**INSTITUTO TECNOLÓGICO DE MILPA ALTA**

**INGENIERIA EN SISTEMAS COMPUTACIONALES**

**MANUAL TÉCNICO**

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** 4](#_Toc439975976)

[**2.** **OBJETIVOS** 5](#_Toc439975977)

[**2.1 GENERAL DEL SISTEMA**. 5](#_Toc439975978)

[**2.2 OBJETIVOS ESPECÍFICOS** 5](#_Toc439975979)

[**3.** **CONTENIDO TÉCNICO** 6](#_Toc439975980)

[**3.1 JAVASCIPT 6** 6](#_Toc439975981)

[3.1.1 Principales novedades de ES6 6](#_Toc439975982)

[**3.2 METEOR** 7](#_Toc439975983)

[**3.3 REACT** 8](#_Toc439975984)

[**3.4 INFLUX** 10](#_Toc439975985)

[**4.** **LENGUAJES DE PROGRAMACION** 11](#_Toc439975986)

[4.1 NodeJS 11](#_Toc439975987)

[4.2 Bower 11](#_Toc439975988)

[4.3 Yeoman 12](#_Toc439975989)

[**5.** **LIBRERIAS** 12](#_Toc439975990)

[**5.1 Momento** 12](#_Toc439975991)

[**5.2 D3JS** 12](#_Toc439975992)

[**5.3 Jquery** 13](#_Toc439975993)

[**5.4 Browserify** 13](#_Toc439975994)

[**6.** **BASE DE DATOS** 13](#_Toc439975995)

[**6.1 Mongodb 3** 13](#_Toc439975996)

[**6.2 Influxdb** 13](#_Toc439975997)

[**7.** **HERRAMIENTAS** 14](#_Toc439975998)

[**7.1 Git** 14](#_Toc439975999)

[**7.2 Bitbucker** 14](#_Toc439976000)

[**7.3 Sublime Text (IDE)** 14](#_Toc439976001)

[**7.4 Source Tree** 15](#_Toc439976002)

[**7.5 Bitbucket:** 16](#_Toc439976003)

[**8.** **Diagramas.** 17](#_Toc439976004)

**TABLA DE ILUSTRACIONES**

[Ilustración I Meteor 8](#_Toc439976005)

[Ilustración II Influx 10](#_Toc439976006)

[Ilustración III Source tree 1. 15](#_Toc439976007)

[Ilustración IV Source tree 2. 15](#_Toc439976008)

[Ilustración V Estructura del proyecto con Meteor.](#_Toc439976009) 17

[Ilustración VI Diagrama de secuencia.](#_Toc439976010) 18

[Ilustración VII Diagrama de caso de uso.](#_Toc439976011) 19

[Ilustración VIII Base de Datos.](#_Toc439976012) 20

# **INTRODUCCIÓN**

Wind dashboard (Consola de monitoreo inteligente), es un sistema para monitorear la operación de un call center, mostrando información en tiempo real del comportamiento de las llamadas de forma gráfica, la cual nos enfocamos a la parte Front-End. Este sistema es desarrollado con tecnologías de punta, las mejores prácticas de programación que hay en la actualidad, para el desarrollo de este sistema se han manejado herramientas como el control de versiones Git, Source Tree, la arquitectura Flux, que es la arquitectura más actual y que promete ser mejor que el modelo vista controlador.

En este documento se explica todo sobre el comportamiento de Wind dashboard y mostramos como se han implementado las tecnologías.

# **OBJETIVOS**

## **2.1 GENERAL DEL SISTEMA**.

Obtener un monitoreo en tiempo real del comportamiento de la operación del Contact Center, mostrando mediante gráficas el resultado del monitoreo, con información precisa y configurable.

## **2.2 OBJETIVOS ESPECÍFICOS**

* Desarrollo del sistema con tecnología de punta.
* Seguridad al sistema.
* Mostrar información oportuna.
* Dar una experiencia de usuario muy amigable

# **CONTENIDO TÉCNICO**

Wind dashboard se ha desarrollado con tecnologías open source, lo cual genera un costo de desarrollo menor.

Todo el proyecto (Frond-end y back-end) se ha desarrollado con las tecnologías Full stack JavaScript: Desarrollo integral de aplicaciones Web con JavaScript, todas las tecnologías son de JavaScript.

## **3.1 JAVASCIPT 6**

ECMAScript 2015 (6ta edición) es la versión actual de la especificación del lenguaje ECMAScript conocida simplemente como "ES6", esta versión define el estándar para la implementación de [JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript) en [SpiderMonkey](https://developer.mozilla.org/en-US/docs/Mozilla/Projects/SpiderMonkey) el motor utilizado por Firefox y otras aplicaciones de Mozilla.

El nombre clave es "ES.next" o "Harmony", el primer borrador de la especificación (basada en [ECMAScript 5.1](https://developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_5_support_in_Mozilla)) fue publicada el 12 de Julio de 2011 como "ES.next", en Agosto de 2014 el borrador de la especificación fue detenido respecto de nuevas características y pasó por un periodo de estabilización y soluciones de bugs, finalmente, el ECMA-262 edición 6 fue oficialmente aprobada y publicada en Junio 17 de 2015 por el ECMA General Assembly. También aparecerá como el ISO/IEC 16262:2016.

### 3.1.1 Principales novedades de ES6

ES6 trae muchos cambios significativos al lenguaje. Veamos algunos de ellos:

Función Arrow, Clases, this, let y const, template Strings, destructuring, valores por defecto, módulos, entre otros.

## **3.2 METEOR**

Meteor permite la creación rápida de prototipos y produce multiplataforma de código (web, Android, iOS). Se integra con [MongoDB](https://en.wikipedia.org/wiki/MongoDB) y utiliza el [Protocolo de datos distribuidos](https://en.wikipedia.org/wiki/Distributed_Data_Protocol) y un [patrón de publicación-suscripción](https://en.wikipedia.org/wiki/Publish%E2%80%93subscribe_pattern) para propagar automáticamente los cambios de datos a los clientes sin que el desarrollador que escribir ningún código de sincronización

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.

Como Meteor usa Node.js, se utiliza JavaScript en el cliente y en el servidor. Y más aún, Meteor es capaz de compartir código entre ambos entornos.

El resultado es una plataforma muy potente y muy sencilla ya que Meteor abstrae muchas de las molestias y dificultades que nos encontramos habitualmente en el desarrollo de aplicaciones web.

Meteor es “full stack” y es fácil de aprender.

Meteor permite crear una aplicación web en tiempo real en cuestión de horas. Y si ya hemos hecho desarrollo web, estaremos familiarizados con JavaScript, y ni siquiera tendremos que aprender un nuevo lenguaje.



## **3.3 REACT**

React components implementan un render () método que toma datos de entrada y devuelve lo que se vea. Es una biblioteca de JavaScript para la construcción de interfaces de usuario normalmente utilizada en el frontend aunque puede ser utilizada en el backend. Proporciona una vista de los datos prestados como HTML.

Es mantenido por Facebook, Instagram y una comunidad de desarrolladores y empresas individuales. Facebook la utiliza en producción para su red social en determinadas partes, como los comentarios

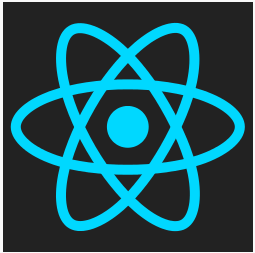


Ilustración I Meteor

En la siguiente imagen mostramos parte del código Meteor con React.

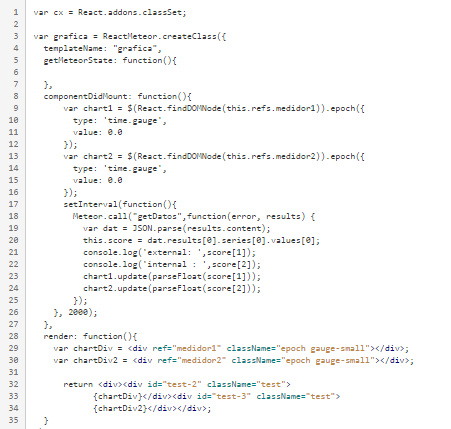


Ilustración ll. Código React.

En la imagen estamos creando el componente llamado medidor1 y medidor2.

## **3.4 INFLUX**

Se enfoca en crear flujos de datos explícitos y entendibles, lo cual hace más sencillo seguir los cambios en el estado de la aplicación y por ende los errores más fáciles de encontrar y corregir.

En MVC el controlador es responsable de coordinar los cambios en 1 o más modelos y lo hace mediante llamadas a métodos en los modelos. Cuando los modelos cambian, se notifican las vistas las cuales a su vez leen los nuevos datos del modelo y se actualizan de acuerdo a esos cambios para que el usuario pueda ver los nuevos datos o estado. Conforme la aplicación crece y se agregan más controladores, modelos y vistas las dependencias se vuelven más complejas

Cuando el usuario interactúa con la interfaz gráfica múltiples ramas de código son ejecutadas y encontrar errores y correr pruebas sobre el estado de la aplicación se vuelve una tarea muy difícil en la cual básicamente se tiene que tratar de adivinar en cuál de todos los puntos se encuentra el error. En los peores casos una interacción causa actualizaciones adicionales en cascada haciendo la tarea aún más dura.

Flux evita este diseño en favor de un flujo de datos en una sola dirección. Todas las interacciones dentro de una vista llaman un “creador de acciones” que a su vez llama el “despachador” de tipo singleton que es responsable de emitir un evento de tipo “acción” al cual los “almacenes” se pueden subscribir. Estos “almacenes” responden a la acción y se auto actualizan.

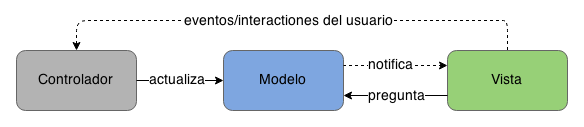


Ilustración II Influx

# **LENGUAJES DE PROGRAMACION**

### 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. C3js

Es un gráfico librería JavaScript en la parte superior de D3js.C3 hace que sea fácil gráficos basados-D3 generar envolviendo el código necesario para construir todo el gráfico. No necesitamos escribir código D3 más. Da algunas clases a cada elemento cuando se genera, por lo que puede definir un estilo personalizado de la clase y es posible extender la estructura directamente por D3. Proporciona una variedad de API y devoluciones de llamada para acceder al estado de la gráfica. Mediante el uso de ellos, se puede actualizar la tabla, incluso después de que se dicte.

### Bower

Es un [sistema de gestión de paquetes](https://en.wikipedia.org/wiki/Package_management_system) de [programación del lado del cliente](https://en.wikipedia.org/wiki/JavaScript#Use_in_web_pages) en la [World Wide](https://en.wikipedia.org/wiki/World_Wide_Web)Web. Depende de [Node.js](https://en.wikipedia.org/wiki/Node.js) y NPM.Funciona con [git](https://en.wikipedia.org/wiki/Git_(software)) y [GitHub](https://en.wikipedia.org/wiki/GitHub)

repositorios.

Los sitios web están hechas de un montón de cosas - marcos, las bibliotecas, los bienes, los servicios públicos, y el arco iris. Bower gestiona todas estas cosas Y trabaja por ir a buscar e instalar los paquetes de todo.

### 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.

# **LIBRERIAS**

## **5.1 Momento**

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

## **5.2 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.

## **5.3 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.

## **5.4 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.

# **BASE DE DATOS**

## **6.1 Mongodb 3**

Es una base de datos orientada a documentos, es decir, en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

## **6.2 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.

# **HERRAMIENTAS**

**7.1 Git** (para el control de versiones)

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

## **7.2 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)) [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)

## **7.3 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), con el tiempo fue creando una identidad propia.

## **7.4 Source Tree**

Es quizás uno de los mejores clientes GUI para manejar repositorios git y mercurial que existen en la actualidad.

Imágenes de Source tree para el control de versiones de Wind dashboard

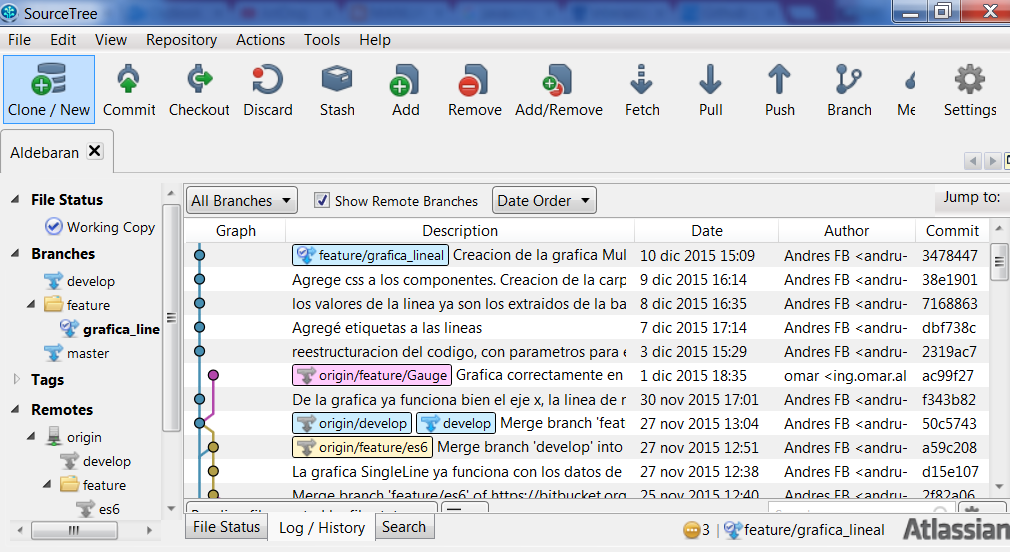


Ilustración III Source tree 1.

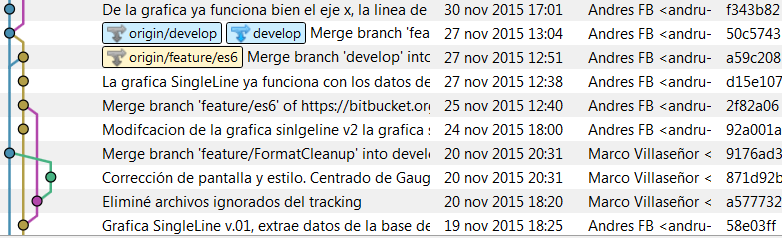
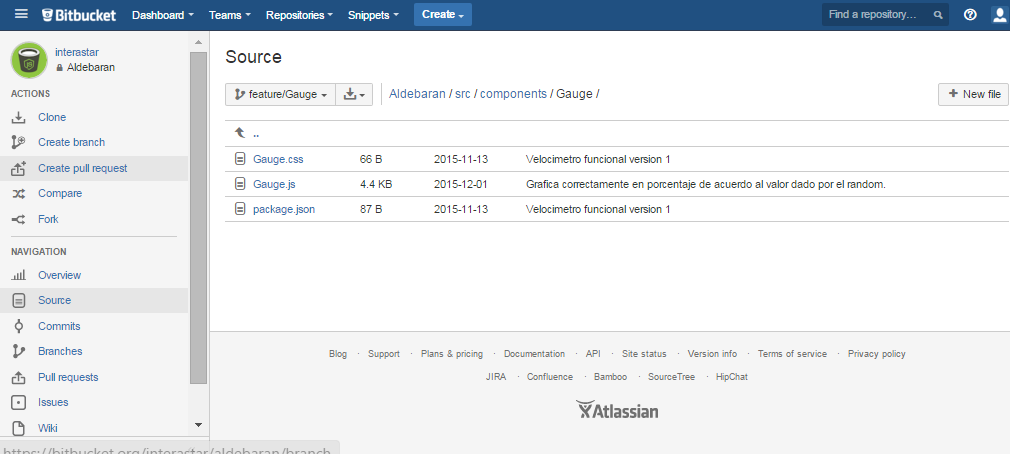


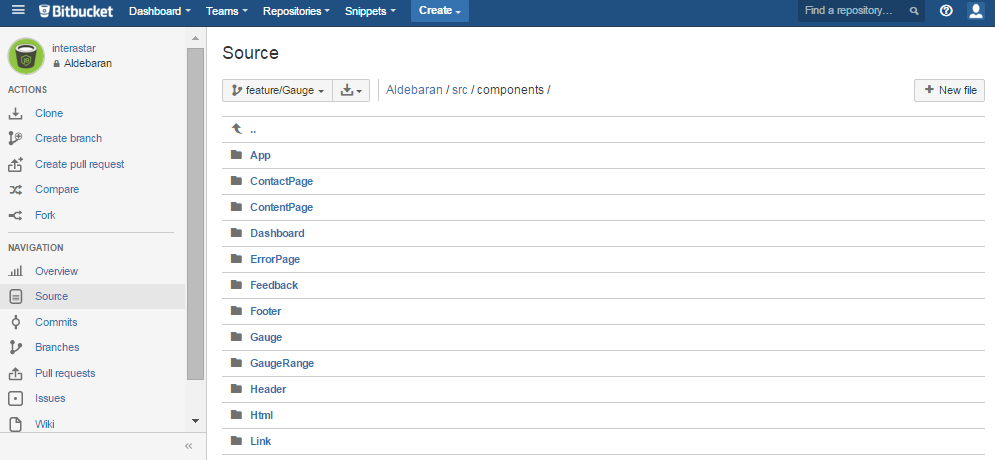
Ilustración IV Source tree 2.

## **7.5 Bitbucket:**

Por varias razones [Git es uno de los controles de versiones más populares](http://bitelia.com/2014/05/git-sistema-control-versiones), pero ¿qué pasa cuando no tienes el hardware, el conocimiento o el tiempo para instalar y mantener tu propio servidor de Git?

Bitbucket es un servicio que te permite administrar tus proyectos usando el popular sistema de control de versiones Git, pero en la nube.





# **Diagramas.**

Para el desarrollo de Wind dashboard manejo el modelo SCRUM que es un modelo ágil orientado a tareas, la mejor práctica de desarrollo de proyectos en equipo, a continuación se muestra la estructura del proyecto con meteor.

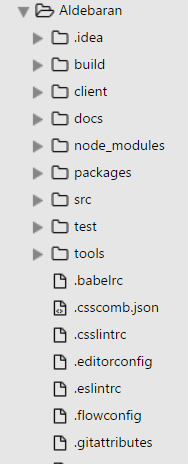


Ilustración V Estructura del proyecto con Meteor.

Esta imagen muestra la estructura del proyecto en meteor. Hay carpetas en donde se colocan solo código para el cliente **client,** carpetas de configuración para todo el proyecto, módulos de node, una carpeta para las pruebas unitarias.

A continuación se muestran los diagramas que se utilizaron:

* Funcionamiento de Wind dashboard, con esta secuencia se hace las tareas generales.

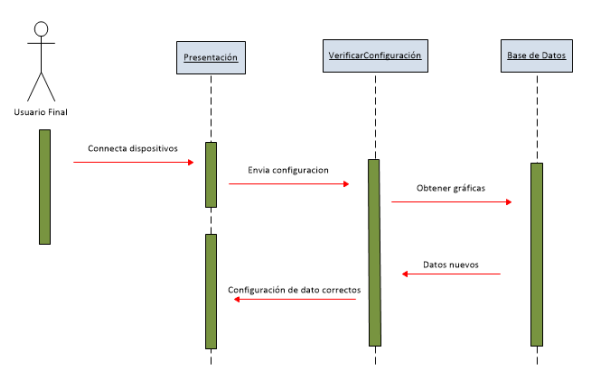


Ilustración VI Diagrama de secuencia.

* Para la visualización de las gráficas.

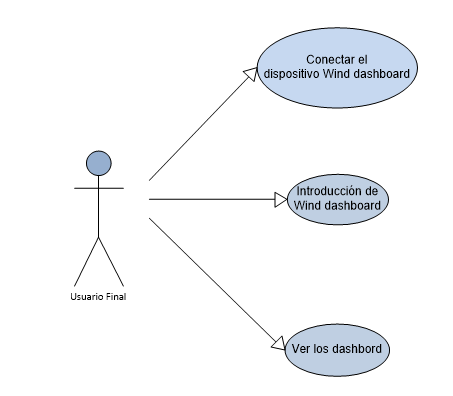


Ilustración VII Diagrama de caso de uso.

* Muestra la base de datos del colector de información a mostrar en el dashboard.

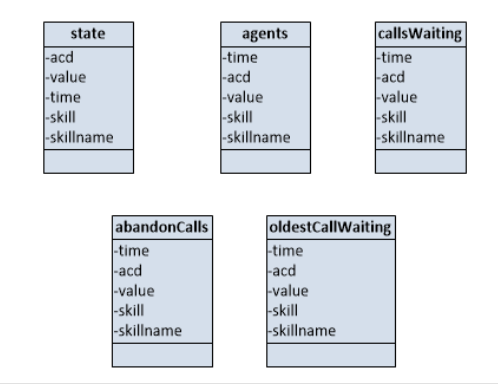


Ilustración VIII Base de Datos.