

UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA



< APLICACIÓN WEB PARA LA CREACIÓN DE CUADROS DE MANDO DE INDICADORES >

Trabajo de Diploma para optar por el título de Ingeniero Informático

Autor: < Luis Raúl Alfonso Caballero >

Tutores: < Dr. Walfredo González Hernández >

Matanzas, 2022

PENSAMIENTOS

La informática se convertirá en una poderosísima fuerza científica, económica e incluso política del país...

Fidel Castro Ruz



DEDICATORIA

El siguiente trabajo se lo dedico a mis padres que desde pequeño me enseñaron el camino a seguir para superarme cada día y convertirme en mejor persona y mejor profesional.

AGRADECIMIENTOS

Le agradezco a mis amigos de la universidad y mi familia, a los trabajadores del XETID y al profesor Walfredo que desde antes de comenzar a ser mi tutor siempre le tuve estima, lo considero un amigo y un gran profesor.

DECLARACIÓN DE AUTORÍA

Yo, Luis Raúl Alfonso Caballero, declaro que soy el único autor del trabajo "APLICACIÓN WEB PARA LA CREACIÓN DE CUADROS DE MANDO DE INDICADORES"



Dr. C. Walfredo González Hernández



Luis Raúl Alfonso Caballero

OPINIÓN DE LOS TUTORES

TÍTULO de la TESIS: Aplicación web para la creación de Cuadro Mando Integral

AUTOR. - Luis Raúl Alfonso Caballero.

TUTOR: Dr. C. Walfredo González Hernández

Los cuadros de mando integral resuelven una problemática esencial en el manejo eficaz de los datos en una empresa. Estos cuadros permiten organizarlos y darles sentido para los directivos que se encargan de tomar las decisiones a corto, mediano y largo plazo.

La tesis cuenta con una estructura y un sustento teórico y metodológico adecuado para este tipo de investigación, la cual fue una plataforma correcta y necesaria para conducir de forma acertada la investigación y aplicar cada método de modo acertado y oportuno. Asumir las complejidades inherentes a la seguridad son resueltos utilizando diversas técnicas. Es necesario reconocer que el estudiante llega a este resultado a partir de un compromiso con su labor formativa en un entorno de complejidad para esta labor sobre todo por la covid. Esta investigación fue resultado de un proceso largo, continuo y sistemático para lograr en Luis Raúl Alfonso enfrentar el desarrollo de un sistema que se va a implantar.

El estudiante ha logrado una madurez apreciable en su proceso de formación; trabajó con independencia, creatividad, disciplina y, sobre todo, constancia ejemplar. Socializó sus resultados de forma sistemática en los diferentes cortes. Evidenció crecimiento profesional y personal en todo el proceso de formación académica. Para quienes suscriben estas opiniones ha sido un placer acompañar al estudiante en esta empresa y ha constituido un proceso de aprendizaje mutuo. La tesis, perfectible como toda obra humana, considero que reúne los requisitos planteados para este tipo de ejercicio.



Dr. C. Walfredo González Hernández

RESUMEN

A medida que se desarrollaba el proyecto “Varadero Ciudad Digital”, surgió la necesidad de una herramienta para la medición del índice de digitalización de las diferentes entidades que se encontraban encerradas en este proceso. Mediante estudios manuales o con métodos digitales básicos como hojas de cálculo en Excel, este análisis se hacía muy engorroso y provocaba pérdidas de tiempo, dinero y no ofrecía un nivel de información histórica de manera sencilla. Esta situación provocó la demanda de revolucionar estos métodos y la elaboración de un cuadro de mando de indicadores que se adaptara a las distintas entidades fue la solución; a partir de esta situación se decide desarrollar una aplicación web para la creación de estos, la cual permite a los especialistas realizar su trabajo de manera segura y en menos tiempo del que se requiere en la actualidad, proporcionando también confidencialidad a los datos con los que se trabaja. Se empleó la metodología ágil de desarrollo de software **XP**, además de apoyarse para una mejor evolución de software en algunos de los diagramas de la metodología pesadas, el lenguaje de programación PHP con su framework Synphony y como gestor de base de datos PostgreSQL. Se obtuvo un software funcional, el cual se le realizaron una serie de pruebas que permitieron constatar su funcionalidad y verificar que cumpliera con las necesidades y exigencias del cliente.

SUMMARY

As the "Varadero Ciudad Digital" project developed, the need arose for a tool to measure the digitization rate of the different entities that were involved in this process. Through manual studies or with basic digital methods such as Excel spreadsheets, this analysis became very cumbersome and caused loss of time and money and did not offer a level of historical information in a simple way. This situation provoked the demand to revolutionize these methods and the elaboration of a scorecard of indicators that was adapted to the different entities was the solution; Based on this situation, it was decided to develop a web application for the creation of these, which allows specialists to carry out their work safely and, in less time, than is currently required, also providing confidentiality to the data with which they are used. it works. The agile XP software development methodology was used, in addition to relying on some of the heavy methodology diagrams for a better software evolution, the PHP programming language with its Symphony framework and PostgreSQL as a database manager. A functional software was obtained, which was carried out a series of tests that allowed verifying its functionality and verifying that it met the needs and demands of the client.

Contenido

< APLICACIÓN WEB PARA LA CREACIÓN DE CUADROS DE MANDO DE INDICADORES > 1

Introducción	1
Marco teórico-referencial	4
1.1 Introducción.	4
1.2 Objeto de estudio.....	4
1.3 Antecedentes	4
1.4 Justificación	4
1.5 Metodología de desarrollo	5
1.6 Métodos de la investigación	8
1.7 Herramientas y tecnologías.....	9
1.7.1 Arquitectura cliente-servidor:.....	9
1.7.2 Software Libre.....	9
1.7.3 Lenguaje de modelado	10
1.7.4 Lenguaje de programación	10
1.7.5 Framework	11
1.7.6 Herramientas de desarrollo	13
1.7.7 Herramientas CASE: yED.....	14
1.7.8 Patrones de diseño.	15
1.8 Conclusiones del capítulo.....	15
2 Propuesta de solución práctica al problema científico.	16
2.1 Introducción.	16
2.2 Descripción del negocio.....	16
2.3 BMPN.....	16
2.4 Propuesta de solución	16
2.4.1 Captura de requisitos.....	17
2.4.2 Requisitos Funcionales.....	17
2.4.3 Requisitos No Funcionales.	17
2.5 Modelado	18
2.5.1 Diseño de la base de datos.	18
2.6 Descripción de la solución.	19
2.6.1 Historia de Usuarios.	20
2.6.2 Análisis de factibilidad.	22
2.7 Conclusiones del capítulo	25
3 Propuesta de solución práctica al problema científico	26
3.1 Introducción	26
3.2 Descripción de la solución	26

3.3	Descripción del software.....	26
3.4	Pruebas de software.....	29
3.5	Resultados de las pruebas realizadas.....	30
3.6	Conclusiones parciales.....	30
4	Conclusiones.....	31
5	Bibliografía.....	32
6	Anexos.....	33

Ilustración 1	BPMN.....	16
Ilustración 2	Diseño de la base de datos.....	19
Ilustración 3	Formulario de inicio de sesión.....	26
Ilustración 4	Formulario de creación de una entidad.....	27
Ilustración 5	Formulario de creación de una variable.....	27
Ilustración 6	Formulario de creación de un indicador.....	28
Ilustración 7	Interfaz de entrada de valores.....	28

Introducción

En las empresas cada vez es más complejo supervisar las actividades de los sistemas y subsistemas de la organización, esto ocasiona pérdidas de tiempo y dinero a las mismas de diversas maneras. La cantidad de variables e indicadores de seguimiento existentes en el día a día son demasiadas, por lo tanto, hace necesario ver nuevas formas de administración o supervisión. No puede centrarse exclusivamente en generar beneficios a corto plazo, aunque esto le suponga una merma de sus beneficios actuales. En los últimos años se ha insistido en una herramienta que pueda revolucionar la gestión de estos aspectos.

Como caso de estudio se escogió el Proyecto “Varadero Ciudad Digital”. En el mismo se hace necesario llevar a cabo un estudio del estado de digitalización existente y dar seguimiento del mismo a medida que vaya avanzando el proyecto, para esto deben medirse una serie de aspectos que tributarán en un final a un índice de digitalización. Debido a la que en todas las entidades no se miden los mismos elementos se complejiza mucho a nivel organizativo este estudio y eventualmente el análisis de los datos recogidos. El sistema actual utilizado para realizar estos análisis se basa en hojas de cálculo electrónicas. Al entrevistar a los encargados del proyecto se confirmó lo que era evidente al trabajar de esta forma, que es la falta de centralización de los datos, la poca portabilidad, la falta de acceso en tiempo real a los datos, la dificultad de almacenar datos históricos para estudios estadísticos y la gran dependencia del factor humano. Todo lo anterior identifica como **situación problemática** la insuficiente disposición de información clave, pertinente, actualizada, centralizada y relevante, para apoyar los procesos de toma de decisiones y eliminar subjetividad de los mecanismos actuales. Por ello se determina como problema científico: ¿cómo elevar la efectividad en la supervisión de entidades y procesos?

Para la solución del problema se trazó la **hipótesis** que rige la investigación, si se desarrolla un Sistema para la Creación de Cuadros de Mando de Indicadores se contribuirá a elevar la efectividad en la supervisión de entidades y procesos.

Objeto de estudio: Cuadros de mando indicadores para entidades y procesos

Campo: Sistemas para la Creación de Cuadros de Mando de Indicadores

Se define como **objetivo general** de la investigación: desarrollar un sistema web para la creación de cuadros de mando de indicadores que contribuya a la elevación de la eficacia y la eficiencia en la supervisión de entidades y procesos.

Con el propósito de dar cumplimiento al objetivo general de esta investigación se trazaron los siguientes **objetivos específicos**:

1. Determinar el marco teórico referencial de la misma para el desarrollo del Sistema para la Creación de cuadros de mando de indicadores

2. Determinar las herramientas y tecnologías acordes al sistema el Sistema web para la Creación de cuadros de mando de indicadores.
3. Diseñar el Sistema web para la Creación de cuadros de mando de indicadores.
4. Implementar el Sistema web para la Creación de cuadros de mando de indicadores.
5. Validar el Sistema web para la Creación de cuadros de mando de indicadores a partir de la aplicación de pruebas de software.

Para el desarrollo del módulo se utilizaron diversos métodos tales como:

Dentro de los métodos teóricos:

- Método de análisis histórico – lógico: permitió estudiar la trayectoria y desarrollo de los sistemas de gestión de indicadores empresarial existentes, así como la utilización de estos en nuestro país.
- Método de análisis y síntesis: este se precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.
- Método inductivo - deductivo: su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

Como métodos empíricos, utilizados por medio de las siguientes técnicas:

- Observación: permitió entender el proceso de análisis de la información en el proyecto utilizado como objeto de estudio.
- Entrevistas: fue útil en distintos momentos de la investigación; fundamentalmente al inicio, cuando se realizó el levantamiento de requisitos para efectuar una exploración preliminar del problema a investigar.

Entre los aportes de la investigación se destacan:

- El teórico-investigativo, al sentar bases para futuras investigaciones.
- El práctico, al desarrollar una herramienta que permita la creación de cuadros de mando de indicadores.

El resultado esperado de este trabajo es contar con una herramienta desarrollada que contribuirá a elevar la efectividad en la supervisión de entidades y procesos que permita la creación de cuadros de mando de indicadores y supervisar en progreso de empresas o procesos.

Atendiendo a lo planteado anteriormente, la tesis queda estructurada en introducción, tres capítulos, conclusiones, recomendaciones y referencias bibliográficas, según sigue:

- Una Introducción, donde se caracteriza la situación problemática y se fundamenta el problema científico a resolver.
- Capítulo I “Marco teórico referencial”. En el que se presenta una exposición detallada de los referentes teóricos que argumentan la propuesta y permiten un acercamiento al objeto de estudio. Además, se comenta sobre las tendencias y tecnologías actuales que serán usadas para el desarrollo del sistema.
- Capítulo II “Diseño de la solución propuesta”. En el que se argumenta la solución que se propone al problema de investigación mediante su descripción. Se presenta la planificación inicial del proyecto, con el empleo de la metodología de desarrollo de software XP, y se realiza un estudio tanto de factibilidad, como de los beneficios tangibles e intangibles para la realización del sistema.
- Capítulo III “Validación de la solución propuesta”. En el que se describe el software y se le realizan las pruebas con el objetivo de entregarle al cliente un producto totalmente funcional, que cumple con todos los requisitos demandados por el mismo y que satisfaga sus necesidades. Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.
- Las recomendaciones en la cual se plasman una serie de propuestas encaminadas a la continuidad de esta investigación.
- Y las referencias de la bibliografía citada.

Marco teórico-referencial

1.1 Introducción.

Es necesario el correcto enmarcado de la investigación a desarrollar, por lo que este capítulo tiene como objetivo un estudio de las bases teóricas existentes que sustentan la investigación referente a la propuesta de solución. Además, en el propio capítulo se describen las principales tendencias tecnológicas que se emplearán en la investigación. Se incluye también, la caracterización del objeto de estudio y el análisis de los antecedentes relacionados con la temática investigada.

1.2 Objeto de estudio.

El objeto de estudio de la investigación se establece a partir del proceso de supervisión del nivel de digitalización del proyecto “Varadero Ciudad Digital”, pues en dicho proceso la gestión de los datos posee poca organización debido a que la información no está centralizada lo que trae consigo una mala manipulación, falta de integridad y persistencia de los datos y como consecuencia de esto, en ocasiones la información emitida carece de veracidad. Además de que la confiabilidad y seguridad en la interactividad entre los usuarios que manejan la información es pobre, ya que no existe ningún mecanismo que regule y garantice este proceso.

El proceso de toma de decisiones también se ve afectado, ya que la información no llega a los decisores en tiempo, ni con la veracidad requerida para que el proceso fluya adecuadamente. La realización de análisis estadísticos a partir de los datos históricos no es posible ya que en numerosas ocasiones esta información no existe.

1.3 Antecedentes

Hasta el momento de esta investigación no se encontró ningún sistema de cuadro de mando de indicadores orientado a la medición del nivel de digitalización de entidades. A nivel internacional existen algunos sistemas con puntos en común con la propuesta de solución con el inconveniente de ser de pago lo que lo que hace que no coincidan con la política de software libre que se promueve en nuestro país y no se ajustan a las normas cubanas en cuanto a nivel organizacional; algunos de estos son: **BSC DESIGNER, Klipfolio, Happymetrix, Data Studio, Power BI**

1.4 Justificación

Como justificación a la elaboración de este sistema, vino de un especialista del parque tecnológico encargado de supervisar el proyecto “Varadero Ciudad Digital” que planteaba la necesidad de analizar el proyecto de forma objetiva con datos que reflejen tanto actual como históricamente la situación del mismo, con los beneficios que no brindan las hojas de cálculo electrónicas como serian: portabilidad, seguridad y acceso en tiempo real a los datos.

1.5 Metodología de desarrollo

Las metodologías de desarrollo de software son un conjunto de pasos y procedimientos descritos a través de un sistema de artefactos y que generan una documentación a seguir para llevar a cabo el desarrollo de software con calidad. Estas brindan un conjunto de detalles organizativos añadiendo rigurosidad y normas permitiendo que los integrantes de un equipo de desarrollo puedan seguir un criterio común a la hora de realizar las tareas durante el desarrollo de un software. La constante innovación tecnológica hace que cada vez sea más necesaria la aplicación de nuevas metodologías adaptadas a los nuevos tiempos y los proyectos que éstos generan.

En la actualidad existen dos grandes grupos de metodologías de desarrollo, las metodologías tradicionales y las metodologías ágiles. Las primeras se enfocan en el uso exhaustivo de documentación durante el ciclo de vida del software, mientras que las segundas dan una mayor importancia a la capacidad de respuesta para la facilidad de cambios. A continuación, se presenta una breve comparación entre ellas.

En la siguiente tabla se muestra una comparación entre metodologías ágiles y no ágiles.

Metodología Ágil	Metodología Tradicional
Pocos artefactos.	Más artefactos.
Pocos roles, más genéricos y flexibles.	Más roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Se hace énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Se hace énfasis en la definición del proceso: roles, actividades y artefactos.
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.

Tabla 1: Comparación entre metodologías ágiles y no ágiles. (Sánchez, 2003)

Luego de haber realizado una comparación de ambas Metodologías de realización del software (Tabla 1), se observa que las Ágiles se ajustan más para conducir el desarrollo del software propuesto, debido a que este presenta pocos artefactos y roles, hay poco tiempo de desarrollo y es mucho más flexible con los cambios en el software.

Entre las metodologías ágiles más conocidas se encuentran: Scrum, Crystal Metodologies y Extreme Programming (XP). Se selecciona entre las metodologías ágiles XP para desarrollar e implementar la solución planteada.

XP como metodología de desarrollo

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en 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 realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP.

Artefactos

En todo proceso de desarrollo de software se generan modelos de información. En XP se generan varios artefactos como las tarjetas de historias de usuario (Story Card), las tarjetas de tareas para la descarga de documentos, el código, las pruebas unitarias y de integración y las pruebas de aceptación. Los artefactos son importantes para conocer cuál fue el proceso de desarrollo del software y lograr entender cómo está construido el sistema, así como la ruta a seguir para agregar funcionalidad al sistema.

Una **historia de usuario** es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final o cliente. El propósito de una historia de usuario es articular cómo un elemento de trabajo entregará un valor particular al cliente.

En programación, una **prueba unitaria** o test unitario es una forma efectiva de comprobar el correcto funcionamiento de las unidades individuales más pequeñas de los programas informáticos. Por ejemplo, en diseño estructurado o en diseño funcional una función o un procedimiento, en diseño orientado a objetos una clase.

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupo.

Las **pruebas de aceptación** XP, también llamadas pruebas del cliente son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente.

Debido a la dimensión del sistema y el nivel bajo de acoplamiento que posee su lógica, para un mejor aprovechamiento del tiempo se decidió omitir las pruebas unitarias y las pruebas de integración utilizando solamente las pruebas de aceptación para validar este, al realizar estas pruebas de forma escalada se puede lograr una correcta valoración. Los componentes del software se validaron por los especialistas de la empresa XETID según sus políticas de calidad por lo tanto no se realizaron pruebas de validación por parte del autor de la investigación; no fueron necesarias las pruebas de seguridad debido a que los servidores donde se hospedarán la misma se encuentran ya configurados por la misma empresa. Aunque en otras fuentes de información aparecen algunas variaciones y extensiones de roles XP, en este apartado describiremos los roles de acuerdo con la propuesta original de Beck.

Programador

El programador escribe las pruebas unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo.

Cliente

El cliente escribe las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es sólo uno dentro del proyecto, pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

Encargado de pruebas (Tester)

El encargado de pruebas ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento (Tracker)

El encargado de seguimiento proporciona realimentación al equipo en el proceso XP. Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. También realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Entrenador (Coach)

Es responsable del proceso global. Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor

Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico.

Gestor (Big boss)

Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

1.6 Métodos de la investigación

Métodos teóricos empleados

Análisis Histórico-lógico: Permitió comprender la forma en la que se analizaban históricamente los distintos aspectos de una empresa o proceso para evaluar su desarrollo o desempeño.

Analítico-sintético: Este se precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división mental del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.

Hipotético-deductivo: Desempeñó un papel esencial en el proceso de verificación del supuesto de partida. A partir de este supuesto se pueden inferir conclusiones y establecer predicciones.

Enfoque de sistema: Proporcionó la orientación general del estudio como una realidad integral formada por elementos que interaccionan unos con otros.

Métodos empíricos empleados

Observación: Acompañó la investigación desde los primeros momentos, a través de la cual se conoció y se observó cómo se lleva a cabo la gestión de los indicadores y las variables de interés para la medición del desempeño de una entidad o proceso y las dificultades que esto conlleva, lo que condujo a la determinación de los procesos y parámetros que se automatizan a partir de la implementación de la aplicación web.

Entrevista: Aportó los datos esenciales de la investigación. Permitió conocer cuáles eran las inquietudes y sugerencias por parte del personal especializado en el tema y qué hacer para agilizar y facilitar su trabajo. Fue útil en distintos momentos de la investigación; fundamentalmente al inicio, cuando se realizó el levantamiento de requisitos para efectuar una exploración preliminar del problema a investigar, de esta forma desarrollarla a gusto del cliente.

Análisis de documentos: Incluye la revisión de documentos como son los manuales y

guías relacionados con el tema. También se consultaron libros y diversos artículos de bancos de información de Internet relacionados con el tema a investigar.

1.7 Herramientas y tecnologías.

Es necesario para el desarrollo de un producto informático que satisfaga una necesidad existente, el empleo de diferentes herramientas y tecnologías con demostrada idoneidad. Ya que, según el problema a resolver, es de suma importancia una correcta elección de las mismas.

Actualmente en el mundo existen disímiles herramientas y tecnologías pertenecientes a la rama de la informática que ayudan a la implementación de diversas aplicaciones. A pesar que nuestro país es subdesarrollado y bloqueado, se realizan numerosos esfuerzos por estar a la altura de las empresas informáticas que compiten a nivel mundial. Es por ello que se encuentra inmerso en la idea de emigrar hacia las tecnologías que utilizan software libre para simular, modelar y dar solución a problemas de la vida cotidiana.

1.7.1 Arquitectura cliente-servidor:

Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un Sistema Gestor de Base de Datos (SGBD). Por otro lado, los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red (Canales, Pena, Peredo, Sossa, & Gutierrez, 2007).

N-capas:

El objetivo de la arquitectura n-capas es separar física y lógicamente los componentes mínimos de una aplicación. Las principales ventajas son (De la Torre Llorente, Castro, Barros, & Nelson, 2010):

- Clara separación de las funciones del control de la interfaz y presentación de datos con la lógica de la aplicación.
- Reusabilidad de componentes.
- Independencia de la interfaz del cliente y la arquitectura de datos.
- Mejores posibilidades de balancear la carga.
- Uso de protocolos abiertos.

1.7.2 Software Libre

Las características del *software* libre, y las implicaciones que tiene su uso, son en gran medida consecuencias de estas libertades que ofrece. La libertad de redistribución proporciona un canal de distribución de gran eficiencia económica, y de muy bajo coste para el productor. Las posibilidades de modificación y de redistribución de las modificaciones facilitan la evolución y mejora técnica de los programas.

El software de código abierto es aquel cuyo código fuente y otros derechos son publicados bajo una licencia compatible con la *Open Source Definition*. El código de un programa que se puede leer, modificar y distribuir tiene mayores probabilidades de que se le realicen mejoras, adaptaciones y correcciones y que así se desarrolle con mayor facilidad.

Para la realización de este trabajo se es consecuente con la situación actual, por lo que todas las herramientas y tecnologías que se utilizan en la realización del sistema con interfaz Web son software libre, ellas son: Sistema Gestor de Bases de Datos (SGBD) PostgreSQL y el Framework Symfony.

1.7.3 Lenguaje de modelado

BPMN en español Modelo y Notación de Procesos de Negocio según lo planteado por (Del Giudice, Soto-Acosta, Carayannis, & Scuotto, 2018), es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. Se utilizó para modelar en la fase de modelación, el negocio y en la fase de implementación

1.7.4 Lenguaje de programación

Es un conjunto de sintaxis y reglas semánticas que definen los programas del ordenador, es una técnica estándar de comunicación para entregarle instrucciones al computador. Le proporciona la capacidad al programador de especificar, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano.

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) según (Orozco Santa María & García Ramírez, 2017) es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Es multiplataforma por lo tanto permite operar en varios sistemas operativos. Así como que tiene un acceso fácil a otras herramientas de la web, además de que es veloz (Php, s,f). Será utilizado en la lógica de negocio de los plugins.

JavaScript (JS): es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo, programación funcional).

HTML, siglas en inglés de HyperText Markup Language ('lenguaje de marcado de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante (Musciano & Kennedy, 2002) siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles. ExtJS es una librería de JavaScript que está basado completamente en la programación orientada a objeto. Cada objeto contiene lo típico: propiedades, métodos y eventos. Una de las grandes ventajas de utilizar ExtJS es que permite crear aplicaciones complejas utilizando componentes predefinidos. Existe un balance entre el cliente y el servidor. La carga de procesamiento se distribuye, permitiendo que el servidor al tener menor carga, pueda manejar más clientes al mismo tiempo. Se utilizará en la modelación de las vistas. Doctrine es una librería de PHP que permite trabajar con un esquema de base de datos como si fuese un conjunto de objetos. Es un ORM (mapeador objeto-relacional) permitiendo la generación automática del modelo de clases y brinda una capa de abstracción de la base de datos. Se empleará para controlar el acceso a los datos (Hodgkinson-Williams & King, 2018).

1.7.5 Framework

Un framework según Emprenderalia, se define como aquella aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y funcionalidades ya creadas para que sean usadas directamente. El objetivo que poseen es hacer que los desarrolladores se centren en los verdaderos problemas y no se preocupen de implementar funcionalidades que son de uso común en muchas

aplicaciones, como podrían ser el proceso de login de usuarios o establecer la conexión con la base de datos. Uno de los motivos que más convence de su uso es que suelen basarse en el patrón Modelo Vista Controlador lo que va a permitir que la aplicación sea más robusta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo, clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

A continuación, se mencionan una serie de características que podemos encontrar en prácticamente todos los Framework existentes.

- **Abstracción de urls y sesiones:** No es necesario manipular directamente las Urls ni las sesiones, el framework ya se encarga de hacerlo.
- **Acceso a datos:** Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos, en BBDD, XML, etc.
- **Controladores:** La mayoría de framework implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.
- **Autenticación y control de acceso:** Incluyen mecanismos para la identificación e usuarios mediante login y password y permiten restringir el acceso a determinadas páginas a determinados usuarios.
- **Internacionalización.**
- **Separación entre diseño y contenido.**

1.7.5.1 *Symfony*

Se elige Symfony entre los frameworks PHP debido a que es un framework extenso y es el único framework que sigue los estándares de PHP y web por completo. Los componentes de Symfony son utilizados por sistemas de gestión de contenido populares como Drupal o PHP Bulletin Board (PHPBB). Es la primera opción para muchos desarrolladores debido a la disponibilidad de bibliotecas y componentes reutilizables con los que se puede completar una amplia variedad de tareas como autenticación, creación de plantillas, configuración de objetos y mucho más.

1.7.5.2 *jQuery*

Es un framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX, además de ser software libre y de código abierto.

JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Otra gran ventaja de JQuery es que permite cambiar el contenido de la página web sin necesidad de recargarla, utilizando DOM y AJAX de manera extremadamente sencilla gracias a su sintaxis (Chaffer & Swedberg, 2011).

1.7.6 Herramientas de desarrollo

1.7.6.1 Sistemas gestores de bases de datos (SGBD)

Se desea almacenar la información de la unidad en un SGBD porque estos son capaces de detectar los datos repetidos. Además, cuentan con un subsistema de seguridad y autorización capaz de crear cuentas y especificar sus restricciones. Permiten la extracción, manipulación y modificación de la información según los diferentes tipos de permisos y el mantenimiento de esquemas. Hay tres características inherentes al enfoque de las bases de datos, la separación de los programas y los datos, el manejo de múltiples vistas de usuario y el empleo de un catálogo para almacenar la descripción de los datos (Yáñez, 2008). El objetivo de la arquitectura de tres esquemas (capas) consiste en formar una separación entre las aplicaciones del usuario y la base de datos física.

PostgreSQL

En la actualidad existen disímiles SGBD, los más conocidos son SQLServer, PostgreSQL, Oracle, MySQL de los cuales se elige PostgreSQL por ser orientado a objetos, libre, código abierto, de fácil instalación y cómodo para la conexión. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema(Sanz, 2010). Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

A continuación, se muestran algunas de las características más importantes y soportadas por PostgreSQL:

- Es una base de datos 100% ACID.
- Integridad referencial.
- Completa documentación.
- Licencia BSD.
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
- Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl.

- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.
- Herencia de tablas (Inheritance).

1.7.6.2 VisualStudio Code

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP y R, al igual que entornos de desarrollo web, como ASP.NET MVC y Django. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros. Se incluye un diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y pruebas de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga. Las ediciones Express se han diseñado para principiantes, aficionados y pequeños negocios, todas disponibles gratuitamente a través de la página de Microsoft. Se incluye una edición independiente para cada lenguaje: Visual Basic, Visual C++, Visual C#, Visual J# para programación .NET en Windows y Visual Web Developer para la creación de sitios web ASP.NET. Las ediciones Express carecen de algunas herramientas avanzadas de programación, así como de opciones de extensibilidad.

1.7.7 Herramientas CASE: yED

YEd es una aplicación de escritorio diseñada para crear de manera eficiente y fácilmente diagramas de calidad. Los usuarios pueden crear diagramas de forma manual, así como conjuntos de datos fuera de importación para facilitar el análisis. yEd tiene algoritmos automáticos para el diseño que puede organizar los conjuntos de datos de todos los tamaños con un solo clic. yEd está disponible para uso gratuito. Es un editor gráfico multiplataforma y es compatible con Windows, Mac OS y Linux. yEd tiene varias características clave que lo hacen ideal tanto para uso personal y profesional. Se pueden importar fácilmente datos de una hoja de cálculo de Excel externo con un formato .xls, o puede importar datos almacenados en archivos XML. La interfaz de usuario intuitiva es fácil de aprender y le permite crear diagramas y sin la necesidad de conocimientos técnicos complejos. Puede organizar automáticamente los elementos de su diagrama para que sean más fáciles de leer. Y, por último, puede exportar sus vectoriales y de mapa de bits en formato JPG, PNG, SWF, SVG y PDF. yEd soporta una

amplia gama de tipos de diagramas. En primer lugar, y los soportes más importantes ilustra diagramas visuales basados en, pero también es capaz de soportar los mapas mentales, organigramas, diagramas entidad-relación, diagramas Swimlane, y mucho más. Cualquier esquema de organización que se pueda imaginar se apoya en el marco yEd.

1.7.8 Patrones de diseño.

El desarrollo de software es un proceso complejo, en especial si se demanda una solución rápida, adaptable y estable, lo que conlleva enfrentar retos continuamente. Con la utilización de patrones de diseño se puede evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.

Modelo vista controlador (MVC): Se estructura de tres componentes fundamentales. Tiene como principio que cada uno de los componentes está separado en diferentes objetos y no se pueden combinar dentro de una misma clase. Separa la lógica del negocio y la capa de presentación. Entre sus ventajas están que permite un proceso de desarrollo más rápido, la modificación no afecta a todo el modelo porque la parte del modelo no es dependiente de algún otro componente como las vistas (Cobián-Alvarado, Damián-Reyes, Andrade-Aréchiga, JRG, & Acosta-Díaz, 2016).

Arquitectura por capas: La arquitectura en capas consta en dividir la aplicación en capas, con la intención de que cada capa tenga un rol muy definido, como podría ser, una capa de presentación, una capa de reglas de negocio y una capa de acceso a datos, sin embargo, este estilo arquitectónico no define cuantas capas debe de tener la aplicación, sino más bien, se centra en la separación de la aplicación en capas

1.8 Conclusiones del capítulo.

Luego de realizar un análisis del objeto de estudio de la investigación, los antecedentes, las herramientas y metodología utilizadas se arriba a las siguientes conclusiones:

- Se determinó el marco teórico referencial, sobre el cual se sustenta la propuesta de trabajo, con vistas al desarrollo del software mediante un estudio de las principales herramientas y tecnologías que se proponen para el proyecto, se sientan las bases que dan pie a su inicio y además de forma general se ha se ha contribuido a la mejor comprensión del objeto de estudio.
- Los sistemas existentes no cumplen con las normas ambientales cubanas, ni el principio de software libre que se promueve en nuestro país
- La combinación de herramientas, tecnologías y la metodología de desarrollo de software establecida seleccionada, es la apropiada para la realización del sistema ya que resuelve la situación problemática planteada en la investigación.

2 Propuesta de solución práctica al problema científico.

2.1 Introducción.

El desarrollo de un sistema informático requiere una gran labor de análisis y diseño para lograr un producto final con la calidad requerida y que a su vez cumpla con las expectativas del cliente. Este proceso tiene como finalidad transformar los requisitos en un diseño del sistema en creación y adaptar el diseño para que se ajuste al entorno de implementación, pensado para el rendimiento. En el presente capítulo se hará una descripción general de las características del sistema a realizar en correspondencia con la metodología de desarrollo de software expuesta en el capítulo anterior y con la ayuda de los procedimientos, técnicas y herramientas que esta nos brinda.

2.2 Descripción del negocio

La conformación del cuadro de mando comienza con la necesidad de una entidad de medir su desarrollo o evaluar un proceso determinado; el especialista creara los indicadores basados en el interés de la entidad y luego las variables que se asocian a las mismas. De esta forma quedará conformado el cuadro de mando y el especialista.

2.3 BPMN

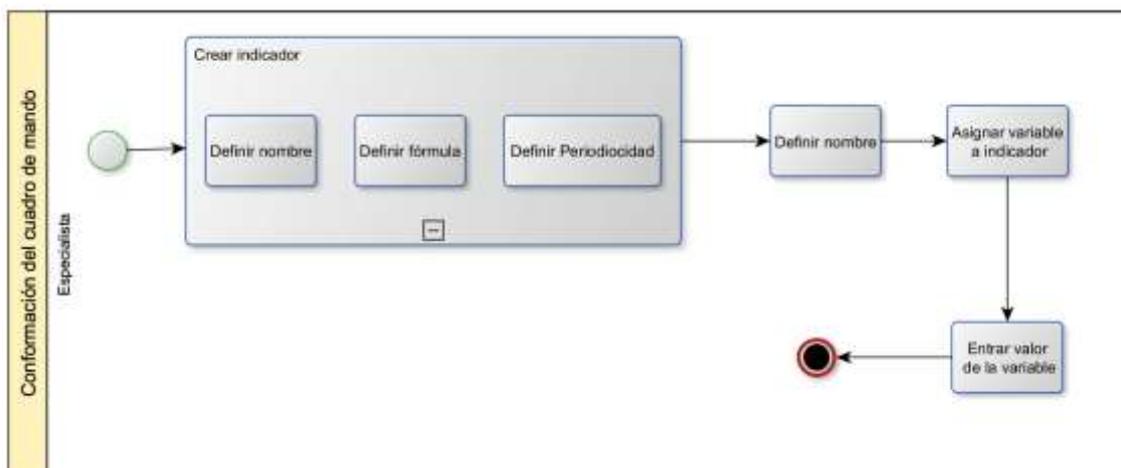


Ilustración 1 BPMN

2.4 Propuesta de solución

Se propone el desarrollo de una aplicación web que les permita a los usuarios con rol de especialistas de los distintos procesos y entidades, crear, ver, modificar y eliminar indicadores y variables asociadas a estos, y a los mismos podrán introducir los datos de las variables según estén disponibles y la aplicación guardara un registro histórico de estos para su posterior análisis.

Los usuarios que interactúan con la aplicación tendrán los siguientes roles:

Administrador total del sistema (Súper Usuario): Tiene la tarea de asignar permisos, crear los usuarios, asignar roles y configurar la aplicación desde otra interfaz.

Especialista: Podrá introducir los nuevos valores a las variables que tengan acceso según su entidad o sus permisos.

Para garantizar la seguridad y la confiabilidad en la información que se procesa es de gran importancia implementar una aplicación web que permita la autenticación para identificar los usuarios que utilizan el programa y a los distintos datos a los que pueden acceder.

2.4.1 Captura de requisitos.

Los requerimientos funcionales de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio, cumplimiento, seguridad u otra índole.

Los requisitos no funcionales se tratan de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. Los requisitos no funcionales se originan en la necesidad del usuario, debido a restricciones presupuestarias, políticas organizacionales, la necesidad de interoperabilidad con otros sistemas de software o hardware, o factores externos tales como regulaciones de seguridad, políticas de privacidad, entre otros.

2.4.2 Requisitos Funcionales

1. Autenticarse
2. Gestionar usuarios
3. Gestionar Entidades
4. Gestionar Indicadores
5. Gestionar Variables
6. Asociar Variables a Entidad
7. Registrar Valores

2.4.3 Requisitos No Funcionales.

1. El módulo del sistema debe ajustarse a los colores y diseño del sistema en general.
2. El módulo del sistema de contener colores acordes al diseño original del sistema.
3. Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
4. Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

5. Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador general del sistema.
6. El nuevo sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.
7. El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
8. El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
9. El sistema debe poseer interfaces gráficas bien formadas.
10. El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
11. El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
12. La tasa de tiempos de falla del sistema no podrá ser mayor al 0,5% del tiempo de operación total.
13. La aplicación debe ser compatible con todas las versiones de navegadores web.
14. La aplicación deberá consumir menos de 500 Mb de memoria RAM al ser operada por un navegador.
15. El sistema deberá ser multiplataforma y ser compatible con los sistemas operativos Windows y Linux. Deberá funcionar con los siguientes requisitos mínimos de software: Sistema operativo: Windows XP o superior. Linux. Gestor de base de datos: Postgresql v8.0 o superior.
16. El sistema deberá encontrarse disponible las 24 horas de todos los días para aquellos usuarios autorizados a acceder al sistema.

2.5 Modelado

2.5.1 Diseño de la base de datos.

Uno de los pasos cruciales en la construcción de un software que maneje una base de datos, es sin duda, el diseño de la misma pues tiene como propósito asegurar que los datos persistentes sean almacenados consistente y eficientemente. En el diseño de la base de datos los modelos de datos tienen un papel significativo pues se encargan de proveer una vista de las entidades lógicas de datos y sus relaciones, si los modelos no son definidos apropiadamente, podemos tener muchos problemas al momento de ejecutar consultas a la base de datos.

3	Gestionar Usuarios	Alta	95
5	Gestionar Entidades	Alta	90
6	Gestionar Variables	Media	85
7	Gestionar Indicadores	Media	80
8	Actualizar Variable	Media	80

Cada una de estas funcionalidades serán descritas en las tablas correspondientes a las historias de usuario.

2.6.1 Historia de Usuarios.

Historia de usuario	
Numero:1	Usuario: Especialista y Administrador total del sistema
Nombre historia: Inicio de sesión	
Prioridad en negocio:1	Riesgo de desarrollo:2
Puntos estimados:2	Iteraciones asignadas:1
Programador responsable: Luis Raúl Alfonso	
Descripción: El usuario ingresa al sistema e introduce su usuario y contraseña para iniciar sesión en el mismo.	
Validación: -Inicio de sesión denegado: El usuario o contraseña ingresado es incorrecto -Inicio de sesión permitido: El usuario y contraseña ingresado es correcto y el sistema muestra la interfaz principal	

Historia de usuario	
Numero:2	Usuario: Administrador Total del Sistema
Nombre historia: Creación de entidad	
Prioridad en negocio:1	Riesgo de desarrollo:2
Puntos estimados:3	Iteraciones asignadas:2
Programador responsable: Luis Raúl Alfonso	
Descripción: El administrador total del sistema crea la entidad o proceso para la posterior conformación de su cuadro de mando por el especialista, asignando su	

nombre, abreviatura, provincia, municipio, organismo, teléfono, dirección, nombre y apellidos del director de la entidad o proceso y su teléfono, y asigna el permiso a este para gestionar los elementos correspondientes al ente creado.

Validación:

-Creación satisfactoria: Los datos ingresados para la creación del elemento han sido validados por el sistema y se ha creado el registro del mismo.

-Creación fallida: El contenido de los datos ingresados no satisface los criterios de aceptación del sistema

Historia de usuario

Numero:3	Usuario: Especialista
Nombre historia: Creación de un indicador	
Prioridad en negocio:1	Riesgo de desarrollo:1
Puntos estimados:2	Iteraciones asignadas:1
Programador responsable: Luis Raúl Alfonso	
Descripción: El especialista mediante las interfaces correspondientes procede a la creación del indicador especificando su nombre, formula, periodicidad tipo y descripción.	
Validación:	
-Creación satisfactoria: Los datos ingresados para la creación del elemento han sido validados por el sistema y se ha creado el registro del mismo.	
-Creación fallida: El contenido de los datos ingresados no satisface los criterios de aceptación del sistema	

Historia de usuario

Numero:4	Usuario: Especialista
Nombre historia: Creación de una variable	
Prioridad en negocio:1	Riesgo de desarrollo:1
Puntos estimados:3	Iteraciones asignadas:2
Programador responsable: Luis Raúl Alfonso	
Descripción: El especialista mediante las interfaces correspondientes procede a la creación de la variable especificando la entidad a la que pertenece, su nombre, su abreviatura y su descripción.	
Validación:	
-Creación satisfactoria: Los datos ingresados para la creación del elemento han sido validados por el sistema y se ha creado el registro del mismo.	

-Creación fallida: El contenido de los datos ingresados no satisface los criterios de aceptación del sistema

Historia de usuario	
Numero:5	Usuario: Especialista
Nombre historia: Asociar variable a indicador	
Prioridad en negocio:1	Riesgo de desarrollo:1
Puntos estimados:4	Iteraciones asignadas:2
Programador responsable: Luis Raúl Alfonso	
Descripción: El especialista mediante las interfaces correspondientes a los indicadores procede a asociar cada una de las variables creadas a su indicador correspondiente.	
Validación: -Asociación satisfactoria: El sistema logro asociar la variable al indicador sin problema alguno. -Asociación fallida: El sistema tubo problemas al asociar la variable al indicador	

2.6.2 Análisis de factibilidad.

La **medición de software** implica analizar las estadísticas de uso de software y ayuda a los administradores de TI a reducir los gastos generales debido a renovaciones y actualizaciones no deseadas. Las funciones de medición de software de Endpoint Central le brindan visibilidad e información sobre los activos de software dentro de su red y lo ayudan a tomar decisiones informadas sobre las renovaciones de licencias de software; esto también reduce los costos de TI. Los Puntos Función constituyen una técnica de medida del software, simple de obtener, pero muy potente en sus resultados. Esta potencia radica en que del valor de la medida en Puntos Función se derivan un conjunto de métricas esenciales para la gestión de la productividad, la calidad y el coste del software. Los Puntos Función proporcionan una medida objetiva, cuantitativa y auditable del tamaño de las aplicaciones, desde el punto de vista de los requisitos especificados por el usuario final de la aplicación. También son un medio de entendimiento entre lo que el usuario quiere y lo que al final se le suministra. En consecuencia, su valoración se deriva a partir de los requisitos funcionales que la aplicación debe satisfacer, modelos de datos, definición de pantallas e interfaces gráficos y diagramas de análisis.

Los sistemas están divididos en cinco componentes generales:

- Entradas externas (EI): Procesos en los que se introducen datos y que suponen la actualización de cualquier archivo interno.
- Salidas externas (EO): Procesos en los que se envía datos al exterior de la aplicación.
- Consultas (EQ): Procesos consistentes en la combinación de una entrada y una salida, en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.
- Ficheros Lógicos Internos (ILF): Grupos de datos relacionados entre sí internos al sistema.
- Ficheros Lógicos Externos (EIF): Grupos de datos que se mantienen externamente

Cálculos por puntos de función:

Valor de dominio de información	Factor ponderado			Total
	Baja	Media	Alta	
Entradas externas (EI)	2 x 3	2 x 4	4 x 6	38
Salidas externas (EO)	3 x 4	3 x 5	2 x 7	41
Consultas externas (EQ)	2 x 3	5 x 4	1 x 6	32
Archivos lógicos internos (ILF)	10 x 7	2 x 10	3 x 15	135
Archivos de interfaz externos (EIF)	0 x 5	0 x 7	0 x 10	0
Total				246

Cálculo de los Puntos de Función Sin Ajustar (PFSA):

$$PFSA = 38 + 41 + 32 + 135 + 0$$

$$PFSA = 246$$

Cálculo del Factor de Ajuste (FA):

No.	Factor de ajuste	Puntos
1	Comunicación de datos	1
2	Proceso distribuido	1
3	Objetivos de rendimiento	1
4	Configuración de explotación compartida	1
5	Tasa de transacciones	1
6	Entrada de datos en línea	2
7	Eficiencia con el usuario final	2
8	Actualizaciones en línea	1
9	Lógica de proceso interno compleja	2
10	Reusabilidad del código	2
11	Conversión e instalación contempladas	1
12	Facilidad de operación	2
13	Instalaciones múltiples	0
14	Facilidad de cambios	1
Factor de ajuste		17

Cálculo de los Puntos de Función Ajustados (PFA)

$$PFA = PFSA * [0.65 + (0.01 * FA)]$$

$$PFA = 246 * [0.65 + (0.01 * 17)]$$

$$PFA = 201,72$$

Cálculo del Esfuerzo

H/H: Horas hombre

$$H/H = PFA * \text{Horas PF promedio}$$

Para obtener las Horas PF promedio, nos apoyamos en la tabla siguiente:

Entorno y lenguaje	Líneas de código por PF	Horas por PF promedio
Lenguajes de 2GL: Ensamblador, C	0	0
Lenguajes de 3GL: Cobol	250	15
Lenguajes de 4GL: PHP	400	8

$$H/H = 201,72 * 9$$

$$H/H = 1815,48 \text{ Horas hombre}$$

Equipo XP	Jornada Laboral (h)	Horas de trabajo al proyecto por día (h)	Horas de trabajo al proyecto por semana (h)	Semanas de Trabajo por mes (u)	Total de horas (h)	Total de días laborables en el mes (días)
Luis Raúl Alfonso Caballero	8	6	30	4	120	24

Duración del proyecto en horas = (1948,32Horas/hombre) / 1 hombre = 1948,32 horas por miembro

Debido al framework que se utilizara se estima una reducción en la cantidad de horas de trabajo de 700 horas, al utilizarse herramientas que automatizan muchas partes del proceso como la creación de interfaces, clases controladoras y modelos de datos.

Contando con la experiencia del programador se estima que pueda utilizar 300 horas menos para realizar este trabajo

$$1 \text{ mes} = 140 \text{ horas laborables}$$

$$\text{Duración del proyecto en meses} = 948.32 \text{ horas/hombre} / (140 \text{ horas/mes}) = 6 \text{ meses}$$

Cálculo del Presupuesto del Proyecto

Sueldo mensual 1 participante = 4000.00 CUP

Costo Total del proyecto = sueldo 1 participante * cantidad participantes * tiempo de desarrollo

Costo Total del Proyecto = 4000.00 * 1 * 6

Costo Total del Proyecto = 24000.00 CUP

2.7 Conclusiones del capítulo.

En este capítulo se da una muestra de cómo fue el desarrollo del software mediante la ingeniería de este. Con las diferentes tablas se fue describiendo el proceso, para un mejor entendimiento de las funcionalidades del mismo. También se estimó un costo aproximado, el cual pudo haberle costado a la empresa si se lo desarrolla un equipo profesional. Por lo que es más eficiente la manera en que se desarrolló.

3 Propuesta de solución práctica al problema científico

3.1 Introducción

Para solucionar la problemática de la creación un sistema web para la gestión de indicadores de las distintas entidades y los procesos del proyecto varadero ciudad digital, se desarrolló utilizando la metodología XP un producto digital libre siguiendo las directivas para la soberanía tecnología por la que el país se encuentra abogando, que cumple con las funciones específicas necesitadas por el cliente.

3.2 Descripción de la solución

Para dar solución a la problemática se desarrolló un sistema que registra las entidades o procesos con su información básica para reconocerla, a la misma se le asocian variables con las informaciones exigidas en los formularios de la misma y se conforman los indicadores asociándole las variables. En una interfaz aparte se permitirá la entrada de valores a las variables con un control para asignar la fecha del registro.

3.3 Descripción del software

En el anterior capítulo se planteó la propuesta de solución de esta investigación, o sea la aplicación web para la creación de cuadros de mando de indicadores. A continuación, se mostrarán algunas de las interfaces del software y se argumentará la funcionalidad que cumplirá cada una.



Ilustración 3 Formulario de inicio de sesión

Descripción: Es donde el cliente ingresa su usuario y contraseña para acceder al sistema

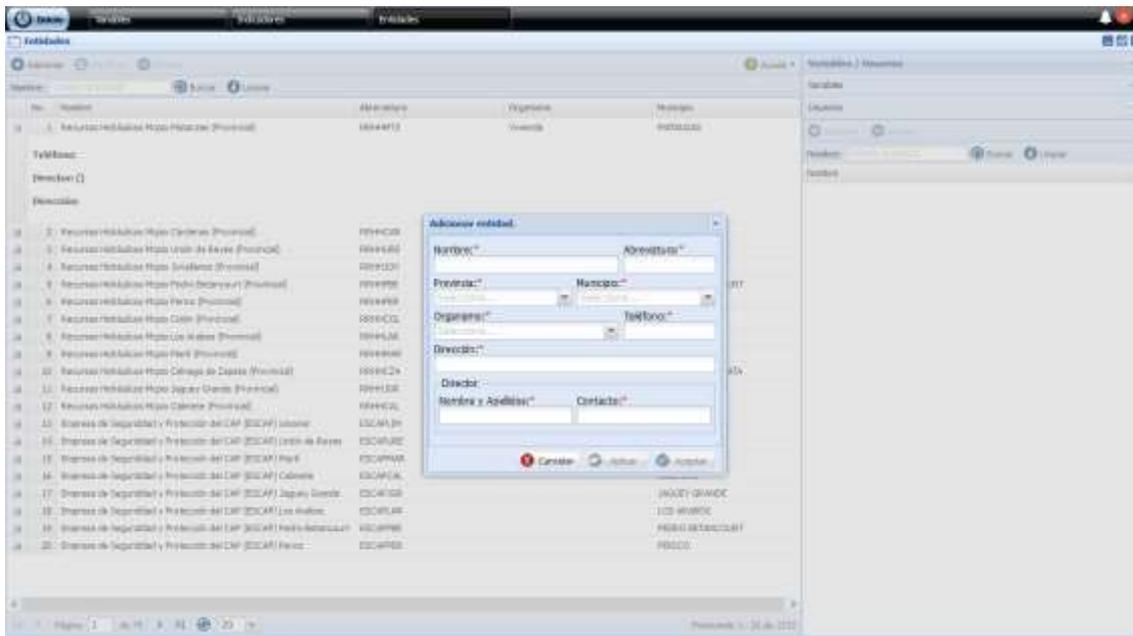


Ilustración 4 Formulario de creación de una entidad

Descripción: Es donde el cliente ingresa las informaciones necesarias para la creación de una entidad

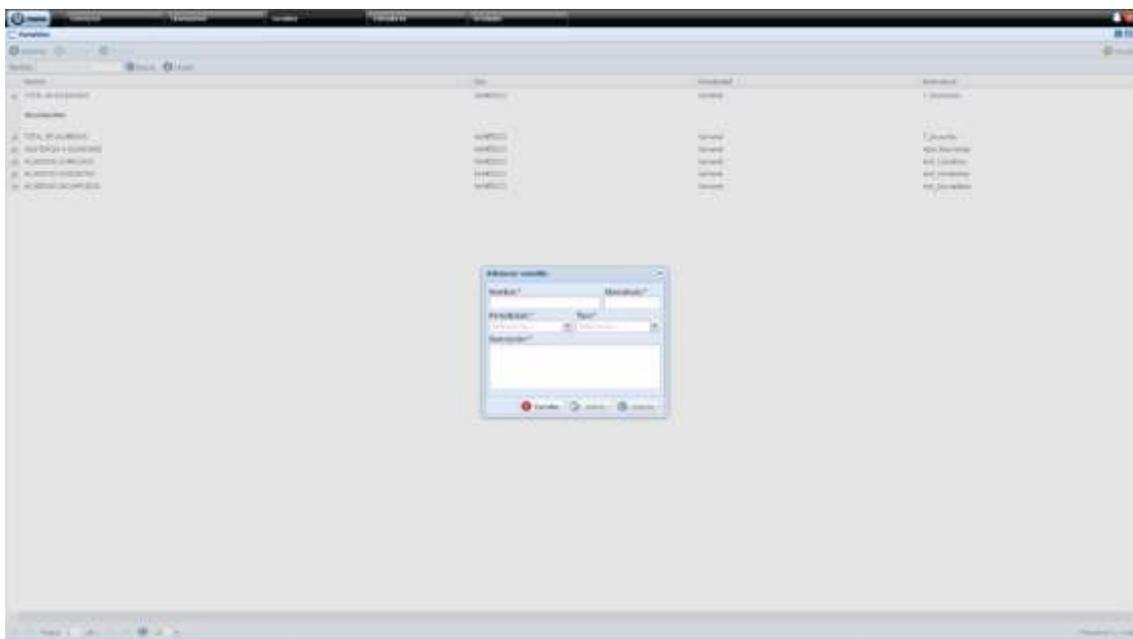


Ilustración 5 Formulario de creación de una variable

Descripción: Es donde el cliente ingresa los datos necesarios para la creación de una variable

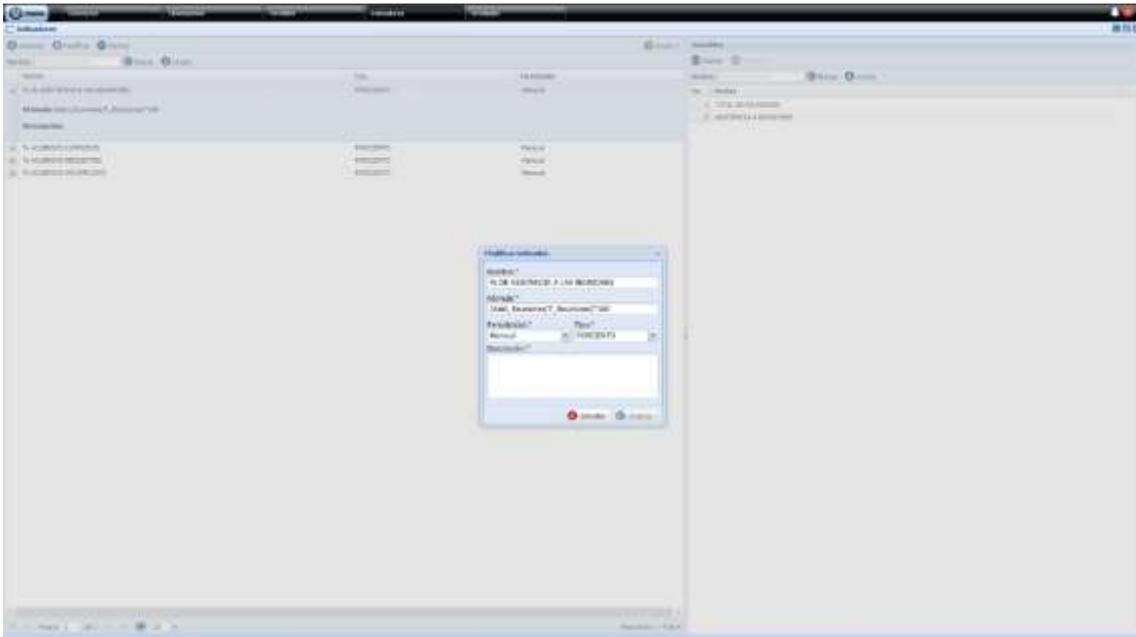


Ilustración 6 Formulario de creación de un indicador

Descripción: Es donde el cliente los datos necesarios para la creación de un indicador.

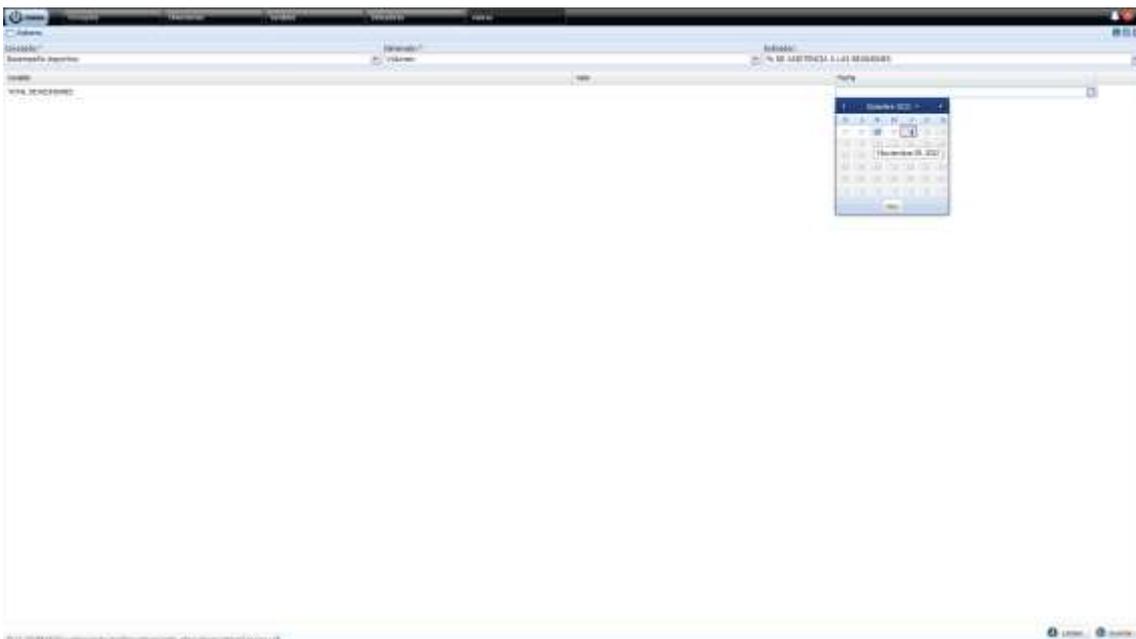


Ilustración 7 Interfaz de entrada de valores

Descripción: Es la interfaz que le permite al cliente la entrada de los valores a las variables y le permite establecer la fecha en la q se estudió el valor.

3.4 Pruebas de software

Las pruebas de software son el artefacto que permite verificar el correcto funcionamiento de los módulos del sistema, a continuación, se exponen la estrategia de pruebas a cada historia de usuario:

Pruebas de Aceptación	
Código:01	Historia de usuario: Inicio de sesión.
Nombre: Prueba de Inicio de Sesión.	
Descripción: Se comprueba que el cliente puede autenticarse en el sistema con sus credenciales.	
Condiciones de ejecución: Encontrarse en la página de inicio de sesión del sistema.	
Entrada: El cliente introduce las credenciales correctas	
Resultado esperado: El sistema inicia una sesión con el usuario introducido por el cliente.	
Resultado obtenido: El sistema inicia la sesión satisfactoriamente.	

Pruebas de Aceptación	
Código:02	Historia de usuario: Creación de una variable.
Nombre: Prueba de creación de una variable.	
Descripción: Se comprueba que el cliente pueda crear una variable sin que ocurra ningún error.	
Condiciones de ejecución: Tener una sesión iniciada y poseer los permisos de creación de variables.	
Entrada: El cliente introduce la información de la variable y pulsa el botón aceptar para crearla.	
Resultado esperado: El sistema crea la variable registrándola en la base de datos y muestra una notificación al cliente de la creación de la misma.	
Resultado obtenido: El sistema creó la variable registrándola en la base de datos y mostró una notificación al cliente de la creación de la misma.	

Pruebas de Aceptación	
Código:03	Historia de usuario: Creación de un indicador.
Nombre: Prueba de creación de un indicador.	
Descripción: Se comprueba que el cliente pueda crear un indicador sin que ocurra ningún error.	
Condiciones de ejecución: Tener una sesión iniciada y poseer los permisos de creación de indicadores.	
Entrada: El cliente introduce la información del indicador y pulsa el botón aceptar para crearlo.	
Resultado esperado: El sistema crea el indicador registrándolo en la base de datos y muestra una notificación al cliente de la creación del mismo.	
Resultado obtenido: El sistema creó el indicador registrándolo en la base de datos y mostró una notificación al cliente de la creación del mismo.	

Pruebas de Aceptación	
Código:04	Historia de usuario: Asociar variable a entidad.
Nombre: Prueba de asociar variable a entidad.	
Descripción: Se comprueba que el cliente pueda asociar una variable a una entidad sin ningún tipo de error.	
Condiciones de ejecución: Tener una sesión iniciada y poseer los permisos de asociar variables a entidades.	
Entrada: El cliente desde la interfaz que muestra las entidades selecciona la entidad a la que se va a asociar una variable y presiona el botón asociar, selecciona la variable que desea asociar a la entidad y presiona aceptar.	
Resultado esperado: El sistema asocia la variable a la entidad registrando la relación en la base de datos y envía una notificación al cliente para informar.	
Resultado obtenido: El sistema asoció la variable a la entidad registrando la relación en la base de datos y envió una notificación al cliente para informarlo.	

3.5 Resultados de las pruebas realizadas

Luego de la realización de todas las pruebas previstas el resultado fue positivo, se desplegaron las pruebas para cada historia de usuario marcada como de alta prioridad por el cliente obteniendo y se comprobó con el cliente las funcionalidades del sistema siendo valoradas positivamente por este.

3.6 Conclusiones parciales

En este capítulo se expuso la solución dada al problema científico. Se argumentó las decisiones tomadas en el proceso de desarrollo, así como se proporcionó una descripción detallada de los elementos más importantes del sistema. Se realizó la estrategia de pruebas y posteriormente se ejecutaron dichas pruebas en el sistema arrojando los resultados esperados

4 Conclusiones

Esta investigación le presento al proyecto “Varadero Ciudad Digital”, la solución a los distintos inconvenientes que se encontraban a la hora de medir y supervisar el desarrollo de las entidades y procesos que encierra el mismo. Utilizando la metodología XP y las tecnologías utilizadas por la empresa XETID para el desarrollo, luego de investigar sobre las bases teóricas de cuadros de mando de indicadores, se logró la producción de un software que permita la creación de estos. Aplicando un proceso de pruebas de aceptación se logró validar el funcionamiento de este y en las interacciones con el usuario se comprobó que cumple con los objetivos de la investigación y las necesidades del mismo.

5 Bibliografía

- Canales, A., Pena, A., Peredo, R., Sossa, H., & Gutierrez, A. J. E. s. w. a. (2007). Adaptive and intelligent web based education system: Towards an integral architecture and framework. *33*(4), 1076-1089.
- Chaffer, J., & Swedberg, K. (2011). *Learning jQuery*: Packt Publishing Ltd.
- Cobián-Alvarado, M., Damián-Reyes, P., Andrade-Aréchiga, M., JRG, P., & Acosta-Díaz, R. (2016). Combatiendo la deserción de estudiantes universitarios con el uso de tecnologías de información.
- De la Torre Llorente, C., Castro, U. Z., Barros, M. A. R., & Nelson, J. C. (2010). Guía de Arquitectura N-Capas orientada al Dominio con .NET. In: España: Microsoft Iberica.
- Del Giudice, M., Soto-Acosta, P., Carayannis, E., & Scuotto, V. J. B. p. m. j. (2018). Emerging perspectives on business process management (BPM): IT-based processes and ambidextrous organizations, theory and practice. *24*(5), 1070-1076.
- Hodgkinson-Williams, C., & King, T. (2018). Researching OER in the open: Developments in the ROER4D project.
- Musciano, C., & Kennedy, B. (2002). *HTML & XHTML: The Definitive Guide: The Definitive Guide*: " O'Reilly Media, Inc."
- Orozco Santa María, A. M., & García Ramírez, M. J. C.-e. R. d. I. E. (2017). Autopercepción de habilidades de aprendizaje en ambientes virtuales. (25), 144-167.
- Sanz. (2010). Pruebas de aceptación en Sistemas Navegables. *Revista Española*.
- Yáñez, C. J. I. e. (2008). Sistema de gestión de calidad en base a la norma ISO 9001. *9*, 1-9.

6 Anexos

Historia de usuario	
Numero:6	Usuario: Especialista
Nombre historia: Asociar variable a entidad	
Prioridad en negocio:1	Riesgo de desarrollo:2
Puntos estimados:2	Iteraciones asignadas:1
Programador responsable: Luis Raúl Alfonso	
Descripción: El especialista después de configurar las variables accede a la interfaz de entidades y asocia cada una de las variables con la entidad que le corresponde a través de los controles designados en la aplicación con estas funciones.	
Validación:	
-Asociación Satisfactoria: El sistema asocia la variable a la entidad registrándola en la base de datos y enviando un mensaje de confirmación al cliente	
- Asociación Fallida: El sistema no logra asociar la variable a la entidad y envía un mensaje de error al cliente	

Historia de usuario	
Numero:7	Usuario: Administrador total del sistema
Nombre historia: Asociar usuario a entidad	
Prioridad en negocio:1	Riesgo de desarrollo:2
Puntos estimados:2	Iteraciones asignadas:1
Programador responsable: Luis Raúl Alfonso	
Descripción: El administrador del sistema después de configurar las entidades asocia a cada una de los usuarios que tendrán acceso a trabajar con ellas, a través de los controles designados en la aplicación con estas funciones.	
Validación:	
-Asociación Satisfactoria: El sistema asocia el usuario a la entidad registrándolo en la base de datos y enviando un mensaje de confirmación al cliente	
- Asociación Fallida: El sistema no logra asociar el usuario a la entidad y envía un mensaje de error al cliente	

Historia de usuario	
Numero:8	Usuario: Especialista

Nombre historia: Asignar valor a variable	
Prioridad en negocio:1	Riesgo de desarrollo:2
Puntos estimados:2	Iteraciones asignadas:1
Programador responsable: Luis Raúl Alfonso	
Descripción: El usuario accede a la interfaz de introducir valores y accede a las variables según la entidad que tiene asociada, ingresa los nuevos valores de las variables y la fecha de recogida del dato y presiona el botón guardar.	
Validación: -Guardado Correcto: El sistema consigue guardar el nuevo valor de la variable en la base de datos y envía un mensaje de confirmación. -Inicio de sesión permitido: : El sistema no consigue guardar el nuevo valor de la variable en la base de datos y envía un mensaje de error.	