

|  |
| --- |
| Bcutzz  Barbershop |

|  |  |  |
| --- | --- | --- |
| Proyecto final |  |  |
| Nombre: Antonio Moscoso  Fecha: Diciembre 2019 | 2º DAM  IES Almunia |  |

ÍNDICE

[DOCUMENTACIÓN USUARIO 3](#_Toc12548547)

[Objetivo 3](#_Toc12548548)

Introducción\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

[Requisitos previos 3](#_Toc12548550)

[Finalidad 3](#_Toc12548551)

Funciones\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_3

Requisitos para el desarrollo

Guia de uso

Funciones del sistema

Principales funciones del sistema

Solución de problemas

[RESUMEN FINANCIERO 3](#_Toc12548552)

[BALANCES GENERALES 3](#_Toc12548553)

[Balance de la posición financiera 3](#_Toc12548554)

[Balance de ingresos completo (pérdidas y ganancias) 3](#_Toc12548555)

[Balance de variaciones de capital 3](#_Toc12548556)

[Balance de flujo de efectivo 3](#_Toc12548557)

[NOTAS DE LOS BALANCES GENERALES 3](#_Toc12548558)

[Cuentas 3](#_Toc12548559)

[Deuda 3](#_Toc12548560)

[Deuda 3](#_Toc12548561)

[Negocio establecido 3](#_Toc12548562)

[Obligaciones contingentes 3](#_Toc12548563)

[Aportes 3](#_Toc12548564)

[INFORME DE AUDITORÍA INDEPENDIENTE 3](#_Toc12548565)

[Informe del Auditor 3](#_Toc12548566)

# OBJETIVO

La aplicación es un gestor de citas de una barbería desarrollada en Android nativo. Con ella el cliente podrá solicitar una cita para el servicio que desee. Y el administrador podrá gestionar su agenda a través de otra aplicación desarrollada en Ionic.

Con esto se le permitirá al barbero no tener que estar continuamente atendiendo llamadas y podrá gestionar y llevar su agenda en cualquier dispositivo.

## INTRODUCCIÓN

Tendremos dos aplicaciones, una que será la que instalen los clientes y otra que solo tendrá y usara el administrador

Usaremos un host gratuito donde almacenaremos la base de datos.

En dicho host también almacenaremos los distintos webServices que iremos llamando desde las aplicaciones para gestionar todas las tareas necesarias.

La parte de cliente nos permitirá primeramente registrarnos si aún no lo hemos hecho o loguearnos. Una vez hecho esto ya podremos solicitar la cita y ver más información como donde se encuentra la barbería, ver información de mis citas, ver redes sociales desde la propia aplicación y hacer logout.

La parte de administrador nos permitirá ver todas las citas asignadas y cancelar las citas. También tendremos otro panel donde podremos ver todos los clientes que tenemos registrados, y un último panel que nos permite añadir los días que no quisiésemos trabajar, que bloqueara ese día en el calendario de los clientes.

¿Cree que es difícil aplicar formato a un documento tan atractivo como este? Para nada. Para aplicar fácilmente cualquier formato de texto que vea en este documento con un solo clic, vaya al grupo Estilos de la pestaña Inicio de la cinta de opciones.

# Finalidad

Aplicación Android:

1. Registro de usuarios.
2. Acceso de usuarios.
3. Gestión de citas.
4. Consultor de citas.
5. Módulo de Google Maps.
6. Módulo de redes sociales.
7. Cierre de sesión.

Aplicación Ionic:

1. Gestor y filtro de citas.
2. Gestor de clientes.
3. Gestor de días.

## FUNCIONES

Aplicación Android:

1. Ventana de logueo o registro.
2. Menú con distintos fragments:
   1. Seleccionar servicio y solicitar cita.
   2. Ver localización en Google maps.
   3. Ver mis citas.
   4. Ver redes sociales
   5. Cerrar sesión.

Aplicación Ionic:

1. Ver, filtrar y eliminar citas.
2. Ver clientes.
3. Ver y gestionar días libres.

# Requisitos para el desarrollo

1. Una base de datos donde almacenaremos todos los datos relacionados con los clientes, los días y las horas de cada día.
2. Aplicación Android dónde permitiremos el registro de cualquier nuevo cliente, cada persona solo podrá tener un usuario, para ello le solicitaremos el DNI.
3. Una vez registrado ya podrá loguearse.
4. Utilizaremos una librería para el calendario que nos permitirá bloquear los fines de semanas, así como los días que el administrador quiera.
5. Una vez logueado el usuario, podrá solicitar una cita, primero seleccionando un servicio y después un día. Al seleccionar el día le cargarán las horas disponibles para ese día según el servicio seleccionado.
6. Utilizaremos un fragment que nos permitirá ver nuestras citas, ver la localización del sitio en Google Maps, las redes sociales desde la propia aplicación y cerrar sesión.
7. Ya en la parte del administrador tendremos tres tabs con los que podremos ver y eliminar las citas, otro donde veremos los clientes y un último que nos permitirá agregar o eliminar aquellos días que no trabajemos.
8. Utilizaremos web services para conectar y hacer las distintas gestiones con la base de datos.

## REQUISITOS PREVIOS

Cliente

Necesitamos mínimo: SdkVersion 28 (Android 9).

Solamente se necesita una conexión a internet para poder conectar al servidor web.

APK de la aplicación que nos permita instalarla

Administrador

Se necesita una conexión a internet para poder conectar al servidor web.

APK para instalar en Android, IPA para IOS o el enlace web para cualquier otro dispositivo.

# guía de uso

Android

1. Una vez tengamos la app instalada lo primero que tendremos que hacer será registrarnos, para ello debemos insertar todos nuestros datos correctamente ya que se comprobaran que todos los campos sean válidos (Dni, tfno, correo, etc).
2. Tras registrarnos o si ya lo habíamos hecho antes tendremos que loguearnos en la aplicación y podremos marcar si queremos o no que nos recuerde nuestro usuario y contraseña.
3. Entraremos al menú principal, en el que por defecto nos aparecerá para solicitar la cita, también podremos seleccionar otros fragment para ver la localización en Google Maps, ver nuestras citas, ver las redes sociales desde la propia app o cerrar sesión.
4. Si queremos solicitar una cita, primeramente, elegiremos un servicio, ya sea barba, pelado o barba + pelado, que tendrán una duración de 15, 30 o 45min respectivamente.
5. Una vez aceptemos, se nos llevará a otro fragment donde nos carga un DatePicker con los días que haya disponible (Los fines de semanas siempre estarán bloqueados ya que el barbero no trabaja esos días, pero también aparecerán bloqueados los días que estén completos o los días que quiera tomarse libres)
6. Cuando aceptemos el dia del DatePicker automáticamente nos cargaran en un ListView las horas que hay disponible para ese día y ese servicio\*.
7. \*(Si selecciono un servicio de 45min y las 9:00 está libre pero las 9:30 tiene una cita ya asignada, no me mostrará las 9 ni las 9:15 ya que estas horas me chocarían con las 9:30).
8. Finalmente, si aceptamos, se nos asignará la cita y se nos informará de ello.

Ionic

1. Podremos bien instalar la aplicación o acceder mediante la web.
2. En cuanto accedamos por defecto, nos aparecerán las citas que tenemos asignadas. Tendremos un select y una opción para buscar, bien por el nombre del cliente, o bien por el día.
3. En cada cita nos aparecerá un botón de borrar, que en caso de que lo pulsemos, nos borrará la cita y se recargará automáticamente. En la base de datos no borrará las horas, si no que las asignará de nuevo como disponibles.
4. Tenemos otra opción donde podremos ver otra lista con la información completa de nuestros clientes: Nombre, Apellidos teléfono, correo, fecha de nacimiento.
5. Y por último tendremos una lista con los días que bloquearemos en el DatePicker para que nadie pueda solicitar una cita.
6. Podremos tanto insertar días nuevos, que se agregarán y se recargarán automáticamente una vez los agregemos, como borrarlos y volver a marcarlos como disponibles.

# sOLUCIÓN DE PROBLEMAS

Tanto la parte de Android como de Ionic, nos podrá dar error porque no tengamos acceso a internet o al conectar con la base de datos. En cualquier caso y en ambas aplicaciones, se nos informará de ello a través de un toast.

Para solucionarlo, debemos asegurarnos simplemente de tener una conexión decente a internet.

En la parte Android, nos podrá dar error únicamente en el registro si el usuario no introdujese bien algún campo como un nombre o apellidos con caracteres numéricos, un correo incorrecto, un DNI incorrecto, o un teléfono con caracteres y se le informará colocando en rojo ese campo y mostrándole un Toast.

Si esto estuviese correcto, el usuario podrá pulsar el botón de registro y en caso de que haya un usuario con el mismo nombre de usuario o con el mismo DNI cargado en nuestra base de datos. Si así fuese se informará al usuario a través de un toast y no se agregará el nuevo usuario a la base de datos.