

**DESARROLLO DE UN SERVICIO DE PAGOS
MEDIANTE APLICACIONES MÓVILES
MULTIPLATAFORMA**

SYNERGY GLOBAL BUSINESS, C.A

Firma: _____

Br. Javier Antonio Ortuño Ramírez

C.I.: 23.662.632

Tutor Industrial: Ing. Alexander Ramírez

Tutor Académico: Ing. José Pirrone

FECHA Y LUGAR: Caracas, abril de 2016.

Resumen

La demanda de soluciones que involucran ambientes tecnológicos como las aplicaciones móviles aumenta día a día. Actualmente el desarrollo de estas aplicaciones móviles consume mucho tiempo debido a que se deben hacer desarrollos específicos por plataforma. Sin embargo, ya es posible desarrollar dichas aplicaciones mediante tecnología web, sin sacrificar significativamente el desempeño de la aplicación.

Es por ello que lo relacionado con los servicios de pagos de manera presencial conocidos como TDC (Tarjetas de Crédito) y TDD (Tarjetas de Débito), es posible ofrecer una alternativa de pago mediante aplicaciones móviles multiplataforma.

Este proyecto de pasantía consiste en el diseño y desarrollo de un servicio de pagos a través de aplicaciones móviles multiplataforma que mediante códigos QR (Quick Response Code) permiten almacenar datos en una matriz de puntos y entendibles por lectores ópticos y dispositivos de capturas de imágenes como la cámara fotográfica de los teléfonos inteligentes o tabletas, dando la posibilidad a una empresa de generar una información de cobro utilizando códigos QR y a un cliente leer dicho código para realizar el pago mediante los instrumentos que posea (Teléfonos inteligentes o tabletas).

Índice General

| | | |
|---------|---|----|
| 1. | SECCIÓN INTRODUCTORIA..... | 1 |
| 1.1. | Introducción | 1 |
| 1.2. | Descripción de la empresa..... | 2 |
| 1.3. | Antecedentes de la investigación, Marco teórico | 4 |
| 1.3.1. | HTML | 4 |
| 1.3.2. | HTML5 | 5 |
| 1.3.3. | CCS | 5 |
| 1.3.4. | JavaScript..... | 6 |
| 1.3.5. | Framework | 6 |
| 1.3.6. | Modelo-Vista-Controlador..... | 7 |
| 1.3.7. | AngularJS..... | 7 |
| 1.3.8. | Ionic | 8 |
| 1.3.9. | Cordova..... | 8 |
| 1.3.10. | QR code (Quick Response Code) | 8 |
| 1.3.11. | Kiraso..... | 9 |
| 1.3.12. | Tecnología web | 9 |
| 1.4. | Objetivos del estudio | 9 |
| 1.4.1. | Objetivo General | 9 |
| 1.4.2. | Objetivos Específicos:..... | 9 |
| 1.5. | Alcances y premisas | 10 |
| 1.6. | Metodología empleada | 11 |
| 1.6.1. | FASE I - Planteamiento del problema..... | 11 |
| 1.6.2. | FASE II - Investigación inicial | 11 |
| 1.6.3. | FASE III - Diagnóstico de la situación actual | 12 |
| 1.6.4. | FASE IV - Diseño de propuestas | 12 |
| 2. | SECCIÓN CENTRAL | 13 |
| 2.1. | Descripción detallada de la situación actual y del problema propuesto | 13 |
| 2.2. | Evaluación de la situación mediante la presentación y el análisis de datos..... | 14 |
| 2.3. | Investigación y planteamiento de posibles soluciones | 14 |
| 2.4. | Diseño de propuestas y selección de la mejor opción | 15 |
| 3. | SECCIÓN FINAL..... | 22 |
| 3.1. | Conclusiones | 22 |

| | | |
|------|-----------------------|----|
| 3.2. | Recomendaciones | 22 |
| 3.3. | Bibliografía | 23 |
| 3.4. | Anexo A | 24 |

Índice de Figuras

| | |
|--|----|
| Figura 1. Organigrama de la empresa Synergy-GB. Fuente: Documentos de la empresa..... | 2 |
| Figura 2. Estructura básica de un documento web utilizando HTML. Fuente: (W3schools, 2016) | 5 |
| Figura 3. Mockups en Balsmiq de la aplicación móvil de la empresa. Se observan las pantallas de Bienvenida, Principal y Configurar. Fuente: Elaboración Propia y propiedad de Synergy-GB | 16 |
| Figura 4. Mockups en Balsmiq de la aplicación móvil de la empresa. Se observan las pantallas de Configuración, Cobrar y Consulta de Cobros. Fuente: Elaboración propia y propiedad de Synergy-GB .. | 16 |
| Figura 5. Mockups en Balsmiq de la aplicación móvil del cliente. Se observan las pantallas de Bienvenida, Principal y Lectura de QR. Fuente: Elaboración Propia y propiedad de Synergy-GB | 17 |
| Figura 6. Mockups en Balsmiq de la aplicación móvil del cliente. Se observan las pantallas de Pagar, Resumen de pago y Consulta de Pagos. Fuente: Elaboración Propia y propiedad de Synergy-GB..... | 18 |
| Figura 7. Aplicación móvil de la empresa, mostrando pantalla de Bienvenida, Principal y Cobrar. Fuente Elaboración propia y propiedad de Synergy-GB | 19 |
| Figura 8. Aplicación móvil de la empresa, mostrando pantalla de generar código QR, Configurar y Consulta de Cobros y la opción de cobra. Fuente Elaboración propia y propiedad de Synergy-GB | 19 |
| Figura 9. Aplicación móvil del cliente con su pantalla de Bienvenida, Principal y Consulta de Pagos. Fuente: Elaboración propia y propiedad de Synergy-GB | 20 |
| Figura 10. Aplicación móvil del cliente con su pantalla de Captura de código QR, Pagar y Resumen de pago. Fuente: Elaboración propia y propiedad de Synergy-GB..... | 21 |
| Figura 9. Desarrollo de la interfaz gráfica en HTML de la pantalla principal de la aplicación móvil para la empresa, vista desde el editor de texto Sublime. Fuente: Elaboración propia y propiedad de Synergy-GB. | 24 |
| Figura 10. Estilos CSS empleados para el desarrollo de la pantalla principal de la aplicación móvil para la empresa, vista desde el editor de texto Sublime. Fuente: Elaboración propia y propiedad de Synergy-GB. | 25 |
| Figura 11. Desarrollo de la parte lógica en JavaScript de la pantalla principal de la aplicación móvil para la empresa, vista desde el editor de texto Sublime. Fuente: Elaboración propia y propiedad de Synergy-GB. | 25 |

Figura 12. Simulación en Google Chrome de la pantalla de Bienvenida de la aplicación de la empresa, apariencia en las plataformas IOS y Android. Fuente: Elaboración propia y propiedad de Synergy-GB... 26

Figura 13. Inspector de elementos del browser de Google Chrome, utilizado para editar los componentes de la aplicación. Fuente: Elaboración propia y propiedad de Synergy-GB. 26

1. SECCIÓN INTRODUCTORIA

1.1. Introducción

En la actualidad el desarrollo tecnológico ha evolucionado exponencialmente debido a la alta eficiencia que se obtiene, por lo tanto los problemas comunes que surgen en el día a día demandan de soluciones tecnológicas y de bajo costo, tal es el caso de los servicios de pagos actuales que involucran TDC (Tarjetas de Crédito) y TDD (Tarjetas de Débito) que dependen de un punto de venta para poder ser utilizadas, sin embargo tanto las TDC ,TDD y punto de venta no siempre están disponibles al momento de realizar un pago, generando así un problema.

El siguiente informe de proyecto de pasantía presenta una solución tecnológica que involucre el desarrollo de aplicaciones móviles multiplataforma, capaces de brindar una alternativa a los servicios de pagos, mediante el desarrollo de una aplicación móvil destinada a la empresa que desea generar una información de cobro mediante un código QR y una aplicación móvil para el cliente capaz de leer dicho código y realizar el pago respectivo.

Por lo tanto la estructura de este informe de pasantía tendrá una sección introductoria que contendrá la descripción de la empresa Synergy-GB donde se realizó el proyecto, un marco teórico que compone todos los estudios e investigaciones necesarias, los objetivos de este proyecto con sus respectivos alcances y premisas, y siguiendo una metodología que sustente el desarrollo del proyecto como se observa en la sección central donde se plantea el problema, la solución propuesta y la construcción de las aplicaciones móviles con sus respectivos resultados y por último en la sección final se muestran las conclusiones y recomendaciones del proyecto.

1.2. Descripción de la empresa

SYNERGY GLOBAL BUSINESS, C.A. es una empresa especializada en las áreas de Consultoría y desarrollo de aplicaciones Móviles y Web, así como la integración móvil, corporativa e inteligencia de negocio promoviendo siempre la calidad en sus productos y servicios. Synergy-GB es una empresa que ha estado adquiriendo experiencia desde el 2007 llevando así más de nueve (9) años en el mercado de la innovación y tecnología de punta, transfiriendo las ventajas de la web 2.0 a los sectores de la banca, seguros y empresas de Telecomunicaciones.

El organigrama de la empresa se muestra en la figura 1:

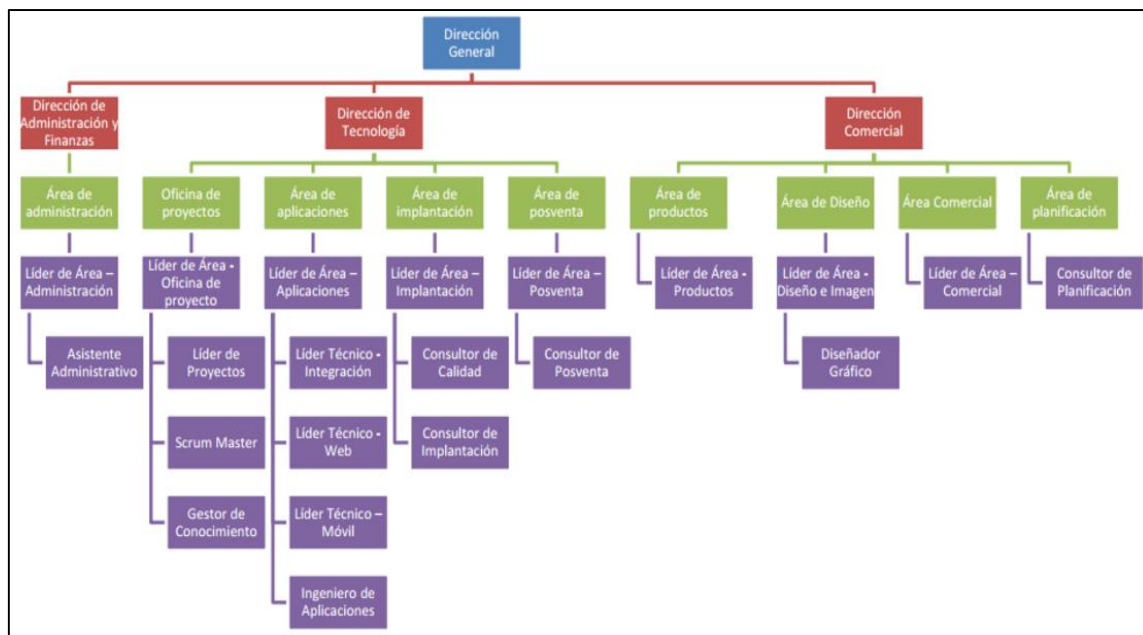


Figura 1. Organigrama de la empresa Synergy-GB. Fuente: Documentos de la empresa

Donde la dirección de tecnología se divide principalmente en tres (3) grandes secciones conocidas como habilidades (*Skills*):

- **Movilidad:** Ingeniero de Proyectos que se basa en el desarrollo de aplicaciones móviles en Android, BlackBerry, IOS, Windows Phone u otros. Conociendo las diferencias entre las plataformas y de las posibilidades que ofrece cada una de ellas en sus diferentes versiones.
- **Web:** Ingeniero de Proyectos que desarrolla aplicaciones web funcionales en varias familias de browsers de escritorio y móviles. Entiende el desarrollo de aplicaciones *responsive* o responsables y de las tendencias en el desarrollo de aplicaciones web en diversos lenguajes ya sean o no asistidos por manejadores de contenido.
- **Integración:** Ingeniero de Proyecto con facilidad para diseñar e implementar arquitecturas de integración de sistemas con conocimiento de diversos estándares de Redes e Internet en general. Cuentan con capacidad para desarrollar APIs de propósito específico o general. Enfocado en la integración de sistemas mediante protocolos de comunicación estándar, que permiten integrar aplicaciones móviles y web con las fuentes de datos y reglas de negocio. Desarrolla servicios web REST/SOAP, desarrolla en Java entre otros lenguajes, diseña y modela las estructuras de datos para base de datos SQL y No-SQL

Alguno de los clientes de Synergy-GB, son por nombrar algunos como Mercantil Banco, Banesco, Banco Activo, BFC, Banco Nacional de Panamá, CrediCorp, Banco Bicentenario, Banco Nacional de Crédito, entre otros ... en el sector bancario a los cuales se les ha dado el servicio del desarrollo de sus aplicaciones móviles con su respectiva integración.

1.3. Antecedentes de la investigación, Marco teórico

1.3.1. HTML

Es un lenguaje que se emplea para el desarrollo de páginas de internet. Está compuesto por una serie de etiquetas que el navegador interpreta (Google Chrome, Mozilla Firefox, Internet Explorer, Safari, etc...) y da forma en la pantalla de nuestro navegador. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc. Sin embargo HTML es un lenguaje que no dispone de sentencias básicas de programación como instrucciones de repetir procesos (ciclos) o condicionales (if- else) entre otros.

Cualquier documento de HTML se compone de elementos HTML los cuales tienen una etiqueta de apertura, un contenido y una etiqueta de cierre (`<p> Mi primer párrafo </p>`). Mientras que un atributo provee una información extra sobre los elementos y vienen representados en pares: nombre = "valor", siendo colocados dentro de la etiqueta de apertura (`<p title="Titulo de mi párrafo"> Mi primer párrafo </p>`). La estructura de un documento HTML según w3schools (2016) en su sección de HTML, la integran:

- El DOCTYPE: define el tipo de documento de HTML. Define la versión del documento.
- El texto entre `<html>` y `</html>` describe un documento HTML.
- El texto entre `<head>` y `</head>` proporciona información sobre el documento.
- El texto entre `<title>` y `</ title>` proporciona un título para el documento.
- El texto entre `<body>` y `</ body>` describe el contenido de la página visible.
- El texto entre `<h1>` y `</ h1>` describe un encabezamiento.
- El texto entre `<p>` y `</ p>` describe un párrafo.

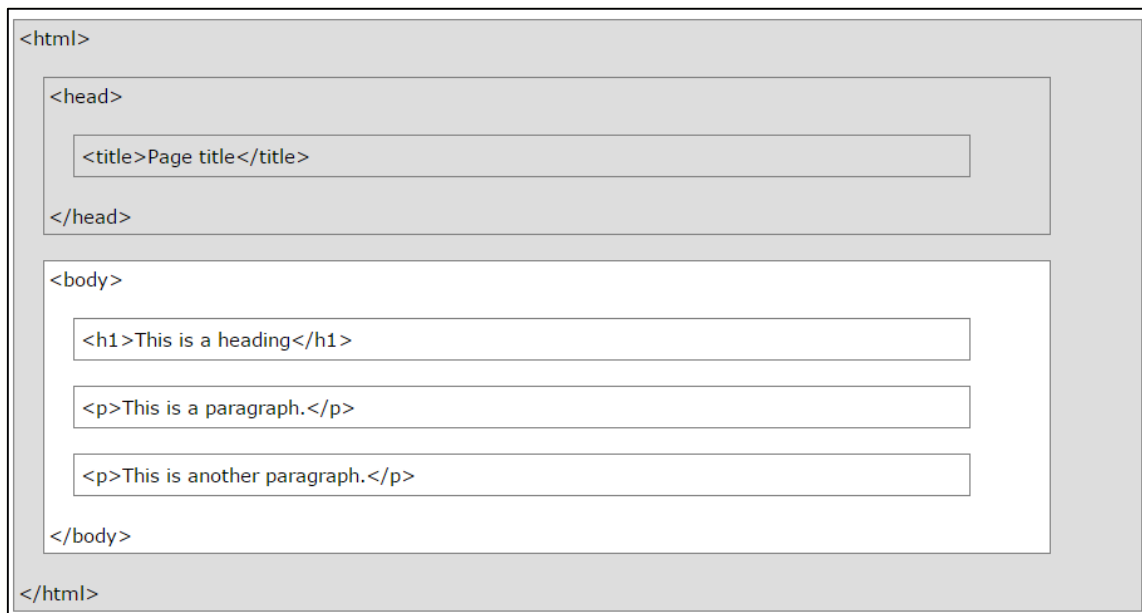


Figura 2. Estructura básica de un documento web utilizando HTML. Fuente: (W3schools, 2016)

1.3.2. HTML5

Según W3schools (2016) en su sección de HTML5 establece que es la última versión de HTML que permite añadir al lenguaje de etiquetas nuevas semánticas como: `<header>`, `<footer>`, `<article>`, y `<section>`, nuevas formas de control de atributos como el número, fecha, hora entre otros. Nuevos elementos gráficos: `<svg>` y `<canvas>`, nuevos elementos multimedia: `<audio>` y `<video>`. Además de incluir nuevas APIs como geolocalización, almacenamiento local, uso de caché, etc... Para hacer uso de HTML5 basta con declarar la etiqueta DOCTYPE, de la siguiente forma: `<!DOCTYPE html>`.

1.3.3. CCS

CSS es un lenguaje de estilo de hoja (styleSheet) que describe la presentación de un documento HTML (o XML), es decir, CSS describe cómo los elementos de un documento HTML deben ser presentados en la pantalla del navegador, en papel o en otros medios de comunicación, permitiendo manipular los elementos.

Para ello CSS permite implementar sus estilos mediante 3 formas según W3Schools (2016) en su sección de CSS:

- Inline – usando atributos de estilos en elementos HTML.
- Interno – usando el elemento `<style>` ubicado dentro de la sección `<head>` del documento HTML.
- Externo - usando uno o más archivos css externos.

1.3.4. JavaScript

Es un lenguaje de programación del tipo orientada a objetos que se utiliza en la web para establecer instrucciones lógicas, de tal forma que los documentos HTML permiten integrar JavaScript para añadir proceso o rutinas que posee la mayoría de los lenguajes de programación, por lo tanto define el comportamiento lógico de una página web. Según W3Schools (2016) en su sección de JavaScript.

JavaScript puede ser colocado en el `<body>` o en el `<head>` de un documento HTML, haciendo uso de la etiqueta `<script>` y `</script>`. También puede ser implementado mediante de archivos externos que se almacenen bajo la extensión “.js” y utilizando la etiqueta con el respectivo valor del atributo `<script src=”myScript.js”>`.

Sin embargo JavaScript para poder manipular el documento HTML hace uso del concepto de DOM (Document Object Model) el cual es creado por el browser y representa el documento HTML como un objeto manipulable, de esta forma JavaScript permite que las vistas estáticas que ofrece HTML puedan ser cambiadas.

1.3.5. Framework

Según Nubelo (2014) un Framework se entiende como una estructura de trabajo que permite guiar a uno o varios programas la forma de interrelacionarse y de trabajar, de esta

manera se expande la funcionalidad de un programa o lenguaje con el objetivo de construir algo más extenso, útil y sencillo.

1.3.6. Modelo-Vista-Controlador

Es una estructura que permite separar los datos de un programa con la lógica del mismo, de tal forma que el modelo representa la información, el controlador define el comportamiento de estos datos y la vista el lugar donde se exhiben dichos datos. Según Alvarez (2014)

1.3.7. AngularJS

Es un framework de JavaScript, permitiendo ser una librería que soporte funcionalidades adicionales en JavaScript, provocando que un documento HTML pueda tener una funcionalidad dinámica sobre sus elementos. AngularJS utiliza el mismo “tag” o etiqueta de JavaScript, es decir; `<script>` y `</script>` para poder ser empleado por HTML. Los archivos de AngularJS por ser un framework de JavaScript deben ser guardados bajo la extensión “.js”. Además AngularJS se basa en el modelo MVC (Modelo-Vista-Controlador). (AngularJS, 2016)

AngularJS extiende a HTML con nuevos atributos en sus elementos, definidos como directivas (`ng-directives`) las cuales generan nuevas funcionalidades sobre los elementos HTML, según W3Schools (2016) en su sección de AngularJS se pueden resumir en:

- *ng-app*: Permite añadir el funcionamiento de un archivo de AngularJS a un documento HTML.
- *ng-control*: Permite añadir un controlador que regule el funcionamiento de AngularJS con la vista, es decir; el documento HTML.
- *ng-bind*: Permite unir los elementos de un modelo (datos) con la vista (elementos HTML).

1.3.8. Ionic

Es un framework de código-abierto que potencia la funcionalidad de HTML5, permitiendo construir aplicaciones móviles con comportamiento nativo sobre plataformas como IOS y Android, para ello hace uso de HTML, CSS y JavaScript además del recurso necesario de AngularJS. De tal forma que Ionic se enfoca en la UI (Interfaz de usuario) de la aplicación móvil. (Ionic, 2015)

1.3.9. Cordova

Es un framework de código-abierto para el desarrollo de aplicaciones móviles y utilizando HTML, CSS y JavaScript, añadiendo plugins los cuales extienden la funcionalidad de un programa, permitiendo emplear elementos como la cámara, conexión de red, bluetooth, reproductor de música, GPS, etc., del dispositivo móvil. (Cordova, s.f.)

1.3.10. QR code (Quick Response Code)

Es un código que permite almacenar datos en una matriz de puntos, entendibles por lectores ópticos y dispositivos de capturas de imágenes como la cámara fotográfica de los teléfonos inteligentes o tabletas. Tiene una capacidad en que la información que se guarda puede de ser de numérico, caracteres, símbolos, binarios y códigos de control. (How To Geek, s.f.)

A su vez un código QR puede ser leído desde cualquier ángulo ya que posee una capacidad de decodificación de 360° permitiendo ser leído desde cualquier dirección por parte de la cámara. Cuenta a su vez con una alta capacidad de corrección de errores en caso de que el código QR sufra algún tipo de daño de manera física, ya que utiliza elementos de redundancia sobre el tipo de dato que se codifique.

1.3.11. Kiraso

También conocido como *Megazord*, es un framework desarrollado por Synergy-GB para crear aplicaciones móviles multiplataforma (tanto para IOS como Android) usando componentes reusables. Este framework trabaja sobre otros elementos como Ionic Framework, AngularJS y Cordova Framework. Para lograr esto, el framework utiliza archivos de configuración que sirven para definir la estructura de una aplicación y los componentes, con su respectiva configuración, que utilizará. También permite definir el enrutamiento y configuración global de la aplicación. (Synergy-GB, 2015)

1.3.12. Tecnología web

Se refiere al conjunto de lenguajes que se utilizan en el desarrollo de páginas web como HTML, CSS y JavaScript así como sus respectivos frameworks, además del entorno de trabajo representado por los browsers (Google Chrome, Internet Explorer, entre otros) que permiten el desarrollo y “debug” de cualquier página web. Se incluye además la capacidad que tienen estos lenguajes que mediante frameworks como Ionic, Cordova, PhoneGap entre otros, permiten la construcción de aplicaciones móviles.

1.4.Objetivos del estudio

1.4.1. Objetivo General

Diseñar y desarrollar aplicaciones móvil multiplataforma para ofrecer un servicio de pago a través de códigos QR, bajo la integración del framework de la empresa Synergy-GB utilizando tecnología web.

1.4.2. Objetivos Específicos:

- Analizar los requerimientos necesarios que involucran el desarrollo de las aplicación móvil multiplataforma

- Estudiar las herramientas y el ambiente de desarrollo necesarios para el desarrollo de las aplicaciones móviles
- Diseñar una aplicación móvil de la empresa en base a los requerimientos detallados.
- Diseñar una aplicación móvil del cliente en base a los requerimientos detallados.
- Desarrollar la aplicación móvil correspondiente a la empresa que permita generar un código QR que almacene la información correspondiente al cobro de un producto o servicio que ofrezca la empresa Synergy-GB.
- Desarrollar la aplicación móvil correspondiente al cliente que permita leer mediante un dispositivo de captura de imágenes como teléfonos inteligentes o tabletas y posteriormente realizar el pago bajo la integración de frameworks de la empresa Synergy-GB.
- Probar y evaluar el funcionamiento de las aplicaciones móviles.
- Elaborar el informe académico.

1.5.Alcances y premisas

El desarrollo de ofrecer un servicio de pagos multiplataforma mediante códigos QR se limitara únicamente en la construcción de las aplicaciones móviles, donde la información necesaria para realizar una transferencia como número de cuenta, monto, concepto, entre otros datos primordiales que necesite el cliente para realizar la transacción, serán almacenados en un código QR mediante la aplicación de la empresa para luego ser decodificada por la aplicación del cliente y seleccionar la cuenta desde la cual se desee debitar. Cualquier cuenta de banco que represente al cliente o la empresa será únicamente simulada bajo datos no reales.

En cuanto al proceso de ejecutar la transacción del cliente hacia la empresa y debitar de la cuenta de banco del consumidor no compete a este proyecto de pasantía.

1.6. Metodología empleada

1.6.1. FASE I - Planteamiento del problema

Investigación del problema que plantea la empresa Synergy-GB en cuanto los servicios de pagos presenciales actuales que involucran tecnología, con el fin de establecer los objetivos generales y específicos que puedan construir una solución al problema así como el empleo de un plan de trabajo mediante un cronograma que se acople a las seis (6) semanas que comprende este proyecto de pasantías.

1.6.2. FASE II - Investigación inicial

Identificado el problema con sus requisitos, se realizan las investigaciones y estudios que puedan construir una solución al problema planteado, por ello se sigue la siguiente modalidad:

- Documentación e investigación de lenguajes de programación que involucren tecnología web orientada al desarrollo de aplicaciones móviles multiplataforma.
- Estudio de los lenguaje HTML, CSS y JavaScript.
- Estudio de los frameworks AngularJS, Ionic y Cordova.
- Estudio del framework que ofrece la empresa Synergy-GB.
- Documentación de códigos QR.

1.6.3. FASE III - Diagnóstico de la situación actual

Una vez obtenido los estudios y documentaciones necesarias es posible identificar y enfrentar el problema planteado, mediante las investigaciones realizadas se puede ofrecer una solución que involucre la tecnología web y framework de trabajo que ofrece la empresa Synergy-GB con el fin de desarrollar aplicaciones móviles de forma rápida y sin sacrificar significativamente el desempeño de la aplicación. Siguiendo lo establecido en el cronograma de trabajo.

1.6.4. FASE IV - Diseño de propuestas

Esta fase consiste en el desarrollo y los resultados obtenidos de la solución al problema planteado, siguiendo los lineamientos que establece la empresa Synergy-GB en cuanto al desarrollo de aplicaciones móviles:

- Diseños de los *mockups* los cuales definen un boceto de la interfaz gráfica y funcionalidad que tendrá las aplicaciones móviles.
- Aprobación por parte del tutor empresarial de los *mockups* previamente realizados.
- Los diseños aprobados mediante *mockups* son pasados al área de diseño de aplicaciones con el fin de que se establezcan la apariencia visual que tendrá la aplicación móvil.
- Desarrollo de la aplicación móvil según los *mockups* y recursos generados para el entorno gráfico.

2. SECCIÓN CENTRAL

2.1.Descripción detallada de la situación actual y del problema propuesto

El progreso de la tecnología en la última década ha tenido un desarrollo exponencial en lo que se refiere a la red de telefonía móvil que ofrece el mundo digital mediante el concepto de transmisión de datos, así como también a los dispositivos móviles el cual genero un nuevo término tecnológico para los teléfonos móviles definidos como “Smartphone”, los cuales cuentan con una capacidad para ejecutar aplicaciones de alto nivel mediante el aprovechamiento de los recursos que posea el dispositivo electrónico, tales como GPS, red de datos, Wi-Fi, Bluetooth , entre otros, por otro lado es posible la interacción con el usuario de una manera más amigable provocando el termino de GUI (Graphics User Interface) que re-define la forma en como la aplicación móvil es presentada ante el cliente.

Está nueva interacción que existe entre el usuario y el teléfono móvil ha provocado una demanda de aplicaciones móviles que permita al usuario final extender sus capacidades de resolver problemas. En la actualidad en lo que se refiere con los servicios de pagos de manera presencial conocidos como TDC (Tarjetas de Crédito) y TDD (Tarjetas de Débito), cuenta con la limitación de solo ser usados si la empresa o comercio para generar el cobro cuenta con un punto de venta el cual depende de tres factores elementales para poder funcionar como son el papel de facturación, fuente eléctrica y la comunicación con la banca a través del cable telefónico, los cuales no siempre están disponibles en todas las regiones del país, por otro lado esta forma de pago electrónica solo podrá ejecutarse si la tarjeta está disponible y es presentada al momento del pago.

2.2.Evaluación de la situación mediante la presentación y el análisis de datos.

Esta etapa consiste en las investigaciones teóricas que soportan una base para entender el proceso y desarrollo de la aplicación, se contemplan estos estudios en el marco teórico antes mencionado.

Los estudios y documentaciones sobre tecnología web involucran a los lenguajes HTML, CSS y JavaScript, de donde los dos primeros lenguajes permiten construir la interfaz gráfica que tendrá la aplicación móvil mientras que el último lenguaje de programación es orientado a objetos y permite definir la lógica que tendrá la aplicación.

Por otro lado las investigaciones y adquirir las habilidades de los frameworks como Ionic AngularJS y Cordova, permitieron expandir las funcionalidades tanto de la interfaz gráfica como de la lógica en la aplicación móvil y que mediante *plugins* o extensiones Cordova es posible controlar las propiedades del dispositivo móvil, como la cámara fotográfica.

Finalmente el framework Kiraso suministrado por Synergy-GB permite unir todos estos lenguajes y framework antes mencionados en un solo framework, manteniendo los lenguajes HTML, CSS y JavaScript pero siguiendo otras reglas y protocolos para la construcción y generación de aplicaciones móviles multiplataforma.

2.3.Investigación y planteamiento de posibles soluciones

Una vez realizados los estudios en tecnología web y el framework Kiraso para el desarrollo de aplicaciones móviles multiplataforma, se propone desarrollar una solución tecnológica y de bajo costo que involucre la construcción de dos aplicaciones móviles capaces de sustituir o brindar una nueva opción a los servicios de pagos actuales mencionados anteriormente, mediante la creación de una aplicación móvil para el cliente que permita realizar un pago mediante la lectura de un código QR y otra para el comercio o empresa que pueda generar una información cobro mediante la generación dicho código,

de tal forma que se brinde una opción distinta al punto de venta actualmente utilizado en los servicios de pagos, sin embargo esta solución atractiva solo se basara en el desarrollo de las aplicaciones más no la integración con los servicios de la banca electrónica.

2.4.Diseño de propuestas y selección de la mejor opción

Los estudios y experiencias adquiridas en cuanto al empleo de tecnología web y el framework Kiraso, dio la oportunidad de plantear una solución gráfica y funcional en lo referente a las aplicaciones móviles que se van a desarrollar.

Por ello se construyeron unos bocetos de las aplicaciones móviles empleando el software de Balsamiq, dichos dibujos son los que se conocen como *mockups* y permiten dar una aproximación a la interfaz gráfica y a la funcionalidad que tendrá la aplicación.

De tal forma que se realizaron distintas propuestas en cuanto al desarrollo de la aplicación, en lo referente a su interfaz gráfica y funcionalidad. Dicho bocetos o *mockups* se pueden apreciar en las figura 4 y figura 3 para la aplicación móvil de la empresa mientras que las figura 5 y figura 6 representan a la aplicación móvil del cliente.



Figura 3. Mockups en Balsmiq de la aplicación móvil de la empresa. Se observan las pantallas de Bienvenida, Principal y Configurar. Fuente: Elaboración Propia y propiedad de Synergy-GB



Figura 4. Mockups en Balsmiq de la aplicación móvil de la empresa. Se observan las pantallas de Configuración, Cobrar y Consulta de Cobros. Fuente: Elaboración propia y propiedad de Synergy-GB



Figura 5. Mockups en Balsmiq de la aplicación móvil del cliente. Se observan las pantallas de Bienvenida, Principal y Lectura de QR. Fuente: Elaboración Propia y propiedad de Synergy-GB



Figura 6. Mockups en Balsmiq de la aplicación móvil del cliente. Se observan las pantallas de Pagar, Resumen de pago y Consulta de Pagos. Fuente: Elaboración Propia y propiedad de Synergy-GB

Una vez diseñados los *mockups* son presentados al tutor industrial y aprobado por el mismo, según su criterio y experiencia en las interacciones que podría tener la aplicación móvil con el usuario, brindando recomendaciones útiles en el desarrollo de las aplicaciones móviles. Finalmente aceptados los *mockups* son enviados al equipo de diseño para generar los recursos gráficos que contendrá las aplicaciones móviles.

Generado los recursos gráficos, se inicia la construcción de las aplicaciones móviles respectivas empleando el framework de Kiraso para brindar los recursos necesarios que permiten construir la aplicación, un editor de texto como *Sublime* donde se escriben los documentos según su lenguaje (HTML, CSS o JavaScript) y un browser como Google Chrome que permite simular la aplicación móvil. La construcción de las aplicaciones mediante el editor de texto y browser son mostradas en el Anexo A

Los resultados obtenidos del desarrollo de la aplicación y generadas en el dispositivo móvil con sistema operativo Android se pueden observar en la figura 7 y figura 8 la

aplicación móvil de la empresa mientras que la figura 9 y figura 10 para la aplicación móvil del cliente.

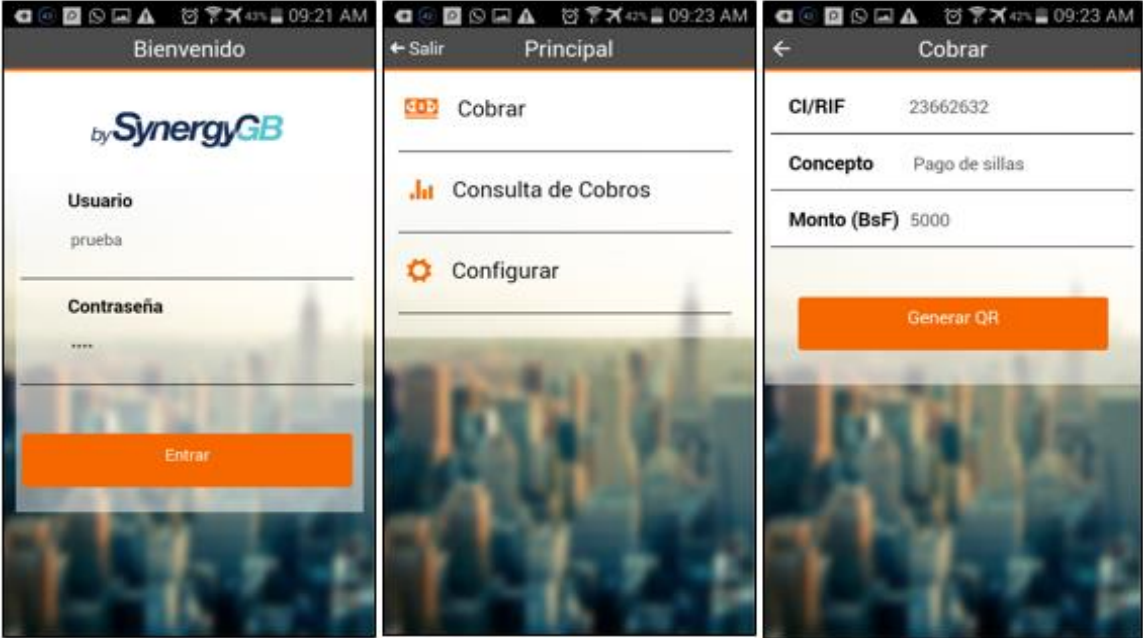


Figura 7. Aplicación móvil de la empresa, mostrando pantalla de Bienvenida, Principal y Cobrar. Fuente Elaboración propia y propiedad de Synergy-GB



Figura 8. Aplicación móvil de la empresa, mostrando pantalla de generar código QR, Configurar y Consulta de Cobros y la opción de cobra. Fuente Elaboración propia y propiedad de Synergy-GB



Figura 9. Aplicación móvil del cliente con su pantalla de Bienvenida, Principal y Consulta de Pagos.
 Fuente: Elaboración propia y propiedad de Synergy-GB

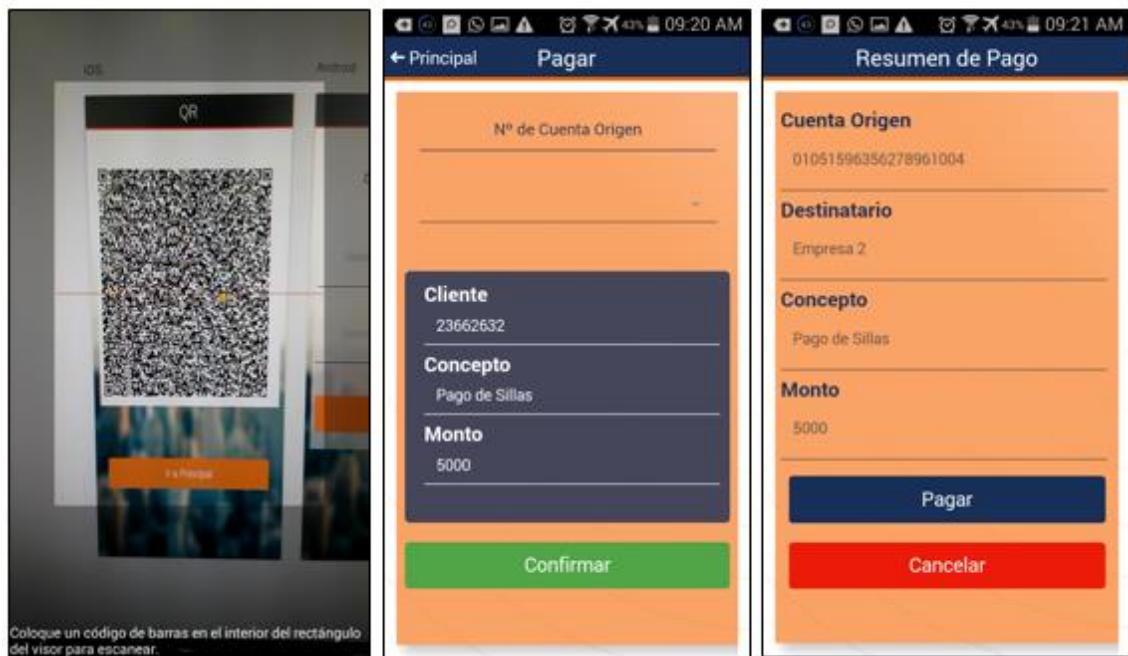


Figura 10. Aplicación móvil del cliente con su pantalla de Captura de código QR, Pagar y Resumen de pago. Fuente: Elaboración propia y propiedad de Synergy-GB

3. SECCIÓN FINAL

3.1.Conclusiones

Las investigaciones y estudios con respecto a la tecnología web y el framework de Kiraso, permitieron la construcción de aplicaciones móviles multiplataforma, es decir, que se desarrollaron las aplicaciones móviles simultáneamente en las plataformas IOS y Android sin utilizar el lenguaje nativo de estos respectivos sistemas operativos, en su lugar se emplearon los lenguajes HTML, CSS y JavaScript y los frameworks como *Ionic*, *AngularJS* y *Cordova* que componen a la tecnología web, mientras que el framework de Kiraso utilizando la misma tecnología y mediante sus plantillas reusables permite que el desarrollador se enfoque solo en la lógica e interfaz gráfica de la aplicación móvil, dejando que el framework haga el resto.

Sin embargo las aplicaciones móviles en lenguaje nativo ofrecen un rendimiento bastante similar a las desarrolladas con tecnología web, donde este último permite desarrollar las aplicaciones móviles en un menor tiempo. De tal forma que la construcción de una aplicación móvil para la empresa, permitió generar un código QR con la información de cobro, con el fin de que la segunda aplicación móvil destinada al cliente leyera dicho código y pudiera pagar, por consiguiente este proyecto de pasantía brindo una alternativa tecnológica, de bajo costo y practica ante los servicios de pagos actuales.

3.2.Recomendaciones

Si bien el desarrollo de las aplicaciones móviles multiplataforma en este proyecto se logró, aún no es posible su implementación por ello se recomienda generar la integración de las aplicaciones móviles con la banca electrónica, con el objetivo de que se implemente este proyecto de pasantía.

De igual forma se recomienda la investigación y estudio de la tecnología NFC, con el fin de que se pueda realizar este mismo proyecto pero empleando las etiquetas NFC ofreciendo otras alternativas en los servicios de pagos actuales.

Finalmente se recomienda implementar este proyecto de pasantía con sus modificaciones pertinentes, en el pago de los servicios de estacionamientos mediante la generación de un código QR en el ticket de entrada y que el cliente pueda pagar por el servicio simplemente leyendo el código QR de su ticket recibido cuando ingresa y mostrando el mismo en la salida una vez realizado el pago.

3.3.Bibliografía

Alvarez, M. A. (02 de Enero de 2014). *¿Qué es MVC?* [Página Web]. Recuperado el 30 de Marzo de 2016, de desarrolloweb.com: <http://www.desarrolloweb.com/articulos/que-es-mvc.html>

AngularJS. (30 de Marzo de 2016). *AngularJS* [Página Web]. Obtenido de <https://angularjs.org/>

Cordova. (s.f.). *Cordova Overview* [Página Web]. Recuperado el 30 de Marzo de 2016, de <http://cordova.apache.org/docs/en/latest/guide/overview/>

How To Geek. (s.f.). *QR Codes Explained: Why You See Those Square Barcodes Everywhere* [Página Web]. Recuperado el 28 de Febrero de 2016, de <http://www.howtogeek.com/162394/qr-codes-explained-why-you-see-those-square-barcodes-everywhere/>

Ionic. (2015). *Ionic Overview* [Página Web]. Recuperado el 30 de Marzo de 2016, de <http://ionicframework.com/docs/overview/>

Nubelo. (16 de Diciembre de 2014). *¿Que son los Frameworks?* [Página Web]. Recuperado el 30 de Marzo de 2016, de <http://www.nubelo.com/blog/que-son-los-frameworks/>

Synergy-GB. (2015). *Megazord Framework*. Recuperado el 25 de Febrero de 2016, de <http://s3.amazonaws.com/megazord-framework/docs/index.html#/index>

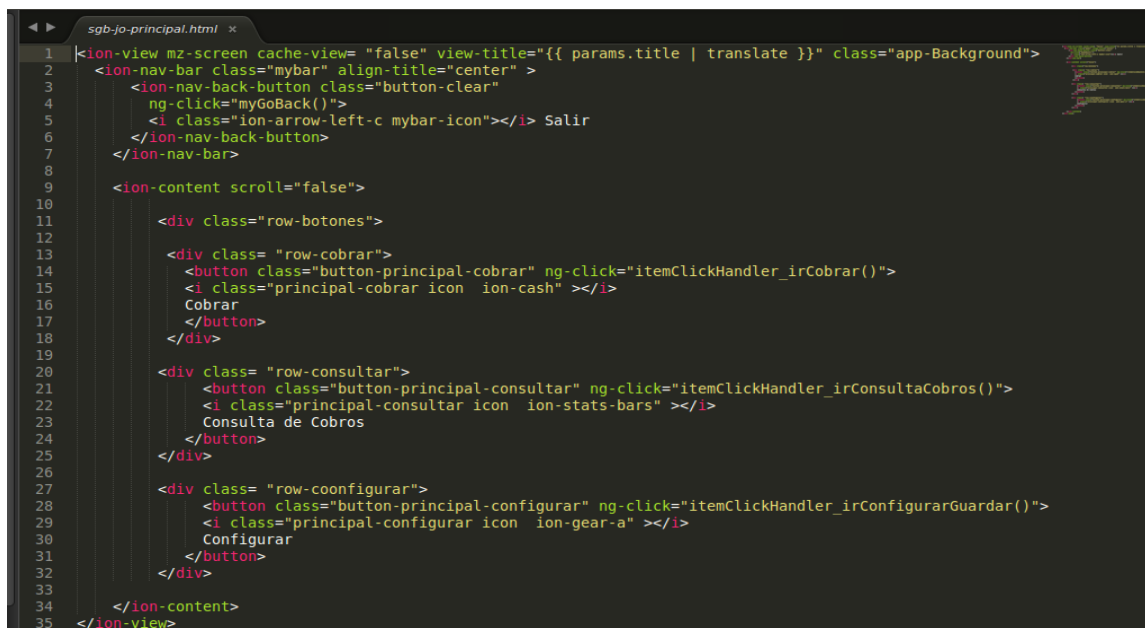
W3Schools. (2016). *AngularJS Introduction* [Página Web]. Recuperado el 30 de Marzo de 2016, de http://www.w3schools.com/angular/angular_intro.asp

W3Schools. (2016). *CSS Introduction [Página Web]*. Recuperado el 29 de Marzo de 2016, de http://www.w3schools.com/css/css_intro.asp

W3schools. (2016). *HTML Introduction [Página Web]*. Recuperado el 29 de Marzo de 2016, de <http://www.w3schools.com/html/default.asp>

W3Schools. (2016). *JavaScript tutorial [Página Web]*. Recuperado el 29 de Marzo de 2016, de <http://www.w3schools.com/js/default.asp>

3.4.Anexo A

A screenshot of a code editor window titled 'sgb-jo-principal.html'. The code is written in HTML and uses AngularJS directives. It defines an `ion-view` with a title from a parameter and a class 'app-Background'. Inside, there's an `ion-nav-bar` with a back button labeled 'Salir' and a clear button. The main content area is an `ion-content` with a scroll attribute. It contains three rows of buttons: 'Cobrar' with a cash icon, 'Consulta de Cobros' with a stats bars icon, and 'Configurar' with a gear icon. Each button has a specific `ng-click` handler and an icon class.

```
1 <ion-view m2-screen cache-view= "false" view-title="{{ params.title | translate }}" class="app-Background">
2   <ion-nav-bar class="mybar" align-title="center" >
3     <ion-nav-back-button class="button-clear"
4       ng-click="myGoBack()">
5       <i class="ion-arrow-left-c mybar-icon"></i> Salir
6     </ion-nav-back-button>
7   </ion-nav-bar>
8
9   <ion-content scroll="false">
10
11     <div class="row-botones">
12
13       <div class= "row-cobrar">
14         <button class="button-principal-cobrar" ng-click="itemClickHandler_irCobrar()">
15           <i class="principal-cobrar icon ion-cash" ></i>
16           Cobrar
17         </button>
18       </div>
19
20       <div class= "row-consultar">
21         <button class="button-principal-consultar" ng-click="itemClickHandler_irConsultaCobros()">
22           <i class="principal-consultar icon ion-stats-bars" ></i>
23           Consulta de Cobros
24         </button>
25       </div>
26
27       <div class= "row-coconfigurar">
28         <button class="button-principal-configurar" ng-click="itemClickHandler_irConfigurarGuardar()">
29           <i class="principal-configurar icon ion-gear-a" ></i>
30           Configurar
31         </button>
32       </div>
33
34     </ion-content>
35 </ion-view>
```

Figura 11. Desarrollo de la interfaz gráfica en HTML de la pantalla principal de la aplicación móvil para la empresa, vista desde el editor de texto *Sublime*. Fuente: Elaboración propia y propiedad de Synergy-GB.

```

1 //Define any sass variables and styles here.
2 //They may be overridden by the application using the screen.
3
4 $myColor: #00dda0;
5
6 .customHeader {
7   color: $myColor;
8 }
9
10 .row-botones {
11   background-image: url("resources/5 Othdr/componente-12.svg");
12   color: white;
13   text-align: center;
14   margin: 0px 0 0 -9px;
15   padding: 0 0 24px 0px;
16 }
17
18
19 .principal-cobrar{
20   margin: 0 14px 0 0px;
21   color: #F76700 ;
22   font-size: xx-large;
23 }
24
25 .principal-consultar{
26   margin: 0 14px 0 0px;
27   color: #F76700 ;
28   font-size: xx-large;
29 }
30
31 .principal-configurar{
32   margin: 0 14px 0 0px;
33   color: #F76700 ;
34   font-size: xx-large;
35 }
36
37
38 .row-cobrar {
39   background-color: rgba(51, 51, 51, 0);
40   color: white;

```

Figura 12. Estilos CSS empleados para el desarrollo de la pantalla principal de la aplicación móvil para la empresa, vista desde el editor de texto Sublime. Fuente: Elaboración propia y propiedad de Synergy-GB.

```

1 |'use strict';
2
3 angular.module('sgb-jo-principal', ['megazord'])
4   .controller('sgb-jo-principal-controller', ['$scope', '_router', '_screen', '_screenParams', '_data', 'func
5     _screen.initialize($scope, _screenParams);
6
7     $scope.params = _screenParams;
8     $scope.cuentaSeleccionada = _data;
9     console.log("Principal/*/$scope.cuentaSeleccionada", $scope.cuentaSeleccionada)
10    /*****
11     $scope.myGoBack = function() {
12       $scope.itemClickHandler_irLogin();
13     };
14
15     $scope.itemClickHandler_irCobrar = function(){
16       //Nothing to do but fire the event
17       _router.fireEvent({
18         name: 'irCobrar',
19         params: $scope.cuentaSeleccionada
20       })
21     };
22
23     $scope.itemClickHandler_irConsultaCobros = function(){
24       //Nothing to do but fire the event
25       _router.fireEvent({
26         name: 'irConsultaCobros',
27         params: $scope.cuentaSeleccionada
28       })
29     };
30
31     $scope.itemClickHandler_irConfigurar = function(){
32       //Nothing to do but fire the event
33       _router.fireEvent({
34         name: 'irConfigurar',
35         params : $scope.cuentaSeleccionada
36       })
37     };
38
39

```

Figura 13. Desarrollo de la parte lógica en JavaScript de la pantalla principal de la aplicación móvil para la empresa, vista desde el editor de texto Sublime. Fuente: Elaboración propia y propiedad de Synergy-GB.

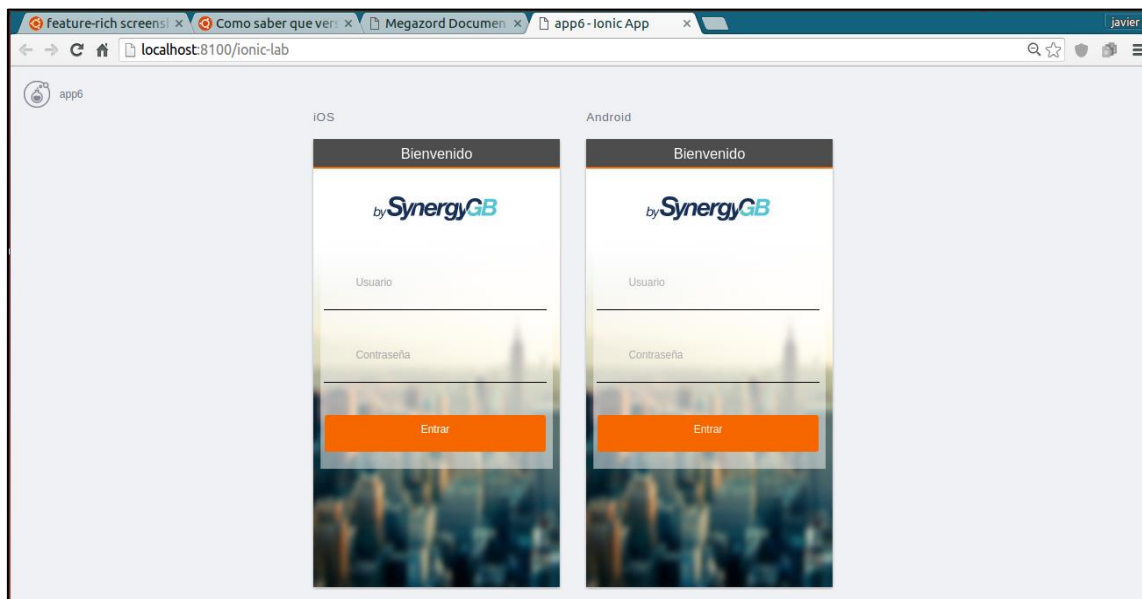


Figura 14. Simulación en Google Chrome de la pantalla de Bienvenida de la aplicación de la empresa, apariencia en las plataformas IOS y Android. Fuente: Elaboración propia y propiedad de Synergy-GB

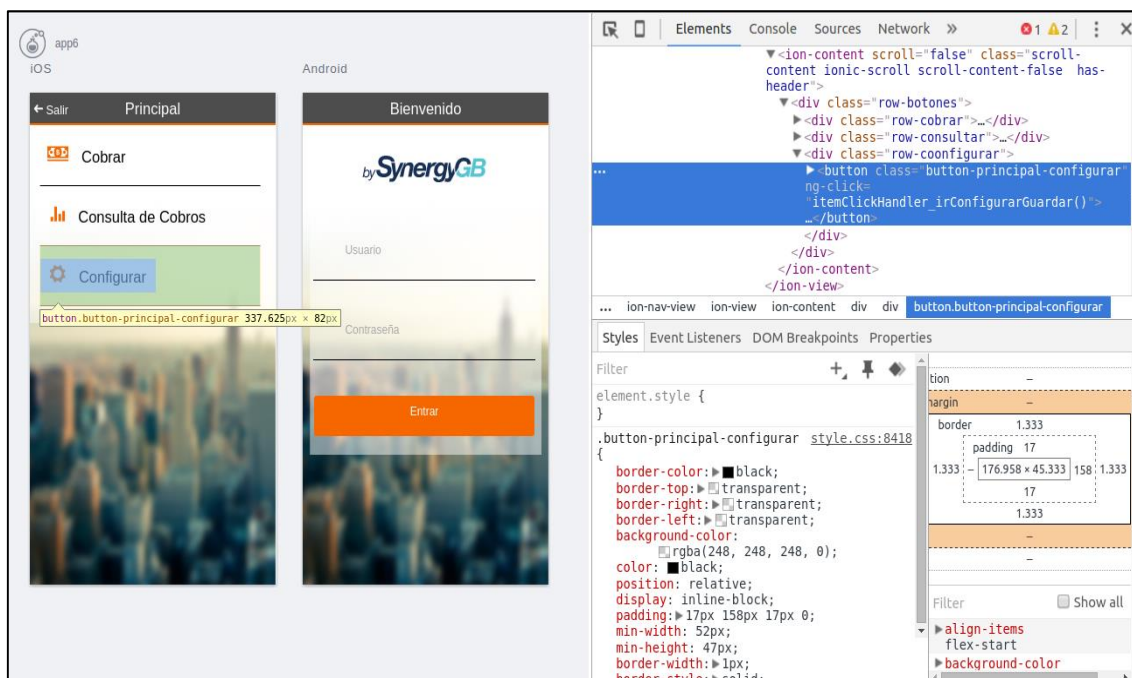


Figura 15. Inspector de elementos del browser de Google Chrome, utilizado para editar los componentes de la aplicación. Fuente: Elaboración propia y propiedad de Synergy-GB.