

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

TRABAJO DE DIPLOMA

Título: Sistema de gestión de la producción del sector privado en la Empresa UEB de Porcino de Villa Clara.

Autor: Yeralys Li Balmaseda.

Tutores: Dr.CT Martha Beatriz Boggiano Castillo.
Ing. Jandro Daniel Velázquez Hernández

Curso: 2020-2021.

"Año 63 de la Revolución."

Santa Clara, noviembre de 2021
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-14190



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del tutor

Firma del jefe del Dpto

PENSAMIENTO

“Las oportunidades grandes nacen de haber sabido aprovechar las pequeñas”

Bill Gates

AGRADECIMIENTOS.

A Dios todopoderoso por brindarme la oportunidad de obtener otro triunfo personal, y darme salud, sabiduría y entendimiento para lograr esta meta.

A mi querida madre Yanetza Balmaseda, por ser siempre incondicional y darme su apoyo en todo momento. Gracias por existir, y que Dios te bendiga siempre.

A mi padre Jose Felix Li quien con sus palabras de aliento no me dejó decaer para que siguiera adelante y siempre sea perseverante y cumpla con mis ideales.

A mi gran amigo Carmelo Miranda por brindarme su apoyo incondicional a cada momento.

A todos mis amigos y compañeros de la UCLV gracias por sus atenciones y colaboración prestada para llegar y conseguir mis objetivos trazados.

A todos mis profesores y personal de la UCLV por su apoyo y dedicación.

DEDICATORIA.

A mis padres por la confianza que siempre pusieron en mí y que en muchas ocasiones ha sido lo que me ha motivado a seguir adelante, por apoyarme siempre en todas mis decisiones por descabelladas que fuesen, por ser mi guía, mi refugio y mi ejemplo.

RESUMEN

Los dispositivos móviles en la actualidad son mucho más que un terminal con servicios de telefonía. Proporcionan multitud de servicios y funciones que pueden ser explotados de diversas maneras a través de las aplicaciones. Debido a su portabilidad y el gran número de funcionalidades que ofrecen se han convertido en uno de los avances tecnológicos más utilizados de este siglo. En el ámbito productivo de la UEB Porcina de Villa Clara, la aplicación móvil que se presenta responde a una demanda a causa de la necesidad del acceso ágil a la información de los convenios por parte de los productores privados. Se desarrolla una aplicación con Sistema Operativo Android, con el objetivo de facilitar el acceso y uso de la información referente a los convenios por parte de los productores, para gestionar su ejecución, y seguimiento por parte de ellos; dicho acceso se hará a través de un servicio web con arquitectura REST el cual permitirá la conexión con la base de datos de la plataforma de la UEB y la extracción de información de la misma.

ABSTRACT

Mobile devices today are much more than a terminal with telephone services. They provide a multitude of services and functions that can be exploited in various ways through applications. Due to their portability and the large number of functionalities they offer, they have become one of the most used technological advances of this century. In the productive area of the Villa Clara Porcino UEB, the mobile application that is presented responds to a demand due to the need for quick access to information on agreements by private producers. An application with Android Operating System is developed, in order to facilitate the access and use of the information regarding the agreements by the producers, to manage their execution, and follow-up by them; Said access will be made through a web service with REST architecture which will allow the connection with the database of the UEB platform and the extraction of information from it.

INDICE.

INTRODUCCIÓN.....	1
CAPITULO 1: MARCO TEÓRICO.....	6
1.1. Antecedentes sobre la gestión de la información para la producción de carne porcina.....	6
1.2. Sistema Operativo Android	9
1.2.1. Arquitectura de Android.	10
1.2.2. IDE de programación	11
1.3. Aplicación Móvil.	12
1.4. Tipos de aplicaciones móviles según su desarrollo.	13
1.4.1. Aplicaciones nativas.....	14
1.4.2. Aplicaciones Web	15
1.4.3. Aplicaciones Híbridas.....	15
1.4.4 Versiones de Android	16
1.5. Metodologías para el desarrollo del software.	18
1.5.1. Metodologías ágiles.	18
1.5.2. Selección de la metodología de desarrollo.	19
1.5.3. Ciclo de vida de Mobile-D	21
1.6. Diseño de aplicaciones móviles para el sistema operativo Android. ...	23
1.6.1. Principio de experiencia de usuario.	24
1.6.2. Interfaces nativas o personalizadas.	26
Conclusiones parciales.....	28
CAPÍTULO II. MODELO DEL NEGOCIO Y REQUISITOS DEL SISTEMA.....	29
2.1 Modelado de negocio.....	29
2.2 Requisitos del sistema.....	33
Conclusiones Parciales.	39
CAPÍTULO III. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.	40
3.1. Exploración.....	40
3.2. Inicialización.	41
3.3 Producción y Estabilización.....	46

3.4 Pruebas del sistema	63
Conclusiones parciales.....	73
CONCLUSIONES	74
REFERENCIAS BIBLIOGRÁFICAS.....	75
ANEXOS	78
Anexo 1: Modelo físico de la Base de Datos de la Empresa UEB de Porcino de Villa Clara	78
Anexo 2: Manual de Usuario.....	79

INTRODUCCIÓN.

Las tecnologías de la información y las comunicaciones han pasado a formar parte de la cotidianeidad del hombre actual, sobre todo los dispositivos móviles, los teléfonos inteligentes y las tabletas.

El dispositivo móvil, se puede definir, como un aparato de pequeño tamaño, el cual posee un sinnúmero de funciones, entre las cuales podemos mencionar, el procesamiento e intercambio de información, la conexión a alguna red, todo esto a través de una memoria interna y limitada (Luna, Vaca and Vásquez, 2017).

El dispositivo móvil, principalmente el teléfono inteligente y el table, se han convertido en artefactos cotidianos en la vida de las personas.

Según cifras mundiales del 2018, se vendieron más de 1969,6 millones de celulares, de los cuales el 95,6 % fueron teléfonos inteligentes o “smartphones” (ROLDAN and THOMPSON, 2018).

El viejo concepto del teléfono utilizado para establecer una comunicación oral o escrita ha evolucionado a un concepto de aparato inteligente, portátil, con acceso a Internet, que permite movilidad y que ha cambiado la forma en que se interactúa en el mundo utilizando aplicaciones móviles, las cuales establecen nuevos modelos para hacer negocios, para estudiar, para comprar y vender, para trabajar, para buscar información; en fin, el contar con un dispositivo móvil inteligente junto con las respectivas aplicaciones, permite a sus dueños disponer de una poderosa herramienta capaz de traer su mundo al alcance del bolsillo de una camisa o de un pantalón (Cuello and Vittone, 2013).

En Cuba al igual que el resto de los países del mundo se ha extendido el uso de las tecnologías de la información y las telecomunicaciones donde las generaciones más jóvenes, fundamentalmente en el ambiente universitario conviven con los smartphones en todo momento (Peñuela et al., 2014). A partir de esta problemática el ministerio de informática y comunicaciones tiene una política de informatización de la sociedad cubana reflejada en los Lineamientos resultantes del 7mo Congreso del Partido Comunista de Cuba (de Cuba, 2017) en su artículo 108 plantea:

“Avanzar gradualmente, según lo permitan las posibilidades económicas, en el proceso de informatización de la sociedad, el desarrollo de la infraestructura de telecomunicaciones y la industria de aplicaciones y servicios informáticos. Sustentar este avance en un sistema de ciber-seguridad que proteja nuestra soberanía tecnológica y asegure el enfrentamiento al uso ilegal de las tecnologías de la información y la comunicación. Instrumentar mecanismos de colaboración internacional en este campo”.

El proceso de informatización de la sociedad cubana busca utilizar ordenadamente y de forma masiva las Tecnologías de la Información y las Comunicaciones en la vida cotidiana, buscando satisfacer todas las necesidades de la sociedad, logrando cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos (Jover and Arriete, 2018). La necesidad creciente de informatización crece cada día en los sectores productivos.

La UEB Porcina de Villa Clara no queda atrás en este proceso de informatización.

En 1998 comienza el vínculo de esta entidad con el sector privado, con el objetivo de mejorar la producción de este sector y de obtener un mejoramiento de la raza de puercos para su mejor rendimiento en carne, grasa, reproducción y fortaleza genética.

El control de los datos de la producción y la contabilidad se realizaba de forma manual y a partir del 2001 se empezaron a aplicar los primeros softwares para la gestión del modelo VPO9 Visual Frontron, que permitía el control de los datos antes mencionados, En el 2006 a nivel nacional se produce una versión más integral denominada Regente 1(Controlaba y guarda la contabilidad de la empresa junto con la producción realizada).

En el 2011 sale la Regente 2 apoyado en la Resolución 11 del 2010, esta asumía lo anterior con mejora y también el control de los convenios, en todos los casos este procedimiento manual demoraba mucho y tenía que realizarlo un especialista, la empresa nacional ha venido mejorando estos softwares que en la actualidad (se utiliza) la Resolución 101 del 2018.

Como todas las tecnologías de la información y la comunicación, los dispositivos móviles se han instaurado en nuestras vidas. Hoy en día prácticamente todos llevamos en nuestro bolsillo un terminal móvil la gran mayoría con sistema operativo Android. En la UEB Porcina de Villa Clara se tratan de aprovechar al máximo estos recursos poniendo al alcance de los productores del sector privado diversas aplicaciones que faciliten, la consulta y acceso a la información disponible, referente a los convenios y otros aspectos relacionados con su producción.

De lo anterior expuesto se deriva el siguiente problema de investigación:

Cómo viabilizar la gestión de consulta de los productores sobre los convenios en la UEB Porcina de Villa Clara para superar las limitaciones actuales en cuanto al acceso a la información acerca de ejecución, seguimiento y control de dichos convenios, contando con la utilización de los dispositivos móviles.

Para contribuir a la solución del problema de investigación antes planteado, se formuló el siguiente objetivo general de investigación:

Implementar una Aplicación para Móviles que facilite a los productores privados la consulta de la información de sus convenios con la empresa porcina y mantenerse actualizados de la ejecución, seguimiento y control.

Este objetivo general fue desglosado en los objetivos específicos siguientes:

1. Identificar las necesidades de información de los convenios que deben cubrirse con una aplicación móvil para satisfacer las expectativas de los productores
2. Implementar una aplicación para móviles sobre los contenidos de los convenios por parte de los productores del sector privado, con el cual se pueden mantener actualizado su ejecución, seguimiento y control.
3. Realizar pruebas de software de caja negra y pruebas de compatibilidad de la aplicación.

Preguntas de Investigación:

¿Cuáles son las necesidades informativas de los productores porcinos que son indispensables para el control sistemático de los convenios?

¿Cuáles herramientas son las más adecuadas para desarrollar una aplicación en el sistema operativo Android para el acceso a las actividades de la información referente a los convenios por parte de los productores, con el cual se pueden mantener actualizado su ejecución, seguimiento y control con los productores del sector privado?

¿Qué aplicación móvil puede desarrollarse que cubra las necesidades actuales de los productores privados y de la UEB para el seguimiento y ejecución de sus convenios contando con las expectativas de los usuarios y clientes de la misma?

Hipótesis.

Si se utiliza una aplicación para dispositivos móviles de los usuarios de la Plataforma en la UEB Porcina de Villa Clara se obtendrá un mayor nivel de acceso y uso del control de la información referente a los convenios por parte de los productores del sector privado de forma oportuna.

Justificación de la Investigación.

El producto responde a una demanda de la UEB Porcina de Villa Clara ante la necesidad de acceso ágil a la información de los convenios por parte de los productores, con el cual se pueden mantener actualizado su ejecución, seguimiento y control con los productores del sector privado. La aplicación a desarrollar cuenta con una gran importancia, la misma soluciona el inconveniente derivado de los largos tiempos de espera para acceder a la información actualizada de los convenios en las oficinas de la empresa porcina y la imposibilidad de realizar seguimiento y control de los mismos en cualquier momento.

El valor práctico del trabajo está dado en que la aplicación para dispositivos móviles que se propone, dispone de medios para facilitar el acceso y uso del control de la información referente a los convenios por parte de los productores, con el cual se pueden gestionar su ejecución, seguimiento y control con los productores del sector privado.

Teniendo en cuenta lo anterior, se justifica la creación y generalización de una aplicación móvil, cómoda, sencilla, rápida y flexible que dé respuesta al objetivo

descrito y que permita a la vez una actualización periódica de los elementos que la componen.

Este trabajo está estructurado en tres capítulos:

En el Capítulo 1 se describe la problemática del negocio que da pie a esta investigación, así como se identifican los requisitos del sistema, se identifican las tecnologías actuales para desarrollar aplicaciones móviles, seleccionando los principales conceptos, métodos, tecnologías y aplicaciones existentes en la actualidad, relacionados con la posible solución del problema de investigación planteado. Se llega a conclusiones sobre los métodos, tecnologías y software a desarrollar.

En el Capítulo 2 se caracteriza el negocio, la gestión de la producción de los productores del sector privado en la empresa de Porcino de Villa Clara, exponiendo las actividades fundamentales que se desarrollan. Se da a conocer el modelo de casos de uso del negocio y los requisitos que debe cumplir el sistema.

El Capítulo 3 muestra la arquitectura de la solución y se documentan los principales artefactos utilizados siguiendo la metodología de desarrollo seleccionada, Mobile-D. Se describen las pruebas de software de caja negra realizadas y se refleja las pruebas de compatibilidad de la aplicación.

CAPITULO 1: MARCO TEÓRICO.

En este capítulo se exponen aspectos generales acerca la historia del proceso de informatización en la Empresa UEB de Porcino de Villa Clara, de la introducción de sistemas informáticos para la gestión y el control de la producción de carne porcina específicamente de los sistemas que permiten la gestión de los convenios con el sector privado.

También se caracteriza el sistema operativo Android y las tecnologías utilizadas en el diseño de aplicaciones móviles como herramientas para la creación de dichos sistemas. Se describen los principales conceptos, métodos, tecnologías y tipos de aplicaciones móviles existentes en la actualidad, para enmarcar la solución del problema de investigación planteado; se estudian las metodologías de desarrollo de software, métodos y tecnologías relacionados con las aplicaciones móviles. Además, se relacionan los principios de experiencia de usuario que deben ser considerados para el desarrollo de estas aplicaciones.

1.1. Antecedentes sobre la gestión de la información para la producción de carne porcina.

La UEB Porcina de Villa Clara ubicada desde su creación en el reparto Brisas del Capiro en el municipio de Santa Clara se desempeña en la gestión de un grupo de actividades e información relacionada con los productores asociados del sector privado o estatal que tienen producciones de carne de cerdo. El objetivo económico de la empresa desde sus inicios ha sido la producción de carne porcina. Su Misión: Mantener los altos niveles de bioseguridad, previniendo la proliferación de foco de PPC, Introducir y aplicar nuevos indicadores científico-técnicos a la producción, bioseguridad y la reducción de la contaminación ambiental. Elevar sosteniblemente los niveles de producción y perfeccionar los indicadores de calidad el sistema de control y la efectiva lucha contra la corrupción.

Consta en las memorias escritas de la empresa que en una primera etapa se dedica al trabajo con socios del sector estatal y más tarde se incorpora el sector privado. La introducción de la tecnología en la empresa comenzó a finales de la

década del 80 con la incorporación de una computadora 8086, a partir de este momento se introducen todos los datos, tomados en papel, a una pequeña base de datos. En 1993 se introducen más computadoras 80486, a partir de este momento empiezan a introducirse las primeras versiones de Office y se llevan todos los datos de contabilidad a un sistema informático llamado CONE; y junto con el MROT que trabajaban los medios de rotación y el control de almacenes y para el trabajo con nóminas se utilizó un programa que se llamaba SOPRE, en tanto no existía soporte informático para llevar la actividad de producción porcina.

En 1998 se introducen las primeras Pentium II, ya con un sistema operativo más fuerte que trabajaban con Windows 95 y operaban con el paquete Office completo, en este propio año, ante la necesidad de incrementar la producción porcina surge la actividad de convenio con particulares y esto sirvió como pilar para la creación de dos pequeños sistemas para llevar la contabilidad de esa actividad que se programaron en VisualFoxPro. Para llevar el control de la producción de carne convenido con particulares se pone en explotación el programa VP01 y para el control de los convenios el VP09 que este extendía del CONE, ambos desarrollados por informáticos de la empresa en aquellos momentos.

En el 2006 se implementa la primera versión del Regente hecha en lenguaje de programación Delphi con una base de datos en Access que unía la producción resultante de los convenios con las compras de carne, este software controlaba el plan por primera vez. Se confeccionaron varias versiones del software regente tratando de solucionar el problema que se presentaba en el proceso de producción, al no poder cumplir con lo convenido con el productor en cuanto a la entrega de alimentos, era necesario renegociar o realizar un reajuste en el convenio del productor algo muy complicado para este tipo de programa que estaba vigente y sus propias limitaciones no le permitía hacer este proceso de almacenamiento de la información. Para poder resolver esta problemática, cosa que complicaba en los municipios pues ellos traían los resultados obtenidos durante el mes de forma manual y aquí en la empresa se conformaba su base

de datos a través de la consolidadora que importaba el proceso, se las incorporaban a una memoria de los informáticos y regresaban a sus respectivas unidades con esta información más completa. En el 2011 sale la versión 2 del Regente desarrollado en el lenguaje de programación Delphi con una base de datos en SQL, versión que ya estaba preparada para la utilización en las redes, a través de servidores, que introducía el control de las carnes y el control de los piensos. A raíz de la introducción del control de los piensos en el año 2014-2015 se le realizaron adecuaciones al Regente 2 y se hizo necesario el desarrollo de una nueva versión 3 del Regente, que se logró desarrollar, pero no se pone en explotación por decisiones de instancias superiores.

En la actualidad con la versión 2 del Regente se gestionan los datos de los productores asociados (productores del sector privado) para la confección de las fichas de los productores, dichos datos son el nombre y apellidos, carne de identidad, el tipo de formas productivas en la que se encuentra cumpliendo labor y a qué Cooperativa de Producción Agropecuaria o Cooperativa de Créditos y Servicios está asociado. Estos datos captados de los asociados son editables y se pueden inhabilitar cuando el productor no haya cumplido el convenio o lo esté incumpliendo en el plazo acordado, no se eliminan porque queda reflejado en el histórico de la empresa.

Unido a los registros de los productores asociados se gestionan los convenios realizados por ellos y la empresa. En estos convenios quedan plasmados el municipio, año, número de identidad, la fecha inicial y la final, el tipo de convenio al que va a asociarse, la variante alimenticia que se le entrega al productor, así como las cantidades, en una segunda sección queda la cantidad de cabezas a retornar con su peso inicial y final y su precio inicial y final. Para la recopilación de información sobre la compra de carne, se registra el código de convenio, municipio, año, fecha actual en la que se va a realizar, el nombre del productor y el tipo de convenio, la cantidad de cabezas que se van a comprar ya sea tanto a precio oficial como a precio diferenciado.

1.2. Sistema Operativo Android

Existen diferentes sistemas operativos móviles, pero sin duda el más utilizado universalmente es Android.

Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución) (Báez et al., 2012).

Como podemos apreciar en la Tabla N°.1 Android es el Sistema Operativo que está dominando el mercado de Smartphone en la actualidad.

Sistema Operativo	Cuota de Mercado (%)
Android	71.81
iOs	27.43
Otros Sistemas Operativos	0.68
Total	100

Tabla 1 Cuota de mercado de los Sistemas Operativos en la actualidad

La Tabla N°.1 muestra el estado actual de los sistemas operativos móviles según un estudio realizado por Gartner en marzo de 2021. Se puede apreciar que Android supone casi un 71.81 del mercado, mientras que los demás sistemas operativos conformarían alrededor de un 28%. Este predominio de Android sobre otros sistemas operativos móviles se basa en el hecho de que Android se ejecuta en muchos más móviles que la competencia ya que es un sistema de código abierto, libre de implementar por cualquier compañía de telefonía. iOS, sin embargo, es privativo y solo se producen móviles con este software desde Apple.

1.2.1. Arquitectura de Android.

La arquitectura interna del sistema operativo o plataforma Android, está básicamente formada por 4 componentes: aplicaciones, almacén de aplicaciones, librerías y kernel/Linux, como se muestra en la siguiente gráfica (Rivera, 2012):

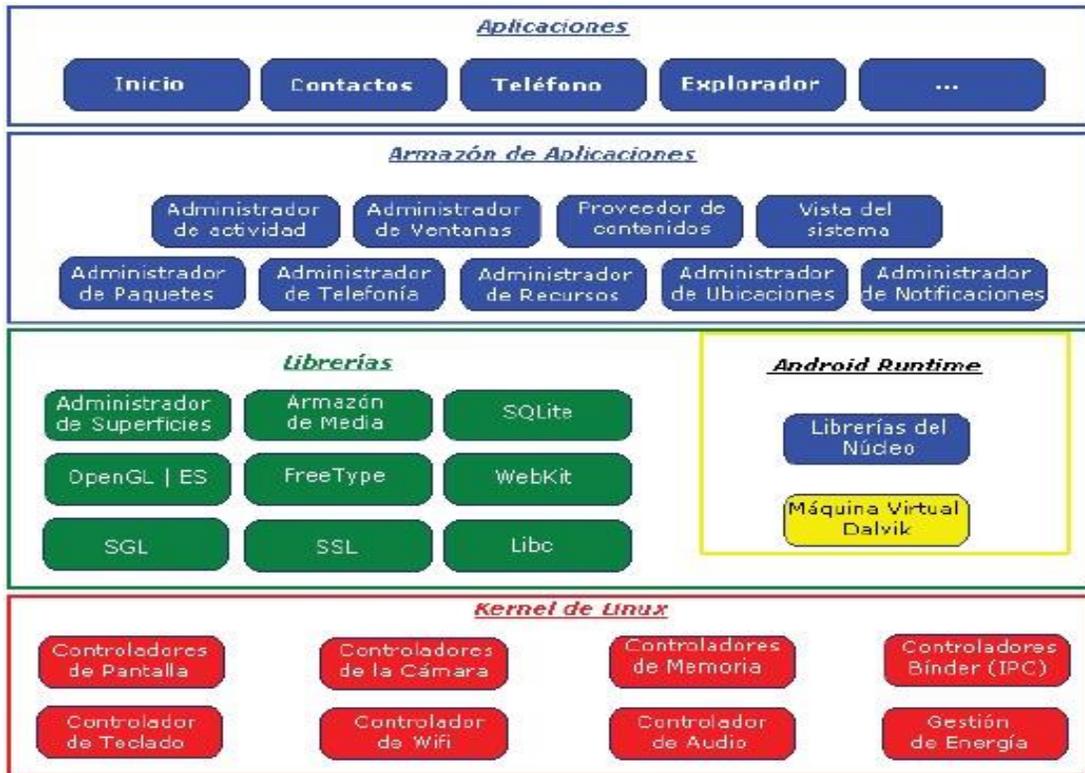


Figura 1 Arquitectura del Sistema Operativo Android.

Si se parte de una vista ascendente, lo primero que se encuentra es el núcleo de Android. Basado en el kernel del sistema operativo Linux. Según (Vanegas, 2012) esta es una capa de abstracción del hardware (hardware abstraction layer), que permite que las aplicaciones accedan a través de controladores (drivers) asumiendo la administración de los recursos del teléfono y del sistema operativo.

A continuación, se localiza el entorno de ejecución (Android Runtime), basado en la máquina virtual de Java. Dadas las limitaciones de los dispositivos Google decidió crear la máquina virtual Dalvik, la cual respondió mucho mejor a dichas

limitaciones. Entre sus características destacan, la optimización de recursos debido a la ejecución de ficheros Dalvik ejecutables (.dex) y la delegación al kernel de Linux procesos como el threading y el manejo de memoria a bajo nivel.

Al mismo nivel se ubican las librerías nativas, escritas en C/C++ y compiladas en código nativo del procesador. Muchas librerías utilizan código abierto.

El entorno de aplicación (Application Framework) ofrece una plataforma de desarrollo libre para aplicaciones, donde su principal riqueza reside en la reutilización de componentes. Componentes desarrollados por Google, o por usuarios.

Por último, se encuentra la capa de aplicaciones (Applications) formada por el conjunto de aplicaciones del dispositivo, ya sean nativas o instaladas por el usuario.

1.2.2. IDE de programación

Un IDE es una herramienta que ayuda a desarrollar de una manera amigable las aplicaciones, brindando ayudas visuales en la sintaxis, plantillas, wizards, plugins, etc.

Android Studio (Benbourahla, 2015), es el entorno de desarrollo específico de Android, basado en IntelliJIDEA (entorno de desarrollo conocido en la comunidad de desarrolladores Java). Aporta muchas novedades respecto a Eclipse.

Entre las características principales de Android Studio, se tiene las siguientes:

- Renderizado en tiempo real
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.

- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.
- Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.

1.3. Aplicación Móvil.

Según (Puig, 2012) una aplicación es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos. Esto lo diferencia principalmente de los sistemas operativos (que hacen funcionar la computadora), las utilidades (que realizan tareas de mantenimiento o de uso general), y las herramientas de desarrollo de software (para crear programas informáticos).

Es una solución informática para la automatización de ciertas tareas complicadas, como pueden ser la contabilidad, la redacción de documentos, o la gestión de un almacén.

De acuerdo a (Hernández Yáñez, 2014), “un programa informático es una secuencia de instrucciones, escritas para realizar una tarea específica en una computadora, ejecutando sus instrucciones en un procesador central directamente. El programa en su formato de código fuente legible para humanos, del cual se derivan los programas ejecutables (ej. compilados), le permite a un programador estudiar y desarrollar sus algoritmos.”

Una aplicación móvil según (Correa Sánchez, 2018), “es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles que permiten al usuario efectuar un conjunto de tareas de cualquier tipo: profesional, de ocio, educativas, de acceso a servicios, etc., facilitando las gestiones o actividades a desarrollar.”

Por lo general, se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, BlackBerry OS, Windows Phone, entre otros.

Ventajas

De acuerdo a (Molina, 2017) al ser aplicaciones residentes en los dispositivos están escritas en algún lenguaje de programación compilado, y su funcionamiento y recursos se encaminan a aportar una serie de ventajas tales como:

- ✓ Acceso más rápido y sencillo a la información necesaria sin necesidad de los datos de autenticación en cada acceso.
- ✓ Almacenamiento de datos personales que, a priori, es de una manera segura.
- ✓ Gran versatilidad en cuánto a su utilización o aplicación práctica.
- ✓ La atribución de funcionalidades específicas.
- ✓ Mejorar la capacidad de conectividad y disponibilidad de servicios y productos (usuario-usuario, usuario-proveedor de servicios, etc.).

1.4. Tipos de aplicaciones móviles según su desarrollo.

En los últimos años el mercado de los dispositivos móviles, en especial los Smartphone, ha mostrado un crecimiento notable en todo el mundo. Siendo las plataformas más sobresalientes iOS y Android. Cada una de estas plataformas cuenta con una infraestructura de desarrollo particular, siendo un reto para los desarrolladores de aplicaciones proporcionar soluciones para todas las plataformas, lo cual tiene un alto costo. La solución ideal a este problema es crear y mantener una única aplicación para todas las plataformas. El desarrollo multiplataforma tiene como objetivo mantener la misma base de código para diversas plataformas. De esta forma el esfuerzo y costo de desarrollo se reduce notablemente (Marichal et al., 2019).

A nivel de programación, existen varias formas de desarrollar una aplicación. Cada una de ellas tiene diferentes características y limitaciones, especialmente desde el punto de vista técnico. Aunque a primera vista esto no parezca incumbencia del diseñador, la realidad es que el tipo de aplicación que se elija, condicionará el diseño visual y la interacción (Delia et al., 2013).

Se describen tres tipos de aplicaciones móviles: aplicaciones nativas, aplicaciones Web y aplicaciones híbridas.

1.4.1. Aplicaciones nativas.

Las aplicaciones nativas son aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo a los programadores, llamado genéricamente Software Development Kit o SDK., Android, iOS y Windows Phone tienen uno diferente y las aplicaciones nativas se diseñan y programan específicamente para cada plataforma, en el lenguaje utilizado por el SDK. (Marichal et al., 2019)

Este tipo de apps (aplicaciones) se descarga e instala desde las tiendas de aplicaciones, con ciertas excepciones en el caso de Android, que sacan buen partido de las diferentes herramientas de promoción y marketing de cada una de ellas.

Las aplicaciones nativas se actualizan frecuentemente y en esos casos, el usuario debe volver a descargarlas para obtener la última versión, que a veces corrige errores o añade mejoras. Una característica generalmente menospreciada de las apps nativas, es que pueden hacer uso de las notificaciones del sistema operativo para mostrar avisos importantes al usuario, aun cuando no se esté usando la aplicación, ejemplo de esto son los mensajes generados por la aplicación.

Las aplicaciones nativas no requieren Internet para funcionar, por lo que ofrecen una experiencia de uso más fluida y están realmente integradas al teléfono, lo cual les permite utilizar todas las características de hardware del terminal, como la cámara y los sensores (GPS, acelerómetro, giroscopio, entre otros).

A nivel de diseño, esta clase de aplicaciones tiene una interfaz basada en las guías de cada sistema operativo, logrando mayor coherencia y consistencia con el resto de aplicaciones y con el propio sistema operativo. Esto favorece la usabilidad y beneficia directamente al usuario que encuentra interfaces familiares.

1.4.2. Aplicaciones Web

La base de programación de las aplicaciones web, también llamadas web apps, es el HTML, conjuntamente con JavaScript y CSS (Cascade Style Sheet) (Lie et al., 2005), estas son herramientas muy utilizadas por los programadores web.

En este caso no se emplea un SDK, lo cual permite programar de forma independiente al sistema operativo en el cual se usará la aplicación. Por eso, estas aplicaciones pueden ser fácilmente utilizadas en diferentes plataformas sin mayores inconvenientes y sin necesidad de desarrollar un código diferente para cada caso particular.

Las aplicaciones web no necesitan instalarse, ya que se visualizan usando el navegador del teléfono como un sitio web normal. Por esta misma razón, no se distribuyen en una tienda de aplicaciones, sino que se comercializan y promocionan de forma independiente.

Al tratarse de aplicaciones que funcionan sobre la web, no es necesario que el usuario reciba actualizaciones, ya que siempre va a estar viendo la última versión. Pero, a diferencia de las apps nativas, requieren de una conexión a Internet para funcionar correctamente (Lujan-Mora, 2002).

Adicionalmente, tienen algunas restricciones e inconvenientes en factores importantes como gestión de memoria y no permiten aprovechar al máximo la potencia de los diferentes componentes de hardware del teléfono. Las aplicaciones web suelen tener una interfaz más genérica e independiente de la apariencia del sistema operativo, por lo que la experiencia de identificación del usuario con los elementos de navegación e interacción, suele ser menor que en el caso de las nativas (Delia et al., 2013).

1.4.3. Aplicaciones Híbridas

Las aplicaciones híbridas son una especie de combinación entre las aplicaciones web y nativas. La forma de desarrollarlas es parecida a la de una aplicación web, usando HTML, CSS y JavaScript, y una vez que la aplicación está terminada, se compila o empaqueta de forma tal, que el resultado final es

como si se tratara de una aplicación nativa. Esto permite casi con un mismo código obtener diferentes aplicaciones, por ejemplo, para Android y iOS, y distribuirlas en cada una de sus tiendas. A diferencia de las aplicaciones web, estas permiten acceder, usando bibliotecas, a las capacidades del teléfono, tal como lo haría una app nativa (Solanky et al., 2016).

Las aplicaciones híbridas, también tienen un diseño visual que no se identifica en gran medida con el del sistema operativo. Sin embargo, hay formas de usar controles y botones nativos de cada plataforma para adherirse más a la estética propia de cada una. Existen algunas herramientas para desarrollar este tipo de aplicaciones. Apache Córdova es una de las más populares, pero existen otras alternativas como Icenium o Xamarin.

¿Qué tipo de aplicación implementar?

Dadas las características de cada una de las aplicaciones, decidirse por una u otra estará determinado por un conjunto de factores fundamentales y por la forma en que afectan finalmente la experiencia de uso. Cuando la disponibilidad de la app sin Internet, la posibilidad de usar notificaciones y el acceso a los recursos de hardware del teléfono sean importantes, una aplicación nativa será la opción más indicada.

Si ninguna de estas características es realmente importante para la aplicación, diseñar una aplicación web sería una mejor opción, si es que ya se dispone del conocimiento para ello, heredado del desarrollo de sitios web. En este caso, el costo de desarrollo es más bajo y la forma de trabajar un poco más ágil. Independientemente de esto, las aplicaciones nativas son las que ofrecen una mejor experiencia de uso y, sobre todo, rendimiento. Algunas apps como Facebook o LinkedIn, que antes eran híbridas, han pasado a ser nativas por este motivo. Adicionalmente, ellas responden más a las guías de diseño de cada sistema operativo. (Marichal et al., 2019)

1.4.4 Versiones de Android

Android tiene un gran número de actualizaciones desde su primer lanzamiento. En la figura N^o.2 se muestran las diferentes versiones.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.8%
4.2 Jelly Bean	17	99.1%
4.3 Jelly Bean	18	98.2%
4.4 KitKat	19	97.9%
5.0 Lollipop	21	93.4%
5.1 Lollipop	22	91.4%
6.0 Marshmallow	23	83.3%
7.0 Nougat	24	71.0%
7.1 Nougat	25	62.8%
8.0 Oreo	26	56.9%
8.1 Oreo	27	49.1%
9.0 Pie	28	34.8%
10. Q	29	2.8%

Figura 2. Versiones del Sistema Operativo Android

Android es un sistema operativo móvil basado en Linux enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros. Es desarrollado por la Open Handset Alliance, liderada por Google.

La estructura del sistema operativo Android, se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica, un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic.

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK). Existen otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C, C++ u otros lenguajes de programación.

Android se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias, donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

En la actualidad según (Cuenca & Marjorie, 2015), existen más de 700.000 aplicaciones para Android y se estima que 1.000.000 de teléfonos móviles se activan diariamente.

Android es criticado por la fragmentación que sufren sus terminales al no ser soportados con actualizaciones por parte de los fabricantes, algo que se podría considerar obsolescencia programada. Esto supone problemas para los desarrolladores que deben lidiar con la retro compatibilidad entre diferentes versiones del sistema operativo.

Ventajas

Según (Su et al., 2019) entre las ventajas más importantes de desarrollar una aplicación Android, se tiene las siguientes:

- ✓ Android es compatible con varios dispositivos y actualmente es el Sistema Operativo más utilizado por dispositivos móviles, por lo tanto, la aplicación se podría instalar en la mayoría de dispositivos móviles.
- ✓ Android es un software libre gracias a la licencia Apache, lo cual le convierte en un sistema operativo totalmente abierto, donde cualquier desarrollador pueda crear, modificar y mejorar el código de la aplicación.
- ✓ El desarrollo de una aplicación Android no requiere de permisos a terceros, existe total libertad.
- ✓ Existe una amplia comunidad de desarrolladores en la red.

1.5. Metodologías para el desarrollo del software.

Hoy en día existen diferentes metodologías que apoyan al proceso de desarrollo de software de aplicación. Se muestran las características principales entre los distintos métodos ágiles que más se conocen actualmente para el desarrollo de aplicaciones y que finalmente permitieron escoger uno que satisfaga las necesidades, el tamaño, el tipo y la complejidad del proyecto.

1.5.1. Metodologías ágiles.

Las metodologías ágiles se basan en valores y principios establecidos en el manifiesto ágil, que permiten a los equipos la entrega rápida de software de alta

calidad y que responden a los cambios que puedan surgir mientras se desarrolla el proyecto.

En la actualidad, existen varias metodologías ágiles, entre las más notables y aplicables se encuentran Extreme Programming (XP) y Scrum, junto con estas también se tiene Mobile-D, que en términos generales procuran responder a los principios del manifiesto ágil. Se describen las particularidades de cada una de ellas.

1.5.1.1 Extreme Programming(XP).

XP según (Bustamante, 2014), es una metodología para equipos de desarrollo que pueden tener un tamaño pequeño o mediano (de dos a diez programadores), que promete enfrentarse a la imprecisión y cambios rápidos de los requerimientos del proyecto.

1.5.1.2 Scrum.

Scrum es un marco de trabajo iterativo e incremental que se centra en la entrega de un producto al final de un período de trabajo al que se le denomina Sprint, y en el caso del software, se refiere a que el código se ha integrado, probado y puede ser potencialmente productivo (M, 2015).

1.5.1.3 Mobile-D

Mobile-D es una metodología ágil que está pensada para un equipo con un menor número de desarrolladores y se orienta en superar las dificultades implicadas en el desarrollo de aplicaciones móviles en un tiempo corto. (Su et al., 2019)

1.5.2. Selección de la metodología de desarrollo.

Para poder realizar la aplicación para el Sistema de control de los convenios de producción con el sector privado y la UEB Porcina en la provincia de Villa Clara, es importante usar la metodología ágil que se centre en los requerimientos específicos que el desarrollo de aplicaciones móviles necesita.

De la bibliografía consultada acerca de cada una de las metodologías estudiadas, se pudo obtener una comparación entre ellas, mostrada en la siguiente tabla (Fernando Diego & Vinicio Jaime, 2014).

Tabla comparativa de metodologías		
	Ventajas	Desventajas
XP	<ul style="list-style-type: none"> -Ofrece proyecciones de software de calidad y se preocupa de los límites de tiempo. -Prueba detenidamente todos los aspectos del software, lo que produce un software de calidad. -El proceso de desarrollo puede ser completamente visualizado y medido. -Los casos de estudio son sencillos 	<ul style="list-style-type: none"> -Impone un desarrollo basado en el código, no en el diseño. -La documentación sobre el diseño es escasa. -Es difícil de implementar pues requiere de grandes equipos de desarrolladores, además de una gran disciplina para completar el proyecto. -El diseño incremental no favorece a los requerimientos de software actuales.
SCRUM	<ul style="list-style-type: none"> -Promueve una acelerada corrección de errores. -Ofrece fácil visualización de la implementación del proyecto. 	<ul style="list-style-type: none"> -La falta de tiempo límite permite que el usuario continúe solicitando cambios. -Los requerimientos deben ser perfectamente definidos
	<ul style="list-style-type: none"> -Promueve la entrega de software de calidad dentro de los cronogramas. -Ofrece feedback permanente de los clientes. 	<ul style="list-style-type: none"> para que se pueda hacer una estimación real de costos y tiempos.
Mobile-D	<ul style="list-style-type: none"> -Los casos de estudios son fáciles de entender. -Promueve detección y corrección de errores de manera temprana. 	<ul style="list-style-type: none"> -Pensado para equipos de desarrollo de 10 personas o menos. -El equipo debe estar enteramente involucrado y

	<p>-Promueve la entrega de prototipos de calidad en corto tiempo.</p> <p>-Se tiene feedback constante de los clientes.</p> <p>-Promueve el trabajo en equipo.</p> <p>-El proceso de desarrollo puede ser completamente visualizado y medido.</p>	<p>comprometido con el proyecto para que este sea exitoso.</p> <p>-Hace énfasis en la refactorización del software, el tiempo, el proceso de implementación, lo que puede disminuir la productividad de otros aspectos.</p>
--	--	---

Tabla 2 Comparativa de metodologías.

Se recomienda aplicar los beneficios de Mobile-D y RUP proporciona las siguientes razones para ser la metodología seleccionada en el desarrollo del proyecto:

- Es una metodología ágil con ciclos de desarrollo cortos y para equipos pequeños.
- Está diseñada para el desarrollo de aplicaciones móviles.
- Tiene facilidad para detectar y resolver tempranamente problemas técnicos.
- Garantiza baja densidad de defectos en las liberaciones de los productos.
- Se basa en el desarrollo basado en pruebas que es una de las mejores formas de asegurar la calidad.
- Tiene un enfoque centrado en la satisfacción del usuario final, permitiendo mejorar el producto al realizar iteraciones cortas.
- Las tareas a realizar en cada fase están bien detalladas.

1.5.3. Ciclo de vida de Mobile-D

El ciclo de vida de Mobile-D se divide en cinco fases: exploración, inicialización, producción, estabilización y prueba.

La fase de exploración tiene como propósito la planificación y el establecimiento del inicio del proyecto. La fase de Exploración puede desvincularse oportunamente de las fases posteriores y también se superpone con la fase de iteración 0. La fase de Exploración es una fase importante que sienta las bases

para una implementación controlada del desarrollo del producto de software, por ejemplo, las cuestiones relacionadas con la arquitectura del producto, el proceso de desarrollo del software y la selección del ámbito. Los diferentes grupos de interés (stakeholders) son necesarios para proporcionar su conocimiento en la fase de Exploración.

El propósito de la fase de Inicialización es permitir el éxito de las próximas etapas del proyecto mediante la preparación y verificación de todos los temas críticos del desarrollo, de modo que todos ellos estén en plena disposición al final de la fase, para luego realizar la implementación de los requisitos seleccionados por el cliente.

El propósito de la fase de Producción es implementar la funcionalidad requerida en el producto, mediante la aplicación del ciclo del desarrollo incremental.

El propósito de la fase de Estabilización es asegurar la calidad de la ejecución del proyecto.

El propósito de la fase de pruebas y corrección del sistema es determinar si el sistema producido implementa la funcionalidad definida por el cliente de manera correcta, proporcionando al equipo encargado del proyecto, la realimentación de la funcionalidad del sistema y la corrección de los defectos encontrados.

En la figura 4 se muestran las fases de la metodología Mobile-D (Muñoz Muñoz, 2020)



Figura 3 Fases de la metodología Mobile-D

Las salidas correspondientes de las 5 fases de la metodología Mobile-D se describen con más detalles en (Muñoz Muñoz, 2020).

1.6. Diseño de aplicaciones móviles para el sistema operativo Android.

El diseño de aplicaciones para dispositivos móviles es fundamental, pues de ello depende el éxito una vez que sea desplegada y los usuarios comiencen a utilizarla. Se debe hacer énfasis en tres objetivos fundamentales a tener en cuenta en el desarrollo y diseño de estas aplicaciones. Las apps deben ser estéticamente placenteras en cada nivel o sección. Las transiciones deben ser rápidas y limpias, la organización y las tipografías deben ser atractivas y llenas de significado. Los iconos son como obras de arte si están bien utilizados ya que una simple imagen, bien conformada, dice más que muchas palabras. Como una herramienta bien hecha, las aplicaciones deben combinar belleza, simpleza y propósito para crear una verdadera experiencia con mínimo esfuerzo, pero poderoso a su vez.

Deben simplificarnos las tareas que realizamos comúnmente y a su vez deben ser fáciles de entender, intuitivas. Cuando una persona utiliza una app por

primera vez, esta debe ser capaz de intuir fácilmente las características o funcionalidades más importantes. Hacer las operaciones más complejas de una forma transparente, dejando al usuario la menor cantidad de interacción posible. Las personas de cualquier edad o cultura deben sentirse totalmente en control y no abrumadas por muchas decisiones o señalamientos innecesarios. No es suficiente con desarrollar una aplicación que sea fácil de utilizar. Las aplicaciones tienen que inspirar a las personas a probar cosas nuevas y utilizar las aplicaciones de formas innovadoras. Permitir a las personas combinar aplicaciones a diferentes flujos de trabajo a través de multitareas, notificaciones e incluso compartir contenido entre aplicaciones (Treude and Aniche, 2018). Todos los sistemas operativos proponen diferentes formas de interactuar con los elementos en pantalla. Conocer la diferencia entre ellos y utilizar elementos familiares para el usuario, asegura que este se sienta cómodo y seguro usando la aplicación.

1.6.1. Principio de experiencia de usuario.

Cada sistema operativo tiene su propia identidad que es reflejada en la apariencia y comportamiento de cada uno de los elementos que componen su interfaz. En ellos imprime su personalidad, lo que hace que la experiencia sea diferente a las demás. Sin embargo, todos comparten algunos puntos de vista fundamentales que se manifiestan en el diseño de sus interfaces (Hassan Montero and Martín Fernández, 2005), estos son los principios de experiencia de usuario:

Simplicidad

La simplicidad visual está directamente relacionada con la usabilidad. Ser simple implica en cierta medida ser mínimo, contar con pocos elementos, sobre todo, que aquellos presentes en la interfaz tengan una función bien definida que contribuya a cumplir el objetivo de la app y ayude al usuario.

Los móviles no son dispositivos para mostrar mucha información en pantalla. Por esta misma razón, la simplicidad consiste también en manejar la economía visual y tener un buen criterio para determinar que incluir y que no en el diseño.

Una gran cantidad de elementos puede abrumar al usuario, por eso, lo que está en pantalla tiene que ser necesario en ese momento y en esa situación de uso. Hacer el diseño simple es bastante complicado, pero reporta grandes beneficios en la experiencia de uso de la aplicación.

Consistencia

Una app tiene diferentes pantallas que la componen y al mismo tiempo, está dentro de un sistema operativo que propone un determinado aspecto visual e interacción. El usuario de Android, iOS o Windows Phone ya está habituado a ellos y espera que las aplicaciones se comporten de la misma manera. La consistencia, entonces, se trata de respetar estos conocimientos y costumbres del usuario, no solo en el interior de la aplicación, sino también en relación con el resto del Sistema Operativo. Esto favorece el uso intuitivo de la app, ya que el usuario puede prever su comportamiento sin demasiado esfuerzo.

La relación existente entre apariencia y comportamiento también tiene que ser consistente. El aspecto visual de un elemento interactivo determinado (como un botón con un icono) puede llevar a esperar un comportamiento específico de acuerdo a la forma en que se ve. Por ejemplo, si se usa un botón que representa la acción “eliminar” en el sistema operativo, el usuario esperara que también dentro de la app haga lo mismo. Cumplir con esa expectativa habla de consistencia.

Navegación intuitiva

Un aspecto que merece mucha atención en una aplicación es la forma de navegar entre contenidos, de manera que resulte fácil de comprender para el usuario, evitando la sensación de desorientación que puede ocasionar una navegación confusa. La navegación intuitiva esta también relacionada con la consistencia. Cada sistema operativo propone diferentes elementos para navegar por la app como botones, pestañas y paneles. Hacer uso de ellos hará que el usuario los reconozca a primera vista y, solo con estos componentes, ya sepa como ir de una sección a otra.

Por otro lado, para el usuario es importante saber y prever que pasará después de pulsar un botón o como se mostrarán las pantallas. Distinguir donde se está

dentro de los contenidos de la aplicación y conocer como volver hacia atrás son factores muy importantes que alivian al usuario y le ahorran esfuerzos inútiles por intentar comprender como ir de un sitio a otro. Una navegación intuitiva permite, justamente, lograr un uso fluido y sin esfuerzo de la aplicación.

El estilo de las interfaces

La interfaz de una aplicación es como la ropa que viste. Es también la capa que hay entre el usuario y el corazón funcional de la app, el lugar donde nacen las interacciones. En mayor medida está compuesta por botones, gráficos, iconos y fondos, que tienen una apariencia visual diferente en cada uno de los sistemas operativos, porque Android, iOS y Windows Phone tienen su propia forma de entender el diseño. El trabajo del diseñador consiste en interpretar la personalidad de cada sistema operativo, aportando su propia visión y estilo de diseño, para conseguir aplicaciones que, además de ser fáciles de usar, sean distintas a las demás y tengan coherencia visual con la plataforma que las acoge (Farman, 2013).

1.6.2. Interfaces nativas o personalizadas.

Las interfaces nativas se basan en elementos (botones, listas y encabezados) que vienen preestablecidos en cada plataforma. Tienen un aspecto ya definido en cuanto a las características básicas de su apariencia como color, tamaño o tipo de fuente, que pueden ajustarse en mayor o menor medida para que se correspondan con la estética buscada. En el momento de comenzar a diseñar es recomendable definir la interfaz con elementos nativos. De esta forma, se consigue una buena base sobre la cual trabajar y no es necesario crear todos los elementos desde cero.

El inconveniente de las interfaces nativas es que limitan la personalidad del diseño y, en algunos casos, es necesario ir un paso más allá. En situaciones como esta, todos o algunos elementos de la interfaz pueden ser personalizados, lo cual se logra creándolos de nuevo como imágenes. Por ejemplo, un elemento visual personalizado podría ser un campo de entrada de texto de un formulario que se genera como imagen, para aprovechar la

posibilidad de incluir texturas, relieves o sombras específicas, que un elemento nativo no ofrece.

Diseñar una interfaz personalizada tiene que planearse de antemano porque representa una mayor complejidad y tiempo de desarrollo. De la misma manera, no siempre los diseños de este tipo de interfaces se trasladan a la aplicación funcional de forma fidedigna, pues su correcta implementación queda en manos de la pericia del desarrollador.

¿Qué tipo de interfaz utilizar?

En la mayoría de los casos no se trata de elegir entre una u otra, sino de alcanzar el balance adecuado combinando ambas. La situación más habitual suele ser partir de una interfaz nativa y personalizar solo aquellos elementos que se consideren necesarios.

El tipo de aplicación también tiene incidencia directa en este asunto. Por ejemplo, las aplicaciones que dan valor a los detalles visuales y a la experiencia en general, suelen contar con más de un elemento personalizado. Tal es el caso de Path y su forma de añadir “momentos”, algo diseñado y desarrollado a medida. Por el contrario, hay otras aplicaciones que dan más valor al cumplimiento de tareas y que se benefician de tener una apariencia más limpia que no distraiga del proceso. Un ejemplo de ellas es WhatsApp, cuyo aspecto visual está completamente basado en elementos nativos del sistema operativo. Además del objetivo o del tipo de aplicación, hay otras variables para considerar. Las interfaces nativas tienen un punto a favor y es que están constituidas por elementos que el usuario ya conoce y a los que está habituado, por lo tanto, no representan un nuevo aprendizaje. Esto puede incidir favorablemente en la usabilidad de la app. Por otro lado, las interfaces personalizadas pueden ofrecer una apariencia más acabada, pero al trabajar con ellas hay que considerar la compatibilidad con múltiples dispositivos, es decir cómo se visualiza el mismo diseño en teléfonos diferentes y el rendimiento general.

Conclusiones parciales.

En este capítulo se fundamenta teóricamente un conjunto de metodologías, herramientas y principios para el desarrollo de aplicaciones móviles

A partir de los aspectos estudiados, para resolver la problemática planteada, se selecciona la metodología de desarrollo Mobile-D, por sus características para el desarrollo de aplicaciones móviles en específico, se determina el desarrollo una aplicación nativa, acorde al avance de los dispositivos móviles en cuanto a la compatibilidad con las distintas versiones del sistema operativo Android.

Se implementará sobre sistema operativo Android, mayoritariamente difundido en Cuba, y se tendrán en cuenta los principios de experiencias de usuario, etc.

CAPÍTULO II. MODELO DEL NEGOCIO Y REQUISITOS DEL SISTEMA

En este capítulo se caracteriza el negocio, la gestión de la producción de los productores del sector privado en la empresa de Porcino de Villa Clara, exponiendo las actividades fundamentales que se desarrollan. Se da a conocer el modelo de casos de uso del negocio y los requisitos que debe cumplir el sistema.

2.1 Modelado de negocio.

La empresa UEB Porcina de Villa Clara, realiza sus actividades de negocio fundamentales a partir de la contratación de los productores privados, que interactúan con la misma por medio de la compra de insumos a la empresa y la venta de sus producciones. Cuando un productor de carne de cerdo desea afiliarse a la empresa UEB Porcina, y hacer un contrato con la misma, debe dirigirse personalmente de forma presencial al departamento comercial de la empresa, lugar donde debe entregar una serie de documentos legales y de sanidad indispensables para su contrato. Luego el comercial revisa que todos los requerimientos estén en orden para registrarlo en el sistema de la empresa como un nuevo asociado.

Una vez asociado el productor, puede realizar convenios con la empresa. Para ello, el productor debe dirigirse a la empresa nuevamente en el departamento comercial en donde se pone de acuerdo con el comercial sobre los por menores del acuerdo, cantidad de carne que debe entregar, cantidad de alimento que la empresa debe brindarle para la cría de cerdos, y otras. Para ver el seguimiento y cumplimiento de lo acordado con la UEB, el productor asociado debe estar constantemente dirigiéndose a la empresa y consultar esta información.

El seguimiento de los convenios por los productores es de gran importancia para estos y a la empresa le interesa un constante flujo de información actualizada entre la propia empresa y productores, así que viabilizar estas actividades es de gran valor para aumentar la eficiencia en el trabajo de los mismos y disminuir la pérdida de tiempo de trabajo de los productores y la concentración de personas en las oficinas comerciales de a UEB.

La UEB está motivada a contar con un sistema de información que permita a UEB, controlar la interacción con los productores. A continuación, se describe el entorno del negocio y los procesos asociados a la relación de la UEB con los productores privados.

Como **actor del negocio** se reconoce al productor del sector privado, eje fundamental y principal beneficiado con los procesos de negocio que se realizan.

Teniendo en cuenta la descripción del negocio y las actividades que se realizan en el mismo, se obtiene el siguiente **diagrama de casos de uso del negocio** el cual se muestra en la figura 4.

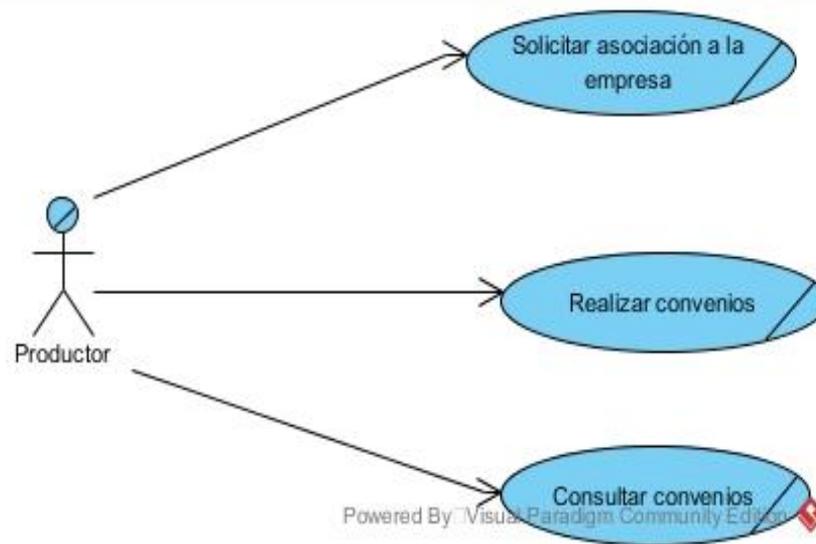


Figura 4 Diagrama de caso de uso del negocio.

Un trabajador del negocio es una abstracción de un humano o software que representa un rol dentro del desarrollo de una realización de un caso de uso del negocio, (Lorenzo, 2016b). Se reconocen como **trabajador del negocio**:

- Comercial: Trabajador que se encarga de atender a los productores para la realización y gestión de los convenios.

Para la gestión de todo el flujo de información relacionado con la empresa se usa actualmente el programa desarrollado para este fin “Regente”, el cual fue desarrollado en el lenguaje Delphi con una conexión a una base de datos de

tipo SQL Server localmente. El funcionamiento de este sistema actualmente empleado, está dirigido a: gestionar los productores asociados a la empresa, gestionar los convenios, compras de carne, la venta de alimentos entre otras funcionalidades. Las entidades más importantes que se manejan en dicho sistema se muestran en las relaciones o tablas de la base de datos, entre estas las fundamentales son: convenios, compra, productores y alimentos, el diseño de la base de datos aparece en el Anexo 1.

Las relaciones o tablas más importantes son:

Productores: Almacena los datos que caracterizan a los productores del sector privado, Sus atributos más relevantes son:

- Id: identificador del productor, su valor es un entero autoincrementado
- Código: es el código del asociado establecido por la empresa cuando asocia un producto, su valor es un entero
- Nombre: nombre completo del productor, su valor es una cadena de texto
- Municipio: llave foránea de la tabla municipio, su valor es un entero
- Grupo: grupo al que pertenece el productor. su valor es un entero
- Tipo: llave foránea de la tabla tipo de productor, su valor es un entero
- Móvil: Numero del teléfono del productor para poder estar en contacto con el productor, su valor es un entero
- Email: Correo del productor para poder estar en contacto con el productor, su valor es cadena de texto

Convenios: Almacena los datos referentes a los convenios establecidos por los productores. Sus atributos más importantes son:

- Id: identificador del convenio en la tabla, su valor es un entero autoincrementado
- Número: es un número que se le da al convenio por la empresa cuando el productor establece un convenio con la misma

- Municipio: llave foránea de la tabla municipio, municipio en el cual se acordó en el convenio, su valor es un entero
- Año: Año en la que se estableció el convenio, su valor es de tipo entero
- Tipo: Tipo de convenio que se estableció, este dato es una llave foránea de la tabla tipoconv (Tipo de Convenio), su valor es un entero
- Productor: productor que estableció el convenio con la empresa porcina, en este se guarda la llave foránea de la tabla productor, su valor es un entero
- Cab: Cantidad de cerdos acordados en el convenio
- peso inicial: Peso inicial que tiene los cerdos al establecerse el convenio, su valor es de tipo double
- peso final oficial: Peso final que se debe entregar los cerdos al concluir el convenio, su valor es de tipo double
- peso final diferenciado: Diferencia del peso con la que se entrega el cerdo con el peso oficial acordado en el convenio, su valor es de tipo double
- fecha inicial: Fecha inicial en la que se estableció el convenio, su valor es de tipo fecha
- fecha final: Fecha final en la que se acordó en el convenio, su valor es de tipo fecha
concluido: Estado en el que está el convenio, si está concluido o no el mismo, su valor es booleano.
- alimentos: Alimentos establecidos en el convenio

Donde un productor puede tener varios convenios en ejecución.

Compra: Almacena los datos referentes a las compras de carnes que hace la empresa porcina a los productores. Sus atributos más importantes son:

- Id: identificador de la compra en la tabla, su valor es un entero autoincrementado
- Código: es el código de la compra establecido por la empresa cuando esta la realiza, su valor es un entero
- Fecha: Fecha en la que se hizo la compra, su valor es de tipo fecha

- **Productor:** Productor al que le hicieron la compra de la carne, en este se guarda la llave foránea de la tabla productor, su valor es un entero
- **Municipio:** llave foránea de la tabla municipio, municipio en el cual se realizó la compra de la carne, su valor es un entero
- **tipo de compra:** Tipo de la compra, este dato es una llave foránea de la tabla tipocompra (Tipo de Compra), su valor es un entero
- **importe:** Importe pagado al productor por la compra de la carne, su valor es de tipo double
- **cabezas:** Cantidad de cerdos de la compra, su valor es un entero
- **kgs:** Cantidad de kgs de carnes de la compra, su valor es un double
- **fecha de pago:** Fecha en la que se le pago al productor por la compra de la carne, su valor es de tipo fecha

A solicitud de los especialistas de información de la UEB debe utilizarse esta base de datos, sin modificaciones para sustentar la aplicación móvil.

2.2 Requisitos del sistema.

El sistema a desarrollar lleva consigo la satisfacción de una serie de requisitos funcionales. Ellos son una condición y/o exigencia que debe cumplir el sistema a implementar a petición del usuario o cliente, (Lorenzo, 2016c). Se recogen los siguientes requisitos que debe cumplir el software:

RF1	Autenticarse
RF2	Registrarse
RF3	Consultar convenios
RF4	Consultar carnes vendidas
RF5	Consultar alimentos comprados
RF6	Consultar resumen de carnes
RF7	Consultar resumen de alimentos
RF8	Modificar información de perfil

Tabla 3 Requisitos funcionales

De igual manera se pueden identificar requisitos no funcionales. Estos son una restricción sobre la operación del sistema. Se identifican los siguientes:

RNF1	El Sistema será desarrollado para la plataforma Android.
RNF2	El Sistema será desarrollado en el lenguaje de programación Kotlin.
RNF3	El sistema utilizara una interfaz amigable y responsiva (sensible).
RNF4	El sistema es instalable en equipos móviles con versión Android 4.4.2 kitkat o superior.

Tabla 4 Requisitos no funcionales

Los **actores del sistema** son toda entidad externa al mismo que guarda una relación con este, y que le demanda una funcionalidad. Esto incluye a los operadores humanos, pero también incluye a todos los sistemas externos. En el caso de los seres humanos, estos se pueden ver como definiciones de rol, por lo que un mismo individuo puede corresponder como uno o más actores, (Gutiérrez, 2018).

Como actor del sistema se encuentra:

Actor	Descripción
Productor Asociado	Todas las personas que tiene convenio con la Empresa UEB Porcina Villa Clara

Tabla 5 Actores del sistema

Teniendo en cuenta los requisitos anteriores se puede obtener el **diagrama de casos de uso del sistema** que se representa en la figura 5:

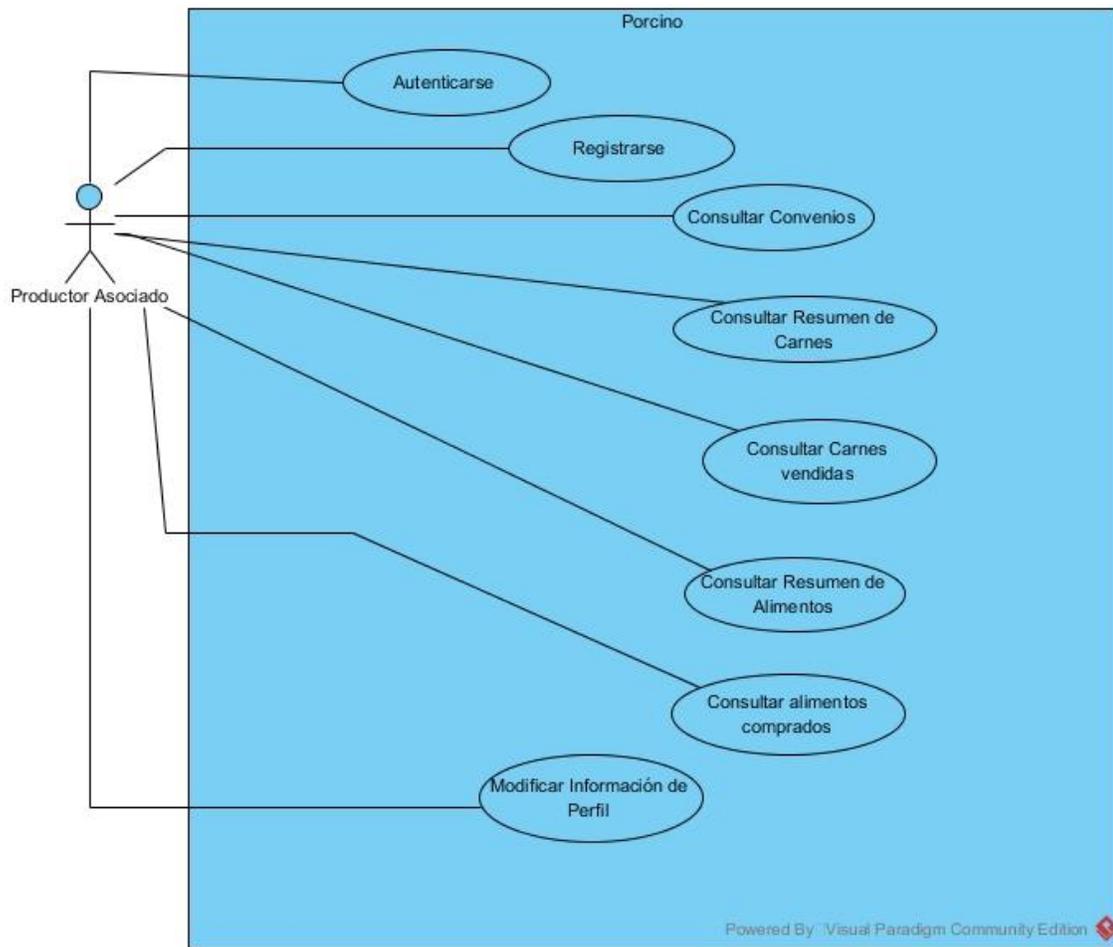


Figura 5 Diagrama caso de uso del sistema

Los casos de uso del sistema responden a qué se quiere que haga el producto. Los casos de uso deben ser especificados.

Se muestra la especificación del caso de uso del sistema Autenticarse en la tabla 6:

Caso de uso: Autenticarse	
Actores	Asociado
Propósito	Autenticarse en el sistema
Casos de uso del negocio asociado	--
Caso de uso del sistema precondition	--
Flujo de eventos	
Acción del actor	Respuesta del sistema

1)El usuario rellena el formulario con usuario y su contraseña y pulsa la opción Login	2)El sistema verifica las credenciales del usuario en el sistema. Si esta correcta las credenciales cambia para la pantalla del menú principal
Flujo alterno	
3)Se repiten los pasos 1 y 2	2)El sistema verifica las credenciales del usuario en el sistema. Si esta incorrecta las credenciales se le notifica el error

Tabla 6 Caso de uso del sistema Autenticarse

Se muestra la especificación del caso de uso del sistema Modificar información de Perfil en la tabla 7:

Caso de uso: Modificar información del perfil	
Actores	Asociado
Propósito	Modificar la información del usuario en el sistema
Casos de uso del negocio asociado	--
Caso de uso del sistema precondición	Autenticarse
Flujo de eventos	
Acción del actor	Respuesta del sistema
1)El usuario selecciona la opción perfil	2)El sistema le muestra un formulario con toda la información guardada del mismo ya insertados.
3)El usuario actualiza los campos deseados en el formulario.	4)El sistema comprueba la información insertada por el usuario. Si está correcta esta se guarda
Flujo alterno	

5)Se repiten los pasos 3 y 4	4)El sistema comprueba la información insertada por el usuario. Si no está correcta, se notifica del error.
------------------------------	---

Tabla 7 Caso de uso del sistema Modificar información de Perfil

Se muestra la especificación del caso de uso del sistema Consultar Convenios en la tabla 8:

Caso de uso: Consultar Convenios	
Actores	Asociado
Propósito	Consultar los convenios del productor con la Empresa UEB Porcina Villa Clara
Casos de uso del negocio asociado	Consultar convenios
Caso de uso del sistema precondición	Autenticarse
Flujo de eventos	
Acción del actor	Respuesta del sistema
1)El usuario selecciona la opción Convenios en la pantalla principal	2) El sistema le muestra una lista de todos los convenios del productor, según los filtros especificados, si el usuario no selecciona los filtros deseados este le mostrara todos los convenios que este posee
3)El usuario selecciona el convenio que desee conocer	
Flujo alterno	
	2) El sistema le muestra una lista de

<p>3) El usuario selecciona los filtros deseados para optimizar su búsqueda</p> <p>5) El usuario selecciona el convenio que desee conocer</p>	<p>todos los convenios del productor, según los filtros especificados, si el usuario no selecciona los filtros deseados este le mostrara todos los convenios que este posee</p> <p>4) El sistema le muestra una lista de todos los convenios del productor, según los filtros especificados</p>
<p>Flujo alterno</p>	
<p>4) Se repiten los pasos 1 y 2</p>	<p>2) Si el sistema no posee acceso a internet le muestra un error de conexión</p>

Tabla 8 Caso de uso del sistema Consultar Convenios

Conclusiones Parciales.

En este capítulo se realizó la descripción del negocio en la UEB Porcina de Villa Clara, los procesos principales se refieren a:

Asociar un productor privado a la UEB y Gestionar los convenios del productor.

.

Se describe la base de datos actual existente que debe ser utilizada también por la aplicación móvil, esta almacena los datos de los productores, los convenios y aquellos relacionados con los convenios de compra de alimentos y venta de los productos.

A partir del análisis del negocio se extraen requisitos del sistema que permiten diseñar e implementar las funcionalidades del mismo para satisfacer las necesidades de información de los productores privados.

CAPÍTULO III. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.

En este capítulo se describe la propuesta de solución, para el desarrollo de este proyecto, en el cual se aplicará la metodología Mobile-D, la cual fue descrita en el capítulo 1 (sección 1.5.3)

A continuación, se describe las actividades realizadas en cada una de las fases que hacen parte de la metodología

3.1. Exploración.

En esta fase se definen los requerimientos, el alcance, así como las bases fundamentales para el desarrollo del producto consecuente con lo que se desea.

Establecimientos de los grupos de interés o stakeholder:

Desarrollador: Es la persona encargada del análisis, desarrollo y pruebas de la aplicación.

UEB Porcina de Villa Clara: Personas que conocen la estructura de la información. También definen junto al desarrollador la información que deciden llevar a la aplicación.

Usuario que usarán la aplicación: Todos los productores del sector privado que tienen convenios con la UEB Porcina de Villa Clara.

Definición del alcance:

El alcance del Proyecto se determina a través de las limitaciones y de los supuestos y dependencias.

Limitaciones

Las limitaciones de la aplicación son:

- La aplicación solo puede ser ejecutada en dispositivos con el Sistema Operativo Android.
- La aplicación móvil requiere de una conexión a internet para poder obtener consultar la información.

Supuestos y dependencias:

Los supuestos y dependencias de la aplicación son:

- Los usuarios manejan como idioma principal el español, siendo este el lenguaje que se muestra en la interfaz de la aplicación.
- La aplicación móvil solo puede ser utilizada por personas que tienen convenios con la Empresa UEB Porcino Villa Clara
- La información manejada por la aplicación se obtiene a través de un servicio API REST

Establecimiento del proyecto

En esta etapa se indica que tipo de herramientas se usaron en el desarrollo del aplicativo móvil que se describe a continuación:

Android Studio: es el entorno de desarrollo integrado (IDE) para el sistema operativo Android de Google, construido sobre el software IntelliJ IDEA de JetBrains y diseñado específicamente para el desarrollo de Android. Permite el desarrollo del proyecto de la aplicación móvil

Kotlin: es un lenguaje de programación de tipado estático que corre sobre la máquina virtual de Java y que también puede ser compilado a código fuente de JavaScript, el cual fue desarrollado principalmente por JetBrains. Permite codificar las funcionalidades de los botones de la aplicación.

Insomnia: es una plataforma API para crear y usar API. Este simplifica cada paso del ciclo de vida de la API y optimiza la colaboración para que pueda crear mejores API, más rápido. Permite probar las peticiones que se realizan a los servicios Web API Rest, devolviendo el resultado de la petición en formato JSON

Adobe XD: Es una herramienta de diseño digital que permite elaborar diseños completos y prototipados tanto para aplicaciones web y móviles, tratando de lograr un mayor nivel de experiencia para el usuario. Para este proyecto en cuestión se utiliza para diseñar los prototipos de las actividades.

3.2. Inicialización.

En esta fase tiene como resultado la realización de actividades al desarrollo y diseño de la aplicación móvil

Configuración del Ambiente del Desarrollo

Esta actividad es específicamente para el desarrollador del aplicativo móvil ya que tiene como propósito el de instaurar un ambiente idóneo para llevar a cabo el desarrollo, en esta actividad además el desarrollador puede ejecutar pruebas de los requerimientos, sin tener la necesidad de una implementación real.

- **Tipo de Proyecto:** Aplicación nativa
- **Lenguaje de Programación para la aplicación móvil:** Kotlin

Preparación del Ambiente

Instalación de las siguientes herramientas: Android Studio, Insomnia.

Planificación de Fases

Fase	Iteración	Descripción
Exploración	Iteración 0	Establecimiento del proyecto, establecimiento de los grupos de interés, limitaciones, supuestos y dependencias
Inicialización	Iteración 0	Análisis de requisitos iniciales
	Iteración 1	Implementación de la funcionalidad de la autenticación de usuarios. Refinamiento y actualización de storycards. Refinamiento de Interfaces.
	Iteración 2	Implementación de la funcionalidad de registro de usuarios. Refinamiento y actualización de storycards. Refinamiento de Interfaces.
	Iteración 3	Implementación de la funcionalidad de consultar convenios. Refinamiento y actualización de storycards. Refinamiento de Interfaces.
	Iteración 4	Implementación de la funcionalidad de resumen de carnes vendidas. Refinamiento y actualización de storycards. Refinamiento

Producción		de Interfaces.
	Iteración 5	Implementación de la funcionalidad de resumen de alimentos comprados. Refinamiento y actualización de storycards. Refinamiento de Interfaces.
	Iteración 6	Implementación de la funcionalidad de consultar de carnes vendidas. Refinamiento y actualización de storycards. Refinamiento de Interfaces.
	Iteración 7	Implementación de la funcionalidad de consultar los alimentos comprados. Refinamiento y actualización de storycards. Refinamiento de Interfaces.
	Iteración 8	Implementación de la funcionalidad de modificar usuario. Refinamiento y actualización de storycards. Refinamiento de Interfaces.
Fase	Iteración	Descripción
	Iteración 9	Refactorización de la funcionalidad de la autenticación de usuarios. Establecimiento de las interfaces definitivas.
	Iteración 10	Refactorización de la funcionalidad de registro de usuarios. Establecimiento de las interfaces definitivas.
	Iteración 11	Refactorización de la funcionalidad de consultar convenios. Establecimiento de las interfaces definitivas.
	Iteración 12	Refactorización de la funcionalidad de resumen de carnes vendidas. Establecimiento de las

Estabilización		interfaces definitivas.
	Iteración 13	Refactorización de la funcionalidad de resumen de alimentos comprados. Establecimiento de las interfaces definitivas.
	Iteración 14	Refactorización de la funcionalidad de consultar de carnes vendidas. Establecimiento de las interfaces definitivas.
	Iteración 15	Refactorización de la funcionalidad de consultar los alimentos comprados. Establecimiento de las interfaces definitivas.
	Iteración 16	Refactorización de la funcionalidad de modificar usuario. Establecimiento de las interfaces definitivas.
Pruebas del Sistema	Iteración 17	Se procede a realizar pruebas del sistema, como las pruebas de caja negra y pruebas de compatibilidad.

Tabla 9 Planificación de Fases

Arquitectura de la aplicación

El patrón arquitectónico a utilizar es MVVM (Model View ViewModel) ya que este nos facilita la separación de la lógica de la interfaz gráfica de usuario y la lógica de negocio o modelo de datos de la aplicación. En MVVM el ViewModel tiene la responsabilidad de convertir los objetos de datos del modelo en un formato que permita manejar y presentar fácilmente. En este sentido, el ViewModel contiene toda la lógica de presentación de la vista. Dado al patrón arquitectónico MVVM se obtuvo el diagrama de paquetes figura 6

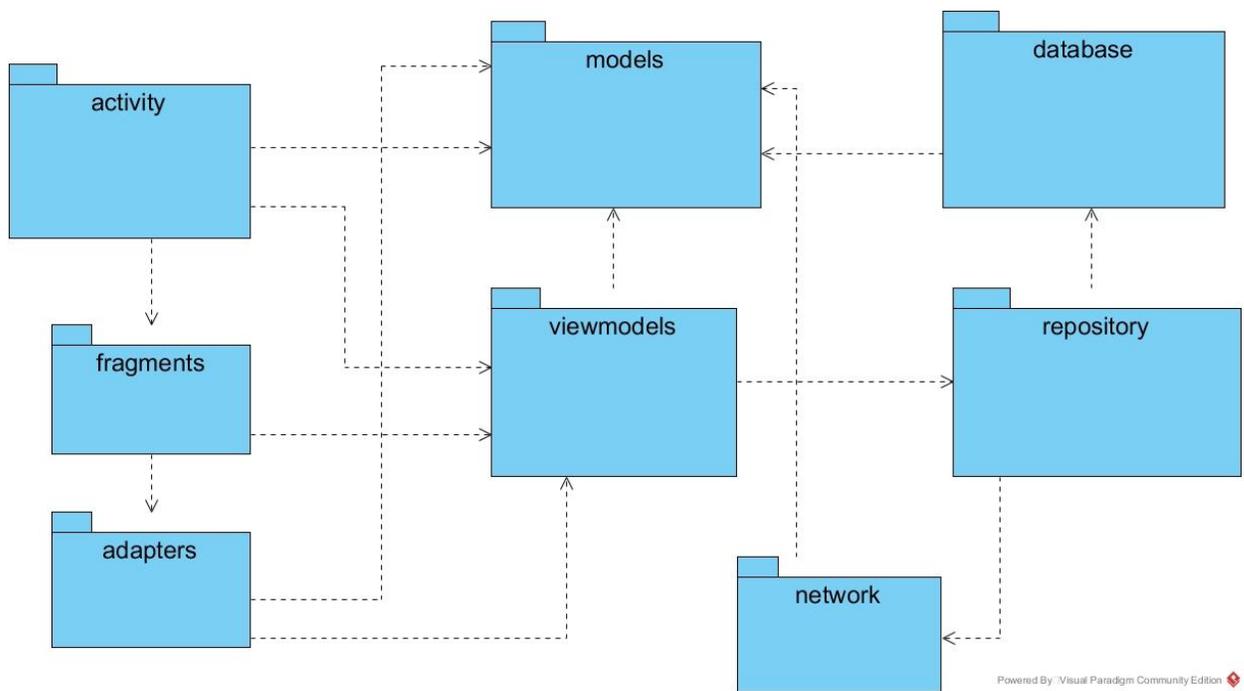


Figura 6 Diagrama de paquetes

En el paquete **Model** se encuentran las clases encargadas para el manejo de los datos de la base de datos o los obtenidos del ApiPorcino. El paquete **Adapters** contiene los adaptadores y los ViewHolder, que son los encargados de mostrar la información en las listas de forma personalizada mejorando la experiencia del usuario. En el paquete **Activity** se encuentran las actividades principales de la aplicación, son las encargadas de navegar a otras vistas o la navegación entre fragmentos. El paquete **Fragment** contiene los distintos fragmentos que serán cargados en las actividades para ejecutar las diferentes funcionalidades. El paquete **ViewModel** contiene todos los viewmodels de las diferentes vistas los cuales manejan la lógica de presentación de la vista. En el paquete **Repository** se encuentras las clases responsables de manejar la información de datos que incluye dónde obtener los datos de un servicio web o de los modelos de datos persistentes.

Esquema de navegabilidad:

En la figura 7, se describe el esquema de navegabilidad de la aplicación móvil, cuyo propósito es describir ante una ventana cuáles son las vistas a las que se pueda acceder y así se precisan las relaciones que hay entre las diferentes vistas

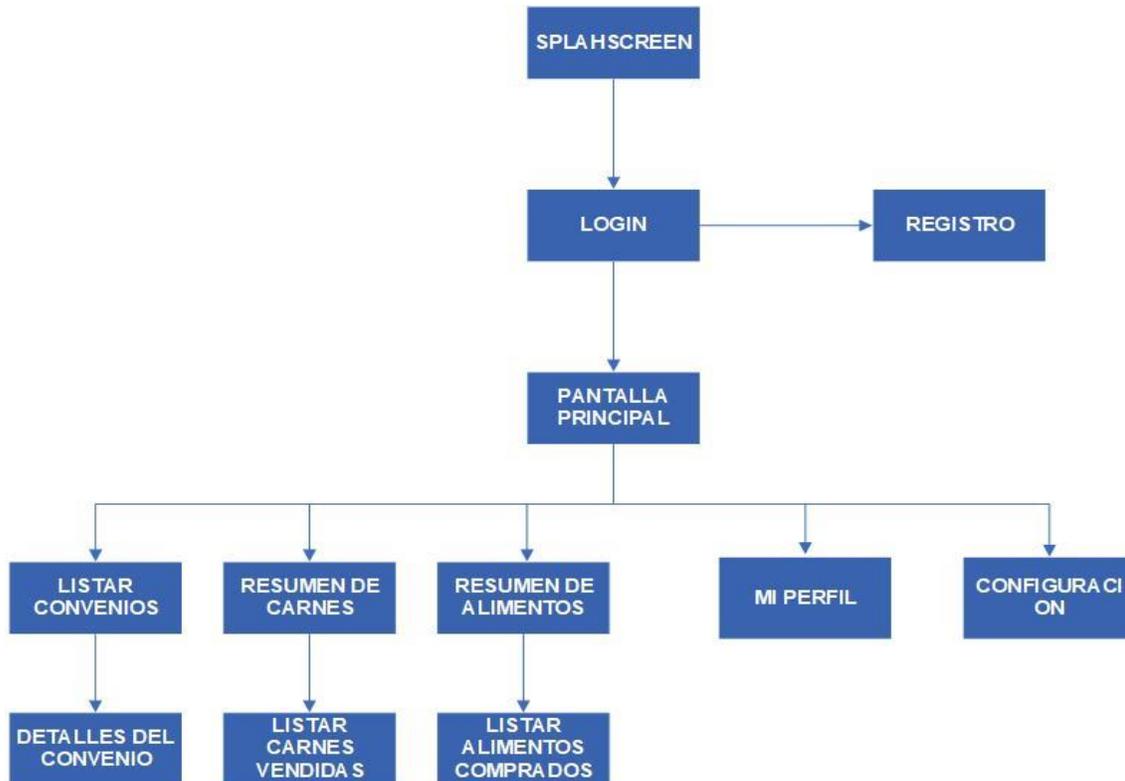


Figura 7 Esquema de navegabilidad

3.3 Producción y Estabilización

En la fase tanto de producción como estabilización se trata de proceder a realizar la funcionalidad de la aplicación móvil junto con todos los requisitos recolectados en las fases anteriores para asegurar que esta cumpla con la calidad deseada de una manera correcta.

Estándares de Codificación

- **Paquetes:** El nombre de los paquetes está escrito en minúscula.

- **Clases:** El nombre de las clases está escrito con la primera letra en mayúscula, si se compone de dos o más palabras la primera letra de cada una de ellas está en mayúscula.
- **Métodos:** Los métodos llevan un nombre relacionado al proceso que ejecutan, están escritos con la primera letra en minúscula, si se compone de dos o más palabras la primera letra de ellos está en mayúscula, a excepción de la primera palabra.
- **Interfaces:** Para los nombres de las interfaces se utilizará la primera letra mayúscula del objeto de la clase esta permite ahorrar líneas de código al momento de ser invocada por algún servicio o función.
- **Layouts:** Los layouts tienen todo su nombre en minúsculas, si poseen más de dos palabras tendrán un guion bajo para separar cada una de ellas.

StoryCard

El propósito de una storycard es escribir cómo un proyecto devolverá valor al usuario. Entonces es el trabajo del equipo de desarrollo ocuparse de cómo desarrollar el código que satisfaga los requisitos de la historia del usuario. En el mejor de los casos, los desarrolladores colaboran estrechamente con los propietarios de negocios y las partes interesadas para aclarar los detalles a medida que se desarrolla el código (Muñoz Muñoz, 2020).

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
-	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	-	-	Baja Media Alta
Descripción						
Fecha	Estado			Comentario		

Tabla 10 Diseño de un StoryCard

Descripción de una storycard

- En el campo **ID** se asigna un número identificador a la historia de usuario.
- En el campo **TIPO** se define qué tipo de actividad se va a describir en esta historia de usuario. Puede ser la implementación de un nuevo requerimiento, el arreglo o la mejora de un requerimiento ya implementado. Este valor varía dependiendo de la fase en la que se encuentre el proyecto. En la fase inicial el tipo siempre se define como Nuevo.
- En el campo **DIFICULTAD** se describe el nivel de dificultad que representará para el equipo de desarrollo el implementar ese requerimiento. Se tiene que ingresar un valor estimado antes y después de la implementación del requerimiento. Los valores referenciales recomendados son Fácil, Moderado, Difícil.
- En el campo **ESFUERZO** se detalla el número de horas que le tomará al equipo de desarrollo implementar este requerimiento. Al igual que con el campo anterior se debe ingresar un valor estimado antes de la implementación y el valor real que tomó la implementación.
- El campo **PRIORIDAD** debe incluir un valor que denote lo sustancial de este requerimiento para el proyecto. Este campo acepta valores referenciales sugeridos de Baja, Normal y Alta.
- El campo **NOMBRE** contiene el nombre de la historia de usuario.
- El campo **DESCRIPCIÓN** lleva el nombre de la historia de usuario que se está cumpliendo y en el campo inferior se describe el desempeño que se va a llevar a cabo en esta historia de usuario, así como los posibles escenarios de éxito y falla que puedan generarse. Esta descripción, según se señala en los buenos procedimientos de las metodologías ágiles, debe ser lo más común posible, sin profundizar en demasiados detalles técnicos.

Para implementar la aplicación se siguen las descripciones y excepciones que se describen en las tablas del StoryCard, se indican todos los datos que

recogen la historia de su desarrollo, incluyendo esfuerzo estimado y real, así como las fechas en que se alcanzan los diferentes estados de implementación. A continuación, se describen las actividades que tiene la aplicación, utilizando la imagen y los significados de los botones, y posteriormente el StoryCard utilizado como guía del desarrollo del mismo:

Login

El Login se le puede definir como un conjunto específico de información que se requiere para poder autenticar al usuario y que al momento de realizar este proceso pueda acceder al servicio que el desee realizar, en muchas aplicaciones móviles se utiliza un Login que generalmente contiene un correo electrónico y contraseña para lo guiarse en la aplicación esto es como una medida de protección para que usuarios terceros no puedan acceder a la aplicación ni a los servicios que brinda en la figura 8.



Figura 8 Login

StoryCard del Login de la aplicación móvil

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
01	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	6	8	Baja Media Alta
Descripción						
Cuando el usuario quiera acceder a la aplicación para consultar toda su información de los convenios con la empresa Porcina, esta le pedirá información como su correo electrónico y contraseña en una pantalla de Login para autenticarse y poder utilizar sin problema los servicios que brinda esta aplicación móvil. Para proteger la información del usuario la contraseña se encuentra encriptada, y además el inicio de sesión cuenta con un token para que accedan sin problema a la aplicación.						
Excepciones						
En caso de que el usuario ingrese un correo electrónico y contraseña incorrecto se le mostrara un mensaje comunicando que se equivocó y que ingrese de nuevo sus datos, esta vez correctamente para acceder a la aplicación, además si un usuario desee registrarse llenando el formulario de registro y el email ya existe este le mostrara una alerta diciendo que ya está registrado un usuario con ese correo y que ingrese otro correo.						
Fecha	Estado		Comentario			
2/02/2021	Definido		Sin Comentario			
3/02/2021	Implementado		Sin Comentario			
5/02/2021	Hecho		Sin Comentario			

Tabla 11 StoryCard del Login

Menú Principal

El usuario una vez autenticado en la aplicación procederá a navegar libremente en el menú principal, donde se le da la opción de convenios para poder consultar todos sus convenios con la UEB porcina, la opción carne para poder consultar el resumen de las carnes vendidas y listar todas las carnes vendidas en todos los convenios, la opción alimentos para poder consultar el resumen de los piensos comprados y listar todos los alimentos comprados en todos los convenios, también se le da la opción de perfil para consultar y poder modificar los datos del mismo, ver figura 9.

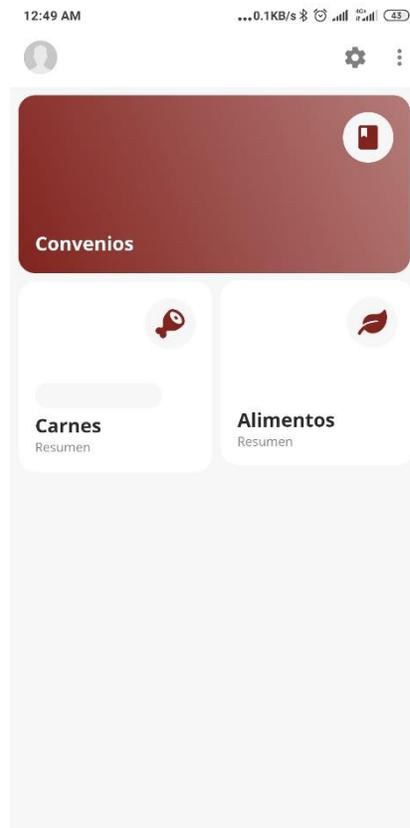


Figura 9 Menú Principal

StoryCard del Menú Principal de la aplicación móvil

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
02	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	2	2	Baja Media Alta
Descripción						
Una vez que el usuario haya procedido a autenticarse esta espera unos pocos segundos para dirigirse a la vista principal del menú, donde podrá observar que puede hacer en la aplicación además de dirigirse a distintas opciones que el desee conocer						
Excepciones						
El usuario debe haberse autenticado antes para dirigirse al menú principal en caso de que no se haya autenticado no podrá dirigirse al menú principal.						
Fecha	Estado		Comentario			
6/02/2021	Definido		Sin Comentario			
7/02/2021	Implementado		Sin Comentario			
9/02/2021	Hecho		Sin Comentario			

Tabla 12 StoryCard del Menú Principal de la aplicación móvil

Consultar Convenios

El productor podrá listar todos sus convenios con la UEB Porcina en función de los filtros aplicados. Este puede filtrar la lista de convenios por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas. Ver figura 10.

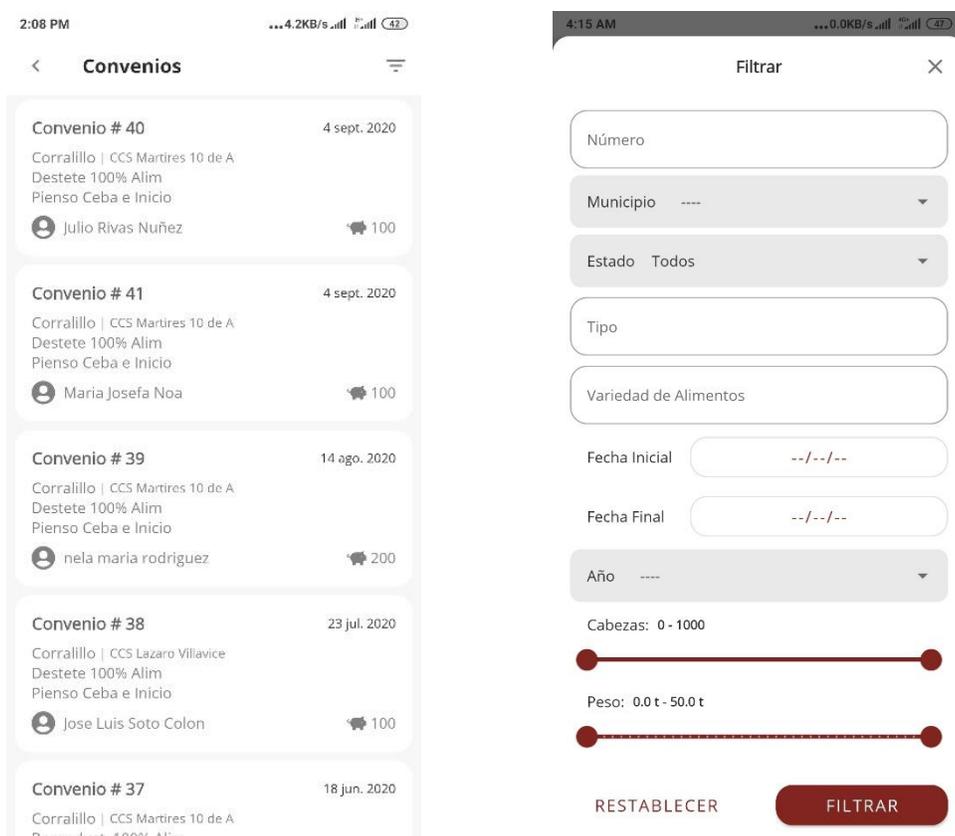


Figura 10 Convenios + Filtro

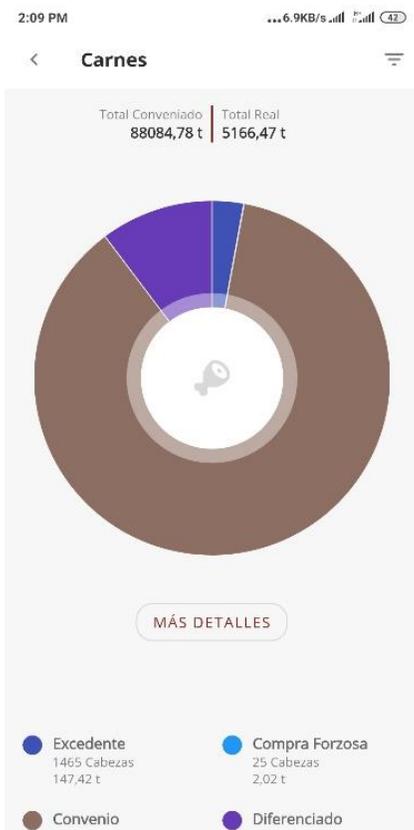
StoryCard de Consultar Convenios

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
03	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	8	10	Baja Media Alta
Descripción						
El productor podrá listar todos sus convenios con la UEB Porcina en función de los filtros aplicados. Este puede filtrar la lista de convenios por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas.						
Excepciones						
Si el usuario no posee una conexión a internet, se le notificara que no posee una conexión al servidor principal						
Fecha		Estado		Comentario		
10/02/2021		Definido		Sin Comentario		
11/02/2021		Implementado		Sin Comentario		
13/02/2021		Hecho		Sin Comentario		

Tabla 13 StoryCard de Consultar Convenios

Resumen de Carnes

El productor podrá observar un gráfico de pastel con las cantidades de cabezas y el peso en toneladas de las carnes vendidas a la empresa porcina de todos los convenios en función de los filtros aplicados. Se le permite filtrar por el número del convenio, por el municipio del convenio, por el importe mínimo y máximo, por un rango de fecha inicial a fecha final de los convenios, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de pienso. Ver figura 11



2:16 PM ...0.1KB/s

Carnes

Filtrar

Número convenio

Municipio

Importe mínimo

Importe máximo

Fecha Inicial: --/--/--

Fecha Final: --/--/--

Año

Cabezas: 0 - 1000

Peso: 0.0 t - 50.0 t

RESTABLECER FILTRAR

Figura 11 Resumen de carnes + Filtro

StoryCard de Resumen Carnes

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
04	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	10	11	Baja Media Alta
Descripción						
El productor podrá observar un gráfico de pastel con las cantidades de cabezas y el peso en toneladas de las carnes vendidas a la empresa porcina de todos los convenios en función de los filtros aplicados. Se le permite filtrar por el número del convenio, por el municipio del convenio, por el importe mínimo y máximo, por un rango de fecha inicial a fecha final de los convenios, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de pienso.						
Excepciones						
Si el usuario no posee una conexión a internet, se le notificara que no posee una conexión al servidor principal						
Fecha	Estado		Comentario			
15/02/2021	Definido		Sin Comentario			
16/02/2021	Implementado		Sin Comentario			
17/02/2021	Hecho		Sin Comentario			

Tabla 14 StoryCard de Resumen Carnes

Resumen de Alimentos

El productor podrá observar un gráfico de pastel con las cantidades por tipo de pienso que le ha comprado a la empresa porcino de todos los convenios en función de los filtros aplicados. Se le permite filtrar por el número del convenio, por el municipio del convenio, por el importe mínimo y máximo, por un rango de fecha inicial a fecha final de los convenios, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de pienso. Ver figura 12

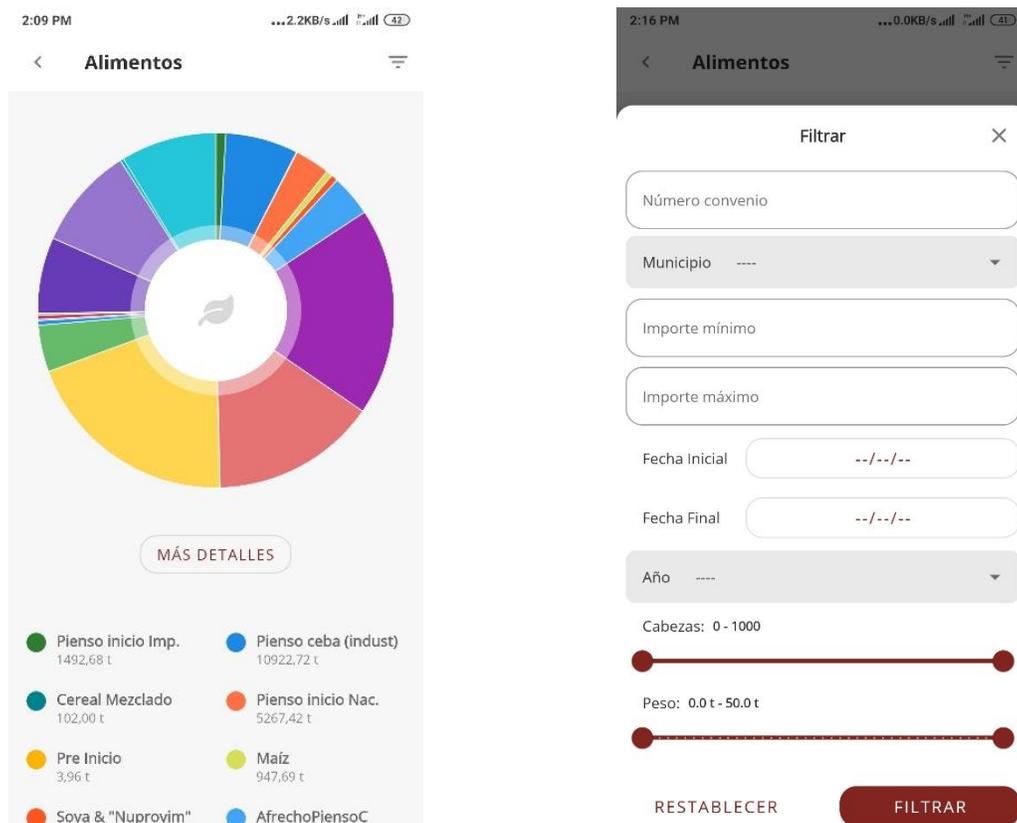


Figura 12 Resumen de Alimentos + Filtro

StoryCard de Resumen de Alimentos.

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
05	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	10	11	Baja Media Alta
Descripción						
El productor podrá observar un gráfico de pastel con las cantidades por tipo de pienso que le ha comprado a la empresa porcino de todos los convenios en función de los filtros aplicados. Se le permite filtrar por el número del convenio, por el municipio del convenio, por el importe mínimo y máximo, por un rango de fecha inicial a fecha final de los convenios, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de pienso.						
Excepciones						
Si el usuario no posee una conexión a internet, se le notificara que no posee una conexión al servidor principal						
Fecha	Estado		Comentario			
22/02/2021	Definido		Sin Comentario			
23/02/2021	Implementado		Sin Comentario			
25/02/2021	Hecho		Sin Comentario			

Tabla 15 StoryCard de Resumen de Alimentos

Consultar Carnes Vendidas

El productor podrá listar todas las carnes vendidas con la UEB Porcina en función de los filtros aplicados. Este puede filtrar la lista de carnes vendidas por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas. Ver figura 13.

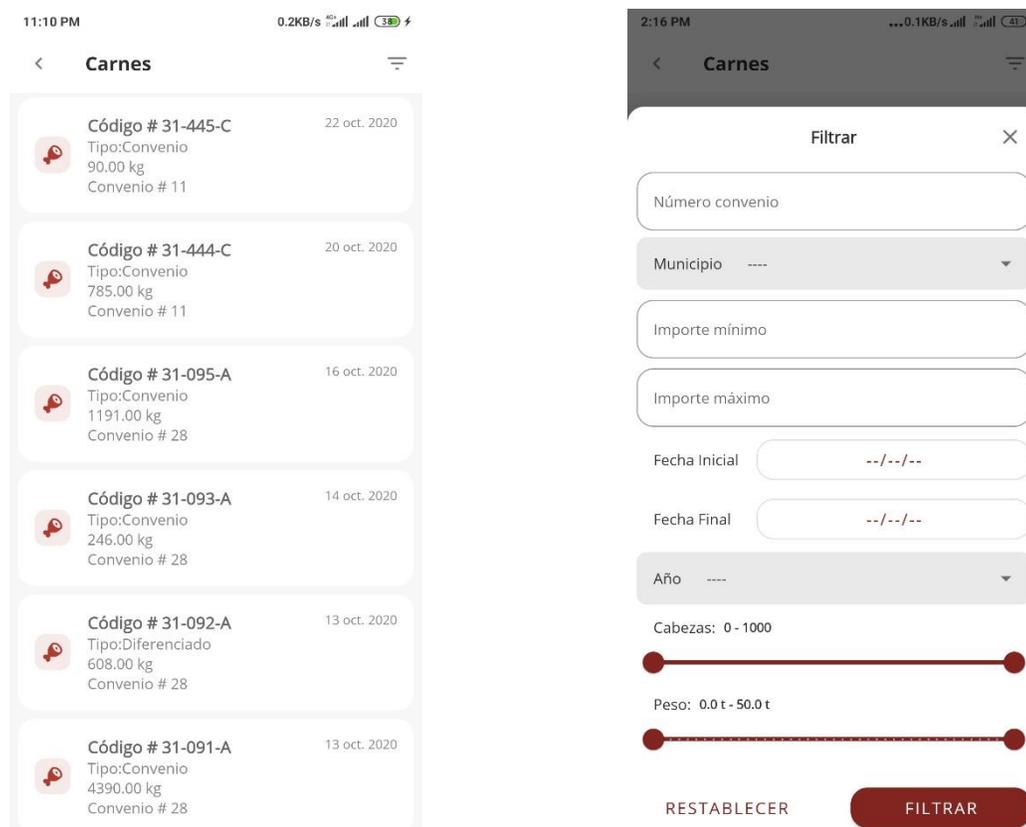


Figura 13 Consultar Carnes vendidas + Filtro

StoryCard de Consultar Carnes Vendidas

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
06	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	5	6	Baja Media Alta
Descripción						
El productor podrá listar todas las carnes vendidas con la UEB Porcina en función de los filtros aplicados. Este puede filtrar la lista de carnes vendidas por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas.						
Excepciones						
Si el usuario no posee una conexión a internet, se le notificara que no posee una conexión al servidor principal						
Fecha	Estado			Comentario		

27/02/2021	Definido	Sin Comentario
28/02/2021	Implementado	Sin Comentario
1/03/2021	Hecho	Sin Comentario

Tabla 16 StoryCard de Consultar Carnes Vendidas

Consultar Alimentos Comprados

El productor podrá listar todos los piensos comprados a la UEB Porcina en función de los filtros aplicados. Este puede filtrar la lista de los piensos comprados por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas. Ver figura 14.

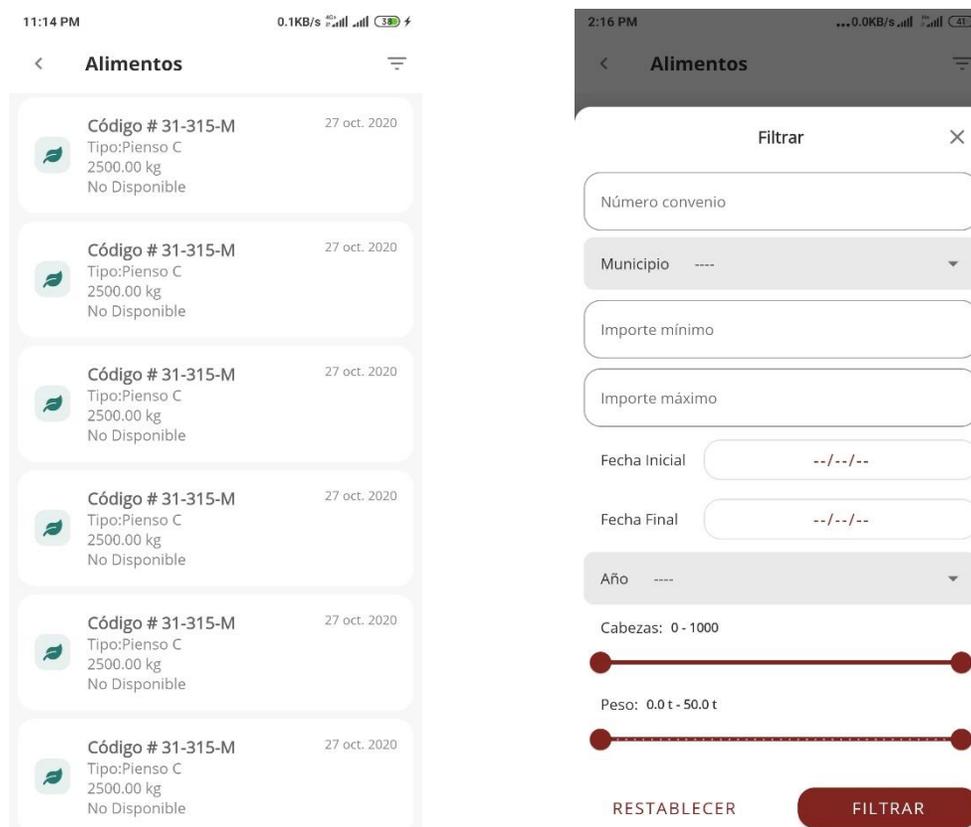


Figura 14 Consultar alimentos comprados + Filtros

StoryCard de Consultar Alimentos Comprados

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
07	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	5	6	Baja Media Alta
Descripción						
El productor podrá listar todos los piensos comprados a la UEB Porcina en función de los filtros aplicados. Este puede filtrar la lista de los piensos comprados por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas						
Excepciones						
Si el usuario no posee una conexión a internet, se le notificara que no posee una conexión al servidor principal						
Fecha	Estado		Comentario			
4/03/2021	Definido		Sin Comentario			
5/03/2021	Implementado		Sin Comentario			
7/03/2021	Hecho		Sin Comentario			

Tabla 17 StoryCard de Consultar Alimentos Comprados

Perfil

El usuario podrá actualizar su perfil, en caso de que el crea sea necesario modificar algún tipo de información que se haya registrado anteriormente ver figura 15.



CAMBIAR CONTRASEÑA

Nombre	Fernando	
Apellidos	cbnnggh	
Celular	55555555	
Carnet de Identidad	6699555545	
Género		

Figura 15 Perfil

StoryCard del Perfil

Número/ Id	Tipo	Dificultad		Esfuerzo		Prioridad
		Antes	Después	Estimado	Gastado	
08	Nuevo Fijo Moderado	Fácil Moderado Duro	Fácil Moderado Duro	8	10	Baja Media Alta
Descripción						
El usuario podrá actualizar su perfil, en caso de que el crea sea necesario modificar algún tipo de información que se haya registrado anteriormente.						
Excepciones						
Si el usuario no posee una conexión a internet, se le notificara que no posee una conexión al servidor principal						
Fecha	Estado		Comentario			
10/03/2021	Definido		Sin Comentario			
11/03/2021	Implementado		Sin Comentario			
13/03/2021	Hecho		Sin Comentario			

Tabla 18 StoryCard del Perfil

Estructura de los directorios

La aplicación queda configurada es una estructura de directorios en forma de árbol, cuya carpeta de mayor jerarquía es Porcino, directorio padre.

La figura 16 describe esta estructura mostrando los directorios más importantes.

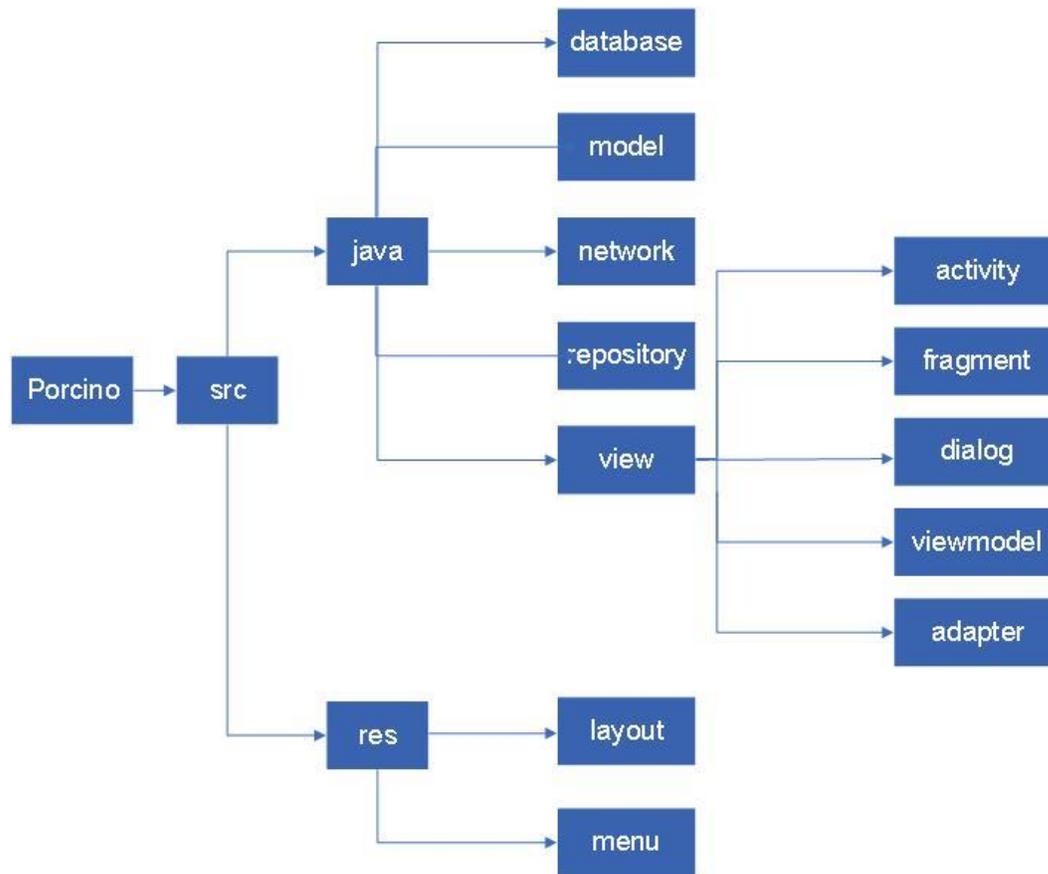


Figura 16 Estructura de los directorios

El subdirectorio **java** tiene todas las clases que dirigen el funcionamiento de la aplicación, es decir, la parte lógica. En esta se encuentran todo lo correspondiente a la base de datos interna de la aplicación que almacena los datos correspondientes a municipios y provincias, además las clases de las vistas, los modelos, entre otras.

El subdirectorio **res** contiene todos los recursos utilizados por la aplicación, ya sea imágenes, vectores, audios, textos, entre otros. En el cual se destacan el subdirectorio layout, el cual contiene los ficheros XML de las vistas de la aplicación, estas nos permitirán configurar las diferentes pantallas que

compondrán la interfaz de usuario de la aplicación. Y en el subdirectorío menú están los XML con los menús de cada actividad.

Servicios utilizados de la API Rest para consultar la BD de UEB porcina.

Los servicios utilizados de la API Rest ApiPorcino para consultar la base de datos de la UEB porcina son:

- Petición de tipo GET para listar todos los convenios del productor autenticado en la aplicación móvil según el filtro especificado:
<https://porcino-vc.xutil.cu/api/mis-convenios>.
- Petición de tipo GET para obtener el resumen de los piensos comprados por el productor a la empresa porcina según el filtro especificado:
<https://porcino-vc.xutil.cu/api/resumenes?tipo-resumen=comida>
- Petición de tipo GET para obtener el resumen de las carnes vendidas por el productor a la empresa porcina según el filtro especificado:
<https://porcino-vc.xutil.cu/api/resumenes?tipo-resumen=carnes>
- Petición de tipo GET para listar todas las carnes vendidas por el productor a la empresa porcina según el filtro especificado:
<https://porcino-vc.xutil.cu/api/compras>.
- Petición de tipo GET para listar todos los piensos comprados por el productor a la empresa porcina según el filtro especificado:
<https://porcino-vc.xutil.cu/api/piensos>
- Petición de tipo GET para obtener la información del perfil del productor en el sistema:
<https://porcino-vc.xutil.cu/api/perfil>

Diagrama de despliegue

La aplicación para móviles que se implementó, se despliega como aparece en el diagrama de despliegue de la figura 17, se conoce que accederá a los servicios Web API Rest existentes en la ApiPorcino, el que a su vez accederá a una base de datos operacional en ejecución en un servidor de la empresa porcina



Figura 17 Diagrama de despliegue

3.4 Pruebas del sistema

Pruebas de caja negra

Las pruebas de caja negra se centran principalmente en lo que se quiere de un módulo o sección específica de un software, o sea, es una manera de encontrar casos específicos de ese módulo que atiendan a su especificación. Son dedicadas a mirar en el exterior de lo que se prueba. Se enfoca solamente en la entrada y salida del sistema, sin preocuparse en tener conocimiento de la estructura interna del programa, (Milián, 2018).

Para cada funcionalidad del sistema se diseñan y realizan las pruebas de caja negra:

Algunos de los casos de este tipo de prueba realizada aparecen a continuación:

Caso de prueba de la Interfaz usuario del Login

Entrada de datos	Salidas del sistema	Mensaje de la Interfaz
No se introducen los datos del correo y contraseña del usuario	Error: Campos requeridos para autenticarse al sistema	 <p>The screenshot shows a mobile login screen with a dark red circular logo at the top. Below the logo are two input fields: 'Correo electrónico' and 'Contraseña'. Both fields have a red border and a red exclamation mark icon on the right, indicating an error. Below each field is the text 'Campo requerido'. At the bottom of the form is a dark red button labeled 'INICIAR SESIÓN'. Below the button is a link that says '¿No estás Registrado? CREAR CUENTA'. The top of the screen shows the time '4:53 PM' and network status '...36,0KB/s'.</p>

Correo:
ylbg@nauta.cu
Contraseña: 1234
(Incorrecta)

Error: La contraseña
debe tener como
mínimo 8 caracteres



<p>Correo: ylb@nauta.cu Contraseña: pepe1234 (Incorrecta)</p>	<p>Error: Correo y la contraseña especifica no son correctos</p>	 <p>The screenshot shows a mobile application login screen. At the top, the status bar displays '5:07 PM', signal strength, and battery level. The app's logo, a red circle with a white bird-like shape, is centered at the top. Below it are two input fields: 'Correo electrónico' containing 'ylb@nauta.cu' and 'Contraseña' with masked characters. A red 'INICIAR SESIÓN' button is positioned below the fields. A link for '¿No estás Registrado? CREAR CUENTA' is located further down. At the bottom, a red error message box contains a white 'X' icon and the text 'No se reconoce la combinación de usuario y contraseña'.</p>
---	--	---

Tabla 19 Caso de prueba de la Interfaz usuario del Login

Caso de prueba de la Interfaz de usuario del Registro de Usuario

Entrada de datos	Salidas del sistema	Mensaje de la Interfaz
No se introducen datos	Error: Campos requeridos para registrarse al sistema	 <p>The screenshot shows a mobile application interface for user registration. At the top, the time is 4:59 PM and the battery level is 34%. The title of the screen is 'Registro'. Below the title, there are several input fields, each with a red exclamation mark icon indicating an error. The fields are: 'Nombres' (0/50), 'Apellidos' (0/50), 'Email*' (0/30), 'Carnet de Identidad' (0/11), 'Contraseña' (0/50), 'Confirmar contraseña *' (0/50), 'Celular' (0/8), and 'Provincia' (ARTEMISA). The error messages are 'Campo requerido' for each of these fields.</p>

Introducir una contraseña pequeña

Error: La contraseña debe tener como mínimo 8 caracteres

5:00 PM

< Registro

Carnet de Identidad
97864646464 11/11

Contraseña
.... !
Su contraseña debe tener al menos 8 caracteres 4/50

Confirmar contraseña *
.... !
Su contraseña debe tener al menos 8 caracteres 4/50

Celular
0/8

Provincia
ARTEMISA ▼

Municipio
ALQUIZAR ▼

Género
Masculino ▼

REGISTRARSE

<p>Confirmar mal la contraseña</p>	<p>Error: La contraseña y confirmar contraseña deben coincidir</p>	 <p>The screenshot shows a mobile registration screen titled "Registro". At the top, the time is 5:00 PM and the battery is at 33%. The form includes a phone number field with "97864646464" and "11/11" characters. Below it is a "Contraseña" field with "....." and "8/50" characters. The "Confirmar contraseña *" field also has "....." and "9/50" characters, but it is highlighted with a red border and a red exclamation mark icon. Below this field, the text "Las contraseñas no son iguales" is displayed. Further down are fields for "Celular" (0/8), "Provincia" (ARTEMISA), "Municipio" (ALQUIZAR), and "Género" (Masculino). A red "REGISTRARSE" button is at the bottom.</p>
------------------------------------	--	---

Tabla 20 Caso de prueba de la Interfaz de usuario del Registro de Usuario

Caso de prueba en la Interfaz de Listar los Convenios

Acción	Salidas del sistema	Mensaje de la Interfaz
<p>Acceder a los convenios sin conexión a internet</p>	<p>Error: Error de que no se pudo establecer una conexión con de servidor de datos, o que el dispositivo no posee una conexión a internet</p>	 <p>5:15 PM ...0,0KB/s ... 33% 🔋</p> <p>< Convenios ≡</p> <p>Ups! No se encontraron convenios para mostrar</p> <p>Se a producido una error de red. ✖ Compruebe su conexión e inténtelo de nuevo.</p>

Tabla 21 Caso de prueba en la Interfaz de Listar los Convenios

Caso de prueba en la Interfaz de Perfil del Usuario

Acción	Salidas del sistema	Mensaje de la Interfaz
<p>Acceder a al perfil sin conexión a internet</p>	<p>Error: Error de red o que el dispositivo no posee una conexión a internet</p>	

Tabla 22 Caso de prueba en la Interfaz de Perfil del Usuario

Pruebas de compatibilidad

En las pruebas de compatibilidad se asegura que la aplicación móvil funcione correctamente, para esta prueba se seleccionaron varios dispositivos con diferente tamaño de pantalla, resolución y versión de Sistema operativo Android. Pueden ser de inspección manual o automatizada mediante la captura de pantallas que son posteriormente revisadas por el tester

La aplicación está diseñada para funcionar en cualquier smartphone o Tablet con Sistema operativo Android 4.4 o superior, por tanto, se realizaron pruebas en distintos dispositivos los cuales se muestran en l figura 18.

	<ul style="list-style-type: none"> - Motorola Moto G XT-1072 (2015) - CPU Qualcomm Snapdragon Quad-Core 1.2GHz - Memoria RAM de 1GB - Almacenamiento interno 8GB. - Pantalla táctil 5.0" 720x1280 HD 16:9 - Android 4.0.3 - Jellybean
	<ul style="list-style-type: none"> - Samsung Galaxy S6 SM-G920F - CPU Cortex-A53 Quad-Core 1.5GHz & Exynos 7420 Quad-Core 2.1GHz & Qualcomm Snapdragon Quad-Core 1.2GHz - Memoria RAM de 3GB - Almacenamiento interno 32GB. - Pantalla táctil 5.1" 1440x2560 QuadHD 16:9 - Android 7.1.1 - Nougat
	<ul style="list-style-type: none"> - Blu Dash J - CPU MediaTek Dual-Core 1.3GHz + ARM Mail 400 GPU - Memoria RAM de 256MB - Almacenamiento interno 512MB. - Pantalla táctil 4.0" 480x800, 4:3 - Android 4.4 - KitKat
	<ul style="list-style-type: none"> - Samsung P4 - CPU ARM Cortex-A9 Dual-Core 1.6ghz + Nvidia Tegra GPU - Memoria RAM De 768MB - Almacenamiento Interno 16GB. - Pantalla Táctil 10.0" 800x1280 HD 16:9 - Android 5.1.1 - Lollipop
	<ul style="list-style-type: none"> - Huawei T1-7.0 - CPU Spreadtrum Quad-Core 1.2GHz + GPU Mail - Memoria RAM de 1GB - Almacenamiento interno 8GB. - Pantalla táctil 7.0" 600x1024, 16:9 - Android 6.0 - Marshmallow

Figura 18 Dispositivos de Prueba

En los Smartphones probados se pudo apreciar como la app se ajusta de manera automática a las variaciones de tamaño de pantalla y densidad de pixeles, además al instalarse en distintos sistemas operativos la aplicación respondió como se esperaba ya que al instalarla en dispositivos con sistema operativo Android inferior a la versión 4.4 – KitKat, la aplicación no funcionó, ya que se implementó para versiones superiores o la 4.4 – KitKat.

En los Tablets la aplicación se comporta correctamente, variando la forma en que se visualiza el contenido, para ajustarse a pantallas con un mayor tamaño, se utilizaron layouts diferentes con el objetivo de aprovechar las ventajas de tener más espacio para mostrar información en dispositivos más grandes. También se probó en versiones distintas de Android para evaluar su comportamiento, el cual fue exitoso.

Conclusiones parciales.

Se presenta el diseño del sistema, a partir de la arquitectura general del mismo, esbozado en un diagrama de implementación, que se consideró útil para mostrar las interacciones entre la apk, el servidor de servicios Web y el servidor de bases de datos.

A partir de las fases de trabajo de la metodología Mobile-D se obtienen los artefactos de la aplicación. Se implementan las funcionalidades especificadas en los requisitos del sistema, a partir de la especificación de los StoryCard refinados en las iteraciones planificadas. Se muestran las diferentes actividades de la aplicación describiéndose las posibilidades que permiten a los usuarios, productores privados.

Se utilizan pruebas de caja negra para las funcionalidades generales del sistema que permiten obtener un correcto funcionamiento del mismo, así como las pruebas de compatibilidad para asegurar que la aplicación sea compatible para las versiones de Android superior a 4.4 así como a los diferentes tipos de dispositivos, celular o table.

CONCLUSIONES

A raíz del desarrollo del presente trabajo se puede arribar a las siguientes conclusiones:

- Se utiliza Mobil-D como metodología de software ágil para llevar a cabo el desarrollo del software, pues esta se da a conocer como un proceso de desarrollo extendido y con éxito para el desarrollo de aplicaciones móviles, utilizando las salidas fundamentales que proporciona.
- La aplicación móvil desarrollada permite acceder y consultar información de forma ágil y precisa por parte de los productores del sector privado para el seguimiento y control de sus convenios con la Empresa UEB Porcina de Villa Clara. La aplicación accede a la base de datos de la empresa, a través del uso de los servicios web accediendo a la API Rest que fue creada por otros desarrolladores para lograr estos fines.
- Se concibió una interfaz intuitiva y sencilla que permite una interacción viable, rápida, de fácil manejo por parte de los usuarios a través de una aplicación móvil desarrollada según sus exigencias.
- Se realizaron pruebas de las funcionalidades más importantes del sistema para corregir errores con pruebas de caja negra y las pruebas de compatibilidad garantizan asegurar que la aplicación se ejecuta en una gama amplia de dispositivos Android, a partir de las versiones 4.4.

REFERENCIAS BIBLIOGRÁFICAS.

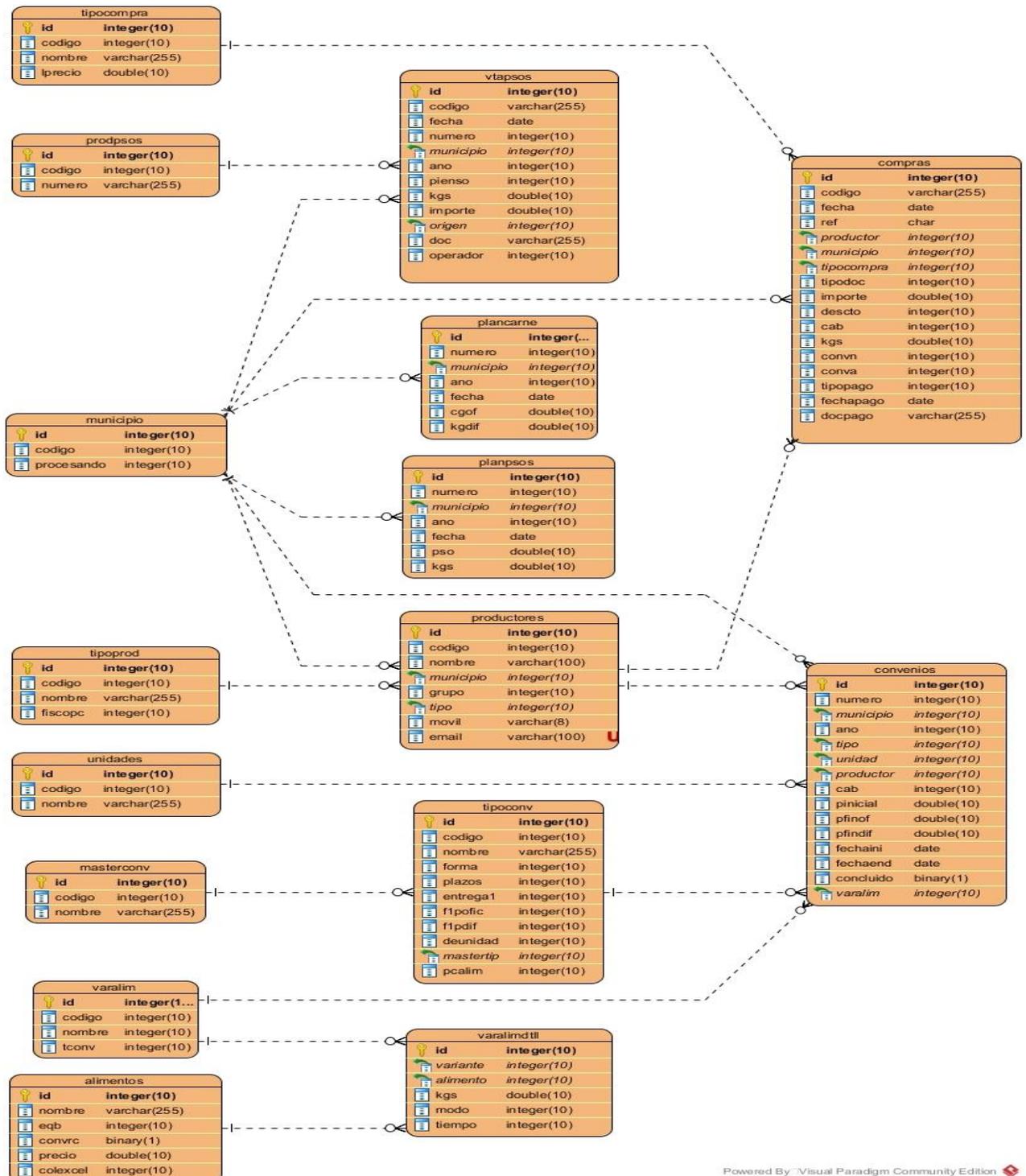
- Adam Warterski, P. (2013). Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone. Revista de Tecnología | Journal Technology, 12 número (4), 181–183. <https://doi.org/10.1145/2048237.2157457>
- Agenjo-Bullon, X., Hernández-Carrascal, F., 2010. La biblioteca virtual: función y planteamiento.
- Báez, M. (2012). INTRODUCCIÓN - ¿QUÉ ES ANDROID?
- Báez, M. et al. (2012) 'Introducción a android', EME Madrid, España, p. 121.
- Borrego, Á. (2012). Introducción a Android.
- Castillo, D. M. B. B. (2014) 'REGLAS DE NEGOCIO DESDE LA PERSPECTIVA DE LOS DATOS EN BASES DE DATOS RELACIONALE'.
- Correa Sánchez, W. (2018). Comercialización de una Aplicación Móvil para Dispositivos Android/iOS, 1–123.
- Cuenca, C., & Marjorie, J. (2015). "Desarrollo de una aplicación móvil Android para la búsqueda de plazas disponibles en un parqueadero."
- de Cuba, P. C., 2017. Lineamientos de la política económica y social del partido y la revolución para el período 2016-2021. Tech. rep.
- Delia, L. N., Galdamez, N., Thomas, P., Pesado, P. M., 2013. Un análisis experimental de tipo de aplicaciones para dispositivos móviles. In: Congreso Argentino de Ciencias de la Computación (CACIC). Vol. 18.
- Farman, J. (2013) Mobile interface theory: Embodied space and locative media. Routledge.
- Fernando Diego, Y. H., & Vinicio Jaime, C. P. (2014). Desarrollo de una aplicación para el mercado inmobiliario. Retrieved from.
- Gutiérrez, N. (2018) 'nataliagutierrez9835ita_ DEFINICION CASO DE USO, ACTORES Y ROLES'.
- Hassan Montero, Y., Martín Fernández, F. J., 2005. La experiencia del usuario. No solo usabilidad (4).
- Hernández Yáñez, L. (2014). Fundamentos de la programación.

- Jover, J. N., Arriete, L. F. M., 2018. La política de ciencia, tecnología e innovación en la actualización del modelo económico cubano: evaluación y propuestas/science, technology and innovation policy in the updating of the cuban economic model: evaluations and proposals.
- Lie, H. W., Bos, B., Lilley, C., Jacobs, I., 2005. Cascading style sheets. Pearson India.
- Lorenzo, G. F. (2016c) 'Conf 5 - Captura de requisitos'.
- Lujan-Mora, S., 2002. Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario.
- Luna, R. M. C., Vaca, K. H. A. and Vázquez, D. A. P. (2017) 'Observaciones acerca de los dispositivos móviles', Dominio de las Ciencias. Polo de Capacitación, Investigación y Publicación (POCAIP), 3(4), pp. 89–103.
- Marichal, D., Tutores, R., Rodr, Y., & Rueda, D. T. (2019). Prototipo de Aplicación para Móviles para la "Enciclopedia Cubana 1000 Preguntas 1000 Respuestas." Universidad Central Marta Abreu de las Villas.
- Molina, M. A. (2017). Propuesta de diseño de una aplicación móvil.
- Morales, M. (2013). Aplicaciones móviles nativas orientadas a servicios y recursos de bibliotecas universitarias.
- Msaffirio (2017) 'Reglas de Negocio – Business Rules – Tecnologías de la Información y Procesos de Negocios (BPM)'.
- Peñuela Epalza, M., Paternina Del Rio, J., Moreno Santiago, D., Camacho Pérez, L., Acosta Barrios, L., De León De León, L., 2014. El uso de los smartphones y las relaciones interpersonales de los jóvenes universitarios en la ciudad de barranquilla (Colombia). Salud uninorte 30 (3).
- Puig, M. (2012). "Creación de una aplicación, programada en Java, para smartphones basados en el sistema operativo Android para un portal turístico."
- ROLDAN, M. M., THOMPSON, D. N., 2018. Aplicaciones móviles nativas orientadas a servicios y recursos de bibliotecas universitarias. In: XVI Congreso Edutec Costa Rica.

- Sánchez Perez, P. A. (2015). Aplicación Android para simular el comportamiento de métodos de Investigación de Operaciones. Universidad Central “Marta Abreu” de Las Villas.
- Santiago, R., et al. (2015). Mobile learning: nuevas realidades en el aula, Editorial Océano.
- Solanky, J., Patil, K., Patel, G., 2016. Resemblance of phonegap and titanium for mobile application development. International Journal of Computer Applications 144 (10).
- Su, L. M., Garc, F. R., & Garc, A. (2019). Management System for Criminal Consultation. Universidad Central “Marta Abreu” de las Villas.
- Treude, C., Aniche, M., 2018. Where does google find api documentation? In: Proceedings of the 2nd International Workshop on API Usage and Evolution. ACM, pp. 19–22. Vol. 9 número 2, julio de 2012, pp 129-145, pp. 129–145.
- Villazán Olivarez, F. J. (2010). Informática I.
- Milián, Y. R. (2018) ‘Pruebas de caja negra - Globe Testing’.
- Muñoz Muñoz, C. A. (2020). *APLICACIÓN DE LA METODOLOGÍA MOBILE-D EN EL DESARROLLO DE UNA APP MÓVIL PARA GESTIONAR CITAS MÉDICAS DEL CENTRO JEL RIOBAMBA.*
- Su, L. M., Garc, F. R., & Garc, A. (2019). *Management System for Criminal Consultation.* Universidad Central Marta Abreu de las Villas.

ANEXOS

Anexo 1: Modelo físico de la Base de Datos de la Empresa UEB de Porcino de Villa Clara



Anexo 2: Manual de Usuario



MANUAL DE USUARIO APLICACIÓN MÓVIL PORCINO

Elaborado Por: Yeralys Li Balmaseda
Versión: 1.0

Reconocimiento y apertura de la aplicación.

La aplicación móvil una vez instalada en su dispositivo móvil se visualiza con el nombre de Porcino y un icono representado de la siguiente forma (Figura 20):



Figura 19 Reconocimiento de la aplicación

Splash Screen

Al tocar el icono de la aplicación, se abrirá el Splash Screen de la aplicación móvil, esto como indicación al usuario que la aplicación está en proceso de carga para utilizarla. (Figura 21)

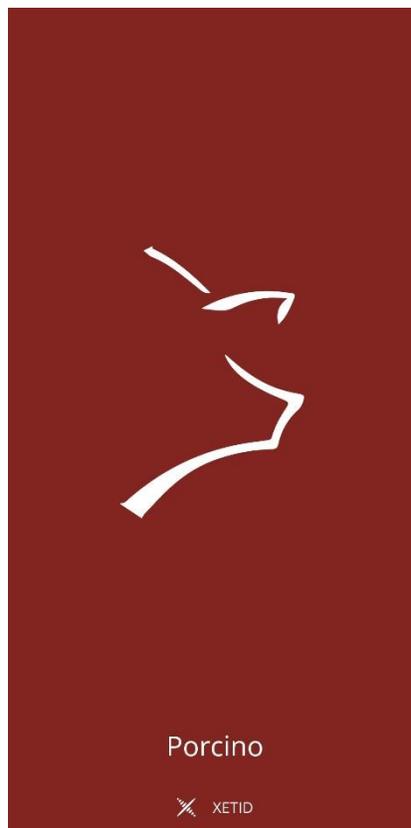


Figura 20 Splash Screen

Inicio de sesión

Una vez cargada nuestra aplicación aparecerá la pantalla de inicio de sesión en el cual se procede a llenar los distintos campos solicitados, una vez completados estos requisitos, se puede proceder al botón de inicio de sesión, pulsamos dicho botón y nos dirigimos al menú principal de la aplicación, en caso de ser un nuevo usuario en el sistema tendrá que elegir la opción Crear Cuenta. (Figura 22)



Figura 21 Login

Crear Cuenta.

Para utilizar la aplicación móvil es necesario que el usuario ingresé los datos solicitados como: (nombres, apellidos, correo electrónico carnet de identidad, número de teléfono, provincia, municipio, género y su contraseña que contenga al menos 8 dígitos entre letras y número). Una vez ingresados los datos el usuario pulsara el botón registrarse. (Figura 23)

11:55 PM ...0.0KB/s 4G LTE 26%

< **Registro**

Nombres 0/50

Apellidos 0/50

Email* 0/30

Carnet de Identidad 0/11

Contraseña 0/50

Confirmar contraseña * 0/50

Celular 0/8

Provincia
ARTEMISA



Figura 22 Crear Cuenta

Menú principal.

El usuario una vez autenticado en la aplicación móvil procederá a navegar libremente en el menú principal de la aplicación y seleccionar cualquier opción que el considere oportuno utilizar figura 24. El usuario puede ir a la opción **Convenios** para consultar todos los convenios que este posee con la Empresa Porcina, la opción **Carnes** para consultar el resumen y toda la información de las carnes vendidas a la empresa, también tiene la opción **Alimentos** para consultar el resumen y toda la información de los piensos comprados a la empresa. Otras de las opciones se encuentran la de Perfil, donde el usuario puede consultar y modificar todos los datos del mismo en el sistema de Porcino.

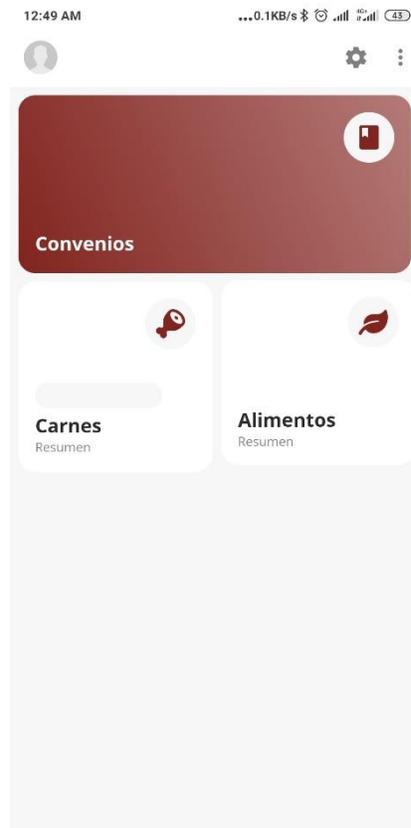


Figura 23 Menú Principal

Perfil

El usuario podrá actualizar su información de usuario a cómo el considere necesario modificar, una vez que el usuario modifique la información de perfil, seguidamente le sale un mensaje, que le indica que guarde su información para no perderla, en caso de que quiera realizar otra opción Figura 25.



Figura 24 Perfil

Convenios

Al pulsar en la opción **Convenios**, el usuario puede observar todos los convenios que este tiene con la Empresa UEB Porcino Villa Clara en función de los filtros aplicados. Este puede filtrar la lista de convenios por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas. Ver figura 26.

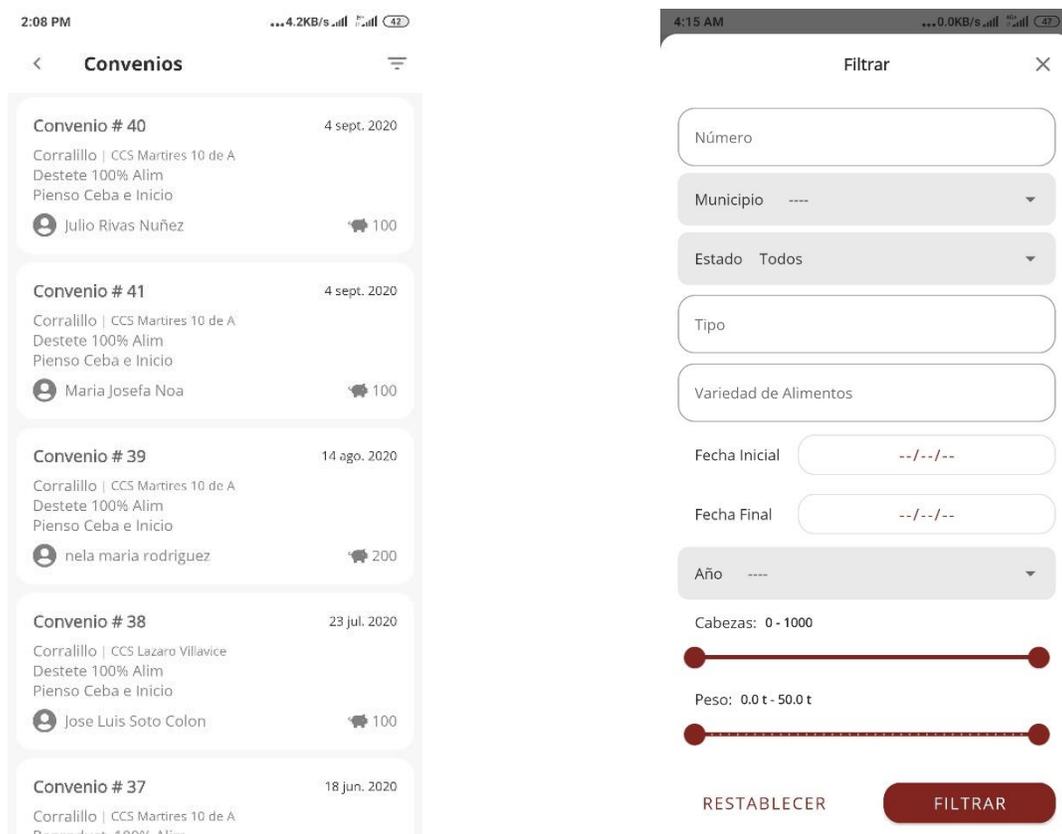


Figura 25 Convenios + Filtro

Carnes

El usuario en esta opción puede observar un gráfico de pastel, con la cantidad de toneladas, la cantidad de cerdos y el tipo de las carnes vendidas del productor a la Empresa Porcina en función de los filtros aplicados (Ver figura 27). Este puede filtrar la lista de carnes vendidas por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniadas y por el peso en toneladas de carnes a vender conveniadas. Este al pulsar la opción **Mas Detalles**, puede observar la lista de las ventas de carnes a la empresa porcina.

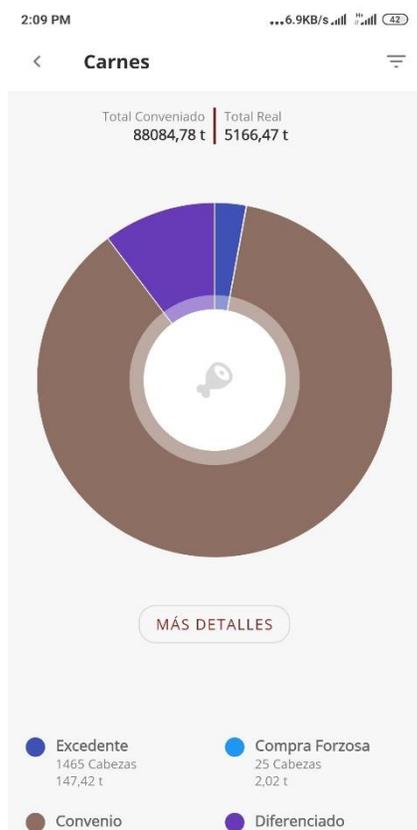


Figura 26 Resumen de Carnes Filtro

Carnes/Más Detalles:

Aquí el usuario puede observar una lista de las carnes vendidas a la empresa porcina en función de los filtros aplicados, este al pulsar uno de los elementos puede ver con más detalles la información de la venta (Figura 28).

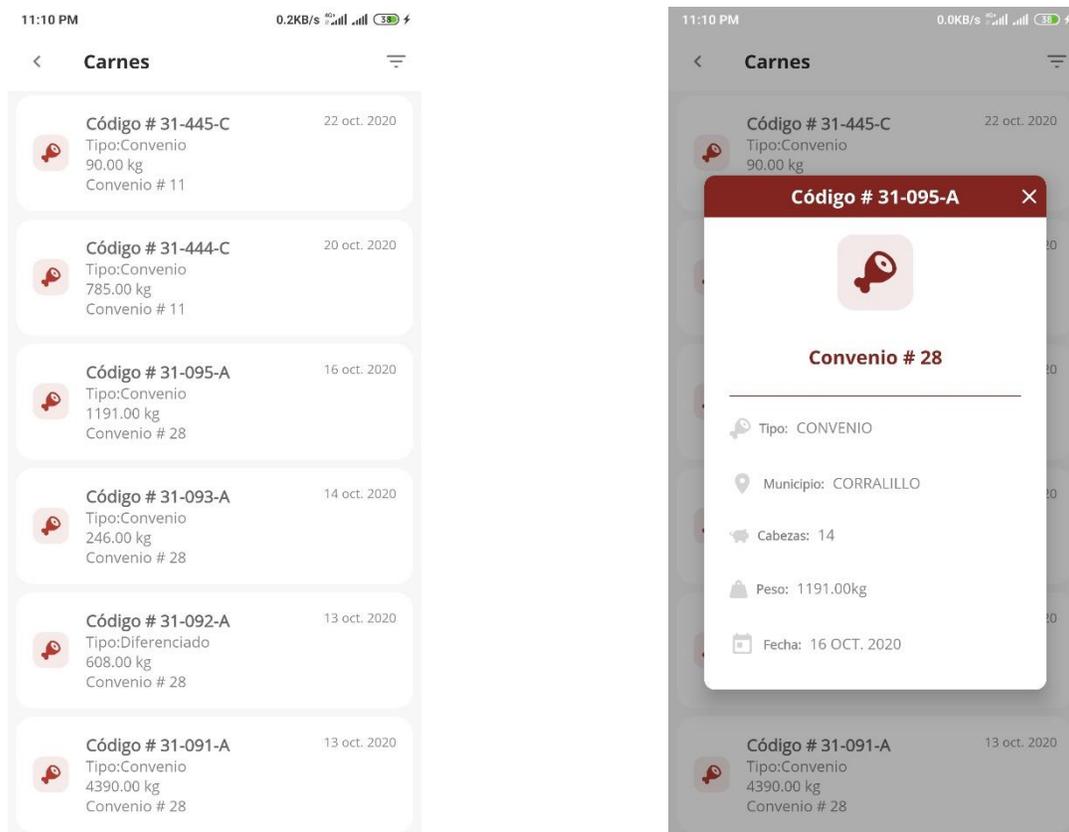


Figura 27 Lista de las carnes vendidas + Detalles

Alimentos

El usuario en esta opción puede observar un gráfico de pastel, con la cantidad de toneladas, y el tipo de pienso que le compro el productor a la Empresa Porcina en función de los filtros aplicados (Ver figura 29). Este puede filtrar la lista de carnes vendidas por el número de convenio, por el municipio, por el estado del convenio si está concluido o no el mismo, por el tipo de convenio, por la variedad de alimentos, por el rango de fecha inicial y fecha final de los convenios, por el año del convenio, por la cantidad de cabezas de cerdos conveniados y por el peso en toneladas de carnes a vender conveniadas. Este al pulsar la opción **Mas Detalles**, puede observar la lista de las compras de pienso a la empresa porcina.



Figura 28 Resumen de Alimentos + Filtro

Alimentos/Más Detalles:

Aquí el usuario puede observar una lista de todos los piensos comprados a la empresa porcina en función de los filtros aplicados, este al pulsar uno de los elementos puede ver con más detalles la información de la compra del pienso (Figura 30).

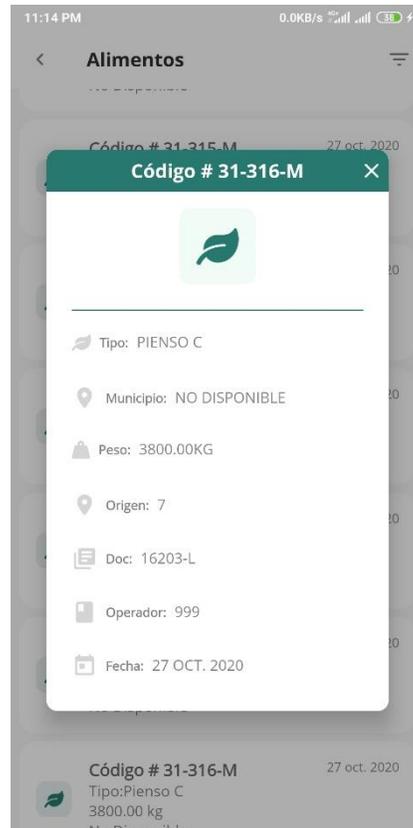
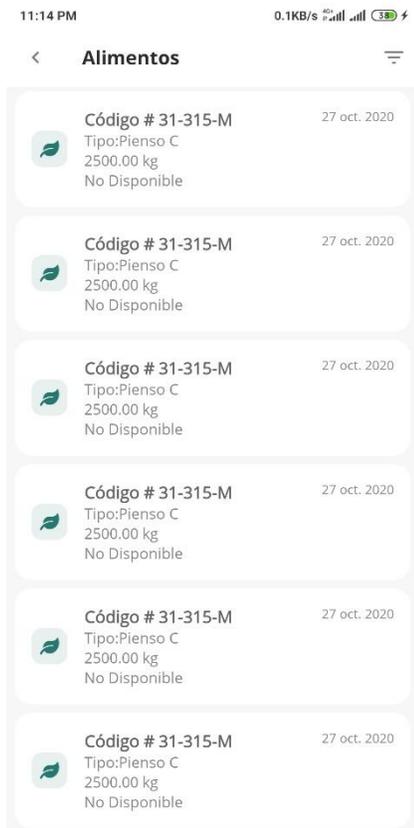


Figura 29 Lista de los piensos + Detalles