

6 de noviembre de 2019

Manual Técnico: Memorias Urbanas

Gabriel Vindas Brenes 2014103931

Erick Muñoz Rojas 2017109369

Sofía Mannix Sánchez 2017134730

Jannio Chaverri Agüero 2015072182

David Valverde Zúñiga 200922986

Instituto Tecnológico de Costa Rica

San José, II Semestre 2019

Contenido

Objetivo General del Manual Técnico	3
Objetivos Específicos del Manual Técnico	3
Descripción de la aplicación.....	3
Requisitos de instalación	3
Instalación de la aplicación	4
Ejecución de la aplicación	4
Requerimientos del entorno de desarrollo	5
Descripción de los actores del sistema	6
Implementación de la aplicación	6
Flutter	6
Estructura	8
Dependencias del proyecto	9
Implementación de la plataforma web	11
React JS	11
Estructura	12
Dependencias del proyecto	12

Objetivo General del Manual Técnico

Ofrecer la información técnica necesaria para la instalación y el mantenimiento de la aplicación móvil y web.

Objetivos Específicos del Manual Técnico

- Determinar la estructura técnica y su diseño.
- Especificar los requerimientos de hardware y software.
- Detallar las tecnologías utilizadas para el desarrollo de la aplicación.

Descripción de la aplicación

La aplicación “Memorias Urbanas” brinda una forma digital de acercar a las personas al ámbito cultural que se encuentra presente en Barrio Amón. Mediante un mapa y un catálogo de lugares con interés histórico o cultural, las personas pueden acceder a la aplicación y aprender o explorar acerca de las localidades que se encuentran en la zona.

Requisitos de instalación

Para instalar esta aplicación se requiere un dispositivo móvil que cumpla las siguientes características:

- Teléfono inteligente (smartphone) con sistema operativo Android, versión 5.0 (Lollipop) o más reciente.
- Teléfono inteligente (smartphone) con sistema operativo IOS, compatible con la versión 9.0 de Xcode o una más reciente.
- El dispositivo debe tener al menos 512MB de memoria RAM y al menos 85,47MB de espacio de almacenamiento en la memoria interna disponibles.

Instalación de la aplicación

Para instalar la aplicación en el dispositivo, deberá obtener el archivo APK instalable a través de alguno de los servicios de distribución utilizados por los desarrolladores (como Google Play, IOS Store) u otros canales de distribución.

Además, si se desea instalar la aplicación en un dispositivo IOS y esta no es encuentra todavía en la IOS Store, se requiere de la posesión de una computadora con sistema operativo MAC OS.

Esta sección será actualizada conforme se incorporen canales de distribución para la obtención de esta aplicación, dado que del momento los únicos medios de obtención son a través del repositorio git y en la carpeta de Google Drive oficial de la aplicación.

Ejecución de la aplicación

Para ejecutar la aplicación móvil, seleccione el ícono de la aplicación a través del menú de aplicaciones de su dispositivo Android o IOS. Por el momento la aplicación no posee un icono definido y usa el establecido por defecto en su respectivo sistema operativo.

Para ejecutar la aplicación web, debe iniciar el servidor local mediante el uso de “node package manager”. Debido a que del momento el acceso de la aplicación web no se encuentra disponible en un url.

Requerimientos del entorno de desarrollo

Para realizar labores de mantenimiento e implementación de la aplicación, se deben tener las siguientes herramientas de software instaladas en los equipos a utilizar:

Móvil	
Lenguaje de Programación	Dart (Lenguaje Orientado a Objetos) v2.5.0
Ambiente de Desarrollo	Android Studio v3.4
Herramienta (Framework) de Desarrollo	Flutter v1.9.1+hotfix.5 (Desarrollo IOS-Android)

Web	
Lenguaje de Programación	Javascript
Ambiente de Desarrollo	Visual Studio Code
Herramienta (Framework) de Desarrollo	React JS

Hasta el momento la aplicación ha sido probado en los siguientes dispositivos:

Dispositivo Digital de Pruebas	Google Nexus 5 (Emulador Android Studio)
Dispositivo Físico de Pruebas	Samsung tablet Galaxy E, Samsung Galaxy S7 Edge y Huawei Mate 20.

De igual manera la aplicación está diseñada para tener completo funcionamiento con las siguientes versiones de Android y IOS.

Android	Android Jelly Bean (API 16) o superior
IOS	IOS 8 o superior

Descripción de los actores del sistema

Nombre del Actor	Descripción
Usuario de la Aplicación	Utiliza la aplicación móvil para explorar lugares de interés a lo largo de Barrio Amón.
Administrador de la Aplicación	Se encarga de manejar los datos a través del sistema web establecido.

Implementación de la aplicación

A través de esta sección se brinda una explicación de las características del diseño del código fuente utilizado para la generación de la aplicación Android / IOS a través de un proyecto de aplicación utilizando alguno de los IDEs mencionados en la sección “Requerimientos de Entorno de Desarrollo”:

Flutter

Flutter es el kit de desarrollo implementado con el fin de producir una aplicación que pueda ser liberada de forma paralela en Android y IOS.

<p>Widgets</p>	<p>Cada elemento dentro de Flutter es un widget. Lo que indica que cada uno posee sus propios atributos, funciones y cualidades específicas. Por ejemplo:</p> <ul style="list-style-type: none"> • Botones. • Barra de navegación. • Menú lateral. • Imágenes. <p>Cada uno representa un widget diferente.</p>
<p>Árbol de Widgets</p>	<p>Flutter implementa una jerarquía en forma de una estructura arbórea de datos con el fin de maximizar eficiencia.</p>
<p>Árbol de Widgets</p>	<p>Flutter implementa una jerarquía en forma de una estructura arbórea de datos con el fin de maximizar eficiencia.</p>
<p>State</p>	<p>En flutter tenemos dos tipos de estado para un widget.</p> <ul style="list-style-type: none"> • Stateful widget. • Stateless widget. <p>En nuestra aplicación usamos ambos. El estado stateful refleja un widget con un ámbito cambiante, es decir es dinámico. Mientras stateless nos refleja un comportamiento más estático.</p>

Estructura

Se especifica la estructura en cómo está organizada nuestra aplicación en un ámbito de archivos, esto con el fin de proveer un entendimiento más profundo a cómo se desarrolló la misma.

Vistas	Las vistas proveen las funciones visuales de la aplicación. Estas están ubicadas bajo la carpeta lib/screens .
Elementos	Hacen referencia a los archivos externos de imagen que se utilizan a lo largo de la aplicación. Estos están ubicados bajo la carpeta lib/assets .
Widgets	Un widget es un elemento programado que provee cierta funcionalidad al estar en pantalla. Flutter provee la mayoría de manera default pero existen algunos programados a mano. Estos están ubicados bajo la carpeta lib/widgets .
Main	Archivo principal de nuestro código, de aquí la aplicación empieza su ejecución. Se encuentra en lib/main.dart
Dependencias	<p>Algunas paqueterías externas fueron implementadas para el desarrollo de la aplicación. Entre ellas están:</p> <ul style="list-style-type: none">• firebase_auth: ^0.11.1• firebase_database: ^3.0.0• cloud_firestore: ^0.11.0+2• google_maps_flutter: ^0.4.0• carousel_pro: ^0.0.1• dio: 1.0.0• flutter_barcode_scanner: ^0.0.9• auto_size_text: ^0.0.2 <p>Para agregar o alterar dependencias se debe acceder al archivo "pubspec.yaml" y dirigirse a la sección de "dependencias".</p>

Android	Flutter nos genera un código android y ios cada vez que se compila el código. Este es código nativo y puede ser alterado manualmente si se desea. Este se encuentra ubicado en la carpeta “android” .
IOS	Flutter nos genera un código android y ios cada vez que se compila el código. Este es código nativo y puede ser alterado manualmente si se desea. Este se encuentra ubicado en la carpeta “ios” .

Dependencias del proyecto

Flutter genera código nativo a partir del código dart que escribimos por lo que las mayorías de las dependencias a utilizar son instaladas y declaradas automáticamente por el SDK al momento de compilar. Estas dependencias simplemente son agregadas al pubspec.yaml de nuestro proyecto.

```
name: memorias_urbanas
description: A new Flutter application.

# The following defines the version and build number for your application.
# A version number is three numbers separated by dots, like 1.2.43
# followed by an optional build number separated by a +.
# Both the version and the builder number may be overridden in flutter
# build by specifying --build-name and --build-number, respectively.
# In Android, build-name is used as versionName while build-number used as versionCode.
# Read more about Android versioning at https://developer.android.com/studio/publish/versioning
# In iOS, build-name is used as CFBundleShortVersionString while build-number used as CFBundleVersion.
# Read more about iOS versioning at
# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html
version: 1.0.0+1

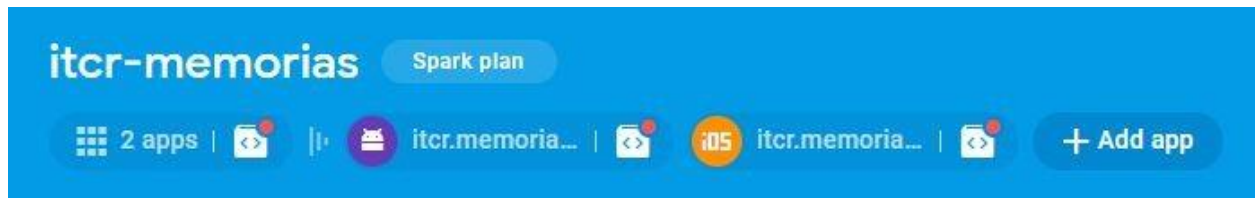
environment:
  sdk: ">=2.1.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter
  firebase_auth: ^0.11.1
  firebase_database: ^3.0.0
  cloud_firestore: ^0.11.0+2
  google_maps_flutter: ^0.4.0
  carousel_pro: ^0.0.1
  dio: 1.0.0
  flutter_barcode_scanner: ^0.0.9
  auto_size_text: ^0.0.2

# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^0.1.2

dev_dependencies:
```

Dicho esto, para este proyecto se requieren hacer ciertos ajustes manuales para la integración de firebase al sistema. Estos ajustes se encuentran explicados de manera detallada en la documentación de firebase al momento de enlazar una aplicación con firebase (<https://console.firebase.google.com>).



×

Add Firebase to your Android app

1

Register app

Android package name ⓘ

com.company.appname

App nickname (optional) ⓘ

My Android App

Debug signing certificate SHA-1 (optional) ⓘ

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

2

Download config file

3

Read the Get Started Guide for Android

× Add Firebase to your iOS app

1

Register app

iOS bundle ID ⓘ

App nickname (optional) ⓘ

App Store ID (optional) ⓘ

Register app

2

Download config file

3

Read the Get Started Guide for iOS

Implementación de la plataforma web

A través de esta sección se brinda una explicación de las características del diseño del código fuente utilizado para la generación de la aplicación web a través de un proyecto de aplicación utilizando alguno de los IDEs y herramientas mencionadas en la sección “Requerimientos de Entorno de Desarrollo”.

React JS

React es una librería de javascript enfocada en el desarrollo de interfaces de usuario para aplicaciones web. La cual permite manejar de forma ordenada y concisa nuestra aplicación, dándonos la capacidad de modular la estructura en componentes diversos y reusables.

Estructura

Se especifica la estructura en cómo está organizada nuestra aplicación en un ámbito de archivos, esto con el fin de proveer un entendimiento más profundo a cómo se desarrolló la misma.

Componentes	Un componente hace referencia a un elemento reusable dentro de la aplicación, así como botones o iconos.
Configuración	Archivo que establece la configuración de conexión con la base de datos en firebase.
Páginas	Cada página establecida dentro de nuestro ambiente web.
Assets	Referencias externas de aspectos visuales como archivos svg.
Dependencias	Todas las dependencias en React se encuentran dentro del archivo package.json y están son manejadas e instaladas mediante la línea de comandos de npm.
Index	Punto de entrada de la aplicación, React recolecta todo el código y lo exporta bajo un solo archivo de entrada.

Dependencias del proyecto

Las siguientes librerías / dependencias son las establecidas y utilizadas dentro de nuestra aplicación web.

React JS	Librería utilizada para la creación de la interfaz de nuestra aplicación.
Material UI	Provee elementos visuales ya existentes que cumplen con la línea de diseño de material.
Firebase API	API utilizado para el acceso y manejo de datos dentro de la base de datos de firebase.

