



Reporte Final de Estadía

José Octavio Sánchez Contreras

SIOF



VERACRUZ
Gobierno del Estado



SEV
ESTADO DE VERACRUZ

VER Educación
SECRETARÍA DE EDUCACIÓN

SEP
SECRETARÍA DE
EDUCACIÓN PÚBLICA



DET
Dirección de Educación
Tecnológica del
Estado de Veracruz



Aplicación móvil para Sistema Integral Oxifuel (SIOF)

Sánchez Contreras José Octavio

Universidad Tecnológica del Centro de Veracruz

Ingeniería en Tecnologías de la Información

Cuitláhuac, Veracruz

Abril 2017

Tabla de contenido

1. Introducción	1
2. Descripción de la problemática	2
3. Objetivos	3
3.1 Objetivo general	3
3.2 Objetivos específicos.....	3
4. Alcance y limitaciones	4
4.1 Alcance.....	4
4.2 Limitaciones	4
5. Justificación	5
6. Marco teórico	6
6.1 Desarrollo de Software.....	6
6.2 Fases del proceso de desarrollo de software	6
6.2.1 Análisis de requisitos.....	6
6.2.2 Diseño y arquitectura	7
6.2.3 Programación.....	7
6.2.4 Pruebas.....	7
6.2.5 Documentación	8
6.2.6 Mantenimiento.....	8
6.3 Metodologías de desarrollo de software	8
6.4 Modelos del Proceso de Desarrollo Software	9
7. Metodología	10
7.1 Modelo de Procesos para la industria de Software.....	11
7.2 Administración de Proyectos Específicos (APE)	12
7.3 Desarrollo y Mantenimiento de Software (DMS)	12
8. Desarrollo	14
8.1 Planteamiento del problema.....	14
8.2 Solución propuesta.....	14
8.2.1 Manejo de servidores.....	14

8.2.2 Manejo de base de datos	15
8.2.3 Desarrollo de back-end.....	15
8.2.4 Desarrollo del front-end	17
8.3 Aplicación de la metodología	18
Conclusiones.....	19
Recomendaciones.....	20
Bibliografía.....	21
Anexos	22
Anexo A - Especificación de requisitos de software	22
1. Introducción.....	22
2. Funcionales.....	22
3. Interfaz de usuario	24
4. Interfaces con otro software y hardware	25
5. Confiabilidad	25
6. Eficiencia	26
7. Mantenimiento	26
8. Portabilidad	26
9. Reusabilidad	26
10. Legales y reglamentarios.....	26
Anexo B - Cronograma de actividades.....	27

Resumen

El presente documento contiene la información que describe el desarrollo de un producto de software para la empresa, la cual se dedica a la venta de combustible con etanol.

Se mostrará solo la parte del desarrollo de la aplicación para el sistema operativo para móvil iOS que es parte de un software más grande denominado Sistema Integral de Oxifuel, cuyo propósito principal es ofrecer al usuario final diversos servicios como la compra de bonos de carbono por medio de una tarjeta llamada Oxiskey.

Es necesario explicar algunas de las tecnologías y metodologías que se utilizarán para tener una mejor comprensión del alcance del proyecto y sus beneficios, además de hacer entendible el proceso que se llevará a cabo y sus limitaciones debidas a las propias restricciones. Se optó por utilizar Cordova, HTML5, CSS, JavaScript, AngularJS y Ionic como tecnologías del lado del cliente (aplicación para iOS) y Apache http Server, PostgreSQL, PHP y Laravel como tecnologías del lado del servidor (computadora remota a la que se accede por medio de internet) por sus características de robustez, compatibilidad con diversos sistemas operativos y navegadores de internet con sus diversas versiones.

También se verá que durante el proceso de desarrollo se cambió de tecnologías del lado del cliente a programación con SWIFT con la herramienta Xcode dentro de un entorno macOS 10.11 por necesidades que surgieron durante dicho proceso.

Todo el proceso está contemplado para un periodo de 15 semanas máximo y mediante el uso del modelo de procesado de software llamado MoProSoft.

Abstract

This document contains the information that describes the development of a software product for the company, which is dedicated to the sale of fuel with ethanol.

This document shows only the part of the development of the application for iOS operating system that is part of a larger software called "Sistema Integral de Oxifuel", whose main purpose is to offer to the end user various services such as the purchase of carbon points through of a card called Oxikey.

It is necessary to explain some of the technologies and methodologies that will be used to have a better understanding of the scope of the project and its benefits, besides making understandable the process that will be carried out and its limitations due to the restrictions themselves. It was chosen to use Cordova, HTML5, CSS, JavaScript, AngularJS and Ionic as client side technologies (iOS application) and Apache http Server, PostgreSQL, PHP and Laravel as server side technologies (remote computer accessed through internet) for its characteristics of robustness, compatibility with various operating systems and internet browsers with its various versions.

It will also be seen that during the development process we switched from client-side technologies to programming with SWIFT with the XCode tool within a macOS 10.11 environment because of needs that arose during that process.

The whole process is contemplated for a maximum period of 15 weeks and by the software process model called MoProSoft.

1. Introducción

El presente documento contiene la información que describe el desarrollo de una aplicación en iOS dentro del centro de desarrollo de software (CEDESOF) para una empresa ubicada en la ciudad de Orizaba, la cual se dedica a la venta de combustible con etanol. El desarrollo de la aplicación para iOS que en lo posterior dentro del propio documento se le denominará como aplicación móvil es parte de un proyecto más grande llamado Sistema Integral Oxifuel (abreviado SIOF), que tiene como propósito principal ofrecer al cliente diversos servicios como la compra de bonos de carbono por medio de una tarjeta.

El proyecto que se desarrollará a lo largo del presente documento mostrará cómo se puede solventar la necesidad de la empresa de publicitar sus servicios y recabar fondos para las fundaciones relacionadas con la misma, además de ofrecer un valor agregado a sus clientes con la compra y recarga de la tarjeta desde su smartphone.

2. Descripción de la problemática

La empresa cliente, se dedica a la venta de combustible con etanol, es de reciente incursión en el mercado del combustible y no tiene un sistema que integre todos sus procesos internos y externos con sus asociados. En investigaciones y diálogos previos con la empresa se identificó que tienen la necesidad de publicitar su negocio y de facilitar a sus clientes la compra de sus productos y a su vez también facilitar el proceso de donación de sus clientes a las fundaciones asociadas.

3. Objetivos

3.1 Objetivo general

Desarrollar una aplicación móvil que permita a sus actuales y potenciales clientes conocer su propuesta de negocio, comprar y recargar Oxikey y realizar donaciones a las fundaciones asociadas desde cualquier iPhone con internet.

3.2 Objetivos específicos

- Recopilar la información necesaria de las reglas de negocio para definir la funcionalidad que deberá cumplir la aplicación móvil.
- Realizar la maquetación de las vistas de la aplicación móvil para su aprobación por parte del cliente.
- Establecer los datos necesarios de la base de datos del proyecto SIOF para poder realizar el diseño de la interfaz de conexión con la aplicación móvil.
- Crear un web service que sirva de modelo de datos centralizado para los dispositivos móviles.
- Crear las vistas para las maquetaciones aprobadas por el cliente.
- Integrar todas las librerías necesarias para poder realizar el pago desde la aplicación móvil con PayPal.
- Hacer los controladores que permitan la comunicación de las vistas con los modelos.
- Realizar las pruebas correspondientes de las funcionalidades requeridas por el software.

4. Alcance y limitaciones

4.1 Alcance

La aplicación móvil mostrará información acerca de la empresa cliente y asociados, permitirá al cliente comprar y recargar su Oxikey con PayPal desde cualquier iPhone con internet que tenga la aplicación instalada y permitirá al cliente realizar donaciones con PayPal a cualquiera de las fundaciones asociadas a la empresa. El usuario de la aplicación móvil será capaz de actualizar su información personal registrada y consultar todos los movimientos de compras, recargas y donaciones realizados. La aplicación móvil será capaz de mostrar la ubicación de las estaciones de los franquiciatarios registrados con la empresa cliente.

4.2 Limitaciones

La aplicación estará probada para su correcto funcionamiento para las versiones 8 y 9 de iOS, siendo estas las más comunes a la fecha de entrega del producto. No está contemplado otro método de pago además de PayPal para la compra y recarga de Oxikey, y también para cualquiera de las donaciones. Algunos de los errores que puedan suceder durante el proceso de pago con PayPal son debidos a limitaciones propias de las librerías de PayPal, por lo que no son corregibles mas que por el desarrollador de dichas librerías. El correcto funcionamiento de la localización de las estaciones de servicio de los franquiciatarios está sujeta a las características de software del iPhone y servicios de internet del usuario final.

5. Justificación

El desarrollar una aplicación móvil para la empresa cliente, que permita a sus actuales y potenciales clientes conocer su propuesta de negocio, realizar sus pagos de Oxikey y donaciones a las fundaciones asociadas permitirá dar a conocer de una manera fácil y atractiva la información que al momento deseen compartir o difundir entre sus clientes, además de ser una manera más cómoda y personal de poder realizar sus pagos, ya que no necesitan rentar alguna computadora, excluyéndose los riesgos concernientes al uso de computadoras de dominio público, además de permitirle realizar cualquiera de sus pagos desde cualquier ubicación donde cuente con conexión a internet. Además, la utilización de la sección de la aplicación que localiza las estaciones de servicios registradas, facilitará al usuario encontrar la estación que se encuentre más cercana de su ubicación actual, generando un beneficio para el usuario final tanto como para la empresa.

6. Marco teórico

6.1 Desarrollo de Software

Cuando se va desarrollar un software intervienen muchas personas como lo es el cliente, el que tiene una necesidad en su empresa y desea que sea solucionada, para esto existe el Analista de Sistema que es el encargado de hacerle llegar todos los requisitos y necesidades que tiene el cliente a los programadores que son las personas encargadas de realizar la codificación y diseño del sistema para después probarlo y que lo instale al cliente. Es así como intervienen varias personas ya que una sola persona podría perder de vista algo necesario, entre más personas estén involucradas es más probable cubrir con todos los requisitos del sistema.

6.2 Fases del proceso de desarrollo de software

6.2.1 Análisis de requisitos

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento ERS, Especificación de Requerimientos del software, cuya estructura puede venir definida por varios estándares, tales como CMMI. Asimismo, se define un diagrama Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software. La captura, análisis y especificación de requisitos (incluso pruebas de ellos), forman una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Aunque aún no está formalizada, ya se habla de la Ingeniería de Requisitos. La IEEE 830-1998 normaliza la

creación de las Especificaciones de Requisitos de Software (Software Requirements Specification).

6.2.2 Diseño y arquitectura

Se refiere a determinar cómo funcionará de forma general sin entrar en detalles. Consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc. Se definen los casos de uso para cubrir las funciones que realizará el sistema, y se transforman las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercano a la programación orientada a objetos.

6.2.3 Programación

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga. La complejidad y la duración de esta etapa está íntimamente ligada al o a los lenguajes de programación utilizados.

6.2.4 Pruebas

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas. En general hay dos grandes formas de organizar un área de pruebas, la primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera pueda entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área

de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en qué condiciones puede fallar una aplicación y así poner atención en detalles que el personal inexperto no consideraría.

6.2.5 Documentación

Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por la modelación (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, etc; todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

6.2.6 Mantenimiento

Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores, o bugs. La mayor parte consiste en extender el sistema para hacer nuevas cosas. De manera similar, alrededor de 2/3 de toda la Ingeniería civil, Arquitectura y trabajo de construcción es dar mantenimiento.

6.3 Metodologías de desarrollo de software

Todo desarrollo de software implica cierta complejidad al momento de realizarlo, pero si se aplica alguna metodología de desarrollo de software que ya tenga años de utilización y una amplia documentación, se disminuyen los riesgos implicados, de lo contrario, se podrían obtener clientes insatisfechos con el resultado y desarrolladores aún más.

Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Con relación a los proyectos que se desarrollan con mayor

envergadura, hay si toma sentido el basarse en una metodología de desarrollo y se empieza a buscar cuál sería la más apropiada para dicho caso. En ciertas ocasiones no se encuentra la más adecuada y se termina por hacer un diseño de metodología propio, por supuesto no está mal siempre y cuando sirva para alcanzar el objetivo.

Muchas veces se realiza el diseño del software de manera rígida, tal como el cliente lo solicitó, pues si se alteran cosas que no se habían previsto, atrasan el proyecto y en algunas situaciones ocasionan no llegar a cumplir con el cambio solicitado, esto conlleva malestar en el cliente puesto que no ha sido tomado en cuenta su pedido; para evitar estos incidentes se debe llegar a un acuerdo formal con el cliente al inicio del proyecto de manera que no perjudique el desarrollo del mismo.

Muchas veces los usuarios finales se dan cuenta que dejaron de mencionar algunas cosas y lo manifiestan en la etapa inicial del proyecto cuando se le muestra el prototipo del mismo. Algunas Metodologías conocidas son:

- La metodología RUP es la más adaptable para proyectos de largo plazo.
- La metodología XP en cambio, se recomienda para proyectos de corto plazo.
- La metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología.

Se puede decir además que es importante antes de elegir la metodología que se debe usar para implementar el software, determinar el alcance que tendrá y luego de allí ver cuál es la que más se acomoda al desarrollo de la aplicación.

6.4 Modelos del Proceso de Desarrollo Software

No existe consenso sobre cuál es el mejor modelo del proceso software. Distintos equipos de desarrollo pueden utilizar diferentes modelos de proceso software para producir el mismo tipo de sistema software. Sin embargo, algunos modelos son más apropiados para producir ciertos tipos de sistemas, de forma que si no se

utiliza un modelo adecuado puede ocurrir que el sistema software resultante sea de menor calidad.

El reparto de costes entre las distintas fases del proceso de desarrollo es difícil de determinar dado los distintos modelos de proceso existentes. El proceso de diseño software debería tener en cuenta la posterior evolución del sistema. Las características deseables de un proceso de desarrollo software son:

- **Claridad:** El proceso de desarrollo es claro cuando se entiende con facilidad.
- **Visibilidad:** Un proceso de desarrollo es visible cuando sus actividades producen resultados claros identificables externamente.
- **Facilidad de soporte:** Exige disponer de herramientas CASE (Computer-Aided Software Engineering) que den soporte a todas o algunas de las actividades del proceso de desarrollo.
- **Fiabilidad:** Un proceso de desarrollo es fiable cuando es capaz de detectar posibles errores.
- **Facilidad de mantenimiento:** Requiere capacidad para incorporar nuevos requisitos o modificar alguno o algunos de los existentes.
- **Rapidez:** Un proceso software es rápido cuando se puede obtener, a partir de la especificación, una implementación del sistema en un tiempo reducido.

7. Metodología

MoProSoft, está definido como un Modelo de Procesos para la Industria de Software, es el modelo de calidad seleccionado para ser parte de los lineamientos de la empresa para la cual de desarrolla y está definido como un conjunto de recomendaciones y mejores prácticas para los procesos de desarrollo de software, sirve de base a la Norma Mexicana para la Industria de Desarrollo y Mantenimiento de Software.

En este capítulo se presenta una descripción del modelo, además de algunos de los procesos que corresponden al mismo y que se tomaron como referencia para la adaptación al desarrollo de este proyecto.

7.1 Modelo de Procesos para la industria de Software

La definición del modelo de procesos MoProSoft tiene tres categorías de procesos: Alta Dirección, Gerencia y Operación que reflejan la estructura de una organización.

La categoría de **Alta Dirección (DIR)** contiene los lineamientos para los procesos de la Categoría de Gerencia y se retroalimenta con la información generada por ellos en apoyo a la estrategia de la organización.

La categoría de **Gerencia (GER)** está integrada por los procesos de Gestión de Procesos, Gestión de Proyectos y Gestión de Recursos, éste último está constituido por los subprocesos de Recursos Humanos y Ambiente de Trabajo, Bienes, Servicios e Infraestructura y Conocimiento de la Organización.

La categoría de **Operación (OPE)** está integrada por los procesos de Administración de Proyectos Específicos y de Desarrollo y Mantenimiento de Software (Oktaba, et al., 2005).

El modelo de procesos MoProSoft está dirigido a las empresas o áreas internas dedicadas al desarrollo y/o mantenimiento de software. Las organizaciones que no cuenten con procesos establecidos, pueden usar el modelo ajustándolo de acuerdo a sus necesidades. Mientras que las organizaciones que ya tienen procesos establecidos, pueden usarlo como punto de referencia para identificar los elementos que les hace falta cubrir (Oktaba, et al., 2005).

La categoría Operación aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Dentro de esta categoría se encuentran los procesos Administración de Proyectos Específicos y Desarrollo y Mantenimiento de Software, referidos en el documento de MoProSoft v1.3 – Por niveles de capacidad de procesos, con las claves OPE.1 y OPE.2, respectivamente.

El propósito de la Administración de Proyectos Específicos es establecer y llevar a cabo sistemáticamente las actividades que permitan cumplir con los objetivos de un proyecto en tiempo y costo esperados. Mientras que el del proceso Desarrollo y Mantenimiento de Software es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de productos de software nuevos o modificados cumpliendo con los requisitos especificados.

Para la realización del proyecto expuesto en este documento, se utiliza una adaptación de los procesos OPE.1-Administración de Proyectos Específicos y OPE.2-Desarrollo y Mantenimiento de Software en conjunto como metodología de desarrollo.

7.2 Administración de Proyectos Específicos (APE)

La Administración de Proyectos Específicos aplica conocimientos, habilidades, técnicas y herramientas, a cada una de las siguientes actividades del proyecto:

- **Planificación:** Conjunto de actividades cuya finalidad es obtener y mantener el plan del proyecto y el plan de desarrollo que regirán al proyecto específico, con base en la descripción del proyecto, generada en el proceso Gestión de Proyectos.
- **Realización:** Consiste en llevar a cabo las actividades del plan del proyecto, de acuerdo a las tareas especificadas.
- **Evaluación y control:** Se trata de asegurar que se cumplan los objetivos del proyecto. Se supervisa y evalúa el proceso para identificar desviaciones y realizar acciones correctivas, cuando sea necesario.
- **Cierre:** Consiste en entregar los productos de acuerdo a un protocolo de entrega y dar por concluido el ciclo o proyecto. Como resultado se tiene el documento de aceptación del cliente (Oktaba, et al., 2005).

7.3 Desarrollo y Mantenimiento de Software (DMS)

El proceso de Desarrollo y Mantenimiento de Software se compone de uno o más ciclos de desarrollo. Cada ciclo está compuesto de las siguientes fases.

- **Inicio:** Revisión del plan de desarrollo por los miembros del equipo de trabajo para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización.
- **Requerimientos:** Conjunto de actividades cuya finalidad es obtener la documentación de la especificación de requerimientos y plan de pruebas

del sistema, para conseguir un entendimiento común entre el cliente y el proyecto.

- **Análisis y Diseño:** Serie de actividades en las cuales se analizan los requerimientos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado se obtiene la documentación correspondiente al Análisis y Diseño y Plan de Pruebas de Integración.
- **Construcción:** Conjunto de actividades para producir componentes de software que correspondan al Análisis y Diseño, así como a la realización de pruebas unitarias.
- Como resultado se obtienen los componentes de software probados.
- **Integración y Pruebas:** Conjunto de actividades para integrar y probar los componentes de software, basados en los planes de pruebas de integración y de sistema, con la finalidad de obtener el software que satisfaga los requerimientos especificados. Como resultado se obtiene el producto de software probado y documentado.
- **Cierre:** Integración final de la configuración de software generada en las fases para su entrega. Identificación y documentación de las lecciones aprendidas. Generación del reporte de mediciones y sugerencias de mejora (Oktaba, et al., 2005).

8. Desarrollo

El contenido de este capítulo comprende la descripción de los elementos que se definieron como parte del proyecto y el manejo de estos para desarrollar un software que dé solución a la problemática propuesta.

8.1 Planteamiento del problema

Para el desarrollo de una aplicación móvil se deben tomar en cuenta varios aspectos que intervienen para dar funcionalidad dinámica a la misma. Los principales aspectos que intervienen en la misma son:

- Manejo de servidores.
- Manejo de base de datos.
- Desarrollo back-end.
- Desarrollo front-end.
- Diseño gráfico.

Cada uno de estos aspectos pueden ser abordados desde diferentes perspectivas y con diferentes tecnologías, a continuación, se detallará el proceso que se llevó para cumplir con todos los aspectos anteriores y dar solución a la problemática definida anteriormente.

8.2 Solución propuesta

A continuación, se detallará en cada una de las diferentes tecnologías utilizadas para cumplir con el desarrollo de cada uno de los aspectos necesarios mencionados en el planteamiento del problema.

8.2.1 Manejo de servidores

Un servidor como su nombre lo sugiere, se encarga de cubrir servicios o necesidades, ya sean de almacenamiento, recuperación o procesamiento de información, un servidor para poder desempeñar su función requiere de dos partes

fundamentales, un equipo de cómputo (**hardware**) y un software que le indique al mismo como deben ser usados sus recursos para cubrir las necesidades (servicios) según la función que este desempeñe. Estos pueden ser instalados dentro de la infraestructura de la misma empresa o ser adquiridos por medio de una empresa externa que le rente el equipo necesario.

La instalación o contratación del servidor para la aplicación móvil fue escogido con anterioridad por la empresa en conjunto con los desarrolladores de la plataforma web.

8.2.2 Manejo de base de datos

El manejo de la base de datos requiere en primera instancia de la compra o renta de un servidor de bases de datos, todo depende de las necesidades de la compañía. En segunda instancia es necesario de un software que ayude con el manejo de la información conocido como **DBMS** (siglas de las palabras de origen inglés Database Management System), este a su vez se especializa sobre algún lenguaje como por ejemplo SQL, que es el más popular al momento de redacción de este documento.

Dado que el aspecto del servidor fue cubierto por la empresa interesada en la aplicación móvil y el DBMS fue elegido con anterioridad durante el proceso de desarrollo del sitio web, solo fue necesario analizar la base de datos existente (**PostgreSQL**) para ver si lo que tenían era suficiente o sería necesario agregar algo más.

8.2.3 Desarrollo de back-end

Este apartado se refiere al desarrollo de la parte del sistema que el usuario no ve, pero que es esencial para el funcionamiento correcto de la aplicación móvil y que normalmente se enfoca en proporcionar servicios.

En primer lugar es necesario un servidor de aplicaciones, que comprende la instalación de un software que hará la función de escuchar todas las peticiones de los distintos dispositivos móviles y dar respuesta a los mismos, cabe mencionar

que dicho servidor de aplicaciones puede estar montado sobre el mismo hardware (equipo de cómputo) que el servidor de base de datos aunque lo más óptimo podría ser tenerlos en equipos de procesamiento distintos ofreciendo un mejor rendimiento del hardware, además de que los datos de la aplicación podrían estar distribuidos a través de varios servidores de bases de datos para mejorar aún más el rendimiento y disponibilidad de la información.

8.2.3.1 Servidor de aplicaciones

El servidor de aplicaciones seleccionado fue “**Apache http server**”, pero este a su vez necesita de otros aditamentos que ayuden a procesar la petición y dar una respuesta satisfactoria a la aplicación móvil, dichos aditamentos se mencionan a continuación.

8.2.3.2 Laravel

Se escogió un **framework** conocido como Laravel por ser un conjunto de librerías escritas en **PHP** que ofrecen una manera limpia de programar y también ofrecen varias medidas de seguridad ya integradas, logrando un grado de robustez del software en un periodo corto de tiempo para los conocedores del framework.

Para la comunicación con la base de datos se utilizó una librería de Laravel conocida como “Query Builder” que permite realizar consultas de una manera muy similar al lenguaje SQL, dado que la forma en la que estaba estructurada la base de datos no era propicia para utilizar “Eloquent” sin reestructurar todo desde el principio. Eloquent es una librería más reciente de Laravel que simplifica el diseño del modelo de datos y ofrece un grado mayor de abstracción de los datos.

8.2.3.3 JSON Web Tokens (JWT)

Un JWT es una forma de representar y transportar de forma segura información a través de la red por medio del formato JSON, el estándar se encuentra documentado en el sitio web de la IETF (siglas de las palabras de origen inglés Internet Engineering Task Force) con la denominación **rfc7519**, este comprende

varios tecnicismos que no tienen una traducción literal al español por lo que si el lector desea ampliar sus conocimientos sobre este estándar puede consultar el texto original en inglés que se encuentra en la URL <https://tools.ietf.org/html/rfc7519>.

El motivo por el que se decidió usar JWT fue porque otros métodos existentes para autenticar usuarios presentaban deficiencias o incompatibilidades. El uso de JWT permite que un usuario se autentique en un servidor y si la aplicación móvil llegara perder comunicación con el servidor, el dispositivo móvil puede autenticarse con otro servidor usando el mismo token proporcionado por el servidor anterior, además de que, el administrador del sistema puede permitir si lo desea que varios usuarios se puedan conectar usando el mismo token, ventajas que presenta frente a otras tecnologías.

Por ser este un estándar, era necesario seleccionar una implementación de este que se adaptara al lenguaje de programación y framework que se seleccionaron, se decidió utilizar un proyecto integrable con Laravel llamado “tymon/jwt-auth” que se puede encontrar en el sitio web de GitHub. Este se escogió de entre varios por su grado de compatibilidad y porcentaje de cumplimiento del estándar.

8.2.4 Desarrollo del front-end

El front-end normalmente es todo aquello que el usuario puede ver, es el medio por el cual el usuario interactúa con el sistema, en este caso el front-end es la interfaz gráfica que ofrece la aplicación móvil que permite al usuario final consultar o realizar operaciones la información registrada y realizar pagos.

Para realizar la aplicación móvil hay que considerar que hay varios tipos de dispositivos móviles y que cada uno tiene un sistema operativo diferente que requiere de técnicas y lenguajes diferentes de programación. Considerando que los dispositivos de mayor uso al momento de redactar este documento son los de tipo Smartphone, existen dos muy populares, aquellos con sistema operativo **Android** y los que tienen sistema operativo **iOS**.

Ya delimitadas las plataformas a tratar queda elegir la tecnología a utilizar para desarrollar la aplicación como tal, se consideró en primera instancia usar

ionic porque se codifica con lenguajes de programación para la web y el código generado lo transforma en código nativo para Android y iOS, sin embargo, la aplicación generada es ligeramente más lenta que la que fue realizada en código nativo para cada plataforma.

Al final por cuestiones de rendimiento y seguridad del código se tomó como solución definitiva realizarlo en código nativo para cada plataforma, para realizar la aplicación para Android se utilizó **Java** y para la aplicación para iOS se utilizó **Swift**, cabe mencionar que Swift no es la única opción, también existe el lenguaje Objective-C que lleva varios años utilizándose para desarrollar aplicaciones para la familia de productos de Apple, pero Swift es un lenguaje creado por Apple para solucionar deficiencias encontradas durante el paso de los años en Objective-C, además de ofrecer nuevas características de lenguajes de programación modernos.

8.3 Aplicación de la metodología

La aplicación resultante del desarrollo del proyecto que se expone en este documento es propiedad de la empresa cliente, por lo que no todos los artefactos resultantes pueden ser anexados al documento, por mencionar entre algunos el código fuente, mockups, pruebas unitarias, pruebas de integración, manuales de usuario, mantenimiento y operación. Se anexan la especificación de requisitos de software y cronograma de actividades como anexos en el documento.

Conclusiones

El desarrollo de la aplicación para iOS en Swift se realizó de forma programática por ser menor el impacto al momento del cambio de versiones, pero se observó que había mucha documentación para la implementación en Objective-C y por medio de Swift con el “Interface Builder” que se trabaja por medio de “Storyboards”. La documentación para realizar el código en Swift de forma programática fue muy escasa durante el desarrollo del proyecto por lo que varias investigaciones tuvieron que hacerse en Objective-C para después traducirlo a Swift. La documentación para Swift todavía estaba complementándose por la empresa Apple al momento de realizar el proyecto, se espera que en un futuro exista una mayor documentación del lenguaje volviéndolo una excelente opción para programar para iOS.

Recomendaciones

El desarrollo de la aplicación no contemplaba guardar información de forma local en el dispositivo móvil, en caso de algún error después de realizar un pago con PayPal se manda un correo electrónico al administrador del sistema para dar soporte al usuario, pero utilizando CoreData en Swift se podrían almacenar los IDs de los pagos realizados con PayPal para después completar automáticamente el proceso de registro del pago sin necesidad de que intervenga directamente un administrador.

Bibliografía

- ARTUR BORONAT, J. I., JOSÉ Á. CARSÍ, ISIDRO RAMOS, ABEL GÓMEZ. Del método formal a la aplicación industrial en Gestión de Modelos: Maude aplicado a Eclipse Modeling Framework1, 2003.
- Bernd Bruegge & Allen H. Dutoit. Object-Oriented Software Engineering, Prentice Hall, Pag. 11.
- Castro, Díaz-Balart, Fidel: CIENCIA, INNOVACIÓN Y FUTURO. Ediciones especiales. Instituto Cubano del Libro, La Habana. 2001.
- Campderrich Falgueras, Benet (2002): Ingeniería de software. Barcelona: Editorial UOC, 2002. 320 páginas.
- Franquet, R.: COMUNICAR EN LA SOCIEDAD DE LA INFORMACIÓN. Universidad Autónoma de Barcelona. 2005
- Ingeniería de Software Código de Ética y Práctica Profesional. SEERI, East Tennessee State University. 1999.
- Ingeniería de software (sexta edición), Ian Sommerville. Addison Wesley. Sitio en inglés.
- Ojalvo, V. y otros: LA COMUNICACIÓN EDUCATIVA. Universidad de la Habana. En formato digital.
- ONET Code Connector - Software Developers, Systems Software - 15-1133.00. Onetcodeconnector.org.
- Pasquali, A.: COMPRENDER LA COMUNICACIÓN. Caracas: Monte Ávila Editores. 1979.
- Presman, Roger, 2002. Ingeniería de Software: un enfoque práctico, Sexta edición, McGraw.Hill/Interamericana de España, 824 páginas. pág. 39, 53-54, 67-72.
- Software Development Manager Position Description. interfacing.com.
- Oktaba, H. (2005). *Modelo de Procesos para la Industria de Software, MoProSoft*. México.

Anexos

En la siguiente sección se presentan los apéndices correspondientes a salidas o entregables de determinadas actividades del desarrollo del proyecto.

- **Anexo A.** Especificación de requisitos de software.
- **Anexo B.** Cronograma de actividades.

Anexo A - Especificación de requisitos de software

ENTREGABLE	CÓDIGO		
Documento de ERS	ERS-v1.2		
ANEXO	FECHA		
Anexo A	26	01	2017

1. Introducción

En el presente documento se da a conocer la Especificación de Requisitos de Software (ERS) para el proyecto, el documento está basado en el estándar proporcionado por el modelo MoProSoft.

1.1 Objetivo

Determinar todas las funcionalidades y restricciones que el proyecto tendrá para su construcción de manera clara y concisa, con la finalidad de conocer a fondo dicha información, concentrando los puntos de información más relevantes en el documento, este servirá de base para el buen desarrollo del sistema y cumplirá una parte en la garantía de la entera satisfacción de las partes colaboradoras en el proyecto.

2. Funcionales

A continuación, se describen las actividades que se podrán realizar en el sistema y la información que utilizará para poder llevar a cabo dichas tareas. Se maneja una nomenclatura con el propósito de identificar los requisitos por medio de una clave. La clave está compuesta por los siguientes elementos:

- REF = Indica que se trata de un requisito funcional.

- 00 = Donde los ceros indican el lugar de un número de dos cifras consecutivo, comenzando con el 01.
- YYY = Indica la primera o primeras letra de cada palabra del enunciado del requisito, solamente se usan tres letras para componer este elemento de la clave.

Cada elemento es separado del otro por un guion, y la clave final es formada por la combinación consecutiva de los elementos, de la siguiente manera: REF-00-YYY.

2.1 Funciones del sistema

El sistema deberá realizar las siguientes tareas:

- REF-01-CTO. Compra de tarjeta digital Oxiskey con PayPal: sirve de medio para guardar los bonos de carbono que el usuario dispone para gastar en el combustible. La forma de pago de combustible es por medio bonos de carbono, ya sea que el usuario final los compre en la estación correspondiente o que el usuario los compre con los bonos de carbono que disponga en su tarjeta.
- REF-02-RBC. Recarga de bonos de carbono: muestra cuántos bonos de carbono puede comprar con el dinero que el usuario desee pagar o calcular el monto de dinero que necesitaría pagar para obtener los bonos de carbono deseados. El proceso de pago se realiza con PayPal.
- REF-03-PIN. Pantalla informativa: Muestra información acerca de la empresa y sus productos.
- REF-04-GES. Geolocalización de estaciones de servicio: Permite localizar geográficamente mediante un mapa digital las estaciones de servicio registradas por la empresa. Además, centra la pantalla en la ubicación actual del usuario del dispositivo móvil, mostrando a su vez, la estación más cercana en la sección visible del mapa.
- REF-05-SCU. Suscripción a Canacintra USA: Permite al usuario de la aplicación suscribirse a Canacintra USA, para poder hacer uso de los beneficios que esta ofrece.

- REF-06-DF1. Donaciones a la fundación I: ayuda a realizar el pago por concepto de donación a la fundación. El pago es por medio de la plataforma de PayPal, haciendo más confiable el proceso.
- REF-07-IF1. Información de la fundación I: sirve de medio para entender mejor los principios y camino que dirigen a la fundación.
- REF-08-DF2. Donaciones a la fundación II: ayuda a realizar el pago por concepto de donación a la fundación por medio de la plataforma de PayPal, haciendo más seguro el proceso.
- REF-09-IF2. Información de la fundación II: sirve para orientar sobre los principios y acciones que dirigen a la fundación.
- REF-10-PUA. Perfil del usuario activo: muestra los datos personales del usuario que está utilizando la aplicación, además, permite actualizar la información que aparece en su perfil, así también su contraseña.
- REF-11-CPR. Consulta de pagos realizados: es una opción del menú que permite visualizar todos los pagos realizados en los diversos conceptos de pago que tiene la aplicación, para que el usuario pueda checar su estado de cuentas.
- REF-12-CSA. Cierre de la sesión activa: destruye el token que autentifica al usuario con el servidor y lo redirige a la pantalla de inicio de sesión.
- REF-13-PIS. Pantalla de inicio de sesión: Aparece al momento de iniciar la aplicación o siempre que se trate de regresar de un estado inactivo y no se tenga un token vigente.

3. Interfaz de usuario

La interfaz de usuario estará constituida por un menú que ofrece distintas opciones, de acuerdo a la funcionalidad y que permitirá que el usuario comprenda, de forma fácil, el funcionamiento del sistema poniendo las herramientas necesarias a su alcance. La estructura del menú de la aplicación es la siguiente:

- Tarjeta electrónica
 - Compra
 - Recarga
- Informativa de la empresa

- Geolocalización
- Suscripciones
- Fundación I
 - Donaciones
 - Información
- Fundación II
 - Donaciones
 - Información
- Perfil
 - Actualizar
 - Cambiar contraseña
- Consultas
- Salir

4. Interfaces con otro software y hardware

Respecto a la interacción con otro hardware, se requerirá que el GPS este activado para que la aplicación pueda acceder a la localización del dispositivo. Además, la aplicación hace uso de las librerías de PayPal que realizan una conexión a los servidores de PayPal para autenticar al comprador y llevar a cabo el proceso de pago.

5. Confiabilidad

Este aspecto es determinado por otras capacidades de acuerdo al contexto de uso del producto software. Las capacidades que influyen sobre el nivel de confiabilidad del sistema son:

- **Efectividad:** Se ha determinado que el sistema cumplirá con las funciones que se han definido para el mismo.
- **Seguridad de uso:** El sistema minimizará el nivel de riesgo del cliente mediante el uso de servicios profesionales de externos que disminuyan el riesgo de que puede suceder algún percance en el proceso de pago.

6. Eficiencia

Este punto se cumplirá debido a las prestaciones que brindará el software, relativas a la cantidad de recursos empleados en consideración al tiempo de uso. Los aspectos o capacidades que influyen para considerar el nivel de eficiencia del producto son:

- **Comportamiento temporal:** Los tiempos de respuesta y proceso del sistema serán adecuados bajo condiciones determinadas de uso y recursos.
- **Utilización de recursos:** Se adecuará la cantidad y tipo de recursos cuando el software realiza sus funciones bajo condiciones determinadas.

7. Mantenimiento

Se requiere que el producto software pueda ser modificado en un futuro, esto incluye correcciones, mejoras o adaptación del mismo a cambios en el entorno, requisitos y/o especificaciones.

8. Portabilidad

Se ha definido que el software no cumplirá con el criterio de instalabilidad, correspondiente al punto portabilidad, esto es porque la aplicación será desarrollada en código nativo para iOS para asegurar el mejor rendimiento y seguridad posibles para esa plataforma.

9. Reusabilidad

Dentro de los requisitos establecidos, se definió que el producto software creado pueda volver a ser usado en otro proyecto relacionado.

10. Legales y reglamentarios

Respecto al código fuente del producto software, se requiere que sea de distribución restringida, limitada y controlada por la organización para la cual se desarrolla, esto para que forme parte de futuros proyectos que se desarrollen.

Anexo B - Cronograma de actividades

ENTREGABLE	Cronograma		
Documento de ERS	ERS-v1.2		
ANEXO	FECHA		
Anexo B	26	01	2017

Propuesta del proyecto SIOF														
Actividad	Semanas (2017)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Diseño de la interfaz de usuarios	P													
	R													
Instalación de librerías	P													
	R													
Instalación de CocoaPods	P													
	R													
Instalación de PayPal Mobile	P													
	R													
Configuración de PayPal	P													
	R													
Configuración de JWT (web service)	P													
	R													
Instalación de Laravel	P													
	R													
Configuración de Laravel	P													
	R													
Instalación de tymon/jwt-auth	P													
	R													
Configuración de JWT	P													
	R													
Creación de rutas	P													
	R													
Creación de JwtController	P													
	R													
Programación de tarjeta electrónica - Compra	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Creación CardController (web service)	P													
	R													
Programación de tarjeta electrónica - Recarga	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Modificación CardController (web service)	P													
	R													
Programación de Fundación I	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Creación FoundationController (web service)	P													
	R													

Propuesta del proyecto SIOF														
Actividad	Semanas (2017)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Programación de Fundación II	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Modificación FoundationController (web service)	P													
	R													
Programación de Geolocalización	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Creación StationController (web service)	P													
	R													
Programación de Suscripciones	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Creación SubscriptionController (web service)	P													
	R													
Programación de consultas (Oxikey, donaciones y Canacintra)	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Modificacion CardController (web service)	P													
	R													
Modificacion FoundationController (web service)	P													
	R													
Modificacion SubscriptionController (web service)	P													
	R													
Programación de Perfil	P													
	R													
Creación vista (movil)	P													
	R													
Creación controlador (movil)	P													
	R													
Creación ClientController (web service)	P													
	R													
Distribución de la aplicación	P													
	R													