



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

**Analizar, diseñar y desarrollar nueva versión del sistema de reportes de
mensajería masiva de la empresa Tedexis.**

Por:
ALEJANDRO JESUS GUEVARA ALLEN

INFORME DE PASANTÍA LARGA
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación

Sartenejas, diciembre de 2016



UNIVERSIDAD SIMÓN BOLÍVAR
DECANATO DE ESTUDIOS PROFESIONALES
COORDINACIÓN DE INGENIERÍA DE LA COMPUTACIÓN

**Analizar, diseñar y desarrollar nueva versión del sistema de reportes de
mensajería masiva de la empresa Tedexis.**

Por:
ALEJANDRO JESUS GUEVARA ALLEN

Realizado con la asesoría de:
Tutor Académico: PROF. MASUN NABHAN HOMSI
Tutor Industrial: LIC. JESELYS E. HERNÁNDEZ MOYA

INFORME DE PASANTÍA LARGA
Presentado ante la Ilustre Universidad Simón Bolívar
como requisito parcial para optar al título de
Ingeniero en Computación

Sartenejas, diciembre de 2016

TIID: Tedexis Interactive Dashboard II

Por:

Alejandro Jesus Guevara Allen

RESUMEN

Tedexis Interactive Dashboard II (TIID) es un servicio de valor agregado para los clientes de tedexis. Su objetivo es ofrecer información sobre las transacciones que se realizan sobre su plataforma de mensajería (SMS, PUSH, EMAIL, RRSS, etc.), brindar mecanismos para el uso de la plataforma y dar información sobre los procesos, productos y métodos de integración asociados a la cuenta del cliente.

El presente informe muestra las actividades realizadas para el diseño e implementación del sistema TIID, como una interfaz web para que el cliente pueda estar informado sobre su actividad dentro de la empresa, donde tendrá acceso a múltiples reportes de tráfico de mensajes, acceso a manuales de productos de la empresa y gestionar tickets de soporte técnico. La información que visualiza cada usuario dependerá de su perfil dentro del sistema.

*A mis padres y mis hermanos
por apoyarme en todo lo que hago.*

*A mis abuelos y mi familia
por siempre estar.*

Agradecimientos

Primordialmente, gracias a Dios por permitirme vivir este momento.

Mis agradecimientos a la empresa Tedexis por brindarme la oportunidad de ser parte de su equipo y guiarme en los momentos difíciles del desarrollo del proyecto. Gracias por la confianza que me brindaron.

Agradezco enormemente a las profesora Masun Nabhan Homsí, por su guía y colaboración por las distintas veces que nos cruzamos durante mi carrera académica, su ánimo que siempre me contagia. De igual manera a todos los profesores de la Universidad Simón Bolívar, quienes día a día dan lo mejor de sí a pesar de las dificultades.

Gracias a mi familia, a los que se quedaron, los que se fueron y a los que ya no están. Muchas gracias a todos, siempre estarán presentes.

Mis amigos eternos e incondicionales: Pedro, Denyre, Daniela, Tabla, Jesús B., Jesús G., Luis, Karina y Marianny. Forman gran parte de quien soy hoy en día.

Muchas gracias a todas esas personas que comenzaron siendo conocidos y terminaron siendo amigos para toda la vida, Carlo, Stefany, Gabriela, Victor, William, Erick, Javier, Angel, Bono, Pedro, Amarant, Veronica y a la agrupación AniUSB. Sin ustedes no hubiera sido lo mismo.

Finalmente, gracias todas las personas que hicieron posible esta experiencia, en especial a las madres de mis amigos quienes me abrieron las puertas de sus casas y siempre estuvieron pendiente de mí. Desde el panadero que se aprendió mi nombre, hasta el señor que recoge la basura quien siempre me regala una sonrisa.

Nuevamente Gracias.

Índice General

Índice General	VIII
Índice de Tablas	XII
Índice de Figuras	XIII
Lista de Símbolos y Abreviaturas	XIV
Introducción	1
1. Entorno Empresarial	4
1.1. Antecedentes	4
1.2. Misión	5
1.3. Visión	5
1.4. Valores	5
1.5. Estructura Organizacional	5
2. Marco Teórico	7
2.1. Framework	7
2.2. Servicio Web	7
2.3. Transferencia de Estado Representacional	8
2.4. Patrón MVC	8
2.5. Hilos	8
2.6. JavaScript	8
2.7. AngularJS	9
2.8. MongoDB	9
2.9. Bower	9

2.10. NodeJS	9
2.11. NPM	9
2.12. HTML	10
2.13. JSON	10
2.14. JAVA	10
2.15. Bitbucket	10
2.16. JAX-RS	10
2.17. Jersey	11
3. Marco Metodológico	12
3.1. Roles	12
3.1.1. Scrum Master	12
3.1.2. Product Owner	13
3.1.3. Development Team	13
3.2. Eventos	13
3.2.1. Sprint	13
3.2.2. Sprint Planning	13
3.2.3. Daily Sprint Meeting	14
3.2.4. Sprint Review	14
3.2.5. Sprint Retrospective	14
3.3. Artefactos	14
3.3.1. Product Backlog	14
3.3.2. Sprint Backlog	15
4. Desarrollo de la aplicación	16
4.1. Familiarizarse con la empresa	17
4.1.1. Objetivos	17
4.1.2. Actividades	17
4.1.2.1. Familiarizarse con la empresa tedexis y el modelo de negocios empleado	17
4.1.2.2. Analizar el modelo de datos local, de la plataforma y demás fuentes de datos de la aplicación	18
4.2. Levantamiento de información	19
4.2.1. Objetivos	19

4.2.2.	Actividades	19
4.2.2.1.	Obtener los reportes que genera el sistema y son usados por los clientes de Tedexis	19
4.2.2.2.	Definir las opciones que existen por cada reporte generado por el sistema e identificar los servicios que son ofrecidos por la interfaz	19
4.2.2.3.	Obtener los roles disponibles y los privilegios asociados . . .	20
4.3.	Desarrollo de documentación	20
4.3.1.	Objetivos	20
4.3.2.	Actividades	20
4.3.2.1.	Creación de documentos iniciales	20
4.4.	Selección de tecnologías	23
4.4.1.	Objetivos	23
4.4.2.	Actividades	23
4.4.2.1.	Selección de frameworks de desarrollo	23
4.5.	Desarrollo del back-end	23
4.5.1.	Objetivos	23
4.5.2.	Actividades	23
4.5.2.1.	Diseño y creación de la base de datos	24
4.5.2.2.	Creación de hilos consolidadores de información	27
4.5.2.3.	Pruebas	28
4.6.	Autenticación	28
4.6.1.	Objetivos	28
4.6.2.	Actividades	29
4.6.2.1.	Creación del servicio Web RESTful	29
4.6.2.2.	Autenticación del usuario	29
4.6.2.3.	Pruebas	30
4.7.	Inicio del desarrollo de Reportes por Empresa	30
4.7.1.	Objetivos	30
4.7.2.	Actividades	31
4.7.2.1.	Creación de vista Saldo Pasaporte	31
4.7.2.2.	Creación de vista Evolución del Tráfico	31
4.7.2.3.	Creación de vistas Evolución Día y Evolución Hora	32
4.7.2.4.	Pruebas	32

4.8.	Fin del desarrollo de Reportes por Empresa	34
4.8.1.	Objetivos	34
4.8.2.	Actividades	34
4.8.2.1.	Desarrollo de vistas Top Empresas y Top Pasaportes	34
4.8.2.2.	Desarrollo de vistas Campañas y Distribución Operadoras	34
4.8.2.3.	Pruebas	35
4.9.	Desarrollo de Reportes por Pasaporte	36
4.9.1.	Objetivos	36
4.9.2.	Actividades	36
4.9.2.1.	Desarrollo de vistas de reporte de tráfico	36
4.9.2.2.	Pruebas	38
4.10.	Desarrollo de ultimas vistas del front-end	39
4.10.1.	Objetivos	39
4.10.2.	Actividades	39
4.10.2.1.	Vista principal	39
4.10.2.2.	Soporte	40
4.11.	Pruebas por el personal de la empresa	40
4.11.1.	Objetivos	40
4.11.2.	Actividades	40
4.11.2.1.	Pruebas del equipo de producto	40
4.11.2.2.	Pruebas del equipo de comercio	41
4.12.	Redacción del libro de pasantía	41
4.12.1.	Objetivos	41
4.12.2.	Actividades	41
4.12.2.1.	Redacción del libro y versión final de los documentos de la empresa	41
5.	Conclusiones y Recomendaciones	42
	Bibliografía	44
A.	Documento de Definición de Requerimientos	46
B.	Documento de Definición de Proyecto	54

Índice de Tablas

4.1. Tabla de duración de Sprints	16
---	----

Índice de Figuras

1.1. Estructura Organizacional de Tedexis	6
4.1. Tedexis	17
4.2. Vista ejemplo de MO y MT	18
4.3. Diagrama de casos de uso.	21
4.4. Diagrama de despligue.	22
4.5. Diagrama de conceptual de base de datos.	25
4.6. Vista de inicio de sesión.	29
4.7. Vista de creación de usuario.	30
4.8. Reporte Empresa: Saldo Pasaporte.	31
4.9. Reporte Empresa: Evolución del Tráfico.	32
4.10. Reporte Empresa: Evolución Día.	33
4.11. Reporte Empresa: Evolución Hora.	33
4.12. Reporte Empresa: Top Empresa.	34
4.13. Reporte Empresa: Top Pasaporte.	35
4.14. Reporte Empresa: Campañas.	35
4.15. Reporte Pasaporte: Evolución del Tráfico.	37
4.16. Reporte Pasaporte: Evolución Día.	37
4.17. Reporte Pasaporte: Evolución Hora.	38
4.18. Vista principal.	39
4.19. Vista soporte.	40

Lista de Símbolos y Abreviaturas

TIID: Tedexis Interactive Dashboard II

MO: Mensaje originado desde un teléfono celular (Mobile Originated)

MT: Mensaje recibido en un teléfono celular (Mobile Terminated)

MR: Mensaje de respuesta a un MO. No es comúnmente usado en telecomunicaciones pero forma parte del lenguaje en Tedexis

MVC: Modelo Vista Controlador (Model–View–Controller)

PNG: es un formato gráfico basado en un algoritmo de comprensión sin pérdida. (Portable Network Graphics)

Introducción

Planteamiento del problema

Actualmente la empresa Tedexis posee un sistema de reporte de mensajería masiva, TID (*Tedexis Interactive Dashboard*), alojado en su plataforma. Dicho sistema provee un servicio a través de una interfaz web donde el cliente puede estar informado sobre sus mensajes enviados, acceder a manuales y tutoriales de diferentes productos, gestionar tiques de soporte técnico y conocer los medios a redes sociales de la organización.

Esta versión del sistema ha cumplido con las necesidades de los clientes satisfactoriamente hasta hace aproximadamente 2 años, cuando la empresa empieza a tener un incremento exponencial de la cantidad de mensajes enviados, lo que produjo una congestión en la consulta de información ya que no estimaron que el crecimiento de la empresa sería de forma tan abrupta.

Es importante señalar que los tipos de mensajes manejados por la empresa Tedexis son:

- **MO:** corresponden a los mensajes originados desde teléfonos celulares. Las siglas MO corresponden a *Mobile Originated*.
- **MT:** corresponden a los mensajes recibidos por los teléfonos celulares. Las siglas MT corresponden a *Mobile Terminated*.
- **MR:** corresponden a los mensajes de respuesta de un MO. No es comúnmente usado en telecomunicaciones pero forma parte del lenguaje en Tedexis.

El problema más notorio que posee actualmente el sistema, es la lentitud en proveer los reportes de mensajería masiva al usuario, llegando a tardar incluso horas en la generación de dichos reportes, ocasionando que muchos clientes dejaran de utilizar el servicio. De igual manera, se presentan muchas quejas por parte del departamento de mercadeo de la empresa

sobre la apariencia y funcionalidades nuevas que debería poseer el sistema.

Por lo expuesto anteriormente, nace la idea de TIID (*Tedexis Interactive Dashboard II*) como versión mejorada del sistema en producción, con el que se desea obtener un producto capaz de mejorar las funcionalidades presentes en TID con una imagen más fresca, ligera, capaz de adaptarse a los dispositivos móviles, actual y con información integra y confiable. Esta nueva versión, debe ser capaz de obtener información de diferentes fuentes y proveer al cliente tiempos de respuesta decentes en la consulta de información.

Solución propuesta

Se plantea el desarrollo de un sistema que permita:

- Autenticación de usuarios al sistema y asegurar el acceso restringiendo a su información.
- Consolidación diaria del tráfico de datos de cada usuario para garantizar la disponibilidad y el rápido despliegue de información.
- Cargar y mostrar información de reportes pertinentes al usuario.

Objetivo general

Se propone como objetivo general optimizar las funcionalidades del sistema TID de la empresa Tedexis, enfocandose en el rápido despliegue de información de sus distintos reportes mediante el análisis, diseño y desarrollo de un nuevo sistema.

Objetivos específicos

- Analizar el modelo de negocio de la aplicación a fin de comprender las necesidades requeridas.
- Especificar los requerimientos del sistema TIID.
- Diseñar una solución que cumpla con las funcionalidades de la versión anterior del sistema (TID).
- Implementar y probar las funcionalidades definidas para el sistema.
- Ejecutar las pruebas unitarias, de integración y de regresión necesarias para el sistema.

Alcance

Se definió como alcance del proyecto la entrega de un prototipo que cumpla con 90 % de funcionalidad. Se estableció que se desarrollará y optimizará todas las funcionalidades que posee el sistema TID actualmente. Además se integrará un módulo de usuarios que permita el inicio de sesión y creación de nuevos usuarios, permitiendo elegir entre dos tipos de perfiles, administrador y general, que se diferenciarán simplemente en la cantidad de vistas a las que pueden acceder. Los requerimientos no funcionales a cumplir son: disponibilidad, confiabilidad, tiempos de respuesta cortos, seguridad y escalabilidad. Su implementación en producción no estuvo prevista como parte del proyecto de pasantía.

Organización del informe

Este informe presenta los resultados del diseño y desarrollo del sistema TIID donde se explican las diferentes fases del desarrollo del mismo y el proceso de transformación de un concepto inicial a un prototipo funcional. Consta de 5 capítulos organizados de la siguiente manera:

- Entorno Empresarial: provee una visión general de Tedexis, la empresa donde fue realizada la pasantía.
- Marco Teórico: presenta algunos conceptos teóricos, así como las herramientas tecnológicas utilizadas en el desarrollo del sistema.
- Marco Metodológico: describe la metodología utilizada en el desarrollo de la pasantía, Scrum.
- Desarrollo de la aplicación: presenta todo el proceso de desarrollo del sistema, actividades realizadas, problemas encontrados y como fueron resueltos.
- Conclusiones y recomendaciones: muestra las conclusiones y recomendaciones del proyecto.

Capítulo 1

Entorno Empresarial

En el presente capítulo se exponen las características fundamentales de la empresa en la que fue realizado el proyecto. En la primera sección se explican los antecedentes o descripción histórica de la empresa. En la segunda, tercera y cuarta sección, se detallan su visión, misión y valores respectivamente. En la quinta sección detalla la ubicación física de la empresa. La sexta sección explica la estructura organizacional de la empresa. Y finalmente, en la séptima sección se expone la ubicación del pasante dentro de dicha estructura empresarial. [1]

1.1. Antecedentes

Tedexis, es una empresa regional pionera en soluciones móviles para compañías, especialmente en las áreas de banca y comercio móvil. Desde su fundación en el año 2000, ha alcanzado el liderazgo regional con el mayor número de implementaciones exitosas en el mercado, convirtiéndose en referencia no sólo en Latinoamérica, sino globalmente.

Tedexis ofrece productos y servicios avanzados que son implementados 100% por talento Venezolano, ayudando a crear un mundo donde las transacciones y el intercambio de información se realice a través de la tecnología adecuada al tiempo adecuado, desde cualquier parte.

1.2. Misión

Exceder las expectativas de nuestros clientes mediante soluciones de tecnología móvil y talento humano comprometido e innovador. [1]

1.3. Visión

Ser una empresa centrada en el cliente, reconocida por su propuesta de valor superior en constante evolución, y por la competencia y motivación de su innovador capital humano. [1]

1.4. Valores

- Fans del servicio: eXceder las expectativas del cliente.
- Compromiso: cumplir lo que se ofrece y ofrecer + de lo que se espera.
- Innovación: haz + piensa diferente :).
- Excelencia: la única manera de hacer las cosas.
- Aprendizaje: ser + cada día.
- Equipo+: orgullosos de ser **TEDEXIS x +** :).

1.5. Estructura Organizacional

En la Figura 1.1 se presenta la estructura organizacional de la empresa, se tienen 5 áreas importantes entre las que se tienen:

- **Dirección de comercialización y mercadeo:** es la encargada de la gestión de ventas y relación con los clientes actuales y prospectos, promoviendo actividades que permitan desde su detección hasta el cierre de negociación.
- **Dirección de producto:** encargada de la creación de productos en la organización. Son los responsables del desarrollo de software desde el inicio hasta su maduración y consolidación. En esta dirección fue ubicado el pasante durante todo el proyecto de pasantías.
- **Dirección de operaciones:** dirección encargada del mantenimiento de los productos en producción de la empresa desde un enfoque técnico.
- **Dirección de ingeniería de soluciones:** esta dirección es la encargada de gerenciar los nuevos y futuros proyectos dentro de la empresa. Generan prototipos de soluciones

a problemas relevantes en los distintos departamentos.

- **Dirección de planificación y seguimiento:** área encargada de velar por la calidad de los servicios que ofrece la empresa en forma global, desde procesos, metodologías de desarrollo, ejecución de proyectos y satisfacción de los clientes.

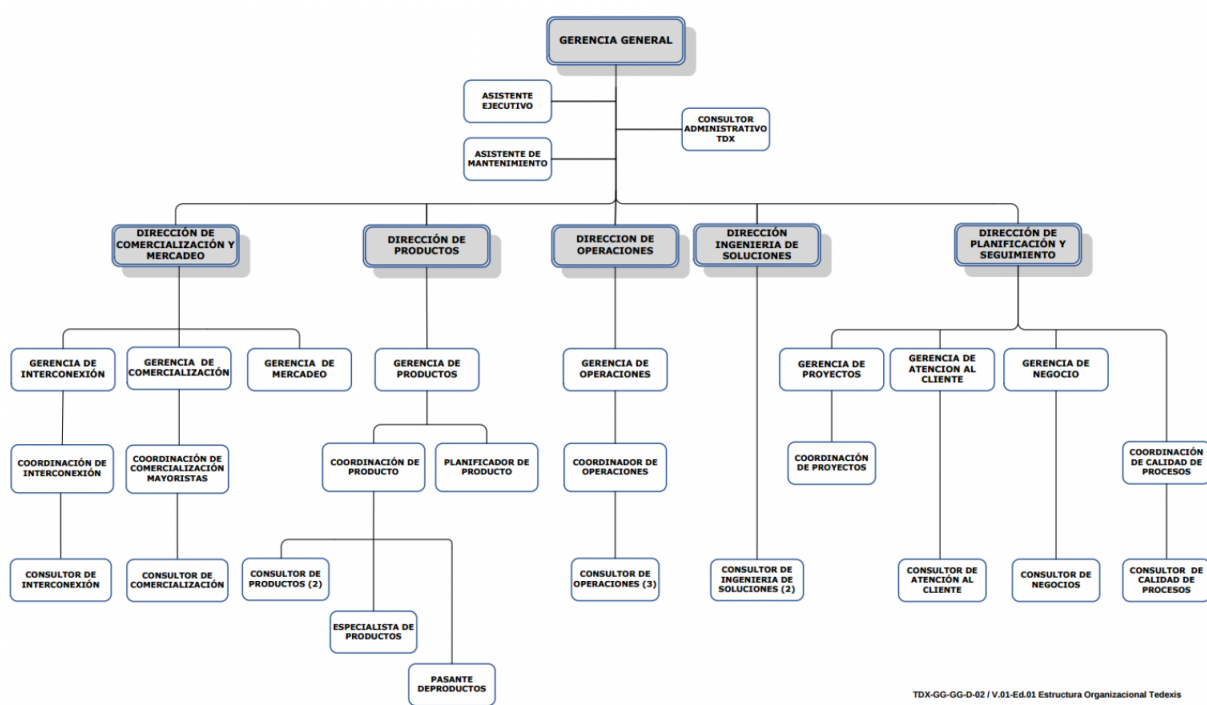


Figura 1.1: Estructura Organizacional de Tedexis

Capítulo 2

Marco Teórico

En este capítulo se exponen los conceptos fundamentales que sustentan el trabajo realizado, adicionalmente se explican las herramientas tecnológicas utilizadas para el desarrollo.

2.1. Framework

Un framework se define como un conjunto estandarizado de conceptos, prácticas y criterios para enfrentar un tipo de problema particular. De igual manera puede ser visto como un esqueleto o patrón para el desarrollo de una aplicación. [2]

2.2. Servicio Web

Un servicio web es un conjunto de protocolos y estándares empleados para intercambiar información entre aplicaciones web, no necesariamente escritas en el mismo lenguaje de programación, ni alojadas en la misma plataforma por lo que brinda versatilidad en el desarrollo. Algunos de los servicios web mas utilizados son: XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), REST (Representational State Transfer) entre otros. Los servicios web dentro de un framework, son generalmente utilizados para establecer comunicación entre el back-end y front-end del producto a desarrollar. [4]

2.3. Transferencia de Estado Representacional

La Transferencia de Estado Representacional (REST), es un estilo de arquitectura para sistemas distribuidos. Se apoya en el protocolo HTTP para definir todas las operaciones que puede realizar, GET, POST, PUT y DELETE, que permiten la comunicación entre el servicio y el cliente. Gracias a la arquitectura desacoplada y versatilidad en la comunicación entre productor y consumidor, la popularidad en la utilización de REST aumente considerablemente. [6]

2.4. Patrón MVC

Modelo-Vista-Controlador es un patrón de diseño que busca estructurar aplicaciones de manera modular. [3] Para esto, se pueden apreciar tres componentes:

- Modelo: Aquí se encuentra encapsulada la estructura y funcionalidad de la aplicación.
- Vista: Este componente representa la interfaz gráfica que es mostrada al usuario.
- Controlador: El propósito del controlador es desacoplar el modelo de la vista. Define la forma en la que la aplicación reaccionará a la interacción con el usuario.

2.5. Hilos

Los hilos, también conocidos como *Threads* es una forma de realizar concurrencia, los distintos hilos de ejecución comparten una serie de recursos tales como el espacio de la memoria, los archivos abiertos, etc. Un hilo es simplemente una tarea que puede ser ejecutada al mismo tiempo con otra tarea. [18]

2.6. JavaScript

JavaScript es un lenguaje interpretado, orientado a objetos, débilmente tipado y dinámico. Los objetos se crean añadiendo métodos y propiedades. Una vez se ha construido un objeto, puede usarse como modelo (o prototipo) para crear objetos similares. [8]

2.7. AngularJS

AngularJS es un framework estructural para aplicaciones web dinámicas. Su uso permite la modularidad e independencia de Mediante el enlace de datos y la inyección de dependencias facilita muchísimo el trabajo de desarrolladores permitiendo que el código sea mucho más simplificado y concreto. Funciona únicamente en el Front-End, lo que permite poder usar cualquier cosa en el Back-End sin que se vea afectado su funcionamiento.

Un uso adecuado de este framework proporciona ventajas como reusabilidad, modularidad e independencia del servidor. [9]

2.8. MongoDB

MongoDB es una base de datos NoSQL basada en esquemas, que gracias a su escalabilidad y agilidad permite que los esquemas puedan cambiar rápidamente a medida que las aplicaciones evolucionan, proporcionando siempre la funcionalidad que los desarrolladores esperan de las bases de datos tradicionales, tales como índices secundarios, un lenguaje completo de búsquedas y consistencia estricta. [10]

2.9. Bower

Bower es un manejador de versiones que administra componentes que tienen HTML, CSS, JavaScript entre otros. Te permite instalar las dependencias con los paquetes necesarios. [11]

2.10. NodeJS

Es una plataforma desarrollada sobre V8 JavaScript Runtime, posee una arquitectura basada en un manejador de eventos capaz de realizar tareas asincrónicas en I/O, lo que lo hace eficiente para aplicaciones que manejan grandes cantidades de datos en tiempo real. [15]

2.11. NPM

Node package manager, también conocido como NPM, es un manejador de paquetes o módulos que permite a los desarrolladores compartir soluciones. Con NPM se puede especificar la versión que se desea instalar y las dependencias necesarias. [14]

2.12. HTML

El lenguaje de marcado de hipertexto (HyperText Markup Language) es el elemento de construcción más básico de una página web y se utiliza para crear y representar visualmente una página web. [12]

2.13. JSON

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. Es fácil de leer y escribir para humanos, como es de fácil de parsear y generar para computadoras [5]. Esta compuesto de dos estructuras:

- Colecciones, que vienen en pares de nombre y valor
- Lista ordenada de valores.

2.14. JAVA

Java es un lenguaje de programación orientado a objetos. Su propósito es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo. [7]

2.15. Bitbucket

Bitbucket es controlador de versiones distribuido, que facilita el trabajo grupal y permite alojamiento web a todo tipo de proyectos. Este servicio está escrito en Python. [13]

2.16. JAX-RS

JAX-RS también conocido como Java API for RESTful Web Services, es una API implementada en JAVA que proporciona los recursos necesarios para la creación de servicios web de acuerdo al estilo REST [20].

2.17. Jersey

Jersey para servicios web RESTful es un framework de código abierto de calidad que provee soporte para JAX-RS y sirve como una implementación de las referencias de JAX-RS. Jersey tiene su propio API con características adicionales y utilidades que ayudan aún más a simplificar el servicio REST y el desarrollo del cliente [21].

Capítulo 3

Marco Metodológico

Este capítulo describe la metodología de desarrollo utilizada durante el proyecto de pasantía. Para este proyecto se empleó la metodología ágil SCRUM, debido a que el departamento de producto, el equipo de desarrollo de software, utiliza esta metodología y es con la que han tenido mejores resultados. [16]

En este proyecto de pasantía, el *Product Owner* fue un consultor de producto y el *Scrum Master* el coordinador de producto, ambos supervisados por la directora de productos. A continuación se presentan los componentes que forman parte de esta metodología:

3.1. Roles

Existen tres roles principales dentro del equipo de trabajo, cada rol posee un conjunto de actividades bien definidas que deben ser realizadas en un período de tiempo específico para que el proyecto pueda culminar de manera exitosa. Estos son los roles:

3.1.1. Scrum Master

El Scrum Master, es responsable de asegurar que el proceso SCRUM sea entendido y aceptado por todos los miembros del equipo, esto se hace haciendo uso de las reglas, prácticas y teorías. Por otro lado, el Scrum Master actúa como mediador entre agentes externos y el equipo de desarrollo, de esta manera se minimiza los problemas que puedan existir con la interacción entre los mismos. [16]

3.1.2. Product Owner

El Product Owner es responsable de maximizar el valor del producto en creación y el trabajo del equipo desarrollador o *development team*, también tiene la visión del producto y trabaja en colaboración con usuarios, clientes y *stakeholders*. El Product Owner es el único responsable de determinar cuales son las funcionalidades a crear en el *product backlog* o pila del producto. [16]

3.1.3. Development Team

Es el equipo de trabajo que se encarga de llevar a cabo el diseño y desarrollo del software requerido. El equipo tiene la responsabilidad de cumplir con las historias del *Product Backlog* para ir desarrollando paso a paso el producto, la manera en como se desarrollan estas historias es una decision tomada por el equipo. Por lo general consta de 3 a 9 personas. [16]

3.2. Eventos

Los eventos en la metodología Scrum son todas las reuniones utilizadas para crear regularidad y actualizar al equipo constantemente del estado del proyecto y las siguientes acciones que se deberian llevar a cabo. Todos estos eventos tienen un tiempo de duración y son aprovechados para revisar y adaptar cada una de las historias desarrolladas. Estos eventos son los siguientes: [16]

3.2.1. Sprint

El *Sprint* es el corazón del *Scrum*, es un período de tiempo donde se desarrollan ciertas historias propuestas en el *Sprint Backlog* y se entrega un producto funcional. Es recomendado que la duración de los sprints sea constante y definida por el equipo de desarrollo en base a su experecia. Un nuevo *Sprint* comienza inmediatamente despues de concluir uno.

3.2.2. Sprint Planning

Sprint Planning, es la reunión que se realiza al inicio de cada *Sprint* en la que se determinará que historias serán realizadas. El Product Owner, como se mencionó anteriormente, es el encargado de definir las historias y organizarlas según su prioridad, seguidamente el equipo de desarrollo determina las historias que pueden desarrollar. [16]

3.2.3. Daily Sprint Meeting

Reunión de máximo 15 minutos diaria que sirve para sincronizar y actualizar el plan de actividades para las siguientes 24 horas de desarrollo. Esta reunión se realiza todos los días y cada miembro resume lo que hizo el día anterior, como le va en el día y que problemas ha presentado. [16]

3.2.4. Sprint Review

Se lleva a cabo al final de cada *Sprint*, donde se muestran los avances realizados en el producto. Dichos avances son presentados al *Product Owner* y todos los demás interesados (pueden estar presentes clientes finales).

3.2.5. Sprint Retrospective

Esta reunión sirve para reflexionar sobre el último Sprint realizado y sirve para inspeccionar los problemas que se tuvieron y buscar una manera de solucionarlos en los sprints venideros. Tiene un máximo de duración de 3 horas. [16]

3.3. Artefactos

Los artefactos de SCRUM representan trabajo o valor de diversas formas que son útiles para brindar transparencia para la inspección y adaptación. Estos artefactos están diseñados para maximizar el entendimiento de la información. [16]

3.3.1. Product Backlog

Product backlog o pila del producto es una lista ordenada de todas las historias que pueden ser necesarias para el producto, con descripciones detalladas y sus prioridades de todos los requisitos funcionales y no funcionales. Esta lista es dinámica y pública para todos los involucrados en el proyecto. La pila del producto es creada por el *Product Owner* durante el *Sprint Planning*. [16]

3.3.2. Sprint Backlog

Es un documento detallado por parte del equipo de desarrollo, donde se predicen las actividades necesarias a desarrollar en el siguiente *sprint* para dar como terminadas ciertas tareas. [16]

Capítulo 4

Desarrollo de la aplicación

Este capítulo describe el desarrollo completo del sistema empleando la metodología *Scrum*. Se mostrarán los 12 *Sprints* que corresponden a las siguientes fases: análisis y especificación de requerimientos, diseño e implementación. De igual manera, se describen los artefactos generados, actividades realizadas y soluciones a problemas presentados durante el desarrollo de cada *Sprint*. A continuación se muestra en la tabla 4.1 la distribución del tiempo de cada *Sprint*.

Tabla 4.1: Tabla de duración de Sprints

Sprint	Nombre	Duración Semanas
1	Familiarizarse con la empresa	2
2	Levantamiento de información	1
3	Desarrollo de documentación	1
4	Selección de tecnologías	1
5	Desarrollo del back-end	2
6	Autenticación	2
7	Inicio del desarrollo de Reportes por Empresa	3
8	Fin del desarrollo de Reportes por Empresa	2
9	Desarrollo de Reportes por Pasaporte	1
10	Desarrollo de ultimas vistas del front-end	1
11	Pruebas por el personal de la empresa	2
12	Redacción del libro de pasantias	2

4.1. Familiarizarse con la empresa

4.1.1. Objetivos

- Integración y conocimiento general de la empresa Tedexis, modelo de negocio y lugar de trabajo.

4.1.2. Actividades

4.1.2.1. Familiarizarse con la empresa tedexis y el modelo de negocios empleado

Desde el año 2000, Tedexis ofrece servicios de alta calidad para que las empresas accedan fácilmente a la tecnología SMS. Las empresas tienen necesidades específicas para integrar el envío de SMS, ya sea para la comunicación interna entre sus empleados, como con sus clientes para el envío masivo de SMS.

Tedexis ha desarrollado un *Gateway* transaccional para el manejo de SMS llamado Mercurio que ofrece la posibilidad de integrarse con todos los proveedores de telecomunicaciones móviles de América Latina, vea la figura 4.1.

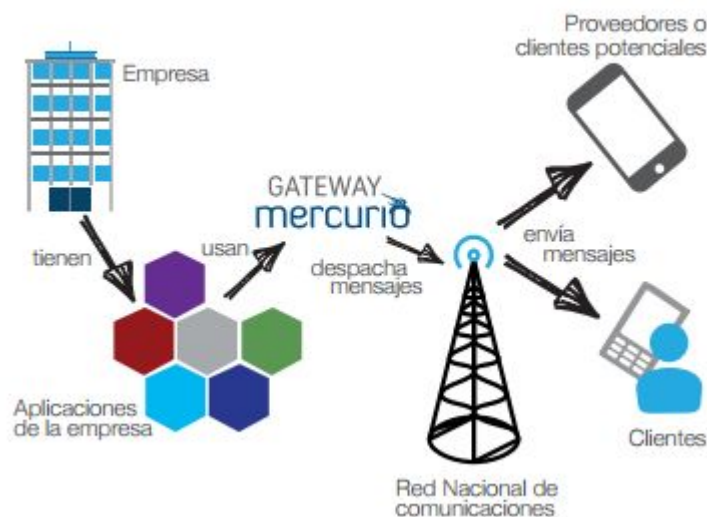


Figura 4.1: Tedexis

Los SMS se pueden clasificar como MO o MT dependiendo de su origen, como podemos ver en el gráfico 4.2. Si el mensaje es enviado por el usuario se le llama un mensaje MO y a la respuesta a este mensaje proveniente de una aplicación, se le llama MT, como por ejemplo la consulta de saldo por parte de un usuario es un MO, y la respuesta con el saldo es un MT.

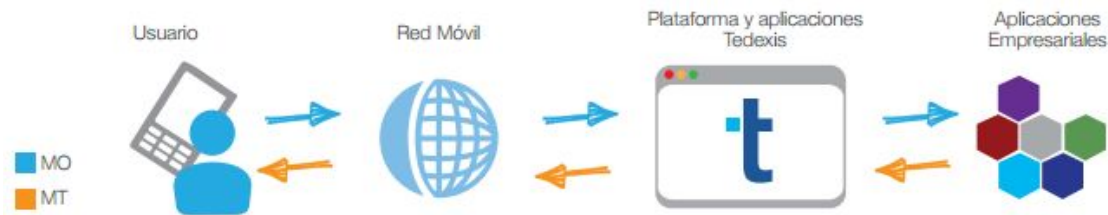


Figura 4.2: Vista ejemplo de MO y MT

4.1.2.2. Analizar el modelo de datos local, de la plataforma y demás fuentes de datos de la aplicación

Inicialmente se realizaron dos reuniones donde se mostraron el modelo de datos actual que posee la empresa Tedexis en su plataforma Mercurio. Mercurio es una puerta de enlace de manejo de SMS, capaz de integrar diferentes proveedores nacionales e internacionales en la lógica de intercambio de los mensajes de texto. La mayoría de las bases de datos están en PostgreSQL, pero poseen como proyecto la migración de dichas bases de datos a MongoDB.

De igual manera se recibieron tres clases donde se mostro parte del código que se utiliza como estándar para el desarrollo de aplicaciones en java dentro de Tedexis. Entre el código mostrado estaban librerías para conexiones a bases de datos PQSL las cuales tuvieron que ser adaptadas para el funcionamiento con MongoDB; librería para el uso de *Logs* o registros; y librerías para el uso de MQ (*Message queue*) las cuales no fueron utilizadas en este proyecto de pasantías.

Estas reuniones fueron realizadas con el Scrum Master, quien realizó la mayoría de las explicaciones y un miembro del área de dirección de productos de Tedexis, quien expuso los conceptos sobre MongoDB.

4.2. Levantamiento de información

4.2.1. Objetivos

- Definir reportes y funcionalidades a desarrollar.

4.2.2. Actividades

4.2.2.1. Obtener los reportes que genera el sistema y son usados por los clientes de Tedexis

Actualmente existe una versión de TID en producción, de la cual se tomaron los reportes minimos que debería poseer la segunda versión del sistema (TIID), por lo que se tomaron los reportes existentes como el alcance mínimo de la nueva versión. Los reportes fueron divididos en dos categorías: Reporte por empresa y reporte por pasaporte.

Reporte por empresa muestra diferentes tipos de reportes referente a la empresa asociada al usuario, dichos reportes son: Evolución de tráfico de mensajes total enviados tanto por día como por hora por la empresa en un rango de tiempo, pasaportes asociados a la empresa, distribución de los mensajes enviados por las operadoras asociadas, campañas de mensajería masiva y top de empresas asociadas según la cantidad de mensajes que hallan enviado.

Reporte por pasaporte muestra los reportes referente a los pasaportes asociados al usuario como individuo, dichos reportes son: Evolución de tráfico de mensajes enviados tanto por día como por hora y distribución de mensajes enviados por operadoras. La consulta de la información de todos los reportes se realiza eligiendo un rango de fechas específico.

4.2.2.2. Definir las opciones que existen por cada reporte generado por el sistema e identificar los servicios que son ofrecidos por la interfaz

Al tener definidos los reportes a implementar, se decidió que la información debería mostrarse en tablas, de igual manera, se definieron los campos específicos que debían mostrarse por cada reporte en dichas tablas y los tipos de gráficos que deberían mostrarse según el tipo de reporte los cuales fueron: Gráficos de líneas y barras en los reportes de evolución de tráfico; gráfico de torta en reporte de operadoras y top empresas.

Posteriormente, se concretó que los datos mostrados en las tablas deberían poder ser exportados por el usuario en formato csv, de igual manera, la tablas deberían poseer algún mecanismo de búsqueda y reorganización de los datos mostrados tanto por campos individuales como de forma global.

4.2.2.3. Obtener los roles disponibles y los privilegios asociados

En la versión actual del sistema, los usuarios son gestionados por un sistema externo a TID por lo que se decidió crear un módulo de usuarios dentro de TIID. Inicialmente se propone que dentro de TIID el usuario administrador sea quien pueda crear y editar los usuarios del sistema, teniendose como perfiles: Tedexis (administrador) y canales (usuario regular). Con la diferencia de que el perfil tedexis posee acceso a todo el sistema, mientras que el perfil canales tiene restringidas ciertas vistas.

4.3. Desarrollo de documentación

4.3.1. Objetivos

- Diseño de la solución global a implementar, parte 1.

4.3.2. Actividades

4.3.2.1. Creación de documentos iniciales

Siguiendo los estandares de Tedexis, primero fue solicitado crear un documento de los casos de usos del sistema, utilizando una plantilla proporsionada por la empresa. Teniendo como versión final el siguiente diagrama 4.3. Donde se muestran las funcionalidades que tiene acceso el usuario como son, autenticación, soporte y los diferentes reportes.

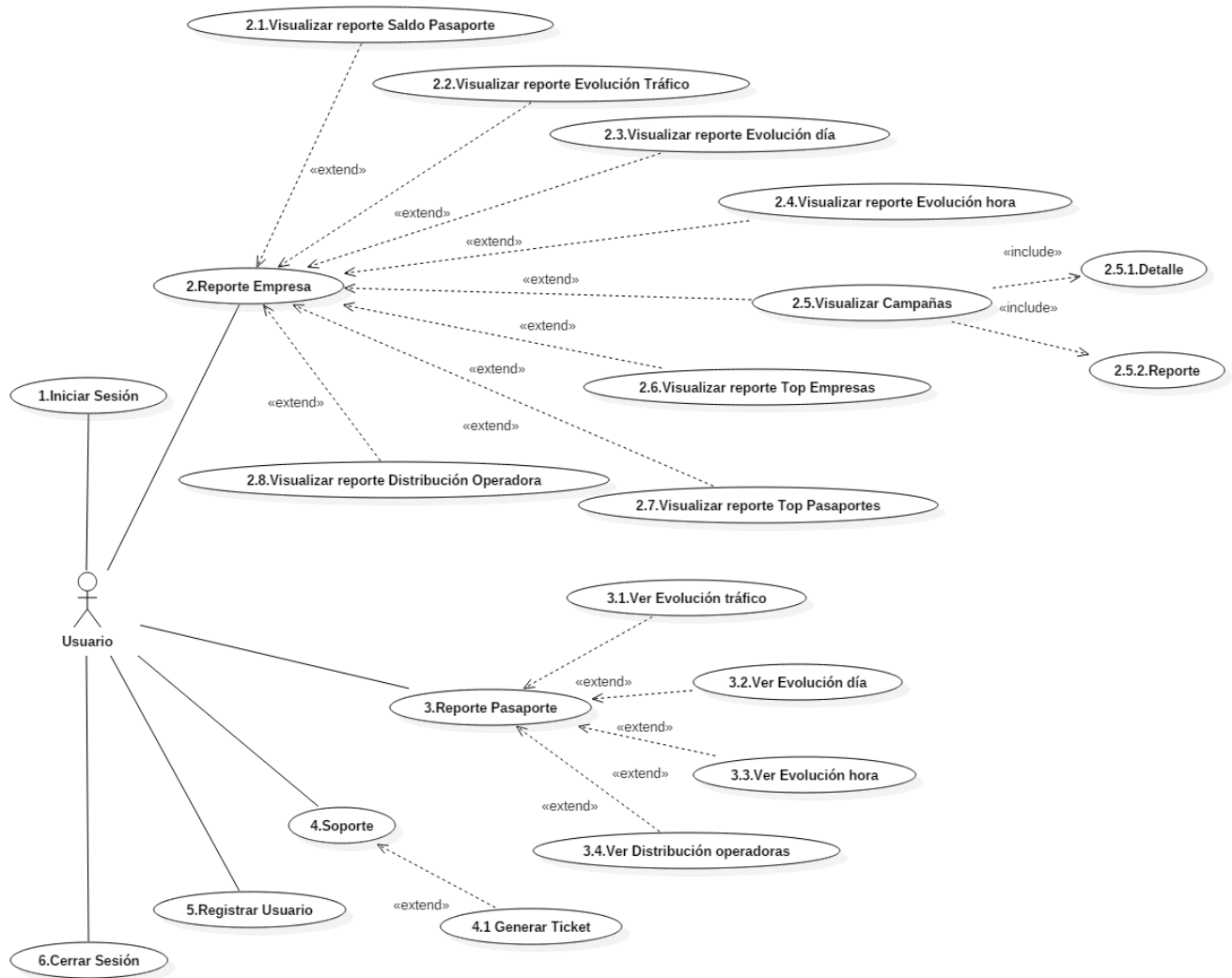


Figura 4.3: Diagrama de casos de uso.

Seguidamente, el *Product Owner* solicitó la elaboración de tres documentos: diccionario de datos, donde se muestra una versión inicial de la base de datos del sistema, dicha base de datos se implementó utilizando el manejador MongoDB por motivos de investigación de la empresa; el documento de vistas del sistema, en el que se muestran vistas tentativas de la interfaz del sistema; por último el documento de diagrama de despliegue el cual muestra dicho diagrama, elaborado siguiendo el prototipo especificado por la empresa, como se puede apreciar en la figura 4.4.

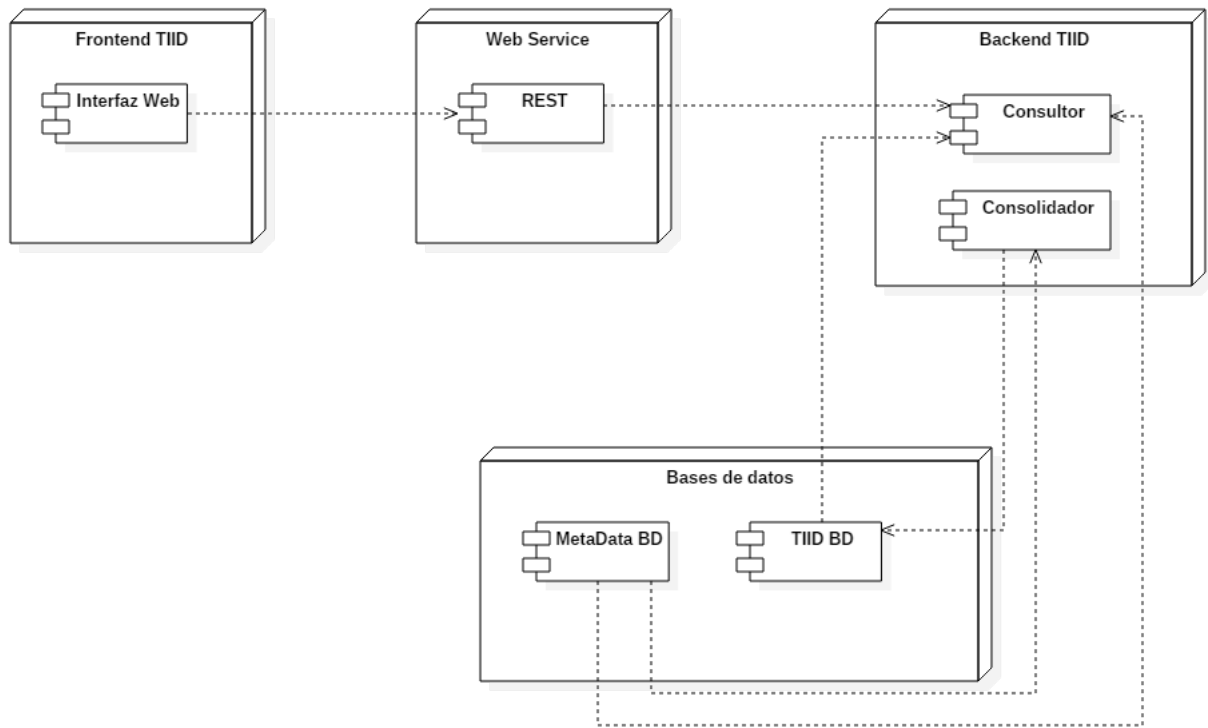


Figura 4.4: Diagrama de despliegue.

Este diagrama de despliegue 4.4 muestra de forma global como esta organizado el sistema, dividido en front-end, desarrollado con el *framework* AngularJS, servicio web desarrollados utilizando la arquitectura REST y back-end. El back-end posee dos apartados: el consolidador, que resume la información proveniente de una base de datos con información detalla, con el fin de optimizar los tiempos de respuesta en las consultas; y el consultor, el cual se encarga de realizar las consultas, provenientes de los servicios web, contra las bases de datos. Las bases de datos consultadas son: MetaDataDB de la que se utiliza la colección RTD (Reporte de tráfico detallado), la cual posee información detallada de todos los mensajes enviados por la empresa. Esta base de datos pertenece a otro sistema ajeno a TIID, la cual es utilizada cuando se activa el consolidador en el back-end ó cuando el cliente requiere reportes del día actual; y la base de datos TIID, la cual posee la información consolidad por cada usuario.

4.4. Selección de tecnologías

4.4.1. Objetivos

- Diseño de la solución global a implementar, parte 2.

4.4.2. Actividades

4.4.2.1. Selección de frameworks de desarrollo

Para el desarrollo del sistema, se estableció desde un principio que el Back-End debía ser desarrollado enteramente en java utilizando el IDE Netbeans, ya que es el lenguaje de programación y el entorno de desarrollo integrado mayormente utilizados dentro de Tedexis. De igual manera el desarrollo del servicio REST cumplió el mismo estandar.

Para el desarrollo del Front-End se probaron diferentes herramientas. El personal de Tedexis tenía mayor preferencia por frameworks de desarrollo basados en java por lo que se probaron Sprint, Play y Grails, haciendo simples implementaciones *Hola Mundo* con cada uno, sin embargo, se decidió utilizar AngularJS por la gran cantidad de documentación disponible y experiencia por parte del pasante con la herramienta. Con la utilización de dicho framework se lograron buenas prácticas de programación utilizando el patrón MVC (Modelo Vista Controlador), la arquitectura REST, el manejador de bases de datos MongoDB y el controlador de versiones Bitbucket.

4.5. Desarrollo del back-end

4.5.1. Objetivos

- Desarrollo del back-end.

4.5.2. Actividades

Este *sprint* muestra el desarrollo del back-end del sistema, el cual fue desarrollado enteramente en java, creandose como un proyecto tipo *Java Application* en *Netbeans* para tomar tu estructura de organización de archivos.

TIID tiene como función principal mostrar diferentes reportes a los usuarios sobre su uso del servicio de mensajería que les provee la empresa Tedexis, por lo cual tiene que manejar una gran cantidad de datos provenientes de la base de datos de la plataforma Mercurio. Esta plataforma consta de tres esquemas en PostgreSQL: Mercurio con seis tablas, DLR con doce tablas y Content con una tabla. Estos tres esquemas sirven como fuente de información para una base de datos en MongoDB, llamada MetaData la cual posee una colección que centraliza la información pertinente de los tres esquemas de la base de datos PostgreSQL. MetaData es una base de datos ajena a TIID, por lo que fue simulada durante el desarrollo del sistema, de igual manera, posee una colección que tiene por nombre RTD (Reporte de Tráfico Detallado), quien es la principal fuente de información que utiliza TIID para generar todos sus reportes.

Ya que la colección RTD tiene la información centralizada de todos los esquemas de Mercurio, posee información que en muchos casos no son pertinentes para los reportes de TIID, por lo cual se crearon dos hilos que consolidan la información necesaria para el sistema.

4.5.2.1. Diseño y creación de la base de datos

Se tomó como manejador de bases de datos MongoDB, debido a que Tedexis tiene futuros proyectos la migración de sus bases de datos actuales de PostgreSQL a MongoDB por la mayor velocidad en el manejo de grandes cantidades de datos, por lo que se decidió investigar con el proyecto de pasantía.

La base de datos de TIID consta de cinco colecciones, como se puede visualizar en el diagrama 4.5, las cuales se muestran a continuación:

- **user:** Almacena la información correspondiente a los usuarios de TIID. Posee los siguientes campos:
 - **user_id:** identificación del usuario dentro de la colección. Ejemplo: Alejandro.
 - **user_name:** nombre del usuario. Ejemplo: Alejandro.
 - **user_mail:** correo electrónico del usuario. Ejemplo: alejandro@gmail.com
 - **company:** corresponde a la compañía donde pertenece el usuario. Ejemplo: Tedexis.
 - **user_profile:** representa el perfil o privilegio del usuario. TIID posee dos perfiles, Tedexis (Administrador) y Canales. Ejemplo: Tedexis.

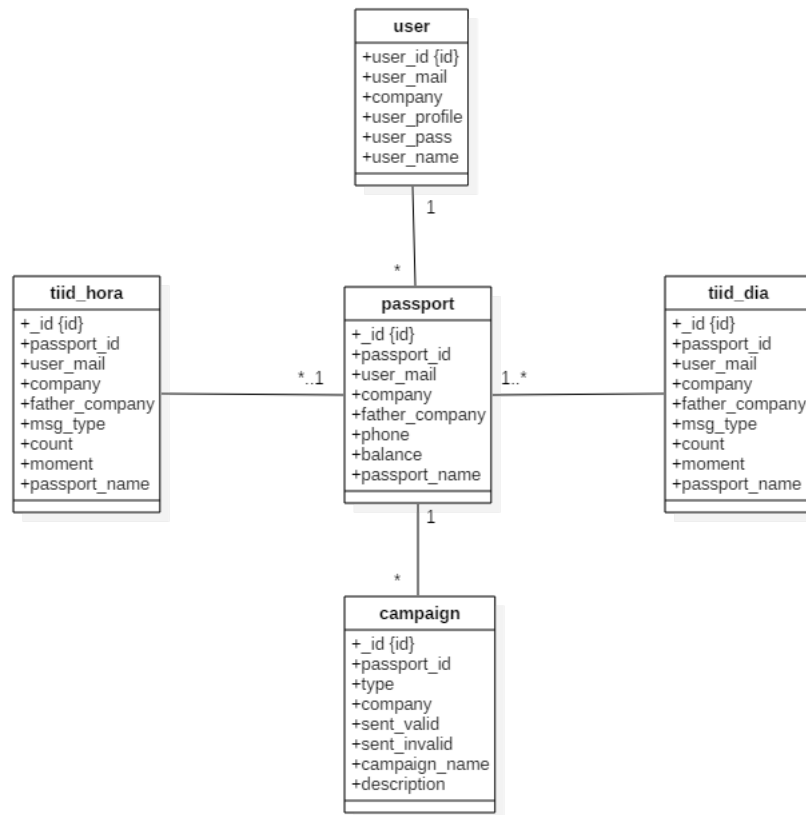


Figura 4.5: Diagrama de conceptual de base de datos.

- **user_pass**: contraseña del usuario.
- **passport**: Almacena la información de los pasaportes asociados a los usuarios de TIID. Los pasaportes son el equivalente a la identificación del usuario.
 - **_id**: identificación del pasaporte dentro de la colección. Creado automáticamente por el manejador.
 - **passport_id**: identificación de los pasaportes. Ejemplo: 42.
 - **passport_name**: nombre del pasaporte. Ejemplo: AlejandroLunch
 - **user_mail**: correo electrónico del usuario que le pertenece dicho pasaporte. Ejemplo: alejandro@gmail.com.
 - **company**: nombre de la compañía asociada al pasaporte. Ejemplo: Banco de Venezuela.
 - **father_company**: nombre de la empresa padre de la compañía asociada al pasaporte. Ejemplo: Tedexis.

- **phone**: número de contacto del propietario del pasaporte. Ejemplo: +584149322879.
- **balance**: saldo disponible del pasaporte. Ejemplo: 1000.
- **campaign**: Almacena los datos de las campañas de mensajería asociadas a las empresas.
 - **_id**: identificación de la campaña dentro de la colección. Creado automáticamente por el manejador.
 - **campaign_name**: nombre de la campaña. Ejemplo: Navidad 2016.
 - **descriptio**: descripción de la campaña. Ejemplo: Promociones para la época de-cembrina 2016.
 - **type**: tipo de mensajes enviados en la campaña. Ejemplo: MT.
 - **company**: nombre de la compañía a la que pertenece la campaña. Ejemplo: Banco de Venezuela.
 - **sent_valid**: número de mensajes enviados exitosamente. Ejemplo: 42.
 - **sent_invalid**: número de mensajes que no pudieron ser enviados. Ejemplo: 9.
- **tiid_hora**: Almacena la información por hora, pertinente para TIID, de la colección RTD de la base de datos MetaData.
 - **_id**: identificación del registro dentro de la colección. Creado automáticamente por el manejador.
 - **passport_id**: identificación del pasaporte que envió los mensajes. Ejemplo: 42.
 - **passport_name**: nombre del pasaporte que envió los mensajes. Ejemplo: AlejandroLunch
 - **user_mail**: correo electrónico del usuario que le pertenece dicho pasaporte. Ejemplo: alejandro@gmail.com.
 - **company**: nombre de la compañía asociada al pasaporte. Ejemplo: Banco de Venezuela.
 - **father_company**: nombre de la empresa padre de la compañía asociada al pasaporte. Ejemplo: Tedexis.
 - **msg_type**: tipo agrupado de los mensajes enviados. Ejemplo: MO.
 - **count**: cantidad de mensajes enviados a cierta fecha y hora. Ejemplo: 100.
 - **moment**: fecha y hora a que fueron enviados los mensajes. Ejemplo: 11-13-2016 16:00:53.
- **tiid_día**: Almacena la información por día, pertinente para TIID, de la colección RTD de la base de datos MetaData.
 - **_id**: identificación del registro dentro de la colección. Creado automáticamente por el manejador.

- **passport_id**: identificación del pasaporte que envió los mensajes. Ejemplo: 42.
- **passport_name**: nombre del pasaporte que envió los mensajes. Ejemplo: AlejandroLunch
- **user_mail**: correo electrónico del usuario que le pertenece dicho pasaporte. Ejemplo: alejandro@gmail.com.
- **company**: nombre de la compañía asociada al pasaporte. Ejemplo: Banco de Venezuela.
- **father_company**: nombre de la empresa padre de la compañía asociada al pasaporte. Ejemplo: Tedexis.
- **msg_type**: tipo agrupado de los mensajes enviados. Ejemplo: MO.
- **count**: cantidad de mensajes enviados a cierta fecha sin hora. Ejemplo: 100.
- **moment**: fecha sin hora a la que fueron enviados los mensajes. Ejemplo: 11-13-2016.

4.5.2.2. Creación de hilos consolidadores de información

El sistema TIID se alimenta de los datos proveniente de la colección RTD, como fue antes mencionado, pero dicha colección posee mucha información no relevante para el sistema lo que ocasiona tiempo de respuestas muy largos y desesperación por parte de los usuarios al consultar su información. Por lo que para optimizar la experiencia del usuario, mejorar las consultas de bases de datos y poseer tiempos cortos de respuesta al momento de la carga de datos en el front-end del sistema, se decidió poseer la mayor cantidad de información procesada dentro de la base de datos de TIID.

Para la consolidación de los datos de RTD, se crearon dos hilos:

- **ThreadDayConsolidator**: esta clase fue creada como extensión de la clase *Thread* de java, lo que permite redefinir el método *run* para luego instanciar la clase y poner en ejecución el hilo con el método *start*. Este hilo lo que hace es recorrer la colección *passport* antes mencionada y por cada pasaporte totaliza la cantidad de mensajes MO, MT y MR enviados por día, mostrados en la colección RTD. Esta clase se apoya en una clase llamada *Mongo* la cual a partir de un archivo de propiedades proporcionado, realiza las conexiones a las bases de datos MongoDB.
- **ThreadHourConsolidator**: esta clase es análoga a la clase *ThreadDayConsolidator* solo que totaliza los mensajes por hora en vez de por día. Es importante mencionar que

ambos hilos se activan una vez por día y consolidan los datos correspondiente al día anterior, por lo que si se realizan consultas del día actual, dichas consultas se llevan acabo directo en colección RTD.

4.5.2.3. Pruebas

Para las prueba de esta sección, primero se creó la bases de datos TIID en MongoDB con la colección *passport* y luego se simuló la colección RTD de la base de datos MetaData. Ambas fueron llenadas mediante el comando *mongoimport* [17] con datos generados de forma aleatoria. Se pusieron a prueba ambos hilos por separado, primero *ThreadDayConsolidator* y luego *ThreadHourConsolidador* probando primero con 100 entradas en la colección RTD y 10 en la colección *passport*, realizando la consolidación de la información sin ningún inconveniente. Luego se aumentó la cantidad de datos con 100000 entradas en RTD y 200 en *passport*, superando la prueba sin ningún inconveniente.

De igual manera, se realizaron pruebas de borde donde se simuló la activación de los hilos a media noche (12:00 am ó 00:00:00), donde se encontraron algunos inconvenientes que no tomaba en cuenta los mensajes enviados justo a media noche pero fue solucionado con un ajuste en el rango de las fechas.

Finalmente se hicieron pruebas de integración, donde se activaron los dos hilos al mismo tiempo, agregando desde 100 hasta 100000 datos en RTD y desde 10 hasta 200 en *passport*, lo que fue superado sin ningún inconveniente. Dichas pruebas fueron mostradas y aprobadas tanto por el *Product Owner* como el *Scrum Master*.

4.6. Autenticación

4.6.1. Objetivos

- Desarrollo del módulo de autenticación.

4.6.2. Actividades

4.6.2.1. Creación del servicio Web RESTful

Para la creación del módulo de autenticación, primero se creó el servicio REST como un proyecto java utilizando Netbeans. Utilizando las herramientas proporcionadas por el entorno de desarrollo integrado, se instaló y configuró el servidor GlashFish 4.2, que es un servidor *open source* para el desarrollo y despliegue de plataformas y tecnologías basadas en java. Al haber instalado el servidor, se procedió a instalar las librerías necesarias para poder invocar el API *Jersey* que proporciona los mecanismos indispensables para el desarrollo de los servicios REST.

4.6.2.2. Autenticación del usuario

Al tener el proyecto REST creado, se procedió al desarrollo del módulo de autenticación. Se utilizó un módulo basado en *tokens* [24] para AngularJS llamado *Satellizer* igualmente *open source* [23]. Con *Satellizer* se logró implementar de forma sencilla el inicio de sesión, cierre de sesión y creación de usuarios para el sistema. En la creación de usuarios se puede elegir entre dos perfiles, Tedexis (administrados) y Canales (usuario regular), diferenciandose en que el perfil Tedexis puede crear nuevos usuarios del sistema mientras que Canales no. Cabe destacar que *Satellizer* permitió una fácil integración con el manejador MongoDB, se partió de un ejemplo desarrollado en java con el manejador H2 el cual fue posteriormente adaptado. A continuación se presentan las vistas del inicio de sesión 4.6 y creación de usuarios 4.7.

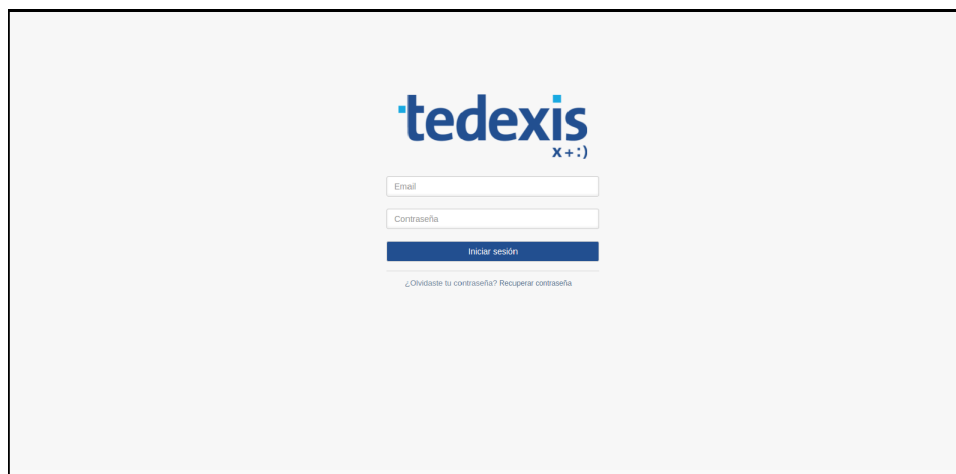


Figura 4.6: Vista de inicio de sesión.

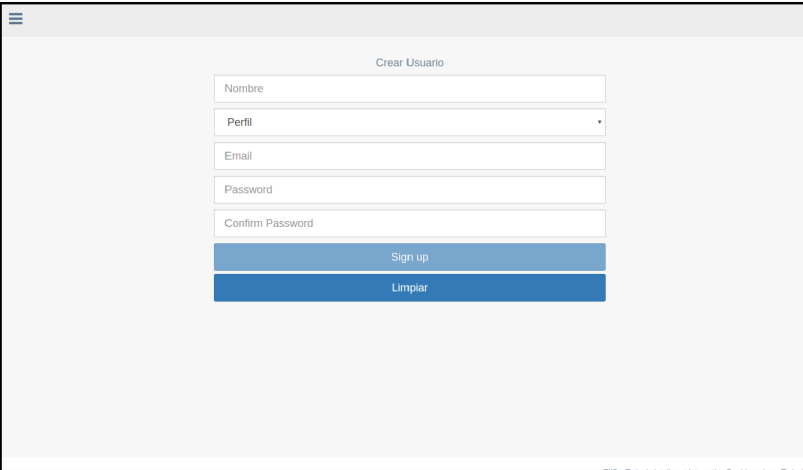


Figura 4.7: Vista de creación de usuario.

Posteriormente se desarrolló la recuperación de contraseña del usuario, el cual dado un correo electrónico verifica si existe en la base de datos y genera una nueva contraseña, la cual es enviada por correo electrónico al usuario.

4.6.2.3. Pruebas

Al crear el proyecto RESTful se crearon métodos GETs de prueba para verificar que se estaba accediendo a la base de datos y las colecciones correspondientes. Seguidamente se integraron los métodos POSTs necesarios para el inicio de sesión y creación de usuario, con el front-end del sistema. Se realizaron distintas pruebas de creación de usuarios y de inicio de sesión donde surgieron las restricciones de no poder crear un nuevo usuario con un correo electrónico ya existente en el sistema y el cifrado de las contraseñas, lo cual se logró con una librería que proporciona *Satellizer*.

4.7. Inicio del desarrollo de Reportes por Empresa

4.7.1. Objetivos

- Desarrollo de reportes por empresa, parte 1.

4.7.2. Actividades

En este *Sprint* se empezó con el desarrollo a fondo del front-end del sistema. El equipo de mercadeo de Tedexis proporcionó una serie de documentos con los recursos visuales utilizados por la empresa, íconos, colores, logos, entre otros. y una plantilla llamada Gentelella [25], la cual sirvió como base para el desarrollo de las vistas de TIID.

4.7.2.1. Creación de vista Saldo Pasaporte

La vista inicial que se creó en el sistema, muestra una tabla con todos los pasaportes asociados al usuario cuya sesión está activa, como se puede apreciar en figura 4.8. En la creación de esta vista surgieron distintos problemas, ya que fue la primera tabla implementada combinando la librería proporcionada por la plantilla Gentelella con los datos provenientes de la llamada de servicio RESTful. Dicha llamada es asincrónica al momento de la carga de la página, lo que ocasionaba que en algunas ocasiones no se mostraran los datos, esto se resolvió utilizando promesas como cuerpo de las tablas al momento de su creación.

Número Pasaporte	Nombre Pasaporte	Email	Empresa	Saldo
82	dtv	1@gmail.com	584149333333	400
83	NoExiste	1@gmail.com	5.8414933	500

Figura 4.8: Reporte Empresa: Saldo Pasaporte.

4.7.2.2. Creación de vista Evolución del Tráfico

Esta vista muestra el reporte de mensajes tipo MT y MR enviados por día por la empresa asociada al usuario en un rango de tiempo específico, el usuario puede elegir dicho rango de

tiempo mediante un *widget* de calendario proporsionado y puede descargar tanto el gráfico mostrado en un una imagen png, como la información del reporte mostrada en la tabla en un archivo de formato csv. De igual manera, el usuario puede aplicar distintos filtros y métodos de búsqueda proporsionados por las funcionalidades de la tabla. Vea la figura 4.9.

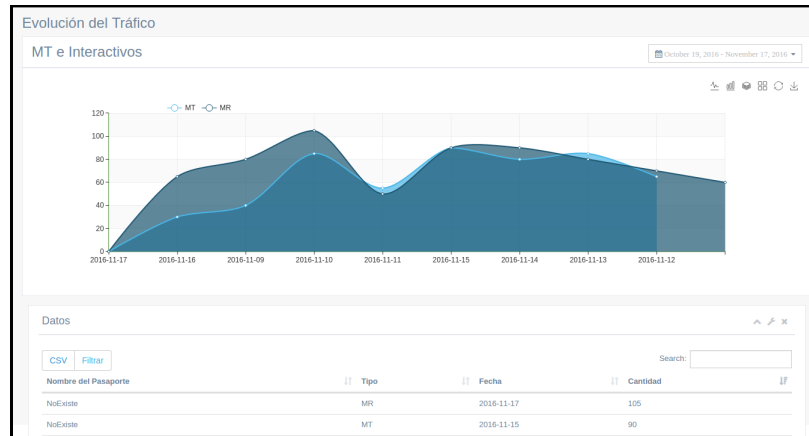


Figura 4.9: Reporte Empresa: Evolución del Tráfico.

Para el desarrollo de esta vista primero se crearon los servicios RESTful que consulta la información consolidada en la colección “tiid_dia” de la base de datos de TIID tanto para el gráfico como la tabla. Luego se escribieron las funciones para generar el gráfico y la tabla, incorporando el rango de fechas proporsionado por un *widget* calendario.

4.7.2.3. Creación de vistas Evolución Día y Evolución Hora

La vista *Evolución Día* es análoga a la vista *Evolución del Tráfico* solo que muestra la información consolidada por día de los mensajes tipo MT y MO, mostrada en la figura 4.10. En cambio la vista *Evolución Hora* sigue la misma lógica que las dos anteriores pero consulta la información consolidad por hora de la colección “tiid_hora”, por lo que se crearon los servicios de consulta REST respectivos. La figura 4.11 muestra el reporte por hora.

4.7.2.4. Pruebas

Las pruebas en este *Sprint* se realizaron a medida que se fueron desarrollando cada vista. Primero se hicieron pruebas unitarias del servicio REST utilizado en la vista *Saldo Pasaporte*, luego se hicieron pruebas de integración del servicio con la tabla mostrada en el fron-end. Las pruebas estan explicitas en el documento de *Scrum* realizado, pero por politicas de confidencialidad de la empresa no pudieron ser presentados en este documento.

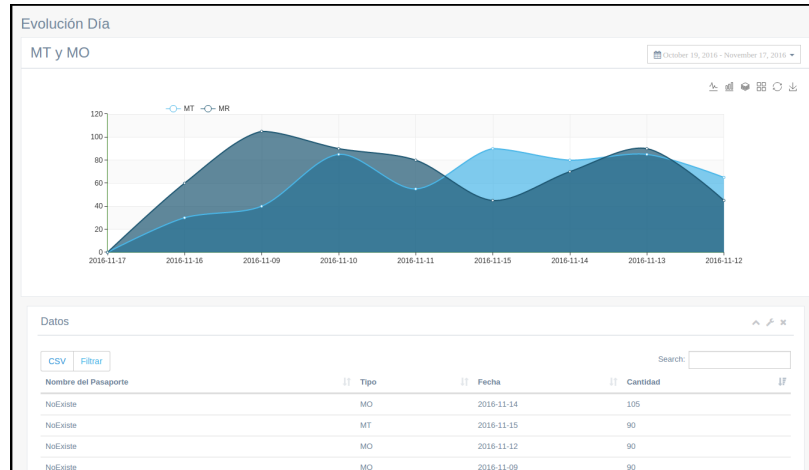


Figura 4.10: Reporte Empresa: Evolución Día.

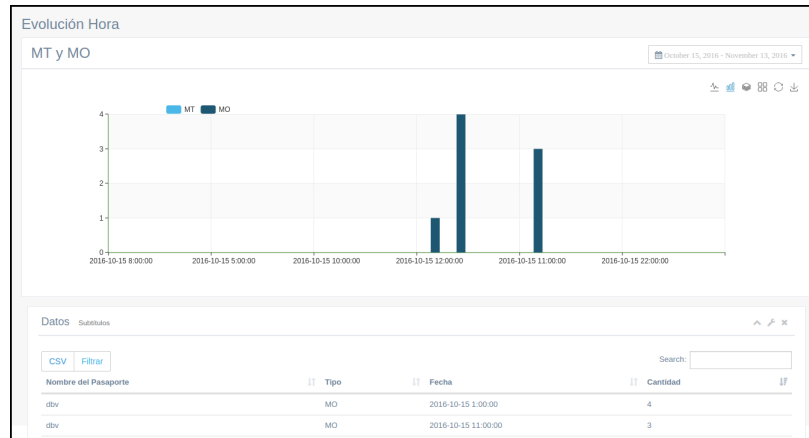


Figura 4.11: Reporte Empresa: Evolución Hora.

Posteriormente se realizaron las pruebas unitarias de los servicios REST utilizados en las vistas *Evolución del Tráfico* y *Evolución Día*. Dichas vistas comparten los mismos servicios. Se realizaron pruebas unitarias verificando que se estaba consultado la colección pertinente y que la presentación de la información fuera la indicada para que pudiera ser utilizada sin mucha modificación en el front-end del sistema. Seguidamente, se realizaron pruebas de integración con el *widget* calendario, las consultas de los servicios y el despliegue de los datos tanto gráficos como en la tabla. En estas pruebas no se presentaron inconvenientes. De la misma manera se hicieron las pruebas de los servicios REST utilizados en el reporte por hora y sus respectivas pruebas de integración con las funcionalidades proposionadas en el front-end.

4.8. Fin del desarrollo de Reportes por Empresa

4.8.1. Objetivos

- Desarrollo de reportes por empresa, parte 2.

4.8.2. Actividades

4.8.2.1. Desarrollo de vistas Top Empresas y Top Pasaportes

Tanto *Top Empresas* como *Top Pasaportes* muestran las 5 primeras empresas y pasaporte, respectivamente, con mayor cantidad de mensajes enviados en un rango de tiempo establecido. Los reportes estan divididos en tres *Top MO*, *Top MT* y *Top Interactivos*, cada uno con su gráfico y su tabla de datos, como son mostrados en las figuras 4.12 y 4.13.

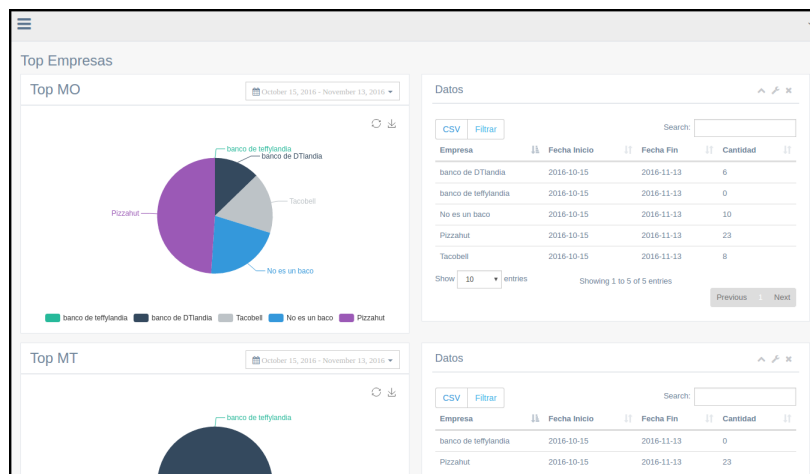


Figura 4.12: Reporte Empresa: Top Empresa.

En este *Sprint* se procedió de manera similar al anterior, primero se desarrollaron los servicios REST para la obtención de datos para ambos reportes y luego se integraron cada uno con sus respectivos graficos y tablas. Finalmente se integraron los *widget* calendario en cada reporte.

4.8.2.2. Desarrollo de vistas Campañas y Distribución Operadoras

La vista *Campañas* muestra todas las campañas de mensajerías asociadas a la empresa a la cual pertenece el usuario. Como se puede observar en la figura 4.14, es una tabla que muestra la información pertinente de las campañas de la empresa, proveniente de la llamada

a un servicio REST.

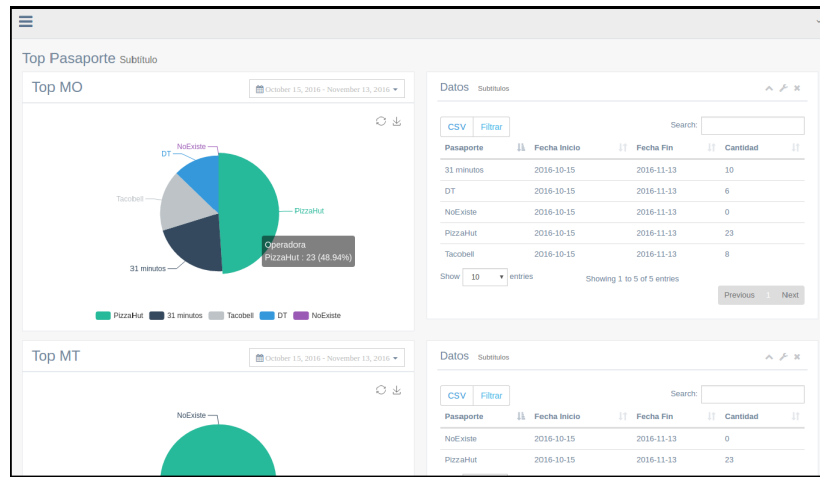


Figura 4.13: Reporte Empresa: Top Pasaporte.

Muestras que *Distribución Operadoras* es un reporte parecido a *Top Empresas*, donde se muestra la cantidad de mensajes enviados por cada operadora asociada a dicha empresa. El reporte de la distribución de las operadoras esta dividido en tres secciones MO, MT e Interactivos, de igual manera que los repotes de *Top Empresa* y *Top Pasaportes*.

Campañas

Campañas asociadas a la empresa:

Campaña	Descripción	Tipo	Empresa	Enviados Válidos	Enviados Inválidos
Navidad 2016	MTs de navidad 2016	Informativo	bando de venezuela	30	10
Proeba2	2016	Informativo	bando de venezuela	42	10

Figura 4.14: Reporte Empresa: Campañas.

4.8.2.3. Pruebas

Las pruebas realizadas en este *Sprint* se realizaron de manera similar que las anteriores. Primero se realizaron pruebas unitarias a cada servicio REST y luego se integraron con lo desarrollado en el front-end. En este caso se tuvieron algunas dificultades en principio con el desarrollo de *Top Empresas*, debido a que es una vista con más de una tabla y más de un gráfico por lo que se generaban algunos conflictos de superposición de gráfico y ciertos filtros no ejercían su función en la tabla correspondiente. Esto se solucionó creando un controlador diferente por cada conjunto gráfico-tabla de cada reporte, es decir, en el caso de *Top Empresas* se crearon 3 controladores diferentes, uno para *Top MO*, otro para *Top MT* y otro para *Top Interactivos*. Esta misma lógica fue aplicada en los reportes *Top Pasaportes* y *Distribución Operadoras*.

4.9. Desarrollo de Reportes por Pasaporte

4.9.1. Objetivos

- Desarrollo de reportes por pasaporte.

4.9.2. Actividades

4.9.2.1. Desarrollo de vistas de reporte de tráfico

Se inició desarrollando la vista *Evolución del Tráfico* para el reporte por pasaporte. Este reporte es similar al realizado anteriormente en los reportes por empresa, con la diferencia que primero se debe elegir un pasaporte del usuario en sesión, mostrados en una tabla al inicio de la vista, y luego se muestra el tráfico de dicho pasaporte, como se puede observar en la figura 4.15.

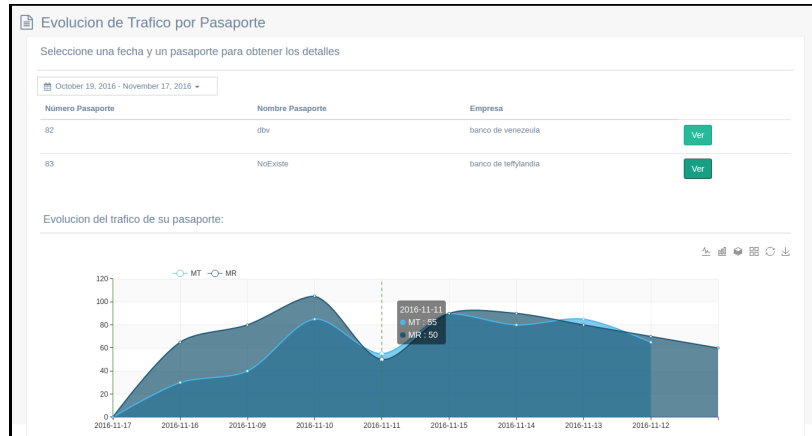


Figura 4.15: Reporte Pasaporte: Evolución del Tráfico.

La información mostrada en los reportes *Evolución del Tráfico*, *Evolución Día* y *Evolución Hora* son análogos a sus versiones en el apartado *Reporte Empresa*, correspondiendo a tráfico de mensajes por día de los tipos MT y MR para *Evolución del Tráfico* y de los tipos MO y MT para *Evolución Día*, mostrado en la figura 4.16, mientras que *Evolución Hora* al seleccionar un pasaporte, muestra el tráfico por hora de los mensajes tipo MT y MO, como se observa en al figura 4.17.

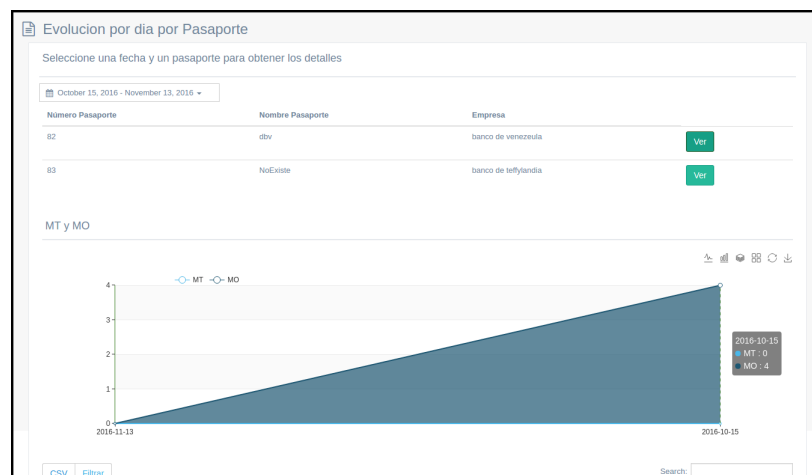


Figura 4.16: Reporte Pasaporte: Evolución Día.

Para el desarrollo de estas vistas, se tomaron como base los servicios REST creados para los reportes por empresa, los cuales fueron adaptados para que hicieran las consultadas a las colecciones consolidadas en base, utilizando los pasaportes en vez de las empresas. De igual manera, para el front-end se tomaron como plantillas las vistas de los reportes por empresa,

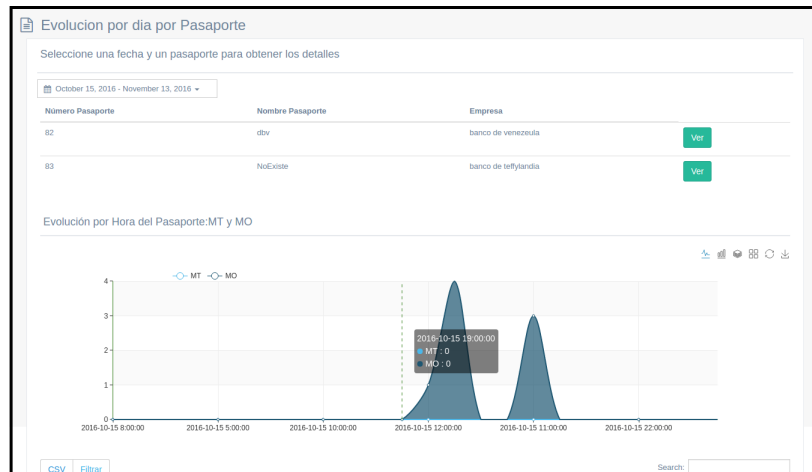


Figura 4.17: Reporte Pasaporte: Evolución Hora.

se le agregaron una tabla nueva donde muestran los pasaportes pertenecientes al usuario y se crearon las funcionalidades pertinentes para integración de todo. No resultó ningún inconveniente en el desarrollo.

El desarrollo de *Distribución Operadoras* se realizó de manera similar a lo expuesto anteriormente, se tomaron el servicio REST y la plantilla de la vista del reporte *Distribución Operadora* por empresa y fueron adaptados de la misma forma. En el servicio REST fueron modificadas las consultas ya no por empresa sino por pasaporte y a la vista se le agregó una nueva tabla donde se elige un pasaporte y se muestra el reporte de la distribución de mensajes, enviados por dicho pasaporte, por operadora.

4.9.2.2. Pruebas

Las pruebas se iniciaron con pruebas unitarias a los nuevos servicios REST desarrollados, verificando que las consultas estuvieran siendo realizadas en base al pasaporte suministrado y no en base a la empresa. Seguidamente, se realizaron pruebas de integración de las nuevas tablas que contienen todos los pasaportes asociados a un usuario y verificando que la información mostrada por pantalla, fueran los reportes pertenecientes al pasaporte seleccionado anteriormente. No resultaron ningún inconveniente en estas pruebas. De igual manera, las pruebas fueron mostradas y aprobadas por el *Product Owner* y el *Scrum Master*.

4.10. Desarrollo de ultimas vistas del front-end

4.10.1. Objetivos

- Desarrollo la vista principal y soporte.

4.10.2. Actividades

Con este *Sprint* se finalizó el desarrollo del sistema, antes de ser montado en un servidor local dentro de Tedexis para que fuera probado por el personal de la empresa.

4.10.2.1. Vista principal

Para la vista principal se solicitó que tuviera un resumen de los reportes por empresa asociado al usuario, mostrando directamente distintos gráficos y la cantidad de mensajes enviados por ese usuario en el mes en curso. Ya que se deseaba que el usuario pudiera visualizar directamente información pertinente desde su entrada al sistema. Como se puede observar en la figura 4.18. Dichos requerimientos fueron cumplidos utilizando los servicios REST creados durante el desarrollo, al igual que las mismas funciones para la creación de gráficos.



Figura 4.18: Vista principal.

4.10.2.2. Soporte

El apartado de soporte se posee un formulario llamado *Generar ticket*, el cual al ser completado con la información requerida, envía un correo al Departamento de Soporte de Tedexis para la gestión de la solicitud del cliente. Se puede visualizar esta vista en la figura 4.19.

Figura 4.19: Vista soporte.

4.11. Pruebas por el personal de la empresa

4.11.1. Objetivos

- Pruebas y certificación.

4.11.2. Actividades

4.11.2.1. Pruebas del equipo de producto

A medida que se fueron realizando los sprint de desarrollo, se fueron realizando pruebas unitarias y de integración tanto en el front-end, como en el back-end y en los servicios rest, dependiendo de lo desarrollado en el sprint.

Posteriormente, el *Scrum Master* decidió montar localmente en un servidor de la empresa el sistema realizado con el fin de recibir mayor retroalimentación por parte del personal

de Tedexis. Inicialmente fue probador por el equipo del Departamento de Producto (los que desarrollan las aplicaciones dentro de la empresa), quienes se enfocaron mayormente en el código realizado, proveyendo recomendaciones de refactorización y documentación del código, específicamente en el back-end y en los servicios rest. De igual manera, propusieron ideas para la optimización de consultas de bases de datos de MongoDB, donde se utilizaron expresiones regulares y métodos de agrupación proporcionados por dicho manejador.

4.11.2.2. Pruebas del equipo de comercio

Seguidamente de las pruebas realizadas por el Departamento de Producto, se les informó al Departamento de Comercio que podían ingresar al sistema y realizaran las pruebas pertinentes. La opinión de este departamento es sumamente importante debido a que son quienes más utilizarían el sistema, además de ser quien lo vende a los clientes de la empresa.

Las observaciones por parte de este departamento fueron en su mayoría enfocadas en el fron-end del sistema, solicitando cambios en la paleta de colores utilizada, la cual ayudaría a asociar al cliente con la empresa de forma más visual y directa; solicitud de que la información exportada de los diferentes reportes solo fuera en formato csv; solicitud de poder descargar los gráficos mostrados en los diferentes repotes en forma de imagenes; solicitud de poder cambiar entre gráfica de barra y gráfica de puntos en los reportes de tráfico de mensajes.

4.12. Redacción del libro de pasantía

4.12.1. Objetivos

- Redacción de libro de pasantía.

4.12.2. Actividades

4.12.2.1. Redacción del libro y versión final de los documentos de la empresa

Como fue propuesto en el plan de trabajo, las últimas dos semanas fueron dedicadas a la redacción del libro de pasantías, adicionalmente, se actualizaron los documentos solicitados por la empresa, que pueden ser encontrados en los apendices.

Capítulo 5

Conclusiones y Recomendaciones

Al culminar el tiempo estipulado para el desarrollo de la pasantía, se puede afirmar que los objetivos iniciales fueron alcanzados en su mayoría y superados en muchos casos, obteniendo un producto final que satisface las necesidades tanto de los clientes como el personal de la empresa que hace uso del sistema.

La versión TIID supera en velocidad de acceso y respuesta a su versión anterior TID. El nuevo sistema posee una interfaz gráfica agradable y cómoda para el usuario, quien tiene acceso a una diversidad de reportes relevantes que lo mantienen al tanto de su actividad con la empresa. Dichos reportes están divididos en dos apartados: Reportes por Empresa, donde se muestran distintos reportes basados en la información proveniente de todos los clientes asociados a la empresa del usuario; y Reportes por Pasaporte, donde se muestran en detalle distintos reportes referentes a cada pasaporte asociado al usuario. Los reportes mencionados poseen distintos mecanismos de búsqueda y filtros para su mejor visualización, de igual manera, pueden ser exportados si el usuario lo desea.

TIID se divide en back-end, front-end y servicios web, donde el back-end se encarga de la consolidación de los mensajes enviados por la plataforma Mercurio en un día, realizado por dos hilos: uno que consolida los mensajes enviados por día y otro que lo hace por hora, esto se almacena en una base de datos NoSQL. Dicha consolidación se lleva a cabo para el rápido acceso de información en las consultas realizadas por los servicios web, que en este caso fueron implementados siguiendo la arquitectura REST, quienes proveen los datos consultados en formato tipo JSON. Finalmente dicha información es desplegada en el front-end del sistema como reportes en forma de gráficas y tablas de datos, los cuales pueden ser consultados

en diferentes rangos de tiempo y descargados si el usuario lo desea.

En cuanto a la metodología de desarrollo se puede afirmar, que *Scrum* resultó ser un buen mecanismo de desarrollo, permitiendo que se realizara de manera organizada, manteniendo constante comunicación con el *Product Owner* y el *Scrum Master* y sobre todo, la rápida y fluida adaptación durante el desarrollo a cambios, sugerencias e imprevistos surgidos.

Como recomendaciones para la empresa se puede mencionar: mejorar la comunicación entre los Departamentos Comercio, Producto y Operaciones, ya que muchas veces proveían diferente información sobre un mismo requerimiento del producto, por lo que había que invertir tiempo adicional en definir nuevamente los puntos y llegar a nuevos acuerdos; disposición a utilizar nuevas tecnologías, debido a que todo lo desarrollan utilizando java y están un poco cerrados a probar diferentes tecnologías que les pueden llegar a ser muy útiles; y realizar y mantener organizada documentación relevante de los productos desarrollados por la empresa.

Esta experiencia sirvió para integrar al pasante en un ambiente laboral real, donde se le permitió un diseñar y desarrollar un producto que resuelve una problemática que sufre actualmente la empresa. Integrarlo en una situación donde el trabajo en equipo y la comunicación fueron pilares fundamentales para el logro de las metas propuestas. De igual manera, permitió al pasante indagar y aprender nuevas tecnologías y herramientas utilizadas en producción por muchas compañías. Una experiencia integra, que permitió el desarrollo tanto personal como profesional.

Finalmente, queda como trabajo futuro el desarrollo a fondo del módulo de autenticación del sistema, desarrollando una mayor cantidad de perfiles y toda la lógica que conlleva y la integración de otros dos sistemas como módulos dentro de TIID, dichos sistemas son: Reporte del tráfico detallado (RTD) y el sistema de programación de envío masivo de mensajes (TrinityKronos), los cuales no fueron desarrollados en este proyecto de pasantías.

Bibliografía

- [1] Tedexis. <https://http://tedexis.com>, consultado el 25 de septiembre de 2016.
- [2] Wikipedia, The Free Encyclopedia. Software Framework. https://en.wikipedia.org/wiki/Software_framework, consultado el 25 de septiembre de 2016.
- [3] THE MVC-WEB DESIGN PATTERN. <http://grove.cs.jmu.edu/groverf/papers/WEBIST2011.pdf>, consultado el 25 de septiembre de 2016.
- [4] Wikipedia, The Free Encyclopedia. Servicio Web. https://es.wikipedia.org/wiki/Servicio_web, consultado el 25 de septiembre de 2016.
- [5] Introducing JSON. <http://www.json.org/>, consultado el 25 de septiembre de 2016.
- [6] TechTarget. REST. <http://searchsoa.techtarget.com/definition/REST>, consultado el 25 de septiembre de 2016.
- [7] What is Java?. https://java.com/en/download/faq/whatis_java.xml, consultado el 25 de septiembre de 2016.
- [8] JavaScript. <https://developer.mozilla.org/es/docs/Web/JavaScript>, consultado el 25 de septiembre de 2016.
- [9] What is AngularJS?. <https://docs.angularjs.org/guide/introduction>, consultado el 25 de septiembre de 2016.
- [10] Reinventando la gestión de datos. <https://www.mongodb.com/es>, consultado el 25 de septiembre de 2016.
- [11] Bower a package manager for the web. <http://bower.io/>, consultado el 25 de septiembre de 2016.

- [12] HTML. <https://developer.mozilla.org/es/docs/Web/HTML>, consultado el 25 de septiembre de 2016.
- [13] Programa, gestiona, colabora, Bitbucket es la solución Git para equipos profesionales. <https://es.atlassian.com/software/bitbucket>, consultado el 25 de septiembre de 2016.
- [14] What is NPM?. <https://docs.npmjs.com/getting-started/what-is-npm>, consultado el 25 de septiembre de 2016.
- [15] Node.js. <https://nodejs.org/en/>, consultado el 25 de septiembre de 2016.
- [16] The Definitive Guide to Scrum. <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf#zoom=100>, consultado el 26 de septiembre de 2016.
- [17] Mongoimport. <https://docs.mongodb.com/manual/reference/mongoimport/>, consultado el 5 de octubre de 2016.
- [18] Clase de Hilos. <http://ldc.usb.ve/~yudith/docencia/ci-3825/ClaseThreads.pdf>, consultado el 8 de octubre de 2016.
- [19] DBCursor class. <https://api.mongodb.com/java/2.6/com/mongodb/DBCursor.html>, consultado el 10 de octubre de 2016.
- [20] JAX-RS. <https://jax-rs-spec.java.net/>, consultado el 12 de octubre de 2016.
- [21] Jersey - RESTful Web Services in Java. <https://jersey.java.net/>, consultado el 12 de octubre de 2016.
- [22] Esquema en PostgreSQL. <https://www.postgresql.org/docs/9.1/static/ddl-schemas.html>, consultado el 12 de octubre de 2016.
- [23] Satellizer, Token-based AngularJS Authentication. <https://github.com/sahat/satellizer>, consultado el 19 de octubre de 2016.
- [24] Autenticacion basada en Tokens. <http://koldohernandez.com/autenticacion-basada-en-tokens/>, consultado el 19 de octubre de 2016.
- [25] Gentelella. <https://github.com/puikinsh/gentelella>, consultado el 19 de octubre de 2016.

Apéndice A

Documento de Definición de Requerimientos

	DDR: TIID	Código: TDX-DP-PR-D-08		Fecha: 31/05/2016
		Versión: 02	Edición: 01	Páginas: 1 de 7

Hoja de trazabilidad del documento

Versión	Fecha	Descripción	Responsable de la Actualización	Responsable de la Revisión	Responsable de la Aprobación
01	04/08/2016	Modificación de Requerimientos funcionales y no <u>funcionales</u>	<u>Alejandro Guevara</u> Pasante	<u>Robinson Rodríguez</u> Consultor de Productos	<u>Yermi Echeverría</u> Coordinador de Productos <u>Vito Mastrogiacomio</u> Director General
01	31/05/2016	Creación de Documento	<u>Jeselys Hernández M.</u> Director de Producto	<u>Robinson Rodríguez</u> Consultor de Productos <u>Yermi Echeverría</u> Coordinador de Productos <u>Ninoska González</u> Gerente de Mercadeo	<u>Elizabeth Navas</u> Director Comercial VE <u>Roberto Angulo</u> Director Comercial TDX <u>Jeselys Hernández M.</u> Director de Producto <u>Vito Mastrogiacomio</u> Director General

	DDR: TIID	Código: TDX-DP-PR-D-08		Fecha: 31/05/2016
		Versión: 02	Edición: 01	Páginas: 2 de 7

Tabla de Contenido

1. Introducción	3
2. Objetivo.....	3
3. Acrónimos, definiciones o términos	3
4. Descripción del producto	4
5. Necesidades	4
6. Premisas.....	5
7. Expectativas de entrega.....	6
8. Especificaciones del sistema: Requerimientos funcionales	6
9. Especificaciones del sistema: Requerimientos no funcionales	6

	DDR: TIID	Código: TDX-DP-PR-D-08		Fecha: 31/05/2016
		Versión: 02	Edición: 01	Páginas: 3 de 7

1. Introducción

En búsqueda de mejoras y obtener nuevos conocimientos hemos estado realizando laboratorios de investigación que nos permitan aprender nuevas tecnologías o nuevas formas de hacer las cosas. En este sentido, este año y como una opción para sobrevivir al déficit profesional que está sufriendo el país estamos incorporando en nuestro portafolio cuatro proyectos que serán llevados a cabo por estudiantes de la Universidad Simón Bolívar durante un período de 20 semanas.

TIID o Tedexis Intelligence and Interactive Dashboard (TID v2) es uno de los proyectos que estará siendo ejecutado por un pasante. El objetivo principal es generar documentación relevante de la aplicación y construir las bases de una nueva arquitectura orientada a servicios. Además, de una actualización tecnológica para el desarrollo de la aplicación web con la que contamos actualmente.

En este documento se pretende definir las necesidades del TIID y las expectativas de entrega.

2. Objetivo:

Definir las necesidades que deben ser atendidas en el desarrollo de la pasantía de TIID.

3. Acrónimos, definiciones o términos

Acrónimo, definición o término	Descripción
SMS (Short Message Service)	Servicio que puede enviar o recibir mensajes entre celulares y otros dispositivos electrónicos e incluso utilizando internet
SMPP (Short Message Peer-to-peer Protocol)	Protocolo estándar de telecomunicaciones pensado para el intercambio de SMS entre equipos que gestionan los mensajes como pueden ser los SMSC o los GSM USSD (Unstructured Supplementary Services Data server) y un sistema de solicitud de SMS como puede ser un servidor WAP o cualquier gateway de mensajería.
SFTP (Secure File Transfer Protocol)	Protocolo para transferencia (carga y descarga) segura de archivos.
Mensaje Masivo	Representa un lote grande de mensajes de texto. Son enviados mediante cargas de archivos de texto.
Mensaje Interactivo	Está formado por un MO y MT. En Tedexis suele llamarse mensaje interactivo a los mensajes manipulados por las campañas de mensajería.
MO (Mobile Originated)	Mensaje originado desde un teléfono celular
MT (Mobile Terminated)	Mensaje recibido en un teléfono celular
MR	Mensaje de respuesta a un MO. No es comúnmente usado en telecomunicaciones pero forma parte del lenguaje en Tedexis.
Trinity	Producto para el envío masivo de mensajes mediante carga de archivos de texto por SFTP.
Kronos	Productos para el envío masivo y programado de mensajes mediante la carga de archivos de texto por web.

	DDR: TIID	Código: TDX-DP-PR-D-08		Fecha: 31/05/2016
		Versión: 02	Edición: 01	Páginas: 4 de 7

Rol	Corresponde a la(s) función(es) que puede realizar un usuario en el TID.
Mercurio	Plataforma de mensajería de texto de Tedexis.
Interfaz ligera	Describe una interfaz web que muestre información con tiempos de respuesta cortos.
IU	Interfaz de usuario. Hace referencia al front-end del TIID y se usa en el documento para definir las necesidades propias de la aplicación web.

4. Descripción del producto

Tedexis Interactive Dashboard 2 (TIID) es un servicio de valor agregado para nuestros clientes. Su objetivo es ofrecer información de las transacciones que se realizan en nuestra plataforma (SMS, PUSH, EMAIL, RRSS, etc.), brindar mecanismos para el uso de la plataforma y dar información sobre los procesos, productos y métodos de integración asociados a la cuenta del cliente.

TID es el antecesor de este servicio y nace como una interfaz web para que el cliente pudiera estar informado sobre los mensajes enviados, acceder a manuales y tutoriales de nuestros productos, gestionar tickets de soporte técnico y conocer los medios a las redes sociales y otros medios de nuestra organización.

Con TIID queremos obtener un producto capaz de mejorar las funcionalidades presentes en TID con una imagen más fresca, ligera, capaz de adaptarse a los dispositivos móviles, actual, con información integra y confiable y con una arquitectura que sea fácil de mantener y evolucionar. Esta nueva versión, debe obtener información de diferentes canales, es decir, debe ser configurable y poder crecer en el acceso a diferentes fuentes de datos.

Actualmente, la información que será manejada por TIID está en diferentes fuentes de datos y en un futuro estas van a crecer en número y cantidad de información y posiblemente cambiaran de tecnología. Por tanto, TIID debe estar en la capacidad de manejar grandes volúmenes de información, ser configurable, fácil de usar y con tiempos cortos de respuesta.

TIID debe estar compuesto por dos partes principales: front-end y back-end; siendo el medio de comunicación la invocación de servicios. El front-end debe realizar las validaciones necesarias para el despliegue de información (roles) e invocación de servicios. El back-end será responsable de gestionar las peticiones y crear una respuesta que recopile información de las fuentes de datos involucradas. Idealmente, el TIID debe poseer un modelo de datos que permita manejar usuarios, roles, configuraciones y aquella información necesaria para cumplir con los requerimientos demandados.

5. Necesidades

NEC.001	IU: Crear una aplicación web fresca, ligera, fácil de usar y responsive
La interfaz de usuario del TIID debe ser una aplicación web que sea fresca, ligera, fácil de usar y responsive, la cual debe comunicarse con el back-end mediante servicios web para obtener la información que se requiere mostrar	
NEC.002	IU: Responder en el menor tiempo posible
Dado que la interacción es a través con la interfaz web, es necesario que las respuestas a las solicitudes de reportes, manuales, cargar, etc., sean en el menor tiempo posible de forma que el usuario pueda obtener	

	DDR: TIID	Código: TDX-DP-PR-D-08	Fecha: 31/05/2016
		Versión: 02	Edición: 01 Páginas: 5 de 7

información oportuna.

NEC.003	IU: Dar opciones en el tipo de gráficos en que se mostrará la información
Ofrecer al usuario la posibilidad de elegir el tipo de gráfico con el cual quiere visualizar sus reportes. Así mismo permitir ver la información en tablas.	
NEC.004	IU: Exportar información en diferentes formatos
Ofrecer al usuario opciones para la descarga de la información, es decir, la posibilidad de exportar la información mostrada en la interfaz web en archivos .csv, .xls o .pdf.	
NEC.005	IU: Disponer los reportes definidos en el TID
Se deben replicar los reportes definidos en el TID para la consulta del tráfico por empresa, pasaporte y campaña según la hora, día o mes (véase TDX-DP-PR-D-28 DDP TIID.pdf). No se incluye los reportes asociados a la opción Detalle tráfico.	
NEC.007	IU: Filtrar la información por cualquier campo mostrado al usuario
Permitir al usuario filtrar y/o buscar información por campos claves mostrados. Por ejemplo: fecha, operadora, país, tipo de mensaje, palabra clave, campaña, número de origen, número de destino, pasaporte, empresa, etc.	
NEC.008	IU: Mostrar el título del filtro aplicado
Para dar la sensación de ubicación y hacer que el usuario del TIID conozca siempre donde está es necesario mantener visible el (los) nombres de los filtros aplicados.	
NEC.009	IU: Acceder las imágenes/archivos de forma centralizada
Crear un repositorio centralizado y remoto que permita manipular las imágenes y archivos correspondientes a logos, manuales, tutoriales, etc., mostrados por las interfaces de usuario.	
NEC.010	IU: Restringir el acceso según el rol del usuario
Durante la creación de un usuario se debe configurar sus características para el acceso a funcionalidades particulares según el tipo de cliente al cual este asociado (mayorista, aliado, local, minorista, interno, administrador, etc.)	
NEC.011	IU: Configurar funcionalidades por rol
Disponer un mecanismo mediante el cual se pueda activar y desactivar las funcionalidades asociadas a un rol.	
NEC.012	IU: Mostrar reporte de facturación histórico*
Además de los reportes definidos en el TID se requiere mostrar los históricos de facturación de los últimos 6 meses en el cual se detalle la cantidad de mensajes enviados por operadora, país, pasaporte y el monto consumido en la moneda correspondiente.	
NEC.013	IU: Mostrar reporte de distribución por pasaporte y operadora*
Además de los reportes definidos en el TID se requiere mostrar un reporte que muestre el tráfico de cada pasaporte clasificado por operadora y resumiendo los totales en ambos casos.	
NEC.014	IU: Incluir el analizador de contenido*
Incluir la opción de analizar el contenido de los mensajes enviados	
NEC.015	Definir módulos desacoplados para el modelo de negocio
Diseñar una aplicación donde la lógica de negocio este desacoplada del acceso a los datos. Para ello se recomienda la construcción de una capa de persistencia que sea agnóstica de la fuente de datos.	
NEC.016	Construir un modelo de datos
Definir y construir un modelo de datos con la información necesaria para validar y atender las solicitudes realizadas por la interfaz web de tal forma que se pueda reducir el tiempo de respuesta de cada solicitud.	
NEC.017	Disponer de un módulo de configuración
Definir y construir un módulo para realizar la configuración de roles, usuarios, privilegios y demás elementos configurables.	

*Estas necesidades quedan para evaluación de una versión 2 del TIID

6.Premisas

ID	Descripción
Prem.001	TIID debe tener y mejorar las funcionalidades actuales del TID
Prem.002	El sistema debe ser desplegado en la nube y estar disponible 24/7
Prem.003	TIID debe mejorar los tiempos de respuesta de las solicitudes

	DDR: TIID	Código: TDX-DP-PR-D-08		Fecha: 31/05/2016
		Versión: 02	Edición: 01	Páginas: 6 de 7

Prem.004	El arte de la aplicación será trabajado en conjunto con el equipo de mercadeo
Prem.005	No se realizarán modificaciones en el modelo de datos de la plataforma Mercurio. Sin embargo, se pueden registrar recomendaciones en este sentido
Prem.006	TIID poseerá un modelo de datos propio con la información detallada de lo que requiere para atender cada reporte definido
Prem.007	La información proveniente de otras aplicaciones será accedida mediante servicios web dummies. Por ejemplo: el procesamiento de archivos y la cantidad de mensajes exitosos y fallidos.
Prem.008	Se debe garantizar el acceso restringido a la información mediante la validación de roles y privilegios por funcionalidad
Prem.009	TIID debe ser configurable

7. Expectativas de entrega

- Se aspira que el pasante logre culminar los objetivos de la pasantía pasadas 18 semanas a partir del inicio de las mismas.
- El inicio de la pasantía se dará el 11/07/2016 y se llevara bajo planificación definida en la oferta de pasantía (véase LM_OfertaDePasantias_Pasantia1.pdf).

8. Especificaciones del sistema: Requerimientos funcionales

REF.001	Gestionar usuario
Permite al usuario administrador crear, modificar y eliminar credenciales de un usuario TIID. Si la empresa asociada al usuario no existe, se debe crear la empresa primero. En la creación del usuario se le asociaran los pasaportes y privilegios correspondientes a dicho usuario.	
REF.002	Gestionar rol
El sistema permitirá la asignación de un rol a los usuarios TIID, dichos roles son: Tedexis, Canales, Solo programación de archivos, Sin programación de archivos y Tikets y ver programación. Un usuario puede tener uno o más roles.	
REF.003	Gestionar funcionalidad/privilegio
Cada rol posee privilegios y restricciones de acceso a funcionalidades en el sistema, los cuales serán asignados según el tipo de cliente asociado. Las restricciones de acceso, de cada rol, podrán ser modificadas mediante una interfaz del sistema por el usuario administrador.	
REF.004	Desplegar gráfico
En la visualización de los gráficos, el usuario podrá elegir entre diferentes tipos de gráficos: Torta, Columnas y Líneas.	
REF.005	Descargar reportes
El sistema facilitará un mecanismo para que el usuario descargue la información mostrada en la interfaz en un archivo .cvs, .xls o pdf.	
REF.006	Gestionar filtros/búsquedas
El sistema proporcionará una interfaz de búsqueda, mediante la cual el usuario podrá realizar búsquedas y/o aplicar filtros utilizando los campos claves mostrados, tales como, pasaporte, fecha, entre otros.	
REF.007	Invocar servicios
La información desplegada en los distintos apartados será obtenida mediante la invocación de servicios REST al back-end del sistema.	
REF.008	Manejar internacionalización
Se mostrará un mecanismo con el cual el usuario podrá ser capaz de cambiar el idioma del sistema, entre inglés y español, siendo español el idioma por defecto.	
REF.009	Gestionar RRSS y Adas
El sistema mostrará los post más recientes de Twitter y blog de la página web de Tedexis, de igual manera, mostrará las campañas publicitarias gestionadas a través del sistema Adas.	

	DDR: TIID	Código: TDX-DP-PR-D-08		Fecha: 31/05/2016
		Versión: 02	Edición: 01	Páginas: 7 de 7

REF.010	Gestionar reportes por pasaporte
El sistema tendrá un apartado "Reporte pasaporte" donde por cada pasaporte se podrá visualizar la evolución de tráfico, tanto por día como por hora, y el tráfico de mensajes correspondiente a las operadoras asociados a dicho pasaporte.	
REF.011	Gestionar reportes por empresa
El sistema tendrá un apartado "Reporte empresa" donde podrá visualizar información sobre saldo de pasaportes, evolución de tráfico, tanto por día como por hora, tráfico de las operadoras, campañas, top empresas y top pasaportes. Información relevante correspondiente a la empresa.	
REF.012	Gestionar configuración
El sistema poseerá una interfaz que permitirá la gestión de usuarios, roles y privilegios, así como todos los elementos configurables.	
REF.013	Cargar información consolidada
El sistema desplegará información consolidada mediante solicitudes a una base de datos local, la cual posee información que es actualizada por un proceso que se invoca con cierta frecuencia.	

9. Especificaciones del sistema: Requerimientos no funcionales

RNF.001	Disponibilidad
El sistema TIID estará disponible 24/7.	
RNF.002	Confiabilidad e integridad de la información
El sistema muestra a los usuarios información relevante y concreta proveniente de la plataforma Mercurio, a la cual se aplican una serie de procesos para su mejor visualización.	
RNF.003	Tiempos de respuesta cortos
TIID mejorará los tiempos de respuesta de TID, esto se logra consultado la información de una base de datos local consolidada, la cual se actualiza con cierta frecuencia.	
RNF.004	Seguridad
El sistema dispondrá de un mecanismo de autenticación de usuarios el cual restringirá el acceso a las distintas funcionalidades del sistema. Con la invocación de servicios REST se emplea el uso de un <i>token</i> el cual autentica al usuario frente al servidor esto permite mantener una sesión de trabajo en aplicaciones nativas (móviles y no móviles).	
RNF.005	Escalabilidad y mantenimiento
Mediante los archivos de propiedades se podrá configurar los módulos TIID. Se proveerá la documentación necesaria y relevante para el posterior mantenimiento y expansión del sistema.	

Apéndice B

Documento de Definición de Proyecto

	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 1 de 10

Hoja de trazabilidad del documento

Versión	Fecha	Descripción	Responsable de la Actualización	Responsable de la Revisión	Responsable de la Aprobación
01	31/06/2016	Creación de Documento	<u>Jeselys Hernández M.</u> Director de Producto	<u>Robinson Rodríguez</u> Consultor de Productos <u>Yermi Echeverría</u> Coordinador de Productos <u>Ninoska González</u> Gerente de Mercadeo	<u>Elizabeth Navas</u> Director Comercial VE <u>Roberto Angulo</u> Director Comercial TDX <u>Jeselys Hernández M.</u> Director de Producto <u>Vito Mastrogiacomio</u> Director General

	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 2 de 10

Tabla de Contenido

1. Introducción	3
2. Objetivo:	3
3. Acrónimos, definiciones o términos	4
4. Descripción del producto	4
5. Funcionalidades del producto	5
6. Consideraciones generales	8
7. Arquitectura de la solución	8
8. Anexos	9
8.1 Anexo 1: Funcionalidades y privilegios por rol	9

	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 3 de 10

1. Introducción

En Tedexis, hemos estado en la búsqueda de herramientas que ofrezcan a nuestros clientes información oportuna y con las cuales se sientan confiados cuando realizan transacciones en nuestra plataforma. Nos vemos interesados en la generación de valor mediante productos que permitan interactuar con esa información y que aporten conocimiento sobre el tráfico que se ha generado para cada usuario. Además de ofrecer opciones para interactuar con nuestra plataforma mediante la diversificación de canales e interfaces de integración dependiendo de las necesidades y naturaleza del cliente.

Tedexis Interactive Dashboard (TID) es una aplicación web con la cual se facilita la interacción de nuestros clientes con los servicios que ofrecemos de forma rápida y sencilla. Entre las funcionalidades disponibles en la aplicación se encuentran:

- Acceso a reportes de tráfico (véase *Anexo 1*).
- Información de los mensajes enviados según éxito y falla para cada una de las interfaces de integración (Archivos programados, campañas o mensajes interactivos, tráfico masivo, etc.)
- Carga de archivos para envío masivo de mensajes.
- Acceso a manuales, tutoriales y preguntas frecuentes
- Gestión de tickets de soporte.
- Acceso a redes sociales (noticias, Twitter, Facebook, correo, etc.).

TID fue desarrollada en Tedexis en el año 2010, donde la capacidad de la plataforma y las condiciones del mercado eran estables, lo cual permitió generar información confiable, integra y con tiempos de respuesta aceptables tanto para la organización como para el cliente. Sin embargo, para 2016 esta situación ha cambiado, nos enfrentamos a clientes más exigentes en cuanto al conocimiento y la búsqueda de información que quieren respuestas inmediatas y una demanda de transacciones que mes a mes excede nuestras expectativas.

Lo anterior más que representar una amenaza se convierte en un reto que queremos atacar y convertir en nuestra fortaleza. Hacer que las necesidades del mercado y de nuestro negocio nos den las pautas para mejorar y evolucionar ante nuestros clientes, aliados y competidores mediante la entrega de una herramienta capaz de ofrecer información de valor en tiempo real, con acceso a productos y servicios para gestionar los canales de comunicación disponibles que se integren de forma natural y que no requiera costos adicionales.

A continuación se describen las necesidades y funcionalidades del TID o conocido como Tedexis Interactive Dashboard 2 (TIID), las consideraciones para su desarrollo, la visión de la arquitectura poco acoplada basada en servicios y el esquema de datos que se amolda a la solución.

2. Objetivo:

Describir detalladamente las necesidades, funcionalidades, consideraciones, arquitectura y elementos de desarrollo de Tedexis Interactive Dashboard 2 (TIID) con el fin de que sirva de insumo en su construcción y desarrollo.

	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 4 de 10

3. Acrónimos, definiciones o términos

Acrónimo, definición o término	Descripción
SMS (Short Message Service)	Servicio que puede enviar o recibir mensajes entre celulares y otros dispositivos electrónicos e incluso utilizando internet
SMPP (Short Message Peer-to-peer Protocol)	Protocolo estándar de telecomunicaciones pensado para el intercambio de SMS entre equipos que gestionan los mensajes como pueden ser los SMSC o los GSM USSD (Unstructured Supplementary Services Data server) y un sistema de solicitud de SMS como puede ser un servidor WAP o cualquier gateway de mensajería.
SFTP (Secure File Transfer Protocol)	Protocolo para transferencia (carga y descarga) segura de archivos.
Mensaje Masivo	Representa un lote grande de mensajes de texto. Son enviados mediante cargas de archivos de texto.
Mensaje Interactivo	Está formado por un MO y MT. En Tedexis suele llamarse mensaje interactivo a los mensajes manipulados por las campañas de mensajería.
MO (Mobile Originated)	Mensaje originado desde un teléfono celular
MT (Mobile Terminated)	Mensaje recibido en un teléfono celular
MR	Mensaje de respuesta a un MO. No es comúnmente usado en telecomunicaciones pero forma parte del lenguaje en Tedexis.
Trinity	Producto para el envío masivo de mensajes mediante carga de archivos de texto por SFTP.
Kronos	Productos para el envío masivo y programado de mensajes mediante la carga de archivos de texto por web.
Rol	Corresponde a la(s) función(es) que puede realizar un usuario en el TID.
Mercurio	Plataforma de mensajería de texto de Tedexis.
Interfaz ligera	Describe una interfaz web que muestre información con tiempos de respuesta cortos.
IU	Interfaz de usuario. Hace referencia al front-end del TIID y se usa en el documento para definir las necesidades propias de la aplicación web.

4. Descripción del producto

Tedexis Interactive Dashboard 2 (TIID) es un servicio de valor agregado para nuestros clientes. Su objetivo es ofrecer información de las transacciones que se realizan en nuestra plataforma (SMS, PUSH, EMAIL, RRSS, etc.), brindar mecanismos para el uso de la plataforma y dar información sobre los procesos, productos y métodos de integración asociados a la cuenta del cliente.

TID es el antecesor de este servicio y nace como una interfaz web para que el cliente pudiera estar informado sobre los mensajes enviados, acceder a manuales y tutoriales de nuestros productos, gestionar tickets de soporte técnico y conocer los medios a las redes sociales y otros medios de nuestra organización.

Con TIID queremos obtener un producto capaz de mejorar las funcionalidades presentes en TID con una imagen más fresca, ligera, capaz de adaptarse a los dispositivos móviles, actual, con información integra y confiable y con una arquitectura que sea fácil de mantener y evolucionar. Esta

	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 5 de 10

nueva versión, debe obtener información de diferentes canales, es decir, debe ser configurable y poder crecer en el acceso a diferentes fuentes de datos.

Actualmente, la información que será manejada por TIID está en diferentes fuentes de datos y en un futuro estas van a crecer en número y cantidad de información y posiblemente cambian de tecnología. Por tanto, TIID debe estar en la capacidad de manejar grandes volúmenes de información, ser configurable, fácil de usar y con tiempos cortos de respuesta.

TIID debe estar compuesto por dos partes principales: front-end y back-end; siendo el medio de comunicación la invocación de servicios. El front-end debe realizar las validaciones necesarias para el despliegue de información (roles) e invocación de servicios. El back-end será responsable de gestionar las peticiones y crear una respuesta que recopile información de las fuentes de datos involucradas. Idealmente, el TIID debe poseer un modelo de datos que permita manejar usuarios, roles, configuraciones y aquella información necesaria para cumplir con los requerimientos demandados.

5. Funcionalidades del producto

- Autenticación de usuario: garantiza el acceso restringido a la aplicación y el despliegue de las funcionalidades según el(los) rol(es) de cada usuario.
 - a. Recuperación de contraseña: permite al usuario reiniciar su contraseña en caso de olvido o desactivación.
- Enlace a redes sociales y contactos de empresa: permite al usuario acceder Twitter y Facebook de Tedexis y/o enviar un correo electrónico.
- Mostrar los post más recientes Twitter y blog de la página web (RSS).
- Manejo de campañas publicitarias (Adas).
- Multilinguaje: ofrece al usuario opciones de lenguaje para el despliegue de la información. Se dispone de español e inglés siendo el español el idioma por defecto.
- Reporte empresa: contiene los reportes tráfico de los mensajes enviados y recibidos por los pasaportes asociados a una empresa. Los reportes de esta sección puede ser exportados.
 - a. Saldo Pasaportes: muestra una tabla con todos los pasaportes asociados al usuario, con la siguiente información: nombre del pasaporte, descripción del pasaporte, correo electrónico de la persona que se puede contactar en caso de problemas, teléfono de la persona de contacto en formato internacional y saldo disponible del pasaporte.
 - b. Evolución tráfico: despliega una interfaz con dos pestañas: gráfico muestra un gráfico con el tráfico mensual MT e interactivo de un periodo determinado para los pasaportes asociados a la empresa; y data muestra una tabla con la información diagramada en la pestaña gráfico. En este reporte se puede manipular el período de tiempo que se quiere visualizar 1 mes, 3 meses, 6 meses o un rango en particular.
 - c. Evolución día: muestra una interfaz con dos pestañas: gráfico se visualiza la gráfica diaria del tráfico MO y MT de un mes específico de los pasaportes asociados a la empresa; data resume en una tabla la información del total de MO y MT enviados por cada día del mes seleccionado. En este reporte puede seleccionarse desde el mes en curso hasta 6 meses atrás.
 - d. Evolución hora: posee dos pestañas: gráfico donde se muestra la gráfica de la cantidad de mensajes MT y MO por hora de los pasaportes de asociados a la empresa;

	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 6 de 10

y data muestra una tabla con un resumen de la información mostrada en el gráfico. En este reporte se puede realizar consulta del día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado.

- e. Distribución operadora: despliega en una tabla y un gráfico de torta la información del tráfico enviado por cada operadora para los pasaportes asociados a una empresa. Se puede consultar el día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado.
- f. Campañas: resume en una tabla las campañas asociadas los pasaportes de la empresa, con la información sobre tipo de campaña, descripción, detalle y reporte. La opción detalle muestra una tabla con fecha, número de teléfono, texto enviado y la respuesta que se envió y permite realizar búsqueda por número de teléfono. La opción reporte se despliega la información resumida de la campaña: una gráfica de puntos de los mensajes totales enviados y recibidos a esa campaña; una tabla con los totales diarios; y una gráfica de torta con la cantidad de mensajes valido e inválidos recibidos. En este reporte se puede realizar consulta del día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado.
- g. Top empresa: muestra la información de las empresas con mayor cantidad de tráfico según el tipo de mensaje: top (10) tráfico MT y top (10) tráfico MO. Además, muestra indicadores sobre el tráfico total MT/MO, sumatoria del tráfico MT/MO del top 5, proporción que representa el top 5 con respecto al total, sumatoria del tráfico MT/MO del top 10 y proporción que representa el top 10 con respecto al total. En este reporte se puede realizar consulta del día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado.
- h. Top pasaporte: muestra la información de los pasaportes con mayor cantidad de tráfico según el tipo de mensaje: top (10) tráfico MT y top (10) tráfico MO. Además, muestra indicadores sobre el tráfico total MT/MO, sumatoria del tráfico MT/MO del top 5, proporción que representa el top 5 con respecto al total, sumatoria del tráfico MT/MO del top 10 y proporción que representa el top 10 con respecto al total. En este reporte se puede realizar consulta del día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado.
- Reporte pasaporte: contiene los reportes del tráfico enviado y recibido por un pasaporte. Los reportes de esta sección puede ser exportados.
 - a. Evolución tráfico: despliega una interfaz con dos pestañas: gráfico muestra un gráfico con el tráfico mensual MT e interactivo de un periodo determinado para un pasaporte seleccionado; y data muestra una tabla con la información diagramada en la pestaña gráfico. En este reporte se puede manipular el período de tiempo que se quiere visualizar 1 mes, 3 meses, 6 meses o un rango en particular.
 - b. Evolución día: muestra una interfaz con dos pestañas: gráfico se visualiza la gráfica diaria del tráfico MO y MT de un mes específico para un pasaporte seleccionado; data resume en una tabla la información del total de MO y MT por cada día del mes seleccionado. En este reporte puede seleccionarse desde el mes en curso hasta 6 meses atrás.

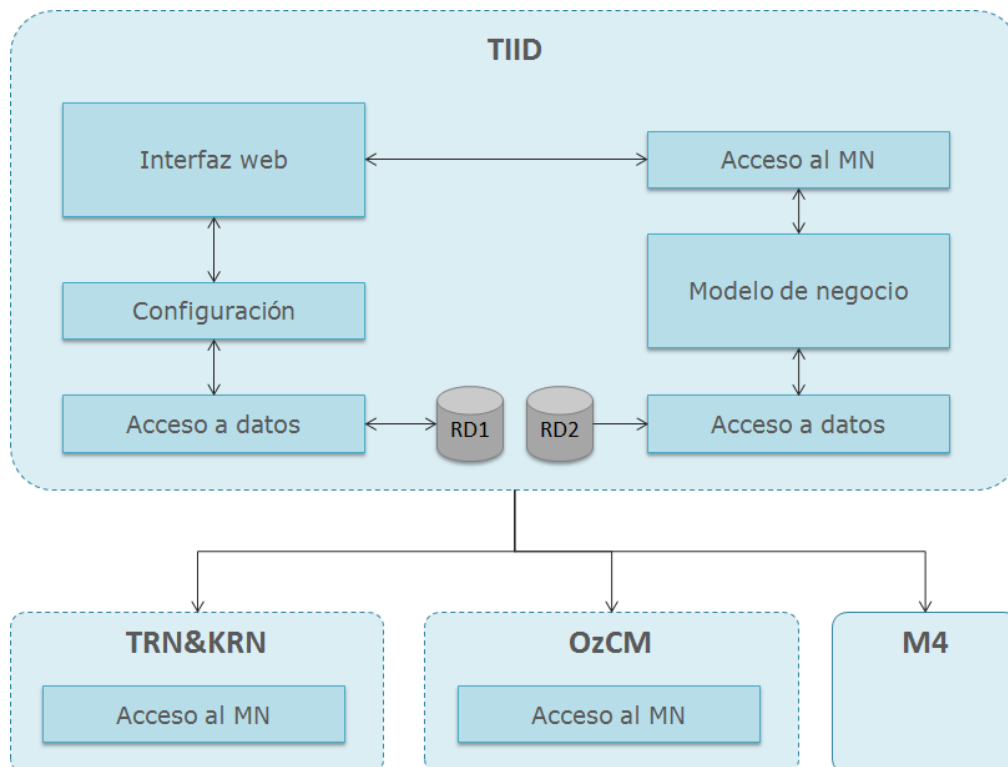
	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 7 de 10

- c. Evolución hora: posee dos pestañas: gráfico donde se muestra la gráfica de la cantidad de mensajes MT y MO por hora de un pasaporte; y data muestra una tabla con un resumen de la información mostrada en el gráfico. En este reporte se puede realizar consulta del día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado.
 - d. Distribución operadora: despliega en una tabla y un gráfico de torta la información del tráfico enviado por cada operadora de un pasaporte seleccionado. Se puede consultar el día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado.
- Detalle Tráfico: contiene las opciones para ver el detalle del tráfico de mensajes enviados según el tipo de interfaz de integración. Los reportes de esta sección puede ser exportados.
 - a. Interactivos: muestra una tabla con fecha, número de origen, operadora destino y número de destino. Se puede consultar el día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado. Además, se puede realizar búsquedas por el número de origen.
 - b. Terminación (MT): muestra una tabla que contiene fecha, operadora origen, short code y número de destino. Se puede consultar el día actual (por defecto), últimos 30 días, los últimos 2 meses, últimos 3 meses, una fecha determinada o un rango determinado. Además, se puede realizar búsquedas por el número de destino.
 - c. Trinity (sFTP): muestra una tabla con fecha, short code, envíos exitosos y envíos fallidos para cada archivo cargado por sFTP. Esta opción permite descargas el archivo completo y la información detallada de los mensajes exitosos y fallidos.
 - d. Archivos programados: despliega una lista con la siguiente información: nombre del usuario que subió el archivo, fecha en que fue cargado, fecha programada para el envío, nombre del archivo cargado, estado de la ejecución programada y la cantidad de mensajes fallidos encontrados al ejecutar el archivo. Esta opción permite descargar el archivo original cargado y cancelar la ejecución futura de un archivo.
 - e. Exportar: permite al usuario configurar la descarga del tráfico detallado de un pasaporte. Se debe definir el tipo de mensaje (MT, MO o todos), el formato (Detalles o resumen masivos) y un rango de fecha (fecha de inicio y fecha de fin).
- Carga de archivos programados (Kronos)*: permite la carga y configuración de envío de archivo de texto para envíos masivos de SMS. Esta opción permite definir la fecha, hora y zona horaria del envío.
- Soporte: contiene toda la información que puede ayudar al usuario a resolver cualquier problema que pueda presentarse al utilizar los productos que ofrece Tedexis mediante la descarga de manuales, tutoriales y la creación de tickets de soporte técnico.
- Descarga: contiene el acceso a la descarga productos para la integración y envío de mensajes.
- Administrar TID: contiene las opciones para modificar la experiencia de uso del TID. En esta opción puede:
 - a. Cambiar Clave: permite al usuario cambiar su clave de acceso al TID.
 - b. Configurar Pasaportes: permite administrar los pasaportes asociados a un usuario.

6. Consideraciones generales

- El desarrollo debe realizarse con tecnología J2EE y su arquitectura deberá estar orientada a servicios
- El levantamiento de información puede/debe involucrar a las áreas comerciales TDX y VE, Atención al cliente, Mercadeo y Operaciones.
- Los requerimientos a desarrollar están detallados en el DDR (véase TDX-DP-PR-D-08 DDR TIID.pdf), el cual consolida información de las áreas involucradas.
- El alcance de esta pasantía está enmarcado en las funcionalidades del TID actual y algunas mejoras puntuales en cuanto a filtros, búsquedas, descargas y elementos visuales que aportan valor.
- Las funcionalidades de reporte de tráfico detallado (RTD) de terminación (MT) e interactivo (MO), Trinity y archivos programados (Kronos) no está incluido en el desarrollo.
- El repositorio centralizado de imágenes/archivos deberá ser creado por el equipo de Productos en un servidor de acceso público.

7. Arquitectura de la solución



	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 9 de 10

8. Anexos

8.1 Anexo 1: Funcionalidades y privilegios por rol

Nombre del reporte	Tipo	Rol	Privilegio
Carga de archivos/programar envío (KRONOS)	Pasaporte	Tedexis Canales Solo Prog Archivos	Carga de archivos Kronos
Saldo de pasaportes	Empresa	Tedexis Canales Sin Prog Archivos	Ver saldo pasaporte
Evolución tráfico	Empresa	Tedexis Canales Sin Prog Archivos	Ver TRF empresa mes
Evolución día	Empresa	Tedexis Canales Sin Prog Archivos	Ver TRF empresa día
Evolución hora	Empresa	Tedexis Canales Sin Prog Archivos	Ver TRF empresa hora
Distribución operadora	Empresa	Tedexis Canales Sin Prog Archivos	Ver Dist operadora empresa
Campañas	Empresa	Tedexis Canales Sin Prog Archivos	Ver campañas
Top empresas	Empresa	Tedexis	Ver top empresa
Top pasaportes	Pasaporte	Tedexis	Ver top pasaporte
Evolución tráfico	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver TRF pasaporte mes
Evolución día	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver TRF pasaporte día
Evolución hora	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver TRF pasaporte hora
Distribución operadora	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver Dist operadora pasaporte
Interactivos	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver detalle MO
Terminación (MT)	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver detalle MT
Trinity (sFTP)	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver Trinity
Archivos Programados	Pasaporte	Tedexis Canales Solo Prog Archivos Tickets & Ver Prog	Ver archivos programados

	DDP: Tedexis Interactive Dashboard 2	Código: TDX-DP-PR-D-28		Fecha: 31/05/2016
		Versión: 01	Edición: 01	Páginas: 10 de 10

Exportar	Pasaporte	Tedexis Canales Sin Prog Archivos	Ver exportar
Filtro por región	Pasaporte/Empresa	Tedexis	Ver agrupar por región