**INSTITUTO TECNOLÓGICO DE MILPA ALTA**

**INGENIERIA EN SISTEMAS COMPUTACIONALES**

**Nombre del proyecto:**

Wind dashboard

(Consola de monitoreo inteligente para Contact Center)

Avenida México 700, oficina 106,

San Jerónimo Lídice Ciudad de México

Distrito Federal, C.P. 10200 México.

Teléfono: 5567180024.

**AUTOR:**

**FARIAS BAUTISTA ANDRES**

**NO.COTROL: 111070176**

**Datos del asesor externo:**

**Ing. Marco Alejandro Villaseñor Amezcua**

**Puesto:** Technology Solutions Director.

**Teléfono**

**Mobile: (52) 1 55 54094045**

**Phone: (52) 55 67180023**

**E - Mail:** marco@interastar.com

Asesor interno:

MC. Ernesto de la Cruz Nicolás.

**TABLA DE CONTENIDO**

[**1.** **INTRODUCCIÓN** 6](#_Toc459909421)

[**2.** **OBJETIVOS** 8](#_Toc459909422)

[2.1 Objetivo general 8](#_Toc459909423)

[2.2 Objetivos específicos 8](#_Toc459909424)

[**3.** **JUSTIFICACIÓN** 9](#_Toc459909425)

[**4.** **CARACTERIZACION DEL AREA EN QUE PARTICIPO** 10](#_Toc459909426)

[4.1 ¿Quiénes somos? 10](#_Toc459909427)

[4.2 Historia. 10](#_Toc459909428)

[a. Organización interna de Interastar. 11](#_Toc459909429)

[4.3 Domicilio de la empresa 12](#_Toc459909430)

[4.4 Ubicación 12](#_Toc459909431)

[4.5 Funciones de la empresa 13](#_Toc459909432)

[4.5.1 Servicios Profesionales (PSO) 13](#_Toc459909433)

[4.5.2 Soluciones de Tecnología (TSG) 14](#_Toc459909434)

[4.5.2.1 Integración de Aplicaciones 14](#_Toc459909435)

[4.5.2.2 Desarrollo de Soluciones 14](#_Toc459909436)

[4.5.2.3 Desarrollo de Productos Propios 15](#_Toc459909437)

[4.6 Misión 15](#_Toc459909438)

[4.7 Visión 16](#_Toc459909439)

[4.8 Valores 16](#_Toc459909440)

[4.9 Asesor externo 16](#_Toc459909441)

[4.10 Actividades del residente 17](#_Toc459909442)

[**5.** **PROBLEMAS A RESOLVER, PRIORIZÁNDOLOS** 18](#_Toc459909443)

[5.1 Calendarización del proyecto 19](#_Toc459909444)

[5.2 Recursos del proyecto 20](#_Toc459909445)

[5.3 Personal involucrado 20](#_Toc459909446)

[5.4 Hardware 20](#_Toc459909447)

[5.5 Software 21](#_Toc459909448)

[**6.** **FUNDAMENTO TEORICO** 22](#_Toc459909449)

[6.1 Lenguajes de programación 22](#_Toc459909450)

[6.1.1 CSS3 22](#_Toc459909451)

[6.1.2 HTML 22](#_Toc459909452)

[6.1.3 NodeJS 23](#_Toc459909453)

[6.1.4 ReactJS 23](#_Toc459909454)

[6.1.5 Meteor 24](#_Toc459909455)

[6.1.6 JavaScript 24](#_Toc459909456)

[6.2 Librerías javascript 25](#_Toc459909457)

[6.2.1 MomentJS 25](#_Toc459909458)

[6.2.2 D3JS 26](#_Toc459909459)

[6.2.3 Jquery 26](#_Toc459909460)

[6.2.4 Browserify 26](#_Toc459909461)

[6.2.5 Yeoman 27](#_Toc459909462)

[6.3 Base de datos 27](#_Toc459909463)

[6.3.1 Influxdb 27](#_Toc459909464)

[6.4 Tecnologías de comunicación Back-End Front-End 28](#_Toc459909465)

[6.4.1 JSON 28](#_Toc459909466)

[6.4.2 API REST 28](#_Toc459909467)

[6.5 Herramientas 29](#_Toc459909468)

[6.5.1 Git 29](#_Toc459909469)

[6.5.2 Bitbucker 29](#_Toc459909470)

[6.5.3 Sublime Text (IDE) 29](#_Toc459909471)

[6.5.4 Source Tree 29](#_Toc459909472)

[**7.** **PROCEDIMIENTO Y DESCRIPCION DE LAS ACTIVIDADES REALIZADAS** 31](#_Toc459909473)

[7.1 Ámbito del sistema 31](#_Toc459909474)

[7.2 Descripción general del software 31](#_Toc459909475)

[7.3 Reglas de negocio 31](#_Toc459909476)

[7.4 Lenguajes de programación 32](#_Toc459909477)

[7.5 Requisitos futuros. 32](#_Toc459909478)

[7.6 Descripción del medio ambiente de desarrollo 32](#_Toc459909479)

[7.7 Protocolo de comunicación 33](#_Toc459909480)

[7.8 Software necesario en el servidor para ejecutar el sistema 33](#_Toc459909481)

[7.9 Descripción del actor del sistema 34](#_Toc459909482)

[7.10 Diagramas de caso de Uso para el inicio de Wind dashboard 35](#_Toc459909483)

[7.11 Diagrama de Clases 36](#_Toc459909484)

[7.12 Diagrama de Secuencia 38](#_Toc459909485)

[7.13 Base de datos 39](#_Toc459909486)

[**8.** **RESULTADOS, PLANOS, GRAFICAS.** 40](#_Toc459909487)

[8.1 Gráfica de Gauge 40](#_Toc459909488)

[8.2 Grafica NumberMinutes 42](#_Toc459909489)

[8.3 Grafica Numeric. 43](#_Toc459909490)

[8.4 Gráfica Range 44](#_Toc459909491)

[8.5 Grafica SingleLine 44](#_Toc459909492)

[8.6 Grafica MultiLine. 46](#_Toc459909493)

[8.7 Wind dahsboard 47](#_Toc459909494)

[**9.** **RECOMENDACIONES** 48](#_Toc459909495)

[**10.** **CONCLUSIONES** 49](#_Toc459909496)

[**11.** **COMPETENCIAS DESARROLLADAS Y/O APLICADAS** 49](#_Toc459909497)

[**12.** **REFERENCIAS BIBLIOGRAFICAS** 49](#_Toc459909498)

**TABLA DE ILUSTRACIONES**

[**Ilustración I Organigrama de la empresa (Elaboración propia).** 11](#_Toc439971757)

[**Ilustración II Ubicación de la empresa** 12](#_Toc439971758)

[**Ilustración III Actividades** 19](#_Toc439971759)

[**Ilustración IV Definición resumida de los actores del sistema** 34](#_Toc439971760)

[**Ilustración V Caso de uso de inicio del sistema** 35](#_Toc439971761)

[**Ilustración VI Inicio del Sistema** 35](#_Toc439971762)

[**Ilustración VII Diagrama de clases** 37](#_Toc439971763)

[**Ilustración VIII Diagrama de secuencia** 38](#_Toc439971764)

[**Ilustración IX Base de datos** 39](#_Toc439971765)

[**Ilustración X Grafica de Gauge mostrando el nivel de servicio.** 41](#_Toc439971766)

[**Ilustración XI Grafica NumberMinutes.** 43](#_Toc439971767)

[**Ilustración XII Número de agentes firmados.** 44](#_Toc439971768)

[**Ilustración XIII Numero de llamas en espera.** 44](#_Toc439971769)

[**Ilustración XIV Grafica Range.** 45](#_Toc439971770)

[**Ilustración XV Grafica SingleLine.** 46](#_Toc439971771)

[**Ilustración XVI Grafica MultiLine.** 47](#_Toc439971772)

[**Ilustración XVII Wind dahsboard 48**](#_Toc439971773)

# **INTRODUCCIÓN**

Las Tecnologías de la Información y la Comunicación están presentes en nuestras vidas y la han transformado. Han transformado la gestión de las empresas y nuestra manera de hacer negocios: Para comunicarnos con nuestros clientes, gestionar pedidos, promocionar nuestros productos, para relacionarnos con la administración pública.

El contenido de este documento trata de las fases generales y de los fundamentos teóricos que se necesitan en el desarrollo de Wind dashboard (consola de monitoreo inteligente para Call Center) con graficas en tiempo real que muestra el comportamiento de las llamadas telefónicas.

Interastar es una empresa enfocada a los sectores de TI (Tecnologías de la información) y comunicaciones, la cual se dedica al desarrollo de software, principalmente en las áreas de: Tecnologías para Centros de contacto y Biometría de voz, Análisis y mejora de experiencias de usuario, entre otros.

Interastar desarrolló un sistema para beneficiar a los Contact Center (Centro de Contacto) y los Call Center (Centro de llamadas), el cual dará un panorama con graficas del monitoreo de llamadas, tomando en cuenta que, el principal giro de estas empresas Call Center son la entrada y salida de llamadas telefónicas (comunicación asistente-cliente),

Para lograr el objetivo deseado el proyecto Wind dashboard se dividió en tres fases principales: la primera es una recolección de datos (Front-End), la segunda fase consiste en la visualización de los datos deseados (Back-End), en esta fase se trabajó y se desarrolló este proyecto y la tercera y última fase consiste en el diseño gráfico. En este proyecto se trabajó en la parte de Front-End.

Wind dashboard ha sido desarrollado con tecnologías de punta, utilizando las mejores prácticas de programación, mucho de ellos enfocados a la visualización de datos, manejando repositorios y control de versiones, simplicidad de código, técnicas muy actuales del mundo de desarrollo de software, entre las cuales destacan programación con pruebas unitarias TDD (Test Driven Development) y técnicas de aceptación en equipo ATDD (Acceptance Test Driven Development), metodología de trabajo SCRUM (metodología ágil), entre otras.

Durante el desarrollo del proyecto se obtuvo muchas limitantes sobre las tecnólogas, las cuales fueron solucionadas y en algunos casos fueron omitidas dado que el equipo de desarrollo encontró mejores alternativas, el tiempo de aprendizaje y la investigación de los lenguajes y herramientas fueron las mayores limitantes al momento del desarrollo.

# **OBJETIVOS**

## 2.1 Objetivo general

Desarrollar el sistema Wind dashboard en su front-end para obtener un monitoreo en tiempo real del comportamiento de la operación del Contact Center, consumiendo información del sistema existente sin alterar la operación y sin poner en riesgo esta información, mostrando mediante gráficas el resultado del monitoreo esto con su primera fase en donde solo será para TV.

## 2.2 Objetivos específicos

* Desarrollar el sistema con tecnología de punta.
* Dar seguridad al sistema.
* Dar un buen diseño al sistema.
* Mostrar información oportuna.
* Dar una experiencia de usuario muy amigable.

# **JUSTIFICACIÓN**

Dado a la pedida de control del flujo de las llamadas, mal atención a los clientes durante las llamadas, falta de una visualización de las entradas y salidas de llamadas surge Wind dashboard, un sistema desarrollado con una arquitectura escalable, desarrollando en esta fase para web y TV, en donde muestra en tiempo real el comportamiento de la operación del Contact Center, en una única pantalla, conectándose a su infraestructura de comunicación multicanal con los clientes: telefonía, chat, email, redes sociales.

Todo esto con una interfaz de usuario amigable, graficas sencillas pero con información deseable, con opciones de personalización tamaños de las gráficas obtenidas del monitoreo del Contact Center así como cambio de posicionamiento.

Dando una visualización rápida del performance de la operación y la toma de decisiones de una forma más ágil y precisa.

# **CARACTERIZACION DEL AREA EN QUE PARTICIPO**

## 

## 4.1 ¿Quiénes somos?

Interastar es una empresa que ofrece soluciones de tecnología para empresas de todos tamaños y giros.

Estamos especializados en Telefonía de gran escala, integración de aplicaciones, procesos de Negocios Inteligentes (BI) y consultoría y planificación de estrategias de tecnología.

Nos apasiona el ritmo con el que la tecnología evoluciona y nos hemos marcado como meta estar siempre innovando nuestros servicios para mantenernos como una alternativa confiable para nuestros clientes.

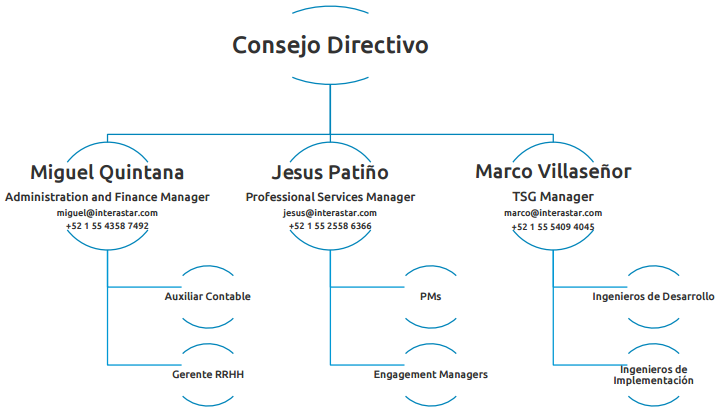
## 4.2 Historia.

Interastar es fundada en el año 2011 por Marco Villaseñor como una compañía dedicada a Servicios de Telefonía para grandes empresas y Call Centers, pero pronto aumentó su oferta de servicios para incluir integración de aplicaciones y el desarrollo de productos propios.

En el 2012, por medio de una alianza estratégica, se incluye en el portafolio de servicios la consultoría y desarrollos de BI, ERP’s (Enterprise Resource Planning – Planificación de Recursos Empresariales) y CRM’s (Customer Relationship Management - Gestión de relaciones con los clientes) así como la inclusión de una Software Factory.

Durante el 2013 se cambia la razón social para incorporar a la empresa a Jesús Patiño y Miguel Quintana, quienes ahora forman parte del equipo directivo de la empresa.

## Organización interna de Interastar.

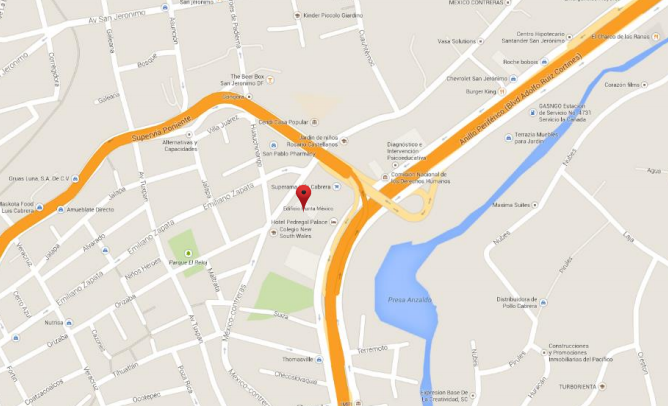


**Ilustración I Organigrama de la empresa (Elaboración propia).**

## 4.3 Domicilio de la empresa

Avenida México 700, oficina 106, San Jerónimo Lídice Ciudad de México Distrito Federal, C.P. 10200 México. Teléfono: 5567180024.

## 4.4 Ubicación



**Ilustración II Ubicación de la empresa**

## 4.5 Funciones de la empresa

### 4.5.1 Servicios Profesionales (PSO)

#### **4.5.1.1 Consultoría en Soluciones de Comunicación Empresarial**

Soluciones de Telefonía para medianas y grandes empresas. Análisis, diseño e implementación de flujos para IVRs.

#### **4.5.1.2 Inteligencia de Negocio**

Análisis y diseño de sistemas que transformen la información aislada y desestructurada para ser explotada directamente o para su análisis y conversión en conocimiento, proporcionando fundamentos sólidos para la toma de decisiones sobre el negocio.

#### **4.5.1.3 Administración de Proyectos**

Proveemos a empresas que no tengan departamento de proyectos con PMs certificados (PMI), con experiencia en IT, para asegurar el cumplimiento de sus objetivos logrando así las metas establecidas en el marco de tiempo deseado.

#### **4.5.1.4 Consultoría y Asesoría en Redes y Datos**

Telefonía para empresas medianas y grandes así como Call Centers: Diseño de infraestructura de datos y planificación para requerimientos futuros que cumplan con las metas del cliente a corto plazo sin comprometer las de largo plazo.

#### **4.6.1.5 Implantación de Soluciones Líderes de Mercado**

Automatic Call Distribution (ACD), Agile Communication Enviroment (ACE), Aura Enablement Services (AES), Avaya Interaction Center (AIC), Avaya Proactive Contact (APC), Aura Contact Centre, Aura Experience Portal, Aura Session Manager, Aura System Manager, Call Center Elite (CC Elite), Contact Center Express (CCE), Call Management System (CMS), Contact Recording, Quality Monitoring, Session Border Controller, Small and Medium Enterprise Communications, Unified Communications, UC Conferencing, Unified Messaging, Aura Messaging, One-X Agent, One-X ClientEnablement Services, One-X Deskphone, One-X Mobile, Flare Experience, Aura Presence Services, Radvision Scopia, Voice Portal (VP), Avaya Voice Self Service (VSS), Workforce Optimization (WFO), Swithces/Datos.

### 4.5.2 Soluciones de Tecnología (TSG)

### 4.5.2.1 Integración de Aplicaciones

Hacemos que dos aplicaciones o sistemas que están incomunicados compartan la información que el cliente requiere de modo que se eviten trabajos innecesarios. Esto eleva la productividad de sistemas y personas, haciendo más eficiente a la organización.

### 4.5.2.2 Desarrollo de Soluciones

Desarrollo de soluciones personalizadas tanto para proyectos de telefonía como de procesos de negocio. Empleamos una metodología ágil (SCRUM) para asegurarnos que no existan desviaciones entre la visión del cliente y el producto final.

### 4.5.2.3 Desarrollo de Productos Propios

Desarrollo productos propios que cubran necesidades específicas del mercado, teniendo en mente que la diversificación de clientes para un mismo producto se refleja en precios más accesibles.

#### **4.5.2.4 Implementaciones de PBX (Conmutadores) para Pequeñas y Medianas Empresas**

Soluciones de PBX IP de bajo costo en basa a tecnología Open Source, que incluye todas las funcionalidades requeridas en un conmutador e incluso en un Call Center pequeño.

## 4.6 Misión

* Ofrecer soluciones innovadoras y costeables para lograr convertir la infraestructura de tecnología del cliente en una ventaja competitiva.
* Asegurar la satisfacción del cliente en cada interacción para que Interastar se convierta en su Top-Of-Mind para necesidades tecnológicas.
* Planificar el crecimiento de la organización prudentemente, ofreciendo estabilidad a colaboradores y proveedores.
* Cuidar a los Colaboradores Proveedores y Business Partners en base a prácticas éticas y transparentes.
* Asegurar que los accionistas reciban un retorno de su inversión adecuado, de acuerdo a las condiciones del mercado, sin comprometer la operación y el crecimiento de la empresa.

## 4.7 Visión

Interastar será un PROVEEDOR CONFIABLE para servicios e integraciones de telefonía de gran escala, consultorías de tecnología y soluciones de BI con el más alto nivel de CALIDAD, las MÁS NUEVAS TECNOLOGÍAS, el mejor entendimiento de las necesidades del cliente y la mejor RELACIÓN COSTO/BENEFICIO, reportando así el máximo beneficio para los STAKEHOLDERS de la empresa.

## 4.8 Valores

**Integridad.** Actuamos siempre con integridad y ética.

**Responsabilidad**. Cumplimos nuestras promesas, respaldamos nuestras decisiones y nos responsabilizamos por nuestros errores.

**Simplicidad**. Buscamos simplificar procesos, productos y servicios, facilitando la operación del cliente.

**Innovación**. Incorporamos y desarrollamos nuevas tecnologías continuamente.

**Trabajo en Equipo**. Crecemos en la medida en que nuestros colaboradores se complementen y ayuden.

## 

## 4.9 Asesor externo

**Nombre:** Ing. Marco Villaseñor Amezcua.

**Puesto**: Director de Soluciones de Tecnología (TSG)

**Descripción**: Coordinar y gestionar el proyecto, llevar el control y flujo de trabajo con la metodología (SCRUM), dar asesorías sobre las tecnologías, dar las herramientas y lenguajes a utilizar.

## 4.10 Actividades del residente

**Nombre:** Andres Farias Bautista

**Puesto**: Analista /programador Front-End.

**Descripción**: Dadas la recolección de información y almacenamiento a la base de datos dicha información, buscar e implementar metodologías y herramientas tecnológicas de punta para mostrar dicha información.

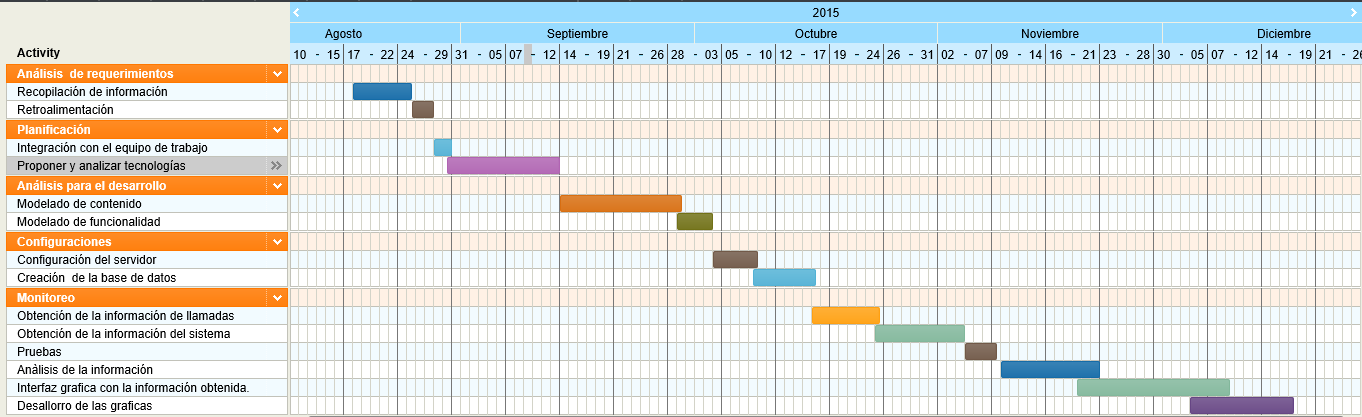
# **PROBLEMAS A RESOLVER, PRIORIZÁNDOLOS**

El proyecto que se presenta en este informe, es desarrollado para el monitoreo en tiempo real de la operación de los Contact Center, el cual nos muestra en gráficas el comportamiento del flujo de las llamadas entrantes, número de llamadas salientes, en espera, esto con la finalidad de llevar un mejor control del trabajo de los asistentes telefónicos.

A continuación se describe las tareas que se realizó para la fase de desarrollo Front-End de Wind dashboard:

1. Mostrar de forma numérica todas las llamadas entrantes, dando parámetros de configuración para el número de minutos de retraso a la hora actual, diseñando esta grafica totalmente configurable (colores, titulo, tiempo de retraso).
2. Mostrar en forma numérica el número de total de llamadas en espera de la misma manera, totalmente configurable (colores, titulo, tiempo de retraso).
3. Mostrar las llamadas totales en forma numérica de acuerdo a un rango de tiempo (8:00 – 11:00), con las mismas características de los anteriores.
4. Mostrar en una gráfica de velocímetro el nivel de servicio de las llamadas totales entrantes al Contact Center. Esto con el rango de los porcentajes configurables.
5. Mostrar en velocímetro el nivel de servicio del Contact Center con rangos de tiempo (8:00 – 11:30 am). Al igual que la gráfica anterior los rangos de los porcentajes configurables.
6. Mostrar en una gráfica lineal el comportamiento del flujo de llamadas entrantes marcando con una línea el nivel de servicio esperado por el Contact Center.

## 5.1 Calendarización del proyecto



**Ilustración III Actividades**

## 5.2 Recursos del proyecto

Referido al personal que participa en el desarrollo del proyecto, desempeñando diversas funciones del proceso de ingeniería de software, el asesor externo a la institución educativa que es el líder del proyecto.

## 5.3 Personal involucrado

**Líder de proyecto:** Marco Villaseñor Amezcua, que se encarga de coordinar las actividades y la arquitectura del proyecto.

**Desarrollador:** Omar Enrique Álvarez, experto en tecnologías JavaScript, dos certificaciones java y actualmente cursando una maestría en Ingeniería de Software en la UNAM, se encargó en el are de Back-End para Wind dashboard.

**Desarrollador:** Andrés Farías Bautista se encargó del Front-End implementando las tecnologías mencionadas anteriormente para una mejor interacción con el usuario final del sistema, uno de los objetivos es resolver gráficamente la necesidad que plantea el usuario.

## 5.4 Hardware

Características de las computadoras en las que se desarrolló el sistema:

**Computadora 1:**

* Marca: Acer.
* Modelo: Aspire One
* Procesador: Intel ® Atom
* Memoria (RAM): 2.00 GB. - Sistema Operativo: Windows 7 - Disco Duro: 250 GB.

**Computadora 2:**

* Marca: Samsung
* Procesador: NP350V4C-A05MX
* Memoria (RAM): 8GB.
* Sistema Operativo: Windows 8.1 Pro
* Disco Duro: 1 TB

**Computadora 3:**

* Marca: Samsung
* Procesador: NP350V4C-A05MX
* Memoria (RAM): 8GB.
* Sistema Operativo: Windows 8.1 Pro
* Disco Duro: 1 TB

## 5.5 Software

A continuación se listan los requerimientos de software para la realización e implementación del proyecto. Software necesario para realizar el sistema:

* + Windows 7 o superior, Linux, Unix.
  + NodeJS.
  + React
  + Language de programación JavaScript.
  + IDE Sublime Text 3.
  + Chrome Versión 43.0.2357.81
  + Yeoman
  + Bower
  + Moment

# **FUNDAMENTO TEORICO**

Para el desarrollo de Wind Dashboard se probaron y analizaron diferentes tecnologías, todas de última generación, a continuación describimos cada tecnología así como las herramientas utilizadas.

## Lenguajes de programación

### CSS3

CSS3 es la última evolución del lenguaje de las Hojas de Estilo en Cascada (Cascading Style Sheets), y pretende ampliar la versión CSS2.1. Trae consigo muchas novedades altamente esperadas , como las esquinas redondeadas, sombras, [gradientes](https://developer.mozilla.org/es/docs/CSS/Using_CSS_gradients), [transiciones](https://developer.mozilla.org/es/docs/Web/CSS/Transiciones_de_CSS) o [animaciones](https://developer.mozilla.org/es/docs/Web/CSS/Usando_animaciones_CSS), y nuevos [layouts](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_Started/Layout) como [multi-columnas](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using_multi-column_layouts" \o "Using CSS multi-column layouts), cajas flexibles o maquetas de diseño en cuadrícula (grid layouts).

Las partes experimentales son particulares para cada navegador y deberían ser evitadas en entornos de producción, o usadas con extrema precaución, ya que tanto la sintaxis como la semántica pueden cambiar en el futuro

(DracotMolver, 2015)

### HTML

HTML es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web. El HTML se creó en un principio con objetivos divulgativos de información con texto y algunas imágenes. No se pensó que llegara a ser utilizado para crear área de ocio y consulta con carácter multimedia (lo que es actualmente la web), de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, sí que se han ido incorporando modificaciones con el tiempo, estos son los estándares del HTML. Numerosos estándares se han presentado ya. El HTML 4.01 es el último estándar a febrero de 2001. Actualización a mayo de 2005, en estos momentos está a punto de presentarse la versión 5 de HTML, de la que ya se tiene un borrador casi definitivo.

(Alvarez, 2001)

### NodeJS

Es un código abierto para el desarrollo [del lado del servidor](https://en.wikipedia.org/wiki/Server-side)[de aplicaciones](https://en.wikipedia.org/wiki/Web_application)web. Es un tiempo de ejecución de JavaScript construido sobre [el motor JavaScript V8 de](https://developers.google.com/v8/)Chrome.  Node.js utiliza el bloqueo no-modelo orientado a eventos de E / S que lo hace ligero y eficiente.

Uno de los puntos fuertes de Node es su capacidad de mantener muchas conexiones abiertas y esperando. En Apache por ejemplo el parámetro, teoría Node puede mantener tantas conexiones como número máximo de archivos descriptores (sockets) soportados por el sistema.

(rmunoz, 2001)

### ReactJS

Es una librería de JavaScript lanzada hace poco más de un año por Facebook, normalmente utilizada en el Front-End aunque puede ser utilizada en el Back-End. Facebook la utiliza en producción para su red social en determinadas partes, como los comentarios y también en Instagram.

(Azaustre, 2015)

### Meteor

Meteor es una plataforma para crear aplicaciones web en tiempo real construida sobre Node.js. Meteor se localiza entre la base de datos de la aplicación y su interfaz de usuario y se encarga que las dos partes estén sincronizadas.

(Greif, 2015)

### 

### 6.1.6 JavaScript

#### **6.1.6.1 Descripción**

**JavaScript**® (a veces abreviado como JS) es un lenguaje ligero e interpretado, orientado a objetos con [funciones de primera clase](https://en.wikipedia.org/wiki/First-class_functions), más conocido como el lenguaje de script para páginas web, pero también [usado en muchos entornos sin navegador](http://en.wikipedia.org/wiki/JavaScript#Uses_outside_web_pages), tales como  [node.js](http://nodejs.org/)**o**[Apache CouchDB](http://couchdb.apache.org/)**.** Es un lenguaje script multi-paradigma, [basado en prototipos](https://en.wikipedia.org/wiki/Prototype-based),  dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa.

El estándar de JavaScript es [ECMAScript](https://developer.mozilla.org/en-US/docs/JavaScript/Language_Resources). Desde el 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en Julio. Se puede seguir el progreso actual de las diferentes, nuevas y mejoradas características en la [correspondiente wiki](http://wiki.ecmascript.org/doku.php?id=harmony:proposals).

(Pablo\_Ivan, 2015 )

En caso de este proyecto Wind dashboard ha sido desarrollado totalmente con tecnologías JavaScript.

#### **6.1.6.2 Funcionamiento**

Se utiliza mucho en la creación de páginas web para conseguir que éstas sean más dinámicas. JavaScript está basado en scripts que se introducen en las etiquetas de una página web para producir los comportamientos que desea su creador. No hay que saber programar con este lenguaje para poder utilizar de forma gratuita los muchos scripts que se encuentran en páginas de Internet. Sólo hay que saber dónde ponerlos dentro del código fuente del documento. Las posibilidades que tiene el lenguaje JavaScript son amplísimas y se utilizan mucho en la Red con los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.

(Valero, 2010)

#### **6.1.6.3 Implementación**

Este lenguaje de programación se utiliza principalmente para establecer y darle funcionalidad intuitiva a la parte del Front-End para Wind dashboard Implementada de manera directa en colaboración con las diferentes librerías que se mencionan a continuación.

## Librerías javascript

### 6.2.1 MomentJS

Es una librería para Javascript que nos permite un manejo avanzado de las fechas: validarlas, manipularlas y formatearlas.

(Atmosfera, 2015)

### D3JS

Es una librería de Javascript utilizada principalmente para crear gráficos y visualizaciones interactivas en un sitio web. Esta herramienta de visualización de datos hace uso principalmente de: gráficos vectoriales (SVG), Javascript, HTML5 y CSS3. D3.js permite usar grandes cantidades de datos en distintos formatos: JSON, CSV, GeoJSON, etc.  Además de todo esto, D3.js nos permite poder crear mapas vectoriales con los cuales podemos crear grandes visualizaciones. Nos enfocaremos principalmente en la creación de mapas.

(Antonio, 2014)

### Jquery

Es una biblioteca JavaScript rápido, pequeño y rico en funciones. Hace las cosas como HTML recorrido y manipulación de documentos, manejo de eventos, animación, y Ajax mucho más simple con una API fácil de usar que funciona a través de una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript.

(Alianza, s.f.)

### Browserify

Es una herramienta open source que nos permite crear módulos en el cliente, utilizando la misma sintaxis que en Node (CommonJS). Por lo tanto, vamos a poder requerir y exportar módulos y manejar sus dependencias como en Node pero en el Browser.

Una gran ventaja es que nos permite usar npm para instalar y manejar las dependencias de nuestros módulos. Por lo que podemos “requerir” cualquier modulo que se encuentre publicado en npm, o bien, utilizar módulos privados.

(Paz, s.f.)

### Yeoman

Es un sistema de generación de proyectos escrito en **JavaScript (Node.js)** que permite **agilizar muchísimo el** [inicio de nuevas aplicaciones](https://www.irontec.com/internet). Está compuesto de un módulo de Node.js sobre el que construir los generadores y una herramienta de línea de comandos para ejecutarlos, todo empaquetado en un único módulo en npm bajo el nombre de ‘yo’. Los generadores se instalan también a través de npm, lo cual permite que se genere [un gran archivo](http://yeoman.io/generators/) con publicaciones open source tanto individuales como de colectivos.

(Jiménez, 2015)

## Base de datos

### 6.3.1 Influxdb

Es un servidor de base de datos de series de tiempo (timeseries), ideal para logs o datos para gráficas que se generen en vivo (dashboards).

Escrita en Go, lenguaje de programación inspirado en la sintaxis de C y desarrollado por Google, InfluxDB es un proyecto open source especialmente pensado para desarrolladores que buscan un sistema de almacenamiento para sus aplicaciones que operen en tiempo real.

(okitup, 2015)

## Tecnologías de comunicación Back-End Front-End

### 6.4.1 JSON

El objeto JSON contiene métodos para analizar [JavaScript Object Notation](http://json.org/) (JSON) y convertir valores a JSON. No puede ser llamado o construído, y aparte de estas dos propiedades, no tiene funcionalidad interesante por sí mismo.

JSON es una sintaxis para serializar objetos, arreglos, números, cadenas, booleanos y nulos. Está basado sobre sintaxis JavaScript pero es diferente a ella: algo JavaScript no es JSON, y algo JSON no es JavaScript.

(ulisestrujillo, 2014)

### 6.4.2 API REST

Actualmente, las API REST están realmente de moda: parece que cualquier aplicación deba proporcionar su “API REST”. Pero... ¿qué significa realmente una API REST?

REST deriva de "REpresentational State Transfer", que traducido vendría a ser “transferencia de representación de estado”, lo que tampoco aclara mucho, pero contiene la clave de lo que significa. Porque la clave de REST es que un servicio REST no tiene estado (es *stateless*), lo que quiere decir que, entre dos llamadas cualesquiera, el servicio pierde todos sus datos. Esto es, que no se puede llamar a un servicio REST y pasarle unos datos (p. ej. un usuario y una contraseña) y esperar que “nos recuerde” en la siguiente petición. De ahí el nombre: el estado lo mantiene el cliente y por lo tanto es el cliente quien debe pasar el estado en cada llamada. Si quiero que un servicio REST me recuerde, debo pasarle quien soy en cada llamada. Eso puede ser un usuario y una contraseña, un *token* o cualquier otro tipo de credenciales, pero debo pasarlas en cada llamada. Y lo mismo aplica para el resto de información.

(Thomas, 2014)

## 6.5 Herramientas

6.5.1 Git (para el control de versiones)

Es un sistema de control de versiones ampliamente utilizado para el desarrollo de software, diseñado para manejar todo tipo de proyectos con rapidez y eficiencia.

(bitbucket.org, s.f.)

### Bitbucker

Es un [servicio de alojamiento basado en la web](https://en.wikipedia.org/wiki/Shared_web_hosting_service) para los proyectos que utilizan ya sea el [Mercurial](https://en.wikipedia.org/wiki/Mercurial) o [Git](https://en.wikipedia.org/wiki/Git_(software)" \o "Git (software)) [de control de revisiones](https://en.wikipedia.org/wiki/Revision_control) sistemas.  Ofrece ambos planes comerciales y cuentas gratuitas. Se ofrece cuentas gratuitas con un número ilimitado de repositorios privados (que puede tener hasta cinco usuarios en el caso de las cuentas gratuitas)

(bitbucket.org, s.f.)

### Sublime Text (IDE)

Es un [editor de texto](https://es.wikipedia.org/wiki/Editor_de_texto) y [editor de código fuente](https://es.wikipedia.org/wiki/Editor_de_c%C3%B3digo_fuente) está escrito en [C++](https://es.wikipedia.org/wiki/C%2B%2B) y [Python](https://es.wikipedia.org/wiki/Python) para los plugins. Desarrollado originalmente como una extensión de [Vim](https://es.wikipedia.org/wiki/Vim" \o "Vim), con el tiempo fue creando una identidad propia.

(wikipedia, https://es.wikipedia.org/wiki/Sublime\_Text, 2015)

### Source Tree

Source Tree es un potente GUI (Graphical User Interface – Interfaz Gráfica de Usuario) para gestionar todos tus repositorios ya sean Git o Mercurial. Con Source Tree podemos crear, clonar, hacer commit, push, pull, merge y algunas cosas más de una forma bastante fácil. Desarrollado por Atlassian e inicialmente solo para Mac, también cuenta con su versión para Windows

(DLRJAD, 2014)

# **PROCEDIMIENTO Y DESCRIPCION DE LAS ACTIVIDADES REALIZADAS**

Para el desarrollo de Wind dashboard manejo el modelo SCRUM ([conjunto de buenas prácticas](http://proyectosagiles.org/fundamentos-de-scrum) para **trabajar colaborativamente, en equipo)** que es un modelo ágil orientado a tareas, la mejor práctica de desarrollo de proyectos en equipo.

## 7.1 Ámbito del sistema

Wind dashboard será un sistema de monitoreo de las operaciones del contact center que mostrará de forma gráfica las entradas y salidas de llamadas.

Estará implementado bajo la arquitectura del FLUX.

## 7.2 Descripción general del software

Wind dashboard es un sistema totalmente independiente y configurable, capaz de almacenar grandes cantidades de información sobre las llamadas, puede hacer una conexión a una base de datos externa sin importar que tipo de base de datos sea utilizando un API REST y almacenando el resultados de esta información a una base de datos propia del sistema.

## 7.3 Reglas de negocio

El usuario solo podrá ver las gráficas de Wind dashboard.

El usuario tendrá que conectar el dispositivo del sistema a una pantalla tv para poder visualizar las gráficas.

El usuario dará a Interastar la configuración deseada.

## 7.4 Lenguajes de programación

Los lenguajes de programación señalados a continuación serán implementados en el desarrollo de Wind dashboard.

* **React:** Para los componentes del sistema.
* **NodeJS:** Procesos del sistema del lado del servidor.
* **HTML:** Construcción de vistas en el sistema.
* **CSS:** Construcción de estilos en las vistas del sistema.
* **JSON**: Configuración en el uso el sistema.

## Requisitos futuros.

Wind dashboard será configurable en su versión 2, en donde el usuario podrá configurar y personalizar los colores tamaños posiciones de las gráficas.

## 7.6 Descripción del medio ambiente de desarrollo

Se muestra un listado de las restricciones que se deben considerar acorde al desarrollo de Wind dashboard.

Restricciones del sistema

Los módulos deberán ser aplicaciones para TV, desarrolladas en lenguaje de programación Meteor y React, utilizando la arquitectura Flux(Arquitectura de programación lineal). El líder de proyecto definirá las herramientas y lenguajes de programación a utilizar en el desarrollo del proyecto.

Reglas del sistema:

* Debe ser intuitiva.
* Segura
* Modular
* Escalable
* Universal
* Arquitectura Flux
* Pruebas
* Reusabilidad de Código
* Que pueda correr con el hardware mínimo requerido

## 7.7 Protocolo de comunicación

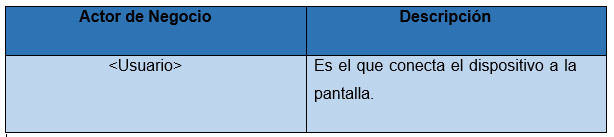
Los protocolos señalados serán implementados en el desarrollo del Wind dashboard.

* **TCP/IP**: Protocolo de control de transmisión/Protocolo de internet.
* **HTTP:** Protocolo de Transferencia de Hipertexto.
* **FTP:** Protocolo de transferencia de archivos
* **SSH:** Seccure Shell, intérprete de órdenes seguro.

## 7.8 Software necesario en el servidor para ejecutar el sistema

* Manejador de base de datos InfluxDB.
* Sistema operativo Raspberry PI.
* Lenguaje de Programación Meteor y React
* Pantalla TV.

## 7.9 Descripción del actor del sistema

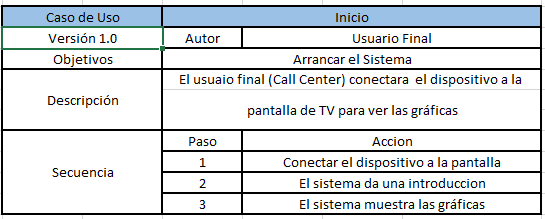


**Ilustración IV Definición resumida de los actores del sistema**

## 7.10 Diagramas de caso de Uso para el inicio de Wind dashboard



**Ilustración V Caso de uso de inicio del sistema**

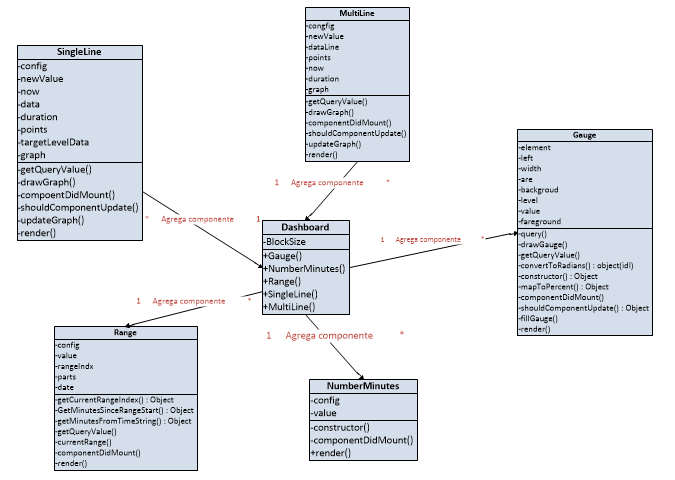


**Ilustración VI Inicio del Sistema**

## 7.11 Diagrama de Clases

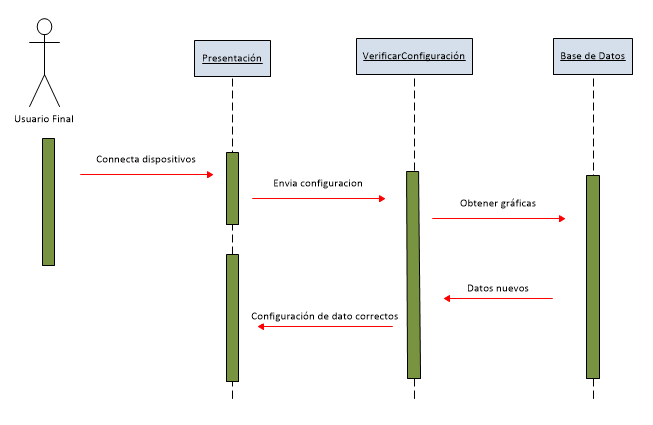
El diagrama de clases muestra la estructura estática de Wind dashboard. La notación utilizada en este diagrama se describe a continuación.

|  |  |
| --- | --- |
|  | Clase. Se representa por medio de un rectángulo dividido en tres partes. La primera parte almacena el nombre de la clase, la segunda parte enlista los atributos que forman parte de la clase, estos atributos tienen asignada una visibilidad, la cual indica si el atributo puede ser accedido por otras clases (+) o solo por la clase que los define (-). |
|  | Asociación. Indica que existe una relacion estatica entre clases. La flecha y etiqueta establecen la dirección y el nombre de la asociación. Los símbolos al inicio y al final de la asociación indica la multiplicidad. La multiplicidad establece el número de objetos que tiene que ver en la asocición. |



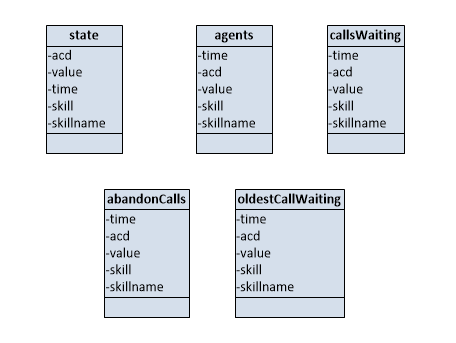
**Ilustración VII Diagrama de clases**

## 7.12 Diagrama de Secuencia



**Ilustración VIII Diagrama de secuencia**

## 7.13 Base de datos



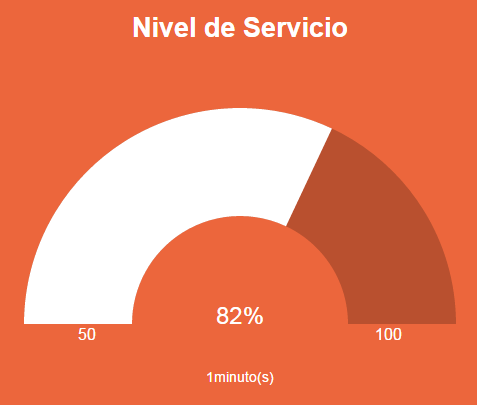
**Ilustración IX Base de datos**

# **RESULTADOS, PLANOS, GRAFICAS.**

Wind dashboard tiene una interfaz completa para la visualización rápida del comportamiento de las llamadas, las tecnologías con las cuales ha sido desarrollado nos han permitido dar una apariencia adecuada. Todas las gráficas obtienen la información en tiempo real.

## 8.1 Gráfica de Gauge

La gráfica de gauge nos muestra el nivel de servicio que está teniendo el call center, mostrando en porcentaje el resultado.

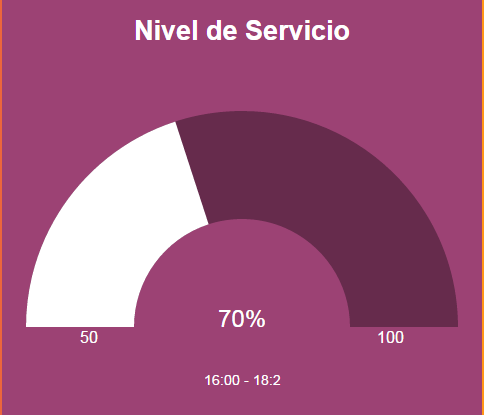


**Ilustración X Grafica de Gauge mostrando el nivel de servicio.**

Esta gráfica puede obtener y mostrar cualquier otro comportamiento, como el porcentaje de llamadas en espera, porcentaje de las llamadas entrantes, ya que solo es cuestión de indicar a que es lo que se desea mostrar.

**8.2 Gráfica GaugeRange.**

La gráfica de GaugeRange nos muestra el nivel de servicio en rango de tiempo que está teniendo el call center, mostrando en porcentaje el resultado.



Esta gráfica puede obtener y mostrar cualquier comportamiento, como el porcentaje de llamadas en espera, porcentaje de las llamadas entrantes, solo que a diferencia del anterior, este lo muestra en rangos de tiempo, y los rangos definen el periodo de tiempos en que se saca el nivel de servicio.

## 8.2 Grafica NumberMinutes

Esta gráfica nos muestra el conteo del número de llamadas en espera, mostrándolo de forma numérica, también muestra el número de minutos de retraso de la hora actual en la que se está obteniendo esta información.



**Ilustración XI Grafica NumberMinutes.**

De igual manera esta grafica puede mostrar cualquier tipo de comportamiento de las llamadas, personalizando el título y la cantidad de minutos de retraso de las consultas.

## 8.3 Grafica Numeric.

Esta gráfica muestra de forma numérica el comportamiento de las llamadas, en este caso, en la ilustración 1(número de imagen) nos muestra el número de agentes formados, obteniendo esta información en tiempo real.



**Ilustración XII Número de agentes firmados.**

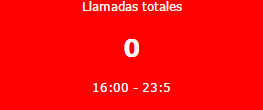
Esta otra gráfica muestra las llamadas en espera.



**Ilustración XIII Numero de llamas en espera.**

## 8.4 Gráfica Range

Esta gráfica muestra el comportamiento de las llamadas con rangos de tiempos, en la ilustracion está mostrando el número de llamadas totales que hay desde las 16:00 hasta las 24:00, la segunda hora (24:00), es la hora actual. Podemos configurar estas horas por rangos de tiempo, por ejemplo: de 10:00 a 01:00, de 01:00 a 16:00, de 16:00 a 24:00, la información a mostrar es configurable, la gráfica puedría mostrar el número de agentes firmados, número de llamas entrantes, etc.

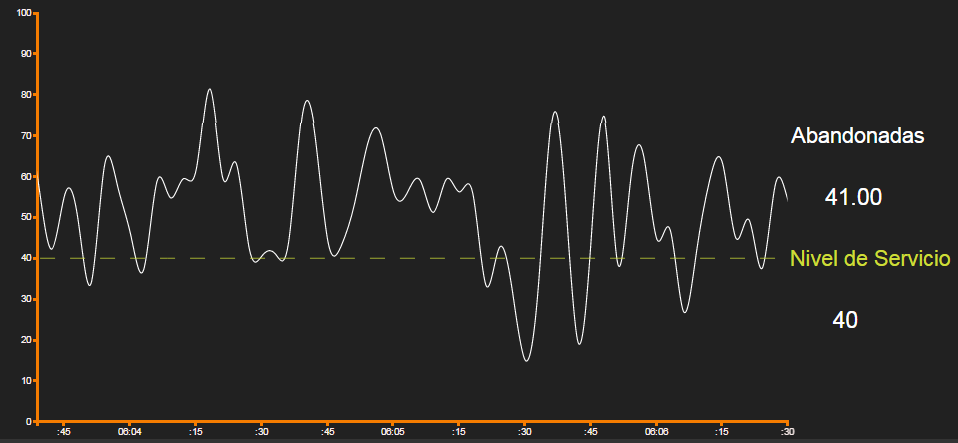


**Ilustración XIV Grafica Range.**

## Grafica SingleLine

Es una gráfica lineal, en su eje x muestra los minutos, y en el eje y muestra el número de llamadas en porcentaje del 10% al 100%. Tiene una línea punteada la cual indica el nivel de servicio esperado por el call center, esta línea es configurable permitiendo pasar como parámetro el nivel de servicio esperado, de igual forma el comportamiento de las llamadas.

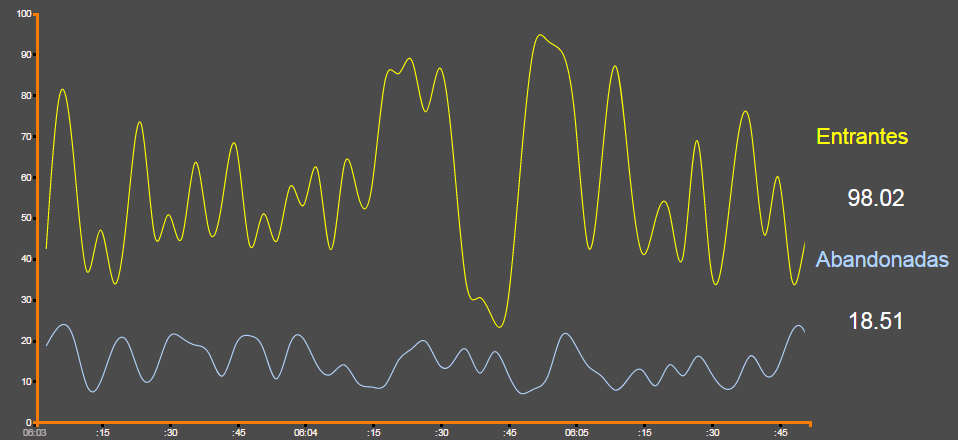
En la ilustración se está mostrando el promedio de las llamadas respondidas en un minuto.



**Ilustración XV Grafica SingleLine.**

## 8.6 Grafica MultiLine.

Esta gráfica tiene las mismas características que la gráfica SingleLine con respecto a los ejes, dicha gráfica puede mostrar dos líneas en tiempo real sobre el comportamiento de las llamadas, en esta imagen estamos mostrando el número de llamadas en respondidas (línea morada) y las llamadas en espera (línea amarilla).



**Ilustración XVI Grafica MultiLine.**

## Wind dahsboard

Esta gráfica es el resultado final, en esta nos muestra de forma general un panorama amplio sobre todos los resultados obtenidos, es decir en una sola pantalla todas las gráficas con sus respectivos resultados sobre el comportamiento de las llamadas.



Ilustración XVII Wind dahsboard

# **RECOMENDACIONES**

Wind dashboard en esta fase está desarrollado principalmente para TV, por lo que para mejor conviene disponer de una pantalla de TV, aunque el sistema es adaptable a resoluciones de pantallas de computadoras y tabletas.

# **CONCLUSIONES**

Llevar el control de todos las entradas y salidas de una organización de manera general es muy importante, este sistema tiene un módulo de recolección de datos, módulos de interfaz gráfica, muestra los números, muestra en graficas hace que sea más fácil la comprensión y análisis y con esto determinar cierto criterios para la mejora de la operación de la empresa, dando soluciones más rápidas y precisas.

Hoy en día gracias a los avances en tecnologías podemos ocupar herramientas que nos facilitan la obtención de datos y mostrarlos en gráficas para su fácil comprensión como en este trabajo se realizó.

Wind dashboard es un herramienta viable para poder llevar la recopilación de datos en tiempo real.

En las siguientes versiones, Wind dashboard podrá ser totalmente configurable y con tecnología Touch, con apoyos de diseñadores gráficos, dando un aspecto totalmente amigable.

# **COMPETENCIAS DESARROLLADAS Y/O APLICADAS**

En este proyecto se desarrolló principalmente el autodidactismo, el miedo es el principal obstáculo que el alumno tiene, en este proyecto se aprendió a trabajar en equipo ordenadamente, cuidando la legibilidad del código línea a línea, detalladamente, se aprendió a trabajar respetando las reglas sugeridas por expertos en cada tecnología utilizada.

# **REFERENCIAS BIBLIOGRAFICAS**

1. Alianza, L. (s.f.). *http://alianza.hol.es/que-es-jquery/*. Obtenido de alianza.hol.es: http://alianza.hol.es/que-es-jquery/
2. Alvarez, P. M. (1 de Enero de 2001). *http://www.desarrolloweb.com/articulos/que-es-html.html*. Obtenido de desarrolloweb.com: http://www.desarrolloweb.com/articulos/que-es-html.html
3. Antonio. (15 de Diciembre de 2014). *http://es.schoolofdata.org/tag/d3-js/*. Obtenido de ESCUELA DE DATOS: http://es.schoolofdata.org/tag/d3-js/
4. Atmosfera. (6 de Agossto de 2015). *https://atmospherejs.com/momentjs/moment*. Obtenido de Atmosfera: https://atmospherejs.com/momentjs/moment
5. Azaustre, C. (22 de Junio de 2015). *https://carlosazaustre.es/blog/empezando-con-react-js-y-ecmascript-6/*. Obtenido de Carlos Azaustre.es: https://carlosazaustre.es/blog/empezando-con-react-js-y-ecmascript-6/
6. bitbucket.org. (s.f.). *http://michelletorres.mx/opciones-de-hosting-con-soporte-para-repositorios-git/*. Obtenido de https://bitbucket.org/: http://michelletorres.mx/opciones-de-hosting-con-soporte-para-repositorios-git/
7. DLRJAD. (9 de Junio de 2014). *https://jadcode.wordpress.com/2014/06/09/source-tree/*. Obtenido de JADCODE: https://jadcode.wordpress.com/2014/06/09/source-tree/
8. DracotMolver. (2015 de Diciembre de 2015). *https://developer.mozilla.org/es/docs/Web/CSS/CSS3*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/CSS/CSS3
9. Greif, T. C. (27 de Junio de 2015). *http://es.discovermeteor.com/pdf*. Obtenido de discovermeteor.com: http://es.discovermeteor.com/pdf
10. Jiménez, A. L. (15 de Septiembre de 2015). *https://blog.irontec.com/estructurando-generadores-de-yeoman/*. Obtenido de irontec.com: https://blog.irontec.com/estructurando-generadores-de-yeoman/
11. okitup. (10 de Abril de 2015). *https://www.okitup.com/blog/que-es-influxdb/*. Obtenido de www.okitup.com: https://www.okitup.com/blog/que-es-influxdb/
12. Pablo\_Ivan. (24 de julio de 2015 ). *MDN*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/JavaScript
13. Paz, G. (s.f.). *https://getmango.com/blog/construyendo-modulos-de-front-end-con-browserify/*. Obtenido de getmango.com: https://getmango.com/blog/construyendo-modulos-de-front-end-con-browserify/
14. rmunoz. (1 de Mayo de 2001). *http://www.rmunoz.net/introduccion-a-node-js.html*. Obtenido de rmunoz: http://www.rmunoz.net/introduccion-a-node-js.html
15. Thomas, E. (25 de Abril de 2014). *http://www.desarrolloweb.com/articulos/que-es-rest-caracteristicas-sistemas.html*. Obtenido de desarrolloweb.com: http://www.desarrolloweb.com/articulos/que-es-rest-caracteristicas-sistemas.html
16. ulisestrujillo. (26 de Septiembre de 2014). *MDN*. Obtenido de MDN: https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos\_globales/JSON
17. Valero, A. (15 de Agosto de 2010). *http://fresno.pntic.mec.es/avaler3/webs/leccion\_08.html*. Obtenido de http://fresno.pntic.mec.es/: http://fresno.pntic.mec.es/avaler3/webs/leccion\_08.html