



Proyecto Final

Clave	Unidad de Aprendizaje
I4893	Programación para Dispositivos Móviles I
Nombre y contacto del profesor	
Dr. César Alejandro García García cesar.ggarcia@academicos.udg.mx	
Unidad, tema y subtema	
Proyecto Final	
Otros temas que refuerza	
2.11 Uso de vistas y layouts 5.3 Uso del GPS	3.5 Intenciones 6.5 Bases de datos
Objetivo General	
Diseña e implementa sistemas de información en múltiples plataformas y entornos operativos	
Objetivo Específico	
Que el alumno adquiera los conocimientos para diseñar, implementar y publicar una aplicación para dispositivo móvil utilizando la plataforma Android Studio	

Instrucciones
<p>Construir una aplicación para manejo de dinero que cuente con las siguientes características:</p> <ul style="list-style-type: none"> • La aplicación debe permitir la creación de cuentas en efectivo, débito y crédito, debiendo siempre existir al menos una cuenta en efectivo. • La aplicación debe permitir la creación de transacciones, cada una asociada a una cuenta y categoría de gasto, que además debe incluir de manera obligatoria cantidad, fecha, hora, y ubicación del GPS; y opcionalmente campos para concepto, beneficiario y notas, editables por el usuario. • La aplicación debe tener al menos tres categorías predefinidas, como por ejemplo: Alimentos, Transporte, Tecnología, etc., para los gastos, permitiendo al usuario la creación de nuevas categorías siempre y cuando no se repitan. • La aplicación debe permitir crear transacciones futuras en las cuentas que no sean de efectivo. Por ejemplo, que cada día 14 del mes se haga el mismo cargo. • La aplicación debe permitir dividir transacciones en parcialidad iguales, solamente para las cuentas de crédito, solamente en partes iguales y siempre recurrentes para el mismo día de cada mes. • La aplicación debe mantener en todo momento un balance de cada cuenta, así como uno general de todas ellas.



- La aplicación debe contar con al menos los siguientes reportes: gasto mensual por categoría, gasto global del mes, gasto por cuenta y categoría en un periodo indicado por el usuario y, dos reportes adicionales creados por los desarrolladores.
- Aspectos técnicos que incluyen el correcto funcionamiento de la aplicación en un dispositivo Android.

Historias de Usuario

Parte 1: Interfaz de usuario

1. El usuario debe tener siempre visible el saldo actual de la cuenta en la que se encuentra.
2. El usuario debe contar con un menú para cambiar de una cuenta a otra.
3. El usuario siempre debe estar en contexto de los gastos, por lo que todos los cargos, abonos y saldos deben tener el símbolo de moneda: \$.

Parte 2: Conexión con la base de datos

1. El usuario debe tener información persistente, que se guarde en la base de datos, para poder acceder a ella cuando vuelva a abrir la aplicación.

Parte 3: Manejo de cuentas

1. El usuario debe poder crear cuentas y, al hacerlo, se captura el nombre, tipo de cuenta (efectivo, crédito, débito) y el saldo inicial.
2. El usuario debe poder eliminar una cuenta y, al hacerlo, todas las transacciones asociadas a ella se deben también eliminar.
3. El usuario no debe poder eliminar la cuenta de efectivo cuando ésta la única cuenta en efectivo.
 - a. Ejemplo: El usuario tiene las cuentas de efectivo “Cartera” y “Puerquito”. El usuario elimina la cuenta “Puerquito” y por ende, las transacciones asociadas a ella. Sin embargo, después de haberla eliminado no podrá eliminar la cuenta “Cartera” al ser la única en efectivo.

Parte 4: Manejo de transacciones

1. El usuario debe poder crear transacciones y, al hacerlo, se capturan de forma obligatoria la cuenta cuyo saldo se afecta, la categoría a la que pertenece, el monto, la



fecha y hora, y la ubicación del GPS. Opcionalmente el usuario puede capturar concepto, beneficiario y notas.

- a. Las transacciones deben forzosamente pertenecer a una cuenta y una categoría.
2. El usuario debe poder crear transacciones recurrentes, que se programan siempre en la misma fecha de cada mes y por la misma cantidad.
 - a. Ejemplo: El día 12 de cada mes, la cuenta “Débito” recibe un cargo por \$ 149 en la categoría “Servicios” y con el concepto “Netflix”.
 - b. Ejemplo: El día 30 de cada mes, la cuenta “Débito” recibe un abono por \$ 20,000 en la categoría “Ingresos” y con el concepto “Salario”.
3. El usuario debe poder crear transacciones a plazos solamente en aquellas cuentas que sean de crédito. Estas transacciones dividen su cargo en 12 partes iguales que se programan en la misma fecha de cada mes y por la misma cantidad.
 - a. Ejemplo: El usuario compra un producto por \$ 12,000 a doce mensualidad el día 15 de octubre, ese mismo día se hace un cargo por \$ 1,000 y se programan otros 11 cargos iguales para los días 15 de noviembre, diciembre, enero, etc. hasta haber completado los 12 cargos.
4. El usuario debe poder crear categorías para las transacciones, las cuales tienen un nombre y si son cargos o abonos. Los cargos siempre reducen el saldo y los abonos siempre lo incrementan. Todas las transacciones de la misma categoría son siempre cargos o siempre abonos.
5. El usuario debe poder crear transacciones entre cuentas, denominadas transferencias, que se cargan en la cuenta origen y se abonan en la cuenta destino.
6. El usuario debe poder eliminar una categoría y, al hacerlo, todas las transacciones asociadas a ella se deben conservar, asociándose a una categoría “Desconocido” o “Default”.

Parte 5: Reportes

1. El usuario debe poder crear un reporte de gasto mensual, que incluye todas las categorías y todas las cuentas, así como la suma total de cargos y abonos de ese mes por categoría.
2. El usuario debe poder crear un reporte por cuenta de todo el mes, e incluye la suma total de cargos o abonos para ese mes en cada categoría.
3. El usuario debe poder crear un reporte por cuenta, donde se le pida una fecha inicial y una final, e incluye la suma total de cargos o abonos para ese periodo en cada categoría.
4. El usuario debe tener disponibles otros dos reportes diferentes a los antes mencionados, creados por los desarrolladores.

Parte 6: Extras

1. Se bonificará con 1 punto adicional a la rúbrica la creación de categorías ocultas que no



- aparecen en los reportes.
2. Se bonificará con 1 punto adicional a la rúbrica si se permite al usuario definir su propio símbolo de moneda y este cambio se propaga adecuadamente en toda la aplicación.
 3. Se bonificará con 1 punto adicional a la rúbrica si se implementan patrones de diseño para la interfaz de usuario.
 4. Se bonificará con 2 puntos adicionales a la rúbrica la sincronización de datos con algún servicio de almacenamiento en la nube, tales como Google Drive, Dropbox o OneDrive.
 5. Se bonificará con 2 puntos adicionales a la rúbrica si se implementa un mapa donde aparezcan las transacciones.

Resultados

Entregables
<ol style="list-style-type: none"> 1. Presentar la aplicación funcionando al profesor 2. Entregar de forma electrónica el documento que detalla los aspectos aquí mencionados 3. Entregar de forma electrónica la carpeta del proyecto de Android Studio, junto con el archivo APK del juego

Este proyecto se compone de 18 puntos posibles y 7 adicionales, donde cada uno equivale al 5.55% de la calificación. La calificación máxima posible es 138, si se alcanzan todos los puntos posibles, así como los adicionales.

Rúbrica para la evaluación del proyecto					
PUNTOS	4	3	2	1	0
Criterio					
Puntualidad	Entrega antes del 20/11/18	Entrega antes del 21/11/18	Entrega antes del 23/11/18	Entrega antes del 28/11/18	Fechas posteriores
Cuentas	N/A	N/A	El manejo de cuentas es el apropiado	Crea cuentas pero no funcionan de la manera esperada	No se pueden crear cuentas
Transacciones	N/A	Crea transacciones y funcionan como se espera	Crea transacciones pero existe un problema al eliminar cuentas o	Crea transacciones pero no funcionan de la manera esperada	No se pueden crear o no se calcula correctamente el saldo



			categorias		
Persistencia de los datos	La persistencia se implementa correctamente	N/A	N/A	N/A	No se conecta a la BD
Interfaz gráfica	N/A	La interfaz es funcional, estética e intuitiva	La interfaz es funcional y estética	La interfaz es funcional	Contiene elementos que no responden
Ubicación por GPS	N/A	N/A	N/A	Agrega la ubicación a las transacciones	No agrega la ubicación a las transacciones
Reportes	N/A	N/A	Implementa los tres reportes solicitados y otros dos de diseño propio	Sólo implementa los tres reportes descritos en el documento	No tiene reportes o no se calcula correctamente el saldo
Dispositivo Android	N/A	N/A	Funciona de manera apropiada en un dispositivo Android	Funciona en el dispositivo pero alguno de los controles falla o interfiere	No funciona en dispositivo Android

Referencias y material adicional

- Ambler, Scott W. (2018). "User Stories: An Agile Introduction." Disponible en línea: <http://www.agilemodeling.com/artifacts/userStory.htm>
- Google Play Store (2018). "Money Lover: Budget App & Expense Tracker". Disponible en línea: <https://play.google.com/store/apps/details?id=com.bookmark.money>
- Google Play Store (2018). "1Money - expense tracker, money manager, budget". Disponible en línea: <https://play.google.com/store/apps/details?id=org.pixelrush.moneyig>
- Android Studio Developers (2018). "Localize the UI with Translations Editor". Disponible en línea: <https://developer.android.com/studio/write/translations-editor>
- Mockplus (2018). "Android UI Design Patterns - 10 Best Mobile App UI Designs in Android". Disponible en línea: <https://www.mockplus.com/blog/post/android-design-patterns>