

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



Título: Aplicación web para la monitorización de los indicadores claves de rendimiento de los funcionarios en el Centro de Información del sistema DALASQ.



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

Autor: Orson Omar Duque Cortina

Tutores: Ing. Yasiel Jorge Martínez

Ing. Yudiel Guerra Valle

CIUDAD DE LA HABANA, CUBA 2015

“Año 58 de la Revolución”

Declaración de autoría

Declaro que soy el autor de este trabajo de diploma y concedo a la Universidad de las Ciencias Informáticas y el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales los derechos patrimoniales del mismo, con carácter exclusivo.

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

Autor

Orson Omar Duque Cortina

Tutor

Ing. Yasiel Jorge Martínez

Tutor

Ing. Yudiel Guerra Valle

Datos de contacto

Tutores

Ing. Yasiel Jorge Martínez

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Categoría docente: Instructor

Años de graduado: 5

Universidad de las Ciencia Informáticas, La Habana, Cuba.

yjmartinez@uci.cu

Ing. Yudiel Guerra Valle

Especialidad de graduación: Ingeniero en Ciencias Informáticas

Años de graduado: 1

Universidad de las Ciencia Informáticas, La Habana, Cuba.

yguerrav@uci.cu



"Los hombres geniales empiezan grandes obras, los hombres trabajadores las terminan."

"Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula, que nunca podrán saber a dónde van."

Leonardo Da Vinci

DEDICATORIA

*A mi familia... en especial a mis padres, mi súper hermano y abuelos.
A la genial isla del Caribe llamada Cuba.*

AGRADECIMIENTOS

A los que de una forma u otra tributaron a mi formación como persona, como profesional.

A mis tutores.

Al tribunal.

A mis profesores a lo largo de estos cortos 5 años.

A la FEU de Mella por inculcarme las ganas de seguir "haciendo", transformando y permitirme el honor de cargar en más de una ocasión con la ingratitud de los hombres como único y preciado premio.

A mis piquetes de la FEU, de todos aprendí la importancia de tener un equipo que te respalde.

A la UCI por ser un lugar especial.

A la Facultad 5 en su totalidad por hacerme sentir en casa.

A mis amigos, en especial: Alexis, Bravo, Arnaldo, Leiser, Yoanita, Erduin, Yandry, Esmaykel, Kiki, Riogel, Yanet, Ernesto, Rosabel, Joyce.

A Michel Sariol más que amigo, hermano.

A Marín, más que amigo, mi hijo.

A mi novia, que escribir algo sería faltarle a la verdad y nunca me explicaría bien.

A mi familia.

A mis abuelos.

A mi ídolo y hermano de sangre, mi guía en todo sentido.

A mi papá... por ser mi apoyo y mi columna en tiempos turbulentos.

A mi mamá... por ser mi máxima confidente y la mejor amiga que puedo tener.

Resumen

La monitorización es una necesidad de cada institución para lograr resultados exitosos. La supervisión de las actividades incluye la recopilación, medición y difusión de información sobre el rendimiento, así como la evaluación de las mediciones para llevar a cabo las mejoras necesarias. Las empresas ante la necesidad de tener un mejor control de los procesos que llevan a cabo, se ven en la necesidad de gestionarlos, de forma tal que le permitan medir su desempeño y poder cumplir con la estrategia trazada. En la presente investigación se realiza un estudio relacionado con los Indicadores Clave de Rendimiento, el *dashboard* y su utilidad como método eficaz para la monitorización de los procesos en la empresa.

El objetivo del trabajo es realizar una aplicación web para el Centro de Información del sistema DalasQ desarrollado por el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales, perteneciente a la Universidad de las Ciencias Informáticas, que permita monitorizar los indicadores generados por dicho sistema, referentes al desempeño de sus funcionarios. De esta manera, se persigue permitir la toma de decisiones basada en datos por parte de los directivos, ahorrar tiempo en la obtención de reportes, crear gráficos, así como la notificación de alertas una vez detectado desvíos en el plan operativo.

Palabras claves: *dashboard*, indicadores claves de rendimiento.

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN.....	5
1.1 INTRODUCCIÓN.....	5
1.2 DEFINICIONES.....	5
1.2.1 <i>Sistemas en Tiempo Real (STR)</i>	5
1.2.2 <i>Proceso de monitorización y control</i>	6
1.2.2.1 <i>Monitorizar</i>	6
1.2.3 <i>Cuadro de Mando Integral (CMI)</i>	7
1.2.4 <i>Dashboard</i>	8
1.2.5 <i>Indicadores claves de rendimiento</i>	9
1.3 ESTUDIO DE SOLUCIONES EXISTENTES	11
1.3.1 <i>Sistemas de monitorización en el mundo</i>	12
1.3.2 <i>Sistemas de monitorización en Cuba</i>	13
1.3.3 <i>Metodologías Ágiles</i>	14
1.3.4 <i>Frameworks</i>	17
1.3.5 <i>Servidor web</i>	20
1.3.6 <i>Herramienta CASE</i>	20
1.3.7 <i>Lenguaje de programación</i>	21
1.3.8 <i>Bibliotecas de programación</i>	22
1.3.9 <i>Entorno de desarrollo integrado</i>	23
1.3.10 <i>Sistema gestor de bases de datos (SGBD)</i>	23
CONCLUSIONES PARCIALES	25
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN	26
2.1 INTRODUCCIÓN.....	26
2.2 MODELO DE DOMINIO.....	26
2.3 PROPUESTA DE SOLUCIÓN	27
2.4 LISTA DE RESERVA DEL PRODUCTO	28
2.5 HISTORIAS DE USUARIO	30
2.6 TAREAS DE INGENIERÍA	32
2.7 PLAN DE RELEASE	33
2.8 ARQUITECTURA DE SOFTWARE (AS)	34
2.9 MODELO DE DISEÑO	35
2.9.1 <i>Patrones de diseño</i>	35
CONCLUSIONES PARCIALES	38
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS AL SISTEMA.....	39
3.1 INTRODUCCIÓN.....	39
3.2 DIAGRAMA DE CLASES	39
3.3 ESTÁNDAR DE CÓDIGO.....	40
3.4 MODELO DE DESPLIEGUE	42
3.5 PRUEBAS	43
3.5.1 <i>Casos de prueba</i>	43
3.5.2 <i>Resultados obtenidos</i>	45
3.6 TRATAMIENTO DE ERRORES.....	46
CONCLUSIONES PARCIALES	47
CONCLUSIONES.....	48
RECOMENDACIONES	49

REFERENCIAS BIBLIOGRÁFICAS	50
ANEXO 1: HISTORIAS DE USUARIO	55
ANEXO 2: TAREAS DE INGENIERÍA.....	64

ÍNDICE DE TABLAS

Tabla 1: Lista de Reserva de Producto. Requisitos Funcionales.....	28
Tabla 2: Lista de reserva del producto. Requisitos no funcionales.	29
Tabla 3: HU_5 Representar las regiones mediante un dashboard según los valores pertinentes de los indicadores.	30
Tabla 4: HU_6 Representar las oficinas de la región seleccionada junto con los estados de sus indicadores.....	31
Tabla 5: Tarea de ingeniería: 1.1	32
Tabla 6: Tarea de ingeniería 2.1	32
Tabla 7: Plan de release.	33
Tabla 8: Caso de prueba de aceptación: HU_1_P1.	44
Tabla 9: Caso de prueba de aceptación: HU_1_P2.	44
Tabla 10: Caso de prueba de aceptación: HU_3_P1.	45
Tabla 11: HU_1: Autenticar usuario.	55
Tabla 12: HU_2: Visualizar usuario.....	55
Tabla 13: HU_3 Representar árbol de ubicaciones de las instalaciones por niveles de división territorial.....	56
Tabla 14: HU_4 Configurar los valores mínimos y máximos de los indicadores de las instalaciones en todo momento.	57
Tabla 15: HU_7 Representar el estado de los indicadores de los funcionarios de la oficina seleccionada.....	58
Tabla 16: HU_8 Crear y modificar las alarmas de las instalaciones mediante el correo del supervisor.....	59
Tabla 17: HU_9 Enviar correo al supervisor de la instalación una vez disparado un indicador de rendimiento.	60
Tabla 18: HU_10 Visualizar estadísticas de los funcionarios.....	61
Tabla 19: HU_11 Generar reportes.	63
Tabla 20: Tarea de Ingeniería HU_3.	64
Tabla 21: Tarea de Ingeniería HU_4.	64
Tabla 22: Tarea de Ingeniería HU_5.	64
Tabla 23: Tarea de Ingeniería HU_5.	65
Tabla 24: Tarea de Ingeniería HU_6.	65
Tabla 25: Tarea de Ingeniería HU_6.	66
Tabla 26: Tarea de Ingeniería HU_7	66
Tabla 27: Tarea de Ingeniería HU_7.	67
Tabla 28: Tarea de Ingeniería HU_8.	67
Tabla 29: Tarea de Ingeniería HU_12.	68
Tabla 30: Tarea de Ingeniería HU_13.	68
Tabla 32: Tarea de Ingeniería HU_14.	68

ÍNDICE DE FIGURAS

Figura 1: Modelo del dominio. (Elaboración Propia)	26
Figura 2: Arquitectura N-Capas (Elaboración Propia).	34
Figura 3: Diagrama de Clases.	39
Figura 4: Diagrama de Despliegue.	42

Introducción

Hoy día las empresas se decantan por la implementación de distintas herramientas o estrategias que les ayuden a alcanzar sus objetivos, en pos de adquirir ventajas respecto a sus homólogos. Esto explica claramente el papel fundamental que juegan las Tecnologías de la Información y las Comunicaciones (TIC) y las herramientas que colaboran en la toma de decisiones haciendo uso precisamente de indicadores claves de rendimiento.

Cuba es un país que tiene como política mantenerse al tanto de los avances científicos y tecnológicos en el mundo. Esto se corrobora luego de las políticas y principios planteados en el resumen de las bases y prioridades para el perfeccionamiento de la informatización de la sociedad en Cuba (1), en el que introduce las TIC con el objetivo de lograr la informatización de la sociedad, trayendo consigo el desarrollo cultural del pueblo y la satisfacción de las necesidades de información y conocimiento de los individuos. Tiene además, impacto directo en la optimización de procesos generando a su vez más divisas.

Como parte del proceso de informatización surge en el año 2002 la Universidad de las Ciencias Informáticas (UCI), la cual mantiene como premisa el desarrollo de software a partir de la vinculación docencia-producción como modelo de formación. Cuenta con una infraestructura productiva conformada por centros de desarrollo de software y dentro de este marco se inserta el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE). Este centro posee como una de sus directrices la confección de productos y tecnologías sobre *Java Enterprise Edition* (JEE). Dentro de las soluciones desarrolladas en este centro se encuentra el sistema de gestión de flujo “DalasQ”.

DalasQ es un software de manipulación de colas de personal encargado de optimizar el tiempo de espera de un cliente. Este producto está destinado a instituciones que brindan servicios públicos y necesiten gestionar el flujo del personal que pasa por sus instalaciones. En general, el proyecto se centra en proveer una solución tecnológica que propicie una mejor calidad de atención a los ciudadanos que acudan a las instalaciones, así como un mejor control de desempeño de la institución en la ejecución de los trámites concebidos en las oficinas.

El sistema DalasQ está compuesto por varias aplicaciones. Una de estas, el Centro de Información del Sistema DalasQ (CISD), es una aplicación web que tiene como objetivo: contribuir a la toma de decisiones de los directivos de la institución que utilice el sistema DalasQ.

Actualmente el CISD es capaz de monitorizar los Indicadores Claves de Rendimiento (Key Performances Indicator, KPI) de los servicios prestados a la población. Sin embargo presenta problemas que dificultan el cumplimiento de su objetivo respecto al desempeño de los funcionarios.

En ocasiones el tiempo de servicio prestado a la población varía, ya sea debido a que el número de clientes es mayor a lo usual o exista algún hecho institucional relacionado con el desempeño de los funcionarios de la institución. Esto conlleva a disminuir los niveles de servicio y calidad. Aumentando el tiempo de espera del cliente en el establecimiento, provocando su insatisfacción e incluso el abandono de la instalación, generando pérdidas financieras para esta entidad.

Existen funcionarios que no cumplen de forma correcta con sus obligaciones. Estos prestan servicios de forma lenta, exceso de tiempo ocioso, así como plazos largos entre una atención (momento en servicio) u otra. Además los supervisores de las distintas instalaciones tienen dificultades para realizar su trabajo de oficina y a la vez chequear el rendimiento de los funcionarios de su instalación, obstaculizándose su control sobre estos.

Estos problemas surgen debido a que los directivos no son capaces de detectar a tiempo, cuál funcionario o grupos de estos, pueden ser la causa de la pérdida de la calidad de los procesos. Así como la dificultad de comparar el desempeño de varios funcionarios e incluso conocer cómo se comporta este a lo largo del día.

Por las razones expuestas se plantea como **problema de investigación**: ¿Cómo monitorizar los indicadores claves de rendimiento de los funcionarios en el CISD para realizar la toma de decisiones basada en datos por parte de los directivos? siendo el **objeto de estudio**: Proceso de monitorización de indicadores claves de rendimiento, enmarcándose en el **campo de acción**: Proceso de monitorización de indicadores claves de rendimiento mediante *dashboard*.

Para dar solución al problema de investigación se propone como **objetivo general**: Desarrollar una herramienta informática que permita monitorizar los indicadores claves de rendimiento de los funcionarios en el CISD, para realizar la toma de decisiones basada en datos por parte de los directivos.

Para dar solución al objetivo general se han planteado las siguientes **tareas investigativas**:

1. Estudio, síntesis y análisis de la bibliografía existente sobre la monitorización de indicadores claves de rendimiento.

2. Selección de la tecnología y metodología de desarrollo más adecuada para la implementación de la aplicación de monitorización de los indicadores claves de rendimiento de los funcionarios en el CISD.
3. Diseño de la aplicación de monitorización de los indicadores claves de rendimiento de los funcionarios en el CISD.
4. Implementación de la aplicación de monitorización de los indicadores claves de rendimiento de los funcionarios en el CISD.
5. Pruebas a la aplicación de monitorización de los indicadores claves de rendimiento de los funcionarios en el CISD.

Idea a defender: El desarrollo de una aplicación de monitorización de los indicadores claves de rendimiento de los funcionarios en el CISD permitirá realizar la toma de decisiones basada en datos por parte de los directivos de la institución.

Lo que presagia como **posible resultado:** Aplicación web que permitirá monitorizar los indicadores claves de rendimiento de los funcionarios en el CISD, para realizar la toma de decisiones basada en datos por parte de los directivos de la institución.

Para la realización de la investigación y elaboración del presente trabajo se han utilizado varios métodos científicos de investigación:

❖ **Métodos Teóricos:**

- **Histórico-lógico:** Se utiliza para realizar una revisión histórica del desarrollo de los *dashboard* y constatar teóricamente cómo ha evolucionado la administración y configuración de la información que estos muestran.
- **Analítico-sintético:** Se utiliza para analizar las teorías, documentos e información referentes al objeto de estudio, permitiendo arribar a la propuesta de solución.

❖ **Método Empírico:**

- **Entrevista:** Consiste en la realización de un diálogo entre dos personas, el investigador y el entrevistado, el primero tiene como objetivo obtener información sobre los *dashboard* a partir del conocimiento que posee el entrevistado, así como los posibles KPI que puede brindar el CISD luego de procesada la información.

El contenido de la presente investigación se encuentra comprendido en tres capítulos estructurados de la siguiente manera:

- **Capítulo 1: Fundamentación teórica de la investigación.**

En este capítulo se brinda una descripción general del objeto de estudio y los sistemas existentes tanto nacionales como internacionales asociados al campo de acción. Se plantea un análisis de varios conceptos relacionados al objeto de estudio, se definen las técnicas, tecnologías y metodología utilizadas para dar solución al problema planteado.

- **Capítulo 2: Propuesta de solución.**

En este capítulo se realiza la captura de requerimientos planteados por el cliente, se formula una propuesta de solución a implementar, se diseña la arquitectura a implementar y se generan los artefactos correspondientes según la metodología seleccionada.

- **Capítulo 3: Implementación y pruebas del sistema.**

En este capítulo se exponen otros artefactos que genera la metodología de desarrollo así como las pruebas utilizadas para la validación de la solución propuesta y los resultados arrojados.

Capítulo I: Fundamentación teórica de la investigación.

1.1 Introducción

En el presente capítulo se fundamenta la investigación y la propuesta de solución a través del estudio teórico realizado, luego de la búsqueda, recopilación y análisis de la información relacionada con el campo y objeto de estudio. Además se refieren los conceptos claves o definiciones asociadas: monitorización, los KPI, el *dashboard* y tiempo real. También se realiza un estudio del Cuadro de Mando Integral (CMI) como herramienta empresarial. Asimismo permite conocer el estado de los sistemas similares existentes en Cuba y el mundo. En adición se describen las tecnologías y herramientas utilizadas para el análisis, diseño e implementación del sistema propuesto.

1.2 Definiciones

1.2.1 Sistemas en Tiempo Real (STR)

Existen muchas interpretaciones de la naturaleza exacta de un Sistema en Tiempo Real, sin embargo, todos tienen en común la noción de tiempo de respuesta.

El diccionario Oxford de computación ofrece la siguiente definición:

Es cualquier sistema donde el tiempo en que se produce su salida no es significativo. Por lo general debido a que la entrada corresponde a algún instante del mundo físico y la salida tiene relación con ese mismo instante. El retraso transcurrido entre la entrada y la salida debe ser lo suficientemente pequeño para considerarse una respuesta puntual. (2)

Otra definición de sistema de tiempo real dada por Philip A. Laplante:

Un sistema de tiempo real puede definirse, entonces, como un sistema que debe satisfacer restricciones explícitas en el tiempo de respuesta o arriesgarse a severas consecuencias, incluida la falla. Por lo tanto un sistema de tiempo real es un sistema que responde a un estímulo externo dentro de un tiempo especificado. (3)

En adición los STR pueden dividirse en dos clasificaciones, en función de las restricciones de tiempos:

- **Sistemas de tiempo real duros** (*Hard real-time systems*): sistemas donde es imperativo que la respuesta ocurra dentro del plazo requerido. Ej: Sistema de Control de Vuelo. (4)
- **Sistemas de tiempo real blandos** (*Soft real-time systems*): sistemas donde los plazos son importantes pero el sistema sigue funcionando si no se cumplen los plazos ocasionalmente. Ej: Sistema de Adquisición de Datos. (4)

En la presente investigación no se considera crítico mostrar la información en tiempo real, pero a petición del cliente se establece un tiempo de latencia de dos segundos. Debido a esto se considera el tipo de sistema de tiempo real blando como parte de la solución, pues puede tolerar un exceso de tiempo de respuesta, ya que no es vital para la institución conocer esta información en el instante exacto en el que sucede.

1.2.2 Proceso de monitorización y control

El proceso de monitorización y control es un conjunto de actividades de gestión que permiten verificar si una institución va marchando según lo planificado. Controla el avance en ejecución de la institución, compara el desempeño de la misma, mide los resultados reales a partir de lo planeado y revisa el comportamiento de los indicadores clave de rendimiento.

La monitorización y control es una necesidad de cada institución para lograr resultados exitosos. La supervisión de las actividades incluye la recopilación, medición y difusión de información sobre el rendimiento, así como la evaluación de las mediciones y tendencias para llevar a cabo las mejoras del proceso. La presente investigación se enmarca en el proceso de monitorización.

1.2.2.1 Monitorizar

Existen muchas definiciones propuestas por varios especialistas e investigadores pero la mayoría coincide en la importancia de la observación buscando posibles anomalías.

Supervisar las actividades en progreso para asegurarse de que están en curso y horario en el cumplimiento de los objetivos y metas de desempeño. (5)

Observar y comprobar el progreso o la calidad de algo durante un período de tiempo; mantener bajo revisión sistemática. (6)

En la presente investigación cuando se refiera al término monitorizar se refiere a la acción de

observar, medir y analizar datos brindados por un sistema, en busca de posibles anomalías previamente definidas por los directivos de la institución.

1.2.3 Cuadro de Mando Integral (CMI)

Es una herramienta de control empresarial que utiliza cuatro perspectivas para controlar la implementación de la estrategia en sus factores críticos de éxito y su adecuación al entorno. Estas perspectivas son: financiera, del cliente, de procesos internos así como de formación y crecimiento. Todas estas ligadas, contribuirían a controlar el cumplimiento de los objetivos estratégicos midiendo los factores claves y sus inductores de actuación para lograr un control anticipado y enfocado no solamente en la actuación financiera.

Según sus creadores, Norton y Kaplan:

El CMI aporta la “receta” que permite que unos ingredientes ya existentes en una organización se combinen para crear valor a largo plazo. (7)

Es la estrategia perfecta que permite a la dirección de una institución inducir la al éxito competitivo. (7)

Perspectiva del cliente o consumidor: ¿Qué esperan de la empresa? El buen servicio al cliente es muy importante. Los clientes esperan los productos en óptima calidad, con un costo adecuado, que se entreguen a tiempo y que su rendimiento sea el convenido.

Perspectiva interna o de procesos internos: ¿En qué puede destacarse? ¿Qué hacer dentro de la empresa para cumplir con las expectativas de los clientes? Los procesos de la empresa deben estudiarse y evaluarse para conseguir la satisfacción de los consumidores.

Perspectiva de la innovación o aprendizaje y crecimiento: ¿Qué se debe continuar mejorando? La competencia es feroz en este nuevo milenio, por ello la empresa debe ser apta para mejorar e innovar.

Perspectiva financiera: ¿Qué esperan los accionistas? Quiénes invierten su dinero esperan, en forma legítima un rendimiento adecuado. Si esto no se complace, es probable que inviertan su dinero en una empresa diferente. Incluye además aquellos aspectos relacionados con los recursos humanos necesarios para poder implementar mejoras.

El CMI utiliza un conjunto de indicadores que a diferencia de los cuadros de mando tradicionales, están plenamente integrados y coordinados, a través de relaciones causa-efecto, con los objetivos y metas de la organización. Se trata de una herramienta de gestión que traduce la estrategia de la empresa en un conjunto coherente de indicadores. En su origen, incorporaba en un único documento diversos ratios para el control financiero de la empresa. Esta herramienta fue evolucionando con el paso de los años y actualmente, combina no sólo ratios financieros, sino también indicadores no financieros que permiten controlar los diferentes procesos del negocio, por lo que se le conoce entonces como CMI. (8)

La implementación del CMI establece la combinación de indicadores financieros y no financieros, permitiendo controlar los diferentes procesos del negocio en una empresa. Además es una filosofía práctica, que logra enlazar: visión, misión y estrategia con el fin de medir el desempeño corporativo.

Para la presente investigación se considera que el CMI no es idóneo para la propuesta de solución pues este brinda una vista integral de la institución y no una de sus perspectivas en específico, como si puede brindar un *dashboard*.

1.2.4 Dashboard

En español no existe una traducción única a este término, al traducir esta palabra al idioma español obtenemos una variedad de posibles sinónimos, entre los más comunes encontramos: cuadro de mando y tablero o panel de control. Para evitar imprecisiones será utilizado el término en inglés.

Es una pantalla que contiene los elementos de información necesarios para conseguir uno o más objetivos; consolidado y creado como única ventana donde toda la información puede ser revisada de un vistazo. (9)

Están diseñados para ayudar a interpretar el estado instantáneo de un elemento de control. Para proveer de un medio para controlar con rapidez el estado de cualquier sistema, deben ser diseñados de manera particular para aprovechar los puntos fuertes de la percepción visual y el conocimiento. (10)

En un *dashboard* se conjugan en una sola pantalla todos los KPI, objetivos y otros gráficos relevantes. Esta información viene estructurada de forma amigable de modo que es muy fácil de leer y visualizar. (11) Es una pantalla mediante la cual se despliega la información de la empresa

en tiempo real extraída de varias fuentes o bases de datos, esta característica le permite a los usuarios tener un conocimiento completo sobre el comportamiento de la empresa y realizar análisis instantáneos.

Para su diseño se deben tener algunas consideraciones: solo deben mostrar los datos más importantes que deben vigilar, no incluir gráficos o alguna información que no sean útiles al usuario. Deben ser interpretados y leídos de una manera rápida y sencilla, deben saber mostrar en un espacio reducido todo el cúmulo de información vital que necesita la empresa, por lo que su diseño debe aprovechar el poder de la percepción visual. Los *dashboard* muestran lo que está pasando, han de permitir reconocer de una manera rápida lo que necesita de atención.

Se pueden definir las siguientes clasificaciones de **dashboard**: (12)

- ❖ Área: Ventas, Finanzas, Marketing, Producción, RRHH.
- ❖ Rol: Analítico, Estratégico, Operacional.
- ❖ Métricas: KPI, Six Sigma.
- ❖ Display: Gráfico, Texto, Gráfico y Texto.
- ❖ Frecuencia: Mes, Semana, Día, Hora, Tiempo Real.
- ❖ Tipos de Datos: Cuantitativos, No Cuantitativos.
- ❖ Alcance: Empresa, Departamento, Individuo.
- ❖ Interacción: Estático, Dinámico.

En el presente trabajo se han utilizado combinaciones de clasificaciones de *dashboard*. Es el caso de los Métricos al implicar KPI junto con el tipo *Display* y Frecuencia, haciendo uso de textos y gráficos en un tiempo establecido. Utilizando además los de tipo de Interacción Dinámico y el Alcance Individuo.

1.2.5 Indicadores claves de rendimiento

Son mediciones cuantificables que reflejan los factores críticos y de éxito de una organización. Utilizados para evaluar los progresos realizados respecto a los objetivos previamente acordados, permitiendo que los usuarios del negocio puedan saber instantáneamente si están dentro o fuera de su plan estratégico. (13)

Se refiere a datos esencialmente cuantitativos, que permiten darse cuenta cómo se encuentra la organización en relación con algún aspecto de la realidad, que sea de interés conocer. Los indicadores pueden ser números, hechos, opiniones o percepciones que señalen condiciones o

situaciones específicas. Además son instrumentos valiosos e importantes para evaluar el proceso de desarrollo y para orientar cómo se pueden alcanzar mejores resultados. Son utilizados para medir el progreso y los logros de un proyecto, así como, para hacer el seguimiento y la evaluación de los mismos. Son herramientas de gran utilidad para evaluar dónde se está y hacia dónde se va, respecto a los objetivos que se persiguen. (14)

En la presente investigación se considera la segunda definición más acertada y la que más se ajusta al desarrollo de la misma, dada en el año 2006 en un artículo publicado por varios autores cubanos referido a las tecnologías de información y las comunicaciones. La selección de esta definición se debe a que está más explícita, detallada y vinculado con la profesión rectora de esta investigación, la Informática.

Características de los indicadores (15)

- Representan fielmente el objetivo a medir mediante una relación directa.
- Son cuantificables a través de datos numéricos o un valor de clasificación.
- Son rentables, superando el beneficio de su uso al coste de su obtención.
- Pueden definir la evolución en el tiempo del objetivo siendo comparables en el tiempo.
- Son fiables para dar confianza a los usuarios sobre su validez.
- Son fáciles de mantener y utilizar.
- No deben interferir con otros indicadores siendo compatible con ellos.

Tipos de indicadores

✓ Indicadores financieros

Los indicadores financieros son los datos que permiten medir la estabilidad, la capacidad de endeudamiento, el rendimiento y las utilidades a una empresa o unidades dentro de la empresa. Solo se centran en resultados financieros. Mediante su uso es posible tener una interpretación de las cifras, resultados o información de su negocio para saber cómo actuar frente a las diversas circunstancias que pueden presentar y su comportamiento en relación con otras empresas de la competencia. (16)

✓ Indicadores numéricos

Un indicador numérico es una relación entre dos o más datos significativos que tienen una relación lógica entre ellos y que entregan información de aspectos críticos de vital importancia

para la conducción de la organización. Existen distintas clases de indicadores numéricos que son obtenidos a través de razones, porcentajes y promedios. Los indicadores numéricos se pueden clasificar en: (17)

1. Indicadores de resultados: A los que también se les conoce como indicadores de efecto. Miden la consecución del objetivo estratégico.
2. Indicadores de causa: A los que también se les conoce como indicadores inductores. Miden el resultado de las acciones que permiten su consecución.

✓ **Indicadores de gestión**

Los indicadores de gestión son la expresión cuantitativa del comportamiento o desempeño de toda una organización o una de sus partes, cuya magnitud al ser comparada con algún nivel de referencia, puede estar señalando una desviación sobre la cual se tomarán acciones correctivas o preventivas según el caso. Sus mediciones están relacionadas con el modo en que los servicios o productos son generados por la institución. Con los resultados obtenidos se pueden plantear soluciones o herramientas que contribuyan al mejoramiento o correctivos que conlleven a la consecución de la meta fijada. (18)

✓ **Indicadores estratégicos**

Los indicadores estratégicos son parámetros cualitativos o cuantitativos que definen los aspectos relevantes sobre los cuales se lleva a cabo la evaluación. Dicha evaluación se emplea con el fin de medir el grado de cumplimiento de los objetivos planteados en términos de eficiencia, eficacia y calidad, además, miden el grado de cumplimiento de los planes estratégicos o de desarrollo. (19)

En la presente investigación se considera que el tipo de indicadores a tener en cuenta, son los indicadores de gestión, más específicamente de rendimiento. Debido a que la meta es monitorizar los indicadores claves de rendimiento de los funcionarios de la institución en el CISD.

1.3 Estudio de soluciones existentes

Actualmente numerosas instituciones utilizan *dashboards* para lograr una mayor gestión teniendo como base el monitoreo de KPI, en ocasiones en tiempo real. Algunas de estas herramientas han sido seleccionadas para su análisis en busca de alguna posible solución al problema de investigación.

1.3.1 Sistemas de monitorización en el mundo

Internacionalmente existen numerosas aplicaciones que brindan la posibilidad de monitorizar procesos reconocidos como importantes por la empresa en cuestión. Por lo general estos sistemas trabajan con un gran volumen de información enfocada en las necesidades o proyección de la empresa.

IBM Cognos Real-time Monitoring

Es una solución de inteligencia empresarial operativa diseñada para proporcionar un acceso fácil a datos coherentes para que pueda reaccionar rápidamente a las oportunidades de ahorro e ingresos. Presenta paneles de instrumentos de autoservicio interactivos con mediciones de KPI para usuarios empresariales, ejecutivos, gestores y analistas de primera línea.

Las características de Cognos Real-time Monitoring incluyen: (20)

- **Análisis, informes y consultas de autoservicio** desde cualquier origen de datos.
- **Alertas definidas por la empresa** sobre problemas imprevistos que requieren atención.
- **Almacén de datos patentado basado en memorias** para la integración continua de datos.
- **Navegación de datos en tiempo real** que le permite investigar la causa raíz de un problema.
- **Entorno del panel de instrumentos de arrastrar y soltar** para poder añadir información con formato en tiempo real a la inteligencia empresarial histórica.

Pentaho *Dashboard*

Conceptualmente, *Pentaho Dashboard* es una plataforma integrada para proporcionar información sobre datos, donde se puede ver todo tipo de informes, gráficos interactivos y los cubos creados con las herramientas de Pentaho como Pentaho Report Designer. Se trata de una interfaz que ofrece una vista centralizada sobre los movimientos de datos profesionales, lo que permite seguirlos y tomar decisiones. (21)

Principales características de la herramienta:

- Está licenciado bajo GNU¹ *General Public License Version 2* (GPLv2).
- Incluye componentes de navegación y el visor de informes y análisis, que pueden ser integrados en los portales o páginas web.
- Genera contenido XML².
- Totalmente personalizable mediante definiciones XML de diseño y hojas de estilo.
- Ejecución de seguimientos de excepciones, permitiendo pre-establecer alertas basadas en reglas del negocio.
- Gran variedad de gráficos, tablas y velocímetros.
- Capacidad de exportar a HTML³, PDF, XLS, DOC, TIFF, CSV, XML.
- Sistemas Operativos compatibles: Windows, Macintosh, Unix y Linux.
- Envío de correos electrónicos.

1.3.2 Sistemas de monitorización en Cuba

A nivel nacional no se ha estado exento del desarrollo alcanzado por los CMI en el mundo. Existen numerosos sistemas de CMI entre los cuales es válido mencionar el de la Empresa de Telecomunicaciones de Cuba S.A (ETECSA) en su Unidad de Negocio Internacional (UNI), la cual automatiza el control de llamadas internaciones en factura de duración de las llamadas totales o al mes, los gastos, la cantidad de llamadas y el importe. Entre estos sistemas se pueden encontrar el de Servicios de Protección de Cienfuegos de la agencia Seguridad y Protección S.A (SEPSA), donde se establece un sistema de alertas sobre los principales procesos relacionados con el cliente y el del Grupo de Electrónica para el Turismo en Varadero (GET Varadero), el cual monitoriza los ingresos por grupo de clientes y el tiempo de respuesta de los servicios informáticos por zona.

En la UCI la monitorización ha sido objeto de numerosas investigaciones enfocados en su mayoría en el análisis y diseño de un CMI. A este grupo pertenece el proyecto PADTSI, uno de los más completos realizados en la institución que permite hacer la gestión necesaria con los diferentes indicadores y brinda una detallada visualización de la información.

¹ GNU GPL: Licencia que garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir y modificar el software.

² XML: Lenguaje de marcas extensibles.

³ HTML: HyperText Markup Language (Lenguaje de Marcado de Hipertexto)

Otro de los sistemas desarrollados por la UCI es el CISD. El cual incluye entre sus características el uso de tecnologías definidas por el proyecto DalasQ para sus aplicaciones web, el cual trabaja sobre tecnologías web, realiza reportes y gráficas a partir de los indicadores definidos pero no tiene a PrimeFaces 5.1 y JSF 2.1 como parte de sus tecnologías, siendo estas, herramientas definidas por el proyecto DalasQ, para sus aplicaciones web.

Estos sistemas nacionales, a excepción de PADTSI el cual constituye una herramienta genérica, no dan solución al problema planteado en la presente investigación, porque cada uno fue diseñado e implementado respondiendo a sus particularidades y requerimientos específicos, no permiten personalizar las interfaces, no viabilizan una posible integración con el sistema DalasQ, ya que fueron desarrollados haciendo uso de tecnologías y herramientas diferentes.

La herramienta IBM *Cognos Real-time Monitoring* es una de las más completas en el mundo, pero posee la característica de ser un sistema de carácter propietario, licencia costosa y solo disponible para su empleo en el sistema operativo Windows. Las herramientas Pentaho *Dashboard* y PADTSI, aunque resuelven parte de la solución no permiten la visualización de un *dashboard* global en forma de mapa haciendo uso de javascript.

Luego del análisis de sistemas que implementan procesos de monitorización de indicadores, se arriba a la conclusión de que aunque algunas satisfacían parcialmente las necesidades o características para darle solución a la problemática de la investigación, detalles importantes como las licencias de los productos y la compatibilidad con la versión actual del CISD definieron el no uso de estas soluciones. Por esta razón se define tomar como base para la solución, el CISD en su versión 1.0.

1.3.3 Metodologías Ágiles

Según Damon B. Poole (22), dar una definición concisa de metodología no es nada fácil, porque es un paraguas de una variedad amplia de metodologías y porque ágil está definido oficialmente como los 4 valores en el Manifiesto Ágil.

1. Individuos e interacciones sobre procesos y herramientas.
2. Software funcionando sobre documentación extensiva.
3. Colaboración con el cliente sobre negociación contractual.
4. Respuesta ante el cambio sobre seguir un plan.

Dado que existen tantas prácticas asociadas al desarrollo ágil, una forma más simple de definir

Ágil es hacerlo en término de beneficios. Es aquel que, en comparación con el desarrollo tradicional, provee beneficios de mayor flexibilidad, Retorno de Inversión más alto, realización más rápida del Retorno de Inversión, más alta calidad, mayor visibilidad, y paz sostenible. (22) A continuación se detallan las características de las metodologías ágiles más usadas en la actualidad.

Metodología Programación Extrema (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y facilidades para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. (23)

Algunas características principales de XP: (24)

- Pruebas unitarias: se basa en las pruebas realizadas a los principales procesos con el objetivo de detectar futuros errores.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Metodología SCRUM

Es una metodología ágil, ideal para administrar y controlar el desarrollo del software, sirve de soporte para acelerar la laboriosidad que se reconoció en XP. Se considera con un perfil de autogestión de los equipos de desarrolladores, está orientada en mayor medida a las personas que a los procesos, o sea, esta interviene para que el equipo trabaje en conjunto con el mismo objetivo y dirección. Su estructura de desarrollo es incremental basada en iteraciones al igual que XP, está diseñada para adaptarse a los cambios de requerimientos. Una de las mayores ventajas lo constituye que es muy fácil de entender y requiere poco esfuerzo para comenzar a usarse. (25)

Metodología SXP

Es un híbrido cubano que toma las buenas prácticas de XP y SCRUM, este ofrece una estrategia tecnológica a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva. Fomenta la responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. Consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto. Basada completamente en los valores y principios de las metodologías ágiles expuestos en el manifiesto ágil. Como método de estimación se utiliza la opinión de expertos y constan con métricas o indicadores para lograr una eficiente calidad. (26)

Consta de 4 fases principales: (27)

- **Planificación-Definición:** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- **Desarrollo:** es donde se realiza la implementación del sistema hasta que esté listo para ser entregado.
- **Entrega:** puesta en marcha.
- **Mantenimiento:** donde se realiza el soporte para el cliente.

En cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto (LRP), definición de las Historias de Usuario (HU), diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, lo que permite mejorar el diseño cada vez que se le añada una nueva funcionalidad. (27)

En la presente investigación se asume esta metodología de desarrollo como la adecuada por las siguientes razones: está especialmente diseñada para proyectos con pequeños equipos de trabajo, corto período de desarrollo y posibilita la existencia de requisitos cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Además, permite seguir de forma clara el progreso del trabajo día a día y no necesita generar demasiados artefactos o una documentación extensa. Siendo también la metodología tomada para guiar el proceso de desarrollo del software de la versión 1.0 del CISD.

1.3.4 Frameworks

Es una estructura conceptual y tecnológica de soporte definido, generalmente con artefactos o módulos de *software* concretos, con base a la cual otro proyecto de *software* puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto. Los *frameworks* son herramientas de desarrollo que facilitan infraestructuras que ahorran horas de trabajo. (28)

Spring 3.2.0

Es un *framework* modular de código abierto para el desarrollo de aplicaciones Java, está dividido en siete módulos donde se destaca su flexibilidad y alto valor de configuración, se acopla a las aplicaciones sin tener que modificarlas para usar sus beneficios. Hace una clara división entre las capas de arquitecturas y está basada en interfaces para enmarcar las funcionalidades, disminuye el margen de errores, se obtiene una mayor limpieza y claridad en el código, reduce potencialmente el enlace entre los diferentes componentes de la aplicación. No necesita muchos recursos para su ejecución por lo que es descrito como liviano para la variedad de servicios que brinda. Propone un módulo que soporta los *frameworks* ORM⁴ más populares del mercado dentro de los cuales se encuentra Hibernate. (29)

Debido a los elementos evidenciados anteriormente, el conocimiento por parte del equipo de desarrollo, su cómoda configuración para interactuar con Hibernate y su factibilidad como punto intermedio entre la capa de presentación y la de acceso a datos del sistema, se determinó su uso para aportar al cumplimiento del objetivo trazado en esta investigación.

Hibernate 3.2.5

Es un *framework* desarrollado en Java, basado en tecnologías libres donde su labor está referida al ORM, que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los *beans*⁵ de las entidades. También facilita la consulta y recuperación de datos, esto reduce el tiempo de

⁴ ORM: Object-Relational Mapping (Mapeo Objeto-Relacional).

⁵ Bean: Componente reutilizable que puede ser manipulado por una herramienta de programación en lenguaje java.

desarrollo que traería hacerlo manual en SQL⁶ y JDBC⁷.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientada a objetos) y el usado en las bases de datos (modelo relacional). Esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. (30)

Se decidió utilizar este *framework* debido a las características antes mencionadas, donde facilita el mapeo de tablas entre la base de datos relacional del sistema DalasQ y el modelo de objetos de la aplicación. Hibernate destaca su alta compatibilidad y facilidad de integración con *Spring*. Se determina la versión 3.2.5 pues es la utilizada en la actual versión del CISD.

Java Server Faces (JSF) 2.1

Es el marco estándar de interfaz de usuario orientada a componentes (UI) para la plataforma JEE, incluida en esta, por lo que se pueden crear aplicaciones que utilicen JSF sin añadir bibliotecas adicionales en el proyecto. Funciona igual de bien como *framework* web independiente, capaz de conectar con otros *frameworks* como Spring. (31)

JSF incluye varios elementos como:

- Un conjunto de API⁸ para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un conjunto por defecto de componentes para la interfaz de usuario, incluyendo los elementos estándares para representar un formulario.

Una de las grandes ventajas de la tecnología JSF es que ofrece una clara separación entre el comportamiento y la presentación, esto permite que cada miembro del equipo de desarrollo se enfoque en su parte del proceso de desarrollo y proporcionar un modelo de programación sencillo para enlazar todas las piezas.

⁶ SQL: Lenguaje de consulta estructurado de acceso a bases de datos relacionales.

⁷ JDBC: Acceso a bases de datos desde aplicaciones Java.

⁸ API: Application Programming Interface (Interfaz de Programación de Aplicaciones).

PrimeFaces 5.1

Es un *framework* de código abierto usado para componentes visuales. Ayuda a ver el proyecto desde el punto de vista de un desarrollador de aplicaciones reconociendo fácilmente las características que faltan y rápidamente corregir los errores. Según el sitio DevRates es el segundo lugar en su lista de *frameworks* con 8.8 puntos. (32)

Es una biblioteca ligera, donde todas las decisiones tomadas por sus desarrolladores se basan en mantener PrimeFaces lo más ligero posible. Por lo general, la adición de una solución de terceros podría traer una sobrecarga, sin embargo, este no es el caso con PrimeFaces. Es sólo un jar sin dependencias y nada que configurar. (33)

Las principales características:

- Soporte nativo de Ajax⁹.
- *Kit* para crear aplicaciones web para móviles.
- Es compatible con otras librerías de componentes, como *JBoss* y *RichFaces*.
- Uso de *javascript*¹⁰ no intrusivo.
- Es un proyecto *Open Source*, activo y bastante estable entre versiones.
- Incorpora jQuery como *framework* de javascript.

Posee aproximadamente 100 componentes bajo el principio de ocultar la complejidad y elevar su flexibilidad a los desarrolladores, un *Showcase*¹¹ con ejemplos de los componentes para su uso y varios paquetes para el desarrollo en tecnología móvil, una documentación en abundancia, además de un fórum o comunidad con especialistas que constantemente postea dando las actualizaciones y soluciones de los problemas que se puedan encontrar con su empleo. Otra de sus características más importantes que lo hacen un *framework* a consumir por el equipo de desarrollo es su integración con la tecnología JSF y Spring, *frameworks* ya seleccionados. Se define cambiar la versión actual 3.5 del CISD a 5.1 pues es política del proyecto DalasQ migrar sus aplicaciones web hacia JSF 2.1 y PrimeFaces 5.1.

⁹AJAX: *Asynchronous JavaScript And XML (JavaScript asíncrono y XML). Técnica de desarrollo web para crear aplicaciones interactivas.*

¹⁰JavaScript: *Es un poderoso lenguaje de programación interpretado utilizado básicamente en la web.*

¹¹Showcase: *Herramienta web para almacenar y mostrar elementos.*

1.3.5 Servidor web

Un servidor web es un programa que procesa una aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales con el cliente, generando o cediendo una respuesta en cualquier lenguaje. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. (34)

Apache Tomcat 8.0.20

Apache Tomcat es un servidor web con soporte de *servlets*¹² y JSP desarrollado bajo el proyecto *Jakarta* en la *Apache Software Foundation*. Su implementación es en el lenguaje de programación Java y es compatible con las API más recientes. Debido a su implementación en tecnologías libres puede ser usado en cualquiera de los sistemas operativos, solo necesitan tener instalado una máquina virtual de Java. Es uno de los servidores web más populares entre los desarrolladores (35). Este software está bajo la licencia Apache 2.0, que permite al usuario usarlo, distribuirlo y modificarlo sin importar su propósito.

Apache Tomcat 8.0.20 incluye a numerosas correcciones para problemas identificados en 8.0.18, así como una serie de otras mejoras y cambios. Los notables cambios desde 8.0.18 incluyen:

- Fijar una regresión desempeño en la implementación de nuevos recursos cuando se utilizan JAR firmados en una aplicación web.
- Corregir varios errores que podrían causar que los registros múltiples para eventos de escritura para una sola toma al usar *Servlet* 3.0 asíncrona. Por lo general, los efectos secundarios de estos múltiples registros serían excepciones que aparecen en los registros.
- Mejorar la fábrica de *beans* utilizado para recursos JNDI¹³. (36)

1.3.6 Herramienta CASE¹⁴

Se utilizan en la planeación y ejecución de proyectos que involucren sistemas de información, pues suelen inducir a sus usuarios a la correcta utilización de metodologías de desarrollo, que le ayudan a llegar con facilidad a los productos de software construidos. Todas las herramientas

¹² *Servlets*: Clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.

¹³ *JNDI*: Java Naming and Directory Interface (Interfaz de Nombrado y Directorio Java). Es una API de Java para servicios de directorio.

¹⁴ *CASE*: Computer Aided Software Engineering (Ingeniería de Software Asistida por Computadora).

CASE prestan soporte a un lenguaje de modelado para acompañar la metodología, la mayoría de ellas soportan UML, teniendo en cuenta la amplia aceptación de este lenguaje, el valor conceptual y visual que proporciona. (37)

Visual Paradigm For UML 8.0

Como herramienta CASE se define Visual Paradigm For UML, que sustenta Lenguaje Unificado de Modelado (UML). Es multiplataforma y facilita la interoperabilidad con otras aplicaciones. Creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de forma simple toda la documentación.

Otro de los beneficios que brinda la herramienta se debe a su de carácter libre lo que evita restricciones de uso. Además sostiene estándares y anotaciones de diversos lenguajes entre los que se encuentra Java y UML. Se acopla con varios entornos de desarrollo integrado entre los cuales se encuentra Eclipse.

1.3.7 Lenguaje de programación

Las máquinas en general, y las computadoras en particular, necesitan de un lenguaje propio para poder interpretar las instrucciones que se les dan y para que nosotros podamos controlar su comportamiento. Ese lenguaje que permite esta relación con las computadoras es el lenguaje de programación. (37)

Java

Java es uno de los lenguajes de programación más populares en la actualidad (38). Permite desarrollar y desplegar de un modo eficiente interesantes aplicaciones y servicios. Ofrece portabilidad de aplicaciones incluso entre los entornos informáticos más dispares.

Características

- Interpretado y compilado a la vez.
- Multiplataforma.
- Seguro.

- Distribuido.
- Orientado a objetos.
- Robusto.
- Multiproceso.

Se elige Java en su versión 7 como lenguaje de programación porque es orientado a objetos, el equipo de desarrollo posee experiencia en el trabajo con este lenguaje, es compatible con los distintos módulos del sistema DalasQ, además imprescindible para la integración con los diferentes *frameworks* propuestos para dar solución al problema de investigación.

1.3.8 Bibliotecas de programación

JasperReports 3.7.5

Está escrito completamente en Java y es capaz de utilizar los datos procedentes de cualquier tipo de fuente de datos y producir documentos de píxeles perfecto que se pueden ver, imprimir o exportar en una variedad de formatos de documentos incluyendo HTML, PDF, XLS, ODT y DOC. (39)

Para la confección de estos reportes se utiliza la herramienta IReport en su versión 3.7.5. Es una herramienta visual de código abierto que sirve para generar ficheros XML con el diseño del informe y es utilizado por JasperReports y JasperReports Server. Provee a los usuarios una interfaz visual para construir reportes y crea diseños muy sofisticados que contienen gráficos, tablas, imágenes y subreporte. (40)

JavaMail 1.4.7

JavaMail es una biblioteca de programación *Open Source* desarrollada en Java para construir aplicaciones de correo y mensajería, o ser usada para ese propósito desde cualquier aplicación de lenguaje java. El envío y recepción de correos los realiza a través de los protocolos SMTP (*Simple Mail Transfer Protocol*), POP3 (*Post Office Protocol*) versión tres e IMAP (*Internet Message Access Protocol*), además implementa la seguridad en la conexión con servidores de correo TLS (*Transfer Layer Security*), SSL (*Secure Sockets Layer*) y la autenticación con usuario y contraseña.

jQuery 2.2

jQuery es una biblioteca JavaScript rápida, pequeña y rica en funciones. Hace tareas como el recorrido de HTML, 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 desarrolladores escriben JavaScript. (42)

1.3.9 Entorno de desarrollo integrado

Eclipse Luna

El Entorno Integrado de Desarrollo (IDE, del inglés *Integrated Development Environment*) seleccionado para el desarrollo de la solución propuesta es Eclipse en su versión 4.4.1 nombrada Luna. Esta versión incluye soporte oficial para Java 8 en las herramientas de desarrollo Java, *plugin*¹⁵ de herramientas de desarrollo, *Eclipse Communication Framework*, integración Maven, Xtend, plataforma de herramientas web, y analizador de memoria. (41)

El compilador de Eclipse incluye mejoras en el idioma, búsqueda y refactorización, asistencia rápida y nuevas opciones para el formateador de lambdas. Es soportada por los principales sistemas operativos tales como Windows, Mac y Linux.

1.3.10 Sistema gestor de bases de datos (SGBD)

Un Sistema Gestor de Bases de Datos (*DataBase Management System*) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipularlos, garantizando su seguridad e integridad.

PostgreSQL 9.2.4

Es un potente sistema de gestión de bases de datos objeto-relacional (O-RDBMS), multiusuario, centralizado y de propósito general, que ha sido desarrollado de diversas maneras desde 1977, está liberado bajo la licencia BSD (*Berkeley Software Distribution*). Es ampliamente considerado como uno de los sistemas gestores de bases de datos de código abierto más avanzado del

¹⁵ *Plugin: Es un programa informático que añade una funcionalidad adicional o una nueva característica al software.*

mundo y uno de los más populares. (42)

Características Generales: (43)

- Integridad referencial.
- Replicación asincrónica/sincrónica.
- Copias de seguridad en caliente.
- Unicode.
- Juegos de caracteres internacionales.
- Regionalización por columna.
- Múltiples métodos de autenticación.
- Acceso encriptado vía SSL.
- Completa documentación.

Se selecciona este SGBD, porque la solución propuesta surge a partir de la necesidad de monitorizar un sistema que se encuentra en su fase de despliegue, a través de los valores existentes en dicha base de dato relacional. Dicha base de datos actualmente se encuentra diseñada en *PostgreSQL*, por lo que la principal razón de su selección estuvo encaminada a mantener la compatibilidad entre sistemas.

Conclusiones parciales

- En este capítulo se realizó un estudio de los *software* existentes con funcionalidades similares a las requeridas por el producto a desarrollar, así como el análisis de los diferentes tipos de indicadores y tecnologías a utilizar con el fin de seleccionar las adecuadas para darle solución a la problemática planteada.
- Fue seleccionada SXP como metodología ágil de desarrollo, Visual Paradigm For UML como herramienta CASE para el modelado de diagramas y artefactos, se definió como lenguaje de programación Java en su versión 7 y como IDE Eclipse Luna, PostgreSQL como SGBD, bibliotecas de programación como JavaMail para el envío de correos electrónicos, JasperReports para generar reportes y jQuery para la construcción del *dashboard* en forma de mapa.
- Además se decidió utilizar diferentes *frameworks* tales como Hibernate, JSF, *Spring* y PrimeFaces. La mayoría de las herramientas y tecnologías seleccionadas son compatibles con la versión 1.0 del CISD siguiendo las políticas del proyecto DalasQ para las tecnologías web.

Capítulo 2: Propuesta de solución.

2.1 Introducción

En este capítulo se describe la propuesta del sistema que da solución a la problemática. Se destacan sus características esenciales y funcionamiento. Se especifican algunos artefactos generados, los requisitos funcionales y no funcionales, mostrando además las historias de usuarios y las tareas de ingenierías asociadas a las mismas.

2.2 Modelo de dominio

Dentro de las actividades más importantes definidas en la metodología SXP se encuentra la definición del Modelo de Dominio (MD). Explica los conceptos significativos en un dominio del problema; es el artefacto más importante a crear durante el análisis orientado a objetos. Es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases u objetos software con responsabilidades. (44) A continuación se presenta el MD para el sistema que se propone:

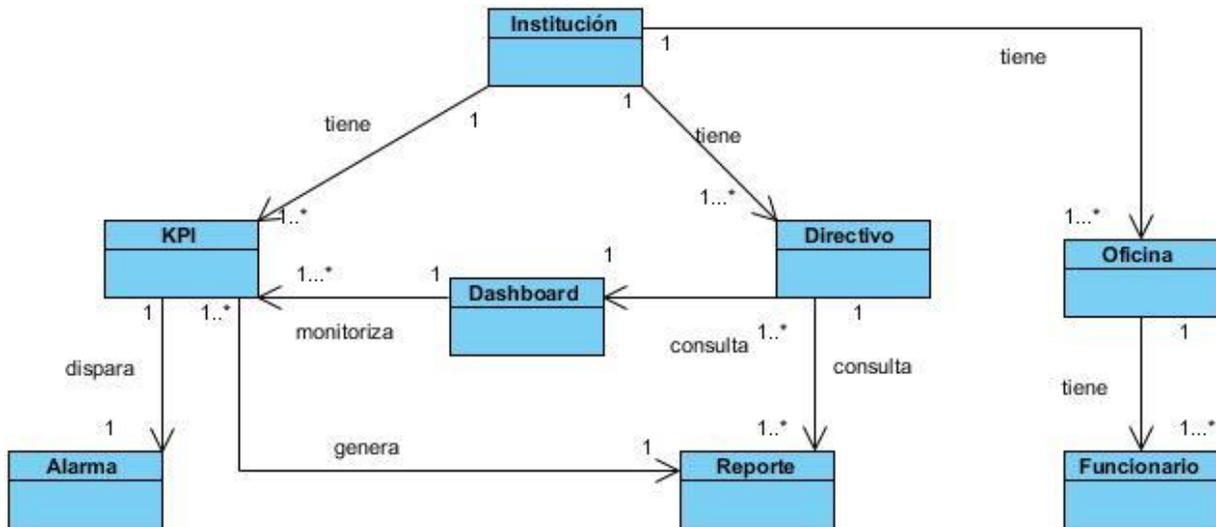


Figura 1: Modelo del dominio. (Elaboración Propia)

Descripción del MD:

- **Institución:** Empresa u organización que tiene Oficinas, Directivos y KPI.
- **Directivo:** Persona que pertenece a la dirección de la Institución. Esta puede consultar Reportes y el Dashboard, tomando decisiones a partir de estos.
- **Dashboard:** Ventana que consulta el Directivo con el estado de los KPI de la Institución.
- **Oficina:** Instalación de la institución que tiene un grupo de Funcionarios.
- **Funcionarios:** Persona encargada de brindar servicios a la población en las Instalaciones.
- **KPI:** Indicadores que tiene definida la Institución. Son monitorizados por el *Dashboard* para la consulta de los Directivos y desencadena una Alarma cuando alguno se encuentra fuera de rango. Además permite generar Reportes.
- **Alarma:** Notificación lanzada cuando un KPI se encuentra fuera de rango.
- **Reporte:** Archivos que se generan con la información de los KPI para ser consultadas por los Directivos.

2.3 Propuesta de solución

En el CISD se necesita monitorizar el estado de los KPI de los funcionarios, para la toma de decisiones basada en datos por parte de los directivos de la institución. Para responder a esta necesidad se toma la decisión de desarrollar una aplicación informática que contribuya a solucionar dicha realidad. La misma facilitará en gran medida la utilización de la información por parte de los directivos de la institución, permitiendo la toma de decisiones basada en datos.

La aplicación permitirá monitorizar la información de las diferentes instalaciones. Además se podrá visualizar globalmente el estado de la organización a través de un mapa interactivo. De los funcionarios se tienen los siguientes indicadores:

- Tiempo desfirmado (no autenticado).
- Tiempo ocioso.
- Tempo de servicio.

- Tiempo de *back-office*.
- Tiempo disponible.
- Tiempo no disponible.

Una vez accedido al sistema y estén configurados los valores mínimos y máximos de los KPI de los funcionarios de determinada oficina, se podrá visualizar información por cada una de las regiones del *dashboard* global o mapa, garantizando tener una mayor percepción de los objetivos. Una vez que algún indicador esté fuera del rango establecido, se envía una notificación automática mediante correo electrónico al supervisor de dicha oficina habiendo establecido previamente el correo electrónico del supervisor en el panel de alarmas. También brinda la posibilidad de generar mediante gráficas, las estadísticas correspondientes a los funcionarios para análisis de comportamiento y la toma de decisiones de los directivos. Además de generar reportes sobre la información de las distintas instalaciones. El sistema a desarrollar será multiplataforma.

2.4 Lista de reserva del producto

Se forma de los requerimientos técnicos y actualizaciones tecnológicas requeridas, específicamente los requisitos funcionales y no funcionales. Representa el primer artefacto que se define en la metodología SXP en la etapa de captura de requisitos.

Tabla 1: Lista de Reserva de Producto. Requisitos Funcionales.

Ítem ¹⁶	Descripción	Estimación (semanas)	Estimado por	Prioridad
01	Autenticar usuario.	1.0	Analista	Alta
02	Visualizar usuario.	0.4	Analista	Media
03	Representar árbol de ubicaciones de las Oficinas por niveles de división territorial.	1.6	Analista	Alta
04	Configurar los valores mínimos y máximos de los indicadores de las instalaciones en todo momento.	1.0	Analista	Alta
05	Representar las regiones (estados) mediante un <i>dashboard</i> según los	3.0	Analista	Alta

¹⁶ Ítem: (para este caso) numeración secuencial de las funcionalidades del sistema.

	valores pertinentes de los indicadores.			
06	Representar las Oficinas de la región seleccionada junto con los estados de sus indicadores.	2.5	Analista	Alta
07	Representar el estado de los indicadores de los Funcionarios de la oficina seleccionada.	2.0	Analista	Alta
08	Crear y modificar las alarmas de las instalaciones.	1.2	Analista	Alta
09	Enviar correo al supervisor de la instalación una vez disparado un indicador de rendimiento.	1.4	Analista	Alta
10	Visualizar estadísticas de los Funcionarios.	1.2	Analista	Alta
11	Generar reportes.	1.0	Analista	Alta

Tabla 2: Lista de reserva del producto. Requisitos no funcionales.

Ítem	Descripción
01	El ordenador donde se instale la solución debe poseer un mínimo de 2 GB ¹⁷ de memoria RAM ¹⁸ , un procesador de 2 GHz y 10 GB de almacenamiento.
02	El ordenador donde se instale la aplicación debe tener tarjeta de red.
03	Utilizar cualquier navegador web como Internet Explorer 8, Mozilla Firefox 16, Google Chrome 18, Safari 5.0, Opera 10 o versiones superiores de cada uno.
04	La aplicación debe poder ser instalado sobre las plataformas Windows y Linux.
05	La aplicación debe actualizar la información visual con una latencia de 2 segundos.

¹⁷ GB: Un gigabyte es una unidad de almacenamiento de información equivalente mil millones de bytes.

¹⁸ RAM: Random Access Memory (Memoria de Acceso Aleatorio), se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del software.

2.5 Historias de usuario

Son utilizadas para especificar las funcionalidades que brindará el sistema. Cada historia de usuario es una representación de un requerimiento de software escrito en una o dos frases utilizando el lenguaje común del usuario. Representan una forma rápida de administrar los requerimientos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Además son la base para realizar las pruebas de aceptación, así como la estimación y planificación del proyecto.

A continuación se presentan dos historias de usuario referentes a los requisitos funcionales de la aplicación enunciados previamente en la tabla 1 de la investigación, el resto se plasma en los anexos. ([Ver Anexos](#))

Tabla 3: HU_5 Representar las regiones mediante un dashboard según los valores pertinentes de los indicadores.

Historia de Usuario	
Número: HU_5	Nombre Historia de Usuario: <i>Representar las regiones mediante un dashboard según los valores pertinentes de los indicadores.</i>
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 3.0 semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 3.0 semanas
Descripción: <i>La presente historia de usuario tiene como objetivo representar las regiones mediante un dashboard según los valores pertinentes de los indicadores. Cada región se visualizará con un color que define el estado en que se encuentra la misma.</i>	
Observaciones: <i>Para poder representar las regiones se debe acceder a la opción Mapa en la interfaz Home.</i>	
Prototipo de interfaz: Dashboard Global	

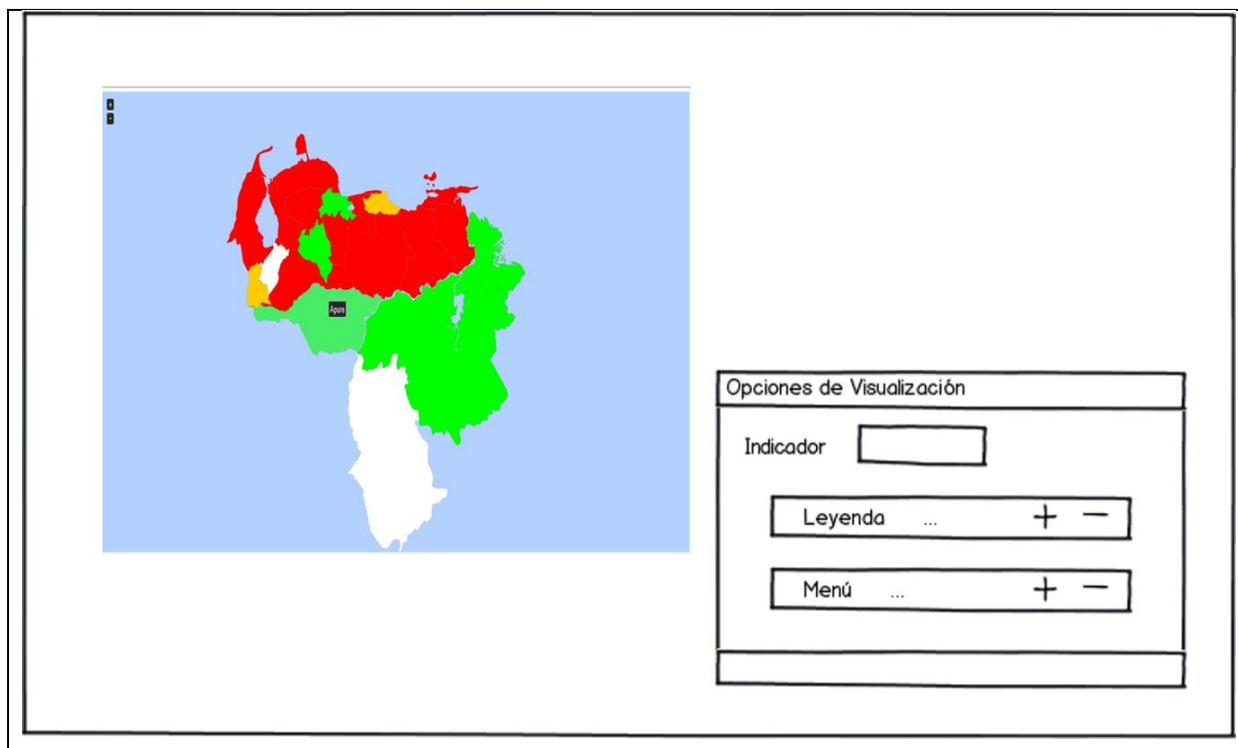


Tabla 4: HU_6 Representar las oficinas de la región seleccionada junto con los estados de sus indicadores.

Historia de Usuario	
Número: HU_6	Nombre Historia de Usuario: Representar las oficinas de la región seleccionada junto con los estados de sus indicadores.
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2.5 semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 2.5 semanas
Descripción: La presente historia de usuario tiene como objetivo representar las Oficinas de la región seleccionada mediante una gráfica (tabla) con los valores de los indicadores así como su estado (azul, verde, rojo).	
Observaciones: Para poder representar las oficinas, se debe presionar sobre la región que se desea obtener información.	
Prototipo de interfaz: Visualizar Región	

Oficinas											
	Nombre	Id	PA	PE	CA	PD	PS	PND	PDES	PB	PO
🔍											
🔍											
🔍											
🔍											
🔍											

2.6 Tareas de ingeniería

Las tareas de ingeniería se generan como parte de la metodología SXP, que integran las distintas historias de usuarios. A continuación se presentan las tareas correspondientes a las historias de usuario originadas anteriormente, el resto en los anexos. ([Ver Anexos](#))

Tabla 5: Tarea de ingeniería: 1.1

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU_1
Nombre Tarea: <i>Diseño e implementación de la funcionalidad autenticar usuario.</i>	
Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: 1.0 <i>semana</i>
Fecha Inicio: 20/01/2015	Fecha Fin: 27/01/2015
Programador Responsable: <i>Orson O. Duque Cortina</i>	
Descripción: <i>Diseñar e implementar la interfaz de acceso a la aplicación con los campos necesarios para realizar la operación (usuario y contraseña). Se introducen datos en los campos y se procesan con una serie de funcionalidades detrás de acuerdo a los valores ingresados.</i>	

Tabla 6: Tarea de ingeniería 2.1

Tarea de Ingeniería	
Número Tarea: 2.1	Número Historia de Usuario: HU_2
Nombre Tarea: <i>Visualización de usuario conectado.</i>	

Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: 0.4 <i>semanas</i>
Fecha Inicio: 28/01/2015	Fecha Fin: 31/01/2015
Programador Responsable: <i>Orson O. Duque Cortina.</i>	
Descripción: <i>Definir e implementar el lugar ideal para mostrar la información y la manera de hacerlo, solicitando la información de usuario a la clase manejadora de la acción.</i>	

2.7 Plan de *release*¹⁹

El plan de *release* es una estimación del tiempo total que debe tomar el desarrollo del proyecto. Se recogen las iteraciones a realizar con sus características, además del orden de las HU con su planificación estimada para ser implementadas. A continuación se presenta el plan de *release* de la aplicación:

Tabla 7: Plan de release.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 1	Se implementa el acceso de usuario y la generación del árbol de ubicaciones, así como se configuran los intervalos permitidos de los indicadores.	HU_1, HU_2, HU_3, HU_4	4.3 semanas
Iteración 4	Se generan los <i>dashboards</i> de las regiones, oficinas de las regiones seleccionadas y funcionarios de las oficinas seleccionadas con los datos de los indicadores correspondientes y se actualizan una vez que presenten cambios en la base de datos.	HU_5, HU_6, HU_7	7.5 semanas
Iteración 3	Se establecen las configuraciones de los indicadores y el sistema de alarmas a los supervisores, además de brindar las	HU_11, HU_12, HU_13, HU_14	5.1 semanas

¹⁹ *Release: Liberación*

	estadísticas de las oficinas y los reportes generados.		
			Total: 17.2 semanas

2.8 Arquitectura de software (AS)

El diseño arquitectónico es la primera etapa en el proceso de diseño y representa un vínculo fundamental entre el proceso de ingeniería de diseño y requisitos. El proceso de diseño arquitectónico se refiere a establecer un marco estructural básico que identifica los componentes principales de un sistema y la comunicación entre estos componentes. (45) En la presente investigación para dar solución a la problemática se define utilizar la arquitectura N-Capas, específicamente tres, conocida de forma común como arquitectura de tres capas.

En la arquitectura tres capas, el objetivo principal es separar los diferentes aspectos del desarrollo en tres capas: Interfaz de Usuario (IU), Lógica de Negocio (LN) y Acceso a Datos (AD). Se puede llevar a cabo en varios niveles y en caso de que ocurra algún cambio solo hay que dirigirse a la capa en cuestión. Ahorra recursos y posibilita un mayor entendimiento, además de ser flexible. Cada capa es una organización jerárquica que le propone servicios a su capa superior y se sirve de las prestaciones que le brinda la inferior.



Figura 2: Arquitectura N-Capas (Elaboración Propia).

La Capa de Interfaz de Usuario es la que presenta el sistema al usuario, le comunica y captura la información. Esta capa se comunica únicamente con la capa de negocio y posee la característica de ser el puente de comunicación entre el usuario y la aplicación. La navegabilidad, el formateo de la información y la validación de datos, son cuestiones que se tratan en esta capa. En la aplicación confeccionada, en esta capa se hace uso de la tecnología JSF y PrimeFaces para los componentes visuales. JasperReports para la generación de reportes en formato PDF.

Además se utiliza el framework de javascript jQuery para la construcción del mapa interactivo como *dashboard*.

La Capa de la Lógica del Negocio es donde residen las clases controladoras y de servicios, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Además se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, así como con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar o recuperar información de él. (40)

En esta capa se utiliza el *framework* Spring para el manejo entre clases y ganar en claridad entre el modelo, el negocio y las vistas.

La Capa de Acceso a Datos permite vincular objetos de las bases de datos relacionales a objetos de lenguaje de programación mediante la propiedad de persistencia de objetos, esto facilita el acceso a datos desde la capa superior. Se hace uso de los DAO (*Data Access Object*) para abstraer el acceso a los datos y el *framework* Hibernate para el mapeo de las tablas de la base de datos y realizar consultas a la misma. (46)

2.9 Modelo de diseño

Debido a que SXP toma algunas buenas prácticas de XP, y dicha metodología define un término llamado metáfora, lo cual según Martin Fowler es una historia compartida que describe como debería funcionar el sistema y define que la práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema.

2.9.1 Patrones de diseño

Un patrón es una solución a un problema en un contexto particular recurrente, lo que hace la solución relevante a otras situaciones, enseña, permite entender cómo adaptarlo a la variante particular del problema donde se quiere aplicar. Los patrones facilitan la reutilización de diseños y arquitecturas de software que han tenido éxito. Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, y describe la esencia de la solución a ese problema, de tal modo que pueda utilizarse esta solución un millón de veces más, sin siquiera hacerlo de la misma manera dos veces. (48)

Los patrones de diseño empleados en el desarrollo de la aplicación son los Patrones Generales

de Software para Asignar Responsabilidades (GRASP por sus siglas en inglés), así como el Patrón Core J2EE DAO.

Patrones GRASP

Bajo acoplamiento: Se utiliza con el objetivo de tener las clases lo menos ligadas o dependientes entre sí. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases. Ejemplo de ello lo constituyen las clases que representan las entidades del modelo en relación con las controladoras del sistema.

Alta cohesión: El problema fundamental que soluciona es cómo mantener manejable la complejidad, por lo que se deben asignar las responsabilidades de manera que la información que almacena sea coherente y relacionada con la clase. Al realizar un cambio en una clase cohesionada todo lo que puede verse afectado y se necesita controlar estará a la vista en la misma área, por lo que facilita el trabajo y entendimiento. Ejemplo de este patrón se puede ver en la clase MapaBean.

Experto: Se encarga de asignar la responsabilidad al experto en la información. Permite conservar el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, lo cual provee además un bajo nivel de acoplamiento.[32] El experto es la clase que cuenta con la información necesaria para cumplir la responsabilidad. Este patrón fue empleado mayormente en las clases controladoras, ejemplo de ello es MapaBean, la cual tiene toda la información y funcionalidades necesarias para crear las gráficas estadísticas y los reportes del sistema.

Controlador: Este patrón se encarga de asignar la responsabilidad a las clases controladoras facultadas de realizar los eventos del sistema. En la capa de lógica de negocio se encuentran todas las clases controladoras, donde se encargarán de atender todas las peticiones que vengan de la capa de presentación. Su uso permite una mayor capacidad de reutilización, ejemplo de ello es la clase MapaBean.

Creador: En un sistema orientado a objetos, la actividad más frecuente es la creación de objetos por lo que se hace necesario controlar la asignación de responsabilidades a través de un principio general. Una vez dada la responsabilidad a una clase esta será la encargada de crear instancias de la otra, dándole visibilidad para su manejo o registro de objetos. Ejemplo de ello el uso del

sessionFactory en la clase AbstractDAOImpl.

Patrón Core J2EE

DAO: Este patrón implementa el mecanismo de acceso requerido para trabajar con la fuente de datos. El componente de negocio utiliza la interfaz más simple expuesta por el DAO para sus clientes. Este patrón oculta completamente los detalles de implementación del origen de datos de sus clientes. Este modelo permite adaptarse a los diferentes esquemas de almacenamiento sin afectar a sus clientes o componentes de negocio. Esencialmente actúa como un adaptador entre el componente y la fuente de datos. Este patrón se evidencia en las clases AbstractDAO y sus relaciones con el modelo. (49)

Conclusiones parciales

- ✓ En este capítulo se obtuvo un mayor entendimiento de la aplicación a desarrollar, mediante la lista de reserva del producto se han definido 11 requisitos funcionales y 7 no funcionales del sistema, que ha permitido identificar las funcionalidades de la aplicación.
- ✓ Fue representado mediante un diagrama de conceptos el MD de la aplicación, así como las historias de usuarios con las tareas de ingeniería.

Capítulo 3: Implementación y pruebas al sistema

3.1 Introducción

En el presente capítulo se reflejan las actividades realizadas como parte de la implementación y pruebas de la solución propuesta. Se define el diagrama de clases de la solución propuesta. Se compone además por el estilo de código definido, los casos de prueba (CP) del sistema, las pruebas de rendimiento y los resultados arrojados.

3.2 Diagrama de clases

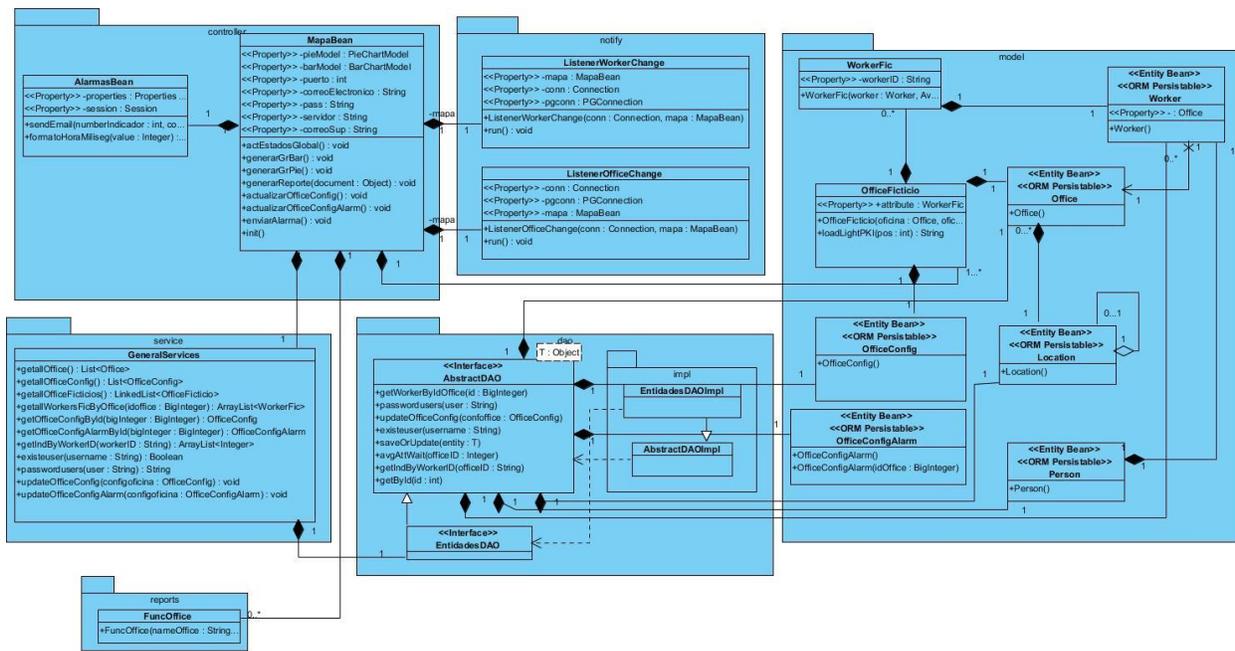


Figura 3: Diagrama de Clases (Elaboración Propia).

Descripción de las clases principales del diagrama de clases:

- **MapaBean:** Clase principal que se encarga de responder las peticiones referentes a la interfaz mapa o *dashboard* global.
- **GeneralServices:** Clase servicio que se relaciona con las entidades DAO permitiendo a la controladora acceder a información basada en las entidades del modelo.
- **AlarmaBean:** Clase encargada de enviar notificaciones.
- **FuncOffice:** Clase que representa los datos a mostrar por el reporte.
- **Paquete DAO:** Clases que implementan el patrón DAO utilizando la variante AbstractDAO.
- **WorkerFic:** Clase que utiliza diferentes datos relacionados con la clase Worker y el

estado de los KPI de dicho Worker.

- **OfficeFicticioso:** Clase que utiliza diferentes datos relacionados con la Clase Office, OfficeConfig y OfficeConfigAlarm.
- **ListenerWorkerChange:** Clase de tipo hilo que se encarga de recibir las notificaciones enviadas por el servidor de base de datos relacionadas con la entidad Worker.
- **ListenerOfficeChange:** Clase de tipo hilo que se encarga de recibir las notificaciones enviadas por el servidor de base de datos relacionadas con la entidad Office.

3.3 Estándar de código

Un estándar de codificación comprende todos los aspectos de la generación de código. Si bien los desarrolladores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. (50)

En particular el de Java, tiene reglas como: las clases inician con mayúsculas, los atributos y métodos inician con minúsculas, las constantes son todas en mayúsculas. También menciona la forma de abrir y cerrar bloques de código y la forma de poner comentarios, a continuación se explica el estándar de código utilizado.

El estándar de código definido para la solución propuesta está condicionado por el estándar previamente establecido en la anterior versión del CISD. (46)

Organización de ficheros: Las clases en Java se agrupan en paquetes, estos paquetes se deben organizar de manera jerárquica, de forma que todo el código desarrollado tendrá que estar incluido dentro del paquete “uci.dalasq”. Dentro del paquete principal las clases se organizarán en subpaquetes en función del código. Por ejemplo, si nos referimos al modelo, las clases del modelo estarán dentro del paquete “uci.dalasq.modelo”.

Nomenclatura de las clases: Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará

UpperCamelCase. Con solo leerlo se reconoce el propósito de la misma, pues cada palabra comienza con letra mayúscula. Ejemplo: `class Office {}` y `class MapaBean {}`.

Nomenclatura de los métodos: Los métodos deben comenzar en minúsculas. Cuando el método esté compuesto por varias palabras cada una de ellas tendrá la primera letra en mayúsculas exceptuando la palabra inicial del nombre del método, conocido como notación lowerCamelCase. Ejemplo: `public void actEstadosGlobal()` y `public void actualizarDatosSegunOfficeConfig()`.

Nomenclatura de las variables: Las variables al igual que los métodos siguen la misma nomenclatura la palabra inicial en minúscula y el resto en mayúscula. Ejemplo: `private String oficinaSeleccionada` y `private List<String> oficinasSeleccionadasAlarma`.

Nomenclatura de los comentarios: Los comentarios se utilizan para describir el código, en ellos se incluye información relacionada con la implementación, tales como descripción de la función de variables locales, las fases de ejecución del método, entre otros aspectos. Se emplean tres tipos de comentarios de implementación:

- Comentarios de bloque: Permiten la descripción de ficheros, clases, bloques, estructuras de datos y algoritmos. Ejemplo:

```
/*  
Esto es un comentario de bloque  
*/
```

- Comentarios de línea: Son comentarios cortos localizados en una sola línea y tabulados al mismo nivel que el código que describen. Ejemplos:

```
/* Esto es un comentario de línea */  
// Esto es otro comentario de línea
```

- Comentario a final de línea: Comentario situado al final de una sentencia de código y en la misma línea.

```
contador++; // Incrementamos el contador  
contador++; /* Incrementamos el contador */
```

Nomenclatura de código CSS²⁰: Los códigos CSS deben seguir la terminología:

```
estilo{  
    propiedad: valor;  
}
```

Los estilos deben estar seguidos con una llave abierta en la misma línea y en letra minúscula. Luego de las propiedades debe terminar con una llave cerrada en la próxima línea con la misma sangría del estilo. Ejemplo:

```
.mystyle{  
    border: 0px;  
    background: #F2F5F7;  
}
```

3.4 Modelo de despliegue

El equipo de desarrollo considera necesario ilustrar la forma en que quedará distribuida la aplicación para su despliegue. Aunque la metodología SXP no tiene concebida dentro de sus artefactos la generación del modelo de despliegue de la aplicación, se define utilizarlo para dar respuesta a una de las peticiones del cliente con el objetivo de mejorar su entendimiento. En la figura se muestra la disposición física que tendrá la aplicación una vez se ponga en explotación.



Figura 4: Diagrama de Despliegue (Elaboración Propia).

²⁰ CSS: Hojas de estilo en cascada.

Descripción del modelo de despliegue: La estación de trabajo es un ordenador cliente que mediante un navegador web, usando el protocolo HTTP²¹ se conecta al servidor web donde reside la aplicación, este a su vez se comunica mediante JDBC con el servidor de base de datos donde residen los datos almacenados del sistema DalasQ. Se integra una impresora a la estación de trabajo de manera opcional para poder completar el funcionamiento de la aplicación. La conexión entre la estación de trabajo y la impresora es mediante cableado USB²² pero esto depende del tipo de impresora que se use. Además se establece la conexión entre el servidor de correo electrónico y el servidor web de la aplicación mediante el protocolo SMTP, a través del cual se realizará el envío de correo electrónico a los supervisores de las instalaciones.

3.5 Pruebas

Las pruebas de software es el proceso de análisis de un elemento de software para detectar las diferencias entre las condiciones existentes y necesarios conocidos como errores (*bugs*) para evaluar las características del elemento software (51). La metodología definida propone para verificar el correcto funcionamiento del software, la realización de pruebas de caja negra, específicamente casos de prueba de aceptación.

3.5.1 Casos de prueba

El propósito de un caso de prueba es especificar una forma de probar el sistema, incluyendo las entradas con las que se prueba, los resultados esperados y las condiciones bajo las que ha de probarse. Los casos de prueba son un producto de desarrollo de software, que ayudan a validar y verificar las expectativas de los *stakeholders*²³.

La técnica de prueba partición equivalente, la cual se utiliza en el presente trabajo, divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. En otras palabras, este método intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia. De tal modo que se pueda asumir razonablemente que una prueba realizada con un valor representativo de cada clase es equivalente, a una prueba realizada con cualquier otro valor de dicha clase.

²¹ HTTP: protocolo de transferencia de hipertexto usado en cada transacción de la World Wide Web.

²² USB: estándar industrial que define los cables y conectores usados para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos electrónicos.

²³Stakeholders: Clientes

Para comprobar si el software funciona correctamente y satisface los requerimientos del cliente se confeccionaron casos de prueba. A continuación se incluyen los CP de las HU_1 y la HU_3.

Tabla 8: Caso de prueba de aceptación: HU_1_P1.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU_1_P1	Nombre Historia de Usuario: Autenticar usuario.
Nombre de la persona que realiza la prueba: Orson O. Duque Cortina	
Descripción de la Prueba: Se ingresan los datos de acceso a la aplicación. (Acceso exitoso)	
Condiciones de Ejecución: Los datos ingresados del usuario deben tener acceso en la aplicación.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none">1. Ingresar usuario.2. Ingresar contraseña.3. Presionar en el botón "Aceptar".	
Resultado Esperado: La aplicación muestra una notificación de bienvenida con el nombre del usuario y otra notificación indicando que se está generando el entorno de la aplicación.	
Evaluación de la Prueba: Satisfactoria.	

Tabla 9: Caso de prueba de aceptación: HU_1_P2.

Caso de Prueba de Aceptación	
Código Caso de Prueba: HU_1_P2	Nombre Historia de Usuario: Autenticar usuario.
Nombre de la persona que realiza la prueba: Orson O. Duque Cortina	
Descripción de la Prueba: Se ingresan los datos de acceso a la aplicación. (Acceso fallido)	
Condiciones de Ejecución: -	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none">1. Ingresar usuario incorrecto o dejar el campo vacío.2. Ingresar contraseña.3. Presionar en el botón "Aceptar".	
Resultado Esperado: La aplicación muestra una alerta de error de inicio de sesión	

indicando que el usuario es incorrecto.

Evaluación de la Prueba: *Satisfactoria.*

Tabla 10: Caso de prueba de aceptación: HU_3_P1.

Caso de Prueba de Aceptación	
Código Caso de Prueba: <i>HU_3_P1</i>	Nombre Historia de Usuario: <i>Representar árbol de ubicaciones de las instalaciones por niveles de división territorial.</i>
Nombre de la persona que realiza la prueba: <i>Orson O. Duque Cortina</i>	
Descripción de la Prueba: <i>Verificar si se muestra dentro de la aplicación el árbol jerárquico de localización de las instalaciones.</i>	
Condiciones de Ejecución: <i>El usuario debe haberse autenticado previamente.</i>	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. <i>Verificar si es visible el árbol de ubicaciones.</i> 2. <i>Desplegar los niveles.</i> 3. <i>Verificar que se visualizan y es correcto el orden jerárquico de las instalaciones.</i> 	
Resultado Esperado: <i>La aplicación muestra en el borde izquierdo un panel que contiene el árbol de localizaciones distribuido de manera jerárquica atendiendo la división territorial.</i>	
Evaluación de la Prueba: <i>Satisfactoria.</i>	

3.5.2 Resultados obtenidos

Se realizaron dos iteraciones de las pruebas de aceptación con el cliente para conocer el grado de satisfacción hacia el producto. Durante la primera iteración se encontraron doce No Conformidades (NC), de estas, 4 eran de importancia Baja, relacionadas con la redacción en la aplicación y 8 de importancia Alta, relacionadas con el diseño de la aplicación. Todas las NC de redacción en la primera iteración fueron resueltas por el equipo de desarrollo así como 4 relacionadas con el diseño de la aplicación. Se realizó una segunda iteración, en la que no se descubrieron NC, donde fueron resueltas las restantes NC relacionadas con el diseño de la aplicación. Luego fue realizada una tercera iteración donde no se descubrieron nuevas NC evaluándose de esta manera todos los CP como “Satisfactorios”.

3.6 Tratamiento de errores

Las aplicaciones informáticas deben estar preparadas para enfrentarse a las entradas erróneas por parte del usuario y errores internos por un comportamiento indebido de un componente en ejecución. En el caso de la aplicación resultante de la investigación, dicho tratamiento se realizó mediante funciones JavaScript, funciones Java y excepciones, mostrando al usuario notificaciones, indicando donde se encuentran los datos incorrectos y evita que sean enviados al servidor mientras no sean corregidos. Haciendo uso del tratamiento de errores se pueden evitar escrituras incorrectas, tipos de datos erróneos y direcciones de correos incorrectos.

Conclusiones parciales

- ✓ Se incluyeron y explicaron los artefactos generados por la metodología SXP para la implementación, así como las pruebas realizadas.
- ✓ Además fue necesario definir un modelo de despliegue para una mejor comprensión por parte del cliente, así como el diagrama de clases.
- ✓ Se explicó el estándar de código utilizado para la implementación de la solución.

Conclusiones

Terminada la investigación y luego del análisis de los resultados obtenidos se arriban a las siguientes conclusiones:

- ✓ El análisis de sistemas de monitorización demostró que a nivel internacional son en su mayoría privativas y las que se utilizan en Cuba están desarrolladas para un propósito en específico o se dificulta la construcción de un *dashboard* en forma de mapa. Evidenciándose la necesidad de actualizar el CISD.
- ✓ Se desarrolló una aplicación web que permitió la monitorización de los indicadores claves de rendimiento de los funcionarios en el CISD para realizar la toma de decisiones basada en datos.
- ✓ Se validó la aplicación mediante las pruebas de aceptación.

Recomendaciones

1. Crear un almacén de datos que permita el análisis de datos a un nivel superior.
2. Crear un mecanismo que permita añadir un mapa de la institución que utilice la propuesta de solución.

Referencias Bibliográficas

1. **Ministerio de Comunicaciones.** Resumen de las bases y prioridades para el perfeccionamiento de la informatización de la sociedad en Cuba. 1er. Taller Nacional de Informatización y Ciberseguridad. *Mincom.* [En línea] 19 de Febrero de 2015. [Citado el: 25 de Febrero de 2015.] <http://www.mincom.gob.cu/?q=node/802>.
2. **Tuck, A., Pyne, Sandra y Ashby, Michael F.** *Oxford dictionary of computing for learners of English.*
3. **Laplante, Philip A.** *Real-Time Systems Design and Analysis.* s.l. : IEEE Press, 1992.
4. **Imparato, Carlos y Shokida, Andrés.** UK. *Universidad Kennedy.* [En línea] 2011. [Citado el: 16 de enero de 2015.] <https://www.kennedy.edu.ar/DocsEsc03/An%C3%A1lisis%20y%20Dise%C3%B1o%20de%20Sistemas%20en%20Tiempo%20Real/An%C3%A1lisis%20y%20Dise%C3%B1o%20de%20Sistemas%20en%20Tiempo%20Real.pdf>.
5. **Business Dictionary.** BusinessDictionary.com. [En línea] 2015. [Citado el: 16 de enero de 2015.] <http://www.businessdictionary.com/definition/monitoring.html>.
6. **Oxford University.** Oxford Dictionaries. [En línea] 2015. [Citado el: 27 de Enero de 2015.] <http://www.oxforddictionaries.com/definition/english/monitor>.
7. **Kaplan, Robert S. y Norton, David P.** *Cómo utilizar el Cuadro de Mando Integral 2ed.* s.l. : Harvard Business Press, 2015.
8. **Dávila, A.** *Nuevas herramientas de control: El Cuadro de Mando Integral.* 1999.
9. **Few, Stephen.** Dashboard Confusion. *Perceptual Edge.* [En línea] 20 de Marzo de 2004. [Citado el: 17 de Febrero de 2015.] http://www.perceptualedge.com/articles/ie/dashboard_confusion.pdf.
10. **DZ, M.P.** *Dashboard Zone.* 2011.
11. **Aguilar, Anuor.** ¿Qué es un Dashboard del Negocio? *Marketing Point.* [En línea] marzo de 2012. [Citado el: 17 de Febrero de 2015.] <http://anuor.blogspot.com/2012/03/que-es-un-dashboard-del-negocio.html>.
12. **Rodríguez, Daniel y Rovira, Pere.** Slideshare. *Cuadros de Mando.* [En línea] 12 de Marzo de 2010. [Citado el: 8 de Marzo de 2015.] <http://es.slideshare.net/pererovira/cuadros-de-mando>.
13. **Imetrix.** Imetrix,. *Imetrix Media Analytics.* [En línea] [Citado el: 20 de Octubre de 2014.] <http://www.imetrix.biz/es/indicadores-claves-desempeno.html>.

14. **Pin, Enrique y Delgado Marrero, Yoalis María.** Indicadores, una herramienta para medir la eficiencia en el uso de las tecnologías de la información y las comunicaciones en los centros de educación superior en Cuba. *GestioPolis.com*. [En línea] 2006. [Citado el: 13 de Diciembre de 2014.] <http://gestiopolis.com/canales6/ger/tecnologia-informacion-comunicaciones.htm>..
15. **Veritier, María Gisela.** FACTIBILIDAD DE DISEÑO E IMPLANTACIÓN DE UN CUADRO DE MANDO INTEGRAL EN UNA SOCIEDAD DEL ESTADO. [En línea] 2009. [Citado el: 13 de Diciembre de 2014.] <http://www.eumed.net/libros-gratis/2009a/479/index.htm>.
16. **Empresamía.** La Solución para el Emprendimiento y el Fortalecimiento Empresarial. ¿Qué es un indicador financiero y para qué sirve? [En línea] 2013. [Citado el: 13 de Diciembre de 2014.] <http://empresamia.com/debe-saber/intem/915-que-es-un-indicador-financiero-y-para-que-sirve>.
17. **Restrepo de Ocampo, I.S., Estrada Mejía, S. y Ramírez Aristizabal, C.** *Modelo de Gestión de Indicadores para una Empresa de Ventas de Vehículos*. Colombia : s.n., 2007.
18. **Cabrera, Liliana.** *Planeación, Estrategias y Táctica*. 2008.
19. **Rubino, Marta Teresa.** Desarrollo de un Cuadro de Mando Integral aplicable a Bibliotecas: descripción de las etapas para su creación y diseño. [En línea] 2007. [Citado el: 14 de Diciembre de 2014.] <http://ciberconta.unizar.es/leccion/BSCbibliotecas.html>.
20. **IBM Corporation.** IBM. *IBM Corporation Web site*. [En línea] 2014. [Citado el: 08 de Marzo de 2015.] <http://www-03.ibm.com/software/products/es/cognos-real-time-monitoring>.
21. **Pentaho.** Pentaho . *Pentaho Community*. [En línea] 2013. [Citado el: 26 de Septiembre de 2014.] <http://community.pentaho.com>.
22. **Poole, Damon B.** Do It Yourself Agile. [En línea] 29 de Septiembre de 2009. [Citado el: 06 de Marzo de 2015.] <http://www.accurev.com/whitepaper/pdf/Do-It-Yourself-Agile.pdf>.
23. **Penadés, M.C. y Letelier, Patricio.** Metodologías ágiles para el desarrollo de software: Xtreme Programming (XP). *Valencia: Universidad Politécnica de Valencia*. [En línea] 2006. [Citado el: 01 de Marzo de 2015.] <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
24. **Mendoza Sanchez, María A.** Metodologías de Desarrollo del Software. [En línea] 07 de Junio de 2004. [Citado el: 01 de Marzo de 2015.] <http://www.willydev.net/descargas/cualmetodologia.pdf>.
25. **INTECO.** *Metodologías y Ciclos de Vida*. 2009.
26. **Peñalver Romero, Gladys Marsi, García de la Puente, Sergio Jesús y Meneses Abad, Abel** . SXP, metodología de desarrollo de software. [En línea] 14 de Noviembre de 2011. [Citado el: 22

- de Enero de 2015.] <http://publicaciones.uci.cu/index.php/SC/article/viewFile/429/482>.
27. **Santana Leiva, Yoani.** *Módulo de Servicios Web para dotProject.DOT-WS*. La Habana : Universidad de las Ciencias Informáticas, 2010.
28. **Rescalvo, E.C.** Framework de Desarrollo de Código Abierto. *Utilización de framework de desarrollo de Código Abierto en el mundo empresarial*. [En línea] 2008. [Citado el: 14 de Diciembre de 2014.] <http://malaga08.opensourceworldconference.com/PAPERS/DIA22/SALA%204/CASAS235.PDF> :p.19.
29. **Johnson, Rod, y otros.** Spring. *Spring Framework Reference Documentation*. [En línea] 2014. [Citado el: 18 de Enero de 2015.] <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/>.
30. **Hibernate.** Hibernate.org. [En línea] 2014. [Citado el: 15 de Diciembre de 2014.] <http://hibernate.org/orm/>.
31. **JavaServer Faces.** JavaServer Faces.org. [En línea] 2014. [Citado el: 13 de Diciembre de 2014.] <http://www.java-serverfaces.org/>.
32. **DevRates.** DevRates. [En línea] 2015. [Citado el: 22 de Enero de 2015.] <http://devrates.com/stats/index>.
33. **Prime Technology.** Primefaces. [En línea] 2014. [Citado el: 17 de Diciembre de 2014.] <http://www.primefaces.org/whyprimefaces>.
34. **Universidad Santo Tomás.** Ustamed. [En línea] 2013. [Citado el: 12 de Enero de 2015.]
35. **Netcraft.** Netcraft.com. [En línea] 02 de 2015. [Citado el: 20 de marzo de 2015.] <http://news.netcraft.com/archives/2015/02/24/february-2015-web-server-survey.html>.
36. **Apache Tomcat.** The Apache Software Foundation. [En línea] 1999-2015. [Citado el: 11 de marzo de 2015.]
37. **Lanzillotta, Analía.** Definición de Lenguaje de Programación. *MasterMagazine*. [En línea] 2015. [Citado el: 03 de Febrero de 2015.] <http://www.mastermagazine.info/termino/5560.php>.
38. **Programming Community Index TIOBE.** TIOBE.com. [En línea] 2015. [Citado el: 20 de 01 de 2015.] <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
39. **Jaspersoft Community.** Jaspersof. *Jaspersoft Community*. [En línea] 2014. [Citado el: 11 de Marzo de 2015.] <https://community.jaspersoft.com/project/jasperreports-library>.
40. **Velázquez Alfonso, Delmis.** *Sistema para la administración de licencias del producto DalasQ*

- (*DalasQLicenseManager*). La Habana : Universidad de las Ciencias Informáticas, 2013.
41. **Eclipse Foundation.** Eclipse. [En línea] [Citado el: 08 de Marzo de 2015.] <https://projects.eclipse.org/releases/luna>.
 42. **DB-ENGINES.** db-engines.com. [En línea] junio de 2015. [Citado el: 20 de 03 de 2015.] <http://db-engines.com/en/ranking>.
 43. **PostgreSQL-ES.** PostgreSQL-ES. [En línea] 02 de Octubre de 2010. [Citado el: 11 de Marzo de 2015.] http://www.postgresql.org.es/sobre_postgresql.
 44. **Larman, Craig.** UML y Patrones 2da Edición. *Un modelo Conceptual*. s.l. : Prentice Hall, 2003.
 45. **Sommerville, Ian.** *Software Engineering, 9th Edition*. 2010.
 46. **Guerra Valle, Yudiel.** *Módulo para el Monitoreo y Control de los Indicadores de Servicios del Sistema DalasQ*. La Habana : UCI, 2014.
 47. **Ortíz Valmaseda, Marcos y Rodríguez Cervantes, Susany .** *Herramienta para la migración de bases de datos Oracle a PostgreSQL*. La Habana : Universidad de las Ciencias Informáticas, 2009.
 48. **Pavón Mestras, Juan.** *Patrones de diseño orientado a objetos*. Madrid : Universidad Complutense Madrid, 2004.
 49. **Alur, Deepak, John, Crupi y Malks, Dan.** Core J2EE Patterns Best Practices and Design Strategies. [En línea] <ftp://ftp.hxu.edu.cn/pub/Develop/J2EE/Core%20J2EE%20Patterns.pdf>.
 50. **Inc., Microsoft.** msdn.microsoft. *Developer Network*. [En línea] [Citado el: 19 de 05 de 2015.] <https://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx>.
 51. **IEEE Computer Society.** 29119-1-2013 - Software and systems engineering —Software testing —Part 1:Concepts and definitions. s.l. : IEEE Standards Association , 2013.
 52. **Española, Real Academia.** Real Academia Española. [En línea] Octubre de 2014. [Citado el: 04 de 03 de 2015.] <http://lema.rae.es/drae/?val=monitorizar>.
 53. **Blanco, Marco Antonio.** Activa Sistemas. *Blog Activa Sistemas* . [En línea] 22 de Enero de 2009. [Citado el: 09 de Enero de 2015.] <http://www.activasistemas.com/blog/2009/01/22/la-necesidad-de-monitorizar-un-sistema-informatico/>.
 54. **Buchele, Robert B.** *How to evaluate a firm*. California : California Managment Review, 1962.
 55. **Tabares Silverio, Yosdani .** *Implementación de funcionalidades al servidor web de Inteligencia Empresarial para el departamento de Estadísticas del MINCEX*. pág. 17.

REFERENCIAS BIBLIOGRÁFICAS

56. **Crespo Pérez, David y Fernández Pérez, Nelson Francisco** . *Desarrollo del Módulo de Configuración del Sistema de Información Hospitalaria alas HIS*. La Habana : Universidad de las Ciencias Informáticas, 2010.
57. **Denoncourt, Don**. Help 400. *Elección del servidor de aplicaciones web* . [En línea] Febrero de 2003. [Citado el: 10 de marzo de 2015.] <http://www.help400.es/asp/scripts/nwart.asp?Num=131&Pag=10&Tip=T>.
58. **Oxford University**. Oxford Dictionaries. [En línea] 2015. [Citado el: 09 de Marzo de 2015.] <http://www.oxforddictionaries.com/definition/english/control>.

Anexo 1: Historias de usuario.

Tabla 11: HU_1: Autenticar usuario.

Historia de Usuario	
Número: HU_1	Nombre Historia de Usuario: Autenticar usuario.
Modificación de Historia de Usuario Número:	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
<p>Descripción: La presente historia de usuario tiene como objetivo permitir y restringir el acceso de usuarios a la aplicación.</p> <p>El usuario debe escribir su nombre de usuario en el campo de texto indicado como "usuario" y su contraseña de usuario en el campo de texto indicado como "contraseña". Luego de dar presiónar el botón "Aceptar" el sistema verificará si los datos introducidos son correctos y actualizará al usuario como "autenticado".</p>	
<p>Observaciones: Solo tendrán acceso a la aplicación los usuarios que formen parte de la tabla "sec_user" de la base de datos del sistema DalasQ.</p>	
<p>Prototipo de interfaz: Acceso de usuario.</p> <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p style="text-align: center;">Autenticar Usuario</p> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="text-align: left;"> <p>Usuario <input style="width: 80px;" type="text"/></p> <p>Contraseña <input style="width: 80px;" type="password"/></p> </div> <div style="text-align: center;">  </div> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div> </div>	

Tabla 12: HU_2: Visualizar usuario.

Historia de Usuario	
Número: HU_2	Nombre Historia de Usuario: Visualizar usuario.

Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 1
Prioridad en Negocio: Media	Puntos Estimados: 0.4 semana
Riesgo en Desarrollo: Bajo	Puntos Reales: 0.4 semana
Descripción: La presente historia de usuario tiene como objetivo que una vez autenticado el usuario, pueda ver en todo momento el usuario conectado.	
Observaciones: Para poder visualizar el usuario es necesario haberse autenticado previamente.	
Prototipo de interfaz: Visualizar usuario conectado.	
	

Tabla 13: HU_3 Representar árbol de ubicaciones de las instalaciones por niveles de división territorial.

Historia de Usuario	
Número: HU_3	Nombre Historia de Usuario: Representar árbol de ubicaciones de las instalaciones por niveles de división territorial.
Modificación de Historia de Usuario Número: -	
Usuario: Orson Omar Duque Cortina	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 1.6 semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1.6 semana
Descripción: La presente historia de usuario tiene como objetivo representar mediante un árbol jerárquico la ubicación de todas las instalaciones reconocidas por el sistema teniendo en cuenta la división territorial.	

Observaciones: La aplicación solo reconocerá aquellas instalaciones que se encuentren creadas por el sistema dentro de la tabla "location".

Prototipo de interfaz: Árbol de ubicaciones.

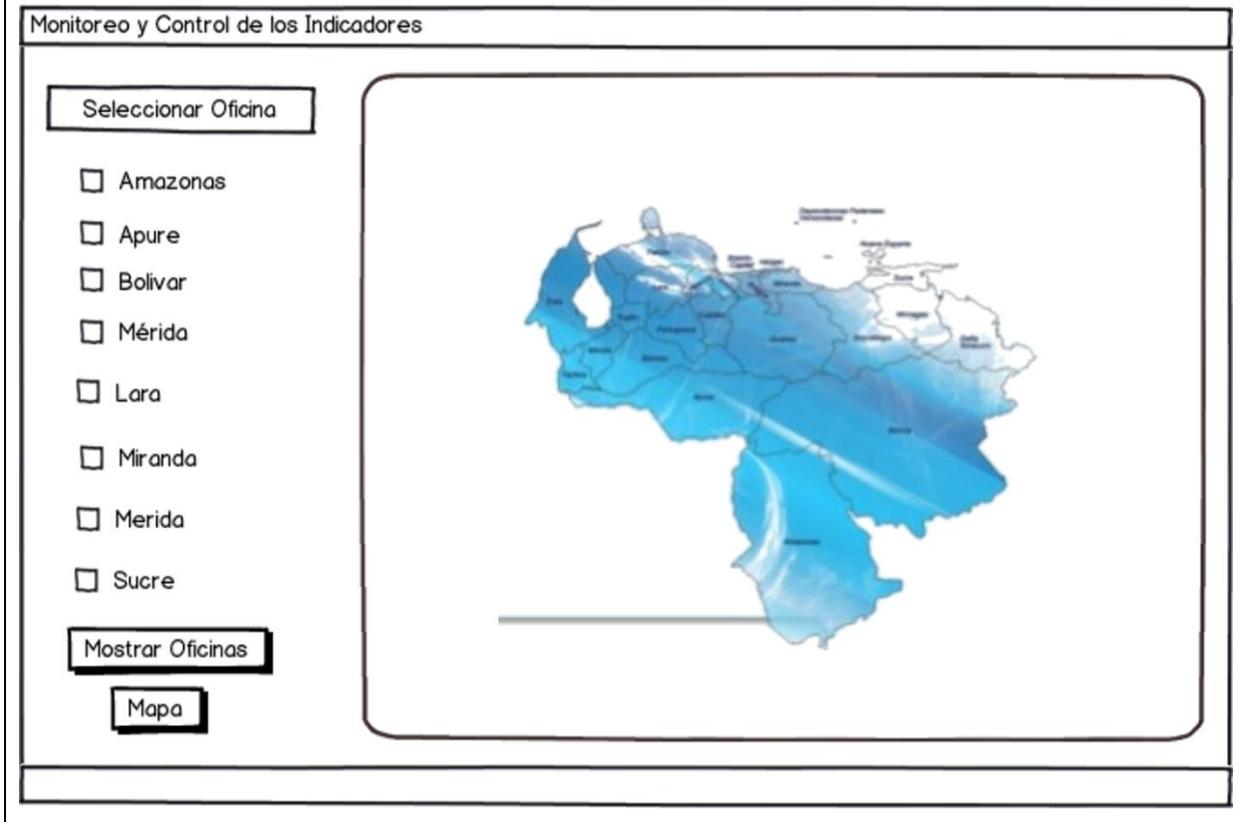


Tabla 14: HU_4 Configurar los valores mínimos y máximos de los indicadores de las instalaciones en todo momento.

Historia de Usuario	
Número: HU_4	Nombre Historia de Usuario: Configurar los valores mínimos y máximos de los indicadores de las instalaciones en todo momento.
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1 semana
Descripción: La presente historia de usuario tiene como objetivo definir los valores mínimos y máximos que pueden tomar los indicadores, estos valores son ingresados por los directivos a cada una de las instalaciones a monitorear.	

Observaciones: Para poder modificar esta información y realizar los cambios en el dashboard se debe presionar en la opción correspondiente a “Configurar Indicadores”.

Prototipo de interfaz: Configuración de indicadores.

Configuración de Indicadores

Seleccione Oficina

Servicio	Valor Mínimo	Valor Máximo
Indicadores		
Tiempo de atención	<input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/>	<input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/>
Tiempo espera	<input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/>	<input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/> <input style="width: 25px; height: 20px; border: 1px solid black;" type="text" value="3"/>
Clientes Abandonan(%)	<input style="width: 100px; height: 25px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/>	<input style="width: 100px; height: 25px; border: 1px solid black;" type="text" value="3"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▲"/> <input style="width: 15px; height: 15px; border: 1px solid black; margin: 0 2px;" type="button" value="▼"/>

De Funcionarios

Tabla 15: HU_7 Representar el estado de los indicadores de los funcionarios de la oficina seleccionada.

Historia de Usuario	
Número: HU_7	Nombre Historia de Usuario: Representar el estado de los indicadores de los funcionarios de la oficina seleccionada.
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 2.0 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 2.0 semanas
Descripción: La presente historia de usuario tiene como objetivo actualizar cada una de los indicadores, siempre que ocurra una inserción en la tabla “attention” o “worker_incident” de la base de datos.	
Observaciones: Para poder representar el estado de los indicadores de los	

funcionarios de la oficina seleccionada, se debe presionar sobre la región que se desea obtener información.

Prototipo de interfaz: Representar Estado Indicadores Funcionario.

Funcionarios							
	ID	PD	PS	PND	PDES	PB	PO
							
							
							
							
							

Tabla 16: HU_8 Crear y modificar las alarmas de las instalaciones mediante el correo del supervisor.

Historia de Usuario	
Número: HU_8	Nombre Historia de Usuario: Crear y modificar las alarmas de las instalaciones mediante el correo del supervisor.
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1.2 semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1.2 semana
Descripción: La presente historia de usuario tiene como objetivo establecer los datos necesarios para poder realizar el proceso de aviso mediante correo electrónico a los supervisores de las instalaciones. Primeramente se establecen los datos del servidor web y acto seguido se ingresan los datos de e-mail de los supervisores.	
Observaciones: Estos datos pueden modificarse durante todo el tiempo de conexión del usuario en la aplicación. Para acceder a esta opción se debe presionar en la opción correspondiente a "Datos para Alarmas".	
Prototipo de interfaz: Datos para Alarmas.	

Configuración de Alarmas

Configuración del servidor

Servidor

Puerto

Autenticación con el servidor

Correo

Contraseña

Oficinas



Tabla 17: HU_9 Enviar correo al supervisor de la instalación una vez disparado un indicador de rendimiento.

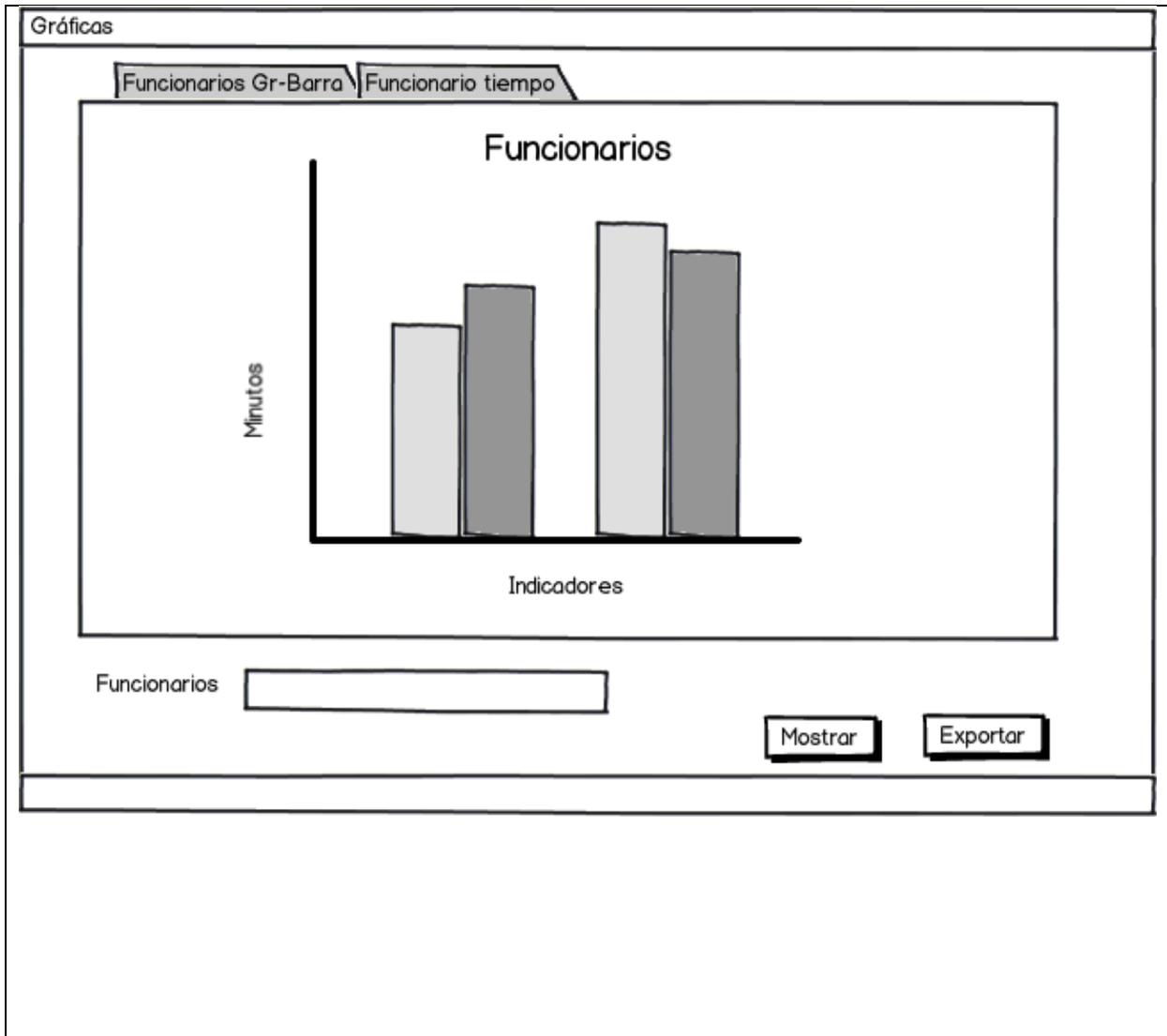
Historia de Usuario	
Número: HU_9	Nombre Historia de Usuario: <i>Enviar correo al supervisor de la instalación una vez disparado un KPI.</i>
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1.4 semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1.4 semana
Descripción: <i>La presente historia de usuario tiene como objetivo el envío de correos electrónicos a los supervisores de las oficinas una vez que se encuentre dicha instalación o funcionario fuera de los objetivos establecidos por los directivos. En el cuerpo del correo se especificará lo ocurrido y el comportamiento en ese instante de la instalación o funcionario.</i>	
Observaciones: <i>En la interfaz de la aplicación aparecerá un mensaje cuando un KPI este fuera de los objetivos establecidos y notificará sobre el envío de correo al supervisor. En caso de que las configuraciones de las alarmas no se hayan realizado</i>	

o estén incorrectos los datos, aparecerá un mensaje especificándolo.

Prototipo de interfaz:

Tabla 18: HU_10 Visualizar estadísticas de los funcionarios.

Historia de Usuario	
Número: HU_10	Nombre Historia de Usuario: Visualizar estadísticas de los funcionarios.
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1.2 semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1.2 semana
Descripción: <i>La presente historia de usuario tiene como objetivo la visualización estadística mediante gráficas de los indicadores de los funcionarios.</i>	
Observaciones: <i>Durante la visualización de las estadísticas se actualiza la gráfica cada vez que cambien los indicadores de los funcionarios seleccionados. Los tiempos promedios se encuentran en minutos, por lo que debajo se encuentra la opción de visualizar el valor en formato hora.</i>	
Prototipo de interfaz: Estadísticas Funcionarios.	



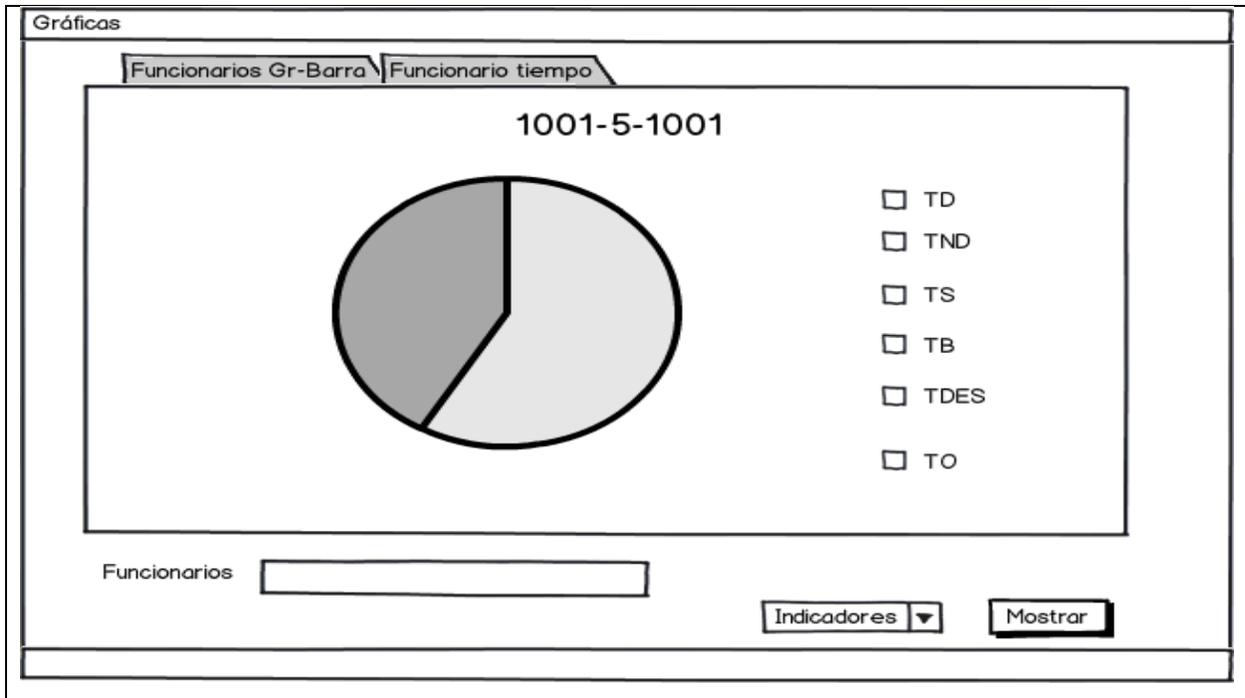


Tabla 19: HU_11 Generar reportes.

Historia de Usuario	
Número: HU_11	Nombre Historia de Usuario: Generar reportes.
Modificación de Historia de Usuario Número: -	
Usuario: Orson O. Duque Cortina	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimados: 1.0 semana
Riesgo en Desarrollo: Medio	Puntos Reales: 1.0 semana
Descripción: La presente historia de usuario tiene como objetivo generar los reportes con la información que muestran los indicadores de las gráficas (tablas) de la región seleccionada u instalación seleccionada.	
Observaciones: El informe aparecerá en pantalla una vez generado correctamente, dando la opción de abrirlo con la aplicación indicada en la estación de trabajo o guardar el informe.	
Prototipo de interfaz:	

Anexo 2: Tareas de Ingeniería

Tabla 20: Tarea de Ingeniería HU_3.

Tarea de Ingeniería	
Número Tarea: 3.1	Número Historia de Usuario: HU_3
Nombre Tarea: <i>Implementación del árbol de ubicaciones.</i>	
Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: 1.6 <i>semanas</i>
Fecha Inicio: 01/02/2015	Fecha Fin: 14/02/2015
Programador Responsable: <i>Orson O. Duque Cortina</i>	
Descripción: <i>Implementar las funcionalidades necesarias para conformar el árbol de ubicaciones, atendiendo a la división territorial a través de los datos en la tabla "location" de la base de datos del sistema DalasQ.</i>	

Tabla 21: Tarea de Ingeniería HU_4.

Tarea de Ingeniería	
Número Tarea: 4.1	Número Historia de Usuario: HU_4
Nombre Tarea: <i>Configurar valores de los indicadores.</i>	
Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: 1.0 <i>semanas</i>
Fecha Inicio: 15/02/2014	Fecha Fin: 21/02/2015
Programador Responsable: <i>Orson O. Duque Cortina</i>	
Descripción: <i>Implementar la interfaz que le aparecerá al usuario para configurar el rango estratégico de los indicadores, con los campos necesarios para su gestión de manera amigable para su uso así como las funcionalidades necesarias para el manejo de los datos durante el proceso de monitoreo.</i>	

Tabla 22: Tarea de Ingeniería HU_5.

Tarea de Ingeniería	
Número Tarea: 5.1	Número Historia de Usuario: HU_5
Nombre Tarea: <i>Crear dashboard.</i>	

Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: 1.0 <i>semana</i>
Fecha Inicio: 22/02/2015	Fecha Fin: 28/02/2015
Programador Responsable: <i>Orson O. Duque Cortina</i>	
Descripción: <i>Implementar las funcionalidades necesarias para crear un dashboard global a partir de los estados de los indicadores de las regiones, con una presentación amigable y coherente. Además mostrar el menú de opciones de visualización y configuración de la aplicación.</i>	

Tabla 23: Tarea de Ingeniería HU_5.

Tarea de Ingeniería	
Número Tarea: 5.2	Número Historia de Usuario: HU_5
Nombre Tarea: <i>Información del dashboard.</i>	
Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: 2.0 <i>semanas</i>
Fecha Inicio: 01/03/2015	Fecha Fin: 13/03/2015
Programador Responsable: <i>Orson O. Duque Cortina</i>	
Descripción: <i>Implementar las funcionalidades necesarias para calcular y mostrar el estado de los indicadores por región. (La región tomará color blanco cuando no tenga información relacionada con la misma en la BD en el día actual. Será de color verde si todas las instalaciones de esa región según el indicador seleccionado están en el rango permitido. En caso de que alguna oficina tenga el indicador seleccionado fuera de rango, esta región será de color rojo.</i>	

Tabla 24: Tarea de Ingeniería HU_6.

Tarea de Ingeniería	
Número Tarea: 6.1	Número Historia de Usuario: HU_6
Nombre Tarea: <i>Crear gráfica (tabla) de la región.</i>	
Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: 1.0 <i>semana</i>

Fecha Inicio: 14/03/2015	Fecha Fin: 20/03/2015
Programador Responsable: Orson O. Duque Cortina	
Descripción: <i>Implementar las funcionalidades necesarias para crear una gráfica (tabla) a partir de los estados de los indicadores de las oficinas de la región seleccionada, con una representación amigable y coherente. Además mostrar la opción de generar reporte a partir de la tabla.</i>	

Tabla 25: Tarea de Ingeniería HU_6.

Tarea de Ingeniería	
Número Tarea: 6.2	Número Historia de Usuario: HU_6
Nombre Tarea: <i>Información de la gráfica (tabla) de la región.</i>	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.5 semanas
Fecha Inicio: 21/03/2015	Fecha Fin: 01/04/2015
Programador Responsable: Orson O. Duque Cortina	
Descripción: <i>Implementar las funcionalidades necesarias para calcular y mostrar el estado de los indicadores por instalación así como su valor. Los indicadores mostrarán su correspondiente valor y el estado en el que se encuentran. (El indicador tomará color blanco cuando no tenga información relacionada con la misma en la BD en el día actual. Será de color verde si el indicador en cuestión está en el rango permitido. En caso de que el indicador esté fuera de rango será de color rojo.</i>	

Tabla 26: Tarea de Ingeniería HU_7

Tarea de Ingeniería	
Número Tarea: 7.1	Número Historia de Usuario: HU_7
Nombre Tarea: <i>Crear gráfica (tabla) de la instalación.</i>	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5 semana
Fecha Inicio: 02/04/2015	Fecha Fin: 06/04/2015
Programador Responsable: Orson O. Duque Cortina	
Descripción: <i>Implementar las funcionalidades necesarias para crear una gráfica</i>	

(tabla) a partir de los estados de los indicadores de las funcionarios de la instalación seleccionada, con una representación amigable y coherente. Además mostrar la opción de generar reporte a partir de la tabla.

Tabla 27: Tarea de Ingeniería HU_7.

Tarea de Ingeniería	
Número Tarea: 7.2	Número Historia de Usuario: HU_7
Nombre Tarea: Información de la gráfica (tabla) de la instalación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.5 semana
Fecha Inicio: 07/04/2015	Fecha Fin: 18/04/2015
Programador Responsable: Orson O. Duque Cortina	
Descripción: Implementar las funcionalidades necesarias para calcular y mostrar el estado de los indicadores por funcionario así como su valor. Los indicadores mostrarán su correspondiente valor y el estado en el que se encuentran. (El indicador tomará color blanco cuando no tenga información relacionada con la misma en la BD en el día actual. Será de color verde si el indicador en cuestión está en el rango permitido. En caso de que el indicador esté fuera de rango será de color rojo.	

Tabla 28: Tarea de Ingeniería HU_8.

Tarea de Ingeniería	
Número Tarea: 8.1	Número Historia de Usuario: HU_8
Nombre Tarea: Crear alarmas de notificación.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.2 semanas
Fecha Inicio: 19/04/2015	Fecha Fin: 27/04/2015
Programador Responsable: Orson O. Duque Cortina	
Descripción: Implementar las funcionalidades necesarias para crear una interfaz donde se establecen los datos necesarios para conectarse al servidor y posteriormente la dirección de correo de los supervisores para el envío de notificaciones. Los datos se pueden modificar durante el tiempo que se encuentre el usuario en línea.	

Tabla 29: Tarea de Ingeniería HU_12.

Tarea de Ingeniería	
Número Tarea: 12.1	Número Historia de Usuario: HU_12
Nombre Tarea: Envío automático de e-mails.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.4 semanas
Fecha Inicio: 28/04/2015	Fecha Fin: 08/05/2015
Programador Responsable: Orson O. Duque Cortina	
Descripción: Implementar las funcionalidades necesarias para enviar de manera automática un e-mail a los supervisores de las instalaciones o funcionarios, una vez que se actualice el estado o estén fuera de los objetivos estratégicos establecidos por los directivos. Además mostrar una notificación del estado del envío, que puede ser: satisfactorio o fallido, de lo contrario de no haber establecido antes los datos de configuración se lanzará una sugerencia indicándolo.	

Tabla 30: Tarea de Ingeniería HU_13.

Tarea de Ingeniería	
Número Tarea: 13.1	Número Historia de Usuario: HU_13
Nombre Tarea: Visualización de estadísticas.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1.2 semanas
Fecha Inicio: 09/05/2015	Fecha Fin: 17/05/2015
Programador Responsable: Orson O. Duque Cortina	
Descripción: Implementar las funcionalidades necesarias para mostrar una interfaz con las gráficas estadísticas de los funcionarios seleccionados, la misma se actualizará acorde al tiempo establecido por el cliente para las gráficas.	

Tabla 31: Tarea de Ingeniería HU_14.

Tarea de Ingeniería	
Número Tarea: 14.1	Número Historia de Usuario: HU_14

Nombre Tarea: <i>Generar reportes de las instalaciones.</i>	
Tipo de Tarea: <i>Desarrollo</i>	Puntos Estimados: <i>1.0 semanas</i>
Fecha Inicio: <i>18/05/2015</i>	Fecha Fin: <i>24/05/2015</i>
Programador Responsable: <i>Orson O. Duque Cortina</i>	
Descripción: <i>Implementar las funcionalidades necesarias para generar un reporte a partir de la información de la tabla.</i>	