

Universidad de las Ciencias Informáticas

Facultad 5



**Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias Informáticas**

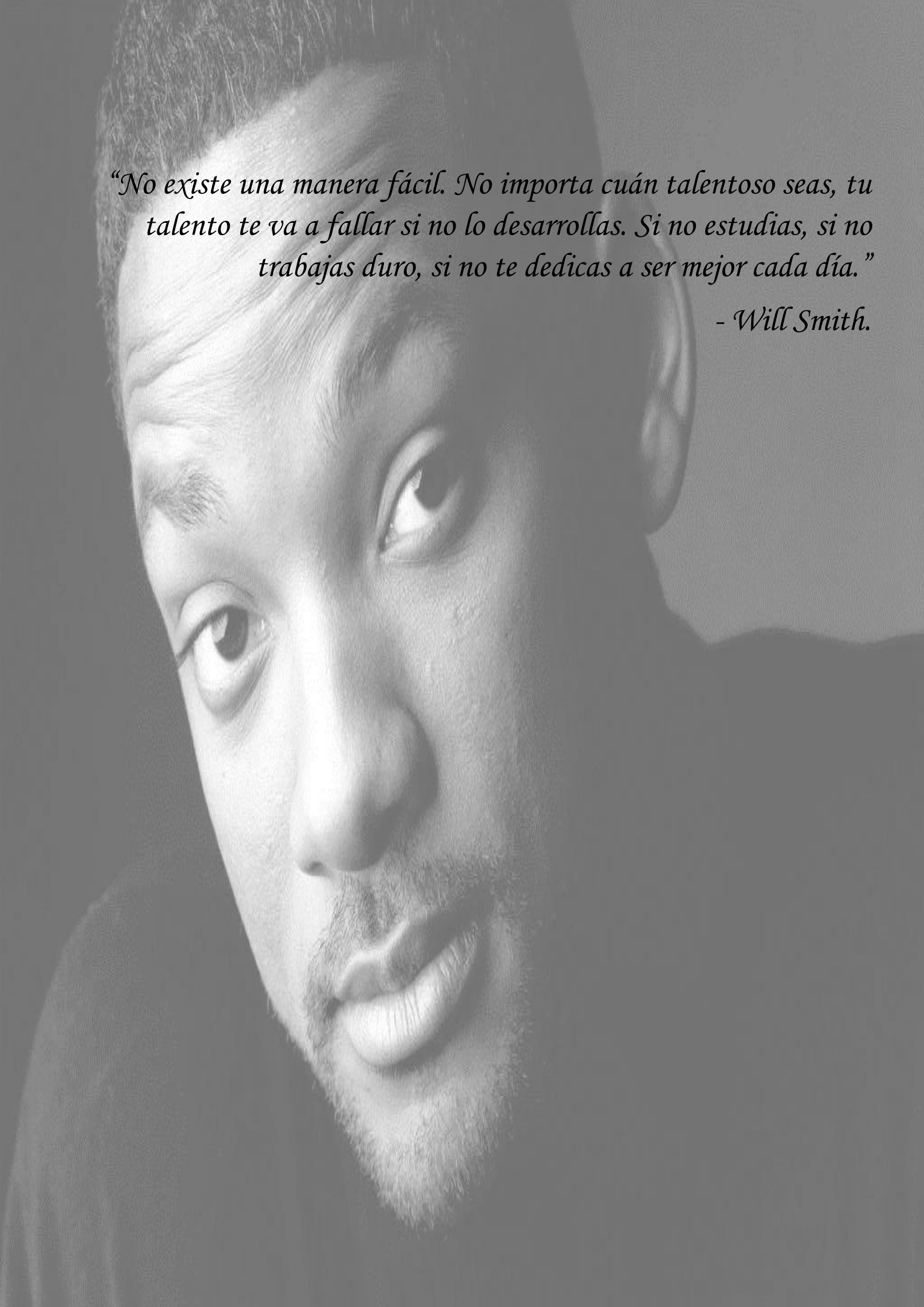
**Módulo de visualización del Cuadro de Mando Integral para el
SCADA SAINUX**

Autor: Luis Alberto García Delgado

Tutor: Ing. Dayrien Corrales Díaz

La Habana, Julio de 2016

“Año 58 de la Revolución”



“No existe una manera fácil. No importa cuán talentoso seas, tu talento te va a fallar si no lo desarrollas. Si no estudias, si no trabajas duro, si no te dedicas a ser mejor cada día.”

- Will Smith.

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Firma del Autor

Luis Alberto García Delgado

Firma del Tutor

Ing. Dayrien Corrales Díaz

Tutor:

Ing. Dayrien Díaz Corrales (dcorrales@uci.cu). Graduado en julio de 2008 de Ingeniero en Ciencias Informáticas. 7 años de experiencia, se desempeña como Especialista “A” en Ciencias Informáticas en el Centro de Informática Industrial, se desempeña como Asesor de Investigación y Postgrado. Maestrante de la segunda edición de la maestría Cibernética aplicada, mención Minería de datos, en el ICIMAF.

Autor:

Luis Alberto García Delgado. (lgdelgado@estudiantes.uci.cu). Estudiante de la Universidad de las Ciencias Informáticas y pertenece al Centro de Informática Industrial.

Dedico esta tesis a mi familia, en especial a mi mamá y mi abuela, por todo el esfuerzo y el apoyo que me han dado durante estos cinco años, por estar siempre en todo momento y por ser mis ejemplos en la vida. A mi novia Jessie por estar siempre a mi lado en los malos y buenos momentos, por entenderme y darme su amor siempre.

Agradezco a mi familia, especial a mi mamá y mi abuela por ser mis guías en la vida y mi ejemplo.

A mi novia Jessie por siempre estar a mi lado, por ser mi amiga y mi apoyo.

A mis amigos, que siempre de una forma u otra han estado presente durante estos cinco años, Alik, Manolo, Yassiel, Yaisel, Juan.

A mis compañeros de grupo, del primero y segundo en el que estuve.

A los profesores que me ayudaron con la realización de este trabajo.

A todos los que de una forma u otra estuvieron presentes en mi trayectoria universitaria.

Resumen

La Universidad de las Ciencias Informáticas cuenta con diversos centros de desarrollo de software, entre ellos el Centro de Informática Industrial, dicho centro desarrolla soluciones que favorecen al sector industrial dentro y fuera del país, soluciones que se denominan Sistemas de Procesamiento Transaccional en Línea(OLTP), estos tipos de sistemas se encargan de generar datos sobre el estado de los componentes que intervienen en los procesos de producción. Actualmente no existe en el centro un sistema capaz de participar en la toma de decisiones de una determinada empresa, involucrándose así en las proyecciones futuras y en un mejor desarrollo a nivel de industria. Por lo antes mencionado, se desea desarrollar una herramienta capaz de generar inteligencia de negocio y participe en las proyecciones futuras de una entidad, supliendo así dicha problemática.

La herramienta que se desea obtener debe aplicar como Cuadro de Mando Integral, generar inteligencia de negocio, extraer información de sistemas de almacenamiento, así como aplicar técnicas de minería de datos para analizar la data extraída mediante módulos dedicados a estas tareas de forma distribuida. Teniendo en cuenta que la información que se maneja en estos tipos de sistemas provienen de disimiles fuentes.

Palabras Clave: Cuadro de Mando Integral, inteligencia de negocios, procesamiento transaccional.

Índice

Introducción	1
Capítulo 1. Fundamentación teórica	5
1. 1 Inteligencia de negocio.....	5
1. 2 Cuadro de mando integral.....	7
1. 3 Formas de representación de la información.....	9
1. 4 Plataformas de software.....	12
1. 5 Aplicaciones web para generar inteligencia de negocio	14
1.5.1 Estructura de una aplicación web.....	14
1.5.2 Características de las aplicaciones web.....	15
1.5.3 Tecnologías para el desarrollo de aplicaciones web.	15
1. 6 Metodología de desarrollo de software.....	17
Conclusiones parciales.....	18
Capítulo 2. Análisis y diseño	19
2. 1 Propuesta de sistema.....	19
2. 2 Descripción del sistema propuesto.....	20
2.2.1 Descripción de los actores	20
2.2.2 Diagrama del dominio	20
2. 3 Descripción de los requisitos de software.....	21
2.3.1 Requisitos funcionales.	21
2.3.2 Descripción de los requisitos funcionales.....	22
2.3.3 Requisitos no funcionales	24
2. 4 Patrón de arquitectura.....	24
2. 5 Casos de uso del sistema	25
2. 6 Diagrama de casos de uso del sistema	26
2.6.1 Patrones de casos de uso.....	26
2. 7 Diagrama de despliegue	27
2. 8 Diagrama de paquetes	28
2.8.1 Descripción de las clases del diagrama de paquetes.....	28
2. 9 Patrones de diseño	29
2. 10 Descripción de los casos de uso del sistema	29
Tabla 2.4 .CU Autenticar usuario	30
Tabla 2.5 CU Gestionar Puntos	30
Conclusiones parciales.....	32
Capítulo 3. Implementación y prueba	33

3. 1	Introducción del capítulo.....	33
3.1.1	Pruebas de caja negra.....	34
3. 2	No conformidades.....	38
3. 3	Resultados de las pruebas de caja negra.....	40
	Conclusiones parciales.....	40
	Conclusiones generales.....	41
	Recomendaciones.....	42
	Referencias bibliográficas.....	43
	Anexo 1. Descripción de Casos de Usos del Sistema.....	47

Índice de tablas

Tabla 2.1. Actores del sistema.....	20
Tabla 2.2. Requisitos funcionales del sistema.....	22
Tabla 2.3. Casos de usos del sistema.....	25
Tabla 2.4 .CU Autenticar usuario	30
Tabla 2.5 CU Gestionar Puntos	30
Tabla 3.1 Diseño de casos de prueba gestionar usuario.....	35
Tabla 3.2 Diseño de caso de prueba autenticar usuario	37
Tabla 3.3. Diseño de caso de prueba definir nombre equipo.	38
Tabla 3.4 No conformidades	39
Tabla 2.6 CU Gestionar usuario.....	47
Tabla 2.7 CU Modo Visualización	49
Tabla 2.8 CU Definir nombres de equipos	50
Tabla 2.9 CU Gestionar Información	51

Índice de figuras

Fig 1.1. Gráfico de barras vertical.	10
Fig 1.2. Gráfico de pastel.	10
Fig 1.3. Histograma.....	11
Figura 1.4. Gráfico de líneas.	11
Figura 1.5. Gráfico de cajas.	12
Fig 2.1 Diagrama del dominio	21
Fig 2.2 Diagrama de casos de uso del sistema.....	26
Fig 2.3 Diagrama de despliegue	27
Fig 2.4 Diagrama de paquetes	28
Fig 3.1 No conformidades	40

Introducción

En la actualidad y debido al desarrollo constante de la industria y los procesos que se llevan a cabo, las empresas cuentan con grandes volúmenes de información que son vitales para el correcto funcionamiento de la misma y así lograr de manera eficiente sus objetivos. Pero a su vez si esta información no es manejada de manera rápida, segura y eficiente pueden ocurrir grandes fallos en el cumplimiento de los objetivos y de la estrategia empresarial que se lleva. Por eso muchas empresas cuentan con herramientas informáticas que le permiten controlar y manejar de manera correcta toda la información referente a la misma. Pero no solo una herramienta informática soluciona dicho problema, hace falta un conocimiento previo y aplicar técnicas y herramientas empresariales para lograr cumplir y mantener siempre el enfoque previsto. Un Cuadro de Mando Integral (CMI) es una de estas herramientas que son de gran ayuda en dicho contexto, y aplicable hoy en día en muchas industrias de diferentes ramas y con distintos objetivos. Se denomina Cuadro de Mando Integral a una herramienta de gestión que facilita la toma de decisiones y que recoge un conjunto coherente de indicadores que proporciona a la alta dirección y a las funciones responsables, una visión comprensible del negocio o de su área de responsabilidad. Visto así el CMI ofrece numerosas posibilidades y ventajas para el proceso dentro de la empresa, de las cuales se pueden mencionar la posibilidad de llevar un control del cumplimiento de los objetivos y de la estrategia empresarial que es aplicada, la generación de información referente a los procesos internos de la empresa para tomar decisiones que mejoren el proceso industrial y no afecten el cumplimiento de los objetivos propuestos. (Robert S.Kaplan)

Pero no solo es aplicar el concepto de Cuadro de Mando Integral y esperar resultados inmediatos, sino que es necesario emplear otros conceptos relevantes en el mundo moderno que son eficientes en el manejo de información para lograr un proceso industrial óptimo desde todos los parámetros posibles. La Inteligencia de Negocio o (*Business Intelligence*, BI) por sus siglas en inglés, es también fundamental dentro de cualquier entidad y el manejo correcto del término antes planteado y su aplicación posibilita numerosos resultados positivos en términos empresariales. El *Data Warehouse Institute*, define la BI como la combinación de tecnología, herramientas y procesos que me permiten transformar datos almacenados en información, esta información en conocimiento y este conocimiento dirigido a un plan o una estrategia comercial. De esta manera se utiliza la información interna de la entidad, a partir de un análisis se logra obtener un conocimiento sobre dicha información que luego es explotado para lograr beneficios en términos económicos o de procesos. (Institute)

Dentro de la Inteligencia de Negocios la representación gráfica de información es fundamental, mediante la cual la toma de decisiones o análisis para posterior empleo es realmente sencillo puesto que las gráficas y diferentes modelos de representar información tienen una manera sencilla de agrupar información de cualquier tipo y brindan una vista amigable y fácil de comprender. En la actualidad son numerosas las empresas que emplean herramientas capaces de representar gráficamente la información relacionada con dicha entidad y así la toma de decisiones por parte de los expertos en analizarla es mucho más sencilla.

La Universidad de las Ciencias Informáticas, centro universitario creado por la Revolución para formar ingenieros informáticos, cuenta con varios centros de producción de software, específicamente el Centro de Informática Industrial (CEDIN), el cual propone soluciones que benefician al sector industrial dentro y fuera del país, donde ninguno de sus productos cuenta con una herramienta que pueda vincularse en la toma de decisiones de una determinada industria, pues casi todos son software del tipo sistemas de procesamiento transaccional en línea (OLTP). El OLTP es una clase de programa que facilita y gestiona aplicaciones orientadas a transacciones, típicamente para ingreso de datos y retención de transacciones en varias industrias.

Entonces se puede hacer la siguiente interrogante ¿cómo brindar la información requerida a los expertos de nivel gerencial, para realizar la toma de decisiones en las industrias?

Una vez planteada la problemática se define como **objetivo general**: Desarrollar un módulo de visualización del Cuadro de Mando Integral para el SCADA SAINUX.

Como **objeto de estudio** se determina: Herramientas que apoyan la toma de decisiones en las industrias.

Como **campo de acción**: Representación de información para apoyar la toma de decisiones en las industrias

Una vez definidas las pautas anteriormente mencionadas se definen tareas para llevar a cabo la investigación con el objetivo de satisfacer los objetivos generales.

Dicho así se especifican un grupo de tareas:

1. Investigación del estado-del-arte sobre sistemas de apoyo a la toma de decisiones a las industrias.
2. Selección de las tecnologías y ambiente de desarrollo.

3. Revisión de la documentación pertinente tratando los conceptos principales abordados
4. Selección de la tecnología específica con la cual se trabajará.
5. Realización de diagramas pertinentes en correspondencia de la metodología de desarrollo de software seleccionada.
6. Identificación de la arquitectura del sistema.
7. Implementación de la herramienta.
8. Realización de pruebas pertinentes a la herramienta obtenida.

Entre los Métodos de trabajo científico utilizados para darle cumplimiento a estas tareas serán empleados:

Métodos teóricos

Histórico-lógico: este método permite realizar una investigación que da inicio con las formas de representación de la información y las tendencias que actualmente existen.

Analítico-sintético: para seleccionar los elementos de mayor importancia, descomponer el problema de la investigación con el objetivo de comprender el área de acción.

Modelación: mediante este método se representó toda la información obtenida hasta el momento mediante diagramas, ya sea de clases o de diseño, los cuales permiten reflejar las relaciones y cualidades de la aplicación que se desea desarrollar.

Método empírico

Observación: mediante este método se realizaron observaciones a los diferentes sistemas que existen en la Universidad de Ciencias Informáticas con el fin de ver similitudes en ellos que proporcionen una perspectiva del empleo de la representación gráfica.

El contenido de este trabajo se encuentra enmarcado en tres capítulos.

Capítulo 1: Fundamentación Teórica.

En este capítulo se realizará una descripción conceptual de las terminologías y/o técnicas que sustentan y dan inicio a la investigación. Además de mostrar una introducción hacia los sistemas de gestión de la formación y la investigación en varios centros.

Capítulo 2: Análisis y Diseño.

En este capítulo se presentan las características de la aplicación, las herramientas para su confección, así como sus funcionalidades y particularidades.

Capítulo 3: Implementación y Prueba.

En este capítulo se describe la estructura del código implementado además de la presentación de los casos de prueba identificados para el sistema y los resultados de las mismas.

Capítulo 1. Fundamentación teórica

En el presente capítulo se hace alusión a las características generales de una aplicación capaz de generar Inteligencia de Negocio y un Cuadro de Mando Integral. Se describen los conceptos fundamentales asociados al dominio del problema y el objeto de estudio, haciéndose un análisis de la situación actual. Se analizan las soluciones semejantes a la propuesta de solución, así como la fundamentación de las tecnologías que se utilizarán para el desarrollo de la aplicación, las herramientas, lenguajes de programación para la implementación del sistema y las propuestas para su desarrollo.

1.1 Inteligencia de negocio

En la actualidad es cada vez más difícil desarrollar herramientas que brinden una ayuda a nivel gerencial o de industria para proveer un mejor rendimiento o calidad en sus servicios y de esta manera crecer financieramente. La inteligencia de negocio juega un papel protagónico en lo antes mencionado.

Se denomina inteligencia empresarial ,inteligencia de negocios o BI (*Business Intelligence*), al conjunto de estrategias y aspectos relevantes enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa . (Cano, 2007)

Es posible diferenciar datos, informaciones, y conocimientos, conceptos en los que se centra la inteligencia empresarial, un dato es algo vago, por ejemplo "500", la información es algo más precisa, ejemplo "Daniel gana 500 pesos de salario", y el conocimiento se obtiene mediante el análisis de la información, ejemplo "Daniel gana 500 pesos de salario, Daniel tiene el mejor salario". Aquí es donde la BI entra en juego, ya que al obtener conocimiento del negocio una vez capturada la información de todas las áreas en la empresa, es posible establecer estrategias y determinar cuáles son las fortalezas y las debilidades. Entre las características principales de los sistemas de inteligencia de negocio se encuentra el empleo de elementos gráficos e interfaces de usuario especializadas que permitan la visualización de los análisis de datos desde perspectivas donde los expertos puedan establecer relaciones y patrones entre ellos, accesibilidad a la información, los datos son la fuente principal de este concepto, apoyo en la toma de decisiones. BI también se refiere al uso de la tecnología para recolectar y usar efectivamente la información, a fin de mejorar la operación del negocio. Un sistema ideal de BI

ofrece a los empleados, socios y altos ejecutivos acceso a la información clave que necesitan para realizar sus tareas del día a día, y principalmente para poder tomar decisiones basadas en datos correctos y certeros. Desde sus raíces en los llamados Sistemas de Información Ejecutiva (EIS, por sus siglas en inglés, *Executive Information Systems*) y los Sistemas para la toma de decisiones (DSS, por sus siglas en inglés *Decision Support Systems*), BI ha evolucionado y se ha transformado en todo un conjunto de tecnologías capaces de satisfacer a toda una gama de usuarios junto a sus necesidades en cuanto al análisis de información.

¿Pero cómo hacer que los negocios pueden crear inteligencia de sus datos?, Además como proveer oportunamente y acertadamente acceso a esa información para sus usuarios finales. Este proceso es dinámico e iterativo. El proceso empieza con preguntas, y las respuestas son resultado de más preguntas o de subsecuentes interacciones del proceso.

Fase 1 – Dirigir y Planear:

Esta fase es el principio y el fin del proceso. Es el principio porque involucra redactar los requerimientos específicos, y el final porque contesta preguntas que guían a otras nuevas. El proceso de BI empieza con los usuarios (Ejecutivos, Directivos, Líderes de Negocio etc.) y aquí se generan las preguntas que les va ayudar a ellos a alcanzar sus objetivos. Ejemplos de esas preguntas son: ¿Cuáles son los clientes más rentables? ¿Cuál es el margen de cada línea de producto?, Etc., Estas necesidades son presentadas por diversos analistas. Esto quiere decir que estos analistas de negocios formulan los requerimientos de los usuarios y dirigen un plan para la recolección de la información y para solución de las respuestas.

Fase 2 – Recolección de Información:

Hay diversas fuentes de información dentro de una compañía. La automatización de los procesos ha creado una fuente de recursos como son: Puntos de ventas, Aplicaciones de Servicios al cliente etc. Los diferentes sistemas crean, procesan y almacenan diferentes tipos de información. Este es un proceso continuo y es importante entender que los datos de esas fuentes son simplemente información y no Inteligencia. Los datos en crudo frecuentemente son incompletos y confusos. La información se convierte en Inteligente a través de procesarla y de analizarla. El proceso de recolección de información es cuando las diferentes fuentes son analizadas para determinar los datos necesarios para encontrar las respuestas a las preguntas.

Fase 3 – Procesamientos de Datos:

Esta fase es la integración de datos en crudo a un formato utilizable para el análisis. Esto puede ser posible, creando una nueva base de datos, agregar datos a bases de datos existente o consolidando información. Esta fase generalmente es vista como Extracción, Transformación y Carga que ocurre en los ambientes de BI.

Fase 4- Análisis y Producción:

El grupo de análisis de negocios utiliza herramientas y técnicas para ordenar sobre los datos y crear inteligencia. El resultado final es la producción de respuestas “inteligentes”, en un contexto propio. En algunos casos es un proceso simple como la creación de un reporte. En otros casos, son la creación de indicadores. Tal vez en esta fase, sean generados requerimientos adicionales pues los analistas puede que encuentren nuevas preguntas que necesiten ser contestadas.

Fase 5- Difusión:

Esta fase de difusión, es entregar productos inteligentes a los diversos clientes que lo requieren. Esto básicamente implica el uso de herramientas BI para la publicación de “tableros de indicadores”, reportes o la posibilidad de tener herramientas de fácil uso para que los mismos usuarios tengan la capacidad de revisar los datos de manera rápida y sencilla.

Es importante entender que las herramientas de soporte a la toma de decisiones, son eso, herramientas, y que la selección y uso, simplifican muchas operaciones y procesos en el negocio, pero que los tomadores de decisiones son la piedra angular.

1.2 Cuadro de mando integral

Previamente se planteaba el término Cuadro de Mando Integral, los Cuadros de Mando (CM) son herramientas de control empresarial orientadas a la monitorización de los objetivos de la empresa o de las diferentes áreas de negocio a través de indicadores. (Robert S.Kaplan) En función de la naturaleza de los indicadores se habla de Cuadro de Mandos Estratégico (CME) si se trata de indicadores estratégicos u Cuadro de mandos Operativo (CMO) si los indicadores son operativos, es decir, indicadores rutinarios ligados a áreas o departamentos específicos de la empresa (las áreas suelen ser procesos).

La periodicidad de los CMO puede ser diaria, semanal o mensual y además está focalizado en indicadores que generalmente representan procesos, por lo que su puesta en funcionamiento es más barata y sencilla y suele ser un buen punto de partida para aquellas compañías que intentan evaluar la implantación de un cuadro de mando integral. El CMO en línea es fundamental en momentos críticos. Es posible aplicar este concepto de CMI a aplicaciones web, de escritorio e incluso nativas, visto de una manera general, el concepto y los puntos que se deben definir en cada tipo de plataforma varían según el propósito y objetivo buscado.

Existen diversas formas de aplicar un CMI, todo depende de los objetivos de una industria determinada, así mismo algunas lo emplean en más o menos pasos o fases. Se describirá las etapas de un CMI aplicado a una empresa x:

- ETAPA I. EMPRESA, NEGOCIO, FÓRMULA DE ÉXITO.
- ETAPA II. LOS GERENTES Y LOS CONCEPTOS DE DIRECCIÓN.
- ETAPA III. EL TABLERO DE COMANDO PARA LA ALTA DIRECCIÓN.
- ETAPA IV. OBJETIVOS, METAS, INDICADORES Y RESPONSABLES POR NIVEL. LISTAS DE INDICADORES.
- ETAPA V. REMUNERACIÓN VARIABLE - TRANSFORMACIÓN EMPRESARIAL.
- ETAPA VI. FORMULA DE CÁLCULO.
- ETAPA VII. BENCHMARKING, COMPARACIONES, REPRESENTACIÓN GRAFICA.
- ETAPA VIII – SISTEMA DE INFORMACIÓN EJECUTIVA (SIE Ó EIS).
- ETAPA IX. COMUNICACIÓN Y CAPACITACIÓN.
- ETAPA X. INTEGRACIÓN A TODAS LAS FASES DE LA GESTIÓN EMPRESARIAL.

Tanto la BI como los CMI tienden a relacionarse a la hora de aplicarlos en una industria debido a las funciones que cumplen. También la forma de representación de una cierta información o datos de una determinada industria tanto en términos monetarios o de productos o simplemente alguna información que se desee representar debe tener un significado, y mucho tiene que ver en esto la forma en la que esta se representa. Sin duda alguna la representación gráfica es sumamente importante y simple de entender, mediante la cual se pueden presentar para su comprensión grandes cantidades de información de forma reducida y fácilmente asequibles para los sujetos a los que va dirigida.

1.3 Formas de representación de la información

Sin lugar a duda la manera en que es representado una información determinada o un juego de datos influye en la interpretación por los receptores o expertos que la asimilan. Los gráficos estadísticos transmiten esa información de modo más expresivo, nos van a permitir, con un sólo golpe de vista, entender de qué se habla, observar sus características más importantes, incluso sacar alguna conclusión sobre el comportamiento de la muestra donde se está realizando el estudio. El principal problema que tiene la representación de datos es su objetividad y comprensión. El proceso de la información requiere tomar decisiones sobre que ejes se tendrán en cuenta, periodo a mostrar, comparar o no. Estas decisiones pueden hacer que la representación ofrecida no sea la que la audiencia espera. Sin embargo, que el gráfico sea fácilmente comprensible requerirá una cierto "ensayo y error" hasta encontrar el modelo perfecto.

El disponer de la capacidad de proceso junto a un tiempo de "ensayo y error" harán que la información sea más fácil de entender y por tanto la toma de decisiones se hará con mayor seguridad y más rápido. Quizás parezca un poco complicado y caro, pero la representación de datos se puede hacer desde un presupuesto "cero" empleando disímiles herramientas. La herramienta que se desea obtener permitirá la visualización de información en los siguientes gráficos:

Diagrama de dunas o anillos:

También conocido como gráfico de sectores, es una representación circular de las frecuencias relativas de una variable cualitativa o discreta que permite ver la distribución interna de los datos que representan un hecho en forma de porcentajes sobre un total. Consiste en dividir un círculo en secciones cuyas superficies sean proporcionales a las cantidades correspondientes a cada categoría. Es recomendable utilizarlo cuando la cantidad de categorías a representar es pequeña. (Abad, 2009)

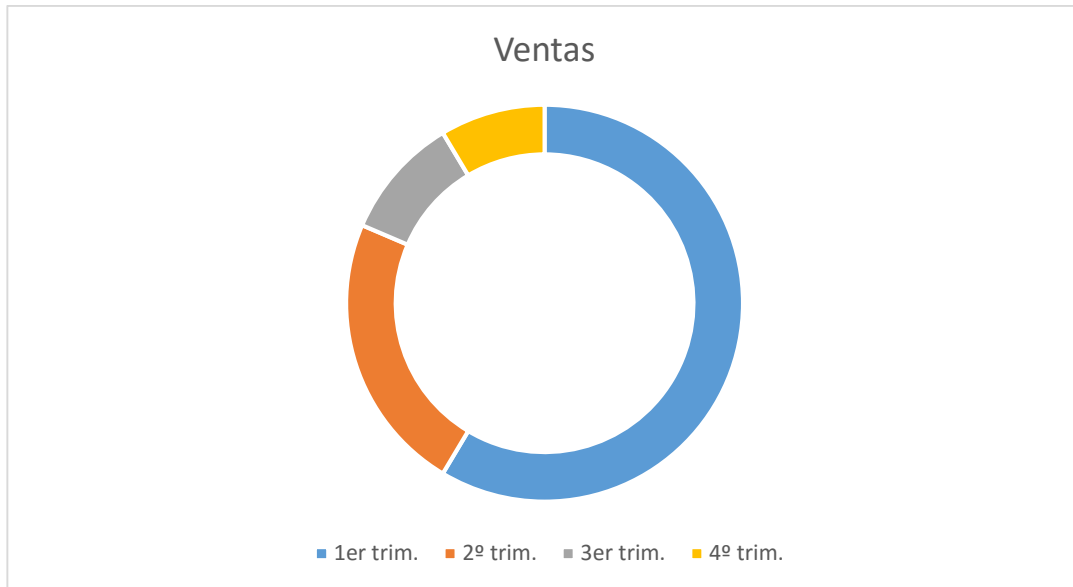


Fig 1.1. Gráfico de dunas.

Gráfico de Pastel:

Aquí se puede ver la contribución de cada parte a un total. Este gráfico se puede utilizar de forma creativa comparando el tamaño de las tartas y el contenido de las mismas. (Desarrolloweb, s.f.) Por ejemplo, si queremos ver el ingreso en un año del capital de una empresa determinada, comparando los ingresos por los productos que esta ofrece, podríamos emplear este tipo de gráfica y así conocer cuál fue el producto más vendido.

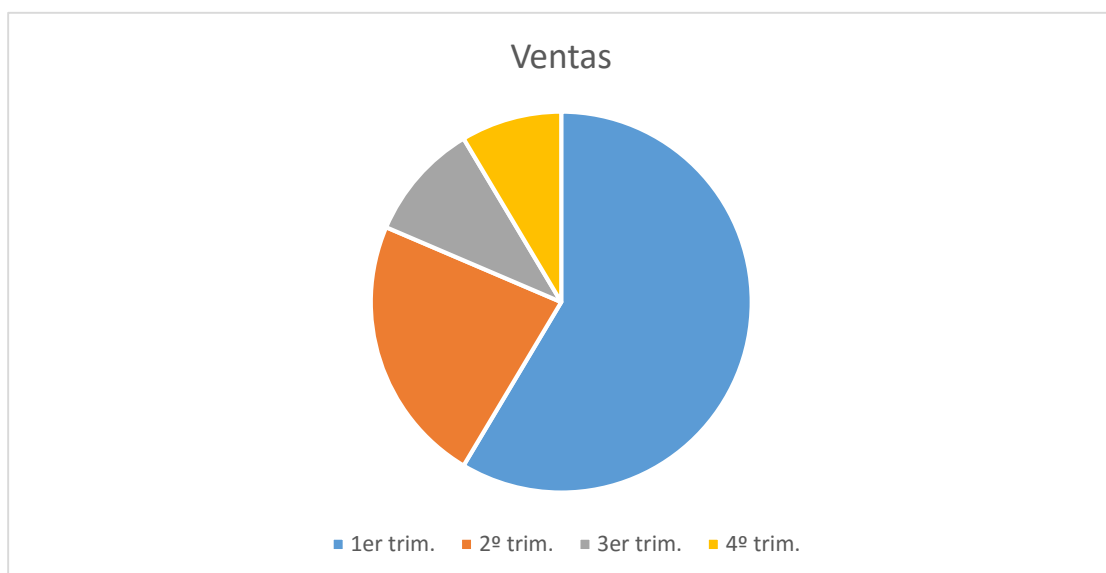


Fig 1.2. Gráfico de pastel.

Histograma.

Un histograma es una representación gráfica de una variable en forma de barras. En el eje abscisas se construyen unos rectángulos que tienen por base la amplitud del intervalo, y por altura, la frecuencia absoluta de cada intervalo. (Vitutor, s.f.) Si se desea visualizar el comportamiento de las alarmas generadas en un periodo de tiempo, el histograma sería el más conveniente para representar dicha información.

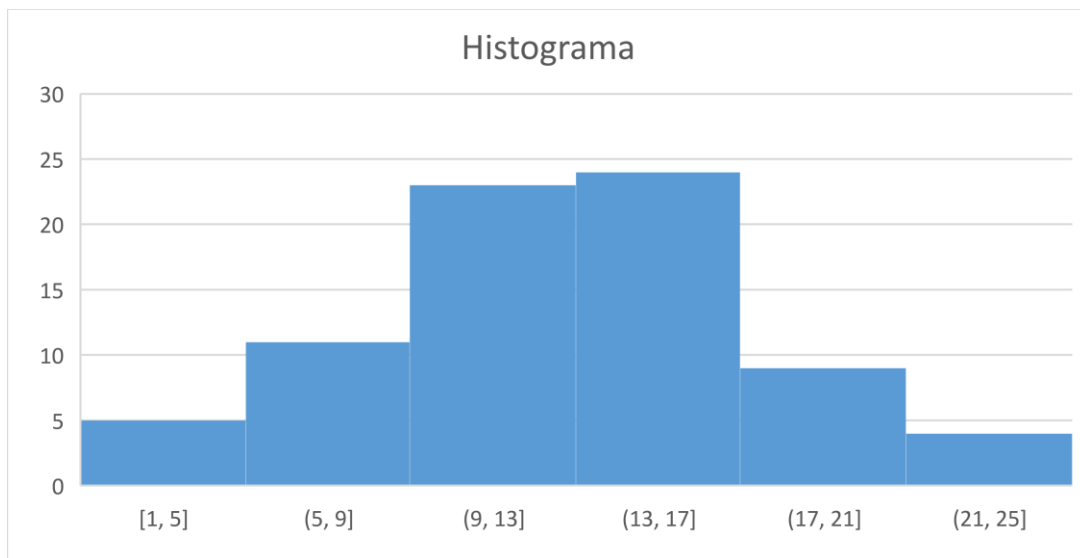


Fig 1.3. Histograma.

Gráfico de líneas.

Los gráficos de líneas muestran los datos en forma de puntos y todos los puntos de la misma serie se unen mediante una línea, de ahí su nombre. Cada valor aparece representado por un punto que es la intersección entre los datos del eje horizontal y los del eje vertical. (Ministerio de Educacion de España, s.f.)

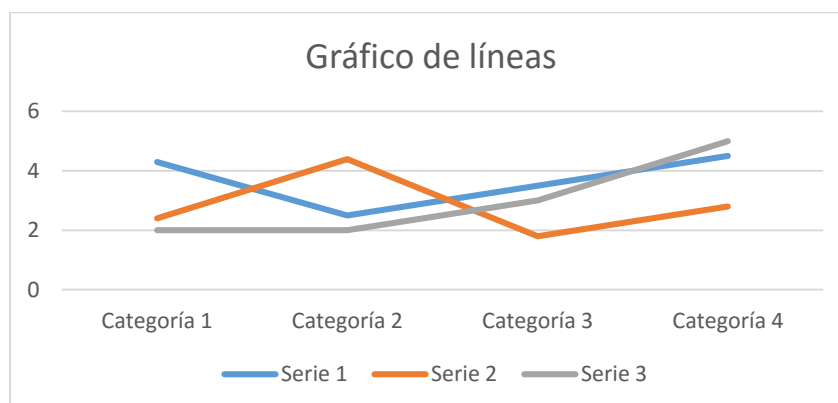


Figura 1.4. Gráfico de líneas.

Gráfico de Cajas

Una gráfica de este tipo consiste en una caja rectangular, donde los lados más largos muestran el recorrido inter-cuartílico. Este rectángulo está dividido por un segmento vertical que indica donde se posiciona la mediana y por lo tanto su relación con los cuartiles primero y tercero. Esta caja se ubica a escala sobre un segmento que tiene como extremos los valores mínimo y máximo de la variable. Las líneas que sobresalen de la caja se llaman bigotes. Estos bigotes tienen un límite de prolongación, de modo que cualquier dato o caso que no se encuentre dentro de este rango es marcado e identificado individualmente. (Estadística para todos, s.f.)

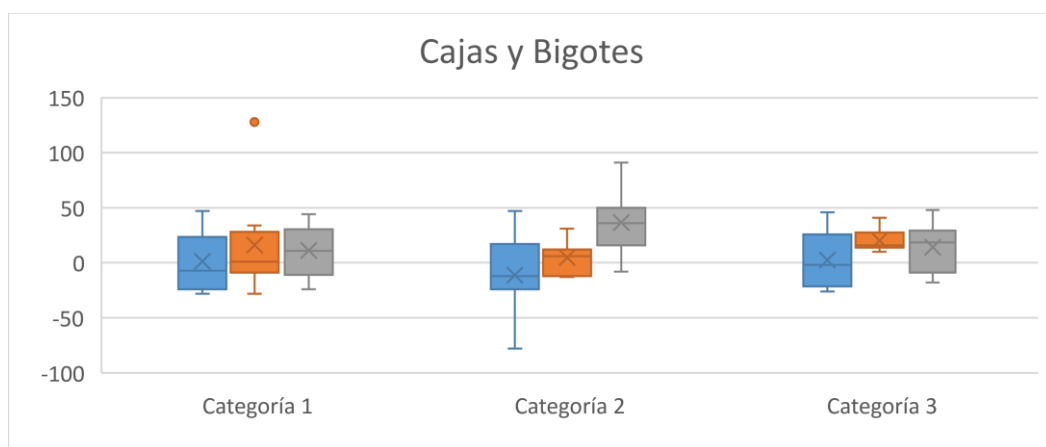


Figura 1.5. Gráfico de cajas y bigotes.

1.4 Plataformas de software

En la actualidad es posible desarrollar una aplicación o software en varias plataformas, dependiendo de la funcionalidad y el objetivo que se persigue. Dígase aplicaciones web, de escritorio o nativas (móviles). Cada una de estas plataformas presentan ventajas las cuales son tenidas en cuenta a la hora de seleccionar en cual de todas se desarrollará la aplicación.

Las ventajas de utilizar aplicaciones web tanto a nivel de usuario como de industrias son notables, ejemplo:

- Actualización. Las aplicaciones basadas en web están siempre actualizadas con el último lanzamiento.
- Inmediatez de acceso. Las aplicaciones basadas en web no necesitan ser descargadas, instaladas y configuradas. Usted accede a su cuenta online a trabajar sin importar cuál es su configuración o su hardware.

- Menos requerimientos de memoria. Las aplicaciones basadas en web tienen muchas más razonables demandas de memoria RAM de parte del usuario final que los programas instalados localmente.
- Múltiples usuarios concurrentes. Las aplicaciones basadas en web pueden realmente ser utilizada por múltiples usuarios al mismo tiempo.

Por otra parte, la tecnología móvil, o aplicaciones nativas utilizan de la misma manera una cantidad de herramientas para generar sus negocios. Ejemplo el sistema IOs de Apple utiliza Cocoa y Objective C para desarrollar sus aplicaciones móviles, dígame también todas las funcionalidades que brindan los mismos, desde aplicaciones con BI para industrias o determinado negocio. Por otra parte, el sistema operativo Android utiliza fuentes de C, C++ y Java, teniendo en cuenta el alcance de los mencionados lenguajes de programación son infinitas las utilidades y ventajas de los mismos, aplicados al mundo de la telefonía móvil.

Dentro del mundo de las aplicaciones móviles también se encuentran ventajas como ejemplo:

- Fortalecer la marca es una de los principales beneficios de desarrollar aplicaciones móviles propias para las empresas.
- Tener presencia en los dispositivos móviles posiciona mucho mejor la marca, dado que los clientes y usuarios pueden consultar la aplicación en cualquier parte.
- La posibilidad de sincronización de la aplicación con las redes sociales mejora la difusión de contenidos.
- Comunicación fluida y sin limitaciones, con la aplicación móvil el cliente tiene acceso a contactar con la empresa en cualquier momento que lo necesite, sin tener que esperar a llamar por teléfono durante los horarios de atención y libre de costo.

En el otro lado están las aplicaciones de escritorio, para las cuales existen muchas alternativas, cada una de ellas con sus propias ventajas, desventajas y curva de aprendizaje, sólo es cuestión de conocerlas y elegir la que sea más adecuada a las necesidades. Ejemplo es Microsoft Visual Studio.NET que no es más que un entorno de desarrollo integrado (IDE, en inglés) desarrollado para crear aplicaciones para el sistema operativo Windows. Este entorno es capaz de ofrecer soporte a múltiples lenguajes de programación como Visual C++, Visual C#, Visual J#, y Visual Basic .NET. Del mismo modo, ofrece soporte a entornos de desarrollo web como ASP.NET.

Entre las principales ventajas de emplear este tipo de aplicaciones se encuentra:

- Pueden ser más robustas
- Tiempo de respuesta más rápido
- Se puede hacer cualquier cosa que permita el Sistema Operativo (cuestión gráfica, control total de las entradas del usuario al momento de capturar)
- Facilita el uso de teclas en caliente

Vistas las principales tecnologías tanto en plataformas Web, aplicaciones de escritorio y en nativas, así como las principales ventajas y desventajas de las mismas se decide realizar la herramienta sobre la plataforma Web. La herramienta que se desea desarrollar debe ofrecer una rápida respuesta al usuario, la misma permitirá seleccionar entre las funcionalidades que brinda y mostrar el resultado de forma gráfica o en forma de texto dependiendo su contenido. Para ello dicha herramienta permitirá la conexión con un servidor que internamente realizará las operaciones que le son especificadas. Así el usuario luego de autenticarse podrá seleccionar la funcionalidad deseada dependiendo del objetivo buscado, internamente la herramienta enviará la solicitud al servidor el cual gestionará la información con el módulo de base de datos, posteriormente se aplica la minería de datos para finalmente ser mostrado el resultado en la aplicación web que se desarrollará.

1.5 Aplicación web para generar inteligencia de negocio

Una aplicación web es aquella que consiste en un documento electrónico que contiene información, cuyo formato se adapta para estar insertado en la World Wide Web, de manera que los usuarios a nivel mundial puedan entrar a la misma por medio del uso de un navegador, visualizándola con un dispositivo móvil como un Smartphone o un monitor de computadora. En otras palabras, es un software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador.

1.5.1 Estructura de una aplicación web.

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador web ofrece la primera capa y un motor capaz de usar alguna tecnología web dinámica (ejemplo: PHP, Java Servlets o ASP, ASP.NET) constituye la capa del medio. Por último, una base de datos constituye la tercera y última capa. El navegador web manda peticiones a la capa del medio, que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario. (Rendimientodesistemas, s.f.)

1.5.2 Características de las aplicaciones web

Cuando se plantea la creación y el diseño de un nuevo sitio web, las opciones que se plantean son básicamente dos: las webs dinámicas y las webs estáticas. La elección varía en función de las necesidades de la empresa o cliente. Las páginas webs dinámicas son aquellas que cuentan con una compleja programación y utiliza bases de datos que cargan la información que el usuario ve en pantalla cuando visita la página. La principal diferencia entre éstas y las estáticas radica en que los contenidos se actualizan según el usuario visita periódicamente el sitio. Las páginas webs estáticas suelen estar desarrolladas en HTML y CSS que ofrecen menor contenido que las dinámicas. Por ello, se suelen utilizar para quienes únicamente publicar contenidos que no necesitan ser modificados ni actualizados en el tiempo, únicamente ofrecen contenidos fijos. (Aplicacionesweb, s.f.) A continuación, se exponen diferentes características entre estos dos tipos de páginas web:

Web estática:

- El proceso de actualización es lento y manual.
- No ofrece bases de datos.
- El usuario no puede seleccionar, ordenar o modificar los contenidos o el diseño de la página.
- Su diseño y desarrollo es más rápido y económico que el de una web dinámica.

Web dinámica:

- Muchas posibilidades en diseño y desarrollo.
- Utiliza varios lenguajes y técnicas de programación en su desarrollo.
- El proceso de actualización es muy sencillo y no requiere entrar en el servidor.
- Permite un gran número de funcionalidades tales como bases de datos, foros, contenido dinámico.
- El usuario puede alterar el diseño, contenidos o presentación de la página a su gusto.

1.5.3 Tecnologías para el desarrollo de aplicaciones web.

Para lograr una herramienta capaz de visualizar información mediante el empleo de gráficas sin importar la plataforma seleccionada ni datos especificados se necesita de una tecnología consecuente a los requerimientos.

NodeJS

Node.js es una librería y entorno de ejecución de entrada y salida dirigida por eventos y por lo tanto asíncrona que se ejecuta sobre el intérprete de JavaScript, proporciona un entorno de ejecución del lado del servidor que compila y ejecuta JavaScript. (Netconsulting, s.f.)

Websocket

WebSocket es un protocolo estándar que permite la comunicación entre un navegador web o aplicación cliente y una aplicación de servidor web mediante una conexión bidireccional, utiliza una secuencia de solicitud-respuesta HTTP estándar para establecer una conexión, también proporciona una interfaz para el cierre asíncrono de la conexión desde cualquier lado. (IBM, s.f.)

JSon

JSON (JavaScript Object Notation) es un formato para el intercambio de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. JSON nació como una alternativa a XML, el fácil uso en JavaScript ha generado un gran número de seguidores de esta alternativa. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías. (Geekytheory, s.f.)

JavaScript

Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache. Es un lenguaje script multi-paradigma, basado en prototipo, dinámico y soporta estilos de programación funcional orientada a objetos e imperativa (Developer.mozilla, s.f.).

Visual Paradigm

Es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos (software.com.ar, s.f.).

1.6 Metodología de desarrollo de software

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) en inglés es una versión simplificada del *Rational Unified Process* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (*test driven development* - TDD en inglés)
- Modelado ágil
- Gestión de Cambios ágil
- Refactorización de Base de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva:

1. Inicio: El objetivo de esta fase es obtener una comprensión común cliente-equipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.

2. Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.

3. Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.

4. Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Al no existir una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI. Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas.

Conclusiones parciales

En el capítulo terminado se abordaron sobre los principales conceptos que se emplearan a lo largo del presente documento, también las diferentes tecnologías existentes en cada una de las diferentes plataformas con más utilidad, así como sus ventajas. Se determinó el tipo de aplicación que se desarrollara y la tecnología que se empleara en el desarrollo de la misma.

Capítulo 2. Análisis y diseño

En el presente capítulo se muestran las particularidades que el sistema debe cumplir. Se muestra una propuesta del sistema y finalmente se realiza una descripción del modelo de dominio, el cual facilita el paso a los requerimientos del sistema, permitiendo el conocimiento de forma general del mismo y la identificación de los actores que interactúan con el sistema y la secuencia de acciones con las que se relacionan.

2.1 Propuesta de sistema

El sistema contará con varias funcionalidades que permitirán la gestión de información tanto de los usuarios como de los datos sobre los cuales se trabajar. El usuario luego de autenticarse accederá al sistema y podrá realizar las funciones pertinentes de acuerdo a su rol y permisos. Las funcionalidades del sistema se exponen a continuación:

1. Gestión de usuarios de la gerencia.

Los usuarios que trabajaran con el sistema pertenecen a la gerencia de la industria y se definirán con los datos siguientes:

- Nombre y Apellidos (obligatorio)
- Rol o Cargo
- Usuario (obligatorio y único)
- Contraseña (obligatorio)

2. Gestión y configuración del equipamiento

Los equipos estarán definidos a partir de datos que se almacenan en la Base de Datos (BD) de configuración, y a partir de ello el sistema realiza una serie de tareas

2.1- Carga de los puntos almacenados en la Base de Datos de Configuración. El resultado ya viene con una propuesta de cuales puntos pertenecen al mismo equipo.

2.2- Modificar nombre de los Equipo (grupos de puntos) y dar la posibilidad de mover los puntos de un grupo hacia otro.

2.3 Almacenar la información de los equipos modificados.

3. Gestión de la visualización de la información.

Se debe dar la posibilidad de que el usuario elija que información desea visualizar, usando un tipo de visualización deseada (gráficos según tipo)

4. Visualización del estado del equipo

- Visualizar la cantidad de alarmas existentes.
- Visualizar el último valor obtenido por cada punto que conforma el equipo.

5. Almacenamiento

Almacenar los datos del perfil del usuario, incluyendo la configuración de las gráficas que selecciono para cada información que maneje.

Así una vez desarrolladas las funcionalidades que debe cumplir el sistema, el usuario se autenticará, accederá a la página de trabajo y podrá elegir qué operación realizar dentro de la aplicación, obteniendo de esta manera el resultado esperado.

2.2 Descripción del sistema propuesto

Una vez descritas las principales funcionalidades que debe realizar el sistema es necesario establecer y describir las características fundamentales que tendrá el sistema una vez terminado.

2.2.1 Descripción de los actores

Tabla 2.1. Actores del sistema.

Actor	Objetivo
Usuario	Se autentica en el sistema para adquirir los permisos pertinentes y así acceder a las funcionalidades.
Administrador	Es el encargado de gestionar los usuarios, los roles y permisos que se conceden por cada rol existente.

2.2.2 Diagrama del dominio

Una vez obtenidos los requisitos funcionales y no funcionales, se obtiene un diagrama de dominio mediante el cual se exponen los conceptos clave del dominio del problema. También

se identifican las relaciones entre todas las entidades comprendidas en el ámbito del dominio del problema.

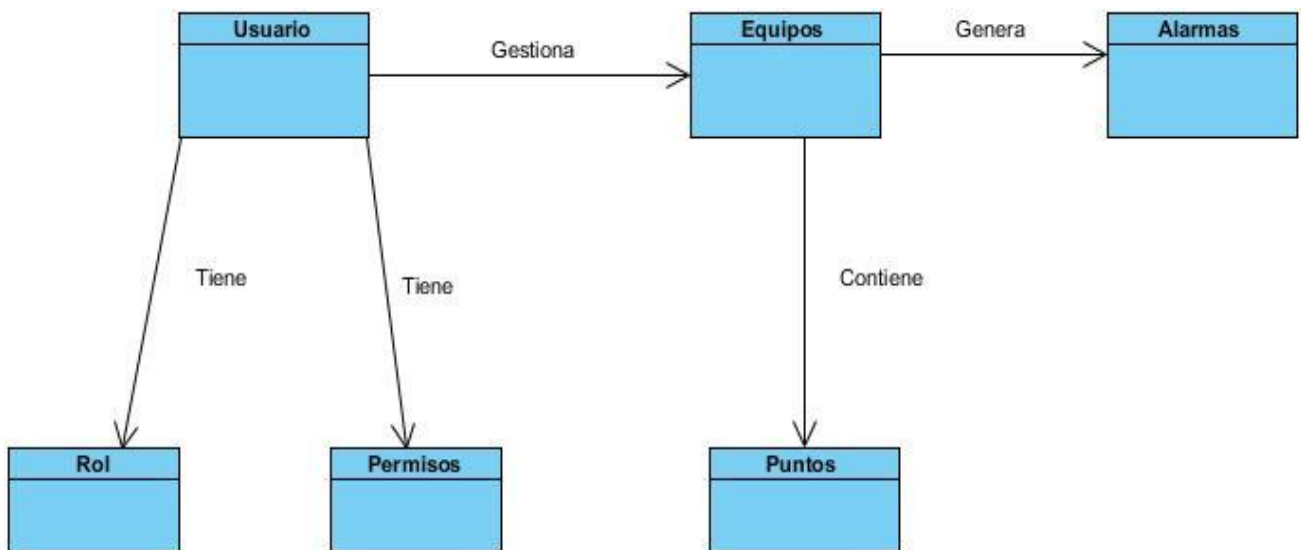


Fig 2.1 Diagrama del dominio

2.2.2.1 Descripción de las clases del diagrama del dominio

Usuario: persona que accede a la aplicación y hace función de las opciones de la misma dependiendo del objetivo deseado.

Equipos: grupo de puntos asociados.

Puntos: variable que contiene un valor y es graficado según ese valor.

Alarmas: variable que contiene una fecha de activación o generación.

Rol: cargo que tendrá los usuarios que interactúen con la aplicación.

Permiso: posibilidades que navegar por el sistema de acuerdo al rol que posea el usuario

2.3 Descripción de los requisitos de software

2.3.1 Requisitos funcionales.

Son aquellos que responden a: ¿qué debe hacer el sistema? y describen las capacidades o condiciones que se deben cumplir.

Tabla 2.2. Requisitos funcionales del sistema

RF1.Autenticar usuario
RF2.Asignar Permiso
RF3.Eliminar usuario
RF4.Crear usuario
RF5.Modificar usuario
RF6.Asignar rol
RF7.Cargar puntos
RF8.Mover puntos
RF9.Almacenar información de puntos
RF10.Visualizar ultimo valor obtenido por punto
RF11. Visualizar cantidad alarmas
RF12.Modos de visualización información
RF13.Definir nombres equipos

2.3.2 Descripción de los requisitos funcionales.

RF1.Autenticar Usuario

El usuario introduce los datos pertinentes en los campos establecidos en la interfaz de autenticación. El sistema verifica si los datos introducidos son correctos, de serlo el usuario accede al sistema, en caso contrario se muestra una notificación indicando que existen datos incorrectos.

RF2.Asignar Permiso

Todos los usuarios que interactúen con el sistema tendrán asociado un rol, mediante el cual tendrán o no permisos para realizar diversas funcionalidades en el sistema.

RF3.Elimiar Usuario

Se elimina un usuario de la Base de Datos.

RF4.Crear Usuario

Se adiciona un usuario a la Base de Datos.

RF5.Modificar Usuario.

Se modifican los datos de un usuario existente en la Base de Datos.

RF6.Asignar Rol.

Cuando se crea un nuevo usuario se le asigna un rol, de acuerdo a las responsabilidades que tendrá al interactuar con el sistema.

RF7.Cargar Puntos.

Se cargan los puntos que existen en la Base de Datos con sus respectivos valores.

RF8.Mover Puntos.

Mueve un punto previamente seleccionado, desde el Equipo inicial hacia el Equipo que se seleccione.

RF9.Almacenar Información de puntos.

Guarda los cambios realizados si se mueve un punto, desde un Equipo hacia otro.

RF10.Visualizar ultimo valor obtenido por punto.

Se visualiza gráficamente el ultimo valor obtenido por cada punto.

RF11.Visualizar cantidad alarmas.

Se visualiza gráficamente la cantidad de alarmas que existen en un período de tiempo, o se visualizan de manera seguida una debajo de la otra.

RF12.Modo de visualización información

El usuario selecciona en qué tipo de grafica visualizar la información de los Equipos que existen con sus Puntos asociados.

RF13. Definir nombres equipos

El usuario modifica el nombre que contiene el Equipo con el cual esté trabajando, almacenándose en la Base de Datos el cambio realizado.

2.3.3 Requisitos no funcionales

Los requisitos no funcionales son las cualidades o propiedades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto. Están vinculados normalmente a requisitos funcionales. La aplicación presenta los siguientes requisitos no funcionales:

Hardware:

- RnF 2. Intérprete de aplicaciones Web Mozilla Firefox 39.0.
- RnF 3. 1 GB de RAM.
- RnF 4. 20 GB de disco duro.
- RnF 5. Procesador Pentium IV o superior.

Soporte:

RnF 6. El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevos servicios en caso de ser necesarios.

Interfaz:

- RnF 7. Diseño bien estructurado y comprensible.
- RnF 8. La comunicación entre el cliente y el servidor Web será realizada a través de HTTP: (HyperText Transfer Protocol), en español Protocolo de Transferencia de Hipertexto

2.4 Patrón de arquitectura

Modelo Vista- Vista Modelo

El patrón Modelo Vista VistaModelo (MVVM) separa los datos de la aplicación, la interfaz de usuario, pero en vez de controlar manualmente los cambios en la vista o en los datos, estos se actualizan directamente cuando sucede un cambio en ellos, por ejemplo, si la vista actualiza un dato que está presentando se actualiza el modelo automáticamente y viceversa. En el patrón MVVM, puesto que la capa "ViewModel" esta débilmente acoplada con la vista (no está

referenciada a los componentes de la vista), se puede utilizar con múltiples vistas sin tener que modificarla. Por lo tanto, los diseñadores y programadores pueden trabajar en paralelo. Si la información y comportamiento no cambian, un cambio en la vista no provoca que se tenga que modificar la capa “ViewModel”. Además, como la capa “ViewModel” es un POJO (acrónimo de Plain Old Java Object), es fácil realizar test unitarios sobre ella sin ninguna configuración ni entorno especial. Lo que significa que la capa “ViewModel” tiene una mejor reusabilidad, testabilidad, y los cambios en la vista le afectan menos. (Perez, s.f.)

2.5 Casos de uso del sistema

Tabla 2.3. Casos de usos del sistema

Código	Título	Requisitos asociados
CU-1	Gestionar Usuario	RF2,RF3,RF4,RF5,RF6
CU-2	Gestionar Puntos	RF7,RF8,RF9
CU-3	Visualizar información	RF10,RF11
CU-4	Seleccionar modo visualización	RF12
CU-5	Definir nombre equipos	RF13
CU-6	Autenticar usuario	RF1

2.6 Diagrama de casos de uso del sistema

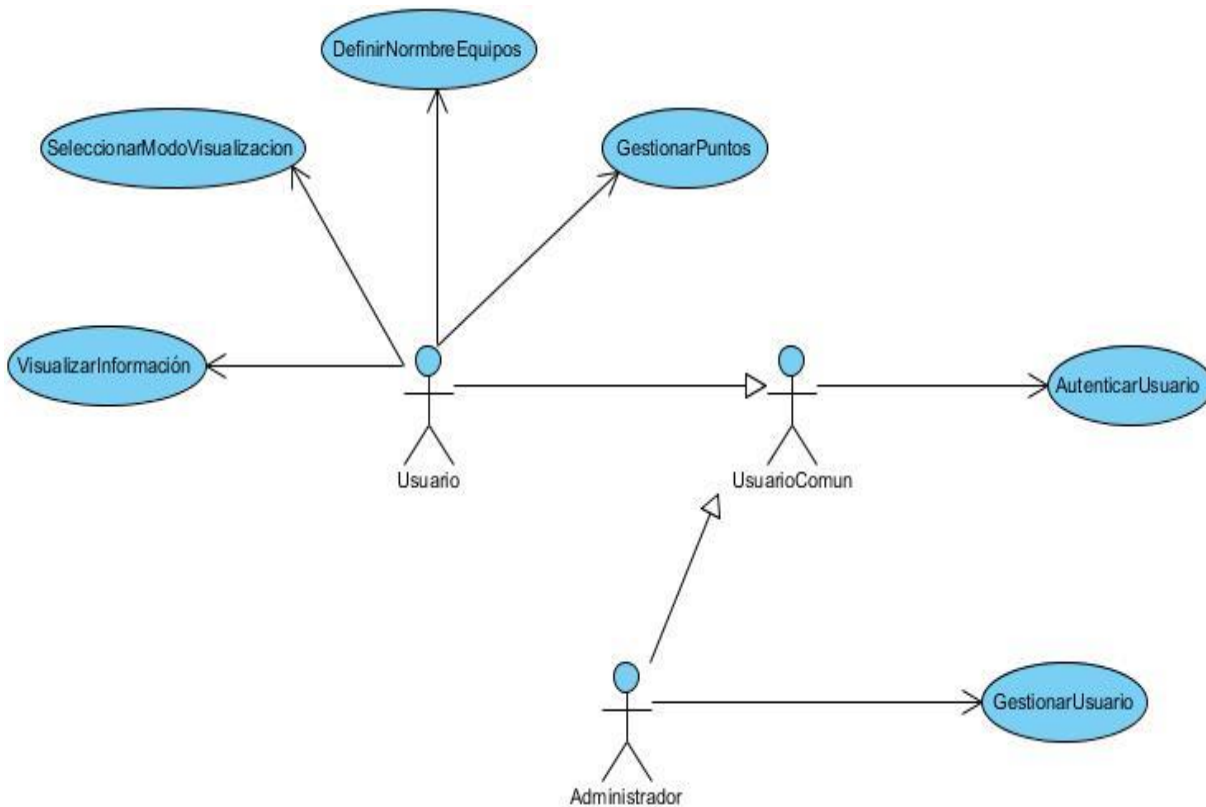


Fig 2.2 Diagrama de casos de uso del sistema

2.6.1 Patrones de casos de uso

Los patrones de casos de uso son comportamientos que deben existir en el sistema, ayudan a describir qué es lo que el sistema debe hacer, describen el uso del sistema y cómo este interactúa con los usuarios. Estos patrones son utilizados generalmente como plantillas que describen como debería ser estructurados y organizados los casos de uso. (EcuRed, 2012) Entre los patrones de caso de uso se encuentra el patrón CRUD y Múltiples Actores.

Patrón CRUD Completo: nos permite modelar las diferentes operaciones para administrar una entidad, tales como crear, leer, cambiar y eliminar o dar de baja.

Patrón CRUD Parcial: indica que en caso de que solo algunas de las cuatro operaciones sean simples se puedan agrupar en un caso de uso y dejar las otras modeladas como un caso de uso separado.

El patrón Múltiples Actores es aplicado cuando dos actores inicializan un caso de uso común para ambos. El patrón tiene dos variantes, Roles Comunes y Roles Diferentes, en el presente caso se aplica la variante Roles Diferentes, puesto que el CU Autenticar usuario es inicializado por el Administrador y el Usuario.

2.7 Diagrama de despliegue

El diagrama de despliegue se emplea en las implementaciones de sistemas, muestra la configuración, disposición física de los distintos nodos de procesamiento en tiempo real y las relaciones entre sus componentes. Describen la topología del sistema: la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. En la investigación en curso solo se desarrolla la interfaz de usuario, por lo que se enmarca el nodo PC-Cliente, aunque se expone una vista completa del despliegue una vez unidos los dos restantes nodos.



Fig 2.3 Diagrama de despliegue

2.8 Diagrama de paquetes

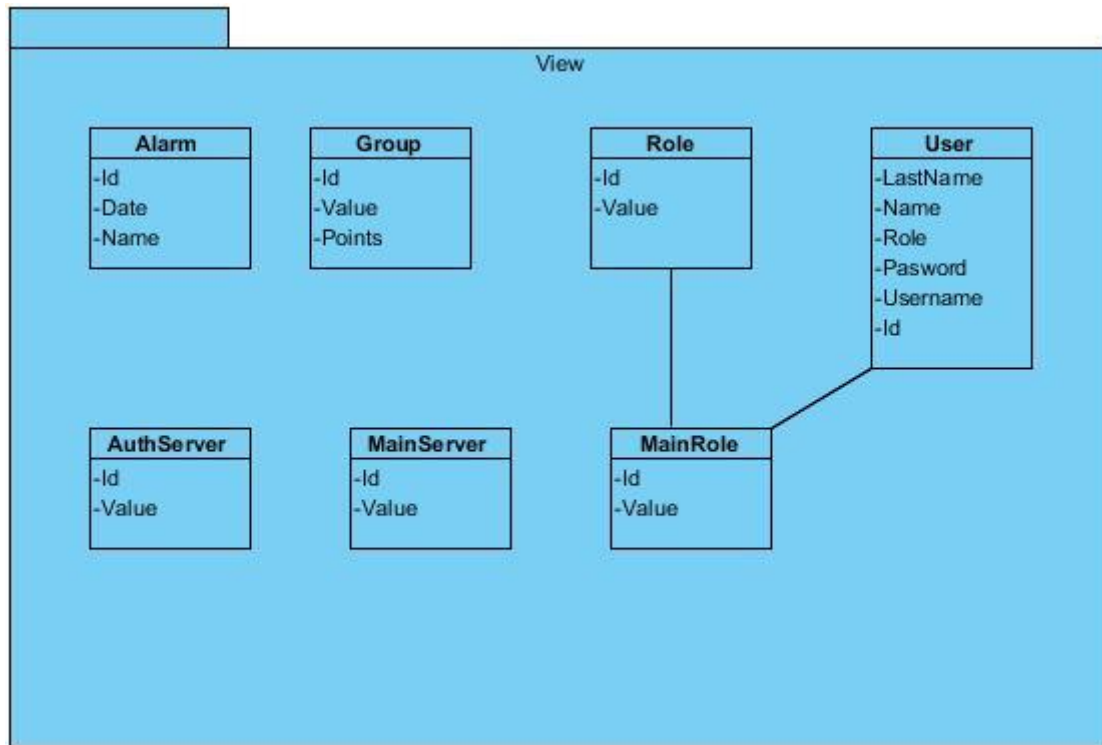


Fig 2.4 Diagrama de paquetes

En la investigación en curso solo es objetivo la interfaz de usuario, por lo que se representa en el diagrama solo las vistas con las que interactuará el usuario.

2.8.1 Descripción de las clases del diagrama de paquetes.

Clase Alarm

Clase que representa una alarma, tiene como atributos un Id, una fecha y un nombre.

Clase Group

Representa un grupo y sus atributos, un Id, una lista de puntos asociados y un valor.

Clase AuthServer

Es la clase que contiene los valores del servidor de autenticación.

Clase MainServer

Clase que contiene los valores del servidor principal.

Clase User

Clase que conforma un usuario con sus datos asociados, nombre, apellidos, rol, contraseña, nombre de usuario.

Clase Role

Clase que tendrá los roles que se definen.

Clase MainRole

Clase que manejará los usuarios y sus roles asociados.

2.9 Patrones de diseño

Los patrones de diseño de software son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. En el presente sistema se emplearon los patrones GRASP del inglés General Responsibility Assignment Software Patterns para la asignación de responsabilidades.

Experto: Las clases desarrolladas serán responsables solo de una actividad específica. Se manifiesta en la clase Generator es la encargada de las opciones de trabajo, graficar, cargar puntos, mover puntos etc. Pero dicha clase no estará asociada a la gestión de usuarios.

Alta Cohesión: Las clases no estarán sobrecargadas, ya que cada una será responsable de una actividad específica sin sobrecargas de métodos de otras. La clase Management es la encargada de gestionar los usuarios del sistema, la clase Generator la de graficar y demás opciones de trabajo y la clase Security de asignar el servidor principal.

Bajo Acoplamiento: Existe una limitada relación entre las clases del sistema. Las clases Generator, Management y Security no están relacionadas entre ellas.

2.10 Descripción de los casos de uso del sistema

A continuación, se muestran algunos de los casos de usos del sistema, los demás están adjuntos en el Anexo 1:

Tabla 2.4 .CU Autenticar usuario

Objetivo	Autenticar usuario	
Actores	Usuario	
Resumen	El usuario se autentica para acceder al sistema	
Complejidad	Baja	
Prioridad	Critica	
Precondiciones	El sistema debe mostrar la interfaz de autenticación	
Poscondiciones	El usuario debe autenticarse y acceder al sistema	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
	1. El usuario accede a la interfaz de autenticación.	Muestra los campos a llenar.
	2. El usuario introduce los datos (usuario y contraseña).	Verifica los datos introducidos.
	3. El usuario accede al sistema.	
Flujos alternos		
1ª. Campos vacíos o datos incorrectos		
		Muestra un mensaje notificando que existen datos incorrectos o campos vacíos. Volver a la acción 1 del actor.

Tabla 2.5 CU Gestionar Puntos

Objetivo	Cargar Puntos, Mover Puntos y Almacenar Información de los Puntos
Actores	Usuario
Resumen	Se deben cargar los puntos existentes en la BD, así como mover los puntos de un grupo hacia otro y almacenar la información de los puntos.

Complejidad	Baja	
Prioridad	Critica	
Precondiciones	El usuario debe estar autenticado y existir la conexión con el servidor.	
Poscondiciones	El sistema carga los puntos, los mueve y almacena la información correspondiente a cada punto correctamente.	
Flujo de eventos		
Sección 1: Cargar Puntos		
Flujo básico		
	Actor	Sistema
	1. Selecciona la opción Puntos.	Carga los puntos de la Base de datos.
	2. Selecciona el grupo que desea visualizar con los puntos asociados.	Muestra los grupos con sus puntos asociados.
Flujos alternos		
Sección 2: Mover Puntos		
Flujo básico		
	Actor	Sistema
	1. Selecciona el grupo al que desea mover los puntos (Sección 1).	Carga el grupo seleccionado.
	2. Selecciona la opción "Habilitar mover".	Desplaza el grupo cargado hacia la derecha.
	3. Selecciona el otro grupo desde el cual se moverán los puntos deseados.	Carga el grupo en la parte izquierda del obtenido previamente.
	4. Selecciona el punto que desea mover hacia el grupo de la derecha y presiona la opción "Mover".	Visualiza en el grupo de la derecha el nuevo punto movido.

Flujos alternos	
Sección 3: Almacenar información Puntos.	
Flujo Básico	
Actor	Sistema
1. Selecciona la opción "Guardar".	Muestra una notificación indicando que se guardó correctamente.
Flujos alternos.	

Conclusiones parciales

En este capítulo, se manifestó la solución propuesta como aspecto inicial para lograr el sistema a desarrollar. Se identificaron los requisitos funcionales y los no funcionales. También se describieron los casos de uso del sistema y se obtuvo una visión clara del proceso que se desarrolla a través del diagrama de dominio

Capítulo 3. Implementación y prueba

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que se cometan errores son comunes, conllevando a que el producto final no sea lo esperado por el cliente. Debido a esto, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad y el cumplimiento de los objetivos propuestos. En el presente capítulo se abordará sobre la validación de la solución propuesta donde se dejan expuestos los resultados obtenidos en las pruebas de caja negra al sistema desarrollado.

3.1 Introducción del capítulo.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo.

Los dos principales objetivos de la fase de pruebas son:

- Maximizar el número de errores detectados (cobertura).
- Reducir al mínimo el número de casos de prueba (costo).

Existen varios tipos de prueba, cada una enfocada a un objetivo único, que a su vez responden al correcto funcionamiento del sistema en cuestión, como, por ejemplo:

- Pruebas unitarias (Realizada a cada módulo del sistema o unidad mínima para mejorar la integración)
- Pruebas de integración (Encontrar fallos en la respuesta de sistemas que dependen de los servicios de otros)
- Pruebas de Humo (Smoke testing o Ad Hoc) (Probar el sistema constantemente)

-Pruebas de cargas (Verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga.)

-Prueba de Stress (Verificar que el sistema funciona correctamente bajo diversas circunstancias de stress (baja memoria, máximo número de clientes conectados, etc.))

-Pruebas de Seguridad y Control de Acceso (Nivel de seguridad de la aplicación)

-Caja Negra (Dado las diferentes entradas posibles se analizan las respuestas del sistema)

Estas solo son un ejemplo de las pruebas existentes que se le aplican a un determinado sistema para comprobar su correcta funcionalidad y que se obtengan los resultados esperados. Específicamente las pruebas de integración y de funcionalidad se le realizarán al sistema desarrollado para garantizar su correcto funcionamiento de acuerdo a los requisitos establecidos.

3.1.1 Pruebas de caja negra.

Idealmente, se determinaron un conjunto de casos de prueba tales que su ejecución exitosa implique que no hay errores en el software desarrollado. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se obtiene el resultado esperado por el usuario.

Tabla 3.1 Diseño de casos de prueba gestionar usuario

Sección	Escenario	Descripción	Variab le 1 Nomb re	Variabl e 2 Apellid os	Variabl e 3 Usuari o	Variable 4 Contrase ña	Variable 5 Repetir contrase ña	Variable 6 rol	Respuesta del sistema	Flujo Central
SC1.Adicio nar usuario	EC1.1 [Adicionar usuario correcto]	El Administra dor introduce los datos del usuario a adicionar.	V	V	V	V	V	V	Crea el usuario y notifica que se completó la operación con éxito.	El usuario introduce los parámetros requeridos en los campos correspondien tes.
			Luis	García Delgad o	Igdelga do	Luis*123	Luis*123	Administra dor		
	EC1.2[Adicio nar usuario incorrecto]	El Administra dor introduce datos no válidos para adicionar usuario.	I	V	V	V	V	Administra dor	Notifica que existen campos vacíos o datos incorrectos.	
			"Vacio"	García Delgad o	Igdelga do	Luis*123	Luis*123			
			V	I	V	V	V	Administra dor		
			Luis	"Vacio"	Igdelga do	Luis*123	Luis*123			
			V	V	I	V	V			

			Luis	García Delgado	“Vacio”	Luis*123	Luis*123	Administrador		
			V	V	V	I	V	Administrador		
			Luis	García Delgado	Igdelgado	“Vacio”	Luis*123	Administrador		
			V	V	V	V	I	Administrador		
			Luis	García Delgado	Igdelgado	Luis*123	“Vacio”	Administrador		
SC2.Modificar usuario	EC2.1[Modificar usuario correcto]	El administrador modifica los datos deseados.	V	V	V	V	N/A	N/A	Notifica que se modificó correctamente	El usuario modifica los campos deseados.
			Pepe	Arias Valdez	parias	Pepe.21				
	EC2.2[Modificar usuario incorrecto]	El administrador deja campos vacíos.	I	V	V	V	N/A	N/A	Notifica que existen errores en los campos.	
			“Vacio”	Arias Valdez	parias	Pepe.21				
			V	I	V	V				
			Pepe	“Vacio”	parias	Pepe.21				
			V	V	I	V				

			Pepe	Arias Valdez	“Vacio”	Pepe.21				
			V	V	V	I				
			Pepe	Arias Valdez	parias	“Vacio”				
SC3.Eliminar usuario	EC3.1[Eliminar usuario]	El administrador selecciona el usuario a eliminar y presiona el botón aceptar.	N/A	N/A	N/A	N/A	N/A	N/A	Elimina el usuario selecciona do.	Selecciona de la lista el usuario que desea eliminar.

Tabla 3.2 Diseño de caso de prueba autenticar usuario

Escenario	Descripción	Variable 1 Usuario	Variable 2 Contraseña	Respuesta del sistema	Flujo central
EC 1.1.[Usuario correcto]	El usuario introduce datos correctos para autenticarse en el sistema	V	V	El usuario queda autenticado en el sistema.	El usuario ingresa su usuario y contraseña en el mecanismo de autenticación del Sistema Generador de Gráficos.
		Igdelgado	Luis*123		
		V	I		

EC 1.2[Usuario incorrecto]	El usuario introduce datos no válidos para autenticarse en el sistema.	lgdelgado	“Vacio”	Muestra un mensaje de error, por lo tanto no se autentica el usuario.
		I	V	
		“Vacio”	Luis*123	

Tabla 3.3. Diseño de caso de prueba definir nombre equipo.

Escenario	Descripción	Variable 1 Nombre equipos	Respuesta del sistema	Flujo Central
EC1.1[Nombre equipo correcto]	El usuario introduce el nombre que desea asignar al equipo seleccionado.	V	Cambia el nombre del equipo y notifica al usuario que se guardó el cambio.	El usuario selecciona el equipo, luego selecciona la opción cambiar nombre, e introduce el nuevo nombre del equipo seleccionado.
		Tanque_Petróleo		
EC1.1[Nombre equipo incorrecto]	El usuario no introduce ningún nombre.	I	No modifica el nombre del equipo seleccionado.	
		“Vacio”		

3.2 No conformidades

A la aplicación se le realizaron tres iteraciones en las que fueron encontradas 5 no conformidades en la primera iteración, dos en la segunda y ninguna en la tercera, las no conformidades fueron corregidas correctamente.

Tabla 3.4 No conformidades

Iteración	Elemento	# NC	Descripción	Ubicación de la NC	Clasificación	Estado
1	Aplicación	1	No cambia de gráfica cuando se desea graficar en una gráfica distinta a la actual.	Icono de la barra de herramientas dentro de la opción Puntos.	Alta	Resuelta
	Aplicación	2	No grafica las alarmas por mes.	Opción comportamiento de alarmas, seleccionar mes.	Media	Resuelta
	Aplicación	3	Falta la tilde en la palabra "Gráfico".	Modulo Generador de Gráficos, opciones de la izquierda.	Alta	Resuelta
	Aplicación	4	No muestra la barra de herramientas en la Sección Puntos.	Sección Puntos del módulo Generador de Gráficos.	Alta	Resuelta
	Aplicación	5	No grafica las alarmas al seleccionar la fecha.	Modulo Generador de Gráficos, sección Comportamiento Alarmas.	Alta	Resuelta
2	Aplicación	1	No accede al menú de Administración.	Página principal luego de autenticarse.	Alta	Resuelta.
	Aplicación	2	No mueve el punto seleccionado de una gráfica a otra.	Opción mover punto de la barra de herramientas dentro de la Opción Puntos.	Media.	Resuelta.

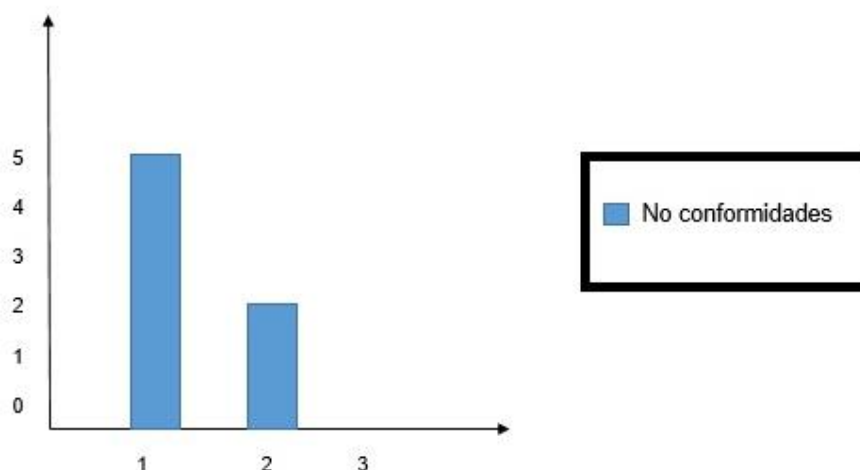


Fig 3.1 No conformidades

3.3 Resultados de las pruebas de caja negra

Para la realización de las pruebas de caja negra, fueron probados tres casos de uso, los que representan el cincuenta por ciento del total. Para cada caso de uso se tuvo en cuenta los diferentes escenarios que pudieran existir, los cuales implican cada una de las posibles interacciones del usuario o combinaciones de situaciones posibles con el fin de evaluar el comportamiento del sistema ante cada funcionalidad. El resultado de cada prueba coincide con el resultado esperado lo que demuestra la correcta implementación del sistema. También se encontraron varias no conformidades las cuales fueron resueltas con éxito.

Conclusiones parciales

Al concluir este capítulo se logró la validación de las funcionalidades del sistema a partir de la realización de pruebas de integración y de caja negra. La integración fue probada con el servidor encargado de enviar las peticiones del sistema. Las pruebas de caja negra se realizaron a partir de los casos de prueba que permitieron explorar cada uno de los escenarios del sistema. Se obtuvo como resultado, un sistema web capaz de graficar datos con el objetivo de fomentarles a los expertos la información de manera graficada para así apoyar la toma de decisiones en determinada industria.

Conclusiones generales

Con la realización de la investigación se logró dar cumplimiento a los objetivos propuestos:

- Desarrollar una herramienta capaz de graficar información y participar en la toma de decisiones.
- Generar Inteligencia de Negocio mediante la solución desarrollada.
- Crear un Módulo de Visualización del Cuadro de Mando Integral para el SCADA SAINUX.

Recomendaciones

Al concluir el presente trabajo se recomienda: - Añadir la posibilidad de graficar la información en otros tipos de gráficas, como gráficas de ramas y troncos, mejorando la visualización gráfica de la herramienta.

Referencias bibliográficas

1-Josep Lluís Cano.2007. BUSINESS INTELLIGENCE: COMPETIR CON INFORMACIÓN [En línea]. http://www.iwith.org/pdf/Libro_BI_Cometir_con_Informacion.pdf

2-Ingeniería del Software 7ma. Ed. - Ian Sommerville. [En línea]. <http://zeus.inf.ucv.cl/~bcrawford/Modelado%20UML/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf>

4-Como utilizar el cuadro de mando integral. Resumen. [En línea]. <https://www.leadersummaries.com/ver-resumen/como-utilizar-el-cuadro-de-mando-integral>

5-GUÍA PARA LA PRESENTACIÓN DE GRÁFICOS ESTADÍSTICOS. [En línea]. <https://www.inei.gob.pe/media/MenuRecursivo/metodologias/libro.pdf>

9-Durandal. [En línea]. <http://durandaljs.com/>.

8-Escribiendo aplicaciones con websocket. [En línea]...https://developer.mozilla.org/es/docs/WebSockets-840092-dup/Writing_WebSocket_client_applications

9-LibrosWeb.JavaScript[En línea]. <https://librosweb.es/libro/javascript/>

10-Tutorial Knockout.js. [En línea]...<https://docs.asp.net/en/latest/client-side/knockout.html>

11-Tutorialspoint.Knockout. [En línea]. <http://www.tutorialspoint.com/knockoutjs/>

- 12- Pressman, Roger S. Ingeniería de software: Un enfoque práctico. Cap. 7. s.l.: Editorial: Mc Graw Hill (Sexta Edición), 2005. 0-07285318-2
- 13-Pressman, Roger. Ingeniería de Software: Un enfoque práctico. Capítulo 10: Diseño arquitectónico. s.l.: Editorial: Mc Graw Hill (Sexta Edición), 2005. 0-07-285318-2
- 14-Fidias G, Arias. El proyecto de investigación. Introducción a la metodología científica 5ta Edición. Caracas, Venezuela: Editorial EPISTEMA, 2006. 980-07-8529-9.
- 15-Alonso, J.F. Curso de metodología de la investigación. Facultad de Ciencias Sociales y Humanísticas. Universidad Central de las Villas.: s.n., 1998.
- 16- Cadavid Andrés Navarro, Fernández Martínez Juan Daniel, Morales Vélez Jonathan. Revisión de metodologías ágiles para el desarrollo de software. 2012.
- 17- 32. Modelo de Implementación: Diagramas de Componentes y Despliegue. [En línea] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf..>
- 18-- Ian Sommerville. Ingeniería de Software (Parte IV Desarrollo). 7ma. s.l.: PEARSON Addison-Wesley, 2005. ISBN 84-7829-074-5.
- 19-- 830-1998., IEEE STD. "Especificaciones de los requisitos del Software".
- 20- Gauchat Juan Diego. El gran libro de HTML5, CSS3 y JavaScript. Barcelona: MARCOMBO, 2012. 08007.
- 21-. Enciclopedia Cubana en la Red. [. [En línea] http://www.ecured.cu/index.php/Visual_Paradigm..

- 22- Sánchez, Tamara Rodríguez. Metodología de desarrollo para la Actividad productiva de la UCI.
- 23- Canós José H., Letelier Patricio y Penadés M^a Carmen. Metodologías Ágiles en el Desarrollo de Software. Valencia: s.n. s/n 46022.
- 24- Adictos al trabajo. Ferri Pérez, Francisco [En línea] <https://www.adictosaltrabajo.com/tutoriales/zk-mvc-mvvm/>
- 25- Wikipedia [En línea] <https://es.wikipedia.org>
- 26- Ministerio de Educación de España. [En línea] http://www.ite.educacion.es/formacion/materiales/180/cd/m4_10/grficos_de_lineas.html
- 27- Estadística para todos. [En línea] <http://www.estadisticaparatodos.es/taller/graficas/cajas.html>
- 28-Software.com [En línea] <http://www.software.com.ar/p/visual-paradigm-para-uml>
- 29-Rendimientodesistemas [En línea] <http://rendimientodesistemas.blogspot.com/p/estructura-de-las-aplicaciones-web.html>
- 30-Aplicacionesweb [En línea] <http://aplicacionesweb-epoch.blogspot.com/p/caracteristicas-principales.html>
- 31-Netconsulting [En línea] <http://www.netconsulting.es/blog/nodejs/>
- 32-IBM [En línea] http://www.ibm.com/support/knowledgecenter/es/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/cwlp_websockets.html

- 33-Geekytheory [En línea] <https://geekytheory.com/json-i-que-es-y-para-que-sirve-json/>
- 34-Developer.mozilla [En línea] <https://developer.mozilla.org/es/docs/Web/JavaScript>
- 35- Abad, P. and Huapaya, E. Guía para la presentación de datos estadísticos.

Anexo 1. Descripción de Casos de Usos del Sistema

Tabla 2.6 CU Gestionar usuario

Objetivo	Crear, eliminar, modificar usuarios, asignar permisos, guardar datos usuarios, asignar rol.	
Actores	Administrador	
Resumen	El administrador gestiona los usuarios, esto le permite añadir, eliminar o modificar los usuarios, así como asignar roles, guardar los datos de usuarios y asignar permisos.	
Complejidad	Baja	
Prioridad	Critica	
Precondiciones	El administrador debe estar autenticado y existir la conexión hacia el servidor y la BD.	
Poscondiciones	Se añade, se elimina, se modifica un usuario, se asignar roles, permisos y guardan datos de usuarios.	
Flujo de eventos		
Sección 1: Crear usuario		
Flujo básico		
	Actor	Sistema
	1. Selecciona la opción Administración.	Muestra la interfaz de gestión de usuarios.
	2. Selecciona el enlace Usuarios.	Muestra las opciones que se realizan en la gestión de usuarios.
	3. Selecciona la opción Crear Usuario.	Muestra los campos que se deben llenar con los datos del usuario.
	4. Introduce los datos del usuario, y presiona el botón "Enviar".	Muestra una notificación indicando que se creó correctamente.
Flujos alternos		
2ª Faltan campos por llenar o datos incorrectos.		
	Actor	Sistema

1.	Muestra una notificación de que existen campos por llenar o datos incorrectos. Vuelve a la acción 2 del actor.
Sección 2: "Modificar usuario."	
Flujo básico	
Actor	Sistema
1. Selecciona el enlace Usuarios.	Muestra las opciones de la gestión de usuarios, así como un buscador de usuarios.
2. Busca o selecciona el usuario que desea modificar, y presiona sobre la opción modificar.	Muestra los datos del usuario seleccionado.
3. Modifica los campos deseados y presiona el botón "Aceptar".	Muestra una notificación señalando que se modificó correctamente.
Flujos alternos	
3ª Campos vacíos o errores en los campos	
1.	Muestra un mensaje de error al llenar los campos o existencias de campos sin datos. Volver a la acción 3 del actor.
Sección 3 "Eliminar usuario"	
Flujo básico	
Actor	Sistema
1. Selecciona el enlace Usuarios.	Muestra las opciones de la gestión de usuarios, así como un buscador de usuarios.

2. Busca o selecciona el usuario que desea eliminar, y presiona sobre la opción eliminar.	Elimina el o los usuarios seleccionados y muestra un mensaje de confirmación.
Flujos alternos	
2ª No selecciona el usuario	
Actor	Sistema
1.	Muestra una notificación de que no existen usuarios seleccionados. Vuelve a la acción 2 del usuario.

Tabla 2.7 CU Modo Visualización

Objetivo	Seleccionar cómo visualizar información.	
Actores	Usuario	
Resumen	El usuario decide mediante qué tipo de graficas visualizar la información.	
Complejidad	Baja	
Prioridad	Critica	
Precondiciones	El usuario debe estar autenticado.	
Poscondiciones	Se guarda la configuración realizada por el usuario	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
	1. Selecciona la opción "Generador de Gráficos".	Muestra las opciones de trabajo.
	2. Selecciona la opción "Gráficos".	Muestra las opciones de trabajo.
	3. Selecciona el grupo que desea visualizar.	Muestra el grupo seleccionado graficado.
	4. Selecciona la opción "Tipo de Gráfica". Y presiona el botón "Aceptar".	Modifica la gráfica previamente visualizada a la nueva elección.
Flujos alternos		

2ª No selecciona ninguna gráfica	
Actor	Sistema
1.	Muestra la información en la gráfica predeterminada.

Tabla 2.8 CU Definir nombres de equipos

Objetivo	Definir el nombre de los equipos.	
Actores	Usuario	
Resumen	El usuario define nombres para los equipos existentes.	
Complejidad	Baja	
Prioridad	Critica	
Precondiciones	El usuario debe estar autenticado.	
Poscondiciones	Se guarda la configuración realizada por el usuario	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
	1. Selecciona la opción "Gráficos" o "Puntos".	Muestra las opciones de trabajo según la elección realizada.
	2. Selecciona el grupo que desea visualizar.	Visualiza el grupo seleccionado.
	3. Selecciona la opción modificar grupo.	Muestra el nombre del grupo con la posibilidad de modificarse.
	4. Modifica el nombre del grupo y presiona el botón "Aceptar".	Guarda los cambios realizados.
Flujos alternos		
4ª No introduce ningún nombre.		
	Actor	Sistema
	1.	Muestra un mensaje indicando que no ha introducido ningún nombre. Vuelve a la acción 4 del actor.

Tabla 2.9 CU Gestionar Información

Objetivo	Visualizar último valor obtenido por puntos y visualizar la cantidad de alarmas.	
Actores	Usuario	
Resumen	Se visualiza último valor obtenido por puntos y la cantidad de alarmas.	
Complejidad	Baja	
Prioridad	Critica	
Precondiciones	El usuario debe estar autenticado.	
Poscondiciones	Se visualiza la información requerida.	
Flujo de eventos		
Sección 1: Visualizar último valor obtenido por puntos		
Flujo básico		
	Actor	Sistema
	1. Selecciona la opción "Generador de Gráficos".	Muestra las opciones de trabajo.
	2. Selecciona la opción "Gráficos".	Muestra las opciones de trabajo relacionadas con la selección.
	4. Selecciona el grupo que desea visualizar.	Muestra el grupo seleccionado, con sus puntos asociados y el último valor de cada punto.
Flujos alternos		
Sección 2: Visualizar la cantidad de alarmas.		
Flujo básico		
	Actor	Sistema
	1. Selecciona la opción "Comportamiento de alarmas".	Muestra la opción de seleccionar si se desea graficar la información por mes, día o año.
	3. Selecciona como agrupar la información a visualizar.	Muestra la información graficada.
Flujos alternos		