



Universidad de Matanzas
Facultad de Ciencias Técnicas
Carrera Ingeniería Informática



Trabajo para optar por el Título de Ingeniero en Informática

Aplicación web para evaluar prerequisites higiénico-sanitarios en entidades gastronómicas del MINTUR en la provincia de Matanzas.

Autora: Lianet Hernández Verdecia

Tutores: MSc. Mayli Estopiñán Lantigua

MSc. Yadrían A. García Pulido

Matanzas, 2018

Pensamientos

Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallen tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dicen tu corazón y tu intuición. Ellos ya saben de algún modo en qué quieres convertirte realmente. Todo lo demás es secundario.

Steve Jobs

Dedicatoria

. A mi familia en general porque gracias a ellos he llegado hasta aquí

. A mi novio por estar ahí para mí en todo momento

. A mis padres por su apoyo incondicional y por hacer posibles mis sueños.

Agradecimientos

. A todos los profesores que han aportado su pequeño granito en mi formación, entre ellos Omar, Mayli, Pulido.

. A mis amigas Zuli, Heidi, Lorena y a todos mis compañeros de aula los quiero.

. A mi amigo Raúl muchas gracias.

Opinión del tutor

Resumen

La presente investigación se fundamenta en desarrollar una aplicación Web que permita realizar evaluaciones a las entidades gastronómicas del MINTUR, en la provincia de Matanzas, teniendo en cuenta indicadores de inversión, constructivos y de gestión que garanticen el cumplimiento de los prerrequisitos higiénicos-sanitarios para asegurar la inocuidad de los alimentos a través de la puesta en práctica del APPCC (Sistema de análisis de peligros y puntos críticos de control). Entre las facilidades que brinda la aplicación se encuentran la generación automática de reportes y estadísticas útiles para posteriores análisis; así como el almacenamiento de todas las evaluaciones que se realicen a las entidades para que puedan ser consultadas lo que permite se realicen con mayor rapidez las consultas y los estudios estadísticos. Para su desarrollo se utilizó la metodología ágil XP (Programación Extrema), como lenguaje de programación PHP con el framework Symfony y gestor de Base de datos MySQL.

Summary

The present investigation is based on the development of a Web application that allows to make evaluations to the gastronomic entities of MINTUR, in the province of Matanzas, taking into account investment, constructive and management indicators that guarantee the fulfillment of the hygienic-sanitary prerequisites to ensure food safety through the implementation of HACCP (Hazard Analysis System and Critical Control Points). Among the facilities provided by the application are the automatic generation of reports and useful statistics for further analysis; as well as the storage of all the evaluations that are made to the entities so that they can be consulted. For its realization the agile methodology XP (Extreme Programming) was used, the PHP programming language with Symfony framework was used as tools. The MySQL Database Manager was selected.

Índice

Contenido

Introducción.	13
Capítulo 1: Marco teórico-referencial.....	17
1 Objeto de estudio.....	17
1.1 Caracterización del MINTUR.	17
1.1.2 Descripción de los conceptos y procesos fundamentales a emplear.	18
1.2 Flujo actual del proceso involucrado en el campo de acción.	19
1.2.1 Análisis crítico de la ejecución actual de este proceso y las causas que originan la situación problemática, así como sus consecuencias.	20
1.3 Antecedentes del trabajo.....	20
1.4 Metodología de la investigación empleada.	22
1.4.1 Métodos teóricos empleados.	22
1.4.2 Métodos empíricos empleados.	23
1.5 Fundamentación de la metodología utilizada.	23
1.5.1 Arquitectura cliente-servidor.	25
1.5.2 Aplicaciones web.	26
1.5.3 Patrones de diseño.	27
1.5.4 Lenguajes de programación.	28
1.5.5 Sistema de gestión de bases de datos.	33
1.5.6. Servidor web.....	34
1.6 Conclusiones parciales del capítulo.	35
Capítulo 2: Análisis, diseño y construcción de la solución propuesta.	37
2.1 Introducción.	37

2.2 Etapa de Planificación.	37
2.2.1 Equipo de trabajo y roles.	37
2.2.2 Historias de Usuario iniciales.	38
2.2.3 Planificación de iteraciones.	47
2.2.4 Plan de entregas.....	47
2.3 Etapa de diseño.	48
2.3.1 Prototipo de interfaz de usuario.	48
2.3.2 Tareas a desarrollar.	49
2.3.3 Tareas de iteración.	50
2.3.4 Diseño de la base de datos	55
2.3.5 Tarjetas de Clase, Responsabilidad y Colaboración.....	55
2.4 Conclusiones parciales del capítulo.	56
Capítulo 3: Validación de la Solución Propuesta	58
3.1 Pruebas	58
3.1.1 Pruebas de aceptación.....	58
3.2 Beneficios tangibles e intangibles. Estimación del costo.....	64
3.3 Conclusiones parciales del capítulo.	64
Conclusiones.....	66
Bibliografía	67

Índice de Ilustraciones

Ilustración 1 Plan de Iteraciones (Elaboración propia) _____	47
Ilustración 2 Plan de Entregas (Elaboración propia) _____	48
Ilustración 3 Vista Principal _____	49
Ilustración 4 Base de Datos _____	55
Ilustración 5 Se muestra en rojo un mensaje de error al autenticarse. _____	62
Ilustración 6 Se muestra el recuadro oscuro en la parte superior derecha anunciando la correcta eliminación de la ayuda. _____	63

Índice de tablas

Tabla 1 Equipo de trabajo y roles (Elaboración Propia)	38
Tabla 2 Historias de usuario iniciales (Elaboración propia)	39
Tabla 3 HU 1.....	40
Tabla 4 HU 2.....	41
Tabla 5 HU 3.....	41
Tabla 6 HU 4.....	42
Tabla 7 HU 5.....	42
Tabla 8 HU 6.....	43
Tabla 9 HU 7.....	44
Tabla 10 HU 8.....	44
Tabla 11 HU 9.....	45
Tabla 12 HU 10	45
Tabla 13 HU 11	46
Tabla 14 Tareas de Iteración (Elaboración propia)	49
Tabla 15 TI 1	50
Tabla 16 TI 5	51
Tabla 17 TI 7	51
Tabla 18 TI 9	52
Tabla 19 TI 13	52
Tabla 20 TI 17	52
Tabla 21 TI 18	53
Tabla 22 TI 20	53
Tabla 23 TI 21	53
Tabla 24 TI 22	54
Tabla 25 TI 23	54
Tabla 26 Tarjeta CRC Usuarios	55
Tabla 27 Tarjeta CRC Criterios	56
Tabla 28 PA 1	60
Tabla 29 PA 2	60
Tabla 30 PA 3	61
Tabla 31 PA 6	62

Tabla 32 PA 8	63
---------------------	----

Introducción.

En los inicios del procesado de información, con la informática sólo se facilitaban los trabajos repetitivos y monótonos del área administrativa. La automatización de esos procesos trajo como consecuencia directa una disminución de los costos y un incremento en la productividad. El desarrollo acelerado en el mundo de las TIC (Tecnologías de la Información y las Comunicaciones) ha provocado toda una revolución.

Las TIC se aplican a numerosas áreas del conocimiento y a la actividad humana, como, por ejemplo: gestión de negocios, almacenamiento y consulta de información, monitorización y control de procesos, industria, robótica, comunicaciones, control de transportes, investigación, desarrollo de juegos, diseño computarizado, aplicaciones/herramientas multimedia, medicina, biología, física, química, meteorología, ingeniería, arte, etc. Actualmente es difícil concebir un área que no use, de alguna forma, el apoyo de la informática. Ésta puede cubrir un enorme abanico de funciones, que van desde las más simples cuestiones domésticas hasta los cálculos científicos más complejos.

La información es un elemento fundamental para el desarrollo, con el de cursar de los años, la gestión de la información ocupa, cada vez más, un espacio mayor en la economía de los países a escala mundial. Se plantea la necesidad inmediata de implantar modelos para la gestión de la calidad total en las instituciones de información, existiendo una estrecha relación entre gestión de la información, del conocimiento y de la calidad en el quehacer de una organización.

En este sentido, Cuba ha planteado la necesidad de dominar e introducir en la práctica social las Tecnologías de la Información y las Comunicaciones, de lograr una cultura digital como una de las características imprescindibles del hombre nuevo, lo que facilitaría a la sociedad acercarse más, al objetivo de un desarrollo sostenible. Al

desarrollo de la informática se le presta gran importancia y se realizan amplios esfuerzos para poner las tecnologías al alcance de todos.

En la provincia de Matanzas son múltiples los avances alcanzados en cuanto a informatización, sin embargo, falta mucho por trabajar para sustituir las viejas técnicas por nuevos sistemas que brinden mayor seguridad, eficiencia y sobre todo ahorren tiempo.

Una de las empresas que tiene entre sus líneas estratégicas la automatización de sus procesos es el MINTUR, organización que dentro del territorio tiene una marcada importancia por ser el turismo un sector que genera considerables ingresos al país y constituye la actividad económica fundamental del territorio matancero. La gastronomía es un elemento esencial del turismo y resulta imposible pensar en turismo sin asociarlo a la gastronomía pues, independientemente del motivo o de la duración del viaje, la alimentación es parte integrante y fundamental. Al MINTUR pertenecen gran cantidad de centros que laboran alimentos a los que se le realizan evaluaciones que determinan si se cumplen o no los prerrequisitos higiénicos-sanitarios, esta evaluación determina si esta apta la entidad para ofrecer un buen servicio y cumplir las expectativas de los clientes.

Luego de estudios hechos a las evaluaciones mencionadas anteriormente se identificó como situación problemática en el MINTUR, que los análisis que se hacen de los resultados que arrojan estas no son eficiente en el cálculo del índice global de cumplimiento además de que sus resultados no se guardaban de manera persistente por lo que se corría el riesgo de la pérdida de esta información, siendo así muy difícil la recuperación de resultados de evaluaciones realizadas anteriormente para su posterior análisis y toma de decisión.

De acuerdo a lo planteado anteriormente, se identificó el siguiente **problema científico:** ¿cómo apoyar el proceso de evaluación de los

prerrequisitos higiénicos-sanitarios en entidades gastronómicas para contribuir a la seguridad alimentaria en las mismas?

El proceso de evaluación de los prerrequisitos higiénicos-sanitarios de los centros gastronómicos del MINTUR constituye el **objeto de estudio**.

El **objetivo general** de la investigación es, desarrollar una aplicación web que ayude a la evaluación de los prerrequisitos higiénicos-sanitarios de manera tal que contribuya a la seguridad alimentaria en las entidades gastronómicas del MINTUR.

Como **campo de acción** se tiene la automatización de las evaluaciones que se realizan a los centros gastronómicos del MINTUR.

Teniendo en cuenta el problema se propone trabajar en la siguiente **hipótesis**: Si se desarrolla una aplicación web para la evaluación de los prerrequisitos higiénicos-sanitarios entonces se apoyará al desarrollo eficiente del proceso de las evaluaciones contribuyendo a la seguridad alimentaria en las entidades gastronómicas del MINTUR.

Se determina como variable independiente la aplicación web para la evaluación de los prerrequisitos higiénicos-sanitarios y como variable dependiente el proceso de las evaluaciones para contribuir a la seguridad alimentaria en las entidades gastronómicas del MINTUR

Los objetivos específicos son:

- Realizar un estudio del marco teórico referencial del proceso de evaluación de los prerrequisitos higiénicos-sanitarios en centros gastronómicos del MINTUR.
- Seleccionar las herramientas y la metodología apropiadas para el desarrollo de la aplicación web.
- Analizar y diseñar la aplicación web que se propone.
- Validar la propuesta, por medio de las pruebas que propone la metodología de desarrollo de software seleccionada y el cumplimiento de los requerimientos funcionales.

Para desarrollar el software se emplean diversos métodos los cuales se dividen en métodos teóricos: histórico-lógico, analítico-sintético, inductivo-deductivo y en métodos empíricos: observación, entrevistas y análisis de documentos.

Se utiliza una metodología de desarrollo de software ágil, Programación Extrema (XP), que permite crear Historias de Usuarios (HU) en colaboración con el cliente.

Con el desarrollo del sistema informático para la evaluación de los prerequisites establecidos se obtendrán beneficios tales como la disminución del tiempo de procesamiento de los resultados generales de cada evaluación para cumplir con los plazos establecidos, la reducción de errores de cálculo que pueden perjudicar directamente el prestigio de los evaluadores además del ahorro de recursos al tener la información almacenada de forma segura y solo proceder a la impresión de los informes necesarios. Como **resultados esperados** se pretende que el MINTUR cuente con una herramienta eficiente para realizar la evaluación de los prerequisites higiénicos-sanitarios evitando errores y agilizando el proceso.

La investigación quedará estructurada de la siguiente forma:

Capítulo 1 - Marco Teórico – Referencial, se plantean las definiciones principales referentes a la evaluación de los prerequisites higiénicos-sanitarios, se hace un estudio sobre las tendencias y tecnologías actuales, se caracterizan los sistemas de bases de datos, las metodologías ágiles de desarrollo de software además de los diferentes lenguajes de programación. Son definidas las herramientas que serán utilizadas y se justifica su uso.

Capítulo 2- Análisis, Diseño de la Solución Propuesta y Construcción, se argumenta la solución que se propone al problema de investigación, presentando una planificación inicial del proyecto, con el empleo de la metodología ágil de desarrollo de software

Programación Extrema. Se realiza además un estudio de los beneficios tangibles e intangibles como resultado de la realización del proyecto de software. Se construye la solución propuesta, presentando una planificación por iteraciones.

Capítulo 3- Validación de la Solución Propuesta, se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio de los clientes y los propios de la metodología de software.

Conclusiones, se redactan las conclusiones generales

Bibliografías, quedan recogidas las bibliografías empleadas para el desarrollo de todo el trabajo.

Capítulo 1: Marco teórico-referencial.

En este capítulo se analizan las bases teóricas que sustentan la investigación y elementos que facilitan el entendimiento del objeto de estudio, así como trabajos relacionados con estos procesos. Se tratan también las diferentes metodologías empleadas, los métodos de investigación estudiados en Ingeniería del Software, se fundamenta la elección de las tecnologías a utilizar y algunos conceptos generales básicos para el desarrollo del trabajo.

1 Objeto de estudio.

1.1 Caracterización del MINTUR.

El Ministerio del turismo es un organismo estatal, rector del sistema de turismo, en el cual participan otras entidades del país. En este sentido el MINTUR elabora la política y controla su aplicación en las entidades que administran directamente las propiedades del sector. Entre sus objetivos estratégicos se destacan: diseñar y concretar una comercialización eficiente del producto turístico, incrementar y diversificar una oferta turística competitiva y sobre todo desarrollar hasta los niveles más avanzados la informática y los sistemas de comunicación(García, 2014).

1.1.2 Descripción de los conceptos y procesos fundamentales a emplear.

-Sistema de análisis de peligros y puntos críticos de control

Este sistema identifica, evalúa y controla los peligros que son significativos para la inocuidad de los alimentos, promueve la buena seguridad alimentaria al identificar los posibles peligros y riesgos y aplicar y supervisar medidas de control adecuadas en aquellos puntos críticos para la seguridad. Si se pone en práctica de forma adecuada, ayudará a evitar la contaminación y la intoxicación alimentaria, a la vez que las reclamaciones de los clientes. En virtud de la legislación de la UE, el APPCC ofrece una base efectiva y factible para una correcta higiene alimentaria en Europa. Además, es aceptado como una norma internacional para las buenas prácticas alimentarias y es un requisito legal en muchos países (ABTA, 2012).

El Plan de APPCC sólo es eficaz si los prerrequisitos funcionan correctamente. Es decir, aunque haya un Plan de APPCC bien diseñado, será difícil garantizar la seguridad de los productos alimenticios si no se siguen los planes de prerrequisitos.

-Prerrequisitos higiénicos-sanitarios.

Los prerrequisitos o planes generales de higiene son las prácticas y condiciones necesarias y esenciales para la Seguridad Alimentaria. Se presentan como planes, cada uno de ellos con la finalidad de cumplir un objetivo específico dirigido a reducir o eliminar peligros generales y, así, garantizar la Seguridad Alimentaria. Son de obligada implantación en todas las industrias y establecimientos alimentarios (Adalis, 2015).

-Indicadores que se avalúan para el cumplimiento de los prerrequisitos.

Los indicadores de gestión son aquellos relacionados con la manipulación, la gestión propiamente y los procesos, además

factibles de mejorar en un corto período de tiempo. Los de inversión se refieren a aquellos aspectos cuya solución implica compra de insumos, productos y/o materiales, reparaciones de envergadura y cuya solución puede requerir un plazo de tiempo mayor en algunos casos. Por último, los constructivos, están relacionados con la infraestructura de las instalaciones o que su mejora conlleva obras de construcción.

1.2 Flujo actual del proceso involucrado en el campo de acción.

El ministerio del turismo necesita evaluar el comportamiento de los prerrequisitos higiénico-sanitarios en todos los centros que elaboran y manipulan alimentos para garantizar la calidad de sus servicios. Estas entidades son evaluadas periódicamente y las evaluaciones que se realizan constan de 10 indicadores cada uno de estos con criterios específicos que miden con rigurosidad todos los aspectos necesarios e imprescindibles para la seguridad alimentaria. El cálculo del índice global de cumplimiento se realiza mediante la fórmula

$$IGHSm = \sum Ipn * Wn \quad Ipn = \frac{\sum Vn}{Vt}$$

Donde:

IGHSm: Índice global de cumplimiento de los prerrequisitos higiénico sanitarios modificado para la implantación del Sistema HACCP en la instalación objeto de estudio.

Ipn: Índice parcial del indicador

Wn: Peso del indicador

Vn: Valor otorgado al criterio

Vt: Total de puntos que podía alcanzar los criterios

Luego de realizar la evaluación a través de esta fórmula matemática se emite el estado de la entidad

IGHSm \geq 80 Bien

80 > IGHSm \geq 50 Regular

IGHSm < 50 Mal

1.2.1 Análisis crítico de la ejecución actual de este proceso y las causas que originan la situación problemática, así como sus consecuencias.

Actualmente estas evaluaciones se realizan de forma manual, dada la cantidad de criterios por cada indicador a evaluar el proceso se hace engorroso, a esto adicionemos la cantidad de recursos y tiempo que demora realizarlo para cada entidad perteneciente al MINTUR, los errores que pueden darse en los cálculos podrían afectar el índice de cumplimiento de los prerequisites higiénicos-sanitarios y además afectar el estado de la entidad tanto de forma específica como el comportamiento general de los indicadores al analizarse en conjunto todos los centros evaluados.

1.3 Antecedentes del trabajo.

El instrumento GHYCAL (Gestión de Higiene y Calidad) propuesto por Gutiérrez, Pastrana y Ramírez (2010) se encuentra contenido en 7 criterios, uno por cada uno de los planes previos de higiene y trazabilidad en un sistema de HACCP; se definieron 77 ítems que corresponden aquellos aspectos enmarcados en los 7 planes mencionados anteriormente que definen el programa de prerequisites de un sistema HACCP. Utiliza una escala discreta de cinco categorías, que se asignan según el nivel de cumplimiento de cada criterio evaluado.

El instrumento S&Q (*Safety and Quality on Food*) corresponde a una aplicación informática desarrollada en el Departamento de Tecnología de Alimentos de la Universidad Politécnica de Valencia, España, con el fin de evaluar preliminarmente la gestión de la higiene y la gestión de la calidad en empresas del sector agroalimentario. La sección

dedicada a la gestión de la higiene está estructurada en 6 grupos de preguntas que contienen aspectos relacionados con los edificios y su entorno, equipamiento, transporte y almacenamiento, control de aguas y limpieza, desinfección y desratización e higiene personal; en total son 75 preguntas que corresponden a una evaluación de los aspectos relevantes que permiten establecer los niveles de gestión de la higiene en cada empresa objeto de estudio (Gutiérrez, Pastrana, & Serra, 2010).

Los mismos autores plantean que estos dos instrumentos permiten la agrupación de acuerdo con los niveles de cumplimiento de gestión de la higiene alcanzados, expresados en valores porcentuales, mediante la aplicación del modelo estándar de auditoría de la calidad en las fases de producción y manipulación de los alimentos del protocolo *International Food Standard (IFS)*; una adaptación a la escala del IFS presentan tres grupos de empresas, así: nivel de cumplimiento bajo, que incluye empresas que obtienen evaluaciones entre el 0 y el 75%; nivel de cumplimiento básico, en el que se clasifican las empresas que obtienen evaluaciones entre el 75 % y el 95 %, y nivel de cumplimiento superior, para empresas que resulten con evaluaciones superiores al 95 %. En ambos casos las herramientas no proponen puntos invalidantes, no da importancia a cada uno de los aspectos a evaluar.

El IGHS propuesto por Cepero y García (2014), evalúa 10 indicadores y dentro de estos, sus criterios correspondientes, llegando a un total de 80; utiliza la escala de medición modificada del instrumento GHYCAL. Este no propone un índice general en fracción, solo toma los puntos otorgados según los pesos y lo suma. Para la cuestión de inocuidad se hace necesaria una decisión dicotómica (CUMPLE/NO CUMPLE) con los criterios planteados; eliminándose de esta forma la subjetividad asumida al otorgar los puntos por criterios, además de objetivar el análisis de los criterios al no establecer valores medios.

Luego de analizar con detenimiento los antecedentes investigativos se tomaron varios aspectos positivos de los documentos analizados ya que ninguno de estos resuelve la situación problemática.

1.4 Metodología de la investigación empleada.

La metodología de la investigación es una disciplina de conocimiento encargada de elaborar, definir y sistematizar el conjunto de técnicas, métodos y procedimientos que se deben seguir durante el desarrollo de un proceso de investigación para la producción de conocimiento. Orienta la manera en que se enfoca una investigación y la forma en que se recolectan, analizan y clasifican los datos, con el objetivo de que los resultados tengan validez y pertinencia, y cumplan con los estándares de exigencia científica. La metodología de la investigación, en este sentido, es también la parte de un proyecto de investigación donde se exponen y describen razonadamente los criterios adoptados en la elección de la metodología, sea esta cuantitativa o cualitativa (Monguel, 2005).

1.4.1 Métodos teóricos empleados.

El método de análisis histórico-lógico: permitió estudiar la trayectoria y desarrollo de los sistemas de evaluación de prerrequisitos existentes, se analizan las ventajas y desventajas y se toman como punto de partida para lograr mejoras. De esta manera encontrar una forma lógica de desarrollar el sistema informático deseado.

Los métodos de análisis y de 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 mental del todo en sus múltiples relaciones para luego unir las partes

analizadas, descubrir las relaciones y características generales entre ellas.

Inducción-deducción: 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.

1.4.2 Métodos empíricos empleados.

La observación científica: acompañó la investigación desde los primeros momentos, a través de la cual se conoció el estado y proceder de todo el proceso de evaluación de prerequisites higiénicos-sanitarios, y se obtuvo la información primaria acerca de los objetos investigados.

La entrevista: aportó datos esenciales a la investigación y fue de gran utilidad en varios momentos fundamentalmente al inicio ya que a través de este método se logró realizar correctamente el levantamiento de requisitos del software además de ayudar a concebir una interfaz agradable al usuario.

Análisis de documentos: la revisión de documentos por medio de los cuales se realiza la evaluación de los prerequisites higiénicos-sanitarios en entidades gastronómicas del MINTUR.

1.5 Fundamentación de la metodología utilizada.

La selección de la metodología a utilizar se hace sobre la base de las características del equipo de desarrollo, las necesidades específicas de la situación y las prioridades demandadas por el cliente. Para elegir una metodología de desarrollo de software se debe tener en cuenta dos factores fundamentales: el tipo de proyecto que se desea desarrollar y el tiempo que se dispone para desarrollar el mismo.

Actualmente existen dos grandes grupos de metodologías de desarrollo, las metodologías tradicionales y las metodologías ágiles;

las primeras se centran en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto, mientras que las segundas dan mayor importancia a la capacidad de respuesta a los cambios. A continuación, se presenta una breve comparación entre ellas:

Metodologías ágiles dan mayor valor a la colaboración con el cliente y al desarrollo incremental del *software* con iteraciones muy cortas. Se basa en la filosofía de que es más importante desarrollar *software* que funcione, que conseguir una buena documentación y es más importante responder ante un cambio, que seguir estrictamente un plan. Se destaca por su popularidad y fácil utilización la metodología *Extreme Programming (XP)*.

La metodología de desarrollo (XP) es una metodología para la ingeniería de software formulada por Kent Beck. Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.(Rodríguez, 2006).

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la Programación Extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. (Ferrer, 2002).

XP tiene como característica la comunicación directa y amplia con el cliente y entre los miembros del equipo de desarrollo, buena

respuesta ante los cambios sin importar la etapa del ciclo de vida de la aplicación donde se encuentre, se promueve la práctica de lo que brinde valor real al software antes que la documentación exhaustiva, la simplicidad en las soluciones y, generalmente los proyectos desarrollados bajo esta metodología tienen un ciclo de vida corto (Jiménez, 2010).

Por las bondades de esta metodología y lo ajustable al proyecto en cuestión se decide usar esta metodología en el desarrollo de la solución propuesta.

1.5.1 Arquitectura cliente-servidor.

La arquitectura cliente-servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor, al proceso que responde a las solicitudes.

Los principales componentes del esquema cliente-servidor son los Clientes, los Servidores y la infraestructura de comunicaciones. En este modelo, las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los Servidores proporcionan un servicio al cliente y devuelven los resultados. En algunos casos existen procesos auxiliares que se encargan de recibir las solicitudes del cliente, verificar la protección, activar un proceso servidor para satisfacer el pedido, recibir su respuesta y enviarla al cliente. Además, deben manejar los interbloqueos, la recuperación ante fallas, y otros aspectos afines (Álvarez, 2007).

Entre las principales características de la arquitectura cliente - servidor, que dan motivo para que se utilice en este proyecto se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz bien definida y única.
- Un servidor puede dar servicios a muchos clientes.
- El cliente proporciona la interfaz entre el usuario y el resto del sistema.
- El cliente necesita solamente conocer la interfaz externa del servidor y no su lógica.
- Los cambios en el servidor implican poco o ningún cambio en el cliente.

1.5.2 Aplicaciones web.

Son consideradas aplicaciones web a aquellas en las que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. Con la introducción de Internet y de las Aplicaciones Web, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web. Afortunadamente, se cuenta con herramientas potentes para realizar esto, ya que han surgido nuevas tecnologías que permiten que el acceso a una base de datos desde la Web, por ejemplo, un simple trámite.

Entre las principales ventajas para su empleo en este proyecto se pueden destacar:

- Ahorra tiempo: se pueden realizar tareas sencillas sin necesidad de descargar ni instalar ningún programa.

- Existen menos problemas de compatibilidad: basta tener un navegador actualizado para poder utilizarlas.
- No ocupan espacio significativo en el disco duro.
- Consumo de recursos bajo: dado que toda (o gran parte) de la aplicación no se encuentra en nuestro ordenador, muchas de las tareas que realiza el software no consumen recursos del cliente porque se realizan desde otro ordenador.
- Multiplataforma: se pueden usar desde cualquier sistema operativo porque sólo es necesario tener un navegador.
- Portables: es independiente del ordenador donde se utilice (un PC de sobremesa, un portátil...) porque se accede a través de una página web (sólo es necesario disponer de acceso a la red).
- La disponibilidad suele ser alta porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- Disminuye la probabilidad de que los virus dañen los datos porque éstos están guardados en el servidor de la aplicación.
- Colaboración: gracias a que el acceso al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene mucho sentido, por ejemplo, en aplicaciones online de calendarios u oficina.

1.5.3 Patrones de diseño.

Al conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo de software se les denomina patrones de diseño. (Gamma *et al.*, 2012).

Algunos patrones de diseño utilizados:

Modelo Vista Controlador (MVC): es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Inyección de Dependencias (en inglés *Dependency Injection, DI*) es un patrón de diseño orientado a objetos, en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto. El término fue acuñado por primera vez por Martin Fowler.

1.5.4 Lenguajes de programación.

PHP es un lenguaje de programación interpretado, originalmente diseñado para la creación de páginas web dinámicas. Sigue un estilo clásico ya que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. Está más cercano a JavaScript6 o al lenguaje C7, pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, lo que permite acceder a los recursos que este tenga, como es una base de datos y el resultado es enviado al navegador, el cual podría ser una página HTML o de cualquier otro tipo (Lerdorf, 2017).

PHP no necesita que el navegador lo soporte, es independiente de este, pero sin embargo para que sus páginas funcionen, el servidor donde están alojadas debe soportar este lenguaje, que tiene

numerables ventajas sobre otros lenguajes de programación que se ejecutan de igual manera (como podrían ser los script CGI Perl), permitiendo intercalar las sentencias PHP en las páginas HTML. Es el lenguaje ideal para el desarrollo de este proyecto.

HTML: Hypertext Markup Language. Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido). La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado.

No tiene nada que ver con un lenguaje de programación, quizás se parezca más al uso de un procesador de texto por la utilización de códigos de comienzo y finalizado de estilo. Además de resultar más sencillo, no se necesita ninguna herramienta de programación, compilador o similar, sino que cualquier editor de texto puede servir para realizar las páginas más sorprendentes (Álvarez, 2012).

Java Scripts: es un lenguaje basado en objetos, muy parecido al Java, pero mucho más sencillo dada su simplicidad sintáctica y su manejabilidad, por lo que casi siempre está dentro de una página HTML. Habitualmente, se utiliza para activar procesos o desarrollar algoritmos que hagan tareas sencillas, (como calculadoras, almanaques, control de errores para formularios, etc.) haciendo la web más viva y dinámica.

Como el HTML no permite activar procesos o desarrollar algoritmos, el Java Script viene muy bien para estas necesidades. No hay que olvidar que está muy limitado frente a lenguajes de programación tradicionales, sobre todo porque no permite la manipulación de

ficheros (para evitar problemas de seguridad). Esto ahorra al servidor mucho trabajo pues el código se interpreta en el cliente (desde su navegador). Combinado con el acceso a bases de datos da mucho juego, pero sobre todo se usa para darle vida a las webs haciendo que cambien imágenes, que se procesen datos, entre otros (Álvarez, 2012).

CSS (Cascading Style Sheets): las hojas de estilo en cascada son un mecanismo que permite aplicar formato a los documentos escritos en HTML (y en otros lenguajes estructurados, como XML) separando el contenido de las páginas de su apariencia. Para el diseñador, esto significa que la información estará contenida en la página HTML, pero este archivo no debe definir cómo será visualizada esa información. Las indicaciones acerca de la composición visual del documento estarán especificadas en el archivo de la CSS. Lo que posibilita crear páginas web de una manera más exacta. Gracias a esto el desarrollador es mucho más dueño de los resultados finales de la página (Eguiluz, 2017).

Framework: en el desarrollo de software es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otras aplicaciones para ayudar a desarrollar y unir los diferentes componentes de un proyecto, es decir, son soluciones completas que llevan incorporado herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución).

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (Pressman, 2010).

Symfony es un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. (Bowler & Bancner, 2009).

Symfony se utilizó porque se ajusta a los siguientes requisitos:

- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, está más indicado para grandes aplicaciones Web que para pequeños proyectos.
- Aunque utiliza MVC (Modelo vista controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.

Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional:

- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo. Código fácil de leer que incluye comentarios de *phpDocumentor* y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código ahorrándonos tiempo de trabajo.

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.

Sonata para administración

Bootstrap es un conjunto de herramientas gratuitas para crear sitios web y aplicaciones web. Contiene HTML y basados en CSS plantillas de diseño para tipografía, formas, botones, navegación y otros

componentes de la interfaz, así como las extensiones de JavaScript opcionales (Risueño, 2013).

Bootstrap es compatible con las últimas versiones de los principales navegadores. Su funcionamiento no es adecuado utilizarlo en navegadores antiguos como Internet Explorer 8.

Algunas de las características que dan motivo a su utilización:

- Desde la versión 2.0 también es compatible con el diseño de respuesta. Esto significa que el diseño de páginas web se ajusta dinámicamente, teniendo en cuenta las características del dispositivo utilizado (de escritorio, tableta, teléfono móvil).
- Bootstrap es de código abierto y está disponible en GitHub. Se anima a los desarrolladores a participar en el proyecto y hacer sus propias contribuciones a la plataforma.
- Recientemente, miembros de la comunidad han traducido la documentación del Bootstrap a varios idiomas, incluido el chino, español y ruso.

1.5.5 Sistema de gestión de bases de datos.

Los sistemas de gestión de bases de datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. La función general de un sistema gestor de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

MySQL: es un gestor de base de datos sencillo de usar e increíblemente rápido sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en

otras aplicaciones desarrolladas también en esos lenguajes. Es un sistema cliente-servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios y asegurando el acceso a solo los autorizados. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo! Finance, Google, Motorola, entre otras. Es gratis para aplicaciones no comerciales. Aprovecha la potencia de sistemas multiprocesadores, gracias a su implementación multi-hilo. Dispone de interfaces de programación de aplicaciones en gran cantidad de lenguajes (C, C++, Java, PHP). Soporta hasta 32 índices por tabla. Implementa una gestión de usuarios y contraseñas que proporciona un buen nivel de seguridad en los datos. Es Software libre (licencia GNU GPL) y Open Source. Trabaja sobre muchas plataformas, incluida Windows, Mac Os X Server, Solaris, Linux, y muchas plataformas de UNIX. Acepta bloqueos y roles de usuario (Martínez, 2014).

Se eligió MySQL, porque es uno de los servidores de bases de datos de código abierto más populares y conocidos del mundo, un sistema de manejo de bases de datos con un gran nivel de estabilidad y facilidad de desarrollo que se integra fácilmente con el lenguaje de programación PHP. Dispone, además, de una arquitectura que lo hace extremadamente rápido y fácil de personalizar. Sumándole a todos estos beneficios, que es un servidor que se adecúa perfectamente a las exigencias del cliente.

1.5.6. Servidor web.

Servidor Web: Apache HTTP Server El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que

implementa el protocolo HTTP y la noción de sitio virtual. Apache tiene amplia aceptación en la red: en el 2005, Apache fue el servidor HTTP más usado, siendo el servidor empleado en el 48% de los sitios Web en el mundo. Sin embargo, ha sufrido un descenso en su cuota de mercado en los últimos años. (Esta dísticas históricas y de uso diario proporcionadas por Netcraft). La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache (Falcón, 2006).

Por lo tanto, las características a destacar para la utilización de este servidor son:

- Es modular.
- El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor Web.
- Open-source. – Es un software distribuido y desarrollado libremente.
- Es gratuito.
- Multi-plataforma.

Presenta mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

1.6 Conclusiones parciales del capítulo.

Mediante la realización de este capítulo se analizaron las herramientas existentes para evaluar prerequisites. Se analizan y seleccionan las tecnologías que facilitarán el desarrollo de la aplicación. Se justifica la utilización de la metodología ágil XP, así

como el lenguaje de programación PHP, el *framework* Symfony, el gestor de bases de datos MySQL.

Capítulo 2: Análisis, diseño y construcción de la solución propuesta.

2.1 Introducción.

En el capítulo se exponen los elementos necesarios para la descripción de la solución propuesta. A través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, es llevado a cabo el análisis de los requerimientos. Por lo anteriormente mencionado se aplica la Metodología XP. Para determinar si es factible el desarrollo del software se realizará un estudio detallado.

2.2 Etapa de Planificación.

La planificación se realiza con el objetivo de lograr una eficiente organización del prototipo inicial del problema y proporcionar así un buen comienzo a una solución eficaz. Con este objetivo y según las ideas del cliente sobre el software se desarrollarán las Historias de Usuarios (HU), mediante la cual se obtendrá un punto de partida para el resto de la planificación del proyecto. Igualmente se realizará un estimado de cada una de las entregas del proyecto y del tiempo, basándose en que la planificación inicial se podría afectar debido a cambios que pudiesen sufrir estos aspectos durante el desarrollo del proyecto.

2.2.1 Equipo de trabajo y roles.

Según la metodología XP, se hace necesario ver como se aplican las prácticas para entender más la composición del equipo de trabajo. Estas traducen valores mencionados en el capítulo anterior en actividades que un programador debe realizar diariamente (Rodríguez, 2006).

A continuación se definen los roles del equipo de trabajo para el desarrollo del software, y los miembros involucrados en el desarrollo del mismo:

Tabla 1 Equipo de trabajo y roles (Elaboración Propia)

Miembros	Roles
Lianet Hernández Verdecia Maily Estupiñán Lantigua	Programadores
Lianet Hernández Verdecia Departamento de Calidad	Encargado de Pruebas
Jefe del Departamento de Calidad.	Cliente

2.2.2 Historias de Usuario iniciales.

Las HU ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Según (Fuentes, 2015) para el establecimiento de las historias se utilizan dos escalas nominales que exponen tres categorías alta, media y baja las cuales significan el riesgo y la prioridad en la escala de riesgo y prioridad respectivamente.

Escalas equivalentes a la prioridad en el negocio:

Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.

Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.

Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

Escala Nominal de Riesgo en Desarrollo:

Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.

Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.

Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales.

Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario.

Estas ayudan en la comunicación entre el cliente y los desarrolladores y pueden ir cambiando a medida que avanza el proyecto y que el cliente vea nuevas posibilidades y soluciones. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

Tabla 2 Historias de usuario iniciales (Elaboración propia)

No	Nombre	Prior.	Riesgo	Iterac.	Entreg.
----	--------	--------	--------	---------	---------

1	Diseño de la interfaz de usuario	Media	Bajo	1	1
2	Diseño y creación de la base de datos	Alta	Alto	1	1
3	Gestionar usuarios y roles	Alta	Medio	1	1
4	Gestionar entidades	Alta	Alto	2	2
5	Gestionar criterios	Alta	Alto	2	2
6	Gestionar ayudas	Alta	Medio	2	2
7	Gestionar indicadores	Alta	Alto	2	2
8	Evaluar criterios en indicadores	Alta	Alto	3	3
9	Generar gráficos	Alta	Alto	3	3
10	Calcular IGHSm	Alta	Alto	3	3
11	Generar reporte	Alta	Medio	3	3

A continuación, se muestran las HU en detalle para que se pueda comprender el proceso:

Tabla 3 HU 1

HISTORIA DE USUARIO	
Número: 1	Usuario: todos
Nombre de Historia: Diseño de la interfaz de usuario	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Lianet Hernández Verdecia	

Descripción:

Se muestra la vista de la interfaz principal en dependencia del rol del usuario que se ha registrado en el sistema.

Tabla 4 HU 2

HISTORIA DE USUARIO	
Número: 2	Usuario:
Nombre de Historia: Diseño y creación de la base de datos	
Prioridad en el negocio: Alta	Riesgo en el desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción:	
Se diseña e implementa la base de datos utilizando como gestor de base de datos MySQL.	

Tabla 5 HU 3

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador
Nombre de Historia: Gestionar Usuarios y Roles	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada:

	1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se permite al administrador gestionar los usuarios y asignar los roles correspondientes a cada uno de los usuarios del sistema.	

Tabla 6 HU 4

HISTORIA DE USUARIO	
Número: 4	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Gestionar entidad	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se puede insertar, modificar y eliminar las entidades que forman parte de la empresa.	

Tabla 7 HU 5

HISTORIA DE USUARIO

Número: 5	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Gestionar criterios	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se puede insertar, modificar y eliminar los criterios pertenecientes a cada indicador.	

Tabla 8 HU 6

HISTORIA DE USUARIO	
Número: 6	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Gestionar ayudas	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Lianet Hernández Verdecia	

Descripción:

Se puede insertar, modificar y eliminar las ayudas correspondientes a cada criterio.

Tabla 9 HU 7

HISTORIA DE USUARIO	
Número: 7	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Gestionar indicadores	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Lianet Hernández Verdecia	
Descripción:	
Se pueden modificar los indicadores.	

Tabla 10 HU 8

HISTORIA DE USUARIO	
Número: 8	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Evaluar criterios en los indicadores	
Prioridad en el negocio: Alta	Riesgo en

	desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Lianet Hernández Verdecia	
Descripción: Permite evaluar los criterios pertenecientes a cada indicador. Se le da valor numérico a los criterios según la decisión tomada (cumple/no cumple).	

Tabla 11 HU 9

HISTORIA DE USUARIO	
Número: 9	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Generar gráficos	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se generan los gráficos con los % de cumplimiento por cada uno de los indicadores evaluados.	

Tabla 12 HU 10

HISTORIA DE USUARIO	
Número: 10	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Calcular IGHm	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se realiza el cálculo del índice global de higiene a través de las fórmulas.	

Tabla 13 HU 11

HISTORIA DE USUARIO	
Número: 11	Usuario: Jefe del departamento de Calidad
Nombre de Historia: Generar reporte	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Lianet Hernández Verdecia	

Descripción:

Se genera un reporte en forma de tabla con los % de cumplimientos de cada indicador y el IGHSm así como el estado de la entidad y si existen invalidantes incumplidos.

2.2.3 Planificación de iteraciones.

En la ilustración 1 se muestran las tres iteraciones y las HU que le corresponden.

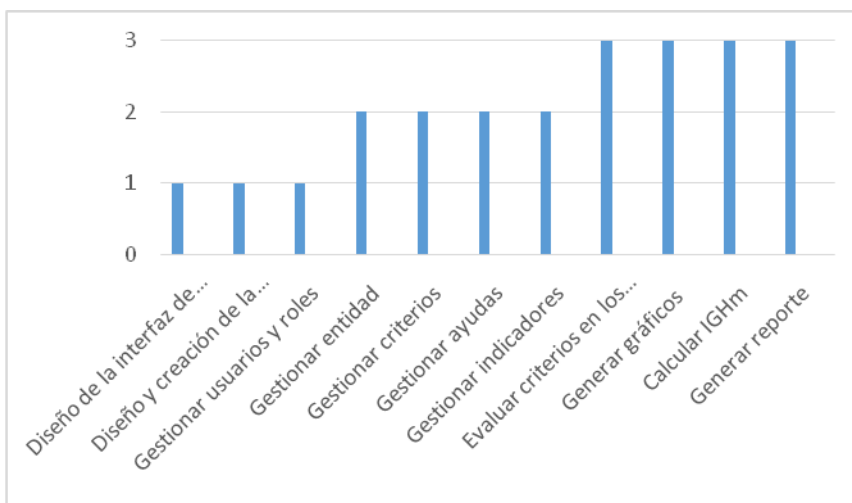


Ilustración 1 Plan de Iteraciones (Elaboración propia)

2.2.4 Plan de entregas.

La planificación de la entrega de acuerdo a la metodología XP propone agrupar una o varias iteraciones para que el cliente tenga versiones funcionales del sistema. En la ilustración 2 se muestra el Plan de Entregas

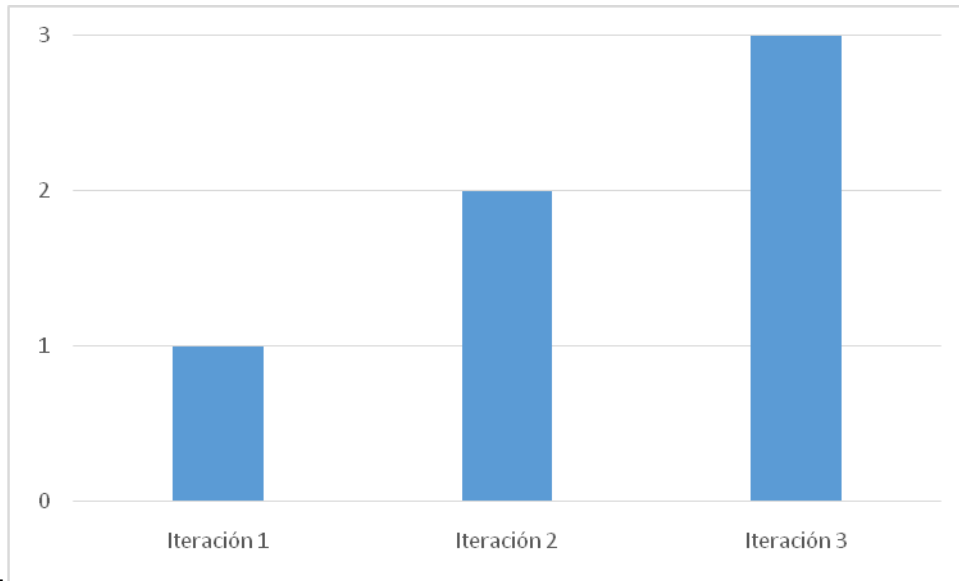


Ilustración 2 Plan de Entregas (Elaboración propia)

2.3 Etapa de diseño.

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo.

Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto.

2.3.1 Prototipo de interfaz de usuario.

A continuación, se muestran una ilustración de la interfaz de usuario, en el acápite de pruebas se pueden ver otras:

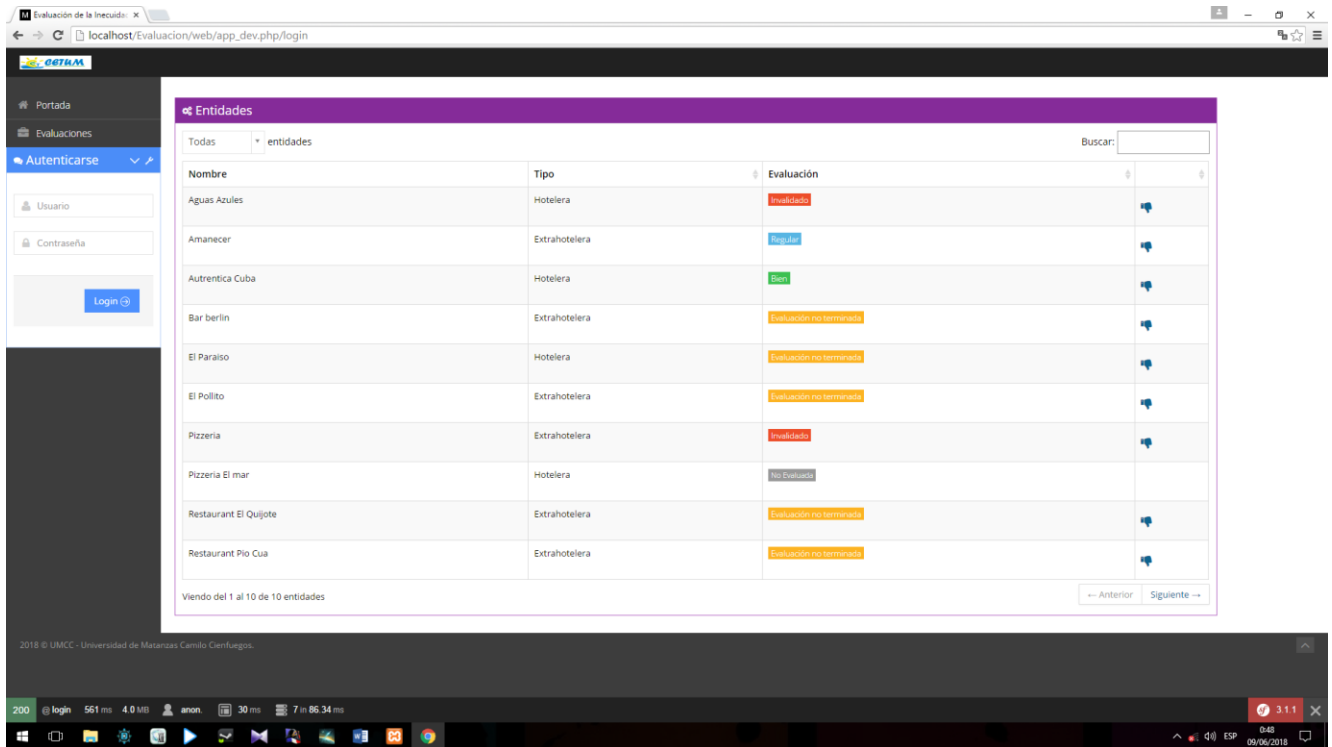


Ilustración 3 Vista Principal

2.3.2 Tareas a desarrollar.

Se muestra en la tabla la cantidad de tarea que corresponden a cada HU.

Tabla 14 Tareas de Iteración (Elaboración propia)

No	Nombre HU	No	Tarea de Iteración	Itrc
1	Diseño de la interfaz de usuario.	1	Interfaz Principal	1
		2	Interfaz Entidades	
		3	Otras Interfaces	
2	Diseño y creación de la base de datos.	4	Diseño de la base de datos	1
		5	Creación de la base de datos	
3	Gestionar usuarios y roles.	6	Gestionar Usuario	1
		7	Autenticarse	

		8	Gestionar Roles	
4	Gestionar entidad.	9	Insertar entidad	2
		10	Modificar entidad	
		11	Eliminar entidad	
5	Gestionar criterios.	12	Insertar criterios	2
		13	Modificar criterios	
		14	Eliminar criterios	
6	Gestionar ayudas.	15	Insertar ayudas	2
		16	Modificar ayudas	
		17	Eliminar ayudas	
7	Gestionar indicador	18	Modificar indicador	2
8	Evaluar criterios en indicadores	19	Evaluar criterios válidos	3
		20	Evaluar criterios invalidantes	
9	Generar gráficos	21	Generar gráficos	3
10	Calcular IGHSm	22	Calcular usando la fórmula	3
11	Generar reportes	23	Reporte General	3

2.3.3 Tareas de iteración.

A continuación, se relacionan algunas tareas de iteración que tenían mayor peso en el desarrollo de este proyecto:

Tabla 15 TI 1

TAREA DE ITERACIÓN

Número: 1	No Historia: 1
Nombre de Tarea: Interfaz principal	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Crear la interfaz visual principal de la aplicación.	

Tabla 16 TI 5

TAREA DE ITERACIÓN	
Número: 5	No Historia: 2
Nombre de Tarea: Creación de la Base de Datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se creó la Base de Datos empleando como herramienta el gestor MySQL	

Tabla 17 TI 7

TAREA DE ITERACIÓN	
Número: 7	No Historia: 3
Nombre de Tarea: Autenticarse	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se otorgan los permisos de acuerdo al tipo de rol autenticado en el	

sistema.

Tabla 18 TI 9

TAREA DE ITERACIÓN	
Número: 9	No Historia: 4
Nombre de Tarea: Insertar entidad	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se permite insertar una entidad en el sistema.	

Tabla 19 TI 13

TAREA DE ITERACIÓN	
Número: 13	No Historia: 5
Nombre de Tarea: Modificar criterios	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se permite modificar los criterios a evaluar.	

Tabla 20 TI 17

TAREA DE ITERACIÓN	
Número: 17	No Historia: 6
Nombre de Tarea: Eliminar ayudas	
Tipo de Tarea: Desarrollo	Puntos estimados: 1

Programador responsable: Lianet Hernández Verdecia

Descripción:

Se permite eliminar las ayudas.

Tabla 21 TI 18

TAREA DE ITERACIÓN

Número: 18

No Historia: 7

Nombre de Tarea: Modificar indicador

Tipo de Tarea: Desarrollo

Puntos estimados: 1

Programador responsable: Lianet Hernández Verdecia

Descripción:

Se permite modificar un indicador

Tabla 22 TI 20

TAREA DE ITERACIÓN

Número: 20

No Historia: 8

Nombre de Tarea: Evaluar criterios invalidantes

Tipo de Tarea: Desarrollo

Puntos estimados: 1

Programador responsable: Lianet Hernández Verdecia

Descripción:

Se evalúan los criterios invalidantes dándole valor 0 para que pueda calcularse el IGHm.

Tabla 23 TI 21

TAREA DE ITERACIÓN

Número: 21

No Historia: 9

Nombre de Tarea: Generar gráficos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se generan gráficos que muestran los % de cumplimiento de la entidad en cada indicador evaluado.	

Tabla 24 TI 22

TAREA DE ITERACIÓN	
Número: 22	No Historia: 10
Nombre de Tarea: Calcular IGHSm	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se calcula el índice global de cumplimiento utilizando la fórmula para conocer el estado de cada entidad.	

Tabla 25 TI 23

TAREA DE ITERACIÓN	
Número: 23	No Historia: 11
Nombre de Tarea: Generar reporte	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Lianet Hernández Verdecia	
Descripción: Se generan reportes con el resultado de la evaluación.	

2.3.4 Diseño de la base de datos

En la ilustración 4 se muestra el modelo de la base de datos que se diseñó y utilizó finalmente en el proyecto.

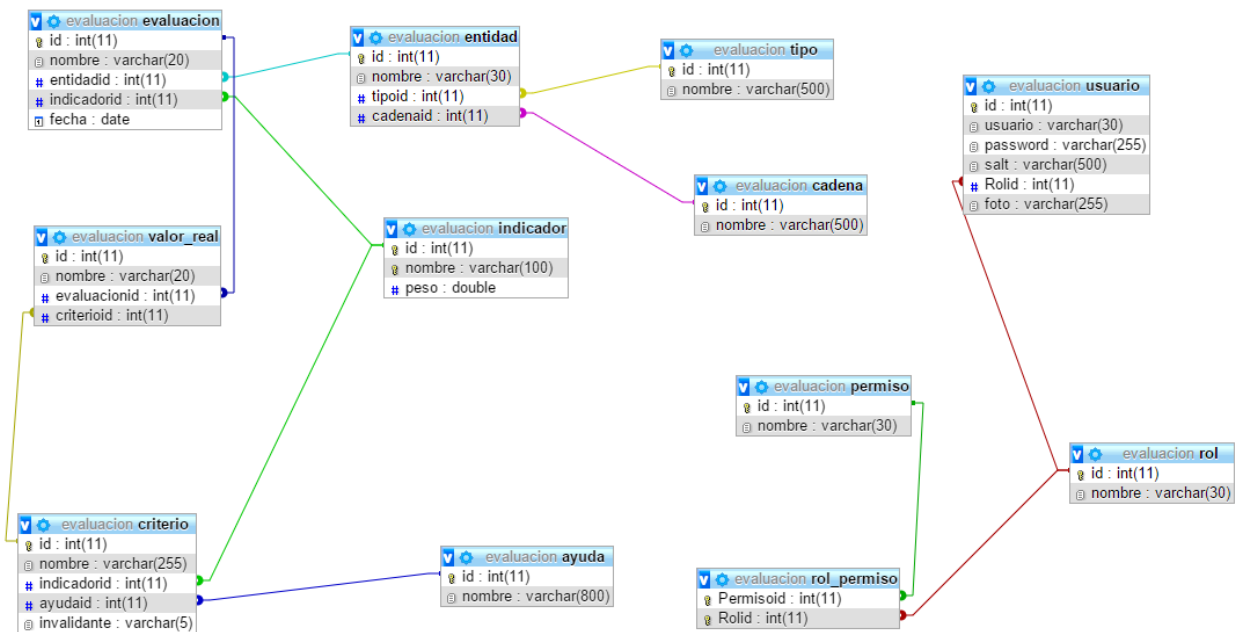


Ilustración 4 Base de Datos

2.3.5 Tarjetas de Clase, Responsabilidad y Colaboración.

A continuación, se muestran algunas tarjetas de Clase, Responsabilidad y Colaboración (CRC) creadas:

Tabla 26 Tarjeta CRC Usuarios

Usuarios
Superclase:
Subclase:
Descripción:
En esta clase se guardan los elementos referidos a los

usuarios.	
ATRIBUTOS	
idUsuario	int
nombreUsuario	String
Contraseña	String
Rol	String

Tabla 27 Tarjeta CRC Criterios

Criterios	
Superclase:	
Subclase:	
Descripción:	
En esta clase se guardan los elementos referidos a los criterios.	
ATRIBUTOS	
idCriterio	int
nombreCriterio	String
idAyuda	int
idIndicador	int
invalidante	String

2.4 Conclusiones parciales del capítulo.

En el presente capítulo se fundamenta la propuesta de solución y para ello se plantea el proceso de desarrollo de la aplicación, según lo establecido por la metodología XP, se muestran todas las etapas

excepto las pruebas funcionales. Se precisa el equipo de trabajo, sus características y los roles que cumple cada uno. Se realizan las historias de usuarios con el objetivo de determinar el comportamiento del sistema, servicios que este debe proporcionar y la especificación de los usuarios que interactúan en el sistema. Se ejecutan las tareas iniciales las cuales se agrupan en iteraciones y entregas. Se obtuvo un sistema informático en completo funcionamiento, después de realizadas las iteraciones.

Capítulo 3: Validación de la Solución Propuesta

En este capítulo se comprueba el cumplimiento de los requerimientos iniciales, se explican los casos de prueba, las estrategias de prueba y de igual forma se abordarán las pruebas de caja negra por medio de las cuales se realizaron las pruebas de aceptación de conjunto con el cliente y el análisis de los resultados obtenidos.

3.1 Pruebas

3.1.1 Pruebas de aceptación

Las pruebas de aceptación son las realizadas por el cliente y usuarios finales de la aplicación. Permiten probar las funcionalidades que exige el cliente. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

Las pruebas persiguen como objetivo, llevar a cabo el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar al menos un error no descubierto hasta entonces (Pressman R, 2010).

Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

- Código: Constituye un indicador de la prueba realizada y es sugerente al nombre de la prueba a la que hace referencia.
- HU: Contiene el nombre de la historia de usuario a la que hace referencia la prueba a realizar.
- Nombre: Nombre que se le asigna a la prueba a realizar.
- Descripción: Se detalla la funcionalidad que se desea probar.
- Condiciones de Ejecución: Muestra las condiciones necesarias y, que deben ser satisfechas antes de efectuar la ejecución del caso prueba para desarrollar el mismo y obtener los resultados esperados.

- Entradas / Pasos de Ejecución: Se describen los pasos seguidos en el desarrollo de la prueba, para ello se consideran todas las entradas que hace el usuario, con el propósito de conocer si se obtiene el resultado esperado.
- Resultado esperado: Se define el resultado que se espera obtener con la prueba realizada.
- Evaluación de la prueba: Una vez efectuada la prueba y obtenido su resultado se expone una valoración de la misma. Se clasifican en:
 - Satisfactoria: Cuando el resultado de la prueba es exactamente el esperado por el usuario.
 - Parcialmente satisfactoria: cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
 - No satisfactoria: Cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida también la HU.

Las pruebas de aceptación se llevan a cabo al efectuar los pasos siguientes:

- 1.** Se redactan los casos de prueba teniendo en cuenta el orden de las historias de usuario y los niveles de prioridad dados a las funcionalidades.
- 2.** Se planifica con el cliente de cuándo y cuáles pruebas serán llevadas a cabo.
- 3.** Se reúnen los miembros del proyecto seleccionados para realizar las pruebas.
- 4.** Se completan cada uno de los campos de las tablas de las pruebas de aceptación con el resultado de la prueba.

A continuación, se muestran varias pruebas de aceptación por su fuerte peso en el proyecto, alternadas con vistas de los mensajes del software que forman parte de la validación del proyecto:

Tabla 28 PA 1

PRUEBAS DE ACEPTACIÓN	
Número Caso de Prueba: 1	No Historia: 1
Nombre Caso de Prueba: Test Interfaz Principal	
Descripción: Verificar que se muestren las interfaces visuales implementadas	
Condiciones de ejecución: Esté corriendo la aplicación.	
Entradas: Interfaces de la aplicación.	
Resultado esperado: Se muestren las interfaces visuales de la aplicación.	
Evaluación: Prueba satisfactoria	

Tabla 29 PA 2

PRUEBAS DE ACEPTACIÓN	
Número Caso de Prueba: 2	No Historia: 2
Nombre Caso de Prueba: Test Base de Datos	
Descripción: Verifica el funcionamiento de la base de datos	
Condiciones de ejecución:	

Estar conectado a la base de datos.

Entradas:

Valores para leer o escribir en la base de datos por ejemplo: indicadores mensuales, usuario, contraseña, etc.

Resultado esperado:

Se muestran o guardan los datos correctamente.

Evaluación:

Prueba satisfactoria

Tabla 30 PA 3

PRUEBAS DE ACEPTACIÓN

Número Caso de Prueba:3

No Historia: 3

Nombre Caso de Prueba: Test Autenticar

Descripción:

Verificar que se autentifique un usuario correctamente en el sistema.

Condiciones de ejecución:

Estar conectado a la Base de Datos

Entradas:

Nombre de usuario y contraseña

Resultado esperado:

Se otorgan al usuario correspondiente los permisos que le corresponden a su rol.

Evaluación:

Prueba satisfactoria

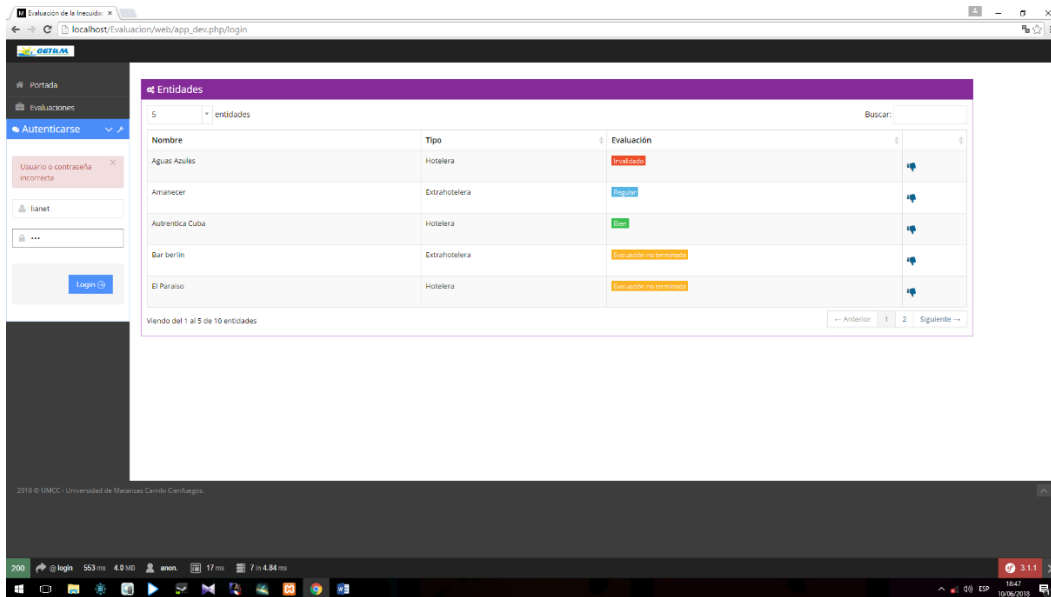


Ilustración 5 Se muestra en rojo un mensaje de error al autenticarse.

Tabla 31 PA 6

PRUEBAS DE ACEPTACIÓN	
Número Caso de Prueba: 6	No Historia: 6
Nombre Caso de Prueba: Test Gestionar Ayudas.	
Descripción: Verificar que se pueda insertar, modificar y eliminar las ayudas.	
Condiciones de ejecución: Estar conectado a la Base de Datos	
Entradas: Descripción de la ayuda.	
Resultado esperado: Se gestionen correctamente las ayudas.	
Evaluación: Prueba satisfactoria	

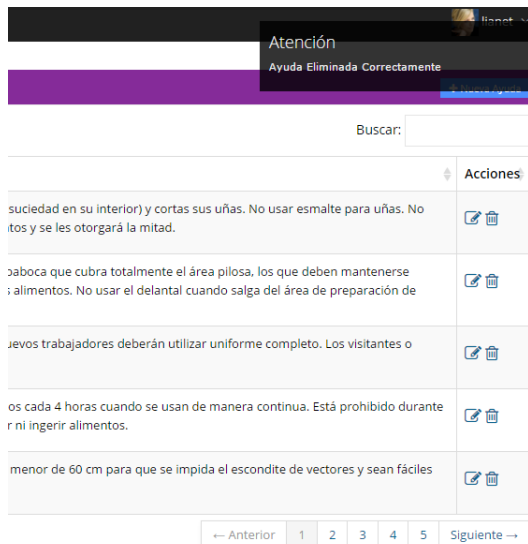


Ilustración 6 Se muestra el recuadro oscuro en la parte superior derecha anunciando la correcta eliminación de la ayuda.

Tabla 32 PA 8

PRUEBAS DE ACEPTACIÓN	
Número Caso de Prueba: 8	No Historia: 8
Nombre Caso de Prueba: Test Evaluar criterios en indicadores.	
Descripción: Verificar que se evalúen correctamente los criterios.	
Condiciones de ejecución: Estar conectado a la Base de Datos	
Entradas: Valor otorgado a los criterios.	
Resultado esperado: Se evalúen correctamente los criterios.	
Evaluación: Prueba satisfactoria	

3.2 Beneficios tangibles e intangibles. Estimación del costo.

Este software resultó de gran beneficio para el cálculo del índice global de cumplimiento de higiene y salud para los centros gastronómicos del ministerio del turismo, por lo expuesto al final del epígrafe anterior.

La fórmula especificada por Bohem plantea que:

$$\text{Costo Total (CT)} = \text{Tiempo de Desarrollo (TD)} * \text{Salario Mensual (SM)} * \text{Cantidad de Hombres (CH)}$$

TD = 24 semanas, lo cual se obtuvo sumando los puntos estimados de las historias de usuarios, agregando el tiempo dedicado a las pruebas. Es aproximadamente 6 meses.

SM = \$500, teniendo en cuenta el salario promedio de un desarrollador en Cuba.

$$\text{CH} = 1$$

$$\text{CT} = \$500 * 6 * 1$$

$$\text{CT} = \$3000$$

Realizando un pequeño análisis de los costos y los beneficios se puede afirmar en primer lugar que para esta entidad la aplicación informática resultó gratuita puesto que fue el resultado de un trabajo de diploma y su costo es inferior a las ventajas que trae.

3.3 Conclusiones parciales del capítulo.

En este capítulo se mostraron los elementos de prueba de la herramienta que se convierten en una herramienta de desarrollo, son un paso imprescindible para la verificación del buen funcionamiento de un software.

Como resultados finales se pudo obtener una aplicación web, con una apariencia agradable y fácil de usar. La planificación inicial se cumplió a un 100%, no existieron gastos para la entidad, se utilizaron las herramientas más actuales para su desarrollo, y el plan de entrega

fue cumplido con éxito. El cliente quedó complacido con el trabajo y se espera que se logren alcanzar mejores resultados en la empresa elevando así los niveles de cumplimiento de los prerrequisitos.

Conclusiones

Se ha desarrollado una aplicación web que evalúa los prerequisites higiénicos sanitarios que contribuye al análisis de la calidad de los servicios gastronómicos en el MINTUR. Siendo consecuente con las tareas planteadas se analizaron los fundamentos teóricos que caracterizan el estado del arte sobre el objeto de estudio.

Se analizaron las herramientas para el desarrollo de la aplicación web para la evaluación de la inocuidad y se seleccionó el framework Symfony y el gestor MySQL. Se utilizó de la metodología XP para el desarrollo de la aplicación web.

Se validó la propuesta implementada con las pruebas de la metodología XP y el cumplimiento de los requerimientos iniciales. Se le dio solución al problema científico planteado con el desarrollo de esta aplicación web.

Bibliografía

.ABTA. 2012. Alojamiento turístico. Manual técnico sobre Salud y Seguridad.

.Adalis.2015.[Prerrequisitos-adalil seguridad alimentaria](http://adalilseguridadalimentaria.com/empresa/appcc/prerrequisitos). Disponible en: adalilseguridadalimentaria.com/empresa/appcc/prerrequisitos. [Consultado: enero 2015]

. Álvarez, M. Á. (30 de 8 de 2007). *DesarrolloWeb*. Obtenido de <http://desarrolloweb.com>

.Álvarez, R. (2012). Introducción al HTML. Obtenido de <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.

. Bowler, T., & Bancner, W. (2009). *Symfony 1.3 web application development*.

.Cepero, S.; García, Y. 2014.Manual para la evaluación del Índice Global de cumplimiento de los prerrequisitos Higiénico Sanitarios en restaurantes. Departamento de Turismo. Facultad de Ciencias Económicas e Informática. Universidad de Matanzas "Camilo Cienfuegos".

.Eguiluz, Javier. (2017). Capítulo 1. Introducción. Libros del Web. [En línea] 2017. [Consultado el: 8 de Abril de 2017.] http://librosweb.es/libro/css/capitulo_1.html.

.FERRER, J. Programación Extrema y Software Libre, 2002.

.Fuentes, J. (2015). Desarrollo de Software Ágil. Extreme Programming y Scrum.

.GAMMA, E.; R. HELM, et al. Design Patterns. Elements of Reusable Object-Oriented Software. 2012.

- .Gutiérrez, Pastrana y Serra, 2010 tomado de Gutiérrez, N.; Castro, J.; Pastrana, E. Julio 2011. Evaluación de prerrequisitos en el sistema HACCP en empresas del sector agroalimentario. Revista EIA. ISSN 1794-1237 número 15, p. 33-43
- .Gutiérrez, N.; Pastrana, E.; Ramírez, E. 2010. Desarrollo de un instrumento para evaluar prerrequisitos en el sistema HACCP. Facultad de Ciencias Agropecuarias. Vol. 8. No. 1.
- .International Food Standard (IFS), 2014. Objectives and main directions. Disponible en: www.ifs.gov [consultado: enero 2018]
- . Jiménez, J. L. (2010). Reglas y Prácticas en eXtreme Programming. Universidad de Vigo, España.
- .Lerdorf, Rasmus. 2017. ¿Qué es PHP? Manual de PHP. [En línea] 2017. <http://php.net/manual/es/intro-what-is.php>.
- .Martínez, D. (2014). PostgreSQL vs MySQL. . Obtenido de <http://danielpecos.com/documents/postgresql-vs-mysql/>
- .Monguel, E.A. 2005. Metodología de la Investigación. México: Universidad Juárez Autónoma de Tabasco, 2005.
- .Pressman, R. 2010. Ingeniería de Software. Un enfoque práctico. 2010.
- .Risueño, P. (2013). Comenzando con Bootstrap, framework responsive.
- .RODRÍGUEZ, R., RAMÍREZ. Metodologías Ágiles para el desarrollo de Software, 2006. [Disponible en: <http://www.cyta.com.ar/ta0502/v5n2a1.htm> [consultado: marzo 2018]