



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN  
LICENCIATURA EN INFORMÁTICA



# **PROGRAMACIÓN DE DISPOSITIVOS MÓVILES**

**Profesor:** Hugo Díaz García

**Grupo:** 2696

**Equipo: DaSCap Desarrollo**  
Pastrana Martínez Carlos Alan  
Silva Aguilar Daniel Omar

**Fecha de entrega:** 27 de abril de 2020

## **CAPITAL EVENTOS APP**

**Primera entrega**

## Contenido

---

Descripción de la aplicación .....	3
Nombre de la aplicación: Capital Eventos App .....	3
Objetivo General de la aplicación .....	3
Objetivos específicos .....	3
Público al que va dirigido .....	4
Especificaciones técnicas .....	4
SDK Android .....	4
Modelo Vista Controlador .....	5
Android Studio .....	5
Kotlin .....	6
Descripción de la plataforma .....	6
Requerimientos mínimos .....	6
Equipo de trabajo .....	7
Nombre del equipo .....	7
Tabla de Roles .....	7
Plan de trabajo .....	8

## Descripción de la aplicación

---

### Nombre de la aplicación: Capital Eventos App

El objetivo del presente proyecto es crear una aplicación para dispositivos móviles con sistema operativo Android, que proporcione una herramienta práctica para la solicitud y programación de servicios de una agencia productora de eventos empresariales y sociales llamada Capital Eventos.



Capital Eventos App, tendrá un catálogo en el cual se mostrará una breve descripción de los diferentes servicios que ofrece la agencia, así como imágenes de estos. Además, mediante un formulario, la aplicación obtendrá información del usuario que haya solicitado alguna cotización, misma que se enviará vía correo electrónico para que el equipo de ventas pueda responder, de forma personalizada, los mensajes de los clientes. La aplicación contará con una opción para enlazar al usuario a la página web de la agencia.

### Objetivo general de la aplicación

El objetivo general de este proyecto es el desarrollo de una aplicación para dispositivos móviles (Capital Eventos App) que sirva de herramienta al equipo de ventas de la agencia Capital Eventos.

### Objetivos específicos

- Desarrollar una aplicación móvil que contenga un catálogo de los diferentes servicios que brinda la agencia.
- Mostrar, en cada uno de los servicios, la descripción y detalle de cada uno.
- Realizar un formulario que permita obtener la información del usuario que requiere una cotización.

- La información obtenida será enviada vía correo electrónico al equipo de ventas de la agencia para que ellos puedan dar seguimiento a las cotizaciones recibidas.

## Público al que va dirigido

Capital Eventos App, está dirigida principalmente a los clientes de la agencia Capital Eventos, pero también a los clientes potenciales que pudieran requerir los servicios que ofrece la agencia.

## Especificaciones técnicas

---

En este apartado, se expondrán las especificaciones técnicas de la aplicación; se explicará brevemente el funcionamiento del Modelo Vista Controlador, a continuación, se indicará el Entorno de Desarrollo Integrado (IDE), Android Studio, el cual será manejado para la realización de este proyecto. Posteriormente, se señalarán algunas ventajas de la utilización del lenguaje de programación Kotlin y, por último, se mencionará la versión mínima de Android para la que será desarrollada nuestra aplicación.

## SDK Android

El SDK de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales. Las plataformas de desarrollo soportadas incluyen GNU/Linux, Mac OS X 10.5.8 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo (IDE) soportada oficialmente es Android Studio.

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. Para nuestro desarrollo, se empleará Android SDK Platform-Tools (revision: 30.0.0).

## Modelo Vista Controlador

El patrón de diseño Modelo Vista Controlador, también llamado diseño MVC, está orientado a objetos dentro de la programación de las aplicaciones móviles; este permite separar la programación de la aplicación en tres capas, es decir, todos los objetos de nuestra aplicación deben caer en una de las tres categorías: ser un objeto del modelo (de datos) o de la vista o de un controlador.

Un objeto del modelo guarda tanto los datos de la aplicación como la denominada “Lógica del Negocio”. Su único propósito consiste en almacenar y gestionar los datos. En aplicaciones Android, las clases del modelo suelen ser creadas desde cero por el desarrollador. Todos los objetos del modelo conforman la Capa de Modelo de Datos.

Una regla simple dice que cualquier cosa que veamos en pantalla es una vista. El conjunto de objetos vista de una aplicación compone la Capa de Presentación o Visualización.

Los controladores se diseñan para que puedan responder a los distintos eventos originados por los objetos de vista y gestionar el flujo de datos entre los objetos del modelo de datos y la capa de visualización.

En resumen, utilizar el patrón de diseño MVC, nos permite separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres capas diferentes, con lo que logramos una fácil distribución del trabajo, una mejor reutilización de los componentes y un mantenimiento más sencillo.



## Android Studio



Para la realización de nuestro proyecto, se utilizará Android Studio, que es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android, basado en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece incluso más funciones que aumentan su productividad.

Android Studio es un sistema de compilación flexible basado en Gradle, contiene un emulador rápido y cargado de funciones, posee un entorno unificado donde puedes desarrollar para todos los dispositivos Android, proporciona una gran variedad de marcos de trabajo y herramientas de prueba, así como herramientas de Lint para identificar problemas de rendimiento, usabilidad y compatibilidad de la versión, entre otros.

Específicamente, se utilizará la versión 3.6, debido a que es la última versión estable, lanzada en febrero de 2020.

## Kotlin



Kotlin, es un lenguaje de programación moderno de tipo estático que aumenta la productividad. Desde su creación en 2011, Kotlin se ha desarrollado continuamente, no solo como lenguaje, sino como un ecosistema completo con herramientas robustas. Ahora, está perfectamente integrado en Android Studio y muchas empresas lo utilizan activamente para desarrollar aplicaciones de Android.

Por esta razón, utilizaremos como lenguaje de programación Kotlin en su versión 1.3.72, que es la última publicada hasta el momento de la realización de este documento.

## Descripción de la plataforma

---

## Requerimientos mínimos

La aplicación será desarrollada para dispositivos móviles con sistema operativo Android con los siguientes requerimientos mínimos:

- Android 5.0 (Lollipop)
- 100 MB de almacenamiento interno
- Conectividad a internet
- Pantalla:  
Tablet: 10.1" (para mayor comodidad del usuario)  
Smartphone: >= 5" (para mayor comodidad del usuario)

## Equipo de trabajo

---

### Nombre del equipo

DaSCaP Desarrollo, está integrado por Daniel Omar Silva Aguilar y Carlos Alan Pastrana Martínez, ambos estudiantes de sexto semestre de la Licenciatura en Informática de la Facultad de Contaduría y Administración de la Universidad Nacional Autónoma de México.



### Tabla de Roles

Administrador del proyecto	Daniel Omar Silva Aguilar
Diseñadores	Carlos A. Pastrana M. y Daniel O. Silva A.
Desarrolladores	Carlos A. Pastrana M. y Daniel O. Silva A.

## Plan de trabajo

Actividad	Meses												Responsable
	Marzo			Abril			Mayo						
1.Estudio Preliminar													Daniel Silva y Carlos Pastrana
2.Estudio de Factibilidad													Daniel Silva y Carlos Pastrana
3. Determinación del nombre													Daniel Silva y Carlos Pastrana
4. Planteamiento de los objetivos													Daniel Silva y Carlos Pastrana
5. Establecimiento de los requisitos mínimos													Daniel Silva y Carlos Pastrana
6. División de actividades por integrante													Daniel Silva y Carlos Pastrana
7. Determinación de especificaciones técnicas													Daniel Silva y Carlos Pastrana
8. Análisis de Requerimientos													Daniel Silva y Carlos Pastrana
9. Diseño del sistema													Daniel Silva y Carlos Pastrana
9.1. Colores utilizar.													Daniel Silva y Carlos Pastrana
9.2. Fuentes a utilizar.													Daniel Silva y Carlos Pastrana
9.3. Logotipo.													Daniel Silva y Carlos Pastrana
9.4. Mapa de navegación.													Daniel Silva y Carlos Pastrana
9.5. Bocetos (Wireframes).													Daniel Silva y Carlos Pastrana
9.6. Guías de diseño que se aplican.													Daniel Silva y Carlos Pastrana
10. Desarrollo													Daniel Silva y Carlos Pastrana
11. Implementación													Daniel Silva y Carlos Pastrana
11.1. Descripción de Módulos													Daniel Silva y Carlos Pastrana
11.2. Lista de archivos a utilizar													Daniel Silva y Carlos Pastrana
11.3. Generación de archivos													Daniel Silva y Carlos Pastrana