU-010
Nombre	Encargar comida extra
Prioridad	Media
Descripción	El sistema deberá comprar la comida del día seleccionado.
Actores	Usuario / Usuaría
Precondiciones	Aplicación instalada e identificada. La comida que se quiere contratar, sin haberla comprado anteriormente. Dinero disponible en el monedero virtual.
Iniciado por	Usuario / Usuaría
Flujo	<ol style="list-style-type: none"> 1. Abre la app 2. Selecciona “ver menú” 3. Escoge una de las tres comidas de un día en particular 4. Recupera la información desde el back-end 5. Muestra la información 6. Selecciona “Contratar” 7. Solicita al back-end la petición 8. Muestra la respuesta
Postcondiciones	El usuario o la usuaria ha comprado la comida extra
Notas	-

3.11 CU-011 Mostrar notificación

Tabla 12- Caso de uso: CU-011

Identificador	CU-011
Nombre	Mostrar notificación
Prioridad	Alta
Descripción	El sistema tiene que mostrar el aviso dado desde el back-end
Actores	Usuario / usuaria
Precondiciones	Aplicación instalada e identificada.
Iniciado por	Back-end
Flujo	<ol style="list-style-type: none"> 1. La aplicación recibe la notificación 2. La aplicación muestra la notificación 3. El usuario o usuaria la selecciona 4. Se abre la app automáticamente 5. Se muestra el aviso
Postcondiciones	El usuario ha recibido la notificación correctamente.
Notas	-

4 Implementación en Android

A continuación, se describen varios aspectos de la implementación de la aplicación, desde las herramientas utilizadas, hasta el resultado obtenido.

4.1 Herramientas utilizadas

La principal herramienta utilizada para el desarrollo de la aplicación ha sido Android Studio. Éste es el IDE oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además de ser un potente editor de códigos y tener las herramientas para desarrolladores de IntelliJ, Android Studio (4) ofrece aún funciones como las siguientes:

- Un sistema de compilación basado en Gradle flexible
- Un emulador rápido con varias funciones
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android
- Instant Run para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código
- Gran cantidad de herramientas y frameworks de prueba
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++.
- Soporte incorporado para Google Cloud Platform

Para el control de versiones, se ha utilizado la herramienta GIT, tanto en su versión de Windows, como en la versión de Ubuntu.

4.2 Back-end

Por unos problemas ajenos a este proyecto, no se ha podido usar el back-end existente. Esto ha llevado a tener que buscar una solución.

En este caso, se ha optado por utilizar una base de datos en tiempo real que proporciona Firebase. Esta herramienta ha permitido seguir el mismo esquema de base de datos del back-end existente, dando pie, de este modo, a que la aplicación pudiese mantener la misma arquitectura. De esta forma, en un futuro no habría que cambiarla para conectarla al back-end inicial.

Por otro lado, al no contemplarse este cambio en el alcance del TFM, se decidió mantener el cambio de back-end como un proyecto paralelo. Con el objetivo de poder

completar a tiempo el proyecto, se decidió no hacer una API intermedia, sino leer y escribir directamente en la BD.

En cuanto a la seguridad y control del esquema de la base de datos, se ha utilizado las reglas que ofrece la base de datos de Firebase con el fin de mantener la seguridad de los datos.

4.3 Estructura del proyecto

En las próximas líneas se explica la estructura del proyecto.

4.3.1 Activity y fragments

La aplicación contiene únicamente dos actividades: la de identificación de usuarios (LoginActivity) y la principal (MainActivity), que sólo se puede acceder a ella si el usuario o la usuaria está identificada. El layout de esta actividad contiene una vista de fragmento que va cambiando dependiendo del fragmento a ejecutar.

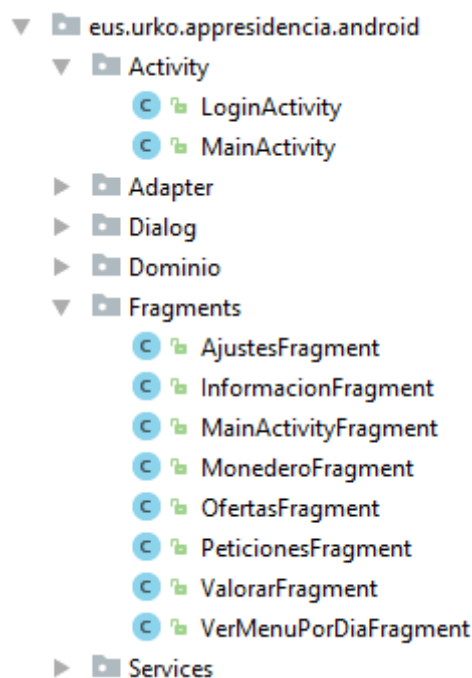


Ilustración 18- Actividades y fragmentos de la aplicación.

En cuanto a los fragmentos:

- Ajustes: Permite modificar la residencia y curso preferido.
- Información: Muestra información sobre la residencia.
- Main: Muestra el logo de bienvenida al abrir la aplicación.
- Monedero: Permite saber el dinero disponible y recargar la cuenta virtual.
- Ofertas: Permite conocer las ofertas disponibles en la residencia.
- Peticiones: Muestra las peticiones de take-away hechas, y al mismo tiempo permite cancelarlas.

- Valorar: Permite valorar el servicio prestado.
- Ver menú por día: Permite ver el menú por día, junto a los alérgenos, y hacer la petición de comida para llevar.

4.3.2 Dominio

Se han generado las clases necesarias para satisfacer todos los elementos de la base de datos.



Ilustración 19- Dominio de la aplicación

En cuanto a la clase CargarDominio, ésta permite añadir a la base de datos “de forma automática” unas clases de prueba (ejemplos).

4.3.3 Servicios

Se ha creado un servicio (MensajeService) con el objetivo de poder recibir los avisos que se dan desde recepción.

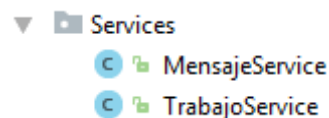


Ilustración 20- Servicios de la implementación

4.3.4 Otros archivos

En el proyecto se encuentran diferentes clases o archivos que componen la aplicación. Estos pueden ser adaptadores, layouts, imágenes, diálogos, etc.

4.4 Traducciones

Gracias a como están pensados los proyectos de Android, estos permiten hacer las traducciones de la aplicación de una manera muy sencilla (5). Para ello, se han almacenado todas las palabras y frases de la interfaz en el archivo “string.xml”, dentro de la carpeta “values” de los recursos.

Esto posibilita, dado el público de la aplicación, ésta se pueda traducir de una manera muy sencilla, haciéndola disponible en los idiomas oficiales del estado, entre otros.

4.5 Resultado

En las próximas imágenes, se pueden ver diferentes capturas de la aplicación en tiempo de ejecución. De esta forma, se puede observar el resultado del proyecto: la aplicación para dispositivos móviles.



Ilustración 21- Pantalla identificación

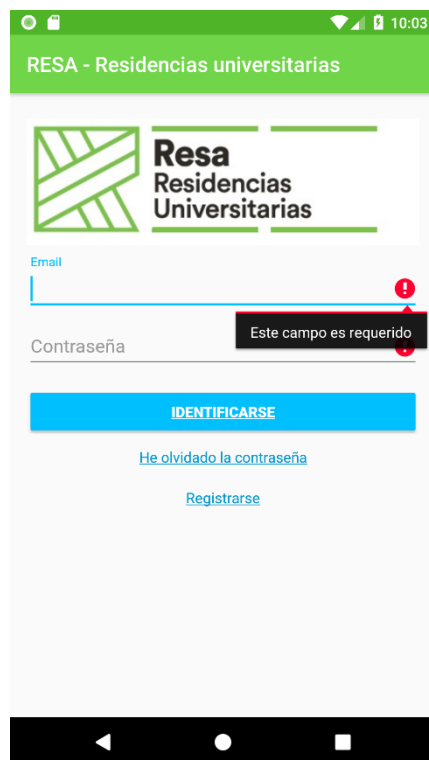


Ilustración 22- Pantalla de identificación con error



Ilustración 23- Pantalla principal

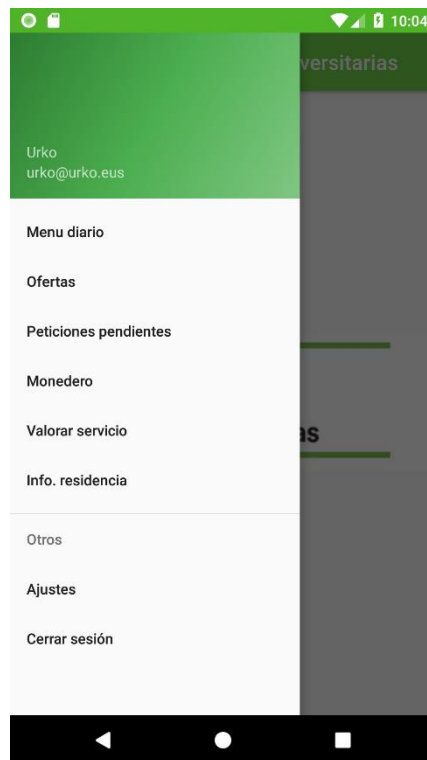


Ilustración 24- Pantalla de navegación



Ilustración 25- Pantalla del menú diario

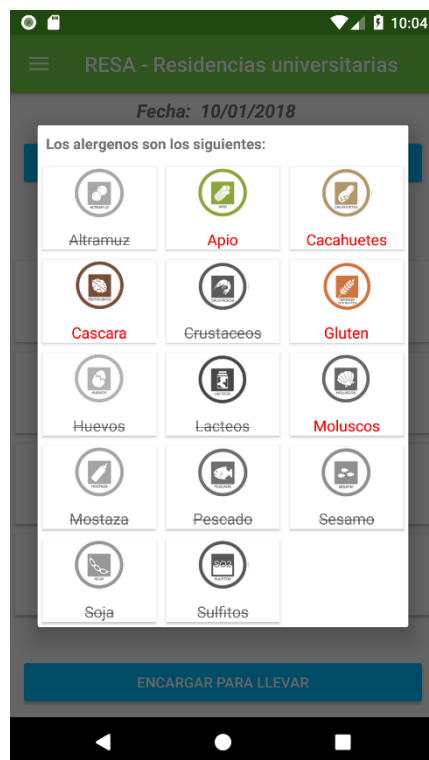


Ilustración 26- Pantalla de alérgenos

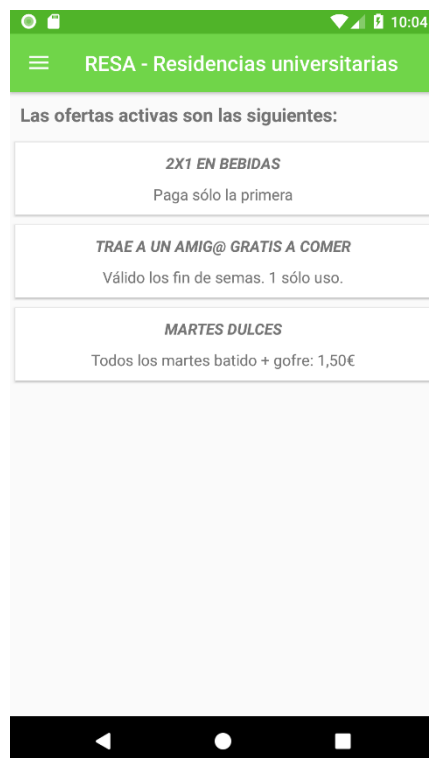


Ilustración 27- Pantalla de ofertas

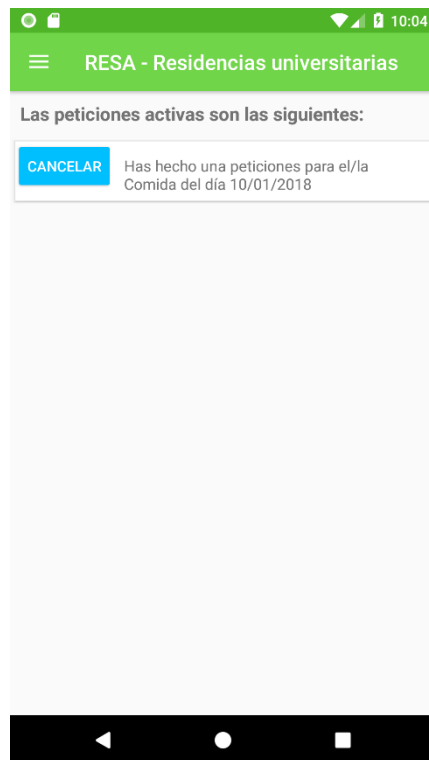


Ilustración 28- Pantalla de peticiones

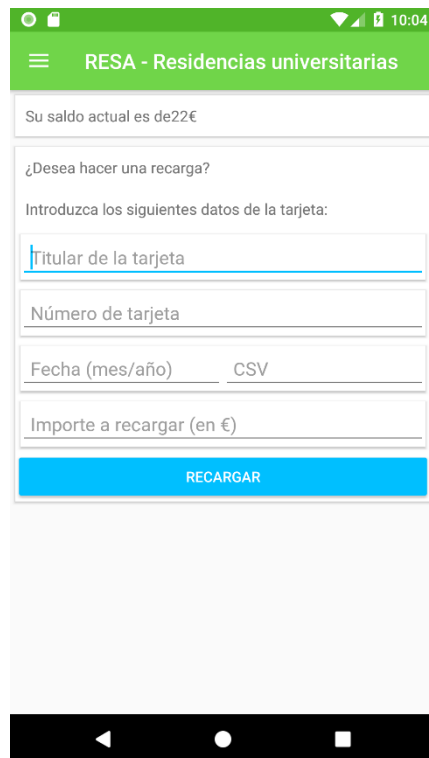


Ilustración 29- Pantalla del monedero

RESA - Residencias universitarias

¿Desea valorar el servicio?

La calidad de la comida ★★★★★

La calidad del menú ★★★★★

La calidad del servicio ★★★★★

Comentario

ENVIAR

Ilustración 30- Pantalla de valoraciones

RESA - Residencias universitarias

A continuación tienes la información sobre la residencia.

Manuel Agud Querol

residentes.maq@resa.es

Paseo de Berio 9, Donostia (20018)

943563007

<https://www.resa.es/es/residencias/san-sebastian/residencia-universitaria-manuel-agud-querol/residencia/>

Ilustración 31- Pantalla de información de la residencia

RESA - Residencias universitarias

Selecciona la residencia y curso preferido:

Selecciona la residencia:

Manuel Agud Querol

Seleccione el curso academico

2016-2017

SELECCIONAR

Ilustración 32- Pantalla de ajustes

5 Control de versiones: Git

Hoy en día, casi todos los proyectos de software son realizados por varias personas a la vez, y en muchas ocasiones, esas personas no se encuentran en el mismo sitio trabajando. Esto, no sólo ocurre en el ámbito empresarial, también sucede, por ejemplo, en el software libre desarrollado por internautas.

Además, cuando el proyecto toma una envergadura significativa, es preciso tener un control del código, sobre todo cuando existen diferentes versiones de éste (desarrollo, producción, pruebas, etc.). A lo largo de la última década, se han ido desarrollando diferentes productos de software, llamados “control de versiones”, para solventar los problemas anteriores. Entre todos ellos, destaca Git, gracias a su facilidad de uso y su gran popularidad, tanto en el sector empresarial, como en el sector del desarrollo libre.

En este capítulo, se explican las características básicas y su funcionamiento. Por último, se detalla el uso que se le ha dado en la implementación de este proyecto.

5.1 Características básicas de Git

A continuación, se explica la instalación de esta herramienta en Ubuntu y los comandos básicos para empezar a utilizar Git (6). Aunque la instalación sólo se describa para el sistema operativo Ubuntu (que ha sido el sistema usado en este proyecto, a excepción de la realización de la memoria), existen versiones compatibles con Windows (7) y Mac OS (8). Eso sí, los comandos serán los mismos en todos los sistemas operativos.

5.1.1 Instalación de Git en Ubuntu

Para instalar Git en Ubuntu, sólo hay que ejecutar el siguiente comando desde la línea de comandos, ya que el repositorio que contiene Git, ya está incluido por defecto en este sistema operativo. Al completar la instalación, la herramienta ya estará lista para su funcionamiento.

```
sudo apt-get install Git
```

5.1.2 Primeros pasos con Git

Para poder explicar mejor el funcionamiento de Git, se supondrá que existe una carpeta llamada “Project”, la cual contiene un proyecto de Android. El código fuente de este proyecto, es al que se le quiere aplicar un control de versiones.

El primer paso, es inicializar un repositorio local dentro de la carpeta “Project”. Para ello, hay que abrir una nueva línea de comandos y cambiar la ruta de ésta, a la de la carpeta del proyecto. Después, se ejecuta el siguiente comando para inicializar el repositorio:

```
git init
```

De esta forma, se habrá creado un repositorio local dentro del proyecto.

En las siguientes páginas se explican los comandos más básicos de Git, suponiendo que sólo se está trabajando con una rama (copias del código fuente para diferentes usos: desarrollo, producción, etc.). Los comandos se tienen que ejecutar en la carpeta del repositorio que se ha inicializado. En este caso, dentro de la carpeta del proyecto.

5.1.2.1 Añadir un nuevo repositorio

Por defecto, un repositorio de Git, aunque esté dentro de la carpeta del proyecto, no lleva seguimiento de ningún archivo. Por esto, hay que indicarle a Git cuales son los archivos del proyecto de los que tiene que llevar control. Para ello, existe el siguiente comando que añade un archivo específico a una carpeta al repositorio:

```
git add <ruta del archivo o carpeta>
```

5.1.2.2 Confirmar los cambios realizados

Cada vez que se realiza una modificación en algún archivo, Git no guarda esas modificaciones hasta que se le indique lo contrario. Por ello, hay que añadir de nuevo el o los archivos modificados al repositorio mediante el comando indicado en la sub-sección anterior. Por último, hay que ejecutar el comando “commit” con el fin de guardar una versión con los cambios realizados.

```
git commit
```

Los dos pasos indicados anteriormente, se pueden simplificar con un solo comando. Hay que tener en cuenta, que de este modo se guardarán todos los archivos que se han añadido anteriormente al repositorio y que en algún momento se hayan modificado.

```
git commit -a
```

En los dos casos, al ejecutar el comando “commit”, pedirá que se escriba un mensaje indicando los cambios en la versión.

5.1.2.3 Revertir los cambios guardados

En el caso de que se quiera revertir los cambios realizados en el último “commit” o anteriores, simplemente hay que ejecutar el siguiente comando:

```
git revert <commit>
```

Siendo <commit> el número único de la versión a la que se quiere volver. Para saber ese número, hay que ejecutar el siguiente comando, el cual mostrará la siguiente información de cada “commit”:

- Número único del “commit”
- Autor
- Fecha
- Mensaje

git log

5.1.2.4 Estado del repositorio

El comando que se especifica a continuación, muestra el estado actual del repositorio, indicando los archivos que no llevan seguimiento y los que no se han confirmado.

git status

5.1.2.5 Añadir y listar los repositorios remotos

Una vez inicializado el repositorio localmente, se puede añadir uno (o más) repositorios remotos para poder tenerlo “sincronizado” en otro PC/servidor/servicio. En las siguientes líneas, se explica la forma de hacerlo con servicios (plataformas) online ofertados gratuitamente.

Para ello, lo primero que hay que hacer es inicializar un nuevo repositorio en la plataforma que se haya escogido (GitHub, BitBucket etc.). Dicha plataforma permitirá obtener una URL (o algo equivalente) con el fin de poder acceder localmente al repositorio. Ahora, hay que añadir el repositorio remoto mediante el siguiente comando, siendo <remoteRepositoryURL> la URL indicada anteriormente:

git remote add origin <remoteRepositoryURL>

Si se quiere ver los repositorios remotos que existen, se ejecutará el siguiente comando:

git remote -v

5.1.2.6 Interactuando con el repositorio remoto

En las siguientes dos páginas, se explican los comandos básicos para interactuar con un repositorio remoto (9).

- Push: Este comando sirve para sincronizar los cambios guardados en el repositorio local, con el repositorio remoto.

```
git push <NombreRepositorioREMOTO> <NombreRAMA>
```

- Fetch: Este comando obtiene todos los cambios del repositorio remoto y los descarga al repositorio local. Todos estos cambios no se combinan con el repositorio local, sino que se guardan en la rama de rastreo remoto.

```
git fetch <NombreRepositorioREMOTO>
```

- Merge: Al ejecutarlo, combina los cambios locales con los cambios del repositorio remoto. Previamente, se habrá tenido que ejecutar el comando fetch.

```
git merge <NombreRepositorioREMOTO>/<NombreRAMA>
```

- Pull: Este comando combina los dos anteriores (fetch y merge), es decir, descarga los cambios que hay en el repositorio remoto y los combina con los locales.

```
git pull <NombreRepositorioREMOTO> <NombreRAMA>
```

- Clone: En el caso de que queramos clonar un repositorio remoto en nuestro ordenador o servidor, existe un comando de Git que aglutina varios pasos, evitando tener que ejecutar varios comandos:

```
git clone remoteRepositoryURL
```

El resultado de la ejecución de dicho comando es el siguiente:

- Crea una nueva carpeta, llamada como el repositorio, donde se encuentra actualmente la línea de comandos.

- Inicializa el repositorio local Git.
- Añade el repositorio a la lista de repositorios remotos con el nombre “origin”.
- Todos los archivos del repositorio remoto se descargan al local.
- La rama por defecto se desprotege.

5.2 Uso de Git en el desarrollo del software en el proyecto

En esta sección se explican qué pasos se han dado para el uso de Git en el entorno de desarrollo, y, a continuación, los motivos de la elección de BitBucket como plataforma online de Git en el proyecto.

5.2.1 Git en el entorno de desarrollo

A la hora de desarrollar el proyecto de Android se han generado varias versiones de código. Todas estas versiones han sido controladas mediante el software Git.

Para poder explicar el uso de Git en el proyecto, hay que mencionar que se han utilizado dos ordenadores para el desarrollo del proyecto. El principal, ha sido el que se encontraba en Donostia (ciudad donde paso la mayor parte del tiempo), y el otro, ha sido un portátil, que se ha utilizado en realización de tareas fuera de la residencia (en viajes, vacaciones, etc.). Debido a esto, se ha necesitado un repositorio remoto para poder tener los dos ordenadores sincronizados. Este, ha estado alojado en BitBucket, como se versa en la siguiente sección.

Los pasos seguidos para montar la infraestructura fueron los siguientes:

- Ordenador Donostia
 1. Crear un nuevo proyecto de Android.
 2. Hacer alguna modificación.
 3. Inicializar el repositorio local dentro de la carpeta del proyecto en el ordenador, mediante el comando “init”.
 4. Añadir, mediante el comando “add”, los archivos que se necesiten agregar al repositorio.
 5. Guardar los cambios (comando “commit”).
 6. Crear un nuevo repositorio en BitBucket.
 7. Añadir (“Git remote add”) el repositorio de BitBucket como repositorio remoto en el local.
 8. Sincronizar los cambios del repositorio local con el remoto, mediante el comando “push”.
- Portátil
 1. Crear un nuevo proyecto de Android.

2. Clonar el repositorio de BitBucket mediante el comando “git clone”.

La mecánica de trabajo seguida ha sido muy simple. Cuando se empezaba a trabajar en un ordenador, se ejecutaba siempre el comando “pull”, para descargar las modificaciones (si las hubiese). Si se hacían progresos, y se querían guardar, los archivos modificados se añadían al seguimiento del repositorio (“git add”) y se guardaban, haciendo un “commit”. Posteriormente, se sincronizaba con el repositorio remoto, mediante el comando “push”. De esta forma, los cambios siempre estaban sincronizados con el repositorio remoto, y se descargaban localmente al empezar a trabajar.

Por último, hay que mencionar que en el repositorio sólo se guardaban los archivos del proyecto, y no todo el proyecto (genéricos para todos los proyectos). Para que fuese más fácil la tarea de añadir únicamente los archivos necesarios al seguimiento del repositorio, se creó el archivo “gitignore”. Éste sirve para indicarle a Git de que archivos no tiene que hacer seguimiento.

5.2.2 Elección de soporte de Git en la nube (servicio escogido)

Hoy en día GitHub es la plataforma más extendida para alojar proyectos Git. Su versión gratuita permite tener repositorios ilimitados públicos, es decir, que cualquier persona puede verlos y contribuir de algún modo. Pero en muchos proyectos tanto de empresa, como de otro tipo, son necesarios repositorios privados. Debido a esto, GitHub tiene versión de pago, la cual está enfocada a empresas.

Aun así, hay muchas otras plataformas que, aunque no tengan la popularidad de GitHub, efectúan las mismas funciones (o incluso más) y además permiten tener gratuitamente repositorios privados. Este es el caso de Bit-Bucket, la plataforma escogida en este proyecto.

6 Gestión del proyecto

Como se ha detallado en el capítulo de introducción, el proyecto ha tenido una planificación dada por las fechas de los diferentes entregables del trabajo. Aun así, durante el transcurso de la realización del proyecto, se ha ido escribiendo un diario, el cual se describe a continuación. Junto a la explicación del diario, se puede encontrar la gestión del alcance, del tiempo y de los costes.

6.1 Diario

Según se iba desarrollado el proyecto, se iba escribiendo un diario, en el cual se detallaba día a día las tareas realizadas, junto con el coste de tiempo que habían supuesto.

También se han ido adjuntando enlaces a páginas web, así como, los títulos de libros que se utilizaban. Asimismo, muchas de las tareas se han explicado de forma similar a la de un manual.

Se ha usado un sistema de “tags”, con el objetivo de clasificar y posteriormente, poder encontrar de forma más sencilla las diferentes tareas.

El motivo de escribir el diario no ha sido otro, que llevar el control de las tareas realizadas cada día, para tenerlas guardadas y que fuese una herramienta útil a la hora de confeccionar este documento.

6.2 Gestión del alcance

Tal y como se ha mencionado en las secciones anteriores, el alcance del proyecto ha ido ampliándose a medida que se iban logrando los objetivos marcados. Dicho alcance, ha quedado dividido en tres fases.

La primera (PEC 1), se centró en explicar detalladamente la idea del proyecto, así como hacer una planificación teniendo en cuenta los hitos más importantes. La segunda fase (PEC 2), se caracterizó por hacer el diseño centrado en el usuario, junto a detallar todos los casos de uso. Y, por último, la tercera fase en implementar la propia aplicación (PEC 3 + Final).

Así mismo, y paralelamente, se han escrito los documentos adjuntados al proyecto, incluido éste.

6.3 Gestión del tiempo

Seguidamente se detallan los hitos más importantes que ha tenido el proyecto:

- Comienzo del proyecto:
- Definición de la idea: 27/09/2017
- Primera fase: 20/09/2017 – 04/10/2017
- Segunda fase: 05/10/2017 – 01/11/2017
- Tercera fase: 02/11/2017 – 10/01/2018

- Entregable intermedio en la tercera fase: 13/12/2017
- Depósito de la versión final de la memoria: 10/01/2018

6.4 Gestión de costes

El coste económico que ha supuesto el proyecto, es totalmente despreciable en comparación con el tiempo que ha supuesto el volumen de trabajo del proyecto. Debido a esto, se excluyen de esta memoria los costes de carácter económico.

En la siguiente tabla, se puede apreciar el coste de tiempo que ha supuesto, dividido en paquetes de trabajo. Hay que tener en cuenta, que gran parte del tiempo dedicado al proyecto ha sido académico (lectura de manuales, pruebas, etc.). Horas redondeadas.

Tabla 13- Gestión de costes (en horas)

Paquete de trabajo	Actividad / Tema	Dedicación (h)
Gestión del proyecto	Alcance, tiempo y costes	10
	Diario	20
	Riesgos	30
	Planificación y comunicación	10
Implementación	DCU	40
	App Android	120
Tecnología a desarrollar (parte académica)	Android	30
	Git	20
Trabajo a desarrollar	Memoria	60
	Manuales, presentación y defensa	10

7 Conclusiones

En este proyecto, se ha alcanzado el objetivo de implementar una aplicación para Android usando para ello los métodos del diseño centrado en el usuario o la usuaria. Es verdad, que no se ha podido conectar la aplicación con el back-end existente, por problemas que no han tenido que ver con el alcance del proyecto (problemas externos). Aun así, se han solucionado esos pequeños inconvenientes con la ayuda del uso de una base de datos en tiempo real de Firebase, no contemplada en el alcance del proyecto.

La estructura de la nueva base de datos ha sido muy similar, por no decir idéntica, a la del back-end existente. Eso sí, por no complicar en exceso la solución al problema, se ha tenido que tomar la decisión de no hacer una API intermedia. De todos modos, para que la aplicación contase con una estructura de datos y seguridad, se han implementado las reglas de la base de datos. Estas controlan quien, y como, puede escribir en ésta, y en que árboles. A pesar de todo, se decidió no añadir este paso al alcance del proyecto, ya que en un inicio no estaba contemplado, es decir, que aunque se haya nombrado en este documento, se puede entender como que la base de datos ha sido un proyecto paralelo externo al TFM.

Por lo demás, la aplicación ha seguido los pasos marcados tanto en la planificación, como en el alcance. Respecto a la primera, ésta se ha tenido que modificar en algunos aspectos para poder tener en cuenta el problema anteriormente planteado. El hecho de que el alcance se haya mantenido, sobre todo en lo que respecta a la interfaz y modo de uso, posibilita que en un futuro se cambie la conexión al back-end de una manera muy sencilla: sólo habría que cambiar las conexiones al back-end, y en todo caso, añadir alguna opción, como errores, que ahora mismo no está contemplados.

En general, se han conseguido todos los objetivos del proyecto, quedando una aplicación que, con las limitaciones del back-end actual, podría ser totalmente funcional de cara a los y las jóvenes residentes.

Por otra parte, creo que es muy importante recalcar la importancia que tiene Internet en este tipo de proyectos. Hoy en día, mediante una simple búsqueda en el ciberespacio se pueden encontrar soluciones a errores de una manera muy rápida, y sin perder demasiado tiempo. Aparte, hay que mencionar todos esos tutoriales, manuales, etc. que hay alojados en diferentes sitios. Eso sí, debido a la velocidad con la que cambian todas las tecnologías, muchas soluciones se han podido quedar desfasadas, o incluso hay que tener en cuenta, que, en el momento de escribir la solución, ésta no fuese la -más-correcta.

Hay que resaltar que la ejecución de este proyecto no hubiera sido posible sin la adquisición académica obtenida durante los cuatro años del Grado, y sobre todo, en estos dos años de Máster en la UOC. Esta adquisición de conocimientos, que en algunas ocasiones se percibe como demasiado teórica o académica, en el momento de ponerla en práctica, se convierte en la herramienta que posibilita la utilización de nuevas tecnologías y el aprendizaje de ellas.

Antes de finalizar, este proyecto no ha sido solamente un factor de enriquecimiento académico y laboral, sino que también ha permitido concluir los dos años

del Máster de Ingeniería Informática cursados en la UOC de Catalunya, de una forma práctico-experimental.

Por último, agradecer a todo el equipo docente que ha hecho posible el master, y con especial ímpetu, a los que han hecho posible este Trabajo de Fin de Master.

8 Próximos pasos

A continuación, se describen algunas ideas que se pueden desarrollar como siguientes pasos, con el objetivo de mejorar y ampliar la aplicación actual.

8.1 Conectar al back-end existente

Como se ha comentado durante esta memoria, el próximo paso sería, cuando éste esté preparado, conectar la aplicación al back-end existente quedando una aplicación totalmente funcional.

8.2 Publicar la aplicación

Se podría publicar la app en las tiendas correspondientes (en este caso, Google Play) para que los y las usuarias la puedan descargar de una manera cómoda.

8.3 Implementar la aplicación para otros Smartphone

Aunque hoy en día el uso de Android es el más extendido, hay otros sistemas operativos que tienen también un amplio público, como iOS de Apple. No cabe duda, que sería una buena opción portar la aplicación a este sistema operativo.

8.4 Añadir nuevas funciones y mejorar las existentes

Aunque el alcance de la aplicación se haya delimitado, hay muchas vías de ampliación en la app. Todas ellas se pueden explorar haciendo un trabajo de campo en varias de las residencias RESA. Un ejemplo, podría ser implementar un chat o dejar la opción de organizar quedadas.

8.5 Traducciones

Hoy en día, en el Estado hay diferentes lenguas oficiales. Dado que la aplicación se puede traducir a diferentes idiomas de una manera muy sencilla, mediante el uso del fichero XML, sería una idea acertada traducirla a esos, y otros, idiomas.

9 Glosario

Activity	Es un componente de la aplicación Android que contiene una pantalla con la que los usuarios y usuarias pueden interactuar para realizar una acción.
Android	Sistema operativo basado en el núcleo Linux, diseñado principalmente para dispositivos móviles con pantalla táctil.
Android Studio	Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.
API	<i>Application Programming Interface.</i>
APK	<i>Android Application Package</i> es un paquete para el sistema operativo Android.
APP	Acortamiento del inglés <i>application</i> (aplicación).
B	Es el nombre de un lenguaje de programación desarrollado en los Bell Labs, predecesor del lenguaje de programación C.
BD	Base de datos.
BitBucket	Servicio de alojamiento basado en web, para proyectos que utilizan el sistema de control de versiones.
C	es un lenguaje de programación originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, ² como evolución del anterior lenguaje B, a su vez basado en BCPL.
C++	es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos.
Caso de uso	Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.
Firebase	Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2011 y adquirida por Google en 2014.
Framework	Estructura conceptual y tecnológica de soporte definido, de software concretos, que puede servir de base para la organización y desarrollo de software.

Git	Software de control de versiones diseñado por Linus Torvalds.
GitHub	Plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.
Google	Empresa multinacional estadounidense especializada en productos y servicios relacionados con Internet, software, dispositivos electrónicos y otras tecnologías.
Google Cloud	Es una plataforma que ha reunido todas las aplicaciones de desarrollo web que Google estaba ofreciendo por separado. Es utilizada para crear ciertos tipos de soluciones a través de la tecnología almacenada en la nube y permite por ejemplo destacar la rapidez y la escalabilidad de su infraestructura en las aplicaciones del buscador.
Gradle	Es una herramienta de automatización de la construcción del código de una aplicación, haciendo uso de las fuentes de otras herramientas.
IDE	<i>Integrated Development Enviorement</i> (Entorno de desarrollo integrado).
Instant Run	Es una función de Android Studio que reduce de manera significativa el tiempo de compilación y actualización de la aplicación que se está probando.
IntelliJ IDEA	Es un ambiente de desarrollo integrado (IDE) para el desarrollo de programas informáticos.
Internauta	Neologismo, utilizado normalmente para describir a los usuarios y las usuarias habituales de Internet.
iOS	Sistema operativo móvil de la multinacional Apple Inc.
JSON	Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos.
Lint	Es el nombre de una herramienta de programación utilizada para detectar código sospechoso, confuso o incompatible entre distintas arquitecturas.
Linus Torvalds	Es un ingeniero de software finlandés, conocido por iniciar y mantener el desarrollo del kernel Linux, basándose en el sistema operativo libre Minix creado por Andrew S. Tanenbaum y en algunas herramientas, varias utilidades y los compiladores desarrollados por el proyecto GNU.
Linux	Kernel (núcleo) del sistema del sistema operativo “GNU/Linux”
Mac OS	Nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh.

Market	Tienda
MarketPlace	Es un sitio que permite a vendedores y vendedoras, así como a compradores y compradoras relacionarse para efectuar una transacción comercial, ya sea de pago o no. Por ejemplo, Google Play.
Open Source	Expresión con la que se conoce al software o hardware distribuido y desarrollado libremente.
PEC	Prueba de Evaluación Continua.
Smartphone	Teléfono móvil inteligente.
TFG	Trabajo fin de grado.
TFM	Trabajo fin de máster.
Ubuntu	Es un sistema operativo basado en GNU/Linux que se distribuye como software libre.
URL	Localizador de recursos uniforme, formados por una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que designa recursos en una red.
Web services	Servicios Web.
Windows	Nombre de una familia de distribuciones de software para PC, smartphone, servidores y sistemas empujados, desarrollados y vendidos por Microsoft.
Windows Phone	Sistema operativo móvil desarrollado por Microsoft.
XML	Lenguaje de marcas, desarrollado por el World Wide Web Consortium (W3C), utilizado para almacenar datos de forma legible.

Algunos de los términos anteriores han sido definidos gracias a la enciclopedia libre Wikipidea.

10 Bibliografía

1. RESA . *Sobre RESA*. [En línea] <https://www.resa.es/es/sobre-resa/>.
2. Android. *Developers*. [En línea] <https://developer.android.com/about/dashboards/index.html>.
3. usability.gov. *Improving the User Experience*. [En línea] <https://www.usability.gov/what-and-why/user-centered-design.html>.
4. Android Developer. *Conoce Android Studio*. [En línea] <https://developer.android.com/studio/intro/index.html?hl=es-419>.
5. Android Developer. *Diseña tu app con una traducción en mente para mejorar la localización*. [En línea] <https://developer.android.com/studio/intro/index.html?hl=es-419>.
6. Straub, Scott Chacon y Ben. *Pro Git book*. s.l. : Apress, 2014. 978-1-484-20077-3.
7. github.io. *Git for Windows*. [En línea] <http://msysgit.github.io/>.
8. Timcharper. *Git for Mac OS X - git-osx-installer*. [En línea] <http://sourceforge.net/projects/git-osx-installer/>.
9. Dudler, Roger. *Git - la guía sencilla*. [En línea] <http://rogerdudler.github.io/git-guide/index.es.html>.

11 Anexos

Por último, se adjuntan diferentes manuales para la compilación, instalación y uso de la aplicación.

11.1 Manual: Compilar la aplicación

Una vez instalado Android Studio, hay que importar el proyecto haciendo uso de la interfaz destinada para ello. Una vez importado correctamente, habrá que darle al botón de la barra superior “Play” que es verde.

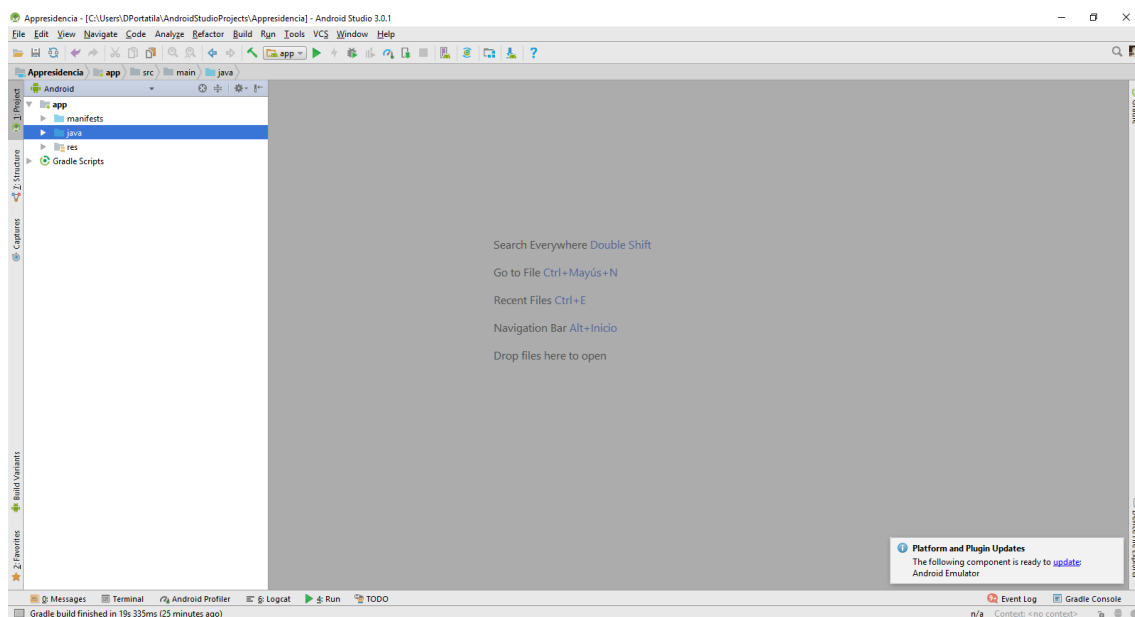


Ilustración 33- Android Studio IDE

Cuando haya terminado de compilar, Android Studio preguntará al usuario o usuaria donde quiere ejecutar la aplicación. En caso de que haya un móvil compatible (con opciones de desarrollador) conectado al ordenador, se podrá probar en este. Si no, se podrá ejecutar en el emulador.

Al compilar el programa se genera un archivo apk, válido para instalar en diferentes dispositivos.

11.2 Manual: Instalar la aplicación en un dispositivo físico

Si se desea instalar la aplicación en un dispositivo mediante la distribución del archivo apk, solamente se tendrá que copiar éste a la memoria del smartphone y ejecutarlo. Hay que tener en cuenta, que la opción de instalar aplicaciones de orígenes desconocidos tiene que estar activada: “Ajustes ➔ Privacidad (o equivalente) ➔ Orígenes desconocidos”.

11.3 Manual: Uso de la aplicación

A continuación, se describen las principales opciones de la aplicación:

11.3.1 Identificación y recuperación de la contraseña

La primera vez que se acceda a la aplicación, se le pedirá al usuario o usuaria que introduzca sus datos de usuario válidos. Por ejemplo:

- Usuario: urko@urko.eus
- Contraseña: urkourko

No se ha desarrollado la opción de “crear cuenta”, ya que, dada la singularidad del proyecto, la idea es que desde la residencia se generen las cuentas y sus contraseñas, con todos los datos del residente. Eso sí, si que se permite recuperar la contraseña desde la pantalla de inicio.

11.3.2 Funcionamiento

Una vez identificado correctamente, se accederá a la pantalla inicial de la aplicación, donde se podrá abrir el menú lateral que llevará a las siguientes opciones:

11.3.2.1 Menú diario

Permite, desde esta pantalla, seleccionar el día del que se quiere saber el menú de la comida o cena. Para ello se dispone de dos botones, uno que añade un día más a la fecha actual, y otro que muestra un calendario para elegir la fecha.

En cada plato, si la información está disponible, aparecerá un botón con el nombre de alérgenos. Al pinchar en él, muestra la información de los alérgenos.

Así mismo, esta pantalla permite contratar comida o cena del día seleccionado para llevar (take-away).

11.3.2.2 Ofertas

Permite conocer las ofertas disponibles en la residencia.

11.3.2.3 Peticiones

Muestra las peticiones de take-away hechas, y al mismo tiempo permite cancelarlas.

11.3.2.4 Monedero.

Permite saber el saldo actual de la cuenta identificada, así como, hacer una recarga con tarjeta de crédito ficticia.

11.3.2.5 Valorar servicio

Permite hacer una valoración del servicio dado, tanto de la comida, como de los y las trabajadoras.

11.3.2.6 Información:

Muestra la información básica de la residencia, como el teléfono, dirección, web, etc.

11.3.2.7 Ajustes:

Permite modificar la residencia y curso preferido. En este caso, se han poblado un curso de una residencia: Curso 2017-2018, residencia Manuel Agud Querol

11.3.2.8 Cerrar sesión

Permite cerrar la sesión actual, llevando al usuario o usuaria a la pantalla de identificación.