



**INSTITUTO TECNOLÓGICO DE CD. GUZMÁN**

**TITULACIÓN INTEGRAL**

**TESIS**

**TEMA:**

**PLATAFORMA DE GESTIÓN PARA HUERTAS AGUACATERAS**

**QUE PARA OBTENER EL TÍTULO DE:**

**INGENIERO INFORMÁTICO**

**PRESENTA:**

**JONATHAN OSORNIO MENDOZA**

**14290766**

**ASESORA:**

**M.C. RAQUEL OCHOA ORNELAS**

**CD. GUZMÁN JALISCO, MÉXICO, AGOSTO DE 2019**



Cd. Guzmán, Municipio de Zapotlán el Grande, Jal. **03/09/2019**

**ASUNTO:** Liberación de Proyecto para Titulación integral

**M.C. FAVIO REY LÚA MADRIGAL**  
**JEFE DE LA DIVISION DE ESTUDIOS PROFESIONALES**  
Presente

Por este medio informo que ha sido liberado el siguiente proyecto para la Titulación integral:

Nombre del Egresado	<b>Jonathan Osornio Mendoza</b>
Carrera:	Ingeniería Informática
No. De Control	<b>14290766</b>
Nombre del proyecto:	"PLATAFORMA DE GESTIÓN PARA HUERTAS AGUACATERAS"
Producto	<b>TESIS</b>

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados.

Atentamente,

  
M.S.C. Omar Cristian Vargas Gonzalez  
Jefe del Depto. Sistemas y Computación



SEP TecNM  
I. T. DE CD. GUZMÁN  
SISTEMAS Y  
COMPUTACIÓN

M.C. Raquel Ochoa Ornelas  Nombre y Firma del Asesor	Dr. Daniel Fajardo Delgado  Nombre y Firma del Revisor	Dra. María Guadalupe Sánchez Cervantes  Nombre y Firma del Revisor
---	---	---

MIAH/CRM/DCVG/ljss



## **RESUMEN**

En el desarrollo del proyecto se utilizaron servidores en la Web, que brindan el espacio para almacenar, administrar y recuperar archivos que físicamente están en otro lugar, y se pueda acceder a ellos como si estuvieran en un dispositivo, a través de Internet.

Existen herramientas para administrar la información que se tiene en una base de datos en la Web, un ejemplo de esto es phpMyAdmin pero trabajan directamente con instrucciones SQL lo que puede resultar problemático para usuarios que no tienen conocimiento sobre estas, por lo cual, es necesario crear interfaces que sean amigables, fáciles de usar y entender por cualquier usuario. Para este proyecto se utilizó Visual Studio 2017 que es un conjunto completo de herramientas de desarrollo de software con el lenguaje Visual Basic.NET para la aplicación de escritorio. Se utilizó Atom para desarrollar la página Web mezclando diferentes lenguajes de programación, de marcado o etiquetas entre ellos HTML, JavaScript, PHP y CSS. Por último, se empleó Android Studio para desarrollar la aplicación móvil.

El proyecto “Plataforma de gestión para huertas aguacateras” está constituido por una aplicación robusta de escritorio que consta de catálogos, movimientos, reportes y consultas. Además, una página Web que incluye catálogos, consultas y reportes y una aplicación para dispositivos móviles Android con la opción de consulta de información. Estas aplicaciones conforman la plataforma con acceso a datos en tiempo real, estadísticas y demás herramientas que ayudan a la toma oportuna de decisiones dentro de la empresa.

# ÍNDICE

1. Introducción.....	1
2. Antecedentes.....	3
3. Objetivos.....	4
3.1 Objetivo general.....	4
3.2 Objetivos específicos.....	4
4. Marco teórico.....	5
4.1 Base de datos.....	5
4.1.1 Ciclo de vida de las bases de datos.....	5
4.1.2 Tablas.....	5
4.1.3 Filas (Registros).....	5
4.1.4 Columnas (Campos).....	5
4.1.5 Claves.....	6
4.2 Sistema gestor de bases de datos (SGBD).....	6
4.3 SQL.....	7
4.4 SQL Server.....	7
4.5 Instrucciones SQL.....	8
4.6 MySQL.....	10
4.7 phpMyAdmin.....	10
4.8 Lenguajes de programación.....	11
4.9 Microsoft.NET Framewok.....	11
4.10 Visual Studio.....	11
4.10.1 Características.....	12
4.11 Java.....	12
4.11.1 Clases.....	13
4.11.2 Métodos.....	13
4.11.3 Identificadores.....	13
4.11.4 Variables.....	14
4.11.5 Constantes.....	14
4.12 Android.....	15
4.12.1 Arquitectura Android.....	15
4.13 HTML.....	16
4.13.1 Sintaxis.....	17
4.13.2 Formatos.....	17
4.14 PHP.....	18
4.14.1 Integración de PHP al proyecto.....	19
4.15 JavaScript.....	19
4.16 CSS.....	20
4.16.1 Integración de CSS en el proyecto.....	20
4.17 JSON.....	21
4.17.1 Integración de JSON en el proyecto.....	21

5. Metodología.....	22
5.1 Análisis.....	22
5.1.1 Políticas .....	22
5.1.2 Almacenes de datos .....	22
5.1.3 Procesos .....	23
5.1.4 Entidades externas.....	23
5.2 Diseño.....	24
5.2.1 Diagrama de Warnier/Orr de la aplicación de escritorio .....	24
5.2.2 Diagrama de Warnier/Orr de la página Web.....	26
5.2.3 Diagrama de Warnier/Orr de la aplicación móvil.....	27
5.2.4 Diagrama de caso de uso general .....	28
5.2.5 Diagrama de clases.....	29
5.2.6 Diagramas de cortes .....	30
5.2.7 Diagramas de compras .....	34
5.2.8 Diagramas de pagos.....	38
5.2.9 Diagramas de cobros.....	42
5.2.10 Diagramas de actividades.....	46
5.3 Codificación.....	50
5.3.1 Diagrama de módulos, formularios, consultas y reportes de la aplicación de escritorio .....	50
5.3.2 Diagrama de módulos, formularios, consultas y reportes de la página Web	52
5.3.3 Diagrama de módulos, formularios y consultas de la aplicación móvil .....	55
5.4 Pruebas .....	57
5.4.1 Pruebas en la aplicación de escritorio.....	57
5.4.2 Pruebas en la página Web .....	61
5.5 Implementación.....	62
5.6 Documentación .....	62
6. Resultados .....	63
6.1 Aplicación de escritorio .....	63
6.1.1 Menú .....	63
6.1.2 Catálogos .....	64
6.1.3 Movimientos .....	65
6.1.4 Consultas.....	66
6.2 Página Web .....	67
6.2.1 Menú .....	67
6.2.2 Reportes .....	68
6.3 Aplicación móvil .....	69
6.3.1 Menú .....	69
6.3.2 Consultas.....	70
6.3.3 Selector de fecha .....	71
7. Discusión .....	72
8. Conclusiones .....	73

9. Referencias bibliográficas y virtuales .....	74
10. Anexos.....	76
10.1 Aplicación de escritorio .....	76
10.1.1 Formulario registrar cortes .....	76
10.2 Página Web.....	84
10.2.1 Formulario consulta empacadoras .....	84
10.3 Aplicación móvil.....	89
10.3.1 Activity seleccionar fecha (código java).....	89
10.3.2 Activity seleccionar fecha (código xml).....	93

## ÍNDICE DE FIGURAS

Figura 1. Arquitectura Android .....	15
Figura 2. Diagrama de Warnier/Orr de escritorio 1.....	24
Figura 3. Diagrama de Warnier/Orr de escritorio 2.....	25
Figura 4. Diagrama de Warnier/Orr Web.....	26
Figura 5. Diagrama de Warnier/Orr móvil .....	27
Figura 6. Diagrama de caso de uso general.....	28
Figura 7. Diagrama de clases.....	29
Figura 8. Diagrama de flujo de datos cortes.....	30
Figura 9. Diagrama de caso de uso expandido cortes.....	31
Figura 10. Diagrama de estado corte .....	32
Figura 11. Diagrama de colaboración cortes .....	33
Figura 12. Diagrama de flujo de datos compras.....	34
Figura 13. Diagrama de caso de uso expandido compras.....	35
Figura 14. Diagrama de estado compras .....	36
Figura 15. Diagrama de colaboración compras .....	37
Figura 16. Diagrama de flujo de datos pagos .....	38
Figura 17. Diagrama de caso de uso expandido pagos.....	39
Figura 18. Diagrama de estado pagos.....	40
Figura 19. Diagrama de colaboración pagos .....	41
Figura 20. Diagrama de flujo de datos cobros.....	42
Figura 21. Diagrama de caso de uso expandido cobros .....	43
Figura 22. Diagrama de estado cobros .....	44
Figura 23. Diagrama de colaboración cobros.....	45
Figura 24. Diagrama de flujo de datos actividades .....	46
Figura 25. Diagrama de caso de uso expandido actividades .....	47
Figura 26. Diagrama de estado actividades .....	48
Figura 27. Diagrama de colaboración actividades.....	49
Figura 28. Diagrama de catálogos de escritorio 1 .....	50
Figura 29. Diagrama de catálogos de escritorio 2 .....	50
Figura 30. Diagrama de movimientos de escritorio .....	51
Figura 31. Diagrama de consultas y reportes de escritorio.....	51
Figura 32. Diagrama de catálogos e inicio de sesión Web.....	52
Figura 33. Diagrama de consultas de catálogos Web.....	52
Figura 34. Diagrama de consultas de movimientos Web .....	53
Figura 35. Diagrama de reportes de catálogos Web.....	53
Figura 36. Diagrama de reportes de movimientos Web .....	54
Figura 37. Diagrama de consulta de movimientos móvil.....	55
Figura 38. Diagrama de inicio, ayuda y cerrar sesión móvil.....	55
Figura 39. Diagrama de consultas de catálogos móvil.....	56

Figura 40. Menú catálogos de escritorio .....	63
Figura 41. Menú movimientos de escritorio.....	64
Figura 42. Registrar empacadoras de escritorio .....	64
Figura 43. Registrar huertas de escritorio.....	65
Figura 44. Registrar cortes de escritorio.....	65
Figura 45. Registrar actividades de escritorio .....	66
Figura 46. Consulta cobros de escritorio.....	66
Figura 47. Consulta cortes de escritorio .....	67
Figura 48. Menú consultas.....	67
Figura 49. Menú reportes .....	68
Figura 50. Reporte empacadoras .....	68
Figura 51. Reporte cortes.....	69
Figura 52. Menú catálogos móvil.....	69
Figura 53. Consulta actividades móvil .....	70
Figura 54. Consulta empacadoras móvil .....	70
Figura 55. Consulta pagos móvil.....	71
Figura 56. Selección de fecha.....	71



## ÍNDICE DE TABLAS

Tabla 1. Prueba registrar corte 1.....	57
Tabla 2. Prueba registrar corte 2.....	58
Tabla 3. Prueba registrar compra 1.....	59
Tabla 4. Prueba registrar compra 2.....	60
Tabla 5. Prueba registrar producto 1 .....	61
Tabla 6. Prueba registrar producto 2 .....	62

# 1. INTRODUCCIÓN

CEDRSSA (2017) en su reporte titulado “Caso de Exportación: El Aguacate”: afirma que las empresas productoras de aguacate en México han crecido de manera acelerada en los últimos años, convirtiéndose en una de las principales fuentes de ingreso y trabajo en el país, exportando aguacate a diversos países principalmente a China, Australia y Chile. Estados Unidos es el principal importador de aguacate mexicano, seguido por Francia, Japón y Canadá.

La mayoría de estas empresas han manejado mucha información de forma manual, teniendo como consecuencia grandes cantidades de papel que no se resguarda en un lugar seguro, pudiendo sufrir algún tipo de daño, aunque también se utilizan archivos digitales de Excel exponiéndose a sufrir daños o pérdida de información por la falta de cuidado que los archivos y documentos necesitan.

Al observar la problemática y la falta de herramientas especializadas en las necesidades de huertas aguacateras, se pensó en desarrollar una herramienta que ayudara a la gestión de la información donde se pueda registrar, consultar, además crear reportes y consultar informes para la toma oportuna de decisiones de una forma fácil que permita el crecimiento económico de la empresa. La herramienta desarrollada está hecha a la medida de las empresas aguacateras tomando en cuenta muchas características que el software similar ya existente no tiene.

En la actualidad, el uso de las tecnologías en las empresas es fundamental para lograr una estabilidad y rentabilidad en el mercado basándose en la información que éstas mismas generan, por eso se puede observar que incluso en tiendas pequeñas ya se cuenta con la ayuda de una computadora donde se puede almacenar la información sobre sus productos, compras y ventas y pedidos a proveedores, venta al público y otras cosas.

La herramienta que se desarrolló es una plataforma integrada por una aplicación robusta de escritorio que cuenta con todas las opciones para registro, consulta, creación de informes y reportes además del respaldo y restauración de la información, una página Web donde se puede registrar información además consultar y generar reportes, y una aplicación para dispositivos móviles Android que tiene opciones para consulta de información. La importancia que tuvo el desarrollar la plataforma en tres escenarios diferentes fue poder registrar y obtener la información de forma rápida, sencilla y que esté siempre disponible para el usuario. El desarrollo se llevó a cabo con ayuda de herramientas y lenguajes tales como Visual Studio 2017, HTML, JavaScript, MySQL y Android Studio.

En esta tesis se exponen las actividades realizadas durante el desarrollo de una plataforma digital y se incluyen diagramas de análisis y diseño correspondientes.

En el capítulo 2 se introduce al área en la que se tomó inspiración para el desarrollo de este trabajo, además, la motivación para llevarlo a cabo.

En el capítulo 3 se establecen los objetivos que se tomaron en cuenta y como base para desarrollar el análisis de los requerimientos y solución que se daría a partir de estos.

En el capítulo 4 se introduce a las definiciones técnicas que se mencionan en el desarrollo de este trabajo, se hace descripción y ejemplificación para que los tecnicismos sean entendidos y comprendidos, se inicia con las bases de datos, mencionando definiciones y herramientas utilizadas, posteriormente se mencionan los lenguajes de programación con ejemplos y definiciones además de las herramientas utilizadas para la codificación de la plataforma.

En el capítulo 5 se aborda la metodología utilizada en el desarrollo del proyecto, primero se expone el análisis de la problemática, las necesidades y posteriormente, se expone la forma de dar solución con diagramas que muestran cómo se conforma la base de datos, el funcionamiento de cada uno de los escenarios (de escritorio, Web y móvil) que componen la plataforma. El capítulo concluye mostrando las pruebas que se realizaron a la plataforma con sus resultados y se hace mención sobre la implementación y documentación de esta.

En el capítulo 6 se muestran los resultados con imágenes obtenidos, es decir la interfaz de la aplicación de escritorio, la página Web y la aplicación móvil además se muestran consultas y reportes como resultado de la funcionalidad de la plataforma en general.

En el capítulo 7 se aborda la problemática que se tuvo al desarrollar el proyecto y la forma de solucionarlos.

El capítulo 8 trata sobre la conclusión del desarrollador, su aprendizaje y experiencia.

En el capítulo 9 se encuentran las referencias donde se encontró información utilizada en el desarrollo de este informe.

El capítulo 10 muestra pequeños ejemplos de cómo se desarrollaron algunos formularios o pantallas de cada uno de los escenarios que integran la plataforma.

## 2. ANTECEDENTES

La SAGARPA (2017) señala que el aguacate es uno de los productos más exitosos de la exportación agroalimentaria nacional. México es el principal proveedor del mercado internacional con una aportación de 45.95% del valor de las exportaciones mundiales. En estados unidos es promovido y consumido en eventos de alcance internacional, como el Super Bowl, en el cual se han llegado a consumir 100,000 toneladas durante el día del evento.

En un artículo Web del periódico Milenio, se explica que las empresas de giro turístico y comercial son las que más han aprovechado las tecnologías y muestran un avance en este aspecto. Estudios y estadísticas sobre el uso de las Tecnologías de la Información y Comunicación (TICs) en el sector productivo agrícola del país manifiestan un rezago significativo en comparación con otros. Sin embargo, las nuevas tecnologías también han llegado a la agricultura.

El sector agrícola, tiene que apostar por aunar tradición e innovación con una única meta: afrontar los nuevos retos de adaptación y conseguir una agricultura competitiva y moderna (Santiago, Zapata y Martínez, 2015).

Investigando en Internet se encontró que existe software similar a este proyecto, algunos de estos son:

- Albor: Aplicación para smartphones que permite llevar el ciclo productivo integrado con gestión confiable y financiera, evalúa distintos escenarios y hace proyecciones operativas, reduce la complejidad en la toma de decisiones, obtiene reportes.
- Sismagro: Software agropecuario. Gestiona las tareas dentro de un calendario, agregar fácilmente datos de los insumos y herramientas, dibuja lotes.
- MiFinca: Es un instrumento para el manejo de los costos, supervisión de tareas, generador de índices de producción control de inventario de insumos, inventario físico, reportes de labores, informe de gastos administrativos.

Sin embargo, ninguno integra los tres escenarios involucrados en el proyecto desarrollado como lo son la aplicación móvil, aplicación Web y aplicación de escritorio interconectadas y trabajando en conjunto en una base de datos en Internet, brindando además diversos informes y estadísticas.

De acuerdo con el especialista en TI Sergio Márquez-Berber, catedrático de la Universidad Autónoma de Chapingo, un problema con el que deben lidiar los agricultores es la llamada “asimetría de información”. Comenta: “Los pequeños campesinos son vulnerables a fenómenos climáticos y variaciones de precios en los mercados consumidores, pero desafortunadamente tienen poco acceso a información que les pudiera ayudar a tomar medidas para proteger sus cultivos o negociar con intermediarios” (Peralta, 2016).

## **3. OBJETIVOS**

### **3.1 Objetivo General**

Desarrollar una plataforma que permita llevar la gestión de huertas aguacateras en sus operaciones diarias con empacadoras, proveedores y otras actividades propias de las huertas.

### **3.2 Objetivos Específicos**

- Solucionar problemas administrativos de contabilidad.
- Disponer de aplicaciones en diferentes escenarios (escritorio, Web, y móvil) para la gestión de las huertas.
- Tener un registro de las empacadoras y proveedores.
- Registrar datos de las huertas y sus encargados.
- Llevar el control de las actividades realizadas.
- Tener un registro de los productos que se utilizan dentro de las huertas.
- Registrar los datos de los cortes de fruta.
- Controlar la entrada de dinero (venta de fruta a empacadoras) de contado y a crédito con sus respectivos cobros.
- Controlar la salida de dinero (compras de productos a proveedores y gastos generales) de contado y a crédito con sus respectivos pagos.
- Tener actualizados los datos de las huertas, productos y demás catálogos.
- Obtener diversas consultas y reportes que ayuden a la toma oportuna de decisiones dentro de la empresa.
- Contar con herramientas de respaldo, restauración de base de datos e histórico para tener siempre los datos seguros.

## 4. MARCO TEÓRICO

Dado que este trabajo se centra en el desarrollo de software, resulta fundamental conocer algunas definiciones que se estarán utilizando en el desarrollo de este.

### 4.1 Base de datos

Silberschatz, Korth y Sudarshan (2006) definen una base de datos como una colección de datos que contiene información relevante para una empresa.

Una base de datos es una colección de datos almacenados y organizados de forma que un programa del ordenador pueda seleccionarlos rápidamente y capaces de ser recobrados, actualizados, insertados y borrados.

#### 4.1.1 Ciclo de vida de las bases de datos

El ciclo de vida de las bases de datos según Bender et al. (2014), inicia desde el análisis de las funcionalidades y datos que debe almacenar el sistema hasta llegar a la implementación, pero para llegar a este, se debe pasar por una serie de etapas que, si se llevan a cabo teniendo en cuenta parámetros de calidad, se llegará a tener una base de datos de mayor funcionalidad. Las etapas del ciclo de vida de una base de datos son:

- Análisis: Análisis de requerimientos.
- Diseño: Diseño de tablas, relaciones, llaves y atributos.
- Implementación: Implementar la base de datos, pruebas y mantenimiento.

#### 4.1.2 Tablas

Son objetos de las bases de datos que contienen información distribuida en filas (registros) y columnas (campos) (Morán, 2015).

#### 4.1.3 Filas (Registros)

Un registro es un conjunto de hechos acerca de una persona, evento o cualquier otro elemento de interés. Cada registro contiene los valores de cada campo de la tabla.

En una base de datos bien estructurada, cada registro debe ser único. Es decir, no deben existir dos o más registros que contengan exactamente la misma información (Gómez Fuentes, 2013).

#### 4.1.4 Columnas (Campos)

Gómez Fuentes (2013) describe un campo como una sola clase del hecho que se puede aplicar a cada registro. Por ejemplo, código postal puede ser un campo de una tabla Empleados. Teléfono puede ser un campo de una tabla Transportistas.

Un valor importante para los campos es el tipo de datos, que puede ser número, texto, moneda, fecha, hora, etc. El tipo de datos limita y describe la clase de información del campo. También determina las acciones que se pueden realizar en el campo y la cantidad de memoria que utilizan los datos.

#### **4.1.5 Claves**

Una clave es un atributo o conjunto de atributos cuyos valores distinguen un registro en una tabla Gómez Fuentes (2013).

##### **Clave primaria**

Gómez Fuentes (2013) define una clave primaria como un identificador único para cada registro. Nunca puede contener entradas nulas.

Para elegir un campo como clave primaria debe cumplir las siguientes características:

- Deberá seleccionarse la que ocupe un menor espacio de almacenamiento.
- Tener una codificación sencilla.
- El contenido de sus valores debe ser conocido y no variar.
- No debe tener valores nulos.
- Podrá utilizarse en otras tablas para construir interrelaciones.
- Deben ser fácilmente recordables por el usuario.
- No debe cambiar con frecuencia.

##### **Clave secundaria**

Es un atributo o combinación de estos, que se usa estrictamente para propósitos de recuperación de información. Cuando una clave está formada por un solo atributo se denomina clave simple, en caso contrario, se denomina clave compuesta (Gómez Fuentes, 2013).

##### **Clave externa**

Es la clave primaria de una tabla que se utiliza en otras tablas para crear una interrelación.

## **4.2 Sistema gestor de bases de datos (SGBD)**

Un sistema gestor de bases de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de información. Además, los sistemas de bases de datos deben proporcionar la

fiabilidad de información almacenada, a pesar de las caídas del sistema o intentos de acceso sin autorización (Silberschatz, Korth y Sudarshan, 2006).

### 4.3 SQL

Silberschatz, Korth y Sudarshan (2006) describen a SQL como un lenguaje de consultas que usa una combinación de álgebra relacional y construcciones de cálculo relacional.

Aunque el lenguaje SQL se considere un lenguaje de consultas, contiene muchas otras capacidades además de la consulta en bases de datos. Incluye características para definir la estructura de los datos, para la modificación de los datos en la base de datos y para la especificación de restricciones de seguridad.

El lenguaje SQL tiene varios componentes (Silberschatz, Korth y Sudarshan, 2006):

- Lenguaje de definición de datos (LDD). El LDD de SQL proporciona órdenes para la definición de esquemas de relación, borrado de relaciones, creación de índices y modificación de esquemas de relación.
- Lenguaje interactivo de manipulación de datos (LMD). El LMD de SQL incluye un lenguaje de consultas, basado tanto en el álgebra relacional como en el cálculo relacional de registros. Incluye también órdenes para insertar, borrar y modificar registros de la base de datos.
- Definición de vistas. El LDD de SQL incluye órdenes para la definición de vistas.
- Control de transacciones. SQL incluye órdenes para la especificación del comienzo y final de transacciones.
- SQL incorporado y SQL dinámico. SQL dinámico e incorporado definen cómo se pueden incorporar las instrucciones SQL en lenguajes de programación de propósito general.
- Integridad. El LDD de SQL incluye órdenes para la especificación de las restricciones de integridad que deben satisfacer los datos almacenados en la base de datos. Las actualizaciones que violen las restricciones de integridad se rechazan.
- Autorización. El LDD de SQL incluye órdenes para especificar derechos de acceso para las relaciones y vistas.

### 4.4 SQL Server

El SQL Server de Microsoft es un sistema de gestión de bases de datos relacionales. Proporciona servicios de duplicación entre varias copias de SQL Server con otros sistemas de bases de datos. Su componente Analysis Services (servicios de análisis), una parte esencial del sistema incluye servicios de procesamiento analítico en línea (OLAP, Online Analytical Processing) y de minería de datos. SQL server proporciona una gran colección de herramientas gráficas y de asistentes. Muchos entornos de desarrollo soportan SQL Server, incluidos Visual Studio. SQL Server proporciona un conjunto de herramientas para gestionar todos los aspectos de desarrollo, consulta, ajuste, prueba y administración de SQL Server. La mayor parte de estas



herramientas giran alrededor de Management Studio que ofrece una interfaz de comandos común para la administración de los servicios, incluye el motor de bases de datos, servicios de análisis, servicios de informes, servidor móvil y servicios de integración (Silberschatz, Korth y Sudarshan, 2006).

## 4.5 Instrucciones SQL

Según Silberschatz, Korth y Sudarshan (2006) las instrucciones de SQL se manejan de la siguiente forma:

La estructura básica de una expresión de consulta en SQL consiste en tres cláusulas: select, from y where.

- La cláusula select corresponde a la operación, se usa para listar los atributos deseados del resultado de una consulta.
- La cláusula from corresponde a las relaciones que deben ser analizadas en la evaluación de la expresión.
- La cláusula where corresponde al predicado que engloba los atributos de las relaciones que aparecen en la cláusula from.

Un ejemplo del uso de las cláusulas básicas de SQL puede ser obtener todos los números de préstamo para préstamos hechos en la sucursal con nombre Navy y puede escribirse como:

```
select número-préstamo  
from préstamo  
where nombre-sucursal = 'Navy'
```

SQL proporciona un mecanismo para renombrar tanto relaciones como atributos. Para ello utiliza la cláusula as, que tiene la forma siguiente:

```
nombre-antiguo as nombre-nuevo
```

SQL ofrece al usuario cierto control sobre el orden en el cual se presentan los registros de una relación. La cláusula order by que hace que los registros resultantes de una consulta se presenten en un cierto orden. Para especificar el tipo de ordenación se puede incluir la cláusula des para orden descendente o asc para orden ascendente.

```
select número-préstamo as np  
from préstamo  
where nombre-sucursal = 'Navy'  
order by np asc
```

En la consulta anterior, se hace uso de la cláusula as para renombrar el atributo y order by para ordenarlo basándose en ese atributo.

Con mucha frecuencia se usa la función de agregación count para contar el número de registros de una relación. La notación para esta función en SQL es count (\*). Así para encontrar el número de registros de la tabla cliente, se escribirá:

```
select count (*)  
from cliente
```

Un borrado se expresa de igual modo que una consulta y se pueden borrar sólo registros completos, es decir, no se pueden borrar valores o atributos concretos. Un borrado se expresa del modo siguiente:

```
delete from cuenta  
where nombre-sucursal = 'Navy'
```

Donde se borran todas las cuentas de la sucursal Navy.

Para insertar datos en una tabla, la instrucción insert más sencilla corresponde a la inserción de un registro. Suponer que se desea insertar en la base de datos el hecho que hay una cuenta C-9932 en la sucursal Navy y que dicha cuenta tiene un saldo de \$ 1,100. La inserción se puede formular del modo siguiente:

```
insert into cuenta  
values ('C-9932','Navy',1100)
```

SQL permite que los atributos se especifiquen en la cláusula insert de la manera siguiente.

```
insert into cuenta (nombre-sucursal, número-cuenta, saldo)  
values ('Navy', 'C-9932',1100)
```

En determinadas situaciones puede ser deseable cambiar un valor dentro de un registro, sin cambiar todos los valores de este. Para este tipo de situaciones se utiliza la instrucción update. Al igual que con insert y delete, se puede elegir los registros que se van a actualizar mediante una consulta. Por ejemplo, si hubiera que realizar el pago de intereses anuales y todos los saldos se incrementan 5 %, habría que formular la siguiente actualización:

```
update cuenta  
set saldo = saldo * 1.05
```

## 4.6 MySQL

MySQL es un sistema de administración de bases de datos relacionales rápido, robusto y fácil de usar. Se adapta bien a la administración de datos en un entorno de red, especialmente en arquitecturas cliente/servidor. Se proporciona con muchas herramientas y es compatible con muchos lenguajes de programación. MySQL se difunde como open source. Así cualquiera tiene acceso al código fuente y puede modificar el código para adaptarlo a sus propias necesidades (Thibaud, 2006).

Para el desarrollo de la aplicación Web fue necesario instalar y configurar el conector MySQL Connector/NET desde la aplicación de Visual Studio.

MySQL Connector/NET permite desarrollar aplicaciones .NET que requieren conectividad de datos segura y de alto rendimiento con MySQL. Implementa las interfaces ADO.NET necesarias y las integra en herramientas compatibles con ADO.NET.

Conexión MySQL utilizada

```
Imports MySql.Data.MySqlClient

Dim conexion As New MySqlConnectionStringBuilder()
    conexion.Server = "remotemysql.com"
    conexion.UserID = "ib91tHJMil"
    conexion.Password = "B1kAohe3uF"
    conexion.Database = "ib91tHJMil"
    conexion.Port = "3306"
Dim con As New MySqlConnection(conexion.ToString)
```

## 4.7 phpMyAdmin

phpMyAdmin es una herramienta de software libre escrita en PHP, su función principal es gestionar la administración de MySQL a través de Internet. phpMyAdmin permite realizar operaciones en MySQL tales como la gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc. Los comandos SQL pueden ser ejecutados a través de la interfaz gráfica de usuario y en el editor se sentencias SQL (Pavón Puertas y Llanera Borges, 2015).

Características:

- Interfaz Web intuitiva.
- Soporte para la mayoría de las características de MySQL.
- Importación datos desde CSV y SQL.
- Exportación de datos a varios formatos: CSV, SQL, XML, PDF, ISO / IEC 26300 - Texto y hoja de cálculo de OpenDocument, Word, entre otros.
- Administrar múltiples servidores.

- Creación de gráficos del diseño de la base de datos en varios formatos.
- Creación de consultas complejas.
- Búsqueda global en una base de datos o un subconjunto de ella.
- Transformación de datos almacenados en cualquier formato usando un conjunto de funciones predefinidas.
- Entre otras características.

## **4.8 Lenguajes de programación**

Según Arias (2016), los lenguajes de programación tienen el propósito de permitir al programador dar instrucciones a la computadora sobre lo que tiene que hacer.

Programar es escribir las instrucciones necesarias, en un lenguaje entendible por la computadora, para resolver un problema particular.

## **4.9 Microsoft.NET Framework**

Es un entorno de desarrollo multilenguaje diseñado por Microsoft para simplificar la construcción, distribución y ejecución de aplicaciones para Internet. Tiene fundamentalmente tres componentes: Una máquina virtual que procesa código escrito en un lenguaje intermedio, una biblioteca de clases y ASP.NET que proporciona los servicios necesarios para crear aplicaciones Web (Ceballos Sierra, 2010).

## **4.10 Visual Studio**

Visual Basic es uno de los lenguajes de programación de alto nivel que pertenecen a .NET. Con Visual Basic.NET se pueden escribir programas convencionales para Internet. Las aplicaciones podrán mostrar una interfaz gráfica al usuario, o bien una interfaz de texto, como hacen las aplicaciones de consola.

Una de las ventajas significativas de Visual Basic sobre otros lenguajes de programación es que el código producido por el compilador Visual Basic puede transportarse a cualquier plataforma que tenga instalada una máquina virtual de .NET y ejecutarse, es decir, es independiente de la plataforma (Ceballos Sierra, 2010).

Visual hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). Sin escribir demasiadas líneas de código para implementar la apariencia y la ubicación de los elementos de la interfaz, ya que se arrastran y colocan objetos prefabricados en la pantalla. (Del Sole, 2017).

Según Fossati (2017), la palabra Basic hace referencia al lenguaje Basic (Beginners All-Purpose Symbolic Introduction Code), un lenguaje utilizado más que ningún otro lenguaje en la historia de la informática o computación. Visual Basic ha evolucionado a partir del lenguaje BASIC original y contiene instrucciones, funciones y palabras clave, relacionadas con la interfaz gráfica de Windows.

El entorno de desarrollo integrado de Visual Studio es un panel de inicio creativo que se puede usar para editar, depurar, compilar código y publicar una aplicación. Un entorno de desarrollo Integrado (IDE) es un programa con numerosas características que se pueden usar para muchos aspectos del desarrollo de software. Más allá del editor estándar y depurado que proporcionan la mayoría de IDE, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos entre otras características para el desarrollo de software (Docs.microsoft.com, 2019).

#### **4.10.1 Características**

- Subrayados ondulados y acciones rápidas: Los subrayados ondulados son rayas con forma de onda debajo de las palabras que alertan de errores o posibles problemas en el código a medida que se escribe. Estas pistas visuales permiten corregir problemas, además se puede ver información adicional sobre el error y acciones rápidas para corregirlo.
- IntelliSense: Es un término que define un conjunto de características que muestran información sobre el código directamente en el editor y en algunos casos, escriben pequeños fragmentos de código automáticamente.
- Cuadro de búsqueda: El cuadro de búsqueda supone una manera de encontrar rápidamente lo que se necesita en Visual Studio.
- Ir a definición: Es una característica que lleva directamente a la ubicación donde se define una función o un tipo.
- Entre un sinnúmero de características más.

#### **4.11 Java**

Flórez Fernández (2012) señala que Java fue creado en 1991 por Sun Microsystems con el fin de elaborar un lenguaje de programación destinado a electrodomésticos. A finales de 1995, Java fue introducido como lenguaje de programación para computadoras. Existen IDE como Eclipse, Netbeans, JNulder, etc. Para escribir código Java, compilarlo y ejecutarlo.

Java surgió en los años 90 como un lenguaje orientado a objetos sencillo, fácil de usar, potente y muy bien adaptado para la programación de aplicaciones en red.

### 4.11.1 Clases

Los programas orientados a objetos se desarrollan mediante el uso de clases predefinidas siempre que sea posible. Únicamente se deben definir clases propias cuando no haya disponibles clases predefinidas adecuadas. En Java, las clases se agrupan en paquetes. Un paquete puede incluir subpaquetes, definiendo una jerarquía de paquetes.

La declaración de una clase se lleva a cabo usando la palabra reservada `class` mediante la sentencia:

```
Public class PrimerPrograma;
```

La palabra reservada `public` indica que esa clase se exporta al exterior, de lo contrario puede ser `private`. `PrimerPrograma` es el nombre de la clase principal y, por extensión de la aplicación (Durán, Gutiérrez y Pimentel, 2007).

### 4.11.2 Métodos

La sintaxis para la declaración de un método es:

```
<modificadores> <tipo devuelto> <nombre de método> (<parámetros>)  
{  
    <cuerpo del método>  
}
```

donde `<modificadores>` es una secuencia de términos para designar distintos tipos de métodos; `<tipo devuelto>` es el tipo de valor de datos devuelto por un método; `<nombre de método>` como se indica, es el nombre del método; `<parámetros>` es una secuencia de valores que pasan al método, y `<cuerpo del método>` es una sentencia de instrucciones (Durán, Gutiérrez y Pimentel, 2007).

### 4.11.3 Identificadores

Según Durán, Gutiérrez y Pimentel (2007), un identificador es el nombre que se asigna a los distintos elementos de un programa, como pueden ser variables, funciones, etc. Se forman utilizando una secuencia de caracteres que pueden incluir dígitos (del 0 a 9), las letras de la “a” a la “z” (mayúsculas o minúsculas), los signos de subrayado (`_`) y de moneda (`$`). La única restricción es que no puede empezar por un dígito.

Las palabras reservadas con identificadores de un lenguaje de programación que tienen un significado especial para el compilador.

Algunos ejemplos de palabras reservadas son:

- Boolean
- Break
- Char
- Class
- Float
- Short
- Try

#### 4.11.4 Variables

Una variable es una representación alfanumérica de una posición de memoria. Como tal, se caracteriza por tres propiedades: Posición de memoria que almacena el valor, tipo de datos almacenado y nombre que se refiere a esa posición de memoria. El tamaño de esa zona, en bytes dependerá del tipo de datos que se almacene en la variable y pueden contener diferentes valores durante la ejecución de un programa y se declaran de la siguiente forma (Durán, Gutiérrez y Pimentel, 2007):

<tipo de datos> <nombre>;

Ejemplos

```
int temperatura;  
float precio;
```

#### 4.11.5 Constantes

Son valores que se pueden asignar desde la declaración o posteriormente, pero solo una vez, este valor no cambia en la ejecución del programa.

La sintaxis de declaración de una constante se usa de la forma siguiente:

Final <tipo de dato> <nombre de constante> = <valor>;

La palabra reservada final indica que la constante tiene un valor que se fija durante todo el periodo de vida que dura la ejecución de un programa (Durán, Gutiérrez y Pimentel, 2007).

Ejemplos

```
final int TEMPERATURA = 37;  
final String PELIGRO = "Peligro. No pasar";
```

## 4.12 Android

Android es un sistema operativo, inicialmente diseñado para teléfonos móviles, en la actualidad se instala también en múltiples dispositivos como tablets, GPS, televisores, etc. Está basado en Linux y permite programar aplicaciones empleando una variación de Java, archivos xml para las vistas y es completamente libre (open source).

Android es un entorno de desarrollo de aplicaciones para Android, este IDE fue desarrollado por Google, por lo que es nombrado el IDE oficial. Cuenta con un editor de código con elementos integrados como “Smart Editing” que proporciona código más legible, Gradle que permite aplicar distintas configuraciones versiones del mismo código para producir distintas versiones del mismo código de aplicación. (Hohenesee, 2014).

### 4.12.1 Arquitectura Android

- Aplicaciones: Todas las aplicaciones están escritas en lenguaje de programación Java.
- Armazón de aplicaciones: Los programadores tienen acceso completo a las mismas APIs (librerías) del Framework utilizadas por aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes.
- Librerías: Android incluye un conjunto de librerías de C/C++ usadas por varios componentes del sistema, entre ellas, librerías de medios, bibliotecas de gráficos 3D y SQLite, entre otras.
- Runtime: Cada aplicación Android ejecuta un proceso con instancia individual de la máquina virtual.
- Núcleo Linux: Android está basado en Linux para los servicios base del sistema, como seguridad, gestión de memoria, procesos y controladores.

En la Figura 1 se muestra la arquitectura de Android antes mencionada.

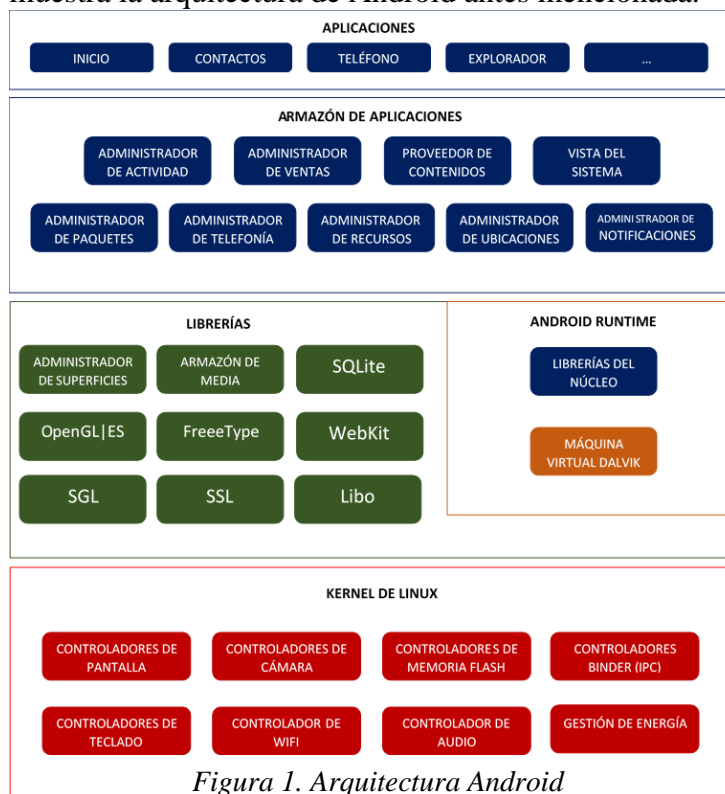


Figura 1. Arquitectura Android



Robledo (2015) comenta que el sistema operativo Android tiene la característica que sus versiones llevan el nombre en inglés de postres y ordenados alfabéticamente de acuerdo con su versión y API son:

- Apple pie versión 1 API 1
- Banana Split versión 1.1 API 2
- Cupcake versión 1.5 API 3
- Donut versión 1.6 API 4
- Eclair versión 2 API 5
- Froyo versión 2.2 API 8
- Gingerbread versión 2.3 API 9
- Honeycomb versión 3 API 11
- Ice Cream Sandwish versión 4.0.1 API 14
- Jellybean versión 4.1 API 16
- KitKat versión 4.4 API 19
- Lollipop versión 5 API 21
- Marshmallow versión 6 API 23
- Nougat versión 7 API 24 y versión 7.1 API 25
- Oreo versión 8 API 26
- Pie versión 9 API 28

Se tiene que tener en cuenta que este IDE necesita de Java por lo que se debe instalar el Java SDK así como el Android SDK necesario

Android Studio ofrece funciones para aumentar la productividad durante la compilación de aplicaciones como lo son:

- Sistema de compilación basado en Gradle flexible
- Emulador rápido
- Instant Run para aplicar cambios mientras la app se ejecuta sin compilar un nuevo apk
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.

## **4.13 HTML**

Según Cobo Yera (2007), HTML es un lenguaje artificial que los ordenadores son capaces de interpretar y diseñado para que los programadores puedan redactar instrucciones que los navegadores ejecuten para originar la página Web. Es decir, HTML es un lenguaje de programación, o un “idioma que la máquina entiende y procesa para dar una respuesta”.

Las siglas HTML significan HyperText Markup Language (lenguaje de marcas de hipertexto). El hipertexto en una computadora es texto que posee referencias (hipervínculos, links o enlaces) a otro texto. Además de texto, el hipertexto puede estar formado por tablas, imágenes u otros elementos.

En esencia, HTML sirve para estructurar documentos (títulos, párrafos, listas, etc.), pero no describe la apariencia o el diseño de un documento, sino que ofrece las herramientas necesarias para dar formato, según la capacidad del servidor Web en el que se almacenan las páginas Web y la capacidad del navegador (tamaño de pantalla, fuentes instaladas, etc.).

El lenguaje HTML tiene dos ventajas que lo hacen prácticamente imprescindible a la hora de diseñar una presentación Web como es su compatibilidad y facilidad que platea su aprendizaje debido al reducido número de etiquetas en las que se apoya.

Para escribir código HTML es necesario un editor de textos básico como bloc de notas de Windows, Gedit de Linux o WordPad y para poder ver el resultado, es necesario tener instalado un navegador Web como Internet Explorer, Safari o Mozilla. El código se guarda en formato .html o .htm, lo cual da como resultado un archivo que ya puede ser abierto con el navegador.

#### 4.13.1 Sintaxis

La sintaxis de un lenguaje de programación son las reglas que indican cómo realizar las construcciones de este. Básicamente la sintaxis de HTML consiste en etiquetas, atributos y los elementos que conforman la página, tanto el texto como las imágenes, tablas, gráficos, etc. Plasmado en comandos sería como lo siguiente:

<etiqueta> elemento afectado </etiqueta>

#### 4.13.2 Formatos

- <P> Sirve para delimitar un párrafo. Inserta una línea en blanco antes del texto.
- <CENTER>...</CENTER> Permite centrar el texto.
- <FONT> </FONT> Permite cambiar las características del texto.
- <FONT color = “ ”> Cambia el color al texto.
- <FONT size = “ “> Cambia el tamaño del texto.
- <FONT face = “ “> Cambia la fuente del texto
- <B> </B> Texto en negrita.
- <I> </I> Texto en cursiva.
- <U> </U> Subrayar algo.
- <HR> Inserta barra horizontal.
- <BR> Salto de línea.
- <H1> </H1> Cabecera de nivel 1, existen 6 niveles.

Ejemplo

```
<HTML>
  <HEAD>
    <TITLE> Ejemplo <TITLE>
```

```

<HEAD>

<BODY>
  <CENTER><H1>Ejemplo</H1></CENTER>
  <HR>
  <P> ejemplo básico de una página HTML
</BODY>
</HTML>

```

## 4.14 PHP

Minera (2008) describe a Hypertext Preprocessor como uno de los lenguajes de programación más utilizados en la actualidad esto debido a que posee características como las siguientes:

- Es libre y gratuito.
- Disponibilidad. El soporte al momento de alojarlo es casi seguro y disponible para sistemas operativos como Mac Os, Microsoft Windows, Unix, Solaris, etc.
- Soporta múltiples bases de datos: Cuenta con extensiones para soportar bases de datos MySQL, SQL Server, Dbase entre otras.
- Evolución. Hay mejoras y avances constantes.
- Facilidad de aprendizaje. PHP se caracteriza por su simpleza.

Las páginas creadas con PHP son simples páginas en HTML que contienen, además de las etiquetas normales, el programa que queremos ejecutar en PHP. Por ejemplo:

```

<HTML>
  <HEAD>
    <TITLE> Ejemplo PHP <TITLE>
  </HEAD>

  <BODY>
    <CENTER><H1>Ejemplo PHP</H1></CENTER>
    <HR>
    <?php
      Código PHP...
    ?>
  </BODY>
</HTML>

```

Su sintaxis básica es muy sencilla, por ejemplo, el programa PHP inicia con la etiqueta <?php y termina con la etiqueta ?>, así se indica que lo que está dentro de estas etiquetas pertenecen a este lenguaje (Eslava Muñoz, 2013).

#### 4.14.1 Integración de PHP al proyecto

Para poder acceder a los datos desde la página Web al servidor, se utilizó un archivo llamado `conexion.php` que contiene el código necesario para realizar el vínculo, este archivo se incluye en todos documentos php donde se necesite realizar la conexión con la base de datos mediante la instrucción `include` (`include 'conexion.php'`).

El código que contiene el archivo es el siguiente:

```
<?php
$h='remotemysql.com';
$u='ib9ltHJMil';
$p='B1kAohe3uF';
$db='ib9ltHJMil';

//Crear conexión y seleccionar Base de Datos
$db = mysqli_connect($h, $u, $p, $db);

if (!$db->set_charset("utf8")) {
    printf("Error al cargar charset utf8: %s\n", $db->error);
}

if ($db->connect_error) {
    die("No se puede conectar con la base de datos: " . $db->connect_error);
}
?>
```

#### 4.15 JavaScript

Sánchez Maza (2001) presenta a JavaScript como un lenguaje de desarrollo de aplicaciones cliente/servidor a través de Internet.

Tiene la particularidad de estar insertado dentro del mismo documento HTML que lo presenta al usuario. Permite crear aplicaciones similares a los CGI. El CGI es un mecanismo que se ha utilizado en los servidores Web para implementar las páginas Web activas.

El programa JavaScript reconoce eventos, son acciones de JavaScript, creados por el usuario definiendo así un sistema interactivo.

JavaScript es un lenguaje compacto, y basado en objetos, diseñado para el desarrollo de aplicaciones cliente-servidor a través de Internet.

JavaScript fue diseñado para ser un lenguaje de elaboración de script que pudieran incrustarse en archivos HTML. No es compilado, sino que, en vez de ello, es interpretado por el navegador.

JavaScript está construido en base a tres bloques particulares:

- Variables: Los tipos de datos soportados por este lenguaje incluyen números, lógicos, cadenas de caracteres y el tipo nulo.
- Objetos: Es un contenedor dado para una colección de propiedades (valores). También puede estar construido de otros objetos. Para crear un objeto primer se debe crear la función constructora de ese objeto.
- Funciones: Procedimientos que una aplicación puede ejecutar. Las funciones asociadas con un objeto en particular son llamadas métodos de un objeto. Para declarar una función se tiene que hacer de la siguiente manera:

```
function cuadrado()  
{  
    Instrucciones de código  
}
```

## 4.16 CSS

Las hojas de estilo en cascada ofrecen propiedad para ampliar el lenguaje HTML en la representación visual de las páginas Web. El lenguaje CSS, definido por primera vez en el año 1996, es el más conocido y utilizado para declarar las propiedades de formato de los diferentes elementos HTML. Este lenguaje permite vincular los documentos HTML con “plantillas de documento”, que además de contener la información topográfica de los elementos visuales de la página, permiten separar la estructura de contenidos de su representación y presentación actuales, no solo en monitor si no en cualquier pantalla (móvil, PDA, etc.), tecnología de soporte (lectores de pantalla, líneas en braille) o en papel impreso (Schulz, 2009).

### 4.16.1 Integración de CSS en el proyecto

En la página Web se utilizó un archivo con nombre estilos.css donde se dio formato a los elementos que esta contenía, un ejemplo del uso de CSS en el proyecto es:

```
.btn__reset {  
    background-color: #FA9535;  
}  
  
h2 span {  
    color: #A21515;  
}
```

## 4.17 JSON

JSON es el acrónimo en inglés de JavaScript Object Notation; es un formato ligero para el intercambio de datos en aplicaciones Web. JSON tiene la ventaja de que no requiere el uso de XML.

JSON se basa en dos tipos de datos y sintaxis del lenguaje JavaScript. Es compatible con cadenas, números, booleanos y valores nulos. También se pueden combinar valores en matrices y objetos.

Los objetos JSON son simplemente conjuntos desordenados de parejas de nombre/valor, donde el nombre es siempre una cadena y el valor es cualquier tipo de datos válido para JSON, incluso otro objeto.

Ejemplo:

```
{
    "producto": {
        "nombre": "widget"
        "numero": "7402-129"
    }
}
```

Android incluye la biblioteca JSON, que permite tratar ese formato de datos (Robledo, 2015). Las clases más importantes de ese paquete son:

- JSONArray: permite cargar y tratar una matriz de elementos en formato JSON.
- JSONObject: permite tratar un único elemento en formato JSON.

### 4.17.1 Integración de JSON en el proyecto

A continuación, se muestra un pequeño fragmento de código utilizado en la aplicación móvil que muestra el uso de JSON.

```
JSONObject jsonObj = new JSONObject(jsonStr);

JSONArray contacts = jsonObj.getJSONArray("contacts");

for (int i = 0; i < contacts.length(); i++) {
    JSONObject c = contacts.getJSONObject(i);
    String xnombre = "Cobro N°..." + c.getString("idCobro");
    String xrfc = "Corte N°..." + c.getString("idCorteCobro");
    String xtel = "Empacadora." + c.getString("e");
    String xdom = "Fecha....." + c.getString("fechaCobro");
}
```

## 5. METODOLOGÍA

En el presente capítulo se aborda la metodología utilizada en el desarrollo del proyecto, primero se expone el análisis de la problemática, las necesidades y posteriormente, se expone la forma de dar solución con diagramas que muestran cómo se conforma la base de datos y el funcionamiento de cada uno de los escenarios (de escritorio, Web y móvil) que componen la plataforma. El capítulo concluye mostrando las pruebas que se realizaron a la plataforma con sus resultados y se hace mención sobre la implementación y documentación de esta.

### 5.1 Análisis

Se acudió a la oficina central de la empresa “Huertas El Cincuenta 1” para analizar la documentación almacenada y a través de intercambio de información se determinó que la empresa tenía principalmente los requerimientos siguientes:

- Disponer de una herramienta para consultar la información por Internet.
- Gestionar los cortes de fruta.
- Registrar la entrada y salida de capital de la empresa administrando compras, gastos y ventas permitiendo grabar y revisar los pagos que se realizan hasta saldar una cuenta.
- Llevar control de las actividades que se llevan a cabo en cada huerta, así como los productos utilizados en el mantenimiento.

#### 5.1.1 Políticas

Las políticas hacen referencia a las delimitaciones que tiene la empresa y deben respetarse al realizar un sistema de información tal cual la empresa los señale.

- El cobro a emparadoras y pago a proveedores puede variar entre efectivo, transferencia bancaria, tarjeta de crédito o débito u otros (en casos especiales con algún bien material).
- Los cortes pueden ser efectuados por diferentes emparadoras.
- El encargado de la huerta cambia de manera constante y puede ser un encargado para varias huertas.
- El precio de los productos se actualiza constantemente en los inventarios.
- Las compras a proveedores y ventas a emparadoras pueden ser a crédito o al contado.

#### 5.1.2 Almacenes de datos

A continuación se muestran los almacenes de datos o tablas de MySQL en las que se registra y consulta la información de la plataforma.

- Actividades

- ActividadesDetalle
- Cobros
- Compras
- ComprasDetalle
- ConceptoGastos
- Cortes
- Empacadoras
- Encargados
- Gastos
- Huertas
- Pagos
- Productos
- Proveedores
- TipoCobroPago
- TipoProducto
- TipoActividades
- Usuarios

### **5.1.3 Procesos**

Los procesos que se mencionan a continuación representan la funcionalidad que tiene que llevar a cabo el sistema para transformar o manipular datos.

- Registrar huertas, encargados, proveedores y empacadoras.
- Guardar productos, tipos de cobro o pago y tipo de productos.
- Registrar cortes y actividades.
- Consultar huertas, encargados, proveedores, productos, empacadoras y catálogos con datos secundarios.
- Modificar todos los catálogos que incluyen huertas, encargados, proveedores, productos y empacadoras.
- Grabar compras, pagos, cobros y gastos varios.
- Obtener reportes de cortes por huerta, empacadora o pendientes de cobro.
- Obtener reportes de compras pendientes de pago, por fecha y por proveedor.
- Obtener gastos por fecha y por concepto.

### **5.1.4 Entidades externas**

Las entidades externas hacen referencia a un ente ajeno al sistema que proporciona o recibe información de este.

- Encargados
- Propietario
- Proveedores
- Empacadoras



## 5.2 Diseño

### 5.2.1 Diagrama de Warnier/Orr de la aplicación de escritorio

En las Figuras 2 y 3 se muestra el diagrama general de Warnier/Orr de la aplicación de escritorio.

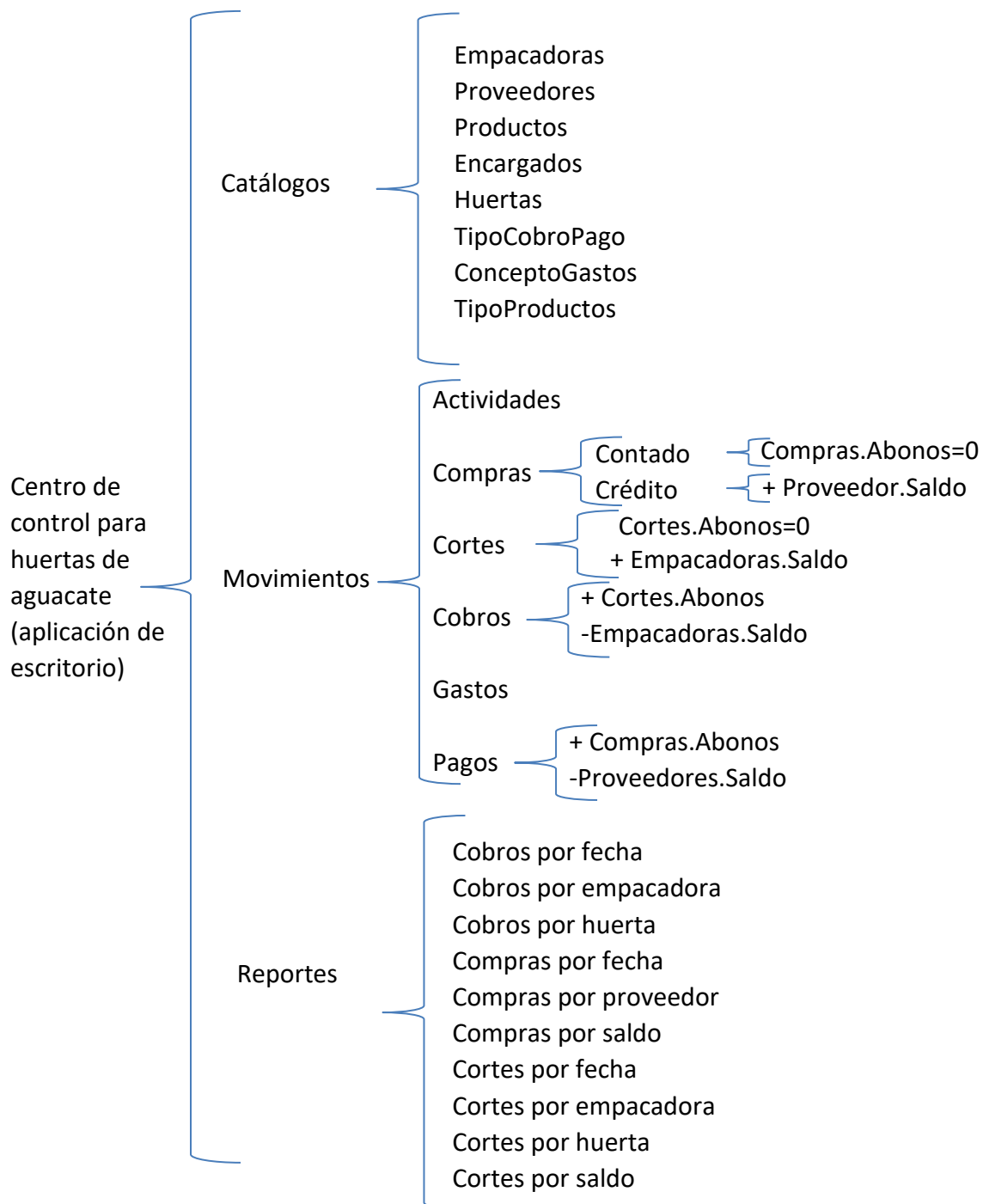


Figura 2. Diagrama de Warnier/Orr de escritorio 1

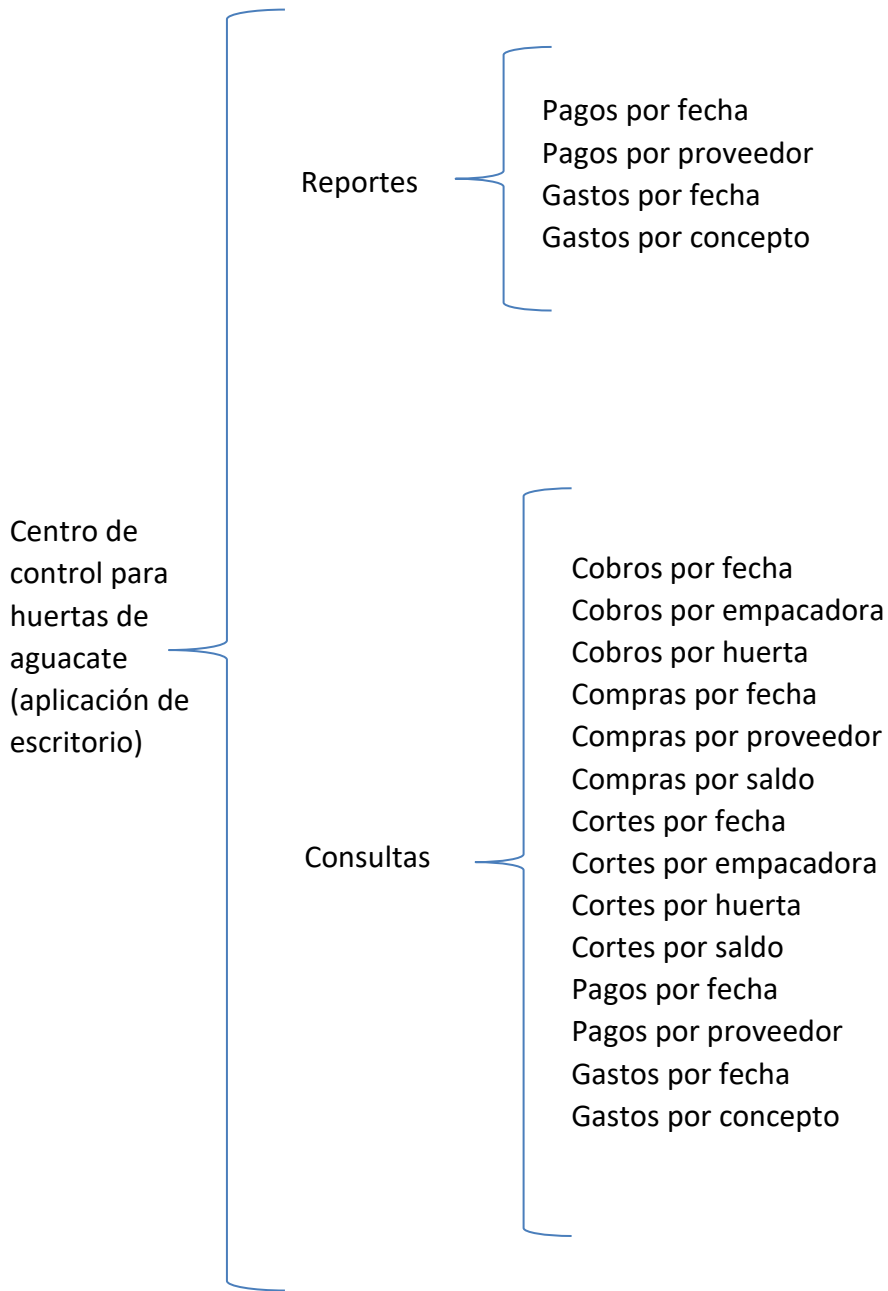


Figura 3. Diagrama de Warnier/Orr de escritorio 2

### 5.2.2 Diagrama de Warnier/Orr de la página Web

En la Figura 4 se muestra el diagrama general de Warnier/Orr de la página Web.

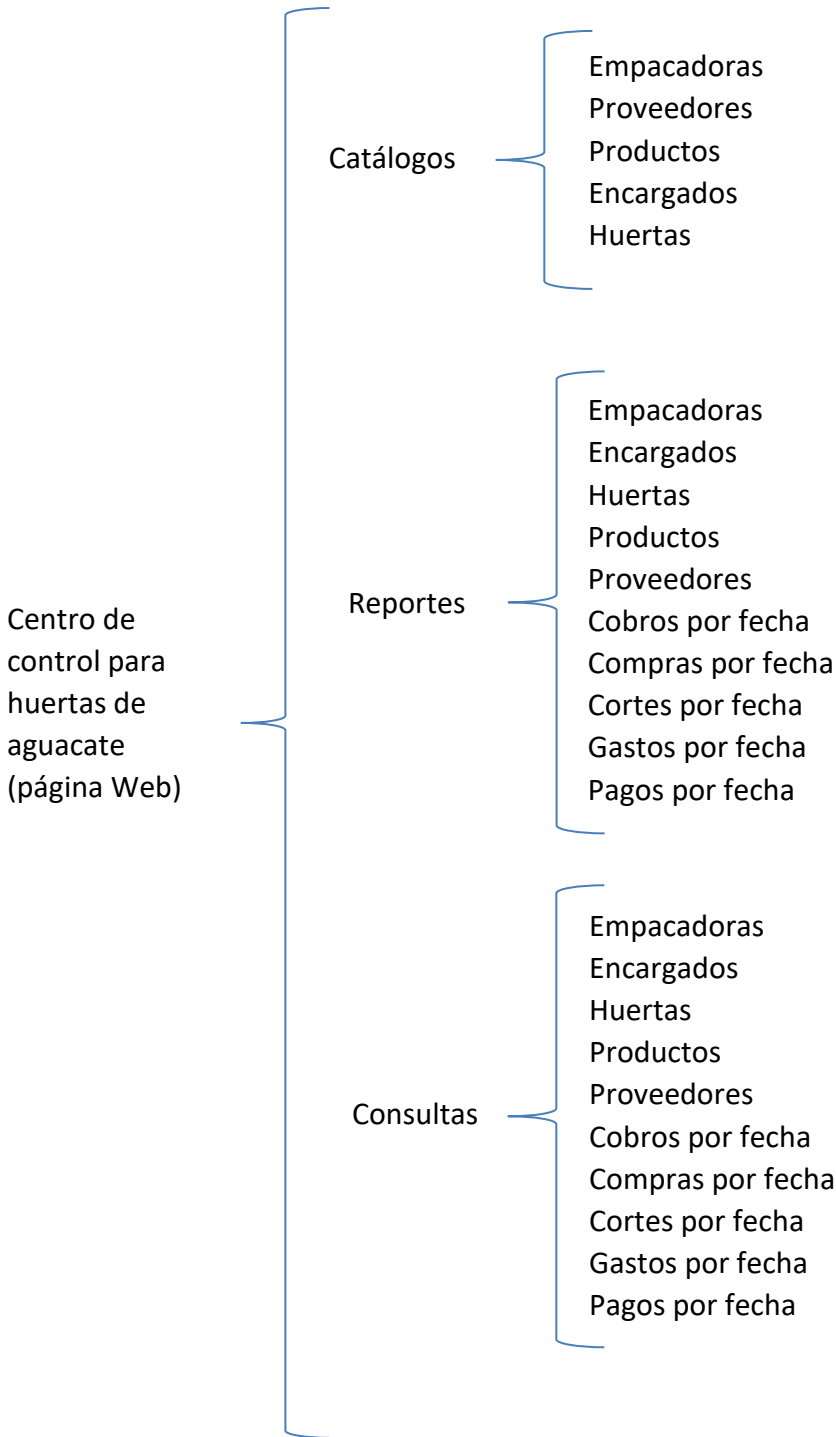
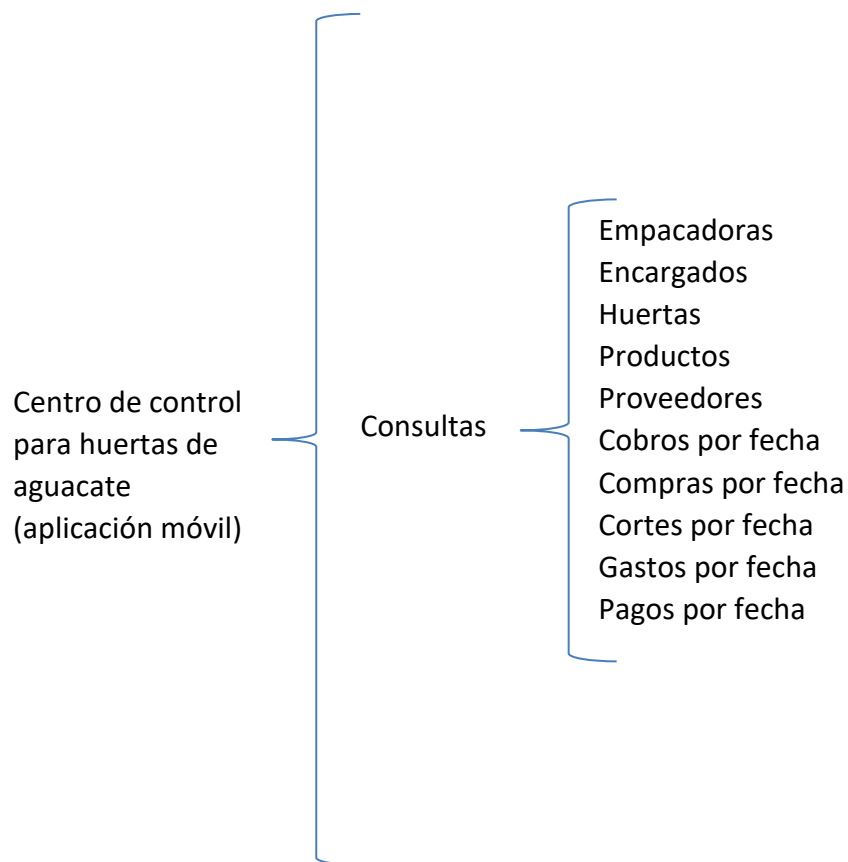


Figura 4. Diagrama de Warnier/Orr Web

### 5.2.3 Diagrama de Warnier/Orr de la aplicación móvil

En la Figura 5 se muestra el diagrama general de Warnier/Orr de la aplicación móvil.



*Figura 5. Diagrama de Warnier/Orr móvil*

## 5.2.4 Diagrama de caso de uso general

En la Figura 6 se muestra el diagrama de caso de uso general de la plataforma.

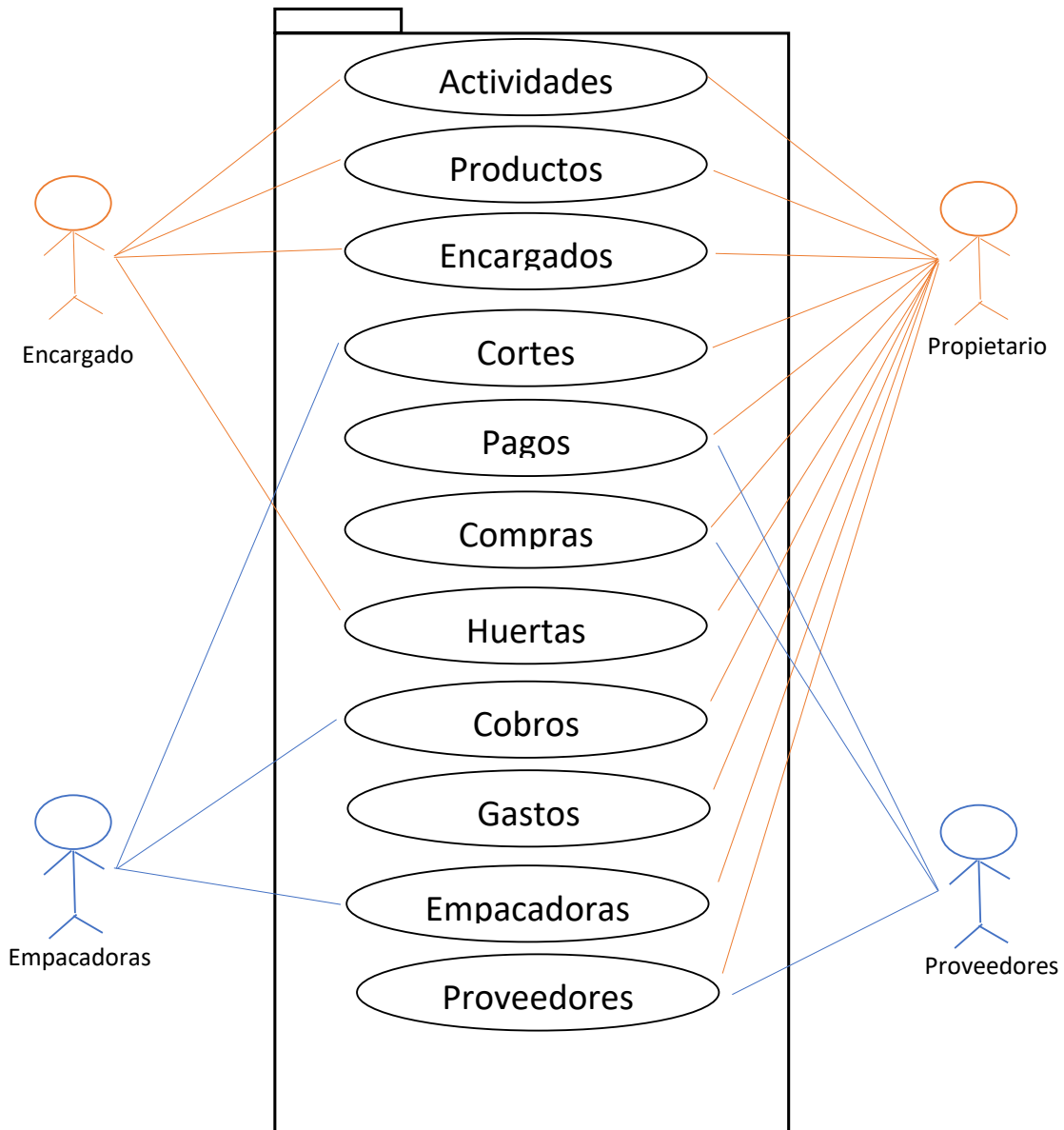


Figura 6. Diagrama de caso de uso general



## 5.2.6 Diagramas de cortes

### Diagrama de flujo de datos para el caso de uso Cortes

En la Figura 8 se muestra el diagrama de flujo de datos para el caso de uso cortes.

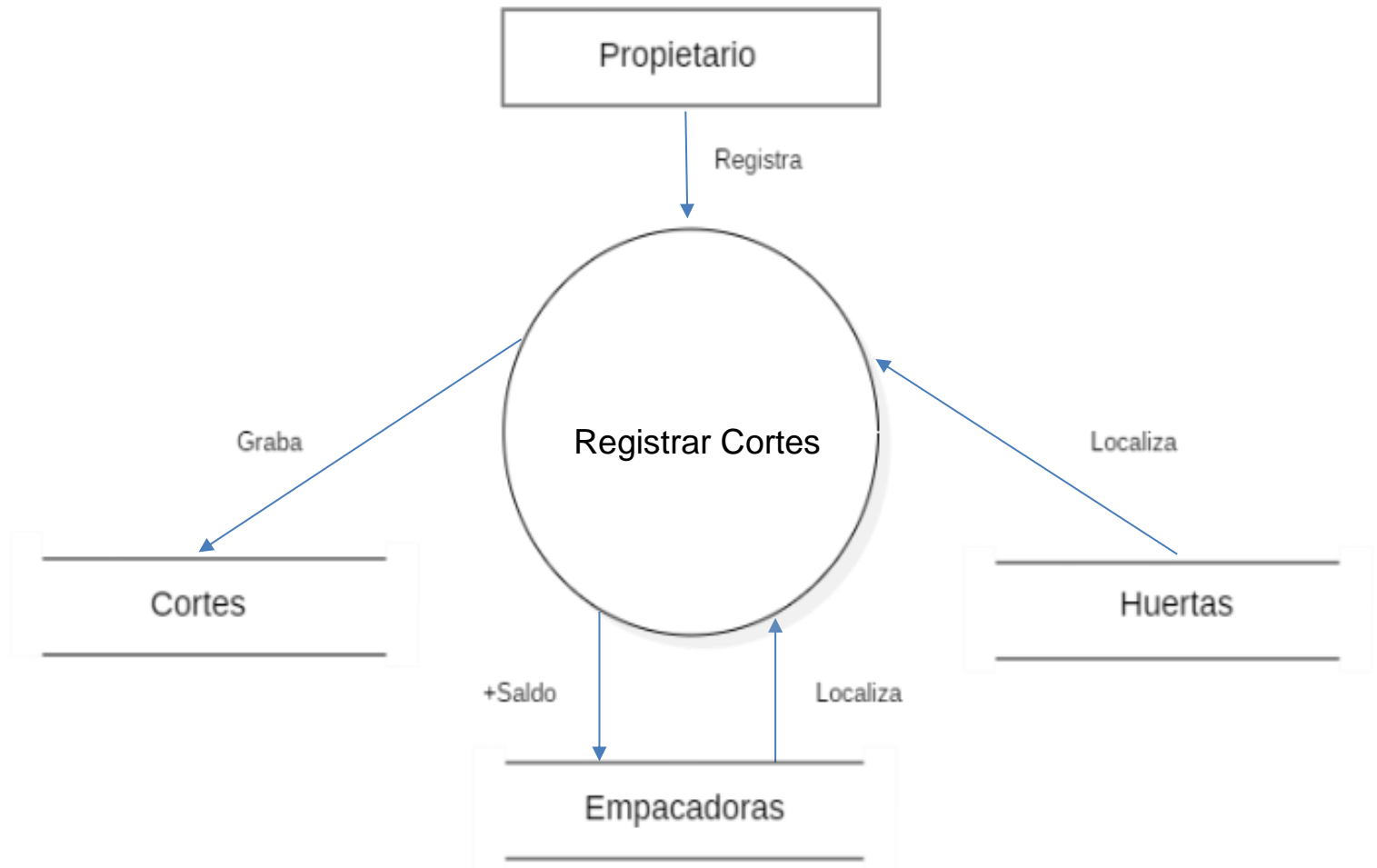


Figura 8. Diagrama de flujo de datos cortes

### Diagrama de caso de uso expandido para el caso de uso cortes.

En la Figura 9 se muestra el diagrama de caso de uso expandido para el caso de uso cortes.

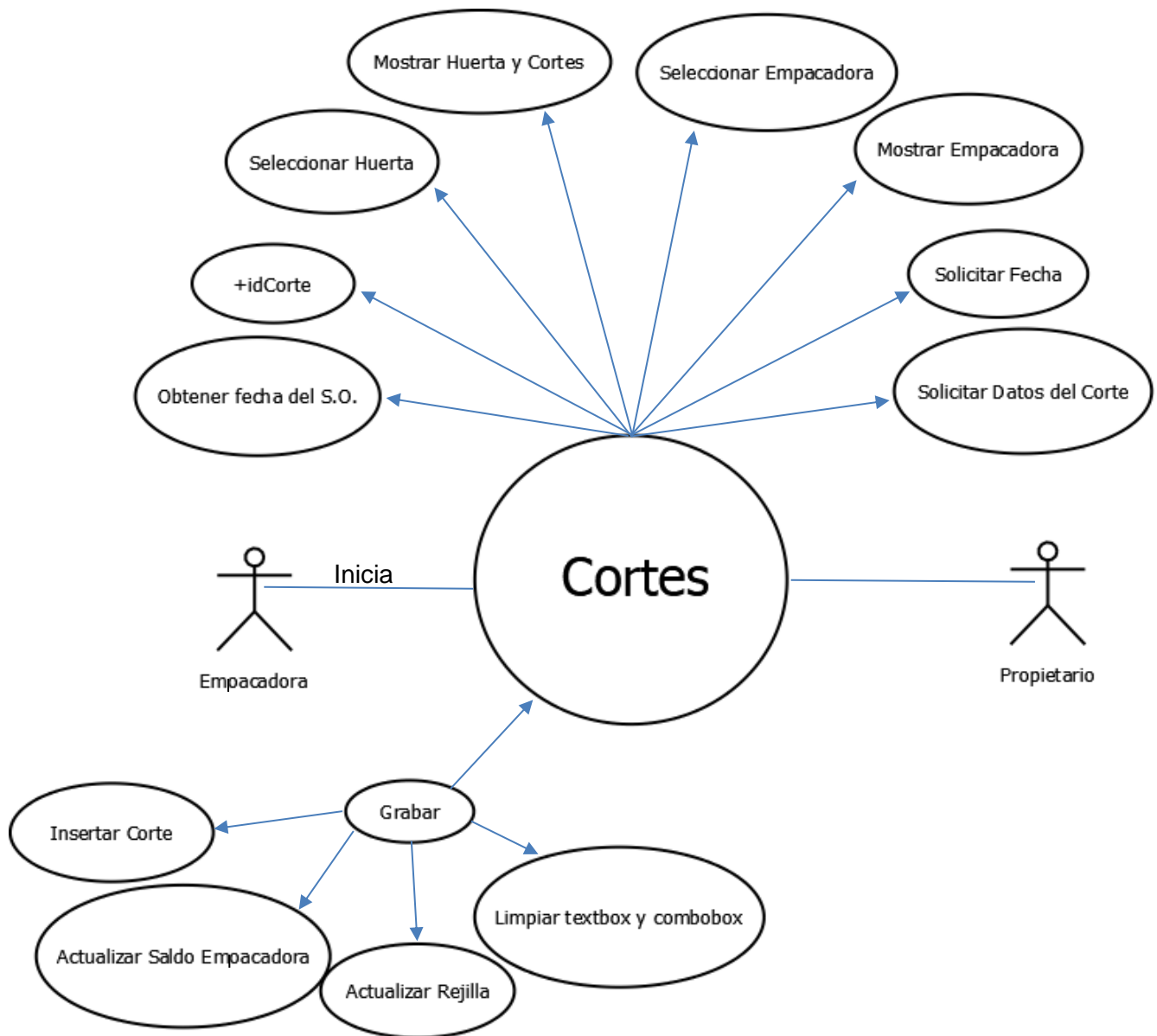


Figura 9. Diagrama de caso de uso expandido cortes



## Diagrama de estado para el caso de uso cortes

En la Figura 10 se muestra el diagrama de estado para el caso de uso cortes.

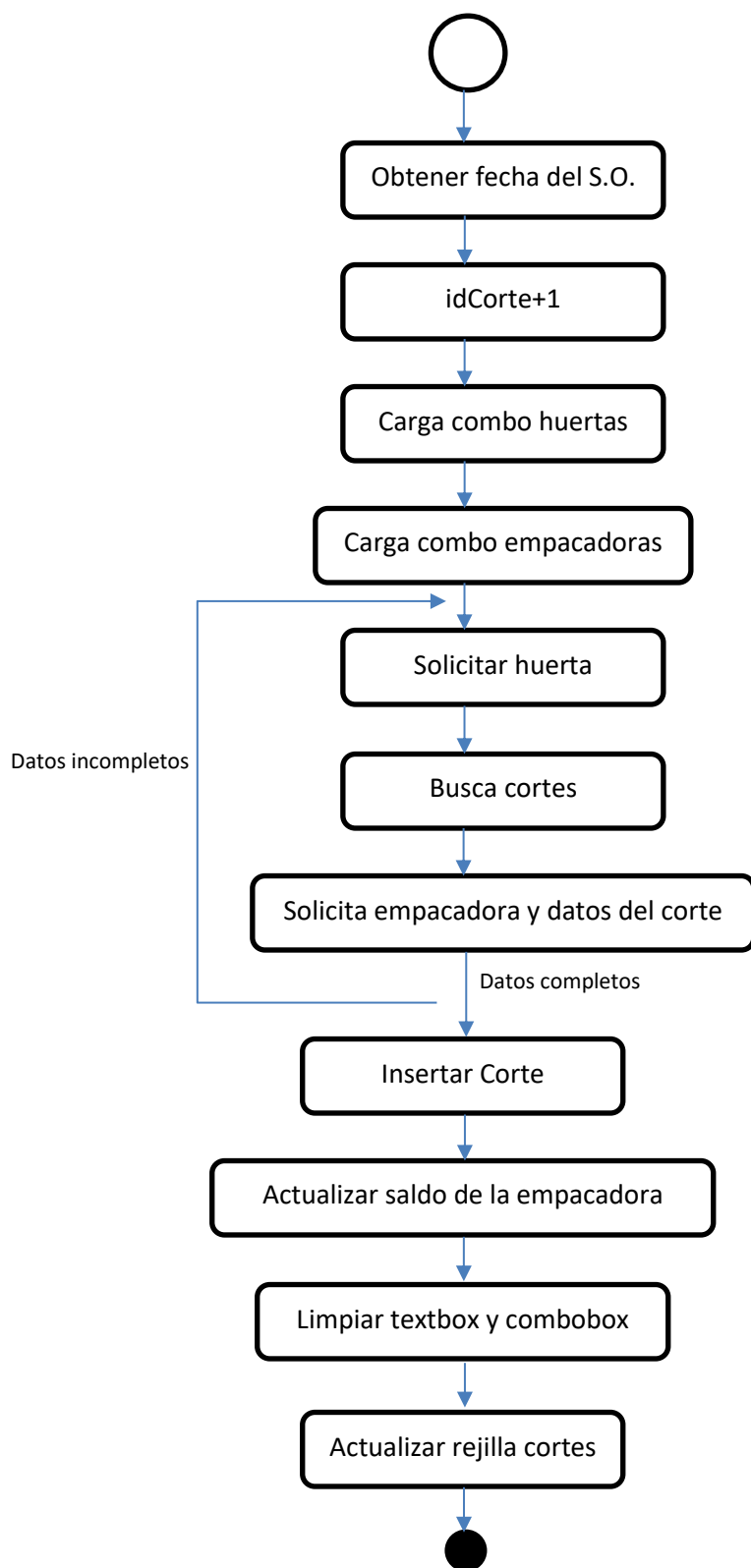


Figura 10. Diagrama de estado corte

## Diagrama de colaboración para el caso de uso cortes

En la Figura 11 se muestra el diagrama de colaboración para el caso de uso cortes.

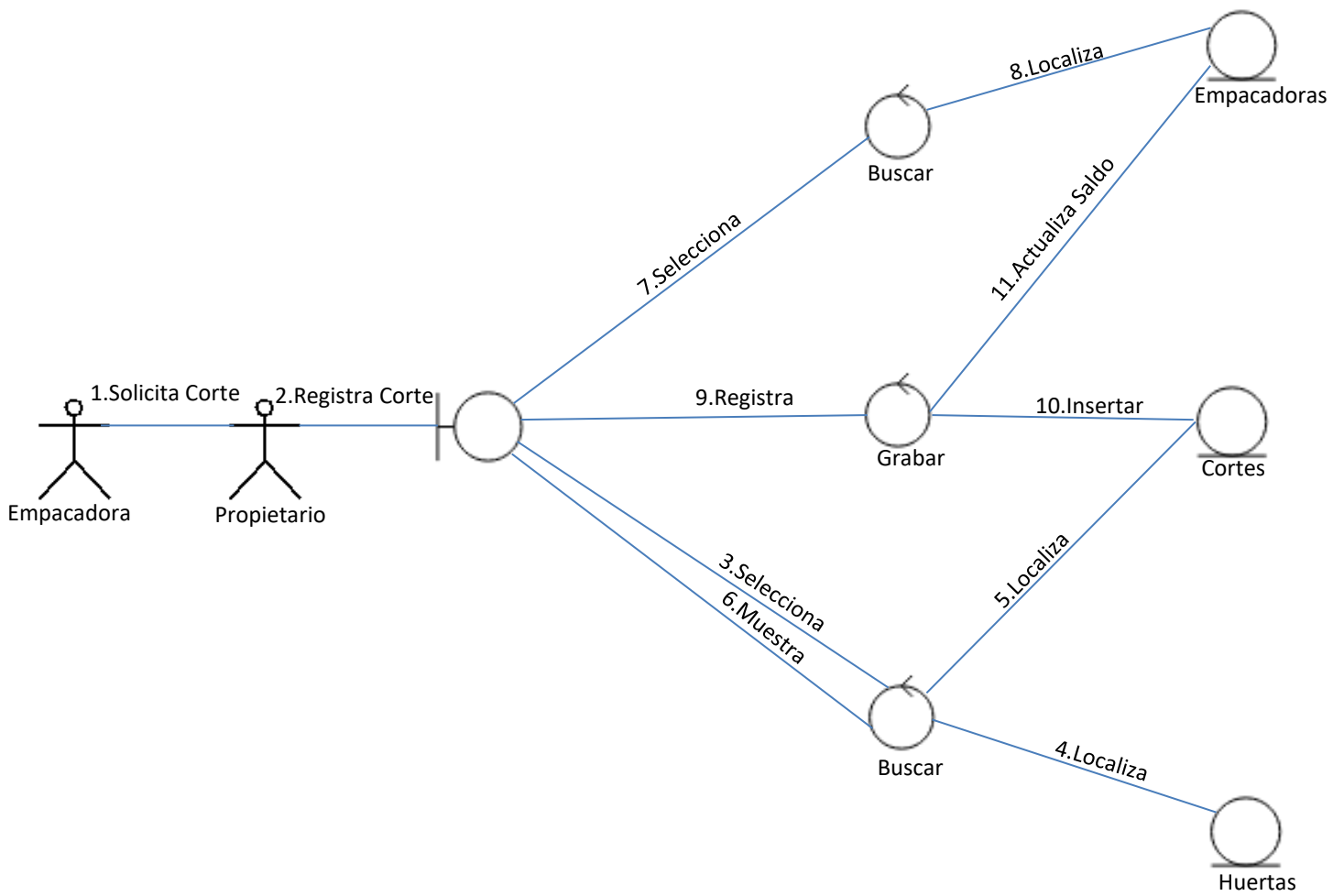


Figura 11. Diagrama de colaboración cortes

### 5.2.7 Diagramas de compras

#### Diagrama de flujo de datos para el caso de uso compras

En la Figura 12 se muestra el diagrama de flujo de datos para el caso de uso compras.

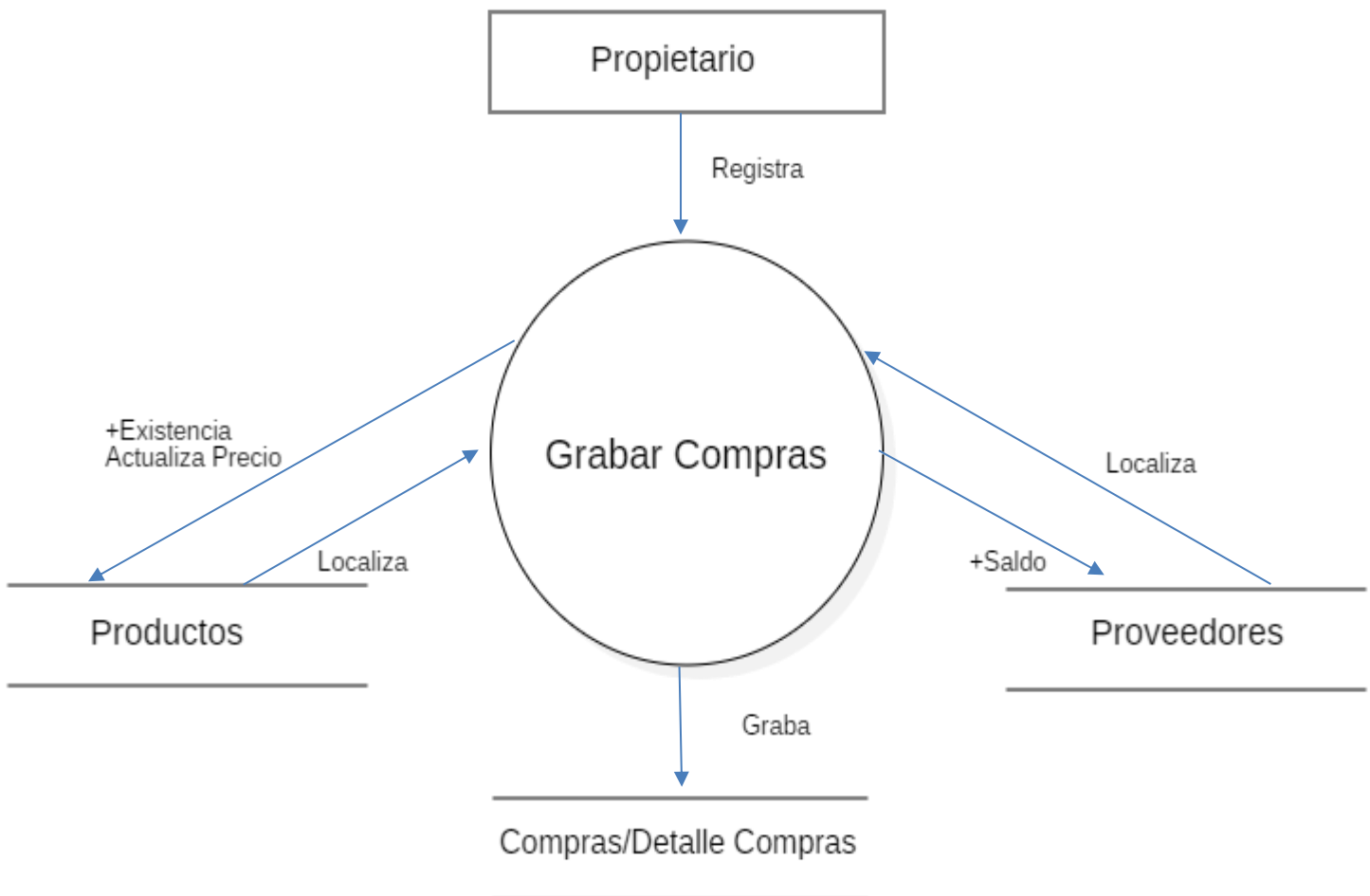


Figura 12. Diagrama de flujo de datos compras

## Diagrama de caso de uso expandido para el caso de uso grabar compras

En la Figura 13 se muestra el diagrama de caso de uso expandido para el caso de uso grabar compras.

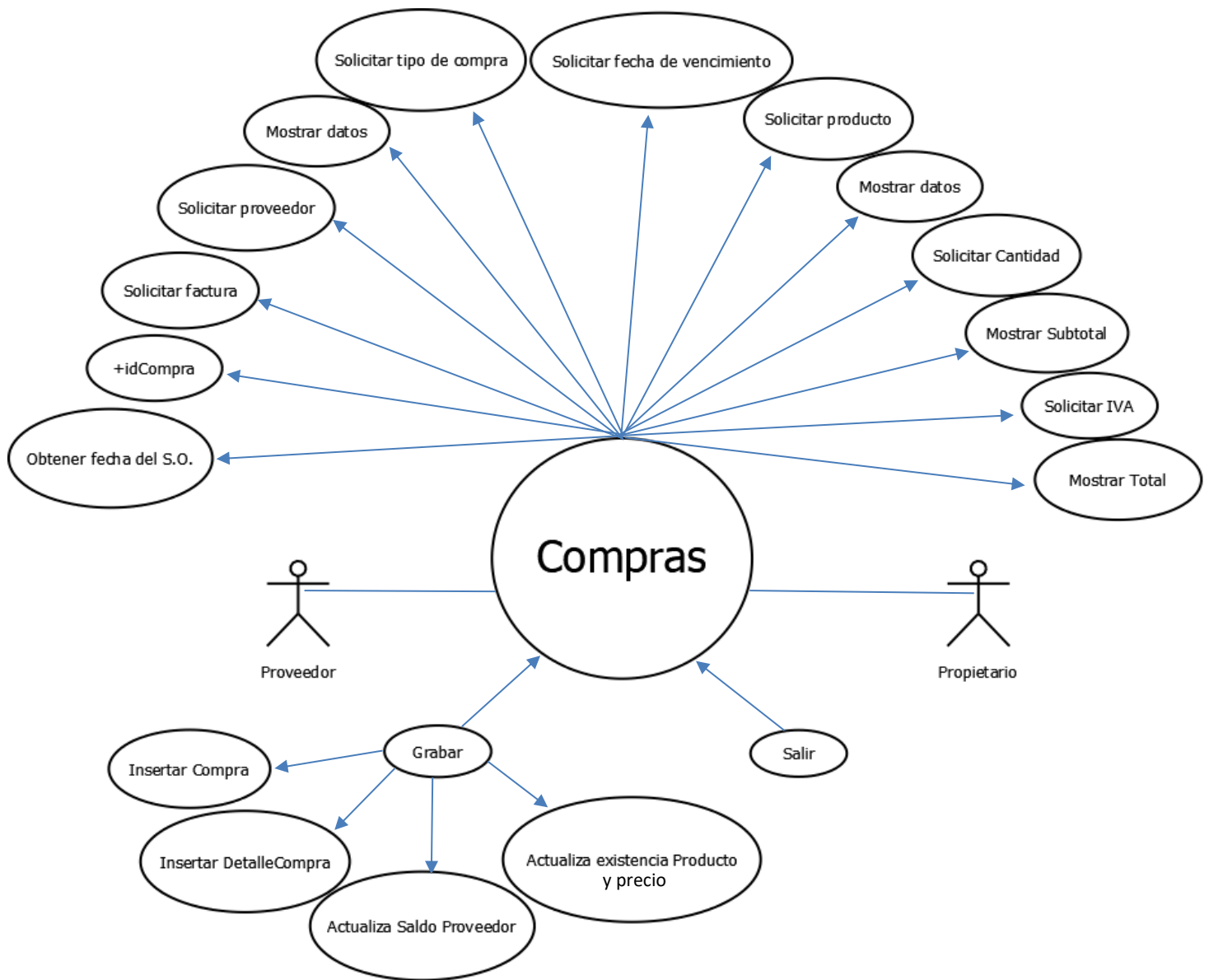


Figura 13. Diagrama de caso de uso expandido compras

## Diagrama de estado para el caso de uso compras

En la Figura 14 se muestra el diagrama de estado para el caso de uso compras.

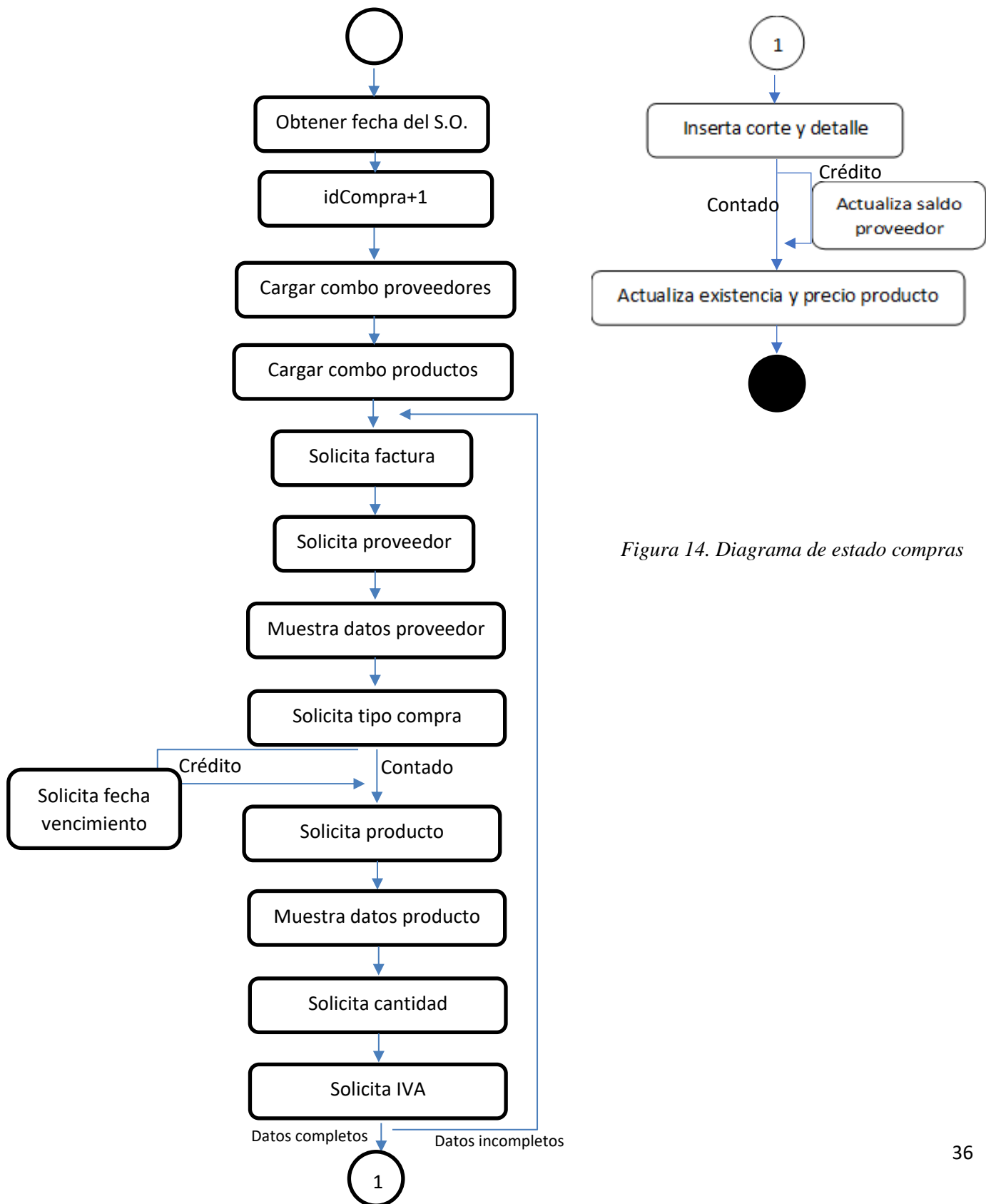


Figura 14. Diagrama de estado compras



## 5.2.8 Diagramas de pagos

### Diagrama de flujo de datos para el caso de uso pagos

En la Figura 16 se muestra el diagrama de flujo de datos para el caso de uso pagos.

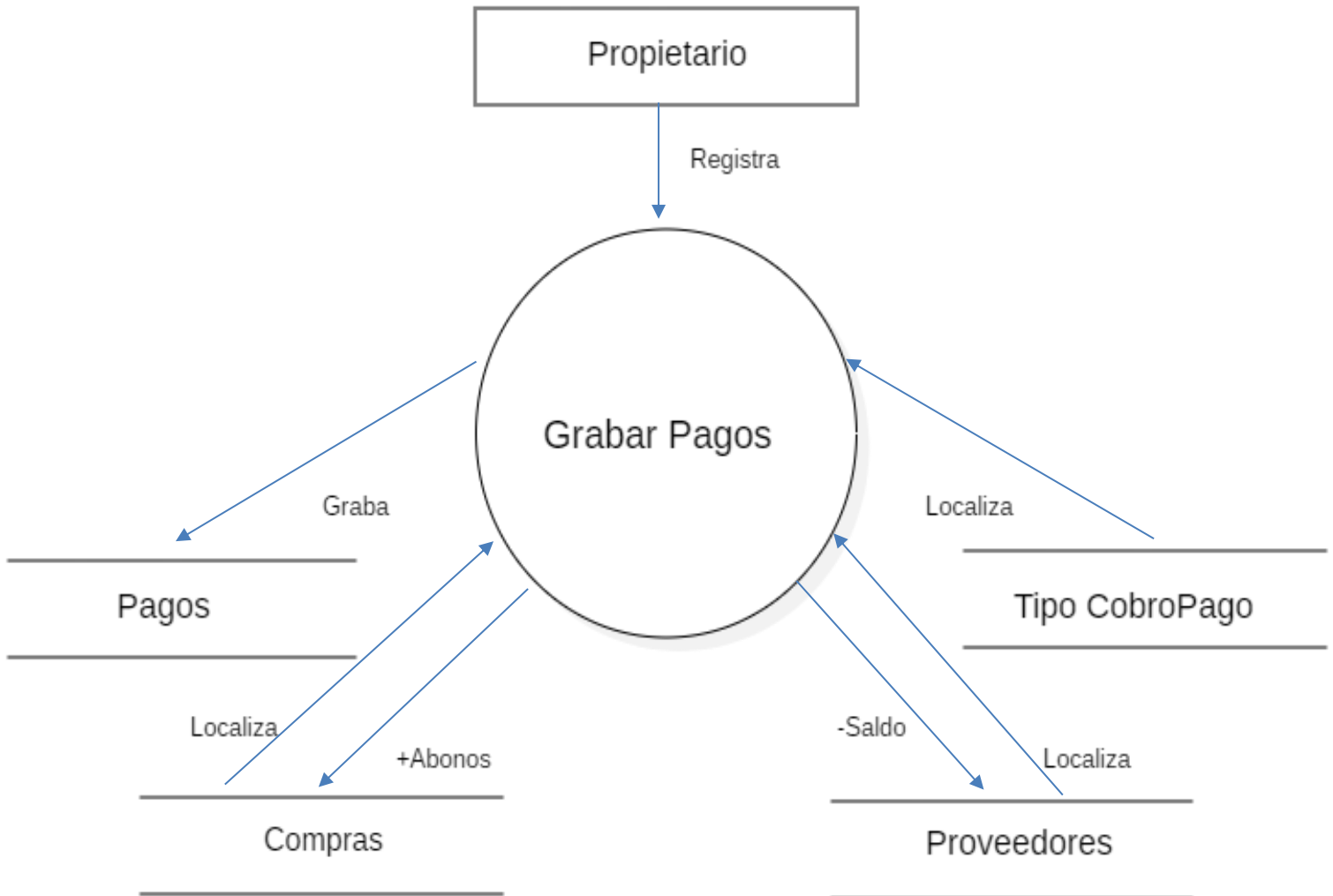


Figura 16. Diagrama de flujo de datos pagos

## Diagrama de caso de uso expandido para el caso de uso pagos

En la Figura 17 se muestra el diagrama de caso de uso expandido para el caso de uso pagos.

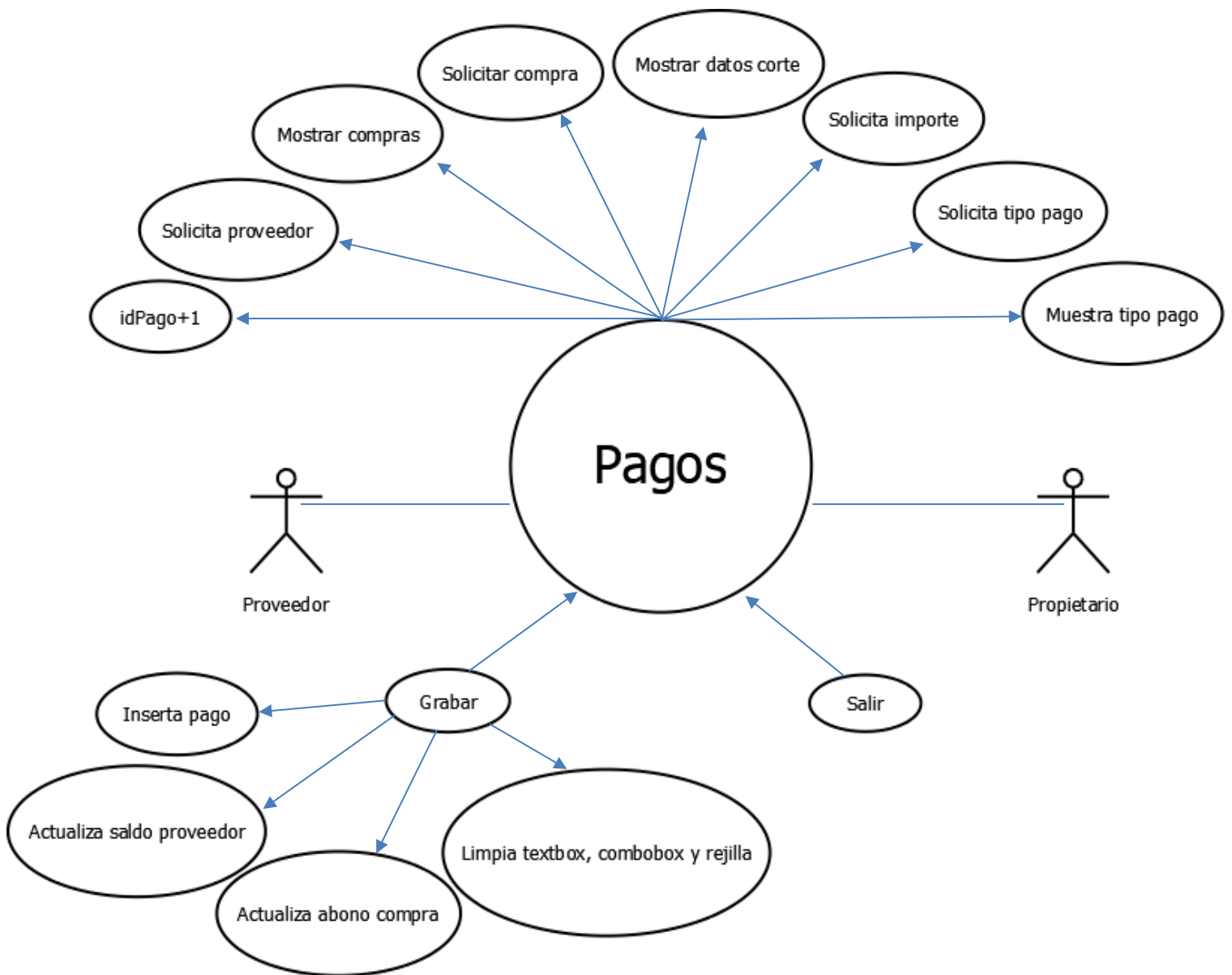


Figura 17. Diagrama de caso de uso expandido pagos



## Diagrama de estado para el caso de uso pagos

En la Figura 18 se muestra el diagrama de estado para el caso de uso pagos.

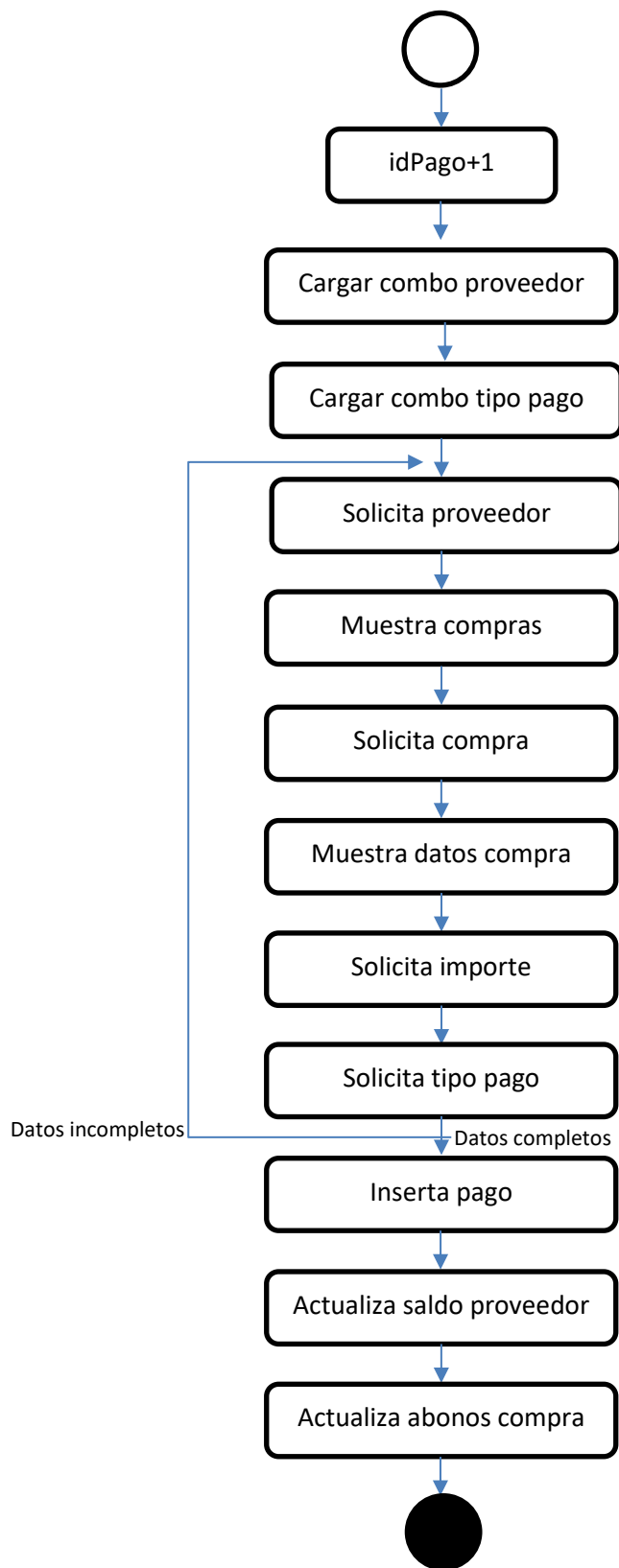


Figura 18. Diagrama de estado pagos



## 5.2.9 Diagramas de cobros

### Diagrama de flujo de datos para el caso de uso cobros

En la Figura 20 se muestra el diagrama de flujo de datos para el caso de uso cobros.

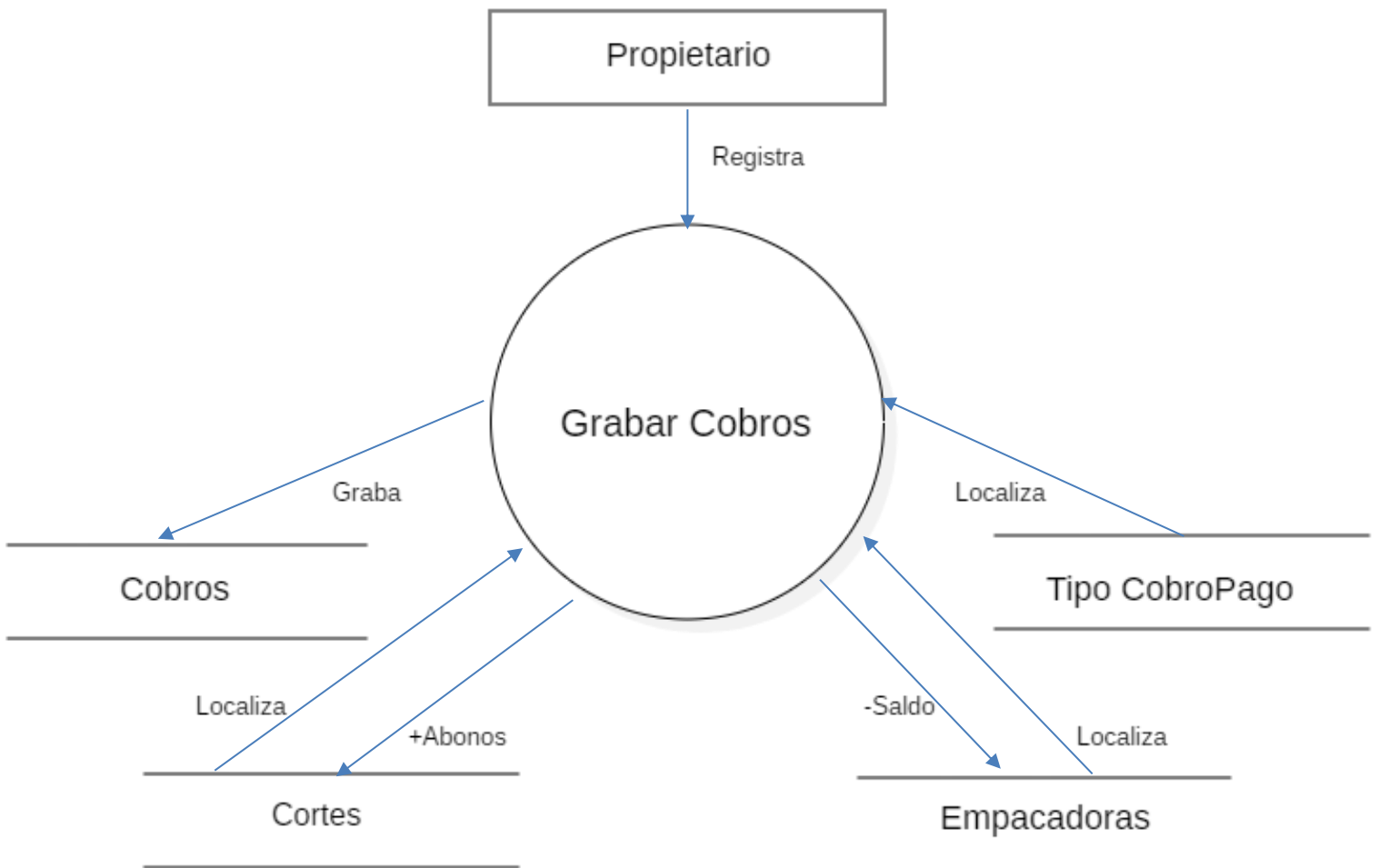


Figura 20. Diagrama de flujo de datos cobros

## Diagrama de caso de uso expandido para el caso de uso grabar cobros

En la Figura 21 se muestra el diagrama de caso de uso expandido para el caso de uso cobros.

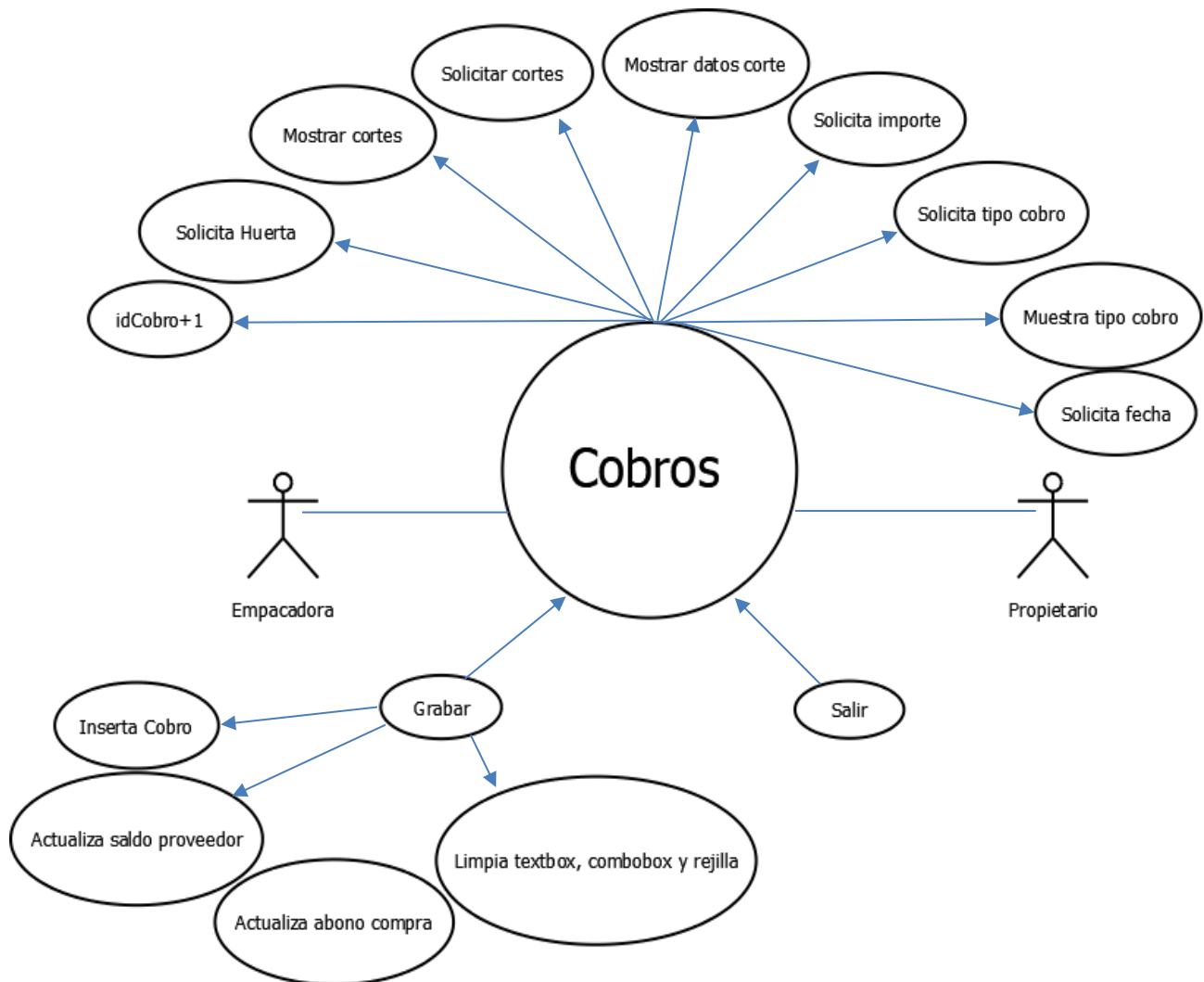


Figura 21. Diagrama de caso de uso expandido cobros

## Diagrama de estado para el caso de uso cobros

En la Figura 22 se muestra el diagrama de estado para el caso de uso cobros.

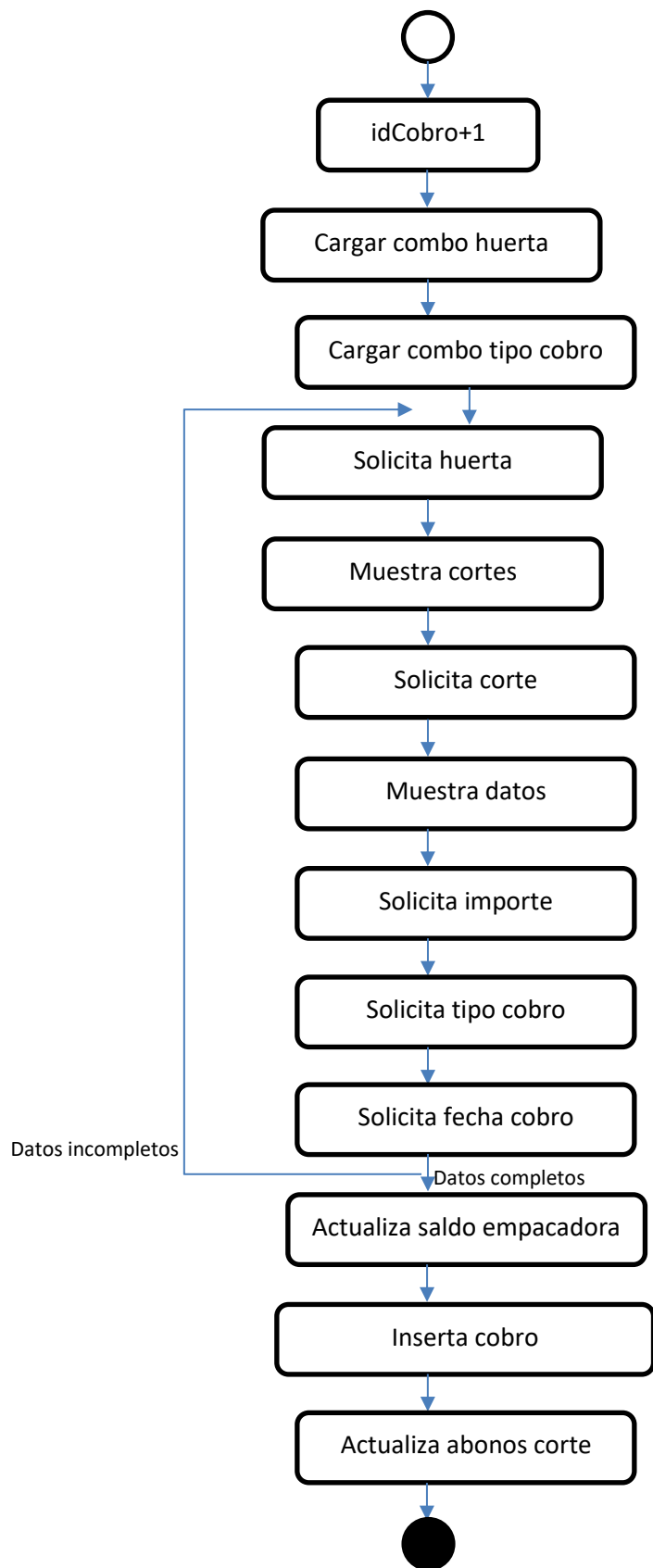


Figura 22. Diagrama de estado cobros

## Diagrama de colaboración para el caso de uso cobros

En la Figura 23 se muestra el diagrama de colaboración para el caso de uso cobros.

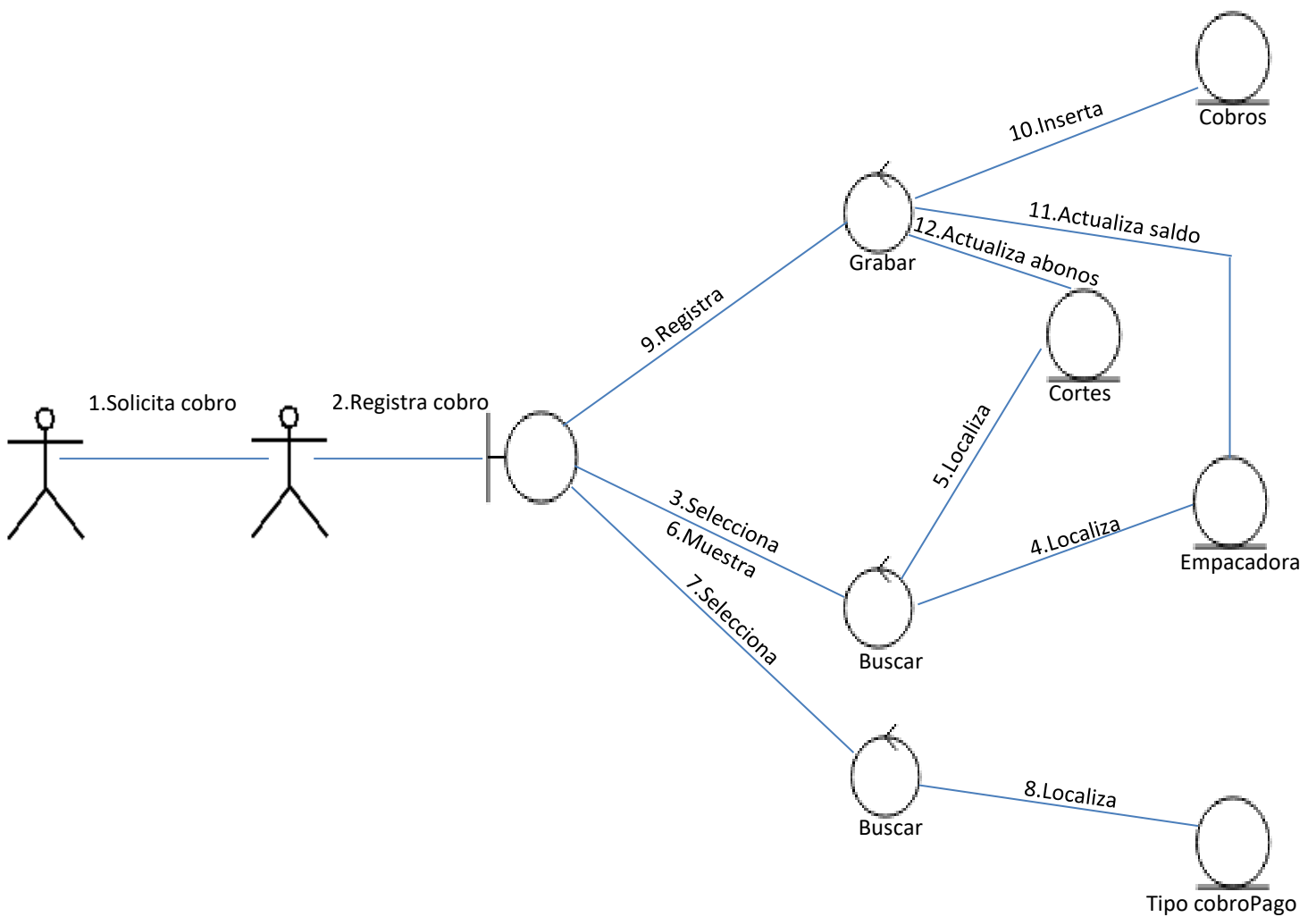


Figura 23. Diagrama de colaboración cobros

## 5.2.10 Diagramas de actividades

### Diagrama de flujo de datos para el caso de uso actividades

En la Figura 24 se muestra el diagrama de flujo de datos para el caso de uso actividades.

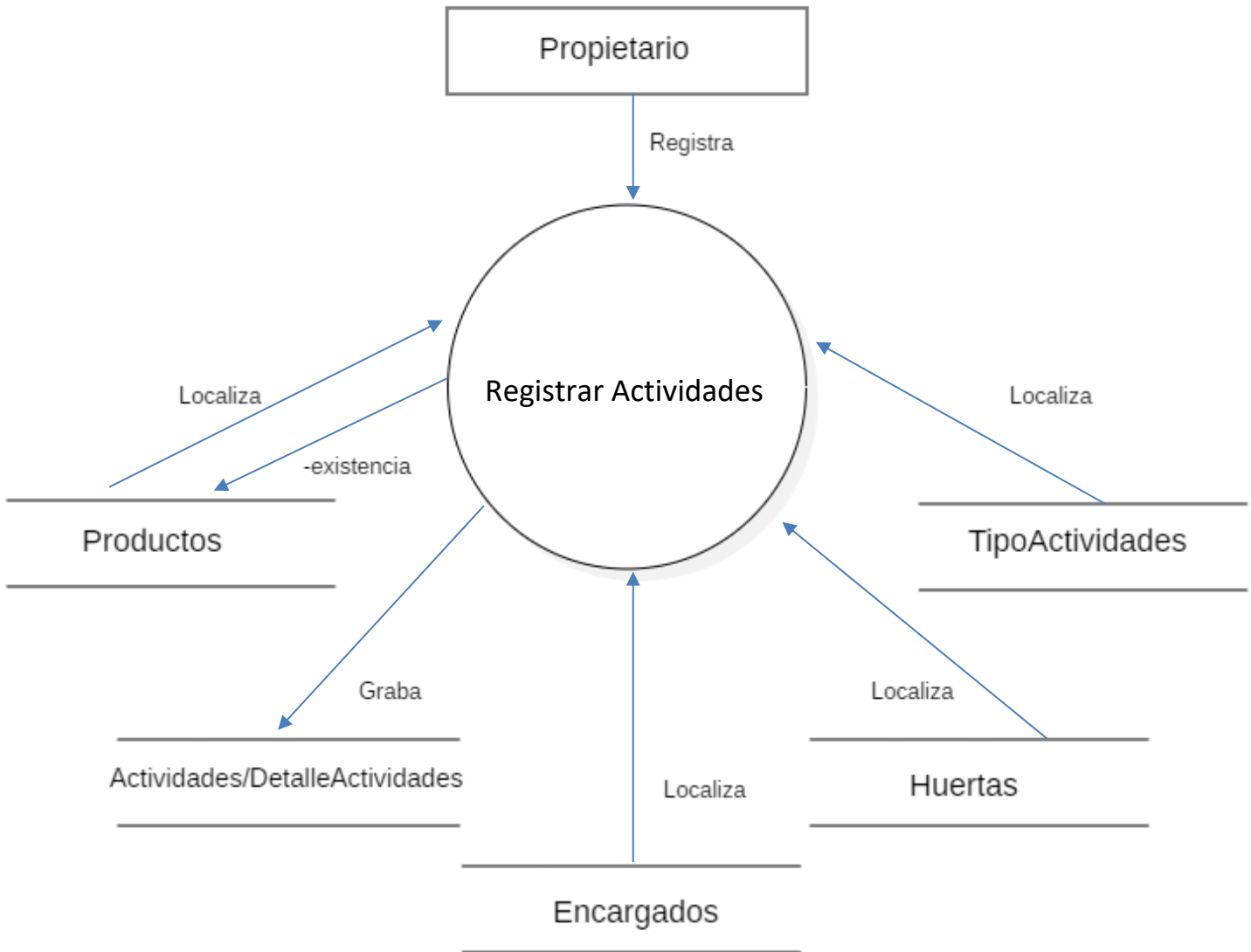


Figura 24. Diagrama de flujo de datos actividades

## Diagrama de caso de uso expandido para el caso de uso actividades

En la Figura 25 se muestra el diagrama de caso de uso expandido para el caso de uso actividades.



Figura 25. Diagrama de caso de uso expandido actividades



## Diagrama de estado para el caso de uso actividades

En la Figura 26 se muestra el diagrama de estado para el caso de uso actividades.

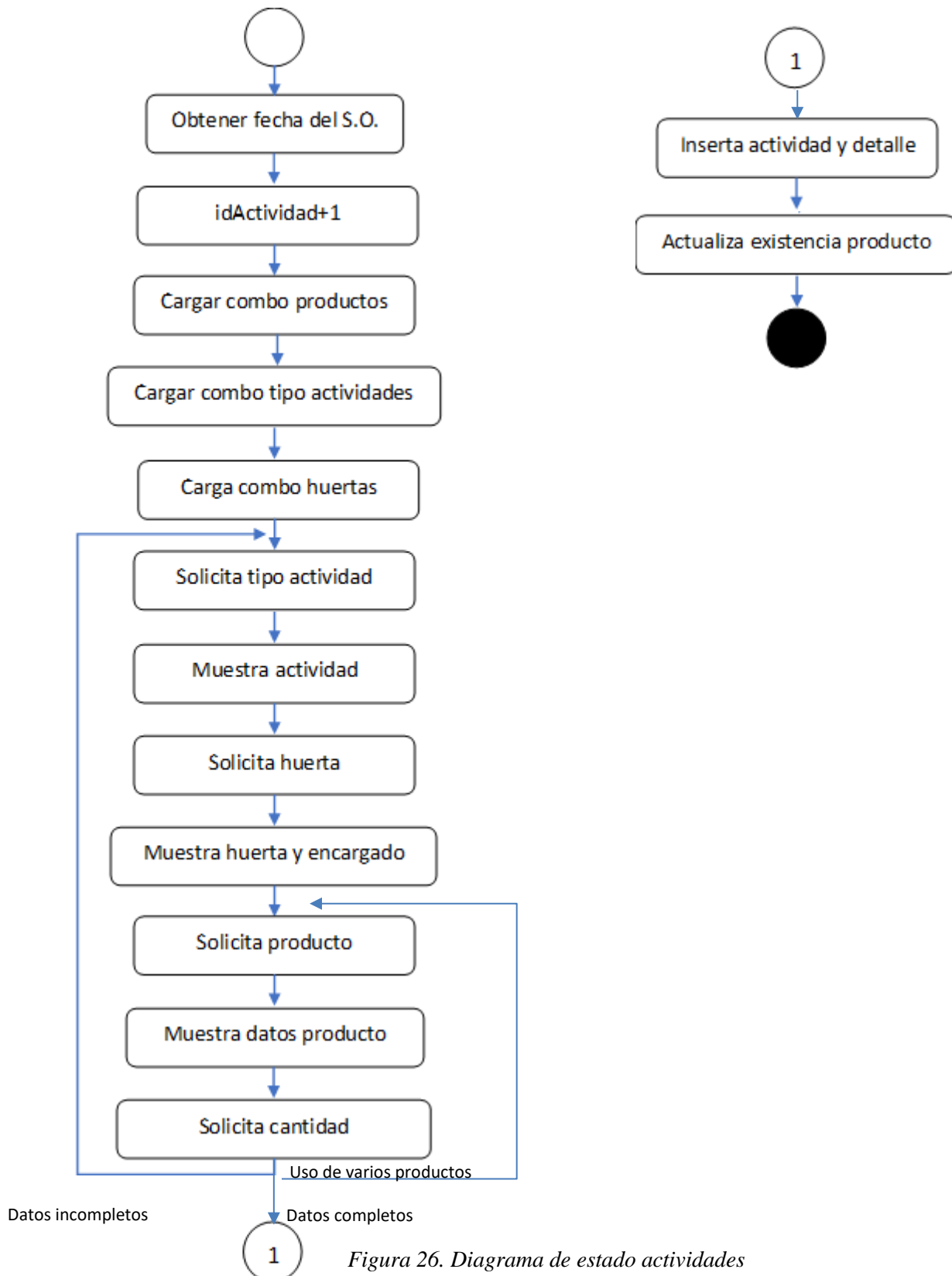


Figura 26. Diagrama de estado actividades

## Diagrama de colaboración para el caso de uso actividades

En la Figura 27 se muestra el diagrama de colaboración para el caso de uso actividades.

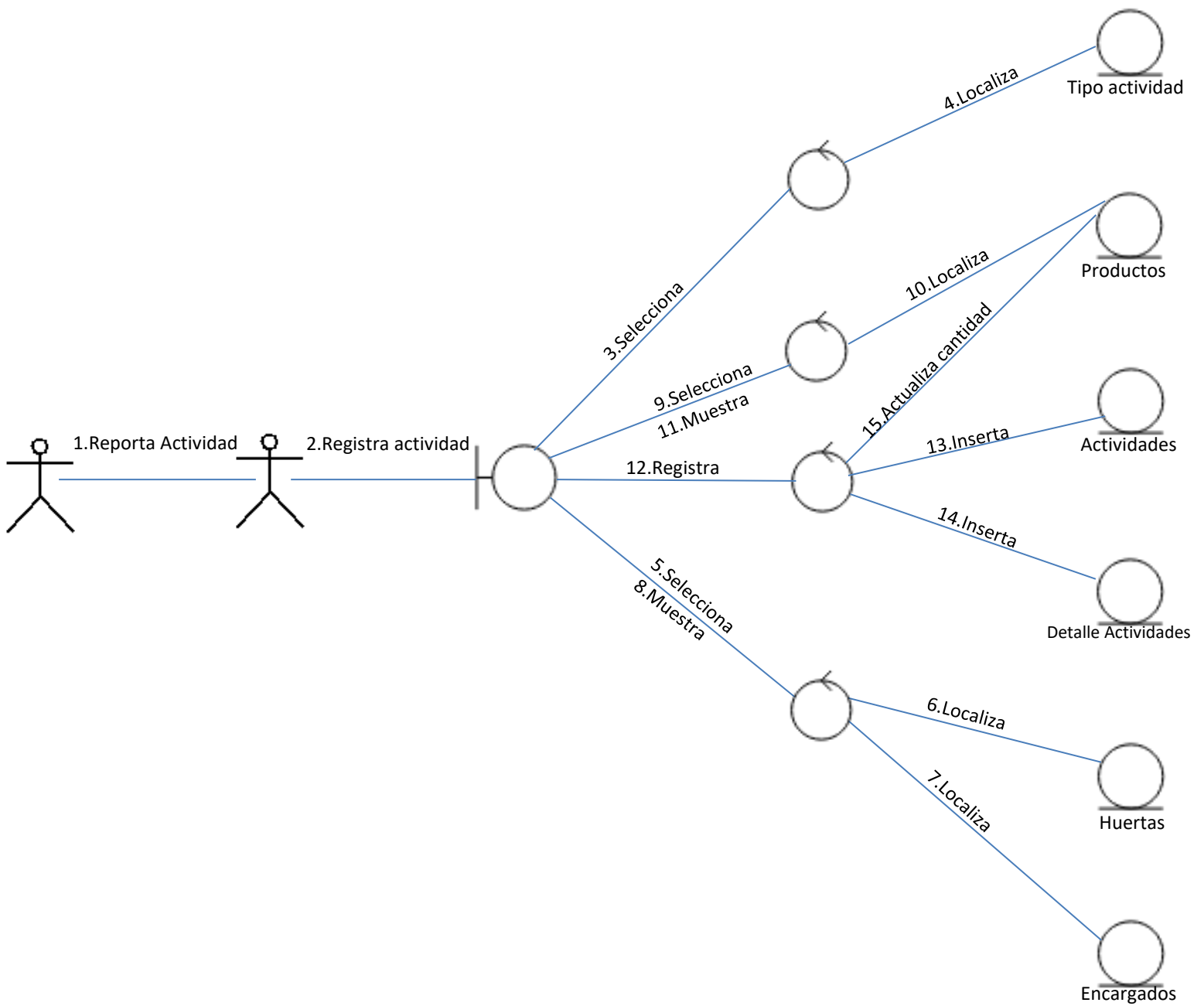


Figura 27. Diagrama de colaboración actividades

## 5.3 Codificación

### 5.3.1 Diagrama de módulos, formularios, consultas y reportes de la aplicación de escritorio

En las Figuras 28 y 29 se muestran los catálogos de la aplicación de escritorio.

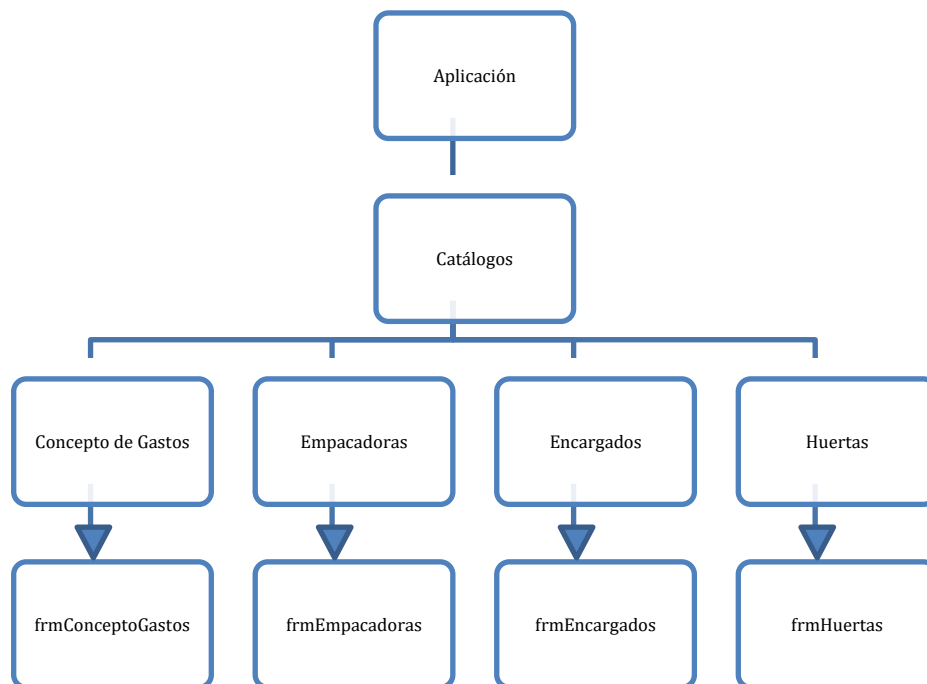


Figura 28. Diagrama de catálogos de escritorio 1

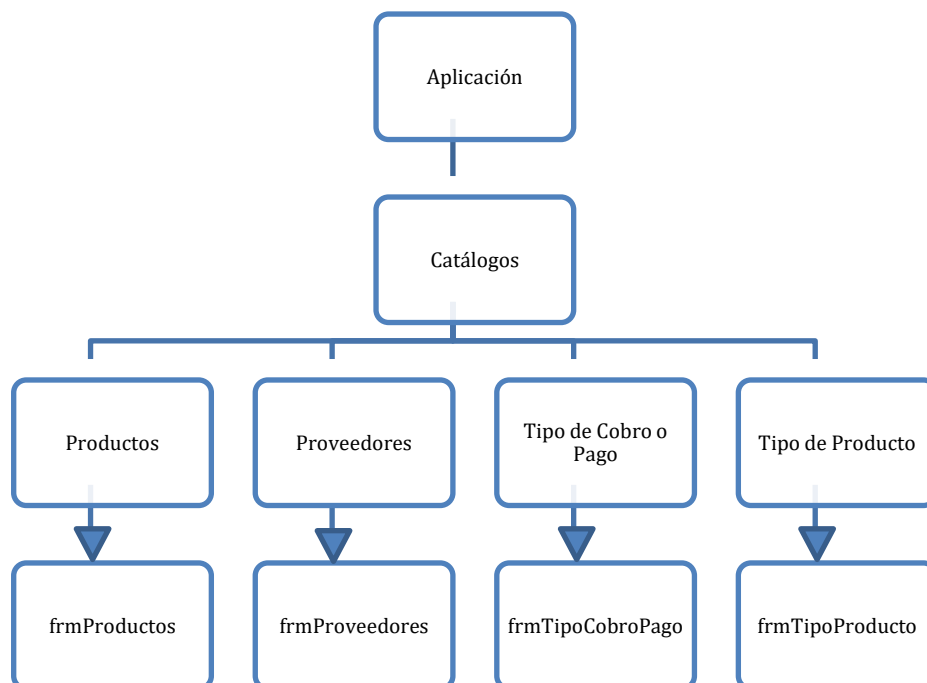


Figura 29. Diagrama de catálogos de escritorio 2

En la Figura 30 se muestran los movimientos de la aplicación de escritorio.

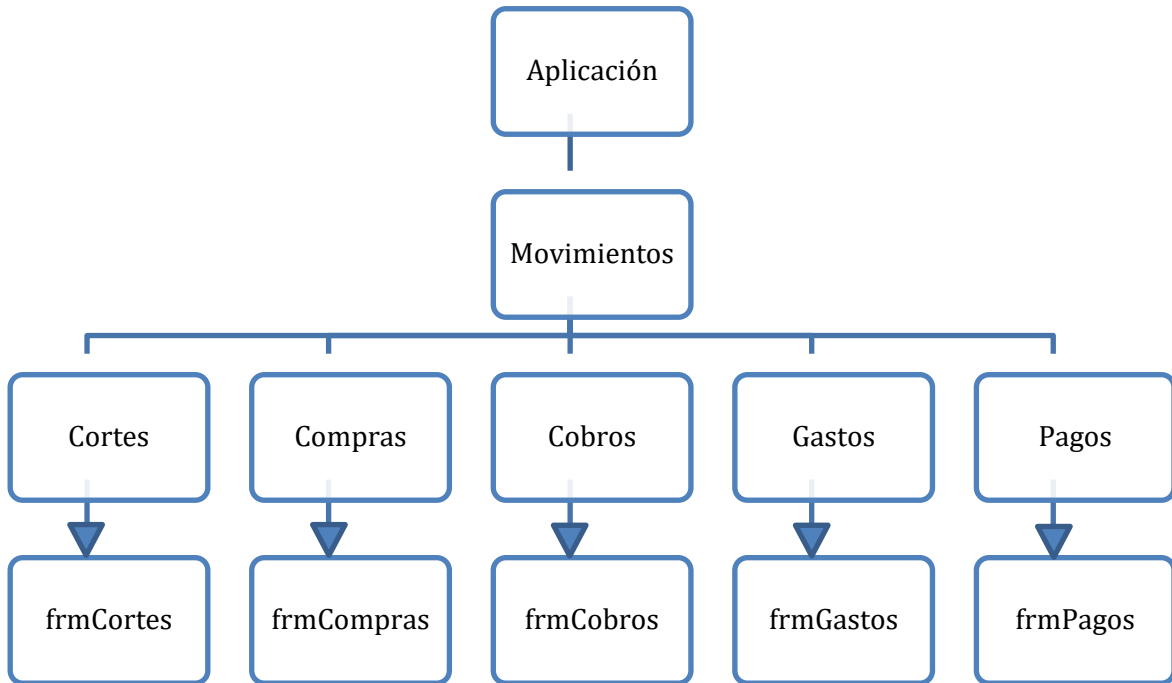


Figura 30. Diagrama de movimientos de escritorio

En la Figura 31 se muestran las consultas y reportes de la aplicación de escritorio.

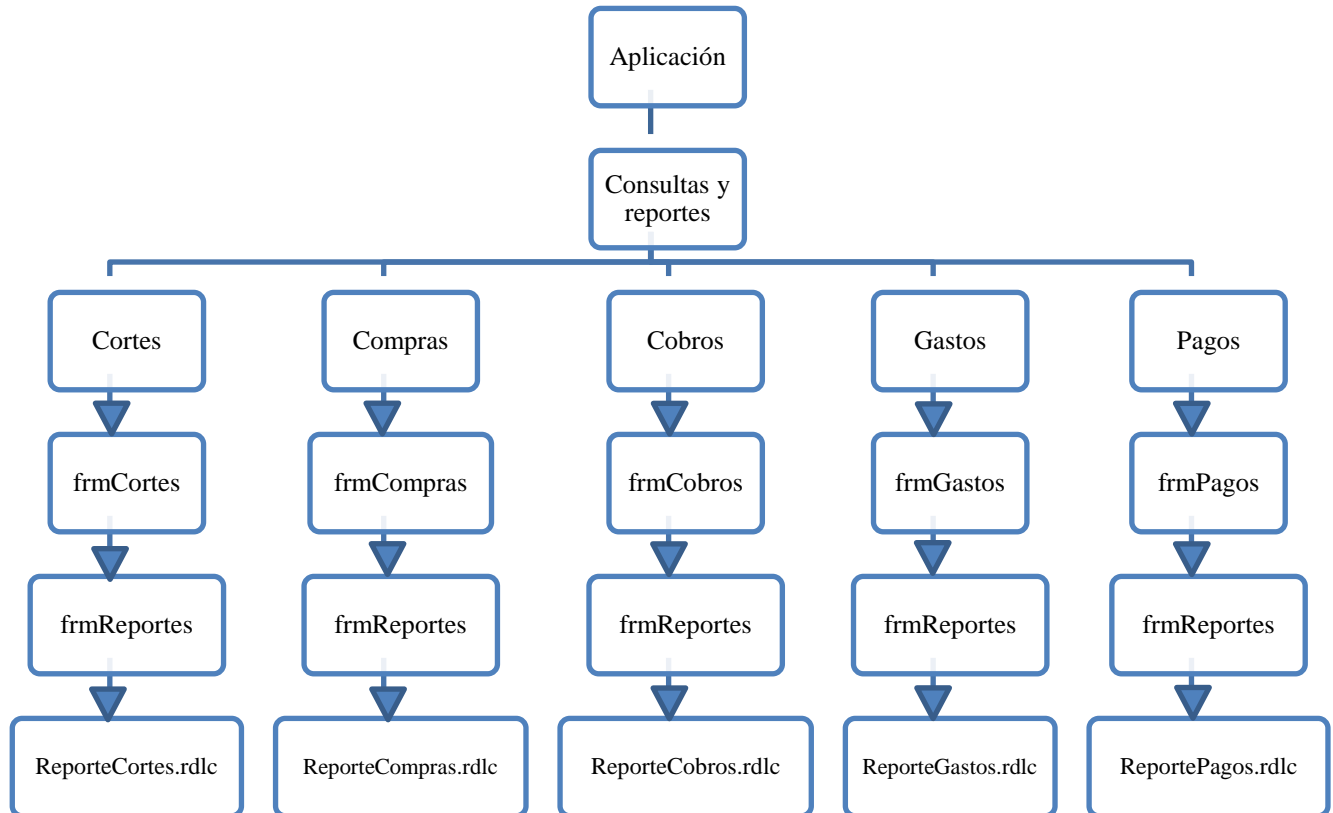


Figura 31. Diagrama de consultas y reportes de escritorio

### 5.3.2 Diagrama de módulos, formularios, consultas y reportes de la página Web

En la Figura 32 se muestran los catálogos y el inicio de sesión de la página Web.

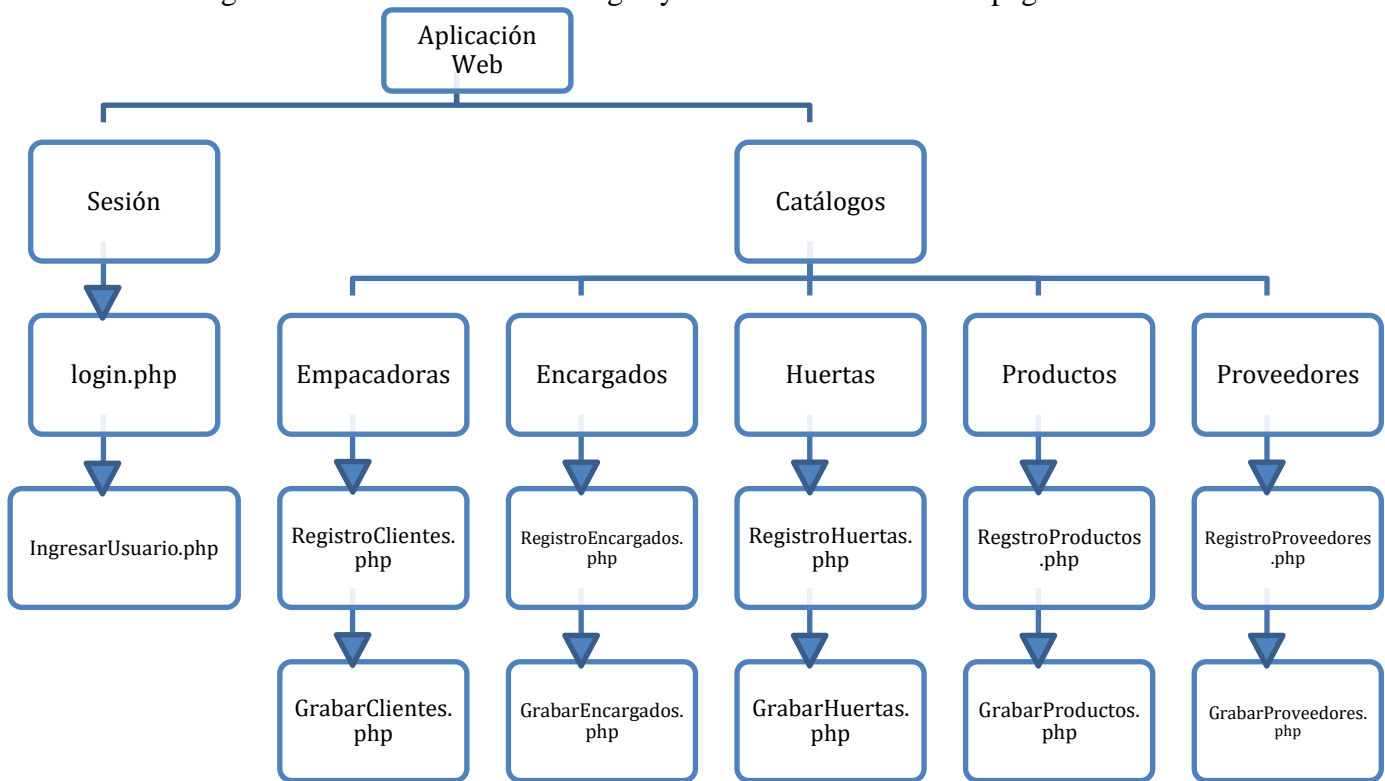


Figura 32. Diagrama de catálogos e inicio de sesión Web

En las Figura 33 se muestran las consultas de catálogos de la página Web.

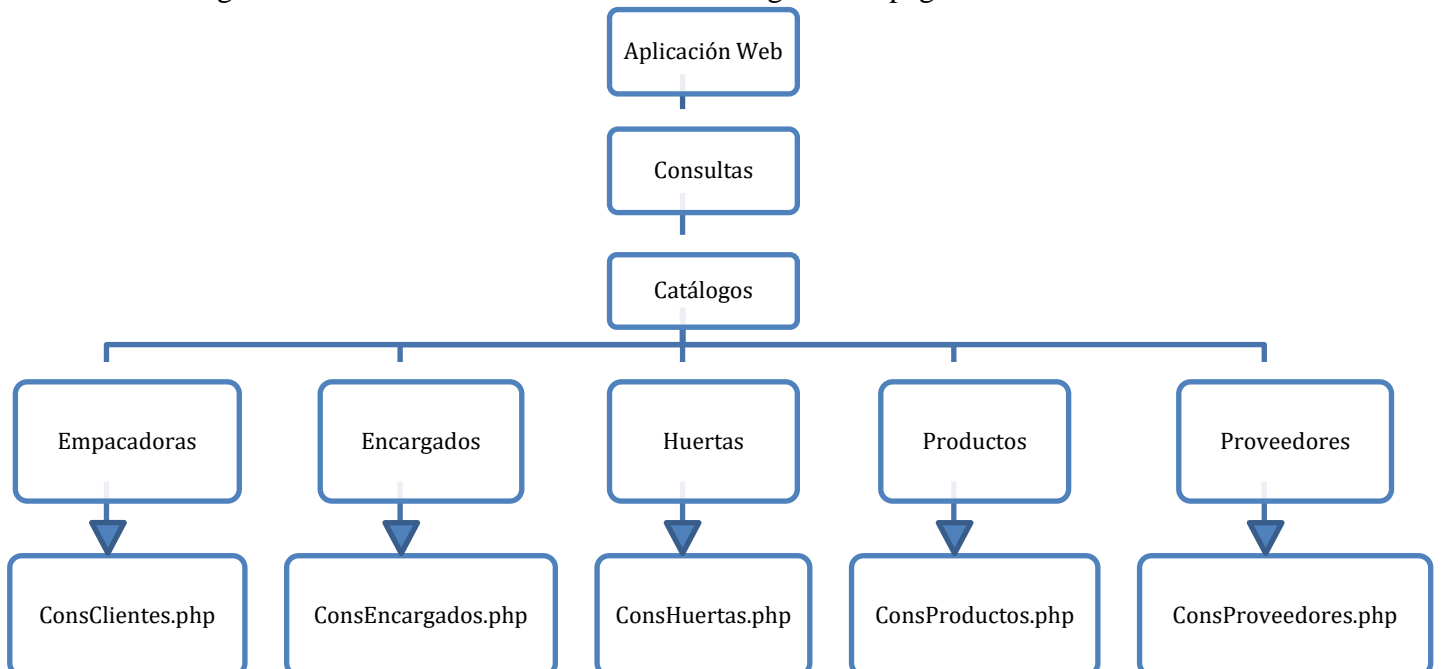


Figura 33. Diagrama de consultas de catálogos Web

En la Figura 34 se muestran las consultas de movimientos de la página Web.

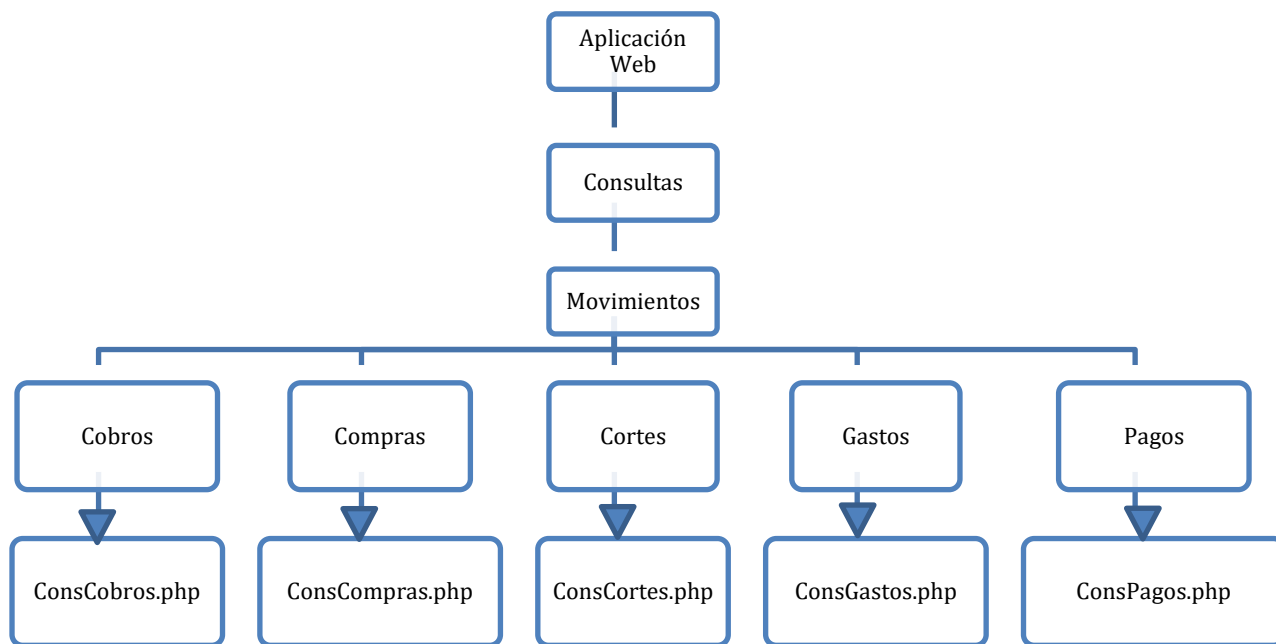


Figura 34. Diagrama de consultas de movimientos Web

En la Figura 35 se muestran los reportes de catálogos de la página Web.

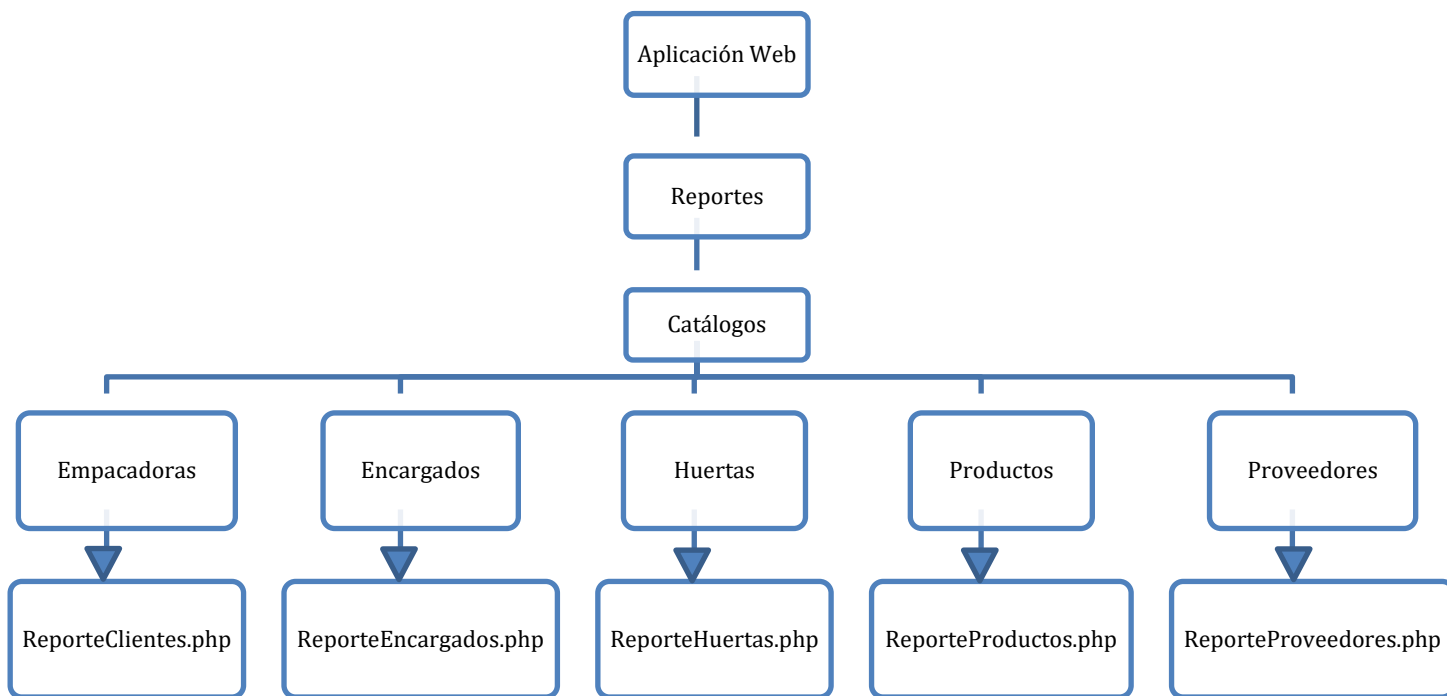


Figura 35. Diagrama de reportes de catálogos Web

En la Figura 36 se muestran los reportes de movimientos de la página Web.

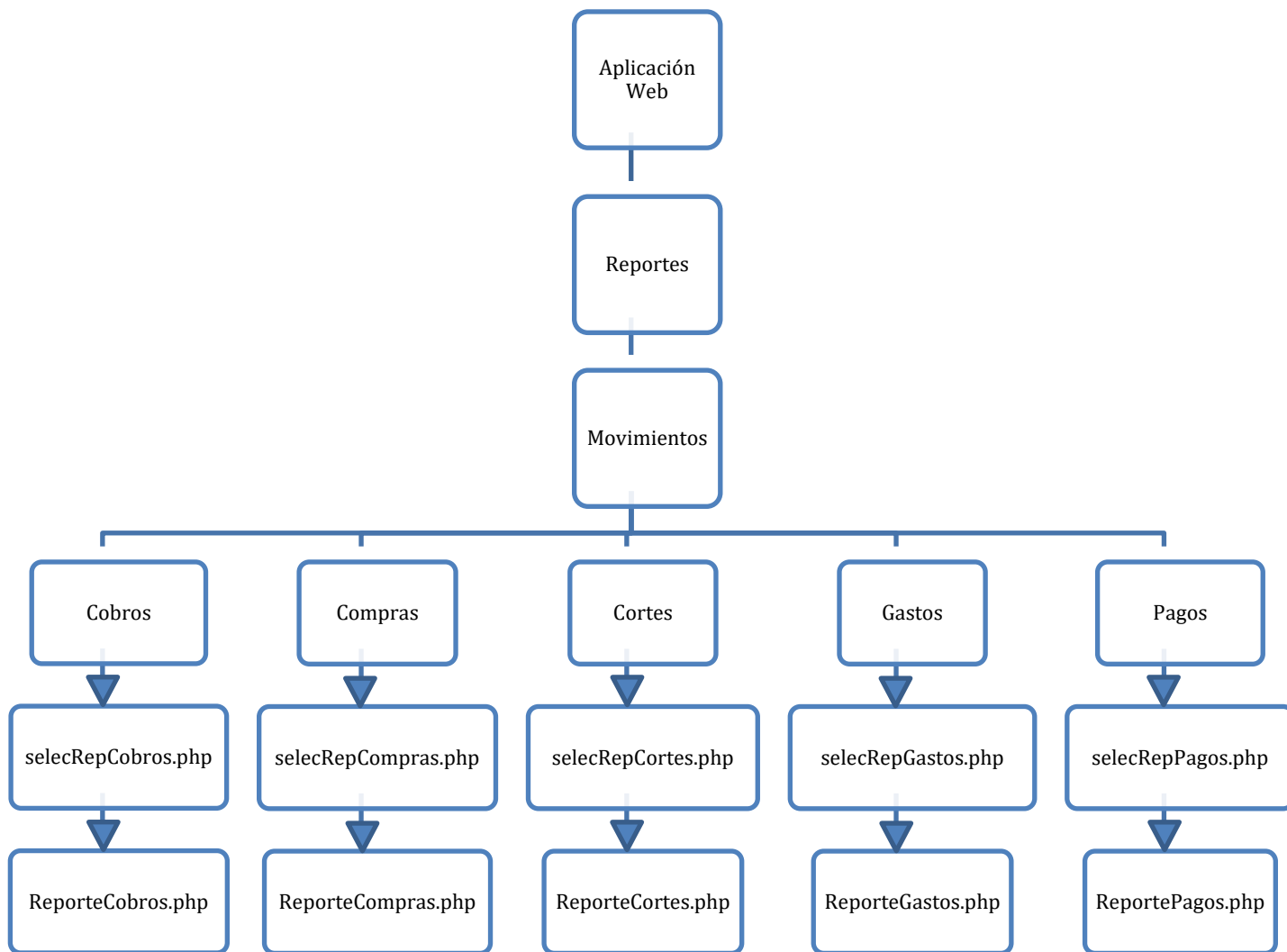


Figura 36. Diagrama de reportes de movimientos Web

### 5.3.3 Diagrama de módulos, formularios y consultas de la aplicación móvil

En la Figura 37 se muestran las consultas de los movimientos de la aplicación móvil.

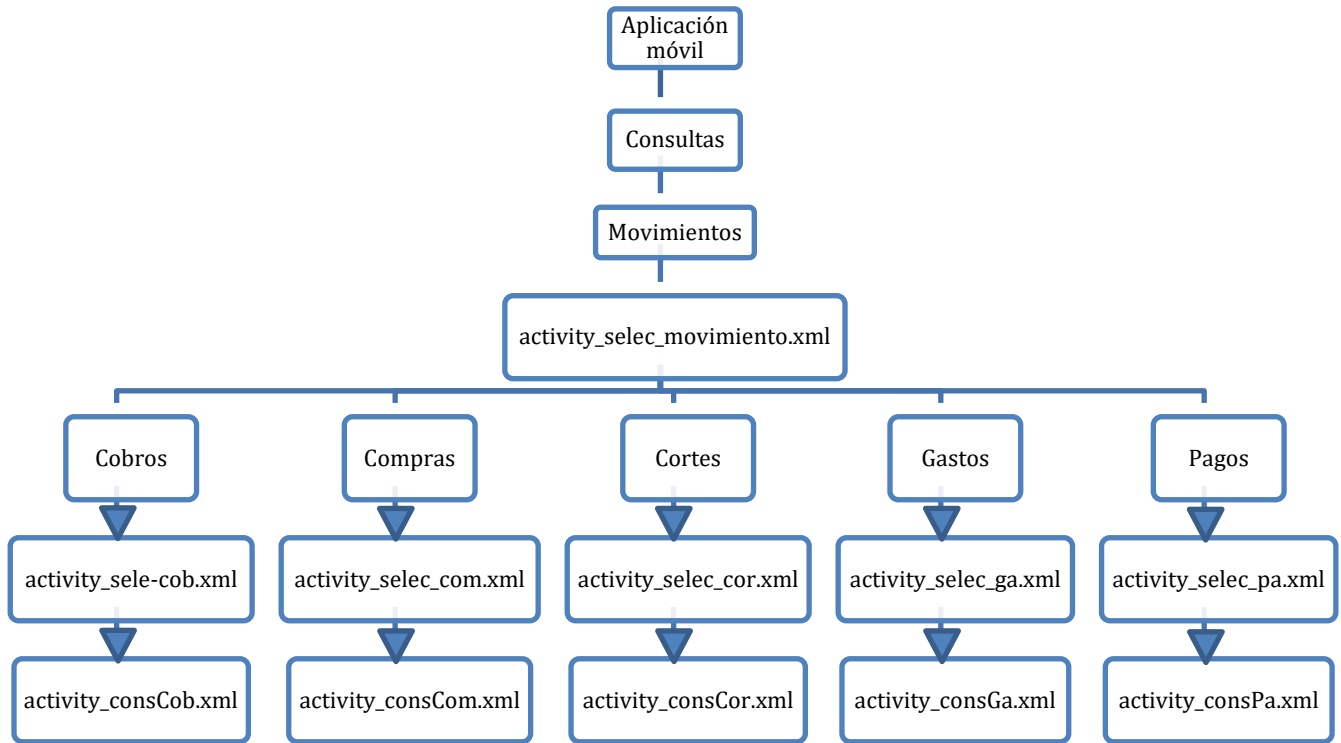


Figura 37. Diagrama de consulta de movimientos móvil

En la Figura 38 se muestra el inicio de sesión, ayuda y cerrar sesión de aplicación móvil.

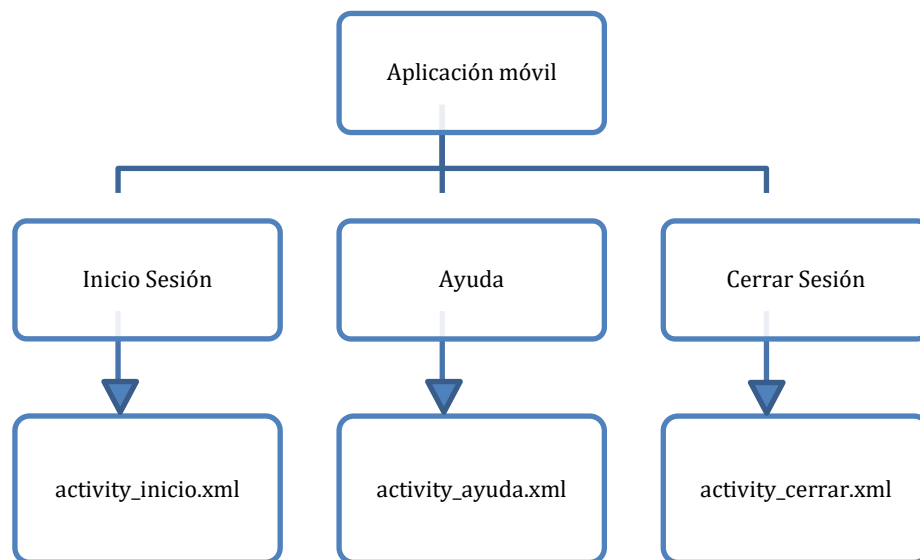


Figura 38. Diagrama de inicio, ayuda y cerrar sesión móvil



En la Figura 39 se muestran las consultas de los catálogos de la aplicación móvil.



Figura 39. Diagrama de consultas de catálogos móvil

## 5.4 Pruebas

### 5.4.1 Pruebas en la aplicación de escritorio

En la Tabla 1 se muestra la primera prueba para realizar el movimiento de cortes.

*Tabla 1. Prueba registrar corte 1*

Nombre	Cortes
Descripción	Registrar un corte en el sistema con 250 cajas, 500 toneladas y \$700,000 de ingreso generado.
Datos de entrada	idCorte: 1 Huerta: El Cincuenta 1 Empacadora: Empacadora 1 Número de Cajas: 250 Fecha: 12 de abril del 2019 Toneladas: 500 Ingreso Generado \$ 700,000
Condiciones	El corte es realizado por la Empacadora 1 en la huerta El Cincuenta 1. El saldo inicial de la empacadora es \$500. El número de cajas es 250, las toneladas 500 y el ingreso generado \$700,000.
Secuencia Principal	<ol style="list-style-type: none"><li>1. El usuario presiona Nuevo para generar el idCorte.</li><li>2. Selecciona la huerta a la que se efectuó el corte.</li><li>3. Selecciona la empacadora por la cual se efectuó el corte.</li><li>4. Ingresa el número de cajas extraído.</li><li>5. Ingresa la fecha en la que se llevó a cabo el corte.</li><li>6. Ingresa las toneladas resultantes del corte.</li><li>7. Agrega el ingreso generado.</li><li>8. El sistema registra el corte.</li></ol>
Alternativas y Errores	<ol style="list-style-type: none"><li>1. No hay conexión a Internet.</li><li>2. La huerta no está registrada.</li><li>3. La empacadora no está registrada.</li></ol>
Salida	Graba en cortes Actualiza saldo en Empacadora Saldo final de la empacadora \$700,500

En la Tabla 2 se muestra la segunda prueba con datos diferentes para realizar el movimiento de cortes.

*Tabla 2. Prueba registrar corte 2*

Nombre	Cortes
Descripción	Registrar un corte en el sistema con 50 cajas, 100 toneladas y \$50,000 de ingreso generado.
Datos de entrada	idCorte: 2 Huerta: El Cincuenta Empacadora: Empacadora 1 Número de Cajas: 50 Fecha: 11 de abril del 2019 Toneladas: 100 Ingreso Generado \$ 50,000
Condiciones	El corte es realizado por la Empacadora 1 en la huerta El Cincuenta. Saldo inicial de la empacadora \$700,500 El número de cajas es 50, las toneladas 100 y el ingreso generado \$50,000.
Secuencia Principal	<ol style="list-style-type: none"> <li>1. El usuario presiona Nuevo para generar el idCorte.</li> <li>2. Selecciona la huerta a la que se efectuó el corte.</li> <li>3. Selecciona la empacadora por la cual se efectuó el corte.</li> <li>4. Ingresa el número de cajas extraído.</li> <li>5. Ingresa la fecha en la que se llevó a cabo el corte.</li> <li>6. Ingresa las toneladas resultantes del corte.</li> <li>7. Agrega el ingreso generado.</li> <li>8. El sistema registra el corte.</li> </ol>
Alternativas y Errores	<ol style="list-style-type: none"> <li>1. No hay conexión a Internet.</li> <li>2. La huerta no está registrada.</li> <li>3. La empacadora no está registrada.</li> </ol>
Salida	Graba en cortes Actualiza saldo en Empacadora Saldo final de la empacadora \$750,500

En la Tabla 3 se muestra la primera prueba para realizar el movimiento de compras.

*Tabla 3. Prueba registrar compra 1*

Nombre	Compras
Descripción	Registrar una compra al contado de 4 ejemplares de un producto con IVA DEL 16%.
Datos de entrada	idCompra: 1 Factura: FACTURA123 Proveedor: Ferretek Condición Compra: Contado Fecha Compra: Obtenida del sistema Producto: FUMIX Cantidad: 4 Precio \$ 500 IVA: 16
Condiciones	La compra es de contado. El IVA es del 16%. Existencia del producto 10. Saldo inicial del proveedor \$500.
Secuencia Principal	1. Ingresar la factura de la compra. 2. Seleccionar de Proveedor al que se realizó la compra. 3. Seleccionar el tipo de pago al Contado. 4. Seleccionar el producto. 5. Ingresar la cantidad. 6. Presionar el botón Agregar. 7. Ingresar el IVA. 8. El sistema registra la compra.
Alternativas y Errores	1. No hay conexión a Internet. 2. El proveedor no está registrado.
Salida	Subtotal \$2,000. Total \$ 2,329. Graba en compras. Actualiza cantidad en producto FUMIX. Existencia de producto final 14. Saldo final del proveedor \$2,829.

En la Tabla 4 se muestra la segunda prueba con datos para realizar el movimiento de compras.

*Tabla 4. Prueba registrar compra 2*

Nombre	Compras
Descripción	Registrar una compra a crédito de 4 ejemplares de un producto y 2 de otro con IVA del 16%.
Datos de entrada	idCompra: 2 Factura: FACTURA666 Proveedor: Ferretek Condición Compra: CRÉDITO Fecha Compra: Obtenida del sistema Producto1 FUMIX Cantidad: 4 Precio \$ 500 Producto 2: ABONIX Cantidad: 2 Precio \$ 100 IVA: 16
Condiciones	La compra es a crédito. El IVA es del 16%. Serán dos productos diferentes con diferente cantidad. Cantidad inicial de producto FUMIX 14. Cantidad inicial de producto ABONIX 0. Saldo inicial del proveedor \$2,829.
Secuencia Principal	1. Ingresar la factura de la compra. 2. Seleccionar proveedor al que se realizó la compra. 3. Seleccionar el tipo de pago a Crédito. 4. Ingresar la fecha de vencimiento. 5. Seleccionar el producto 1. 6. Ingresar la cantidad. 7. Presionar el botón Agregar. 8. Seleccionar el producto 2. 9. Ingresar la cantidad. 10. Presionar agregar. 11. Ingresar el IVA. 12. El sistema registra la compra.
Alternativas y Errores	1. No hay conexión a Internet. 2. El proveedor no está registrado.
Salida	Subtotal \$2,200. Total \$ 2,552. Grabar en compras y actualizar cantidad en productos. Cantidad de producto ABONIX final 2. Cantidad final de producto FUMIX 18. Actualizar saldo en Proveedor. Saldo final del proveedor \$5,381.

## 5.4.2 Pruebas en la página Web

En la Tabla 5 se muestra la primera prueba para realizar el registro de un producto.

*Tabla 5. Prueba registrar producto 1*

Nombre	Productos
Descripción	Registrar un producto con costo \$300 y existencia 5 del tipo herbicida con unidad de medida Litros.
Datos de entrada	idProducto: 3 Tipo: Herbicida Descripción: HERBIX Costo \$ 300 Unidad de Medida: Litros Existencia 5
Condiciones	Tipo herbicida. Unidad de medida litros.
Secuencia Principal	1. Selecciona el tipo de producto. 2. Ingresar la descripción. 3. Ingresar el costo. 4. Selecciona la unidad de medida. 5. Ingresar la existencia. 6. La página registra el producto.
Alternativas y Errores	1. No hay conexión a Internet. 2. El tipo de producto no está registrado.
Salida	Graba en productos.

En la Tabla 6 se muestra la segunda prueba para realizar el registro de un producto con datos diferentes.

*Tabla 6. Prueba registrar producto 2*

Nombre	Productos
Descripción	Registrar un producto con costo \$500 y existencia 2 del tipo abono con unidad de medida Kilos.
Datos de entrada	idProducto: 4 Tipo: Abono Descripción: ABONIX Costo \$ 500 Unidad de Medida: Kilos Existencia 2
Condiciones	Tipo Abono. Unidad de medida kilos.
Secuencia Principal	1. Selecciona el tipo de producto. 2. Ingresa la descripción. 3. Ingresa el costo. 4. Selecciona la unidad de medida. 5. Ingresa la existencia. 6. La página registra el producto.
Alternativas y Errores	1. No hay conexión a Internet. 2. El tipo de producto no está registrado.
Salida	Graba en productos.

## 5.5 Implementación

La base de datos se instaló en remotemysql.com, un servidor gratuito que soporta MySQL. La página Web se subió al servidor <https://www.000webhost.com>, que ofrece alojamiento gratuito para páginas Web con soporte MySQL y PHP.

Se compartió el instalable de la aplicación móvil con la empresa “Huertas el Cincuenta 1” en dos teléfonos celulares con posible instalación en más dispositivos.

Se instaló la aplicación de escritorio en la computadora de la oficina de la empresa y se capacitó al personal en el uso de cada una de las aplicaciones o escenarios que conforman la plataforma (Web, escritorio y móvil), además se realizaron pruebas necesarias para verificar el perfecto funcionamiento de ésta.

## 5.6 Documentación

Se realizó la documentación completa del sistema, incluyendo la ayuda en la aplicación de escritorio, así como el manual de usuario y programación en cada una de las aplicaciones.

## 6. RESULTADOS

En este capítulo se presenta la interfaz de la plataforma, iniciando con las pantallas de la aplicación de escritorio mostrando algunos menús, catálogos, movimientos, consultas y reportes, posteriormente se muestra la página Web, mostrando el menú principal y algunos reportes y finalmente se presenta la aplicación móvil mostrando consultas y algunas pantallas tales el menú.

### 6.1 Aplicación de Escritorio

#### 6.1.1 Menú

En la Figura 40 se muestra el menú catálogos de la aplicación de escritorio.



*Figura 40. Menú catálogos de escritorio*



En la Figura 41 se muestra el menú movimientos de la aplicación de escritorio.

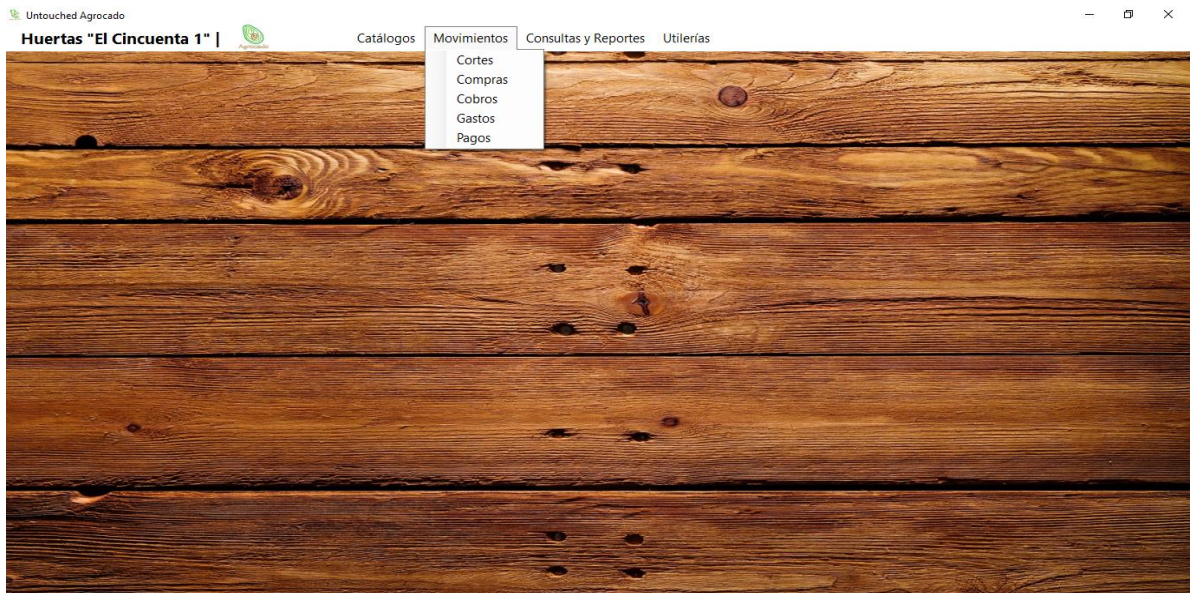


Figura 41. Menú movimientos de escritorio

## 6.1.2 Catálogos

En la Figura 42 se muestra la pantalla registrar empacadoras de la aplicación de escritorio donde se pueden observar algunos datos registrados en la base de datos, los registros más recientes aparecen en esta pantalla.

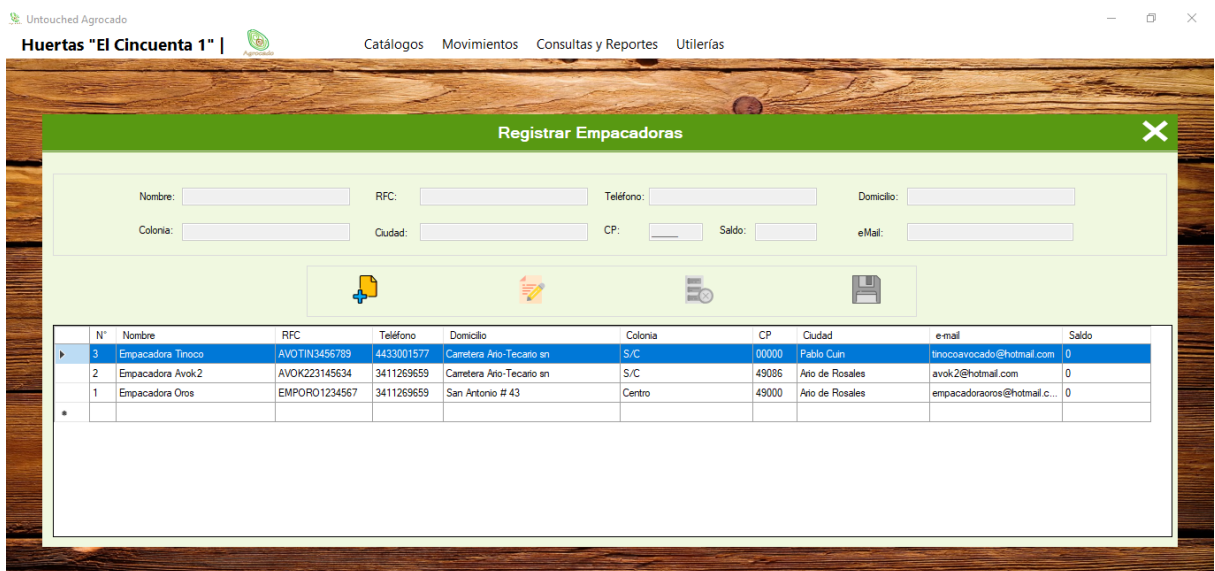


Figura 42. Registrar empacadoras de escritorio

En la Figura 43 se muestra la pantalla registrar huertas de la aplicación de escritorio donde aparecen las opciones para guardar un nuevo registro y una pequeña consulta general donde aparecerán los registros más recientes.

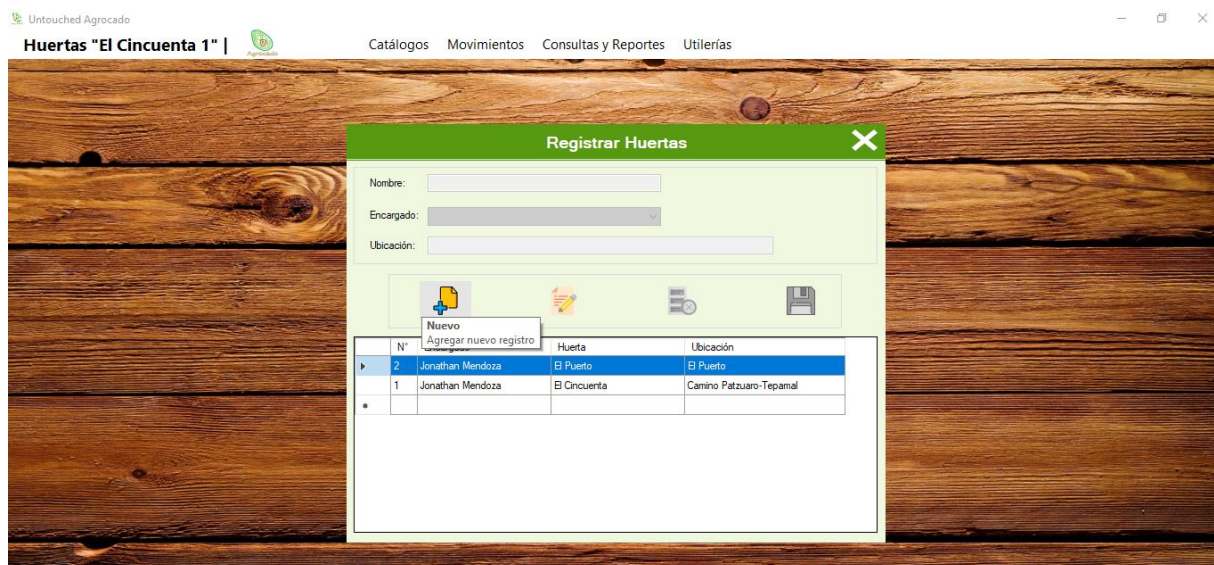


Figura 43. Registrar huertas de escritorio

### 6.1.3 Movimientos

En la Figura 44 se muestra la pantalla registrar cortes de la aplicación de escritorio donde se puede observar el registro de datos de un corte de fruta hecho a la huerta “El Cincuenta” además contiene una pequeña consulta general donde aparecen los registros más recientes.

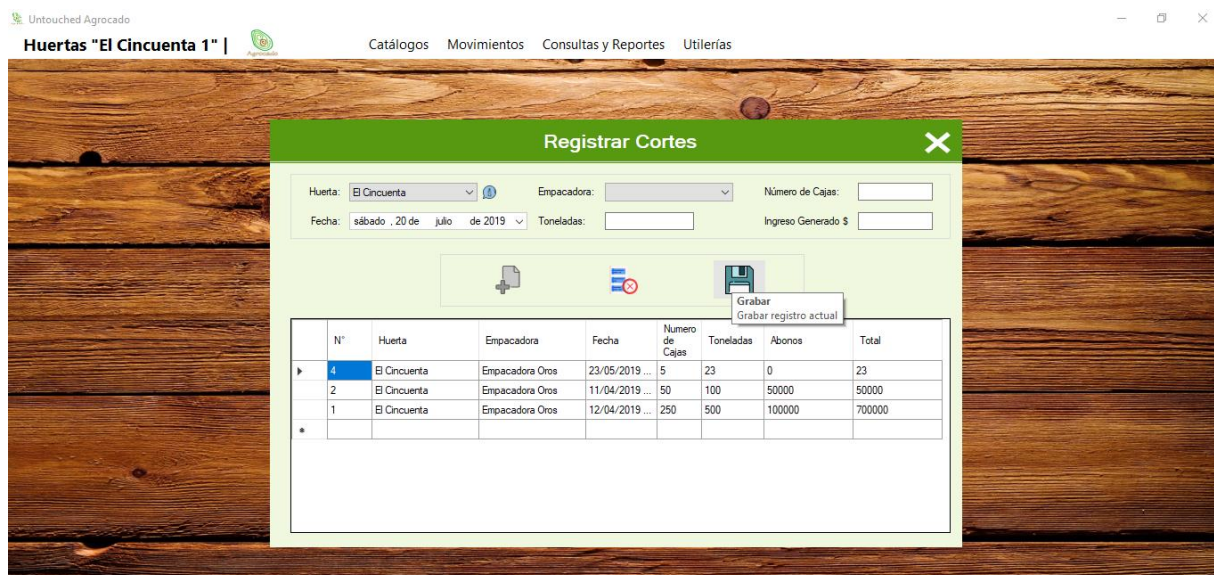


Figura 44. Registrar cortes de escritorio

En la Figura 45 se muestra la pantalla registrar actividades de la aplicación de escritorio en la cual se puede observar la entrada de datos, contiene una lista donde se pueden agregar productos utilizados en la actividad a registrar.

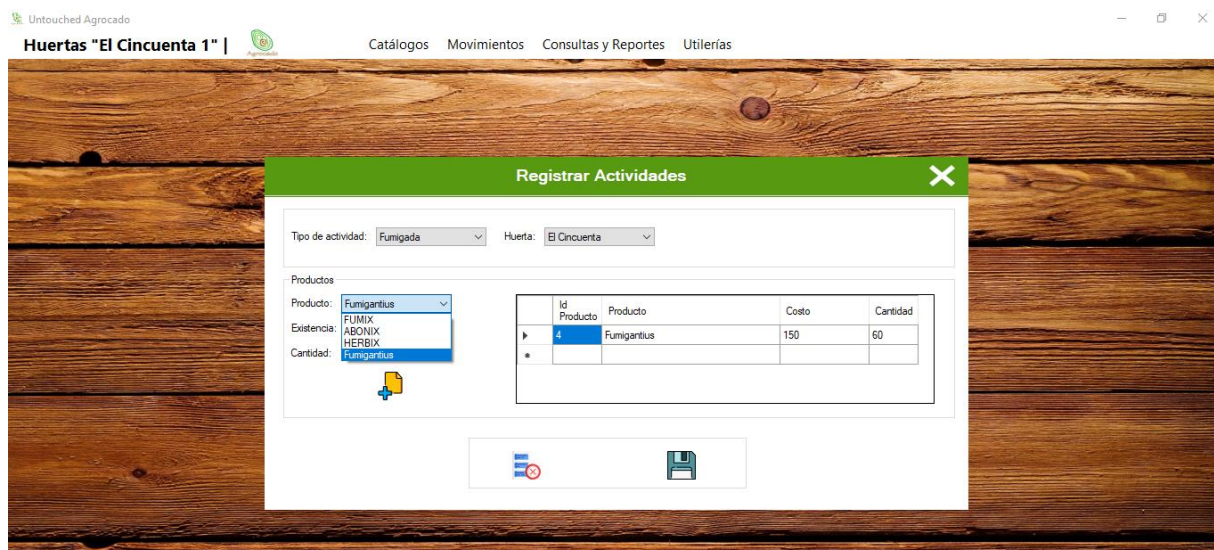


Figura 45. Registrar actividades de escritorio

#### 6.1.4 Consultas

En la Figura 46 se muestra la consulta cobros de la aplicación de escritorio, se puede seleccionar la forma de consultar, por fecha, empacadora y por huerta, en este caso selecciona la empacadora y muestra todos los cobros realizados a esa empacadora.

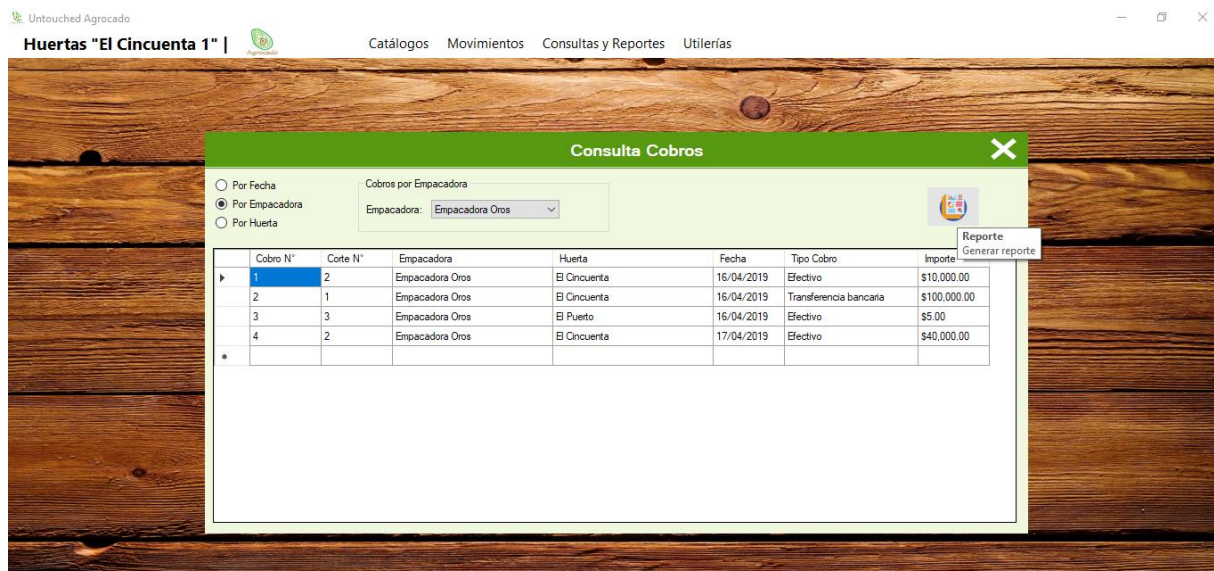


Figura 46. Consulta cobros de escritorio

En la Figura 47 se muestra la consulta cortes de la aplicación de escritorio y se puede escoger la forma de consultar los datos, por fecha, empacadora, huerta o por saldo, en este caso se realizó una consulta de los cortes pagados.

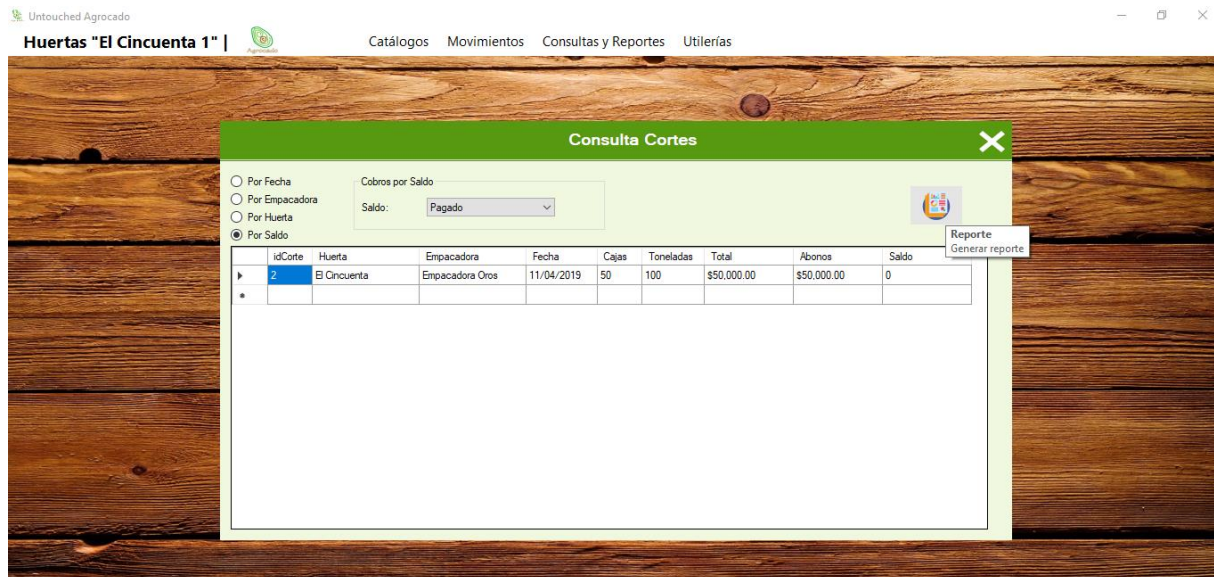


Figura 47. Consulta cortes de escritorio

## 6.2 Página Web

### 6.2.1 Menú

En la Figura 48 se muestra el menú consultas de la página Web.

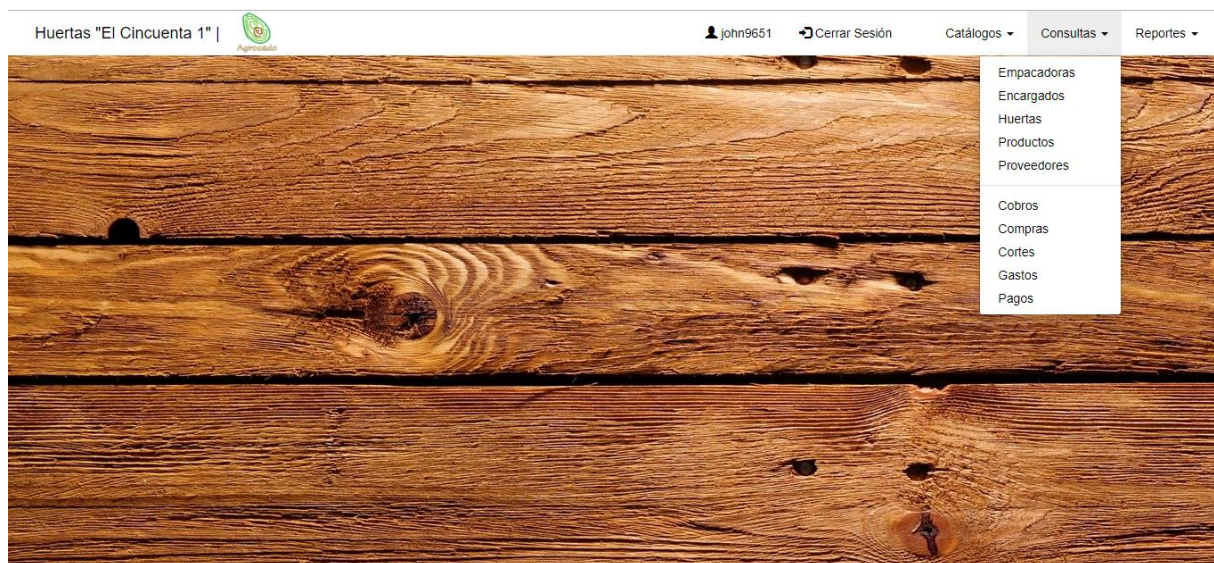


Figura 48. Menú consultas

En la Figura 49 se muestra el menú reportes de la página Web.

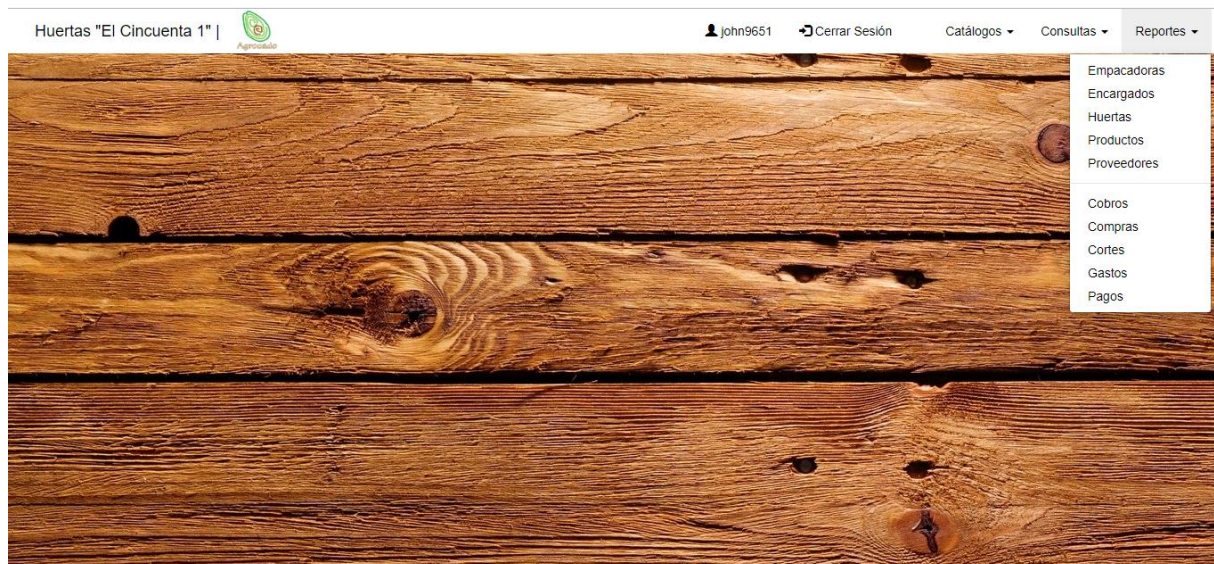


Figura 49. Menú reportes

## 6.2.2 Reportes

En la Figura 50 se muestra el reporte de las empacadoras en la aplicación Web. El informe presenta un listado de empacadoras con saldo pendiente.

ReporteClientes.php 1/1

Huertas "El Cincuenta 1" | Huertas "El Cincuenta 1" | Reporte Empacadoras | Fecha: 20-07-19

ID	Nombre	RFC	Telefono	Domicilio	Colonia	CP	Ciudad	E-Mail	Saldo
1	Empacadora Oros	EMPORO1234567	3411269659	San Antonio # 43	Centro	49000	Ario de Rosales	empacadoraoros@hotmail.com	\$ 0.00
2	Empacadora Avok2	AVOK223145634	3411269659	Carretera Ario-Tecario sn	S/C	49086	Ario de Rosales	avok2@hotmail.com	\$ 0.00
3	Empacadora Tinoco	AVOTIN3456789	4433001577	Carretera Ario-Tecario sn	S/C	00000	Pablo Culin	tinocoavocado@hotmail.com	\$ 0.00
Total									\$ 0.00

Figura 50. Reporte empacadoras

En la Figura 51 se muestra el reporte cortes de la aplicación de escritorio. Se puede apreciar el formato de los reportes y en este caso, datos de los cortes registrados.

ID	Huerta	Empacadora	Fecha	Cajas	Toneladas	Abonos	Total	Saldo
1	El Cincuenta	Empacadora Oros	2019-04-12	250	500	\$ 100,000.00	\$ 700,000.00	\$ 600,000.00
2	El Cincuenta	Empacadora Oros	2019-04-11	50	100	\$ 50,000.00	\$ 50,000.00	\$ 0.00
3	El Puerto	Empacadora Oros	2019-04-16	55	555	\$ 5.00	\$ 5,555.00	\$ 5,550.00
4	El Cincuenta	Empacadora Oros	2019-05-23	5	23	\$ 0.00	\$ 23.00	\$ 23.00
5	El Puerto	Empacadora Oros	2019-05-23	6	65	\$ 0.00	\$ 65.00	\$ 65.00
Total						\$ 150,005.00	\$ 755,643.00	\$ 605,638.00

Figura 51. Reporte cortes

## 6.3 Aplicación móvil

### 6.3.1 Menú

En la Figura 52 se muestra el menú catálogos de la aplicación móvil.



Figura 52. Menú catálogos móvil

### 6.3.2 Consultas

En la Figura 53 se muestra la consulta actividades de la aplicación móvil donde se observan datos de actividades registradas anteriormente en la base de datos.



*Figura 53. Consulta actividades móvil*

En la Figura 54 se muestra la consulta empacadoras de la aplicación de móvil donde se pueden observar datos de una empacadora registrada anteriormente.



*Figura 54. Consulta empacadoras móvil*

En la Figura 55 se muestra la consulta pagos de la aplicación móvil donde aparecen datos de un pago ya efectuado.

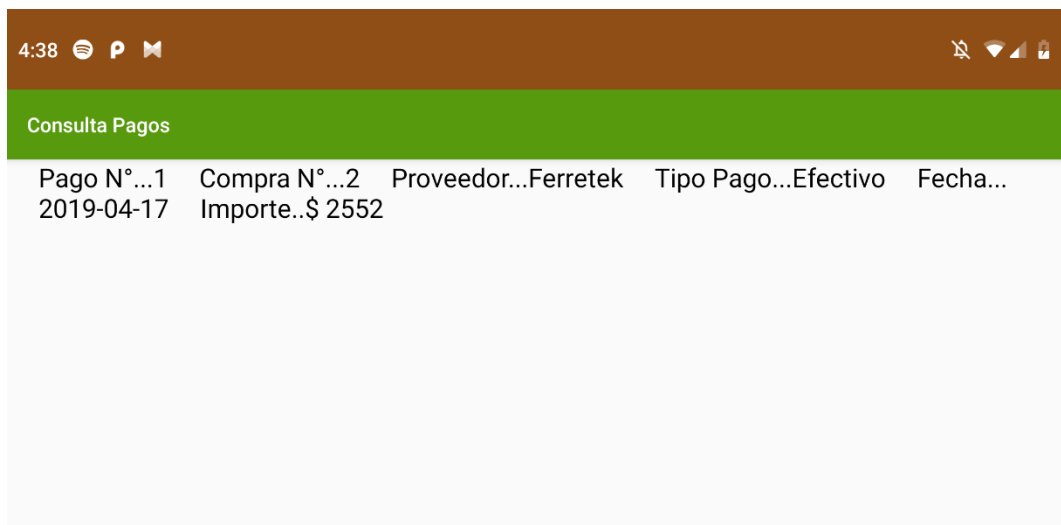


Figura 55. Consulta pagos móvil

### 6.3.3 Selector de fecha

En la Figura 56 se muestra la pantalla de selección de fecha que delimita las consultas en la aplicación móvil. Se selecciona la fecha inicial que indica desde qué fecha se buscarán los registros y la fecha final que indica hasta qué registros mostrar en las consultas.



Figura 56. Selección de fecha



## 7. DISCUSIÓN

El problema inicial que se presentó fue cuando la empresa expuso su preocupación por tener que pagar un servidor, justificando que aún no se tenía el capital suficiente para pagarlo, así que se tuvieron que buscar servidores en la Web que fueran gratuitos y tuvieran las características necesarias para continuar con el trabajo, después de una gran búsqueda se encontró un servidor que soportaba SQLServer pero no soportaba conexiones remotas lo que es fundamental para la plataforma, así que se tuvo que usar MySQL lo que causó un atraso al tener que modificar las conexiones de toda la aplicación de escritorio y de la página Web.

Con el cambio de SGBD se tuvo que descargar e instalar una extensión de MySQL para Visual Studio llamada MySQL.Data.dll, convertir la base de datos tipo SQLServer a tipo MySQL (aunque son similares, hay características que hacen que no se pueda usar la misma base de datos en ambos gestores). Posteriormente se investigó el uso de MySQL en Visual Studio y con páginas Web para cambiar las conexiones que se tenían en cada formulario de la aplicación de escritorio y en la página Web.

Otro problema fue generar reportes a través del llenado de un dataset con consultas en lugar de procedimientos almacenados en la aplicación de escritorio.

Al realizar las tareas de implementación, se observó que el personal se resistía al cambio, es decir preferían continuar con el procedimiento manual a pesar de haber obtenido capacitación al respecto, no veían los beneficios que brinda la plataforma digital. El resultado positivo se obtuvo al capacitar nuevamente al personal por un periodo de tiempo más largo, con casos reales, utilizando cada uno de los escenarios de la plataforma y en constante comunicación, llegando a la conclusión que el personal se resistía al cambio por miedo de utilizar algo diferente a lo que estaban acostumbrados.

La plataforma tuvo un impacto positivo, pues permite el seguimiento a todos los procesos que realiza la huerta. Además, genera consultas a través de aplicaciones móviles y Web mientras que los datos son actualizados en la aplicación de escritorio. Lo anterior, ayuda a obtener información actualizada para la toma oportuna de decisiones.

## 8. CONCLUSIONES

El objetivo principal de este proyecto fue desarrollar una plataforma que permitiera llevar la gestión de huertas aguacateras en sus operaciones diarias con empacadoras, proveedores y otras actividades propias de las huertas.

La aportación principal de este informe consiste en el diseño, desarrollo e implementación de una plataforma digital de fácil acceso mediante una aplicación de escritorio, una página Web y una aplicación móvil. Se decidió trabajar en este proyecto para contribuir con las empresas productoras de aguacate a gestionar su información, además, de cuidar del medio ambiente reduciendo de forma considerable el uso de papel, tinta y otros productos dentro de las huertas.

La participación en este proyecto deja grandes conocimientos sobre las diferentes tecnologías y lenguajes de programación que se pueden implementar para resolver problemas cotidianos. Lo anterior genera una satisfacción personal al plasmar lo aprendido en el Instituto Tecnológico, además que motiva a investigar hasta encontrar la solución a cualquier inconveniente que se presente.

Durante el desarrollo del proyecto se obtuvieron competencias, conocimiento, experiencia y finalmente interés en la carrera reconociendo la cantidad de posibilidades para resolver problemas con creatividad, investigación y trabajo.

Es necesario alimentar correctamente la base de datos para que se genere información correcta, no omitir datos es la clave principal para tener información confiable, por eso es importante la capacitación, para que el personal haga un buen uso de la plataforma, estar en constante comunicación aclarando cualquier duda que pueda surgir hasta comprender completamente la funcionalidad de esta.

## 9. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES

- Arias, Á. (2016). *Programación y Bases de Datos*. 2a ed. IT Campus Academy.
- Bender, C., Deco, C., Gonzáles Sanabria, J., Hallo, M. y Ponce Gallegos, J. (2014). *Tópicos Avanzados de Bases de Datos*. 1er ed. LATIn.
- Ceballos Sierra, F. (2010). *Microsoft Visual Basic .NET. Lenguaje y aplicaciones*. 3ra ed. Madrid, España: RA-MA.
- CEDRSSA. (2017). Caso de Exportación: El Aguacate. Ciudad de México. Palacio Legislativo de San Lázaro. Recuperado de <http://www.cedrssa.gob.mx/files/b/13/54Exportación%20aguacate.pdf>
- Cobo Yera, A. (2007). *Manual avanzado de HTML*. 1er ed. Madrid: Visión Libros.
- Del Sole, A. (2017). *Visual Studio*, Morrisville, USA: Syncfusion.
- Docs.microsoft.com. (2019). *Información general sobre Visual Studio*. [en línea] Disponible en: <https://docs.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2019> [Recuperado el 13 jul. 2019].
- Durán, F., Gutiérrez, F. y Pimentel, E. (2007). *Programación orientada a objetos con Java*. 1er ed. Madrid: Thomsom.
- Eslava Muñoz, F. (2013). *El nuevo PHP paso a paso*. 1ra ed. España: Bubok Publishing.
- Flórez Fernández, H. (2012). *Programación Orientada a Objetos Usando Java*. 1ra ed. Bogotá, Colombia: Ecoe Ediciones.
- Fossati, M. (2017). *Todo sobre Visual Basic*. Recuperado el 13 de Julio del 2019 de <https://books.google.com.mx/books?id=GdwxDwAAQBAJ&printsec=frontcover#v=onepage&q&f=false>.
- Gómez Fuentes, M. (2013). *Notas del Curso: Bases de Datos*. 1ra ed. Ciudad de México: Universidad Autónoma Metropolitana.
- Hohensee, B. (2014). *Introducción a Android Studio*. 1ra ed. Babelcube Inc.
- Minera, F. (2008). *Curso de Programación PHP*. 1er ed. Buenos Aires: Gradi.
- Morán, L. (2015). *SQL Server para Programadores: Aprendiendo a Optimizar Procesos*. 1ra ed. Albaterra, Alicante, España: SolidQ TM Press.
- Pavón Puertas, J. y Llanera Borges, E. (2015). *Creación de un sitio Web con PHP y MySQL*. 5a ed. Madrid, España: RA-MA.
- Peralta, L. (2016). *Apps para el campo mexicano*. [en línea] Milenio.com. Disponible en: <https://www.milenio.com/negocios/apps-para-el-campo-mexicano> [Recuperado el 16 de julio de 2019].

- Robledo, D. (2015). *Desarrollo de aplicaciones para Android I*. 1ra ed. España: Ministerio de Educación, Cultura y Deporte. España.
- SAGARPA.(2017). *Planeación Agrícola Nacional*. [en línea] Disponible en: <https://www.gob.mx/cms/uploads/attachment/file/257067/Potencial-Aguacate.pdf> [Recuperado el 16 jul. 2019].
- Sánchez Maza, M. (2001). *Javascript*. 1ra ed. España: Innovación y Cualificación S.L.
- Santiago Zaragoza, M., Zapata Perusquia, R. y Martínez Martín, G. (2015). *Adopción y uso de las TICS en el sector productivo agrícola como sistemas promotores de la competitividad*. [en línea] Milenio.com. Disponible en: <https://www.milenio.com/opinion/vari- os- autores/ universidad- tecnologica- del- valle- del- mezquital/ adopcion- tics- sector- productivo- agricola- sistemas- promotores- competitividad> [Recuperado el 16 jul. 2019].
- Silberschatz, A., Korth, H. y Sudarshan, S. (2006). *Fundamentos de diseño de bases de datos*. 5a ed. Madrid: McGraw-Hill.
- Schulz, R. (2009). *Diseño Web con CSS*. 1ra ed. Barcelona: Marcombo.
- Thibaud, C. (2006). *MySQL 5 Instalación, Implementación, Administración y Programación*. 1ra ed. Barcelona: Ediciones ENI.

## 10. ANEXOS

### 10.1 Aplicación de escritorio

#### 10.1.1 Formulario registrar cortes

```
Imports MySql.Data.MySqlClient
Public Class Cortes
    Public idH(contar() + 1) As Integer
    Public idE(contar2() + 1) As Integer
    Public Saldo As Integer = 0
    Dim nuevo As Integer = 0

    Private Sub Cortes_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        If My.Computer.Network.IsAvailable Then
            Me.ToolTip1.SetToolTip(PictureBox2, "Puede consultar los cortes seleccionando la huerta")
            dgCortes.Rows.Clear()
            Try
                Dim conexion As New MySqlConnectionStringBuilder()
                conexion.Server = "remotemysql.com"
                conexion.UserID = "ib9ltHJMil"
                conexion.Password = "B1kAohe3uF"
                conexion.Database = "ib9ltHJMil"
                conexion.Port = "3306"
                Dim con As New MySqlConnection(conexion.ToString)
                con.Open()
                Dim command As New MySqlCommand("SELECT * FROM `Huertas`", con)
                Dim lector As MySqlDataReader
                lector = command.ExecuteReader
                Dim num = 0
                While lector.Read
                    cbHuertasCortes.Items.Add(lector(2))
                    idH(num) = lector(0)
                    num = num + 1
                End While
                con.Close()
                con.Open()
                num = 0
                Dim command2 As New MySqlCommand("SELECT * FROM `Empacadoras`", con)
                lector = command2.ExecuteReader
```

```

        While lector.Read
            cbEmpacadorasCortes.Items.Add(lector(1))
            idE(num) = lector(0)
            num = num + 1
        End While
        lector.Close()
        con.Close()
    Catch ex As Exception
        MessageBox.Show("Error: " & Err.Number & " " &
Err.Description)
    End Try
    Me.cbEmpacadorasCortes.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
    Me.cbHuertasCortes.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList
    Else
        MsgBox("Compruebe su conexión a internet")
        Dispose()
    End If
End Sub
Private Sub btnNuevo_Click(sender As Object, e As EventArgs) Handles
btnNuevo.Click
    txtCajasCortes.Enabled = True
    txtToneladasCrtes.Enabled = True
    txtTotal.Enabled = True
    btnGrabar.Enabled = True
    dtpCortes.Enabled = True
    cbEmpacadorasCortes.Enabled = True
    btnCancelar.Enabled = True
    btnNuevo.Enabled = False
    nuevo = 1
End Sub
Private Function contar()
    Dim x As Integer
    Try
        Dim conexion As New MySqlConnectionStringBuilder()
        conexion.Server = "remotemysql.com"
        conexion.UserID = "ib9ltHJMil"
        conexion.Password = "B1kAohe3uF"
        conexion.Database = "ib9ltHJMil"
        conexion.Port = "3306"
        Dim con As New MySqlConnection(conexion.ToString)
        con.Open()
        Dim command As New MySqlCommand("Select count(*) from
`Huertas`", con)
        Dim lector As MySqlDataReader

```

```

        lector = command.ExecuteReader
        While lector.Read
            x = lector(0) + 1
        End While
        con.Close()
    Catch ex As Exception
        MessageBox.Show("Error: " & Err.Number & " " &
Err.Description)
        Return x
    End Function
Private Function contar2()
    Dim x As Integer
    Try
        Dim conexion As New MySqlConnectionStringBuilder()
        conexion.Server = "remotemysql.com"
        conexion.UserID = "ib9ltHJMil"
        conexion.Password = "B1kAohe3uF"
        conexion.Database = "ib9ltHJMil"
        conexion.Port = "3306"
        Dim con As New MySqlConnection(conexion.ToString)
        con.Open()
        Dim command As New MySqlCommand("Select count(*) from
`Empacadoras`", con)
        Dim lector As MySqlDataReader
        lector = command.ExecuteReader
        While lector.Read
            x = lector(0) + 1
        End While
        con.Close()
    Catch ex As Exception
        MessageBox.Show("Error: " & Err.Number & " " &
Err.Description)
    End Try
    Return x
End Function
Private Sub btnGrabar_Click(sender As Object, e As EventArgs) Handles
btnGrabar.Click
    Dim transaccion As MySqlTransaction
    If (txtCajasCortes.Text = "" Or txtToneladasCrtes.Text = ""
Or txtTotal.Text = "" Or cbEmpacadorasCortes.Text = "" Or
cbHuertasCortes.Text = "") Then
        MessageBox.Show("Faltan Datos")
    Else
        Dim hu As Integer =
idH(Val(cbHuertasCortes.SelectedIndex))

```

```

        Dim emp As Integer =
idE(Val(cbEmpacadorasCortes.SelectedIndex))
        Dim cajas As Integer = CType(txtCajasCortes.Text,
Integer)
        Dim fecha As String
fecha = Me.dtpCortes.Value.Date
fecha = Format(CDate(fecha), "yyyy/MM/dd")
        Dim toneladas As Integer = CType(txtToneladasCrtes.Text,
Integer)
        Dim abonos As Integer = 0
        Dim total = Val(txtTotal.Text)
        Dim conexion As New MySqlConnectionStringBuilder()
conexion.Server = "remotemysql.com"
conexion.UserID = "ib91tHJMil"
conexion.Password = "B1kAohe3uF"
conexion.Database = "ib91tHJMil"
conexion.Port = "3306"
        Dim con As New MySqlConnection(conexion.ToString)
con.Open()
transaccion = con.BeginTransaction
        Try
            Dim command As New MySqlCommand("INSERT INTO
`Cortes`(`idHuerta`, `idEmpacadora`, `fechaCorte`, `cajasCorte`,
`toneladasCorte`, `abonosCorte`, `totalCorte`) VALUES (" & hu & "," &
emp & ",'" & fecha & "'," & cajas & "," & toneladas & "," & abonos &
"," & total & ")", con)
            command.Transaction = transaccion
command.ExecuteNonQuery()
            Dim actualiza As New MySqlCommand("UPDATE
`Empacadoras` SET `saldoEmpacadora` = ?sal where
`idEmpacadora`=?emp", con)
            actualiza.Parameters.AddWithValue("?sal", Saldo +
Val(txtTotal.Text))
            actualiza.Parameters.AddWithValue("?emp", emp)
            actualiza.Transaction = transaccion
            actualiza.ExecuteNonQuery()
            If MsgBox("¿Desea Continuar?", MsgBoxStyle.YesNo,
"Guardar datos") = MsgBoxResult.Yes Then
                transaccion.Commit()
            Else
                transaccion.Rollback()
                MsgBox("No se Guardaron los datos")
            End If
        Catch ex As Exception
            MessageBox.Show("Error: " & Err.Number & " " &
Err.Description)

```



```

        transaccion.Rollback()
    End Try
    con.Close()
    llenarRejilla(hu)
    txtCajasCortes.Text = ""
    txtToneladasCrtes.Clear()
    txtTotal.Clear()
    cbEmpacadorasCortes.ResetText()
    dtpCortes.Enabled = False
    txtCajasCortes.Enabled = False
    txtToneladasCrtes.Enabled = False
    txtTotal.Enabled = False
    cbEmpacadorasCortes.Enabled = False
    btnCancelar.Enabled = False
    btnGrabar.Enabled = False
    btnNuevo.Enabled = True
    nuevo = 0
End If
End Sub
Private Sub llenarRejilla(ByVal x As Integer)
    dgCortes.Rows.Clear()
    Dim cont As Integer = 0
    Try
        Dim conexion As New MySqlConnectionStringBuilder()
        conexion.Server = "remotemysql.com"
        conexion.UserID = "ib9ltHJMil"
        conexion.Password = "B1kAohe3uF"
        conexion.Database = "ib9ltHJMil"
        conexion.Port = "3306"
        Dim con As New MySqlConnection(conexion.ToString)
        con.Open()
        Dim command As New MySqlCommand("SELECT `idCorte`,
`Huertas`.`nombreHuerta`, `Empacadoras`.`nombreEmpacadora`,
`fechaCorte`, `cajasCorte`, `toneladasCorte`, `totalCorte`,
`abonosCorte` FROM `Cortes` inner join `Huertas` on
`Huertas`.`idHuerta`=`Cortes`.`idHuerta` inner join `Empacadoras` on
`Empacadoras`.`idEmpacadora`=`Cortes`.`idEmpacadora` where
`Cortes`.`idHuerta`=" & x & " order by `idCorte` DESC", con)
        Dim lector As MySqlDataReader
        lector = command.ExecuteReader
        While lector.Read
            dgCortes.Rows.Add(lector(0), lector(1), lector(2),
lector(3), lector(4), lector(5), lector(7), lector(6))
            cont = cont + 1
        End While
        If (cont = 0 And nuevo = 0) Then

```

```

        MessageBox.Show("Aún no Hay Registros")
    End If
    con.Close()
    Catch ex As Exception
        MessageBox.Show("Error: " & Err.Number & " " &
Err.Description)
    End Try
End Sub
Private Sub obtenerSaldo(ByVal x As Integer)
    Try
        Dim conexion As New MySqlConnectionStringBuilder()
        conexion.Server = "remotemysql.com"
        conexion.UserID = "ib91tHJMil"
        conexion.Password = "B1kAohe3uF"
        conexion.Database = "ib91tHJMil"
        conexion.Port = "3306"
        Dim con As New MySqlConnection(conexion.ToString)
        con.Open()
        Dim command As New MySqlCommand("Select `saldoEmpacadora`
FROM `Empacadoras` WHERE `idEmpacadora`=" & x, con)
        Dim lector As MySqlDataReader
        lector = command.ExecuteReader
        While lector.Read
            Saldo = lector(0)
        End While
        con.Close()
    Catch ex As Exception
        MessageBox.Show("Error: " & Err.Number & " " &
Err.Description)
    End Try
End Sub
Private Sub cbHuertasCortes_TextChanged(sender As Object, e As
EventArgs) Handles cbHuertasCortes.TextChanged
    If (cbHuertasCortes.SelectedIndex > -1) Then
        Dim x As Integer = Val(cbHuertasCortes.SelectedIndex)
        Dim y As Integer = Val(idH(x))
        llenarRejilla(y)
    End If
End Sub
Private Sub btnCancelar_Click(sender As Object, e As EventArgs)
Handles btnCancelar.Click
    txtCajasCortes.Enabled = False
    txtToneladasCrtes.Enabled = False
    txtTotal.Enabled = False
    txtCajasCortes.Text = ""
    txtToneladasCrtes.Text = ""

```

```

        txtTotal.Text = ""
        btnGrabar.Enabled = False
        btnNuevo.Enabled = True
        btnCancelar.Enabled = False
        cbEmpacadorasCortes.Enabled = False
        dtpCortes.Enabled = False
        nuevo = 0
    End Sub
    Private Sub cbEmpacadorasCortes_TextChanged(sender As Object, e
As EventArgs) Handles cbEmpacadorasCortes.TextChanged
        If (cbEmpacadorasCortes.SelectedIndex > -1) Then
            Dim x As Integer = Val(cbEmpacadorasCortes.SelectedIndex)
            obtenerSaldo(idE(x))
        End If
    End Sub
    Private Sub Button1_Click(sender As Object, e As EventArgs)
Handles Button1.Click
        Dispose()
    End Sub
    Private Sub txtCajasCortes_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles txtCajasCortes.KeyPress
        If Char.IsDigit(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsControl(e.KeyChar) Then
            e.Handled = False
        Else
            e.Handled = True
        End If
    End Sub
    Private Sub txtToneladasCortes_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles txtToneladasCortes.KeyPress
        If Char.IsDigit(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsControl(e.KeyChar) Then
            e.Handled = False
        Else
            e.Handled = True
        End If
    End Sub
    Private Sub txtTotal_KeyPress(sender As Object, e As
KeyPressEventArgs) Handles txtTotal.KeyPress
        If Char.IsDigit(e.KeyChar) Then
            e.Handled = False
        ElseIf Char.IsControl(e.KeyChar) Then
            e.Handled = False
        Else

```

```
        e.Handled = True
    End If
End Sub
Private Sub Button2_Click(sender As Object, e As EventArgs)
    Huertas.Show()
End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs)
    Empacadoras.Show()
End Sub
End Class
```

## 10.2 Página Web

### 10.2.1 Formulario consulta empacadoras

```
<?php
  include 'menu.php';
?>
<html>
<head>
  <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<link rel="stylesheet" href="main.css">
  <title>Consulta Empacadoras</title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.m
in.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js
"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min
.js"></script>
</head>
<body>
  <div class="container">
  <div class="form__top">
    <h2>Consulta <span>Empacadoras</span></h2>
<style>
table {
  width: 100%;
  border-collapse: collapse;

}

table, td, th {
  border: 1px solid black;
  padding: 5px;
}
```

```
.pagination>.active>a, .pagination>.active>a:focus,
.pagination>.active>a:hover, .pagination>.active>span,
.pagination>.active>span:focus, .pagination>.active>span:hover {
    z-index: 3;
    color: #fff;
    cursor: default;
    background-color: #5a1605;
    border-color: #ab1815;
}
```

```
.pagination>li>a, .pagination>li>span {
    position: relative;
    float: left;
    padding: 6px 12px;
    margin-left: -1px;
    line-height: 1.42857143;
    color: #13690275;
    text-decoration: none;
    background-color: #fff;
    border: 1px solid #ddd;
}
```

```
a {
    color: #02800a;
    text-decoration: none;
}
```

```
th {text-align: left;}
```

```
.table-striped>tbody>tr:nth-child(odd)>td,
    .table-striped>tbody>tr:nth-child(odd)>th {
    background-color: #fff;
}
.table-striped>tbody>tr:nth-child(even)>td,
.table-striped>tbody>tr:nth-child(even)>th {
    background-color: #6bb5d75;
}
.table-striped>thead>tr>th {
    background-color: #5a1605;
font-style: normal;
color: white;
}
```

```

</style>
</head>
<body>

<?php
include 'config.php';
$num_total_rows=0;
$R =mysqli_query($connexion,'SELECT COUNT(*) as total_products FROM
`Empacadoras`');
while($row = mysqli_fetch_array($R)) {
    $num_total_rows = $row['total_products'];
}
if ($num_total_rows > 0) {
    $page = false;

    if (isset($_GET["page"])) {
        $page = $_GET["page"];
    }

    if (!$page) {
        $start = 0;
        $page = 1;
    } else {
        $start = ($page - 1) * NUM_ITEMS_BY_PAGE;
    }

    $total_pages = ceil($num_total_rows / NUM_ITEMS_BY_PAGE);
    $R =mysqli_query($connexion,"SELECT `idEmpacadora`,
`nombreEmpacadora`, `RFCEmpacadora`, `telEmpacadora`,
`domicilioEmpacadora`, `coloniaEmpacadora`, `cpEmpacadora`,
`ciudadEmpacadora`, `eMailEmpacadora`, `saldoEmpacadora` FROM
`Empacadoras` LIMIT $start, ".NUM_ITEMS_BY_PAGE);

    echo "<table style='font-size:12px' class='table table-
striped'>
    <thead>
<tr>
<th>N°</th>
<th>Nombre</th>
<th>RFC</th>
<th>Teléfono</th>
<th>Domicilio</th>

```

```

<th>Colonia</th>
<th>Ciudad</th>
<th>CP</th>
<th>E-Mail</th>
<th>Saldo</th>
</tr>
</thead>";
while($row = mysqli_fetch_array($R)) {
    echo "<tr>";
    echo "<td>" . utf8_decode($row['idEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['nombreEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['RFCEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['telEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['domicilioEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['coloniaEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['ciudadEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['cpEmpacadora']) . "</td>";
    echo "<td>" . utf8_decode($row['eMailEmpacadora']) . "</td>";
    echo "<td>" . '$ ' . (number_format($row['saldoEmpacadora'], 2, ".",
    ", ")) . "</td>";
    echo "</tr>";
}
echo"</table>";

echo '<nav>';
echo '<ul class="pagination">';

if ($total_pages > 1) {
    if ($page != 1) {
        echo '<li class="page-item"><a class="page-link"
href="ConsClientes2.php?page=' . ($page-1) . '"><span aria-
hidden="true">&laquo;</span></a></li>';
    }

    for ($i=1;$i<=$total_pages;$i++) {
        if ($page == $i) {
            echo '<li class="page-item active"><a class="page-
link" href="#">' . $page . '</a></li>';
        } else {
            echo '<li class="page-item"><a class="page-link"
href="ConsClientes2.php?page=' . $i . '">' . $i . '</a></li>';
        }
    }
}

```



```
    }

    if ($page != $total_pages) {
        echo '<li class="page-item"><a class="page-link"
href="ConsClientes2.php?page=' . ($page+1) . '"><span aria-
hidden="true">&raquo;</span></a></li>';
    }
}
echo '</ul>';
echo '</nav>';
}
?>

    </div>
</div>

</body>
</html>
```

## 10.3 Aplicación móvil

### 10.3.1 Activity seleccionar fecha (código java)

```
package com.example.john.agrocaDOMovil;

import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

import android.widget.Button;
import android.widget.DatePicker;

import android.app.DatePickerDialog;;
import java.util.Calendar;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

public class selecFecha extends AppCompatActivity implements
View.OnClickListener{
    private static final String CERO = "0";
    private static final String BARRA = "/";
    private static String consulta;
    //Calendario para obtener fecha & hora
    public final Calendar c = Calendar.getInstance();

    //Variables para obtener la fecha
    final int mes = c.get(Calendar.MONTH);
    final int dia = c.get(Calendar.DAY_OF_MONTH);
    final int anio = c.get(Calendar.YEAR);
    Button btnEnviar;
    EditText etFechaD;
    ImageButton ibObtenerFechaD;
    EditText etFechaH;
    ImageButton ibObtenerFechaH;
```

```

TextView txConsulta;
private String clase;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_selec_fecha);
    txConsulta = (TextView) findViewById(R.id.txtConsulta);
    consulta = getIntent().getStringExtra("consulta");
    txConsulta.setText(consulta);
    //Widget EditText donde se mostrara la fecha obtenida
    etFechaD = (EditText)
findViewById(R.id.et_mostrar_fecha_pickerD);
    //Widget ImageButton del cual usaremos el evento clic para
obtener la fecha
    ibObtenerFechaD = (ImageButton)
findViewById(R.id.ib_obtener_fechaD);
    //Evento setOnClickListener - clic
    ibObtenerFechaD.setOnClickListener(this);
    getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#579A0E")));

    //Widget EditText donde se mostrara la fecha obtenida
    etFechaH = (EditText)
findViewById(R.id.et_mostrar_fecha_pickerH);
    //Widget ImageButton del cual usaremos el evento clic para
obtener la fecha
    ibObtenerFechaH = (ImageButton)
findViewById(R.id.ib_obtener_fechaH);
    //Evento setOnClickListener - clic
    ibObtenerFechaH.setOnClickListener(this);
    btnEnviar = (Button) findViewById(R.id.btnEnviar);
    switch (consulta){
        case "Cobros":
            btnEnviar.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent i = new Intent(v.getContext(),
MainActivityCobros.class);

```

```

i.putExtra("desde",etFechaD.getText().toString());

i.putExtra("hasta",etFechaH.getText().toString());
        startActivityForResult(i, 0);
    }
});
break;
case "Compras":
    btnEnviar.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(v.getContext(),
MainActivityCompras.class);

i.putExtra("desde",etFechaD.getText().toString());

i.putExtra("hasta",etFechaH.getText().toString());
                startActivityForResult(i, 0);
            }
        });
break;
case "Cortes":
    btnEnviar.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(v.getContext(),
MainActivityCortes.class);

i.putExtra("desde",etFechaD.getText().toString());

i.putExtra("hasta",etFechaH.getText().toString());
                startActivityForResult(i, 0);
            }
        });
break;
case "Gastos":

```

```

        btnEnviar.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(v.getContext(),
MainActivityGastos.class);

i.putExtra("desde",etFechaD.getText().toString());

i.putExtra("hasta",etFechaH.getText().toString());
                startActivityForResult(i, 0);
            }
        });
        break;
        case "Pagos":
            btnEnviar.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent i = new Intent(v.getContext(),
MainActivityPagos.class);

i.putExtra("desde",etFechaD.getText().toString());

i.putExtra("hasta",etFechaH.getText().toString());
                    startActivityForResult(i, 0);
                }
            });
            break;
    }
}
@Override
public void onClick(View v) {
    switch (v.getId()){
        case R.id.ib_obtener_fechaD:
            obtenerFecha();
            break;
        case R.id.ib_obtener_fechaH:
            obtenerFechaHA();
            break;
    }
}

```

```

    }
}
private void obtenerFecha(){
    DatePickerDialog recogerFecha = new DatePickerDialog(this,
new DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int
month, int dayOfMonth) {
        //Esta variable lo que realiza es aumentar en uno
el mes ya que comienza desde 0 = enero
        final int mesActual = month + 1;
        //Formateo el día obtenido: antepone el 0 si son
menores de 10
        String diaFormateado = (dayOfMonth < 10)? CERO +
String.valueOf(dayOfMonth):String.valueOf(dayOfMonth);
        //Formateo el mes obtenido: antepone el 0 si son
menores de 10
        String mesFormateado = (mesActual < 10)? CERO +
String.valueOf(mesActual):String.valueOf(mesActual);
        //Muestro la fecha con el formato deseado
        etFechaD.setText(year + BARRA + mesFormateado +
BARRA + diaFormateado);

    }
    //Estos valores deben ir en ese orden, de lo contrario
no mostrara la fecha actual

    },anio, mes, dia);
    //Muestro el widget
    recogerFecha.show();

}
private void obtenerFechaHA(){
    DatePickerDialog recogerFecha = new DatePickerDialog(this,
new DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int
month, int dayOfMonth) {

```

```

        //Esta variable lo que realiza es aumentar en uno
        el mes ya que comienza desde 0 = enero
        final int mesActual = month + 1;
        //Formateo el día obtenido: antepone el 0 si son
        menores de 10
        String diaFormateado = (dayOfMonth < 10)? CERO +
        String.valueOf(dayOfMonth):String.valueOf(dayOfMonth);
        //Formateo el mes obtenido: antepone el 0 si son
        menores de 10
        String mesFormateado = (mesActual < 10)? CERO +
        String.valueOf(mesActual):String.valueOf(mesActual);
        //Muestro la fecha con el formato deseado
        etFechaH.setText(year + BARRA + mesFormateado +
        BARRA + diaFormateado);

    }

    },anio, mes, dia);
    //Muestro el widget
    recogerFecha.show();
}
}
}

```

### 10.3.2 Activity seleccionar fecha (código xml)

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.john.agrocadomovil.selecFecha">

```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="42dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:text="Consulta"
    android:textAlignment="center"
    android:textColor="@color/colorAccent"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/txtConsulta"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="2dp"
    android:text="TextView"
    android:textAlignment="center"
    android:textColor="@color/colorPrimary"
    android:textSize="30sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```



```
app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

```
<EditText
```

```
    android:id="@+id/et_mostrar_fecha_pickerD"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginEnd="8dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="4dp"  
    android:ems="10"  
    android:hint="00/00/0000"  
    android:inputType="date"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/txtConsulta" />
```

```
<EditText
```

```
    android:id="@+id/et_mostrar_fecha_pickerH"  
    android:layout_width="wrap_content"  
    android:layout_height="45dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="8dp"  
    android:ems="10"  
    android:hint="00/00/0000"  
    android:inputType="date"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.501"  
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/et_mostrar_fecha_pickerD"  
/>
```

```
<ImageButton  
    android:id="@+id/ib_obtener_fechaD"  
    android:layout_width="53dp"  
    android:layout_height="53dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginStart="8dp"  
    android:background="@drawable/calendario"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toEndOf="@+id/et_mostrar_fecha_pickerD"  
    app:layout_constraintTop_toBottomOf="@+id/txtConsulta"  
    tools:ignore="ContentDescription" />
```

```
<ImageButton  
    android:id="@+id/ib_obtener_fechaH"  
    android:layout_width="53dp"  
    android:layout_height="53dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="8dp"  
    android:background="@drawable/calendari"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toEndOf="@+id/et_mostrar_fecha_pickerH"  
    app:layout_constraintTop_toBottomOf="@+id/ib_obtener_fechaD"  
    tools:ignore="ContentDescription" />
```

```
<TextView  
    android:id="@+id/textView4"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"
android:layout_marginTop="16dp"
android:text="Desde:"
android:textColor="@color/colorAccent"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintEnd_toStartOf="@+id/et_mostrar_fecha_pickerD"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtConsulta" />
```

<TextView

```
android:id="@+id/textView5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginEnd="8dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:text="Hasta:"
android:textColor="@color/colorAccent"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintEnd_toStartOf="@+id/et_mostrar_fecha_pickerH"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView4" />
```

<Button

```
android:id="@+id/btnEnviar"
```

```
    android:layout_width="64dp"
    android:layout_height="62dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="2dp"
    android:background="@drawable/seo"
    android:textColor="@android:color/background_light"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/et_mostrar_fecha_pickerH"
/>

</android.support.constraint.ConstraintLayout>
```