



IES Ciudad Escolar

**TRABAJO FINAL DE CICLO SUPERIOR DESARROLLO DE APLICACIONES
MULTIPLATAFORMA**

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)

Autores:

David Ortiz, Jesús Pedroza

Tutor:

José Sala

Versión:

3

Madrid, 22 de diciembre de 2023



1. Abstract

Índice:

1. Abstract
2. Justificación y Antecedentes
3. Introducción
4. Objetivos
5. Estimación de costes de proyecto
6. Desarrollo del proyecto
 - 6.1 Tecnología utilizada
 - 6.2 Planificación
 - 6.3 Especificación de requisitos
 - 6.4 Diseño de la aplicación
 - 6.5 Implementación
 - 6.6 Plan de pruebas
7. Conclusiones y mejoras futuras
8. Bibliografía y referencias
9. Anexos

2. Justificación y antecedentes

El motivo principal por desarrollar esta aplicación es debido al posible aumento de número de alumnos y profesores en el centro causando que en los aparcamientos se encuentren mal organizados y/o que no se encuentren plazas disponibles.

Con esta aplicación, se busca organizar las plazas y que el usuario seleccione la plaza que desea para así controlar dónde aparcar, cuando y por cuánto tiempo desea aparcar ya que no todos están todo el día en el centro.

Hemos decidido programar para android ya que es el S.O móvil más usado y con el lenguaje java debido a que es el lenguaje de programación más usado para crear aplicaciones móviles.

3. Introducción

Este trabajo está realizado por David Ortiz y Jesús Pedroza.

En este trabajo de final de grado se va a realizar una aplicación móvil para android sobre un sistema de reservas de aparcamiento en Ciudad Escolar. El objetivo es que tanto el alumno como el profesor puedan reservar una plaza de aparcamiento en el centro eligiendo el lugar, la fecha y la hora para aparcar. Al ser 2 tipos de usuario, se les otorga unas zonas a las que desea aparcar y las que no para evitar que aparquen en lugares en las que no deben aparcar y si la plaza ya está ocupada en determinada hora y del día, no podrá ocupar esa plaza. Además el usuario podrá ver un historial de las reservas que realizó y cancelar la reserva si la fecha de reserva es anterior a la fecha actual. El usuario deberá tener una cuenta solicitando la secretaría del centro la cual se le entregará un usuario, contraseña y un número secreto que lo debe usar si se le olvida la contraseña.

Debido a que es un proyecto rápido, el modelo de desarrollo será el modelo de prototipo ya que se desarrolla según lo que desea el cliente recibiendo críticas y propuestas de mejoras hasta el punto de que sea satisfactorio para él. La ventaja de usar este modelo es detectar errores en el proyecto, se detecta mayor margen de mejora para poder añadir funcionalidades más fácilmente y que en un futuro se puede reutilizar para crear versiones más avanzadas y complejas en base a esta aplicación (versión para IOS, página web, etc). También con este modelo de desarrollo, se puede mostrar al usuario final el desarrollo del programa.

Ya que la aplicación será para android el diseño de la aplicación se realizará en Android Studio que es el IDE oficial de android para desarrollar aplicaciones Android.

La interfaz donde el usuario interactúa con la aplicación, se usará XML.

El lenguaje de programación que se usará será java para programar las funciones que realizará el usuario y SQLite para implementar la base de datos que almacenará todos los datos sobre aparcamiento y usuarios que es la recomendada para aplicaciones android por su ligereza para que una aplicación android tenga buen rendimiento.

Para que el usuario final pueda utilizar esta aplicación necesita descargarla desde la plataforma de Uptodown que es gratuita y necesitará un dispositivo móvil con android 8.0 o superior para poder ejecutarlo.

4. Objetivos

Los objetivos que tenemos en este proyecto es realizar una aplicación de reservas de aparcamiento la cuales son:

- Organizar mejor las plazas de aparcamiento según el horario y día.
- Evitar que personas ajenas al centro sin autorización aparquen en las plazas.
- Crear parámetros para evitar que se reserven plazas fuera del horario lectivo, festivos, fines de semana y fuera del curso escolar (vacaciones escolares).
- Conocer mejor la distribución de las plazas en el centro.
- Evitar que no se aparque fuera del horario y/o fecha lectiva.

5. Estimación de costes del proyecto

El precio por hora de un desarrollador freelance de aplicaciones móviles es de 20€ la hora. Al ser 2 desarrolladores y el proyecto en total son de 30 horas, será un total de 1200€ el coste.

Para la temporalización y planificación hemos tomado la decisión de separar el proyecto en distintas fases teniendo en cuenta el ciclo de vida software

La primera etapa fue identificar la falta de un servicio para gestionar las plazas de ciudad escolar de una manera más organizada y así asegurando una gestión más eficiente

La segunda etapa es empezar el diseño y proponer ideas de qué tipo de software se adaptará para solucionar este problema y llegamos a la conclusión de que una aplicación móvil desarrollada para android era la más versátil

La tercera etapa es la de desarrollo la cual ocupa la mayor parte de proyecto e implica la creación de todo el código así como de la interfaz y compatibilidad con sistemas móviles esta etapa será desarrollada en conjunto y no dividiremos el trabajo según la dificultad y duración de este, así mismo tendremos reuniones y conversaciones sobre el avance de manera frecuente

La cuarta etapa será la de pruebas en la cual se comprobará la consistencia de la base de datos posibles bloqueos o concurrencia así como acceso a datos de seguridad o claves, además de probar todas las funciones y sus limitaciones

La quinta y última fase sería el despliegue en el cual se crearía el apk y se plantearía el poder lanzarla en la tienda de aplicaciones. (preguntar sobre esta parte a Jesús y sobre el ciclo de vida que hemos elegido)



6. Desarrollo del proyecto

Todo el proyecto está almacenado en nuestro github:

<https://github.com/DavidJesusTFG/ARACE>

Y además se puede descargar desde aquí:

<https://arace.uptodown.com/android>

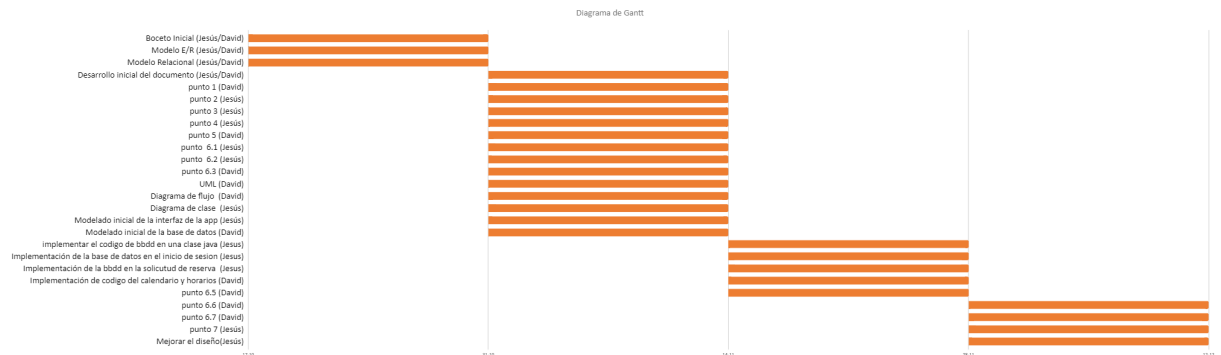
6.1. Tecnología utilizada

Estas son las tecnologías que vamos a usar en nuestro proyecto:

- Android Studio: para desarrollar nuestra aplicación móvil y en este IDE se encuentran las siguientes tecnologías:
 - XML: Crea la estructura completa de la interfaz que interactúa con el usuario final.
 - Java: Lenguaje de programación muy común en el desarrollo de aplicaciones móviles android la cual se programaran las funciones de la interfaz y la base de datos.
 - SQLite: La base de datos para almacenar los usuarios y las plazas de aparcamiento. Para poder manejar mejor la base de datos usaremos:
 - SQLite Browser: Sirve para tener mejor visualización de la BBDD además de realizar el CRUD de manera más sencilla a nivel de administrador y hacer prueba de consultas para luego implementarlo en el java de la app.
- Git: Repositorio de nuestro proyecto para poder trabajar conjuntamente.
- Uptodown: Plataforma gratuita para subir la aplicación sin pagar.

6.2. Planificación

Para distribuir el proyecto, cada uno de nosotros realizaremos distintas partes durante el tiempo que hemos planificado. Esto lo mostramos con el siguiente diagrama de Gantt:



No obstante esto es orientativo pueden cambiar las tareas a última hora.

6.3. Especificación de requisitos

En el nuestra especificación de requisitos vamos a tener en cuenta las necesidades de nuestros clientes así como la manera más óptima de abordar las soluciones ya planteadas y los distintos métodos empleados para estas

Una de las partes más relevantes en el especificacion de los requisitos es el previo estudio de las necesidades que nuestro proyecto va a satisfacer así como también las que en un futuro se podrán llegar a implementar y cuales aunque pueden llegar a ser futuras mejoras no estarían al alcance de nuestro equipo actual, tras tenerlo todo bien analizado hemos llevado a cabo un diagrama de gantt y una planificación de las funcionalidades que vamos a desarrollar y de los tiempos que estas nos requieren, los requisitos va a ser divididos en.

Requisitos funcionales: describen las acciones que la app realizará

Requisitos no funcionales: define los criterios que se emplean a la hora de juzgar la operación de un sistema en lugar de sus comportamientos específicos(revisar la frase)

Para el correcto desarrollo de nuestra app hemos establecido los siguientes requisitos:

- Gestión de la concurrencia a la hora de reservar
- Gestión de acceso a de los usuarios a las zonas
- Gestión de usuarios y claves
- Control de permisos en los usuarios
- Gestión del calendario
- Mantenimiento en la bbdd en torno a las plazas
- Correcta navegación en la interfaz de usuario
- Interfaz intuitiva y sencilla
- Visualización del historial por el usuario
- Limitaciones en el calendario festivo
- Limitaciones en los horarios
- Limitaciones en el número de plazas a reservar
- Limitación acceso a zonas
- Funcionalidad de la gestión de claves
- Implementación de la APP en android
- Implementación de la BBDD SQLITE
- Hashear la contraseña
- Cancelación de la reserva

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)

- Búsqueda de reserva por fecha
- Recuperación de disponibilidad de la plaza al caducar la reserva
- Ocultar reservas ya caducadas
- Prohibición de selección de hora fuera de rango
- Descarga de la aplicación desde una plataforma de descarga de aplicaciones
- Implementación de imágenes dependiendo de la zona de las plazas
- Contador de plazas disponibles en las zonas
- Borrado de datos muy antiguos sobre las reservas

Antes de hacer la aplicación, necesitamos unos requisitos que pide el cliente de la cual lo escribimos para luego comenzar a desarrollar.

Hemos desarrollado unos requisitos que se deben cumplir para el correcto funcionamiento de la aplicación:

Requisito	Tipo de requisito (Funcional/No funcional)	Descripción
RE1	FUNCIONAL	Reserva de plaza
RE2	FUNCIONAL	Vista de zona
RE3	FUNCIONAL	Gestión de acceso
RE4	FUNCIONAL	Gestión de concurrencia
RE5	FUNCIONAL	Gestión Usuarios y claves
RE6	FUNCIONAL	

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)

Para desarrollar todos estos requisitos, es necesario organizarlo y dividirlos en tareas:

Tarea	Descripción	Requisitos usados
TA1		

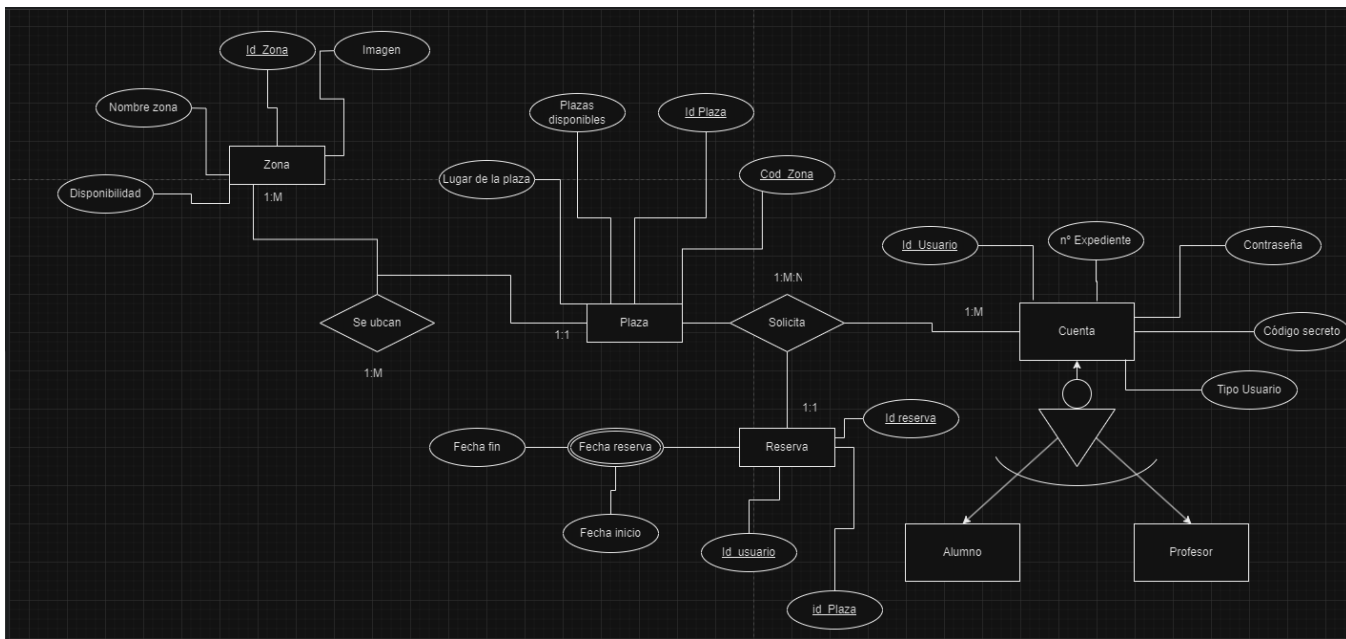
6.4. Diseño de la aplicación

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)

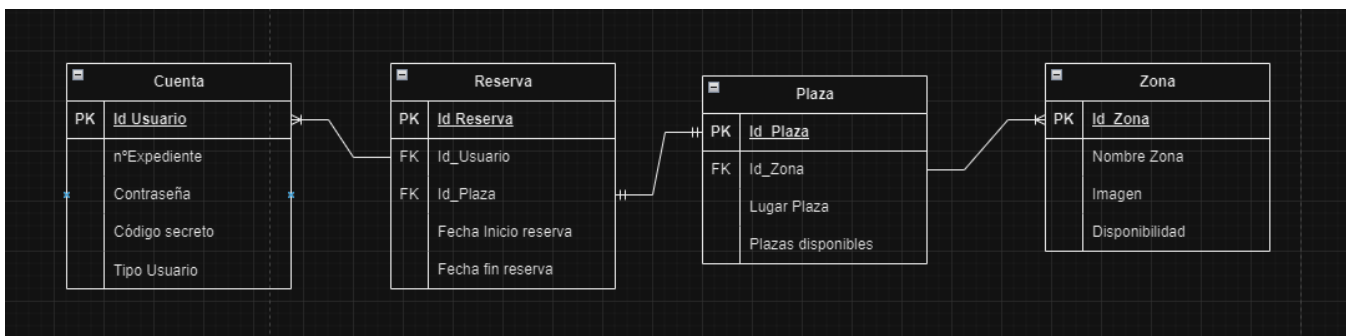
Ya con los requisitos puestos, ya estamos listos para comenzar a desarrollar la aplicación.

La aplicación debe tener una base de datos para obtener los datos para ello, desarrollaremos un modelo entidad relación para conocer cómo se comportaría la base de datos dentro de la aplicación y después un modelo relacional para saber como crear las tablas de la base de datos y que columnas tendrá cada tabla.

Modelo Entidad-Relación

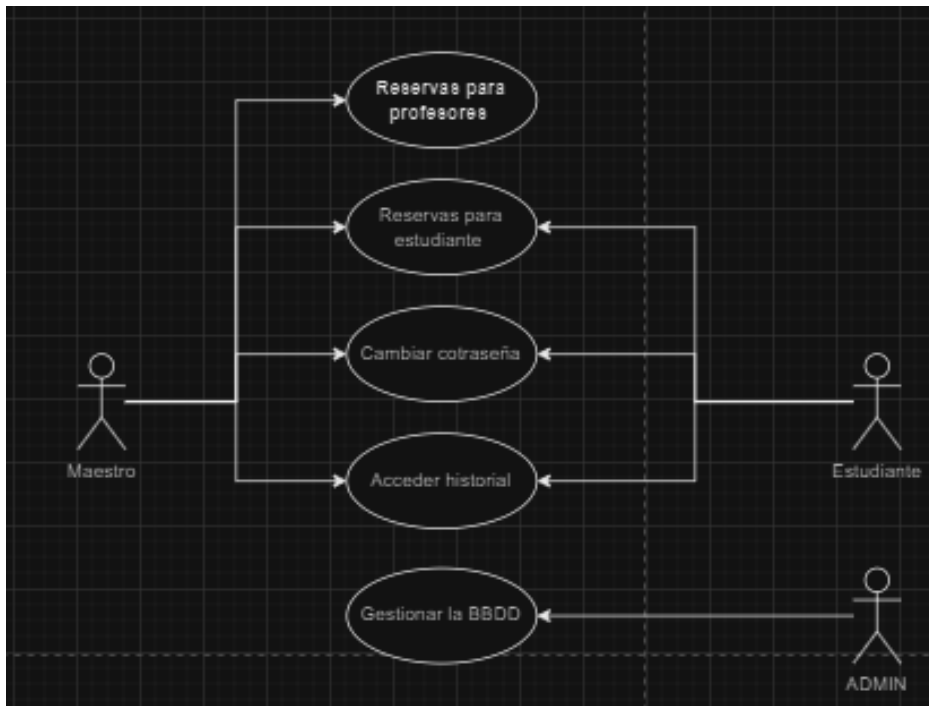


Modelo Relacional:



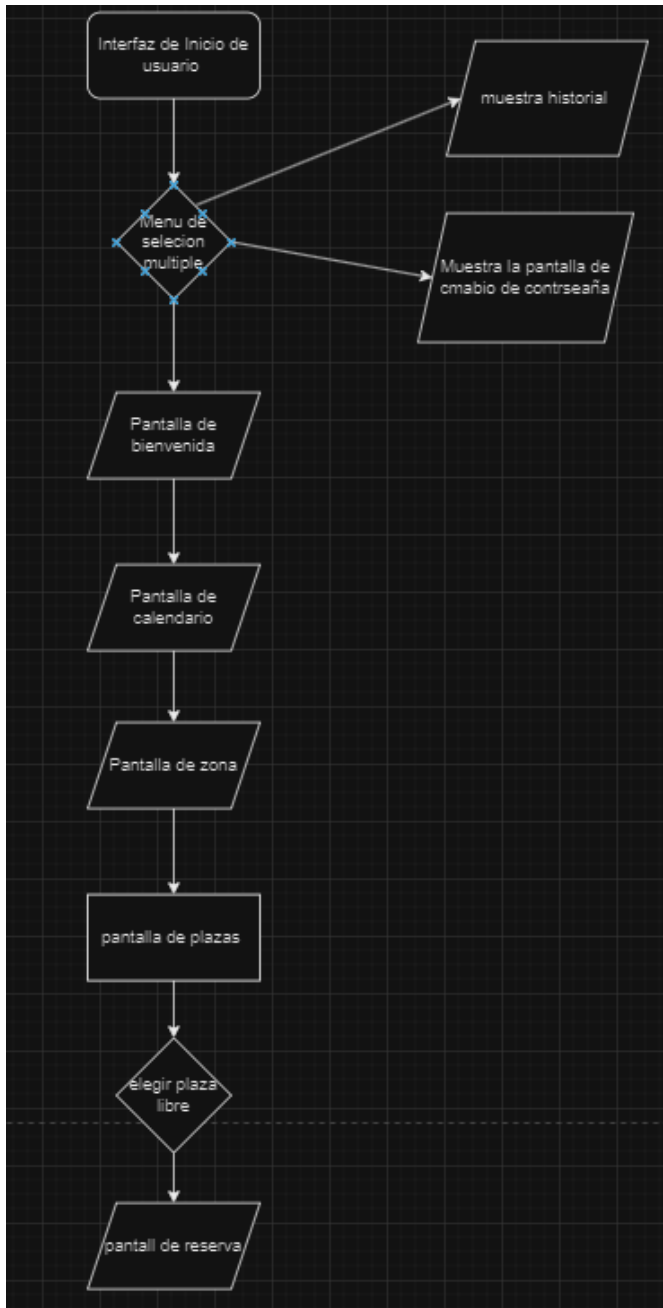
APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)
 Para saber qué rol tendrá según el tipo de usuario y que funciones puede hacer,
 desarrollamos un UML para saber mejor la distribución.

UML:



APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE) también desarrollamos un diagrama de flujo. Este diagrama de flujo sirve para saber cómo se maneja la aplicación a nivel de usuario otorgando las opciones que podrá ver en la pantalla y en qué orden irán.

Diagrama de flujo:



Si queremos conocer mejor la estructura interna del programa, desarrollaremos un diagrama de clases que consiste en distribuir cuales son las clases que va a tener la aplicación, las variables y los métodos que tiene cada clase además de cómo las

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)

Al terminar la fase de diseño vamos a iniciar con la fase de implementación en la cual vamos establecer el lenguaje de programación y el entorno en el cual vamos a programar la aplicación

El cual va a ser java y el entorno Android studio ya que estos son los métodos con los cuales hemos trabajado y que más dominamos a su vez nos a parecido lo más adecuado para desempeñar la programaciones de la aplicación de la manera más satisfactoria, ya que mediante el uso de la interfaz gráfica que está entorno (Android Studio) tiene podemos llevar a cabo el apartado frontend adecuado y "estandarizado" para que cualquier persona que vaya a usar la aplicación se a capaz de usarla de manera intuitiva, además la parte del backend la vamos a desarrollar en java ya que es el lenguaje que más dominamos y hemos empleado en múltiples módulos.
(Parte del modelo de ciclo de vida)

Durante el desarrollo de la aplicación se han creado unas tablas en la base de datos en las cuales el administrador podrá crear los usuario así como las plazas y distintas zonas, el usuario solo podrá crear reservas en la base de datos y cambiar su contraseña que no su código único el cual es empleado para cambiar la contraseña

Las funciones principales que nuestra aplicación va a llevar a cabo serán de crear reservas de una plaza teniendo en cuenta las horas y días que estén permitidos, excluyendo los días festivos así como los fin de semana, esto será llevado a cabo mediante una tabla la cual indicará el tipo de día, en cambio el formato de las hora está contemplado en el código java y no en la base de datos, a la vez se mostrará de manera visual si la plaza está disponible mediante una select en la base de datos comprobando si ese día tiene ya asignada una reserva, a su vez se implementaran fotos de cada plaza para así tener de manera más clara donde están ubicadas, estas imágenes están almacenadas en la aplicación y no en la base de datos

Otro de los requisitos es la contemplación de que se bloquee el inicio de sesión si el usuario falla al intentar entrar en la aplicación un número x de veces

También se planteó desarrollar una función que trate la concurrencia a la hora de reservar una misma plaza, aunque esta función pueda ser planteada para una versión futura de la aplicación más completa.

6.6. Plan de pruebas

6.7. Despliegue y mantenimiento

7. Conclusiones, mejoras futuras

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)
Durante el desarrollo nos dividimos el trabajo y nos reunimos para ir organizando y siguiendo el proyecto.

Al final del desarrollo creamos una aplicación que puede realizar reservas para poder aparcar en el lugar, el día y la hora deseada con la opción también de mostrar un historial y poder cancelar la reserva.

Hemos conseguido los requisitos deseados en el proyecto no obstante, se puede mejorar en los siguientes puntos:

- Crear una base de datos en la red: Al realizarlo todo local los usuarios no pueden ver la base de datos actualizada en distintos dispositivos. Para ello es necesario crear una base de datos en red usando por ejemplo xampp con phpmyAdmin esto facilita poder realizar las concurrencias.
- Crear una versión para IOS: Realizamos la app para android pero solo funciona para android y no para IOS. Para hacerlo debemos crear una versión usando el lenguaje Swift para que funcione o para no complicar el desarrollo utilizar IONIC 5 un framework basado en angular que permite crear la app tanto para android y para IOS.
- Desarrollar una página web: Si el usuario desea realizar una reserva desde el ordenador o por el navegador del móvil, sería crear una página web.
- Utilizar una API para el calendario: Para que tenga una mejor visualización del calendario y evitar ajustar las fechas de manera manual con una tabla, desarrollamos una api para que recoja las fechas sin la necesidad de una tabla de bbdd.
- Realizar un muestreo de las plazas disponibles dependiendo de la fecha y la hora elegida es decir que si el usuario elige primero la fecha y la hora, se muestran las plazas disponibles en ese día y entre las horas elegidas.

8. Bibliografía y referencias

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)

- Documentación de Android Studio:
<https://developer.android.com/docs?hl=es-419>
- Guía de subida de la app a Uptodown:
<https://support.uptodown.com/hc/es/articles/360053260491-C%C3%B3mo-publicar-una-app-en-Uptodown>
- SQLite en android studio:
<https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase>
[e](#)

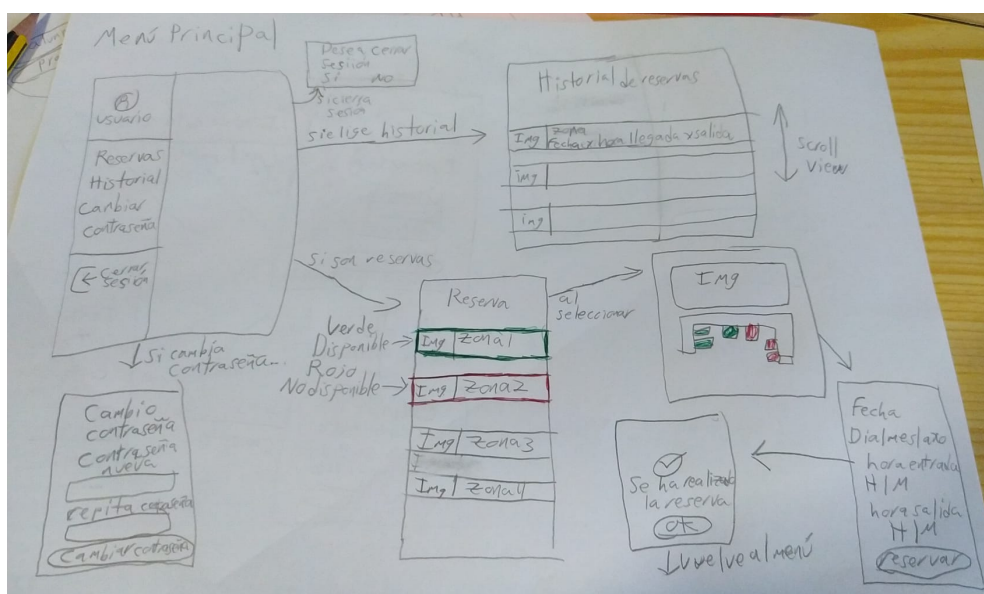
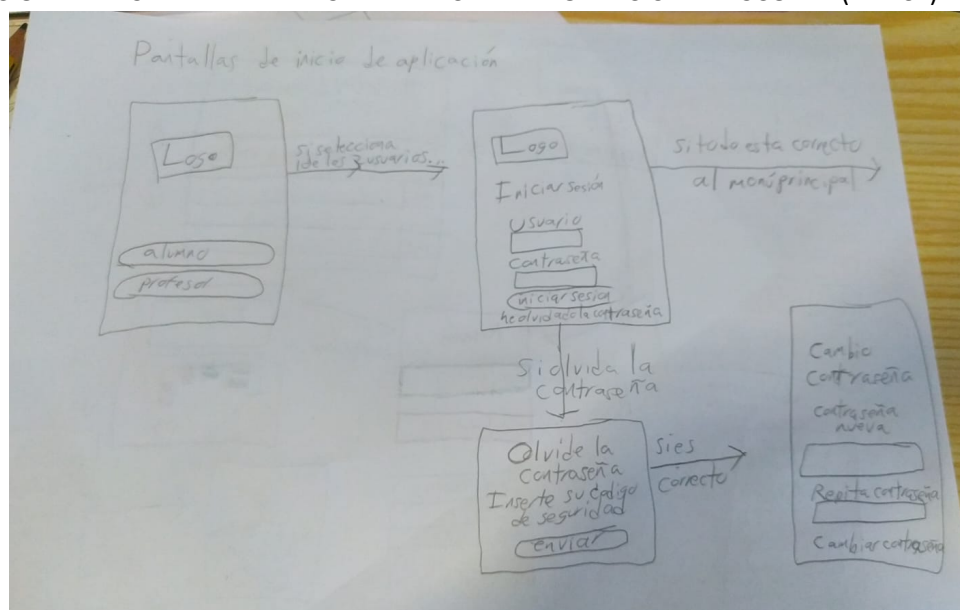
9. Anexos

Glosario:

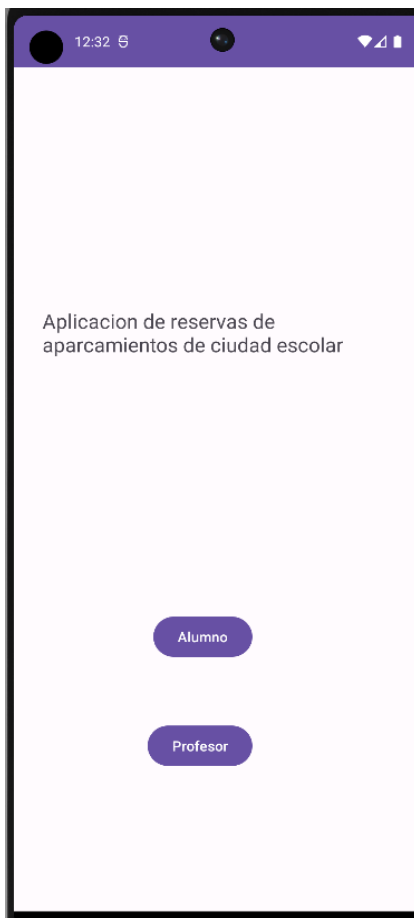
- BBDD: Base de datos
- CRUD: Significa crear, leer, modificar y borrar. Son las funciones básicas de la base de datos.
- UML: Lenguaje para visualizar, especificar o describir métodos y procesos básicamente un plano.
- XML: Extensive Markup Language, es un metalenguaje que permite definir lenguajes de marcas para formar los datos de manera legible.

Anexo 1: Boceto inicial


APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)



Anexo 2: Primera versión de la aplicación



APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)



Inicio Sesión

Nº Expediente

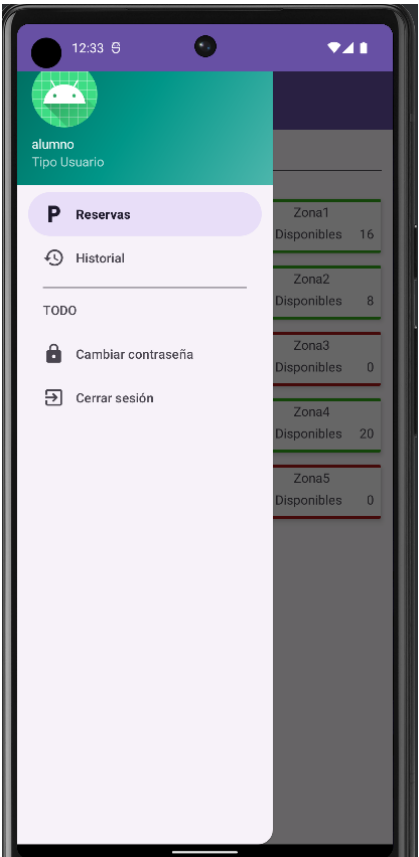
Contraseña

Iniciar Sesión

He olvidado la Contraseña



Buscar	
Zona1	Plazas Disponibles 16
Zona2	Plazas Disponibles 8
Zona3	Plazas Disponibles 0
Zona4	Plazas Disponibles 20
Zona5	Plazas Disponibles 0



APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)



APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)

12:34 6

fragment_cambio_de_contras...

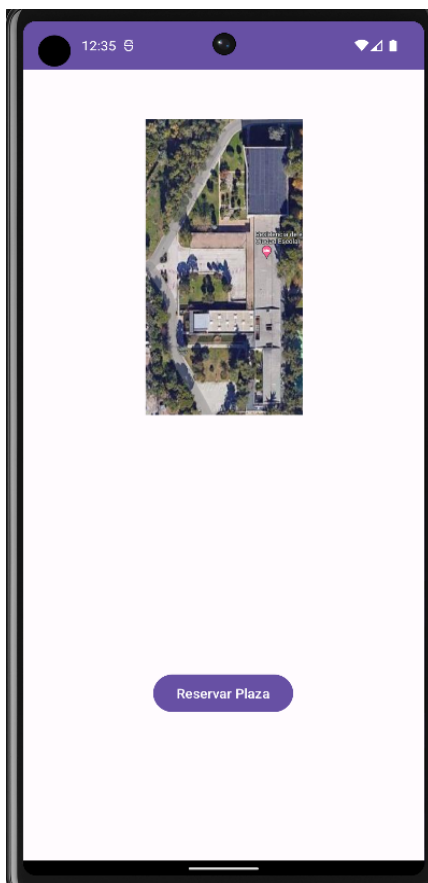
Cambio de contraseña

Inserte su contraseña antigua

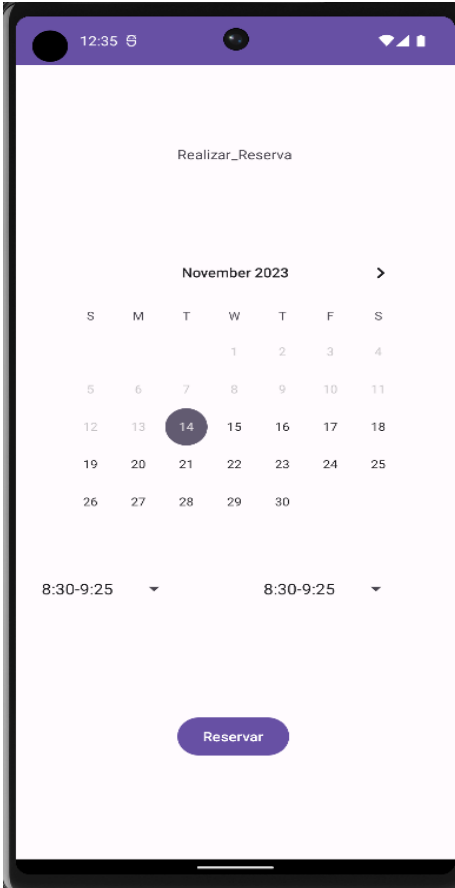
Contraseña

Repita la contraseña

Cambiar contraseña



APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)



Realizar_Reserva

November 2023

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

8:30-9:25 8:30-9:25

Reservar



La reserva se ha realizado correctamente

Ok

Anexo 3: Versión final de la aplicación



Olvidé la contraseña

Inserte su código de seguridad

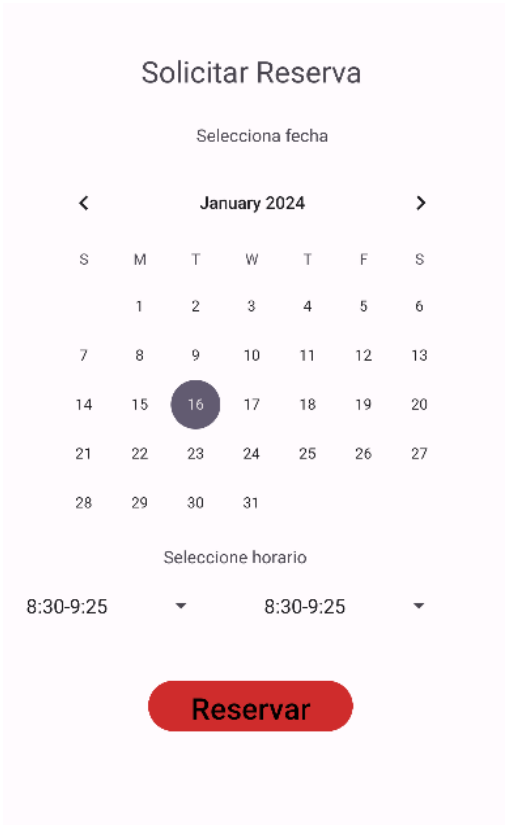
Enviar

Cambio de contraseña







Contraseña

Repita la contraseña

Cambiar contraseña



APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)


	Zona Entrada	Plazas Disponibles: 6
	Zona Cafetería	Plazas Disponibles: 2
	Zona hotel-escuela	Plazas Disponibles: 2
	Zona Pabellón	Plazas Disponibles: 0
	Zona jardín	Plazas Disponibles: 1
	Zona residencia de estudiantes	

Zona Entrada



Plaza: E-1	Disponibilidad: Si
Plaza: E-2	Disponibilidad: Si
Plaza: E-3	Disponibilidad: No
Plaza: E-4	Disponibilidad: Si
Plaza: E-5	Disponibilidad: Si

APLICACIÓN DE RESERVA DE PLAZAS DE APARCAMIENTO DE CIUDAD ESCOLAR (ARACE)



Zona Pabellón

Fecha: 2024-01-16

Cancelar

Hora Entrada: 8:30

Hora Salida: 9:25


Zona Entrada

Fecha: 2024-01-23

Cancelar

Hora Entrada: 8:30

Hora Salida: 9:25



Cambio de contraseña

Inserte su contraseña antigua

Contraseña

Repita la contraseña

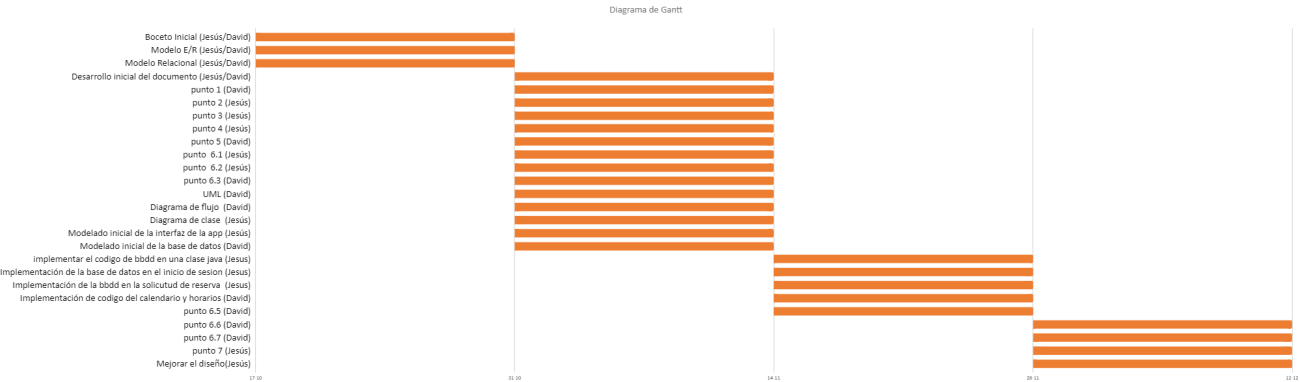
Cambiar contraseña

Anexo 4: Logotipo

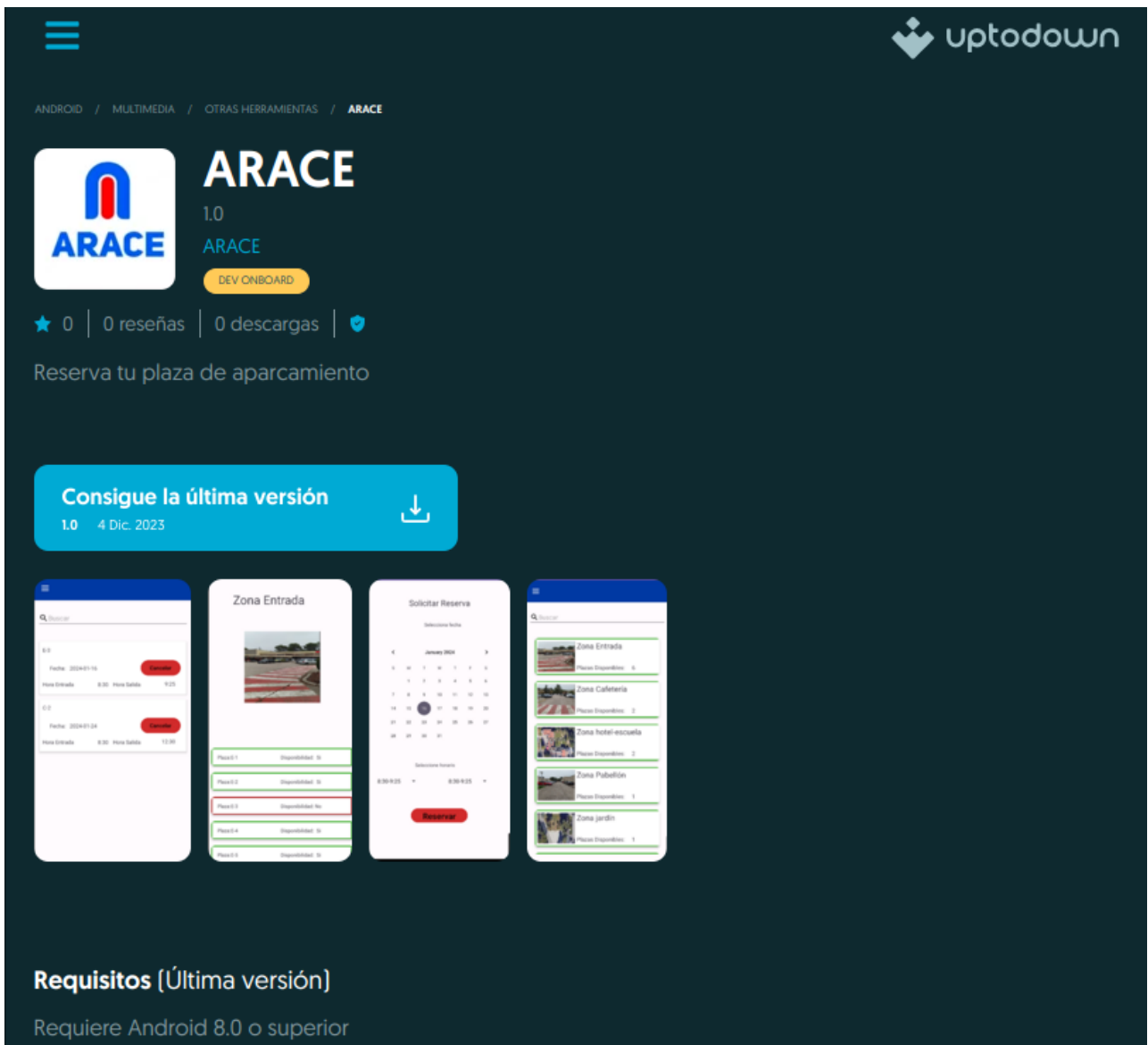




Anexo 5: Diagrama de Gannt



Anexo 6: La aplicación en Uptodown



The screenshot shows the Uptodown website interface for the ARACE app. At the top, there's a navigation bar with the Uptodown logo and a menu icon. Below the navigation bar, the breadcrumb trail reads: ANDROID / MULTIMEDIA / OTRAS HERRAMIENTAS / ARACE. The main section features the ARACE app icon, the name 'ARACE', version '1.0', and a 'DEV ONBOARD' button. Below this, it shows '0 reseñas' and '0 descargas'. A description reads 'Reserva tu plaza de aparcamiento'. A large blue button prompts users to 'Consigue la última versión' (1.0, 4 Dic. 2023) with a download icon. Below the button, four preview images show app screens: 'Zona Entrada', 'Solicitar Reserva', and two other screens showing parking zone details. At the bottom, the 'Requisitos (Última versión)' section states 'Requiere Android 8.0 o superior'.

Consigue la última versión
1.0 4 Dic. 2023

Requisitos (Última versión)
Requiere Android 8.0 o superior