

MANUAL TECNICO
PROTOTIPO DE SOFTWARE PARA LA GESTIÓN DE RESERVAS DE CANCHAS
SINTÉTICAS EN MÓVILES CON ANDROID, CASO DE ESTUDIO CANCHAS SAN
PABLO.

JUAN CARLOS MÚNERA MORALES
JUAN DANIEL PÉREZ PULGARIN

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PEREIRA, RISARALDA

2018

MANUAL TECNICO
PROTOTIPO DE SOFTWARE PARA LA GESTIÓN DE RESERVAS DE CANCHAS
SINTÉTICAS EN MÓVILES CON ANDROID, CASO DE ESTUDIO CANCHAS SAN
PABLO.

JUAN CARLOS MÚNERA MORALES
JUAN DANIEL PÉREZ PULGARIN

Trabajo de grado para optar el título de Ingeniero de Sistemas y Computación.

Docente
CARLOS ALBERTO OCAMPO SEPÚLVEDA, I.S

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PEREIRA, RISARALDA

2018

Tabla de contenido

INTRODUCCIÓN	4
OBJETIVOS.....	5
1. REQUERIMIENTOS TÉCNICOS.....	6
2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DE LA APLICACIÓN MÓVIL.....	7
3. CASOS DE USO.....	10
4. MODELO ENTIDAD RELACIÓN.....	15

INTRODUCCIÓN

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación del aplicativo.

Es importante tener en cuenta que en el presente manual se hace mención de las especificaciones mínimas de hardware y software para la correcta instalación de la aplicación móvil.

OBJETIVOS

Brindar la información necesaria para poder realizar la instalación y configuración del aplicativo.

- Representar la funcionalidad técnica de la estructura, diseño y definición del aplicativo.
- Detallar la especificación de los requerimientos de Hardware y Software necesarios para la instalación de la aplicación.
- Describir las herramientas utilizadas para el diseño y desarrollo del prototipo

1. REQUERIMIENTOS TÉCNICOS.

REQUERIMIENTOS MÍNIMOS DE HARDWARE

- **Procesador:** Dual Core 1.1 GHZ.
- **Memoria RAM:** Mínimo: 1 Gigabytes (GB).
- **Memoria Interna:** Mínimo 1 Gigabytes (GB).

REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- **Permisos:** Teléfono, almacenamiento, Id dispositivo
- **Sistema Operativo:** Android 4.0 y versiones posteriores

2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO DE LA APLICACIÓN MÓVIL.

Android Studio:

Se escoge esta herramienta por ser la oficial para desarrollo de aplicaciones y Android y la gran cantidad de información de ayuda que se puede encontrar de ella.

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA . Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de apps para Android, 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++ y NDK
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine

IntelliJ IDEA es un entorno de desarrollo integrado(IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está

disponible en dos ediciones: edición para la comunidad y edición comercial. IntelliJ IDEA no está basada en Eclipse como MyEclipse u Oracle Enterprise Pack para Eclipse¹.

JAVA:

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible.

Java es un lenguaje orientado a objetos, esto implica que su concepción es muy próxima a la forma de pensar humana. También posee otras características muy importantes:

Es un lenguaje que es compilado, generando ficheros de clases compilados, pero estas clases compiladas son en realidad interpretadas por la máquina virtual java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando.

Es un lenguaje multiplataforma: El mismo código java que funciona en un sistema operativo funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java.

Es un lenguaje seguro: La máquina virtual al ejecutar el código java realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

Gracias al API de java podemos ampliar el lenguaje para que sea capaz de, por ejemplo, comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas, crear aplicaciones visuales al estilo window,

Para poder trabajar con java es necesario emplear un software que permita desarrollar en java, existen varias alternativas comerciales en el mercado: JBuilder, Visual Age, Visual Café, y un conjunto de herramientas shareware e incluso freeware que permiten trabajar con java. Pero todas estas herramientas en realidad se basan en el uso de una herramienta

¹ <https://developer.android.com/studio/intro/?hl=es-419>

proporcionada por Sun, el creador de java, que es el Java Development Kit (JDK). Nosotros nos centraremos en el uso de dicha herramienta².

SQLite:

Es un ligero motor de bases de datos de código abierto, que se caracteriza por mantener el almacenamiento de información persistente de forma sencilla.

A diferencia de otros Sistemas gestores de bases de datos como MySQL, SQL Server y Oracle DB, SQLite tiene las siguientes ventajas:

No requiere el soporte de un servidor: SQLite no ejecuta un proceso para administrar la información, si no que implementa un conjunto de librerías encargadas de la gestión.

No necesita configuración: Libera al programador de todo tipo de configuraciones de puertos, tamaños, ubicaciones, etc.

Usa un archivo para el esquema: Crea un archivo para el esquema completo de una base de datos, lo que permite ahorrarse preocupaciones de seguridad, ya que los datos de las aplicaciones Android no pueden ser accedidos por contextos externos.

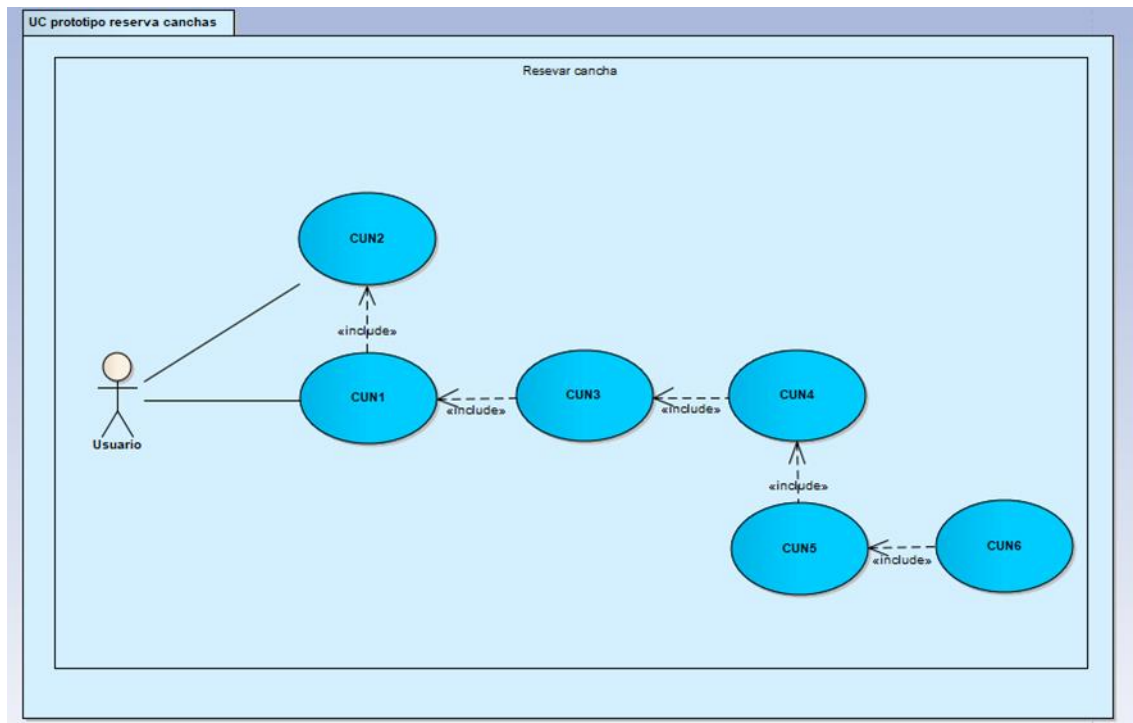
Es de Código Abierto: Está disponible al dominio público de los desarrolladores al igual que sus archivos de compilación e instrucciones de escalabilidad.

Es por eso que SQLite es una tecnología cómoda para los dispositivos móviles. Su simplicidad, rapidez y usabilidad permiten un desarrollo muy amigable³.

² https://es.wikipedia.org/wiki/Java_

³ <http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/>

3. CASOS DE USO



Especificación y modelado de los casos de uso

Definición de los actores:

ACTOR	ACT1 "Usuario Cliente"
DESCRIPCIÓN	Es el usuario principal del sistema, ingresa y gestiona la información que será procesada por el sistema.
RESPONSABILIDADES	<ul style="list-style-type: none"> • Uso general del aplicativo

Definición de los casos de uso:

1.

CASO DE USO	"Inicio de sesión"
ID CASO DE USO	CUN1
ACTOR	ACT1 <Usuario Cliente>
DESCRIPCIÓN	En este caso de uso se presenta una interfaz sencilla en el cual el usuario escribe su cuenta y contraseña para ingresar al sistema.

FLUJO BÁSICO	
Sistema	Usuario
1. El sistema permite visualizar dos campos de texto en el cual se ingresa el usuario y la contraseña. 3. El sistema permite elegir la opción de registro o de regresar si es necesario. Fin del Caso de Uso.	2. El usuario escribe sus datos en los campos.
FLUJOS ALTERNOS	
Sistema	Usuario
1. El sistema falla en el servidor. 4. No hay red disponible.	2. El usuario resetea la aplicación. 3. El usuario no puede hacer nada.
PRECONDICIONES	Usuario registrado
POSTCONDICIONES	Ha iniciado sesión exitosamente
PUNTOS DE INCLUSIÓN	CUN2 “Registrar usuario”
PUNTOS DE EXTENSIÓN	NO.

2.

CASO DE USO	“Registrar usuario”
ID CASO DE USO	CUN2
ACTOR	ACT1 <Usuario Cliente>
DESCRIPCIÓN	Este caso de uso le presenta al usuario un formulario el cual debe completar en todos los campos para realizar un registro exitoso.
FLUJO BÁSICO	
Sistema	Usuario
1. El sistema dispara un formulario con diferentes campos vacíos a llenar. 4. El sistema establece por defecto “valle del cauca” como locación estándar por el caso de estudio planteado en el trabajo de grado.	2. El usuario llena el formulario con sus datos. 3. El usuario acepta términos y condiciones.
FLUJOS ALTERNOS	
Sistema	Usuario
1. El sistema falla en el servidor. 4. No hay red disponible.	2. El usuario resetea la aplicación. 3. El usuario no puede hacer nada.
PRECONDICIONES	Estar en la sección de registro.
POSTCONDICIONES	Usuario registrado exitosamente.
PUNTOS DE INCLUSIÓN	NO
PUNTOS DE EXTENSIÓN	NO

3.

CASO DE USO	“Seleccionar sede”
-------------	--------------------

ID CASO DE USO	CUN3
ACTOR	ACT1 <Usuario Cliente>
DESCRIPCIÓN	Este caso de uso le presenta al usuario un enlace de la cancha registrada, en nuestro caso “San pablo” sede Unión.
FLUJO BÁSICO	
Sistema	Usuario
1. El sistema muestra la locación de la cancha y su correspondiente dirección.	2. El usuario elije la cancha de su preferencia. Fin del Caso de uso.
FLUJOS ALTERNOS	
Sistema	Usuario
1. El sistema falla en el servidor. 4. No hay red disponible.	2. El usuario resetea la aplicación. 3. El usuario no puede hacer nada.
PRECONDICIONES	Inicio de sesión.
POSTCONDICIONES	CUN4 “Elegir cancha”
PUNTOS DE INCLUSIÓN	CUN1 “Iniciar sesión”
PUNTOS DE EXTENSIÓN	NO.

4.

CASO DE USO	“Elegir cancha”
ID CASO DE USO	CUN4
ACTOR	ACT1 <Usuario Cliente>
DESCRIPCIÓN	Este caso de uso le presenta al usuario los datos correspondientes a la sede elegida previamente y sus correspondientes escenarios deportivos. El usuario deberá brindar información de la fecha que tiene previsto realizar la reserva.
FLUJO BÁSICO	
Sistema	Usuario
1. El sistema necesita capturar la fecha en la cual se espera reservar la cancha. 4. El sistema permite visualizar dos escenarios “Cancha principal” y “Cancha auxiliar”.	2. El usuario ingresa la fecha. 3. El usuario elije cuál de los campos presentados desea reservar. Fin de caso de Uso.
FLUJOS ALTERNOS	
Sistema	Usuario
1. El sistema falla en el servidor. 4. No hay red disponible.	2. El usuario resetea la aplicación. 3. El usuario no puede hacer nada.
PRECONDICIONES	Inicio de sesión, CUN3 “Seleccionar sede”.
POSTCONDICIONES	CUN5 “Seleccionar hora”
PUNTOS DE INCLUSIÓN	CUN3 “Seleccionar sede”
PUNTOS DE EXTENSIÓN	NO

5.

CASO DE USO	“Seleccionar hora”
ID CASO DE USO	CUN5
ACTOR	ACT1 <Usuario Cliente>
DESCRIPCIÓN	En este caso de uso se le da la opción de elegir al usuario la hora que desee hacer la reserva y el mismo usuario puede identificar si hay disponibilidad del campo.
FLUJO BÁSICO	
Sistema	Usuario
1. El sistema permite visualizar las opciones que hay respecto a los horarios que maneja la cancha San Pablo, su precio y su disponibilidad.	2.El usuario elige la hora que desee. Fin del caso de uso.
FLUJOS ALTERNOS	
Sistema	Usuario
1. El sistema falla en el servidor. 4. No hay red disponible.	2. El usuario resetea la aplicación. 3. El usuario no puede hacer nada.
PRECONDICIONES	Inicio de sesión, CUN4 “Elegir cancha”.
POSTCONDICIONES	CUN6 “Realizar reserva”
PUNTOS DE INCLUSIÓN	CUN4 “Elegir cancha”
PUNTOS DE EXTENSIÓN	NO

6.

CASO DE USO	“Aceptar reserva”
ID CASO DE USO	CUN6
ACTOR	ACT1<Usuario Cliente>
DESCRIPCIÓN	Este caso de uso el usuario visualiza un resumen de la cancha, hora elegida con su correspondiente precio y disponibilidad.
FLUJO BÁSICO	
Sistema	Usuario
1.El sistema permite visualizar las opciones “aceptar” o “cancelar” correspondientes a la reserva. 3.El sistema notifica al usuario que se ha hecho la gestión, agenda la reserva de la cancha y su disponibilidad cambia. Fin del caso de uso	2.El usuario acepta la reserva.
FLUJOS ALTERNOS	
Sistema	Usuario
1. El sistema falla en el servidor. 4. No hay red disponible.	2. El usuario resetea la aplicación. 3. El usuario no puede hacer nada.

PRECONDICIONES	Inicio de sesión, CUN5 “Seleccionar hora”
POSTCONDICIONES	Reserva realizada con éxito.
PUNTOS DE INCLUSIÓN	CUN5 “Seleccionar hora”
PUNTOS DE EXTENSIÓN	NO

4. MODELO ENTIDAD RELACIÓN

