

Universidad de Holguín
“Oscar Lucero Moya”
FACULTAD DE INFORMÁTICA- MATEMÁTICA

Trabajo de Diploma en opción al Título de Ingeniero Informático

*Sistema para la Gestión de Información de los trámites en
la Oficina de Monumentos y Sitios Históricos de Holguín*

Autora:

❖ Marianny Borjas Zaldívar

Tutores:

- ❖ M.Sc. Michel Velázquez Mariño
- ❖ Arq. Yohanka del Cerro Campano

Holguín, 2014

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo al Departamento de Arquitectos de la Oficina de Monumentos y Sitios Históricos de Holguín para que usen el sistema para facilitar el trabajo del área.

Para que así conste firmo la presente a los _____ días del mes de _____ del _____.

Nombre completo del autor:

Marianny Borjas Zaldivar

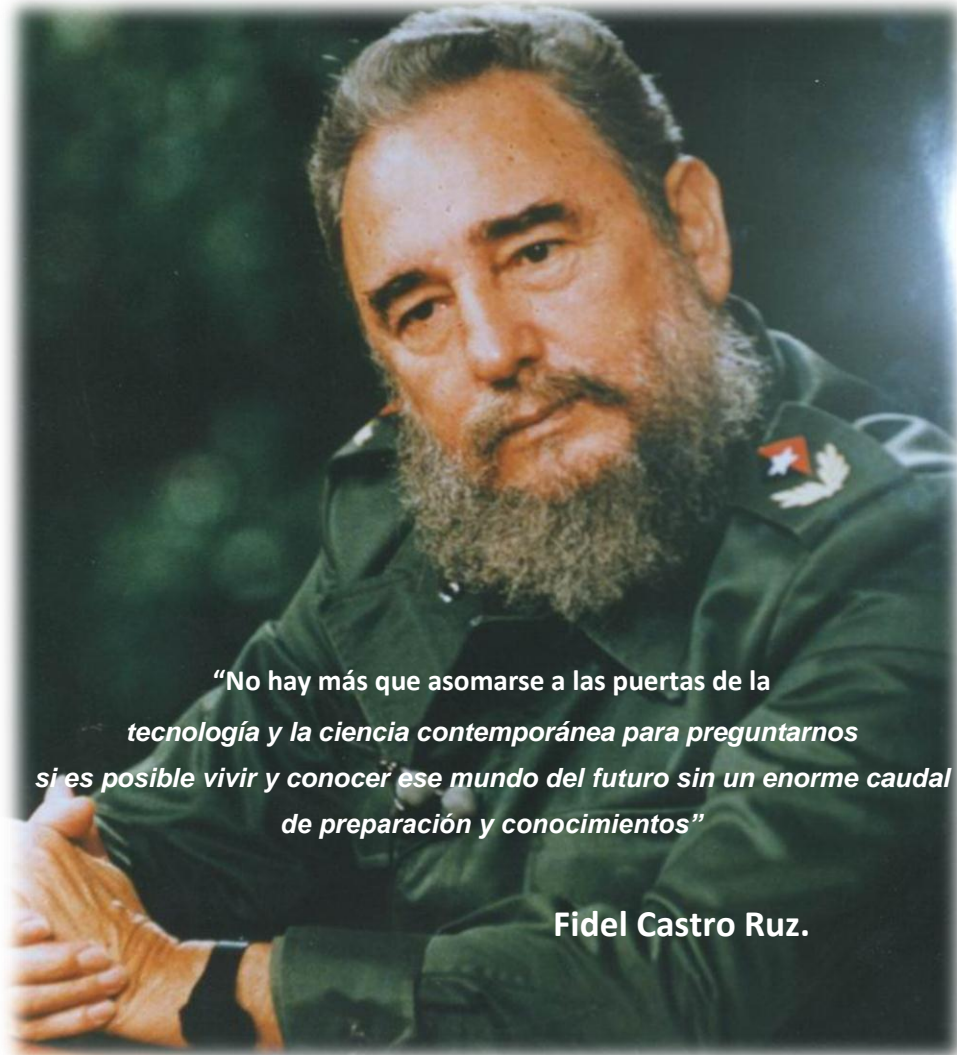
Nombre completo del tutor

M.Sc. Michel Velázquez Mariño

Nombre completo del tutor

Arq. Yohanka del Cerro Campano

Pensamiento



“No hay más que asomarse a las puertas de la tecnología y la ciencia contemporánea para preguntarnos si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación y conocimientos”

Fidel Castro Ruz.

Dedicatoria

A mi bebé Branci por existir.

A mis padres, a mi hermana, mi sobrino, mi esposo por darme siempre todo el apoyo que necesito.

Agradecimientos

A mi bebé Branci, que cuando me sentía desesperada me daba un beso y me decía mamá te quiero mucho.

A mis padres por su apoyo.

A mi hermana y mi sobrino.

A mi esposo.

A mi suegra.

A todos muchas gracias.

Opinión del usuario del trabajo de diploma

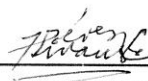
El trabajo de diploma titulado, Sistema para la Gestión de Información de los trámites en la Oficina de Monumentos y Sitios Históricos de Holguín. Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

- Totalmente
 Parcialmente en un _____ %

Los resultados de este trabajo de diploma le reportan a esta entidad los beneficios siguientes:

- Informatización del proceso de gestión de la información de los trámites que se procesan en la Oficina de Monumentos y Sitios Históricos de Holguín.
- Disposición de un sistema flexible y sólido basado en tecnología Web, que satisface los requerimientos del Departamento de Proyecto para la gestión de la información de los trámites.
- Facilitación del trabajo de los técnicos y especialistas, haciendo posible un mejor aprovechamiento de la jornada laboral.
- Reducción de riesgo de pérdida de información: la información se guarda de forma digital, con copias de respaldo.
- Disminución del trabajo manual.

Y para que así conste, se firma la presente a los 9 días del mes de junio del año 2014.



Lic. Hiram Pérez Concepción
Director OMSH





Plc. Jiliau Rodríguez Hdez.
Arq. Yohanka del Cerro Campano
Jefa Departamento Proyecto OMSH

Resumen

La Oficina de Monumentos y Sitios Históricos de Holguín tiene como labor principal proteger el patrimonio cultural y natural, servir como medio de disfrute espiritual, conocimiento, información y difusión de nuestra identidad y memoria histórica, así como desarrollar programas de rescate, conservación y restauración del patrimonio inmueble, tangible e intangible que permita a nuestra población y a los visitantes extranjeros comprender los cambios que se realizan en nuestras comunidades y distinguir los distintos elementos que forman nuestra cultura. Para la Oficina es muy necesario realizar un sistema para el trabajo con los trámites que se procesan en la entidad, ya que toda la información se procesa de forma manual y se guarda en un archivo, llevando su registro, de forma tal que para buscar determinado documento se hace más lenta la operación, se está hablando de los trámites: Regulaciones Urbanísticas, Revisión de Proyectos, Actualización de Título de Propiedad, Autorizo de Construcción, Dictamen y Visita Técnica, de todo el casco histórico que abarca desde la Loma de la Cruz (González Valdés) hasta Vidal Pita y José Antonio Cardet hasta Fomento, así como otros documentos afines a la institución.

Se realizó selección de las tecnologías a utilizar para el desarrollo del sistema: PHP como lenguaje de Programación, con el Framework CodeIgniter, MySQL como gestor de base de datos y el empleo de la metodología AUP para el modelado de la Ingeniería de Software de la aplicación y el gestor Bibliográfico End NoteX3. Una vez terminado el sistema se le realizaron casos de pruebas de aceptación y la valoración de sostenibilidad del mismo, llegando a la conclusión que el sistema en cuestión cumple con los requisitos del cliente y del usuario que una vez implantado en la entidad lo va a usar.

Abstract

The Office of Monuments and Historical Places of Holguín have as main work to protect the cultural and natural patrimony, to serve like half of spiritual enjoyment, knowledge, information and diffusion of our identity and historical memory, as well as to develop rescue programs, conservation and restoration of the patrimony property, tangible and intangible that allows our population and the foreign visitors to understand the changes that are carried out in our communities and to distinguish the different elements that form our culture. For the Office it is very necessary to carry out a system for the work with the steps that are processed in the entity, since all the information is processed in a manual way and he/she stays in a file, taking its registration, in such way that becomes slower to look for certain document the operation, one is speaking of the steps: Regulations Urban, Revision of Projects, Upgrade of Title of Property, I Authorize of Construction, Verdict and Visit Technique, of the whole historical helmet that embraces from Cruz's (González Valdés) Hill until Vidal it Whistles and José Antonio Cardet until Development, as well as other kindred documents to the institution.

She was carried out selection of the technologies to use for the development of the system: PHP like programming language, with Framework CodeIgniter, MySQL like database agent and the employment of the methodology AUP for the modeling of the Engineering of Software of the application and the Bibliographical agent End NoteX3. Once finished the system was carried out cases of tests of acceptance and the valuation of sostenibilidad of the same one, reaching the conclusion that the system in question fulfills the client's requirements and of the user that once implanted in the entity it will use it.

Índice de contenido

Introducción:	3
Capítulo 1: Fundamentación Teórica	8
1.1 Caracterización de la Oficina de Monumentos y Sitios Históricos de Holguín.....	8
1.2 Sistemas informáticos existentes vinculados al campo de acción	11
1.3 Tendencias y tecnologías actuales	11
1.3.1 Arquitectura Cliente-Servidor.....	13
1.3.2 Tecnologías de Servidores Web Apache Web Server	14
1.4 PHP como lenguaje de programación	15
1.5 Patrones de diseño	16
1.6 Marcos de trabajo	17
1.6.1 Características generales de CodeIgniter	18
1.7 Sistemas de gestión de bases de datos	20
1.7.1 MySQL como sistema gestor de base de datos	21
1.8 Metodología para el desarrollo de sistema.....	22
1.8.1 AUP	23
1.9 Visual Paradigm como herramienta CASE	25
1.10 Conclusiones del Capítulo.....	26
Capítulo 2: Descripción de la propuesta de solución.	27
2.1 Requerimientos del Sistema.....	27
2.1.1 Requerimientos funcionales:.....	28
2.2 Requerimientos No Funcionales.....	30
2.3 Actores del Sistema	32
2.3.1 Análisis de los requerimientos.....	32
2.3.2 Diagrama de casos de uso del sistema	32
2.3.3 Casos de uso del sistema	34
2.4 Modelo de Despliegue.....	36
2.5 Arquitectura del Sistema	37
2.6 Diagrama de Clases Persistentes	37
2.7 Modelo de Datos	38
2.7.1 Implementación.....	39
2.8 Pruebas realizadas al Sistema.....	42

2.8.1	Pruebas de aceptación (PA).....	42
2.9	Valoración de sostenibilidad de la propuesta	44
2.9.1	Impacto Socio-Humanista.....	45
2.9.2	Impacto Ambiental	45
2.9.3	Impacto Administrativo	46
2.9.4	Impacto Tecnológico	46
2.9.5	Conclusiones de la valoración de sostenibilidad.....	47
2.10	Conclusionesdel Capítulo	47
	Conclusiones.....	48
	Recomendaciones	49
	Glosario de Términos.....	50
	Bibliografía.....	51
	ANEXOS.....	I
	ANEXO I.....	I
	ANEXO II.....	I

Índice de Figuras

Fig: 1Ciclo de Vida de AUP	25
Fig: 2Diagrama de Procesos del Negocio	27
Fig: 3Diagrama de Paquetes CU	33
Fig: 4Diagrama Casos de Uso del Sistema	35
Fig: 5Modelo de Despliegue	36
Fig: 6Modelo MVC de CodeIgniter	37
Fig: 7Diagrama de Clases	38
Fig: 8Modelo de Datos.....	39
Fig: 9Interfaz de Autenticar Usuario.....	40
Fig: 10Interfaz Principal del Sistema	40
Fig: 11Menú Principal del Sistema	41
Fig: 12 Interfaz Dictamen Técnico	I
Fig: 13Interfaz Nomenclador Empresa.....	I
Fig: 14Interfaz de Reporte.....	II
Fig: 15Interfaz de Gráficos.....	II
Fig: 16Interfaz de Gráficos 1.....	III
Fig: 17Interfaz de Gráficos 2.....	III
Fig: 18Interfaz Configuración del Sistema.....	IV
Fig: 19Interfaz Créditos del Sistema	IV

Índice de tablas

Tabla 1 Actores del Sistema.....	32
Tabla 2 Descripción Gestionar Regulación	35
Tabla 3 Caso de Prueba Agregar Propietario ya existente	43
Tabla 4 Descripción Gestionar Dictamen Técnico	I
Tabla 5 Descripción Gestionar Revisión de Proyecto.	I
Tabla 6 Descripción Gestionar Visita Técnica.....	I
Tabla 7 Descripción Autorizo Construcción.....	II
Tabla 8 Descripción Gestionar Actualización Título Propiedad.....	II
Tabla 9 Descripción Gestionar Propietario.....	II
Tabla 10 Descripción Gestionar Empresas.	III
Tabla 11 Descripción Gestionar Tipo de Trámite.	III
Tabla 12 Descripción Gestionar Especialistas.....	III
Tabla 13 Descripción Gestionar Acciones Constructivas.....	III
Tabla 14 Descripción Gestionar Usuario.	IV
Tabla 15 Descripción Gestionar Log del Sistema.....	IV
Tabla 16 Descripción Gestionar Configuración del Sistema.....	IV
Tabla 17 Descripción Gestionar Cambiar Contraseña.....	V
Tabla 18 Descripción Gestionar Iniciar Sesión.	V
Tabla 19 Descripción Gestionar Cerrar Sesión.	V
Tabla 20 Descripción Gestionar Cantidad de Trámites.	V
Tabla 21 Descripción Gestionar Cantidad de Trámites por Especialista.	VI
Tabla 22 Descripción Gestionar Cantidad de Trámites por Estado de Ejecución.....	VI
Tabla 23 Descripción Gestionar Cantidad de Trámites por Tipo.....	VI
Tabla 24 Descripción Mostrar Cantidad de Trámites.....	VI
Tabla 25 Descripción Mostrar Cantidad de Trámites por Estado de Ejecución.	VII
Tabla 26 Descripción Mostrar Cantidad de Trámites por Tipo.....	VII
Tabla 27 Caso de Prueba Agregar Regulación ya existente.....	I
Tabla 28 Caso de Prueba Agregar Usuario ya existente.....	I

Introducción:

La Oficina de Monumentos y Sitios Históricos, constituida el 13 de noviembre de 2003, se subordina al Centro Provincial de Patrimonio Cultural, tiene como antecedentes el Equipo Técnico de Monumentos creado en 1984 y luego la Oficina de Historia, Conservación y Restauración de Holguín, por la Resolución No. 013/2000, del Presidente de la Asamblea Provincial del Poder Popular de Holguín.[1]

El trabajo de la Oficina de Monumentos está dirigido a proteger el patrimonio cultural y natural, servir como medio de disfrute espiritual, conocimiento, información y difusión de nuestra identidad y memoria histórica, así como desarrollar programas de rescate, conservación y restauración del patrimonio inmueble, tangible e intangible que permita a nuestra población y a los visitantes extranjeros comprender los cambios que se realizan en nuestras comunidades y distinguir los distintos elementos que forman nuestra cultura.

La realización de investigaciones, estudios y talleres para la protección del patrimonio tangible e intangible y el crecimiento del interés turístico por los Proyectos de Iniciativa de Desarrollo Local que realiza la institución, nos obligan a la búsqueda de variantes y alternativas dentro del campo de la informática y las nuevas tecnologías, que permitan obtener ingresos en ambas monedas, a partir de diferentes acciones comerciales derivadas de los servicios que hoy se prestan de forma gratuita, capaces de cubrir parte de los gastos del presupuesto asignado para el funcionamiento de la Oficina de Monumentos.

Después de conocer la función principal de la Oficina de Monumentos, se puede deducir, que por la gran cantidad de información que se procesa en la institución, se hace necesario realizar un sistema que permita digitalizar y guardar, todos los trámites que se desarrollan en la institución, para llevar un control eficiente de los mismos, que sirva de consulta para el cliente en determinado momento.

Para la Oficina es muy necesario realizar este sistema, ya que toda la información se procesa de forma manual y se guarda en un archivo, llevando su registro, de forma tal que para buscar determinado documento se hace más lenta la operación, se está hablando de los trámites: Regulaciones Urbanísticas, Revisión de Proyectos, Actualización de Título de

Propiedad, Autorizo de Construcción, Dictamen y Visita Técnica, de todo el casco histórico que abarca desde la Loma de la Cruz (González Valdés) hasta Vidal Pita y José Antonio Cardet hasta Fomento, así como otros documentos afines a la institución.

El proceso de gestión de la información en la Oficina de Monumentos y Sitios Históricos de Holguín, no cuenta con un sistema informático, capaz de manipular y gestionar la gran cantidad de informaciones que se procesan de forma diaria en la entidad, debido a esto la demanda a la hora de brindar atención al cliente es muy grande, y hasta lenta. Los trámites se archivan en carpetas una vez digitalizados, sin un orden establecido y a la hora de buscar uno en específico la operación es lenta, presentado demoras a la hora de dar una respuesta rápida a los clientes, que vienen a diario a la entidad. La otra forma de archivar los trámites es en papel que sufren deterioro y ataque de plagas, además hay que llevar un registro actualizado de los mismos por año en curso y en ocasiones se demora la búsqueda debido a que hay que buscar en todos los años, que se guardan en otras oficinas de la institución para una mayor seguridad de los mismos, siendo una tarea tediosa para el que lo desempeña, ya que si se contara con una herramienta informática que permita realizar estas búsquedas por años fuera más rápida la operación y la atención al cliente.

A partir de la situación existente en la Oficina debido a estas deficiencias, se pudo identificar el siguiente **problema científico**:

¿Cómo mejorar el proceso de Gestión de Información de los trámites en la Oficina de Monumentos y Sitios Históricos de Holguín?

De esta forma queda definido como **objeto de la investigación**: La Gestión de Información de los trámites en la Oficina de Monumentos y Sitios Históricos de Holguín.

Para resolver el problema se propone como **objetivo**: Desarrollar un sistema informático para la gestión de la información de los trámites en la Oficina de Monumentos y Sitios Históricos de Holguín.

Se determina como **campo de acción**: La informatización del proceso de gestión de la información de los trámites en la Oficina de Monumentos y Sitios Históricos de Holguín.

Para poder ejecutar esta investigación fue necesario dar respuesta a las siguientes **preguntas científicas**:

- ¿Cuáles son los fundamentos teóricos que sustentan la gestión de los trámites en la Oficina de Monumentos y Sitios Históricos de Holguín?
- ¿Cuál es el estado actual de la informatización de la Oficina de Monumentos y Sitios Históricos de Holguín?
- ¿Cómo diseñar un sistema que favorezca la gestión de trámites en la Oficina de Monumentos y Sitios Históricos de Holguín?
- ¿Será sostenible el sistema que se propone?
- ¿Cómo implementar un sistema que favorezca la gestión de trámites en la Oficina de Monumentos y Sitios Históricos de Holguín?
- ¿Satisface el sistema las necesidades que lo originaron?

La investigación se organizó a partir de las siguientes **tareas**:

1. Revisar los fundamentos teóricos que sustentan la gestión de trámites en la Oficina de Monumentos y Sitios Históricos de Holguín.
2. Diagnosticar estado actual de la información en la Oficina de Monumentos y Sitios Históricos de Holguín.
3. Analizar, capturar y diseñar los requerimientos del sistema que se propone.
4. Valorar la sostenibilidad del sistema que se propone.
5. Implementar los requerimientos del sistema informático que se propone.
6. Valorar la efectividad del sistema mediante pruebas de aceptación.

Para cumplimentar estas tareas se han empleado los siguientes **métodos de** investigación científica:

Métodos Teóricos

- ✓ **Análisis y síntesis:** Se utilizó en el estudio y diagnóstico de la gestión de la información. Permitiendo descomponer cada uno de los requerimientos del sistema y sus diferentes partes, para facilitar la comprensión del problema con más detalle y luego integrarlo para obtener una mejor solución.

- ✓ **Histórico-lógico:** Se puso en práctica durante el estudio de todas las herramientas utilizadas en la gestión de información y construcción de sistemas similares utilizados anteriormente con este fin, su evolución; así como de las leyes y principios esenciales en el funcionamiento de cada uno de ellos.
- ✓ **Modelación sistémica:** Este método será utilizado durante casi toda la etapa de elaboración del sistema, desde su análisis hasta su implantación, con la utilización de la metodología de ingeniería de sistema, de modo que se determinen los principales componentes del sistema y relaciones entre ellos.

Métodos Empíricos

- ✓ **Revisión de documentos:** Se utilizó para la recopilación de información en cuanto a los procesos que realiza el Equipo de Proyecto y para definir las tecnologías a usar para el diseño y desarrollo del sistema.
- ✓ **Entrevista:** Posibilitó obtener información sobre cómo se realiza el flujo y gestión de la información en la Oficina de Monumentos y Sitios Históricos de Holguín, ¿cómo la procesan y lo que necesitan? Se entrevistó a la Jefa del Equipo de Proyecto y Arquitectos y Técnicos del área proyecto, lo que facilitó la comunicación entre los usuarios y el desarrollador.
- ✓ **Observación:** Fue empleado para conocer cómo se desarrolla la gestión de la información para llevar a cabo el diagnóstico y la implantación del resultado obtenido de esta investigación.

Resultados esperados:

Con este trabajo se dispondrá de una herramienta informática que garantizará que la información relacionada con el proceso de trámites del Departamento de Proyecto logre más rapidez a la hora de buscar cada trámite, precisión y confiabilidad en los datos, existirá disminución de pérdidas de documentación y se eliminaría el trabajo manual, permitiendo una mayor seguridad de la información que se procesa en la entidad.

El informe consta de introducción, dos capítulos, conclusiones, recomendaciones, glosario de términos, bibliografía y anexos.

Capítulo 1: Fundamentación teórica.

Se hace un análisis de la Gestión de la Información de los trámites que se realizan en la Oficina de Monumentos y Sitios Históricos de Holguín. Se mostrará la descripción de las herramientas que se utilizarán para el cumplimiento a los objetivos de esta investigación. Además se abordará sobre la metodología utilizada para el desarrollo de la aplicación.

Capítulo 2: Descripción de la propuesta de solución.

Contiene los principales flujos de trabajo de la metodología utilizada para el desarrollo del sistema en la ingeniería de sistema como: la modelación del negocio y del sistema donde se exponen claramente los roles de los trabajadores del sistema y se describen los procesos. También se esboza la valoración de sostenibilidad y los resultados obtenidos en la implementación del sistema.

Capítulo 1: Fundamentación Teórica

En este capítulo se hace referencia a las principales características de la Oficina de Monumentos y Sitios Históricos de Holguín y sobre cómo se desarrolla el proceso de tramitación, que se realiza en la entidad. Se analizan el uso de las tecnologías en el campo de la informática y las comunicaciones, los diferentes lenguajes de programación para aplicaciones Web que se utilizan para darle solución al problema. Además del sistema gestor de base de datos que almacenará la información. Así como las metodologías y las herramientas utilizadas para el desarrollo de sistemas informáticos.

1.1 Caracterización de la Oficina de Monumentos y Sitios Históricos de Holguín

La Oficina de Monumentos y Sitios Históricos, se constituyó el 13 de noviembre de 2003, subordinada al Centro Provincial de Patrimonio Cultural, tiene como antecedentes el Equipo Técnico de Monumentos creado en 1984 y luego la Oficina de Historia, Conservación y Restauración de Holguín, por la Resolución No. 013/2000, del Presidente de la Asamblea Provincial del Poder Popular de Holguín.

El trabajo de la Oficina de Monumentos está dirigido a proteger el patrimonio cultural y natural, servir como medio de disfrute espiritual, conocimiento, información y difusión de nuestra identidad y memoria histórica, así como desarrollar programas de rescate, conservación y restauración del patrimonio inmueble, tangible e intangible que permita a nuestra población y a los visitantes extranjeros comprender los cambios que se realizan en nuestras comunidades y distinguir los distintos elementos que forman nuestra cultura.

La realización de investigaciones, estudios y talleres para la protección del patrimonio tangible e intangible y el crecimiento del interés turístico por los Proyectos de Iniciativa de Desarrollo Local que realiza la institución, nos obligan a la búsqueda de variantes y alternativas que permitan obtener ingresos en ambas monedas, a partir de diferentes acciones comerciales derivadas de los servicios que hoy se prestan de forma gratuita,

capaces de cubrir parte de los gastos del presupuesto asignado para el funcionamiento de la Oficina de Monumentos.

Su Capital Humano está conformado por 28 trabajadores de ellos 16 pertenecen al sexo femenino y 12 al sexo masculino. La misma cuenta con 5 departamentos: Administración, Investigación, Proyecto, Arqueología y Dirección.

Por la gran cantidad de información que se procesa en el Departamento de Proyecto se hace necesario realizar una aplicación que sea capaz de gestionar la información de los trámites que se realizan en la Oficina de Monumentos y Sitios Históricos de Holguín. El departamento realiza las siguientes actividades dentro de la institución:

- Atención a las solicitudes de Regulaciones para licencias de obras que realiza el Sistema de la Vivienda y el sistema de Planificación Física. Incluye la atención a la población y los organismos.
- Diseños o proyección arquitectónica o ingeniería para restauración, conservación, rehabilitación, mantenimiento, remodelación, división e inserción en Centros y Sitios Históricos.
- Diseño museográfico, para el montaje y remontaje de Salas de historia y museos.
- Estudio y proyecto de colaboración, así como proyectos socioculturales para edificaciones y centros históricos.
- Servicio de asistencia técnica, asesoría, consultoría, dictámenes y valoraciones técnicas, económicas de recursos materiales y control de autor.
- Dirección facultativa de obra.
- Investigaciones históricas y arqueológicas aplicadas a los trabajadores de conservación y restauración.
- Avaluos y certificados de obras terminadas.
- Cursos, seminarios, recorridos y visitas dirigidas, conferencias para la superación del personal técnico y profesional.[2]

La Oficina de Monumentos y Sitios Históricos de Holguín, demarca su trabajo diario en su Misión, Visión y Objeto Social, ya que estas premisas están presentes en su colectivo y repercuten en la satisfacción del cliente.

Misión, Visión y Objeto Social

✓ Misión

Es salvaguardar el patrimonio cultural y natural, ofreciendo alternativas de información sobre este y servir como medio de disfrute espiritual, conocimiento y difusión de nuestra identidad, así como el desarrollo de programas de rescate y conservación del patrimonio mueble e inmueble, tangible e intangible que permite a nuestra población y a los que nos visitan comprender los cambios que se han sucedido en nuestras comunidades y distinguir los distintos elementos que forman nuestra cultura por medio del conocimiento que ofrecen los diversos museos, zonas urbanas de valor histórico cultural, construcciones de diferentes clasificaciones tipológicas, sitios naturales, arqueológicos e históricos en post de formar una conciencia conservacionista en la tierra del encuentro de las Dos Culturas.[3]

✓ Visión

Contempla la existencia de áreas imprescindibles de consulta por la población, respaldada, atendidas y desarrollada por especialistas con un alto nivel científico que enriquecen con resultados el acervo patrimonial de la provincia.[3]

✓ Objeto Social

Rescatar, preservar, investigar, proteger y difundir, el patrimonio cultural de la provincia en defensa de los valores de nuestra identidad.

Gestión de la Información en el Departamento de Proyecto

El Departamento de Proyecto se subordina directamente a la Dirección. La plantilla de este departamento está formada por 9trabajadores entre Arquitectos (4), Ingenieros (3) y Técnicos (2), que se encargan de prestar los servicios siguientes:

- Elaboración de proyectos, asesoría, consultoría en materiales y técnicas constructivas, control de autor, servicios técnicos, investigaciones aplicadas y superación del personal.[2]

El proceso de gestión de la información en el Departamento de Proyecto no se realiza de forma eficiente por la falta de un sistema informático que posibilite el trabajo de archivar los trámites que se procesan en la institución, las compañeras que atienden esta actividad realizan su trabajo de registro de forma manual, teniendo que consultar registro por registro para obtener la información deseada, ocasionando a veces demora en la entrega de los trámites y haciendo tediosa y lenta la obtención de resultados. Con la aparición de las nuevas tendencias informáticas, se hace necesaria la creación de una herramienta que además de almacenar los datos de los trámites, permita el trabajo con estos, la modificación de los mismos y la obtención de ciertos y determinados reportes.

1.2 Sistemas informáticos existentes vinculados al campo de acción

En la investigación realizada en busca de la existencia de otros sistemas de gestión, no se conoce implantado ni en desarrollo un sistema vinculado a los fines de tramitación de Regulaciones, Proyectos, Actualizaciones, Autorizos, Dictamen y Visita, que se desarrollan en la entidad.

Para la Oficina de Monumentos se hace necesario implementar un sistema que permita realizar los trámites que se llevan a cabo en el Departamento de Proyectos, permitiendo almacenarlos en una base de datos, y que cumpla con las necesidades, y la entidad sea la única propietaria del Sistema.

1.3 Tendencias y tecnologías actuales

Después de analizado y realizado el estudio de la problemática de la presente investigación y los diferentes conceptos de los sistemas de información, se necesita conocer las tendencias, tecnologías y herramientas actuales disponibles para el desarrollo de la posible aplicación informática. Desde el comienzo de toda aplicación se debe tener en cuenta, las tecnologías y herramientas a utilizar para uncorrecto desarrollo. Utilizando aquellas tecnologías que de

acuerdo a sus características le aporten a la aplicación seguridad, confiabilidad, persistencia y portabilidad entre otros aspectos que hacen de un sistema un producto de calidad.

Para la selección de las tecnologías y herramientas a utilizar para la solución de la presente investigación, fue preciso realizar un estudio objetivo y selectivo de las tendencias y tecnologías disponibles en la actualidad para el desarrollo de aplicaciones informáticas, considerando que cumplan con las necesidades planteadas.

Características de una aplicación WEB

Como la aplicación a desarrollar será Web por la existencia de una pequeña red en la entidad, se realizó un estudio de las características de este tipo de aplicación.

Las aplicaciones web son basadas en navegadores. Son programas que se diseñan para funcionar a través de un navegador de internet, es decir, son aplicaciones que se ejecutan de manera directa.[4]

Estas aplicaciones son herramientas efectivas para lograr un objetivo interoperacional o de mucho flujo de información. Durante el estudio de diferentes tecnologías estas son las más populares debido a lo práctico del navegador web como cliente ligero, así como la facilidad para actualizar y mantener las aplicaciones sin distribuir e instalar sistema a miles de usuarios potenciales. Una aplicación web permite que siempre se esté accediendo a una información actualizada en cualquier parte de dicha red, además de ser un conjunto de páginas enlazadas que visualizan diferentes partes de la información que se quiere mostrar a través de ella, siendo una de las mejores herramientas para divulgar, gestionar y compartir la información por lo que trae consigo un aumento de la eficiencia en cuanto a la manipulación del volumen de la misma.

Estas aplicaciones poseen ventajas tales como:

- Disponibilidad: Suele ser alta porque sus usuarios pueden acceder (siempre que se haya concedido este permiso) desde cualquier otra máquina que no sea la suya.

- Actualizaciones inmediatas: Todos los usuarios al conectarse usan siempre la última versión que se haya mejorado hasta ese momento.
- Ahorra tiempo: Se pueden realizar tareas sencillas sin necesidad de instalar sistemas complementarios.
- Poca capacidad en disco duro: No ocupan espacio en el disco duro del cliente.
- Consumo de recursos bajo: Muchas de las tareas que realiza el sistema consumen recursos del cliente porque se realizan desde el propio servidor.
- Colaboración: Es sencillo acceder a los datos y compartirlos por parte de varios usuarios ya que el acceso al servicio se realiza desde una única ubicación.

Ya que el sistema resultante de esta investigación está insertado en un sistema en ambiente web se propone emplear este tipo de aplicación para la elaboración de la solución informática propuesta.[5]

Como ejemplos de aplicaciones web, tenemos las conocidas redes sociales como *Facebook* y *MySpace*, tiendas virtuales como *E-Bay* y *Amazon*, la enciclopedia en línea *Wikipedia*, el buscador *Google*, etc. Dado estos ejemplos, cabe mencionar que la principal característica de las aplicaciones web es que son ejecutadas sobre aplicaciones de escritorio que son conocidas como navegadores web, de los cuales los más conocidos son *Internet Explorer*, *Mozilla Firefox* y *Safari*. [6]

1.3.1 Arquitectura Cliente-Servidor

La arquitectura Cliente-Servidor posee muchas ventajas, la posibilidad de que todos los datos se almacenen en los servidores, permite mejor capacidad del control de la seguridad. El servidor puede controlar el acceso y el recurso al cerciorarse que dejó solamente esos accesos de usuarios permitidos y cambia datos, además cualquier elemento de la red Cliente-Servidor puede ser aumentado fácilmente.

El sistema propuesto corre bajo la arquitectura Cliente-Servidor por los beneficios que aporta, las facilidades para el uso de la red y tener la información almacenada en el servidor y acceder a ella haciendo peticiones a través de los clientes.[7]

1.3.2 Tecnologías de Servidores Web Apache Web Server

Es un sistema de código abierto. Esto significa que la programación que impulsa el sistema puede ser consultada y editada por cualquiera en el mundo. Este diseño abierto permite a cualquier programador crear una solución personalizada basada en el programa núcleo de *Apache*, o ampliar las funciones del sistema. Muchas de estas extensiones personalizadas se han escrito y están disponibles de forma gratuita. Si la función de medida perfecta no existe todavía, se puede crear fácilmente. Al ser de código abierto, *Apache* se actualiza constantemente. Muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web. El enfoque de código abierto significa que los usuarios no tienen que esperar a que versiones completas del sistema sean liberadas. En su lugar, pequeñas actualizaciones pueden ser instaladas tan pronto como estén disponibles.

Es muy portable. Esto significa que se puede instalar en una amplia variedad de servidores y sistemas operativos. Apache es capaz de ejecutarse en todas las versiones del sistema operativo *UNIX*. *Linux* es compatible, así como los sistemas operativos *Windows NT* y *MacOS*. En comparación, el propio servidor de *Microsoft Windows* normalmente sólo se puede instalar en sistemas operativos *Windows*. Desde un punto de vista de *hardware*, *Apache* puede ser utilizado en cualquier servidor con procesador de la serie *Intel 80x86* cuando se combina con *Windows*. Si *Apache* está siendo utilizado en un sistema operativo *Unix* o *Linux*, casi cualquier tipo de procesador es compatible. En general, *Apache* es uno de los sistemas de servidores más adaptables disponibles en la actualidad, y se ejecutará en una amplia gama de entornos técnicos.[8]

1.4 PHP como lenguaje de programación

Por las características de la aplicación en desarrollo se escogió como lenguaje de programación PHP, ya que permite la conexión con la mayoría de los sistemas gestores de base de datos y como se ejecuta en el servidor, no es necesario que su navegador lo soporte, es independiente del navegador.

PHP, acrónimo de *Hypertext Preprocessor* (inicialmente *PHP Tools*, o *Personal Home PageTools*), es un lenguaje de programación Web usado, principalmente, en la interpretación del lado del servidor (*server-side scripting*). La mayoría de su sintaxis es similar a C, Java y Perl.

Cuenta con los entornos *open source*. En su desarrollo hay un proceso de colaboración que permite obtener, de forma gratuita, una enorme cantidad de recursos: el propio lenguaje, el servidor para ejecutarlo, manuales y tutoriales además de scripts que pueden ser descargar y usar inmediatamente. Incorpora el paradigma de la Programación Orientada a Objetos (POO). Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad como *MySQL*, *PostgreSQL*, *mSQL*, *Oracle*, *dbm*, *filepro*, *Hyperwave*, *Informix*, *InterBase* y *Sybase*. [9]

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, similar al ASP de *Microsoft* o el JSP de *Sun*, embebido en páginas HTML y ejecutado en el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. No es un lenguaje de marcas como podría ser *HTML*, *XML* o *WML*. Está más cercano a *JavaScript* o a C, para aquellos que conocen estos lenguajes.

Características de PHP

- **Gratuito:** Al tratarse de software libre, puede descargarse y utilizarse en cualquier aplicación ya sea personal o profesional.

- **Gran popularidad:** Existe una gran comunidad de desarrolladores y programadores que continuamente implementan mejoras en su código.
- **Enorme eficiencia:** con escaso mantenimiento y un servidor gratuito, puede soportar millones de visitas en un día.
- **Sencilla integración con múltiples bases de datos:** Esencial para una página Web verdaderamente dinámica, es una correcta integración con base de datos. Aunque MySQL es la base de datos que mejor trabaja con PHP, puede conectarse también a PostgreSQL, Oracle, dbm, filePro, interbasem y muchas más bases de datos compatibles con *Open DatabaseConnectivity* (ODBC, por sus siglas en inglés).
- **Versatilidad:** PHP puede usarse con la mayoría de sistemas operativos, ya sea basados en UNIX, como con *Windows*, el sistema operativo de *Microsoft*.
- **Gran número de funciones predefinidas:** Está dotado de un gran número de funciones que nos simplificarán enormemente tareas habituales como descargar documentos, enviar correos, trabajar con *cookies* y sesiones, etc.[10]

1.5 Patrones de diseño

Modelo-Vista-Controlador.

MVC es un enfoque de software que separa la lógica de la aplicación de la presentación. En la práctica, le permite a sus páginas web contener mínimo código ya que la presentación está separada del código PHP.

- El Modelo representa sus estructuras de datos. Típicamente sus clases del modelo contendrán funciones que los ayudarán a devolver, insertar y actualizar información de su base de datos.
- La Vista es la información que se presenta al usuario. Una vista será normalmente una página web, pero en *CodeIgniter*, una vista también puede ser un fragmento de página como

el encabezado o pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".

- El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.[11]

1.6 Marcos de trabajo

En el desarrollo de sistema, los marcos de trabajo se han popularizado en los últimos años y se refieren a una estructura conceptual y tecnológica de soporte definida, en la cual un proyecto de sistema puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Además, provee de una estructura y metodología de trabajo que extienden o utilizan las aplicaciones del dominio. En otras palabras, es un esquema o patrón para el desarrollo o implementación de una aplicación completa, o de una parte específica de una aplicación.

Ventajas de utilizar un Frameworks

- ✓ Define una filosofía de trabajo.
- ✓ Proporciona librerías y funciones.
- ✓ Ahorra trabajo y tiempo.
- ✓ Permite producir aplicaciones más fáciles de mantener.
- ✓ Evita código duplicado.
- ✓ Permite crear aplicaciones multi-capa.
- ✓ Permite separar la aplicación en tres capas:
- ✓ La lógica de presentación que administra las interacciones entre el usuario y el sistema.
- ✓ La lógica de datos que permite el acceso a un agente de almacenamiento persistente u otros.
- ✓ La lógica de dominio o de negocio, que manipula los modelos de datos de acuerdo a los comandos recibidos desde la presentación.[12]

1.6.1 Características generales de CodeIgniter

Por las características que tiene el *CodeIgniter*, *framework* para el desarrollo de aplicaciones web, usando PHP, de fácil aprendizaje y menos rígido que otros *Frameworks*, provee un rico conjunto de bibliotecas, para tareas con interfaz sencilla, el cual permite desarrollar proyectos mucho más rápidos que el que se podría hacer si escribiera el código desde cero.

CodeIgniter es un entorno de desarrollo de aplicaciones para gente que construye sitios web usando PHP. El objetivo es habilitar el desarrollo de proyectos de forma mucho más rápida de lo que podría si escribiese código desde cero, a través de proveer un rico conjunto de librerías para tareas comúnmente necesarias, tanto como una simple interface y estructura lógica para acceder a estas librerías. *CodeIgniter* le permite concentrarse creativamente en su proyecto, minimizando el volumen de código necesario para una tarea determinada.[13]

CodeIgniter posee una característica de "Auto-carga" que proporciona la inicialización de librerías, asistentes (*helpers*), y complementos de forma automática durante el proceso de arranque del sistema. Si necesita ciertos recursos a nivel global a lo largo de su aplicación, debe considerar la posibilidad de Auto-carga para su conveniencia. Los siguientes elementos pueden ser cargados automáticamente:

- Clases *Core*: situadas en la carpeta "*libraries*"
- Asistentes (*Helpers*): situados en la carpeta "*helpers*"
- Complementos: situados en la carpeta "*plugins*"
- Archivos de configuración: situados en la carpeta "*config*"
- Archivos de lenguaje: situados en la carpeta "*system/language*"
- Modelos: situados en la carpeta "*models*"

Para autocargar recursos hay que agregar el elemento que deseamos cargar al *arrayautoload*, que se encuentra dentro del fichero de configuración *application/config/autoload.php*.

Algunos de los puntos más interesantes sobre este *framework*, sobre todo en comparación con otros productos similares, son los siguientes:

- **Versatilidad:** Quizás la característica principal de *CodeIgniter*, en comparación con otros *frameworks* PHP. *CodeIgniter* es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.
- **Compatibilidad:** *CodeIgniter*, al menos en el momento de escribir este artículo de desarrolloweb.com, es compatible con la versión PHP 4, lo que hace que se pueda utilizar en cualquier servidor, incluso en algunos antiguos. Por supuesto, funciona correctamente también en PHP 5.
- **Actualizado:** Desde la versión 2 de *CodeIgniter* ya solo es compatible con la versión 5 de PHP. Para los que todavía usen PHP 4 pueden descargar una versión antigua del *framework*, como *CodeIgniter V 1.7.3*, que todavía era compatible. Estas versiones están en la página de descargas de *CodeIgniter*.
- **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir *CodeIgniter* al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- **Flexibilidad:** *CodeIgniter* es bastante menos rígido que otros *frameworks*. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.

- **Ligereza:** El núcleo de *CodeIgniter* es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.
- **Documentación tutorializada:** La documentación de *CodeIgniter* es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto no facilita mucho la referencia rápida, cuando ya sabemos acerca del *framework* y queremos consultar sobre una función o un método en concreto, pero para iniciarnos sin duda se agradece mucho. [14]

CodeIgniter tiene un enfoque bastante poco estricto de MVC, ya que los Modelos no son obligatorios. Si no necesita la separación añadida o encuentra que mantener modelos requiere más complejidad de la que quiere, puede ignorarlos y construir su aplicación mínimamente usando Controladores y Vistas. *CodeIgniter* también le permite incorporar sus propios scripts, o inclusive desarrollar bibliotecas del núcleo para el sistema, permitiéndole trabajar en una forma que tenga más sentido para Ud.[15]

1.7 Sistemas de gestión de bases de datos

Para el trabajo con grandes volúmenes de información se hace necesario organizarlas en bases de datos, las cuales permiten escribir, leer, filtrar y almacenar datos de una manera ágil y segura según los permisos de cada usuario. Los sistemas gestores de bases de datos tienen como propósito simplificar y facilitar el acceso a los datos y hacer que los tiempos de respuesta a las solicitudes de los usuarios sean ágiles.

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayudando a realizar acciones como: definición de los datos, mantenimiento de la integridad dentro de la base de datos, control de la seguridad y privacidad, manipulación de los datos, realizar consultas complejas, importar y exportar desde otras fuentes, actualización constante e información disponible y coherencia e integridad de la información.

Ventajas de los SGBD, se encuentran:

- Facilidad de manejo de grandes volúmenes de información.
- Gran velocidad en muy poco tiempo.
- Independencia del tratamiento de información.
- Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones, consulta.
- No hay duplicidad de información, comprobación de información en el momento de introducir la misma.
- Integridad referencial al terminar los registros.[16]

Dentro de los sistemas gestores de base de datos más utilizados se encuentran Oracle, SQL Server, MySQL. El Frameworks CodeIgniter tiene soporte para el trabajo con los gestores de base de datos MySQL (4.1+), MySQLi, MS SQL, PostgreSQL, Oracle, SQLite, y ODBC.

Para el desarrollo de este sistema, se escogió el MySQL por cumplir con las características que se necesitan para desarrollar el sistema gestor de base de datos.

1.7.1 MySQL como sistema gestor de base de datos

Es un sistema de Gestión de Bases de Datos relacional, multihilo y multiusuario. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del sistema libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de Bases de Datos son:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- Usa tablas en disco b-tree para búsquedas rápidas con compresión de índice.
- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.
- Su bajo consumo lo hacen apto para ser ejecutado en una computadora con escasos recursos sin ningún problema.
- Soporte para control de transacciones en tablas transaccionales (tipo InnoDB), y soporte para procedimientos almacenados, subconsultas y disparadores (Triggers).
- Multiplataforma, incluyendo: GNU/Linux, Windows y BSD.[7]

MySQL es un sistema de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa Licencia Pública General (GPL, por sus siglas en inglés) para definir qué puede hacer y qué no puede hacer con el sistema en diferentes situaciones.

En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.[5]

1.8 Metodología para el desarrollo de sistema

Las Metodologías para el desarrollo de sistemas informáticos son un paquete de técnicas, funciones, procedimientos y herramientas para facilitar la realización de aplicaciones informáticas, las cuales ayudan al Ingenieros a organizar las ideas. El uso de las metodologías garantizan calidad, eficacia, facilidad de mantenimiento del sistema, además

de la satisfacción del cliente pues el equipo o el ingeniero ya cuenta con todos los requerimientos que necesitan para satisfacer el cliente. En muchos casos la selección de una metodología, puede depender del tiempo que se tiene para desarrollarlo.

El desarrollo de un buen sistema depende de un sinnúmero de actividades y etapas, para el éxito del producto es importante elegir la mejor metodología, su papel preponderante es sin duda esencial en un proyecto. El proceso de desarrollo de sistema se ha vuelto cada día más complejo, debido a la necesidad de adecuar la producción de sistema a los requerimientos de los usuarios y el crecimiento de los proyectos de desarrollo. El no hacer uso de una metodología en el proceso de construcción del sistema puede traer consigo una planificación irreal, mala calidad del producto, cambios no controlados y entrega de proyectos excesivamente tarde.

Entre las metodologías tradicionales se encuentran RUP y MSF; dentro de las principales metodologías ágiles, XP, Scrum, Cristal, Methods, AUP, ICONIX, entre otros.

El lenguaje en que se basan estas metodologías es el UML.

Por ser una metodología ágil se escogió para la realización del sistema informático el AUP(Proceso Unificado Ágil)por cumplir con las características principales y necesarias para el desarrollo de la aplicación, brinda un enfoque simple, rápido y organizado del proceso de desarrollo de software, adaptándose a las necesidades del proyecto y permitir, una vez terminado el flujo de trabajo Modelo una independencia marcada entre los desarrolladores y los clientes.[17]

1.8.1 AUP

Es un método ágil para el desarrollo de software basado en RUP, con las mismas prácticas y fases, pero con los flujos de trabajo simplificados y reducidos en número. Describe de forma sencilla un enfoque de desarrollo de software utilizando técnicas y conceptos ágiles. Se puede aplicar a una amplia gama de proyectos adaptándose a las necesidades de cada uno de ellos. Este enfoque aplica técnicas que incluyen Desarrollo Dirigido por Pruebas, Desarrollo Dirigido por Modelado Ágil, Administración Ágil de Cambios y Recodificación de Bases de Datos que permiten ganar en productividad. [18]

Otorga especial importancia a la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo. Para ello, se crean y mantienen listas identificando los riesgos desde etapas iniciales del proyecto. Especialmente relevante en este sentido es el desarrollo de prototipos ejecutables durante la base de elaboración del producto, donde se demuestra la validez de la arquitectura. [19]

En AUP el Modelo abarca los flujos de trabajo de Modelado de Negocio, de Requerimientos y de Análisis y Diseño, además asume las actividades de Gestión de Cambios típicas en la parte de manejo de requisitos. [18]

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren consecutivamente.

➤ Fase Inicio: Durante esta fase el equipo de desarrollo en conjunto con el cliente definen el alcance del nuevo sistema y proponen una o varias arquitecturas candidatas para el mismo.

➤ Fase Elaboración: En esta fase el equipo de desarrollo profundiza en la comprensión de los requisitos del sistema y en validar la arquitectura.

➤ Fase Construcción: Durante la fase de construcción el sistema es desarrollado y probado en el ambiente de desarrollo.

➤ Fase Transición: El sistema se lleva a los entornos de pre-producción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

[18-19]

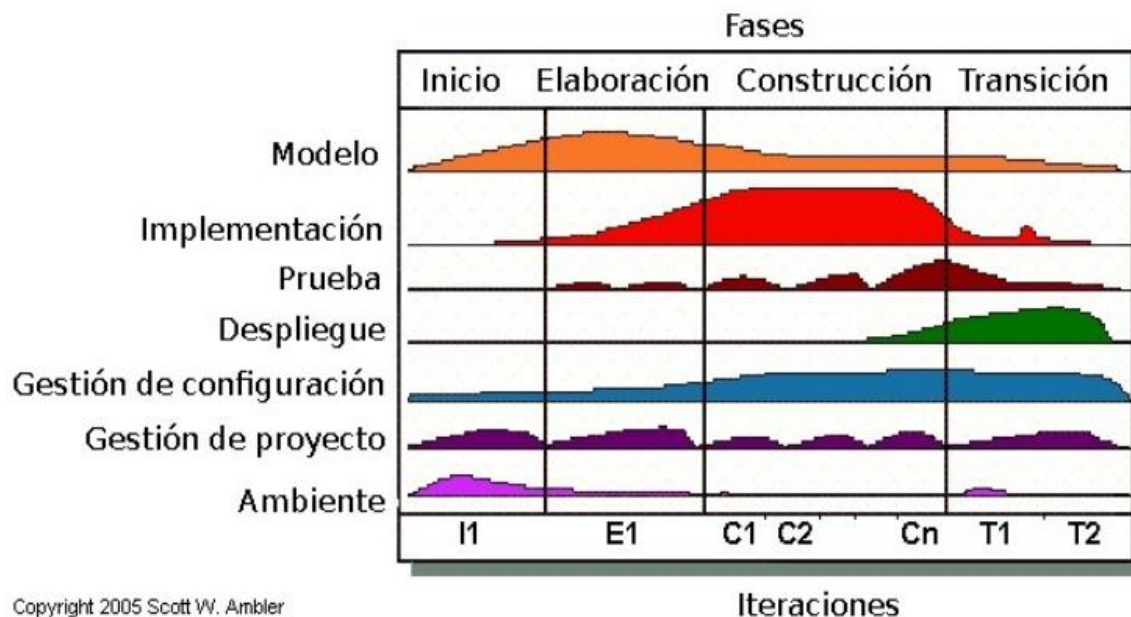


Fig: 1Ciclo de Vida de AUP

1.9 Visual Paradigm como herramienta CASE

Visual Paradigm (VP) facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.

Posibilita la corrección sintáctica, controla que el modelado con UML sea correcto.

Coherencia entre diagramas, al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades. Integración con otras aplicaciones, permite integrarse con otras aplicaciones, lo cual aumenta la productividad.

Trabajo multiusuario, permite el trabajo en grupo, proporcionando herramientas para compartir el trabajo.

Se escogió como herramienta CASE en VP porque facilita la reutilización, ya que se dispone de una herramienta centralizada donde se encuentran los modelos utilizados para otros proyectos. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software, además permite generar diversos informes a partir de la información introducida en la herramienta.[20]

Visual Paradigm for UML Enterprise Edition es una herramienta Case como tantas otras que nos permiten realizar a través de los diagramas, el modelado de de los requerimientos del sistema. En este caso se modelarán los diagramas de la metodología AUP.

1.10 Conclusiones del Capítulo

Se estudiaron los fundamentos teóricos relacionados a la gestión de trámites y de manera general las tecnologías para el desarrollo de aplicaciones web. Dadas las características que debe cumplir el sistema y las bondades que ofrecen las tecnologías antes descritas se escogieron: PHP como lenguaje de programación con las librerías (Highcharts: Librería de Gráficos (escritas en HTML 5 y Java Script), Dompdf: Librería de Reportes en PDF (escrita en PHP, que convierte una página HTML a PDF), GroceryCRUD: Librería de CRUD (para crear, mostrar, editar y eliminar registros, que incluye además buscar y paginar el estado de registros)), con el Frameworks CodeIgniter, como sistema gestor de base de datos (SGBD) MySQL, Apache Web Server como servidor web, Visual Paradigm como herramienta CASE, para el modelado de diagramas, y el uso de la metodología de desarrollo AUP en la elaboración del sistema propuesto.

Capítulo 2: Descripción de la propuesta de solución.

En este capítulo se lleva a cabo el análisis, modelación y diseño de la propuesta de solución que será como resultado el Sistema para la Gestión de Información de los trámites en la Oficina de Monumentos y Sitios Históricos de Holguín. Siguiendo las etapas que componen la metodología AUP permite la construcción del Diagrama de procesos del negocio a partir de la captura de los Requerimientos Funcionales (RF), se especifican los Requerimientos No Funcionales (RNF) para modelar las propiedades que el sistema requiere, se modelan los casos de uso, como guía para describir los pasos hacia la implementación del sistema.

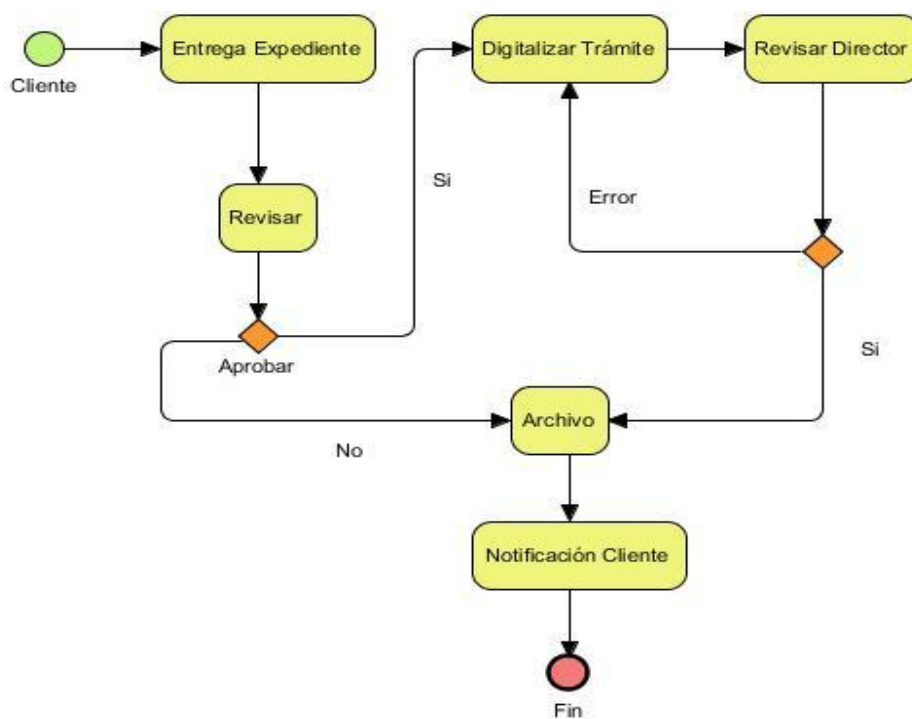


Fig: 2 Diagrama de Procesos del Negocio

2.1 Requerimientos del Sistema

El proceso de captura de requerimientos se lleva a cabo para averiguar aquellas funcionalidades o características que el sistema a desarrollar debe poseer, una buena identificación de estos facilita comprender lo que desea el cliente, brindándole una solución

que satisfaga sus necesidades. Los requerimientos se clasifican en funcionales y no funcionales. [7]

2.1.1 Requerimientos funcionales:

Los requerimientos funcionales hacen una especificación de las acciones que el sistema será capaz de realizar. A continuación se presenta una lista desglosada de estos requerimientos:

- El sistema debe gestionar los datos de los trámites (regulaciones, proyectos, actualizaciones, autorizos, dictámenes y visitas), con toda su información.

Gestionar Trámites

RF1: Gestionar Regulaciones (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF2: Gestionar Dictamen Técnico (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF3: Gestionar Revisión Proyecto (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF4: Gestionar Visita Técnica (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF5: Gestionar Autorizo Construcción (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF6: Gestionar Actualización Título Propiedad (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

- El sistema debe permitir usar nomencladores para facilitar el trabajo con los datos de los trámites (Regulación, Proyecto, Actualización, Autorizo, Dictamen o Visita).

Gestionar Nomencladores

RF7: Gestionar Propietario (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF8: Gestionar Empresas (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF9: Gestionar Tipo de Trámites (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF10: Gestionar Especialistas (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

RF11: Gestionar Acciones Constructivas (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

- El sistema debe gestionar usuario.

Gestionar Administración

RF12: Gestionar Usuario (agregar, editar, eliminar, buscar, exportar).

RF13: Gestionar log del Sistema (mostrar, buscar).

RF14: Gestionar configuración del Sistema (agregar, mostrar, editar, buscar).

RF15: Cambiar contraseña.

RF16: Iniciar sesión.

RF17: Cerrar sesión.

- El sistema debe permitir realizar reportes (regulaciones, proyectos, actualizaciones, autorizos, dictámenes y visitas).

Generar Reportes - Gráficos

RF18: Generar Reporte Cantidad de Trámites (imprimir, exportar).

RF19: Generar Reporte Cantidad de Trámites por Especialista (imprimir, exportar).

RF20: Generar Reporte Cantidad de Trámites por estado de ejecución (imprimir, exportar).

RF21: Generar Reporte Cantidad de Trámites por tipo (imprimir, exportar).

- El sistema debe ser capaz de realizar salvadas de la información procesada por los usuarios (de forma semestral) y actualizar los datos.
- El sistema ha de permitir mostrar de forma gráfica los reportes de los trámites.

RF22: Mostrar Cantidad de trámites

RF23: Mostrar Cantidad de Trámites por estado de ejecución

RF24: Mostrar Cantidad de Trámites por tipo

2.2 Requerimientos No Funcionales

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc. Los requerimientos no funcionales del sistema han ayudado a determinar las propiedades que debe tener el sistema y se describen a continuación.

Los requerimientos no funcionales para este sistema son los siguientes:

Apariencia o Interfaz externa

RNF1: El sistema debe ser fácil de manipular, pues los usuarios finales no son expertos en computación, a pesar de usar la computadora para las labores rutinarias.

RNF2: El flujo de trabajo en el sistema debe asemejarse al proceso rutinario de gestión de trámites que se llevan a cabo en la OMSH.

RNF3: El diseño debe ser agradable y atractivo a los usuarios para lograr una mejor concentración, sin desviar demasiado su atención del contenido de trabajo.

RNF4: Los colores deben encontrarse en la gama del azul, blanco y negro, pero con tonalidades suaves y relajantes para evitar mucho esfuerzo visual, por la gran constancia del uso y dependencia del sistema en el trabajo de los usuarios.

RNF5: La interfaz no debe recargarse con imágenes para proporcionar una navegación cómoda.

RNF6: El idioma utilizado debe ser el español.

Usabilidad

RNF7: El sistema debe ser accesible desde cualquier PC de la OMSH.

RNF8: El sistema debe estar funcionando durante las 8 horas que dura la jornada laboral.

RNF9: El diseño del sistema debe ser sencillo para agilizar su tiempo de conexión.

Rendimiento

RNF10: El tiempo de respuesta en la búsqueda de la información almacenada en el archivo digital debe ser corto (5 - 10 segundos aproximadamente), por lo que el acceso a la base de datos se debe efectuar de forma rápida.

Portabilidad

RNF11: El sistema puede ser usado desde cualquier computadora del área de informática y bajo la plataforma Windows.

RNF12: La computadora Servidor puede funcionar bajo la plataforma Windows.

Seguridad

RNF13: Sólo los usuarios autorizados podrán acceder al sistema.

RNF14: Garantizar que las funcionalidades del sistema se realicen de acuerdo con el nivel del usuario que esté activo.

RNF15: No existe información que se pueda obtener sin ser usuario del sistema.

RNF16: Sólo el administrador del sistema tendrá acceso a la BD, a los ficheros fuentes del sistema y es responsable de la autorización en general del mismo.

RNF17: El sistema debe tener protección contra acciones no autorizadas evitando la corrupción y estados inconscientes que puedan afectar el flujo de trámites de la OMSH o integridad de la información almacenada.

RNF18: Sólo el administrador del sistema podrá hacer salvadas de seguridad de manera periódica, según se estipule en la propia entidad.

Software

RNF19: El Sistema Operativo (SO) de la computadora cliente debe ser Windows 2000 o superior.

RNF20: La computadora servidor debe tener Windows 2000 o superior, Servidor Web Apache, MySQL y funcionalidades para PHP.

Hardware

RNF21: Para ejecutar el sistema como requerimiento mínimo es necesario una computadora con procesador Pentium 3 o superior tanto en la computadora cliente como en el servidor.

RNF22: La computadora servidor y las computadoras clientes deben estar conectadas a la red.

2.3 Actores del Sistema

Nombre del Actor	Justificación
Administrador	Será la persona encargada de realizar cambios en el sistema, y acceder a todas las funcionalidades del mismo.
Técnico	Será la persona que trabajará de forma directa con el sistema, y podrá acceder a las funcionalidades que el administrador determine, que no afecten el buen funcionamiento de la aplicación.
Especialista	Serán aquellas personas que podrán consultar la aplicación sin realizar cambios que afecten el funcionamiento de la aplicación.

Tabla 1 Actores del Sistema

2.3.1 Análisis de los requerimientos

Este flujo de trabajo tiene entre sus objetivos identificar los objetos del dominio, elementos del mundo real. Así como identificar las relaciones de generalización o agregación que existen entre estos objetos, para iniciar con la elaboración de un diagrama de clases de alto nivel. Además como parte de este flujo de trabajo se deben identificar los casos de uso, para esbozar el diagrama de casos de uso. Si es necesario organizarlos en paquetes.

2.3.2 Diagrama de casos de uso del sistema

El diseño de los casos de uso del sistema partió de los requisitos funcionales extraídos del negocio. Los actores identificados fueron: Técnico, Especialista y Administrador que es el responsable de operar todas las funcionalidades del sistema.

Para lograr una mayor modularidad, seguridad y organización en el desarrollo del sistema, los diagramas de CUS se distribuyen en cuatro paquetes o agrupaciones lógicas. Los paquetes son: Trámites, Nomencladores, Reportes-Gráficos y Administración. En estos paquetes se agrupan los requerimientos de acorde a su funcionalidad. En la Figura 2 se muestra el diagrama de paquetes de los casos de uso del sistema.

El paquete Trámites incluye la gestión de los trámites en la entidad. El paquete Nomencladores se centra en la gestión de los nomencladores del sistema. El paquete Reportes-Gráficos permite obtener estadísticas sobre los trámites. El paquete Administración contiene la gestión de los usuarios, el control de acceso, así como el registro de eventos del sistema.

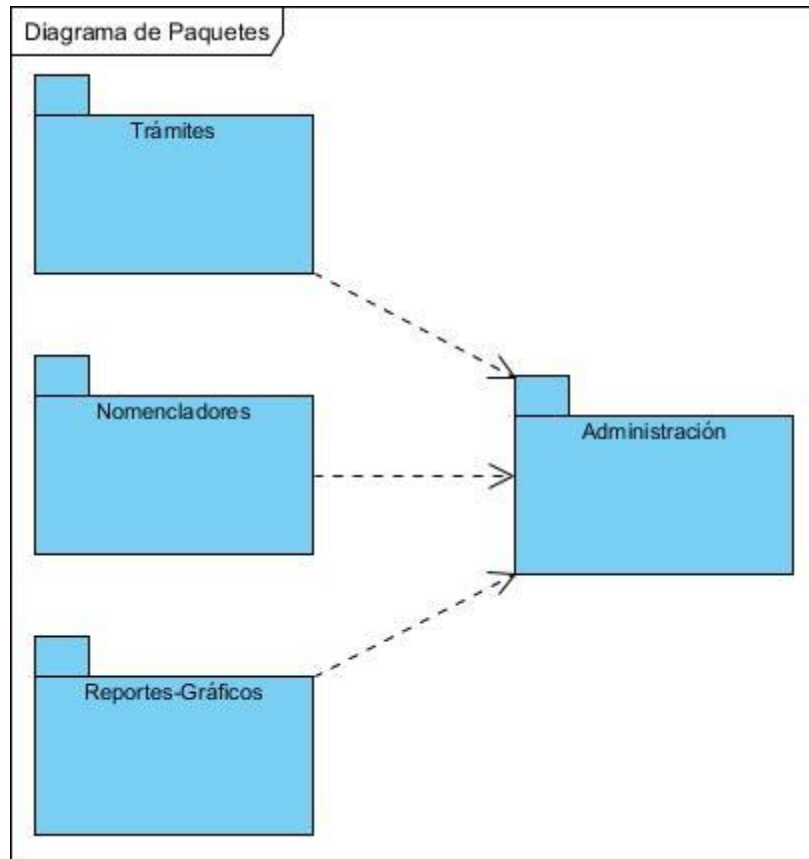


Fig: 3Diagrama de Paquetes CU

2.3.3 Casos de uso del sistema

El diseño de los casos de uso del sistema partieron de los requerimientos funcionales relacionados por el cliente. Se identificaron los actores del sistema (Administrador, Técnico y Especialista), el Administrador con privilegios de operar todas las funcionalidades del sistema y los demás con privilegios específicos.

Casos de uso del sistema:

CU1- Configuración del Sistema: El Administrador realiza configuraciones del sistema: revisa Log del Sistema, Usuarios, Roles y Cambia Contraseña.

CU2- Gestionar Trámites: El Administrador y Técnico procesan cada trámite: Regulaciones Urbanísticas, Revisión de Proyecto, Actualización de Título de Propiedad, Autorizo de Construcción, Dictamen Técnico y Visita Técnica, realizando registros de los mismos, búsquedas, impresión, exportar, etc.

CU3-Gestionar Nomencladores: El Administrador y Técnico procesan la información de los nomencladores según el trámite a realizar.

CU4- Generar Reportes – Gráficos: El Administrador, Técnico y Especialista generan Reportes y Gráficos según especifique el usuario.

CU5- Autenticar Usuario: El Administrador, Técnico y Especialista autentican usuario para acceder al sistema.

CU6- Cerrar Sesión: El Administrador, Técnico y Especialista después de realizar las acciones pertinentes en el sistema cierran sesión.

A continuación se muestra en la Fig: 5, el diagrama de casos de uso del sistema.

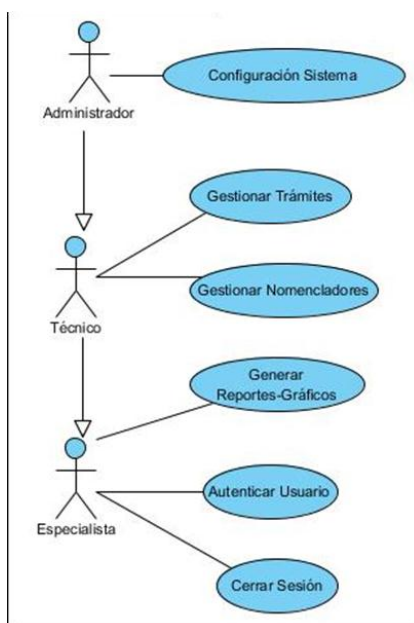


Fig: 4Diagrama Casos de Uso del Sistema

Descripción de casos de uso

Con el fin de detallar las relaciones entre los casos de usos y los actores, se realizaron las descripciones de los casos de uso. Los casos de uso deben estar descritos y enfocados de forma tal que obedezca al esquema de capturar la acción del usuario seguida y asociada a la reacción del sistema.

Dentro del caso de uso Gestionar Trámite se encuentra Gestionar Regulación que a continuación se hace su descripción en la Tabla No. 2, las demás descripciones se encuentran en los anexos.

Caso de Uso	Gestionar Regulación
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con la Regulación (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-1

Tabla 2Descripción Gestionar Regulación

2.4 Modelo de Despliegue

El modelo de despliegue representa los objetos físicos que hacen posible el despliegue de las funcionalidades del sistema y la gestión de estas en los diferentes nodos de cómputo que conforman la aplicación.

La computadora cliente representa al grupo de computadoras, a través de las cuales los usuarios pueden gestionar la información en la aplicación *Web* que se encuentra en el servidor *Web*. Una computadora cliente cuenta con un navegador *Web* que realiza una conexión con el servidor *Web* Apache. La comunicación entre el cliente y el servidor *Web* se realiza a través de la familia de protocolos de red TCP/IP. El servidor de bases de datos soporta la base de datos MySQL y responde las consultas del servidor *Web*, mediante el protocolo de red TCP/IP. [7]

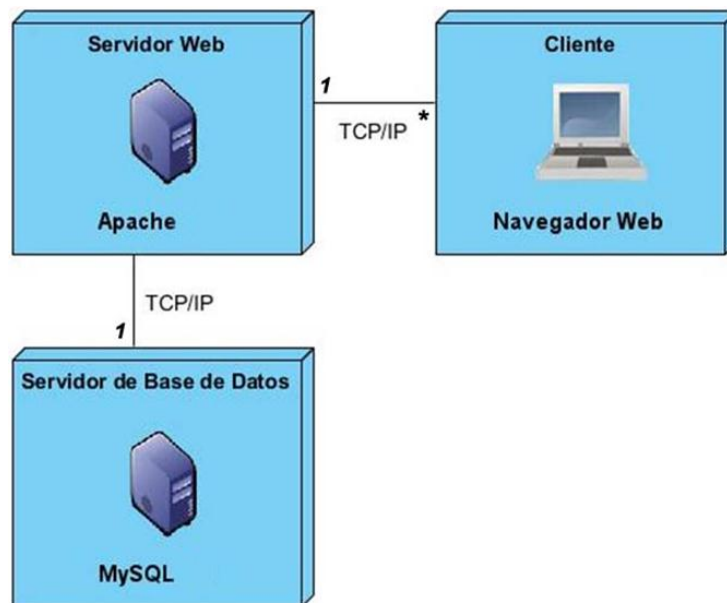


Fig: 5 Modelo de Despliegue

2.5 Arquitectura del Sistema

La arquitectura técnica es conocida como la arquitectura del sistema y de software, con el propósito de describir el sistema que se está intentando construir en términos de estructura, satisfacer los requerimientos del negocio y el nivel de servicio del sistema que se va a desarrollar, incluyendo además la topología del sistema, es decir los nodos del servidor, localización física en la red, la elección del servidor de aplicación.[21]

El estilo arquitectónico seleccionado fue el estilo de llamada-respuesta del Modelo-Vista-Controlador (MVC).

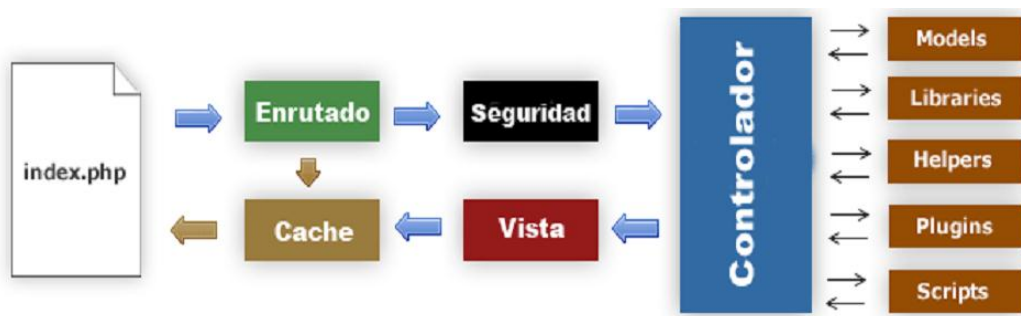


Fig: 6Modelo MVC de CodeIgniter

2.6 Diagrama de Clases Persistentes

El diagrama de clases persistentes modela la información que trasciende en el tiempo, incluso después de cerrado el sistema.

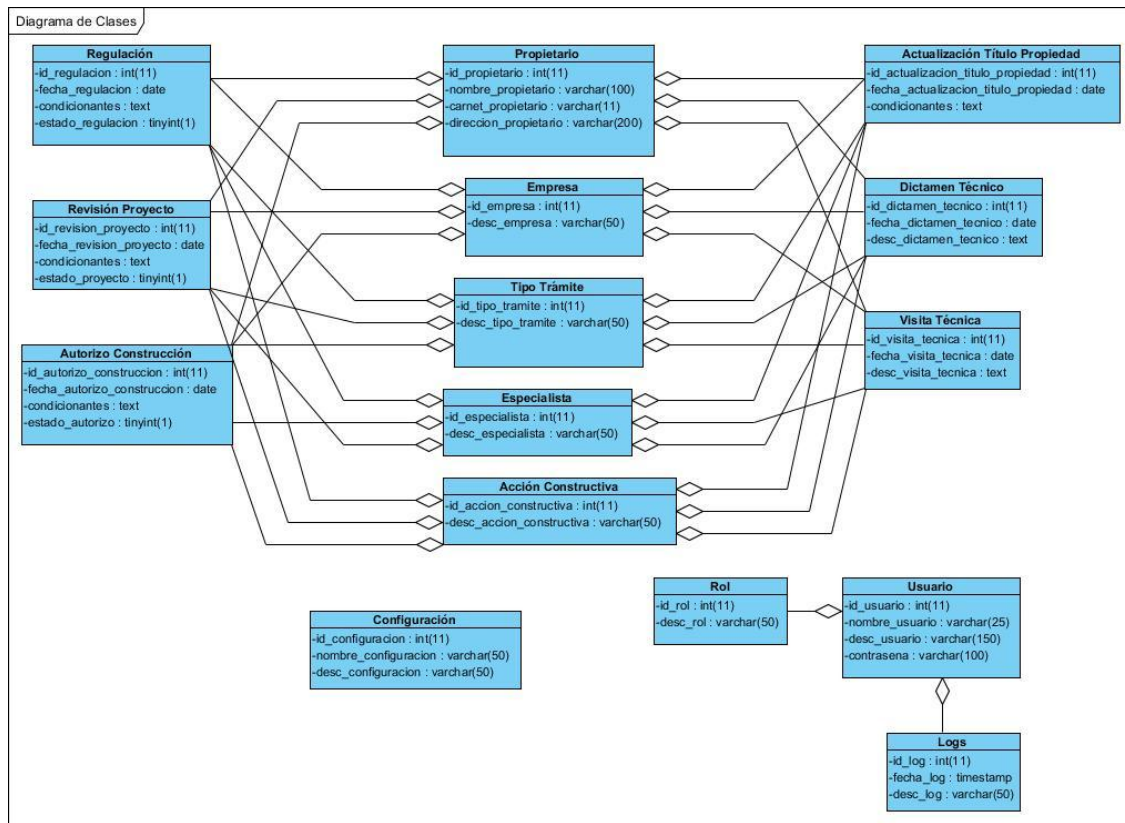


Fig: 7Diagrama de Clases

2.7 Modelo de Datos

El modelo de datos describe de manera abstracta la representación de los datos persistentes que se utilizan en el sistema. Para determinar el modelo lógico de datos se hace necesario primero analizar cuáles son las clases persistentes del componente a desarrollar. Las clases persistentes son aquellas capaces de mantener su valor en el espacio y el tiempo.

Además, describe la estructura lógica de la información que queda almacenada en la base de datos.[9]

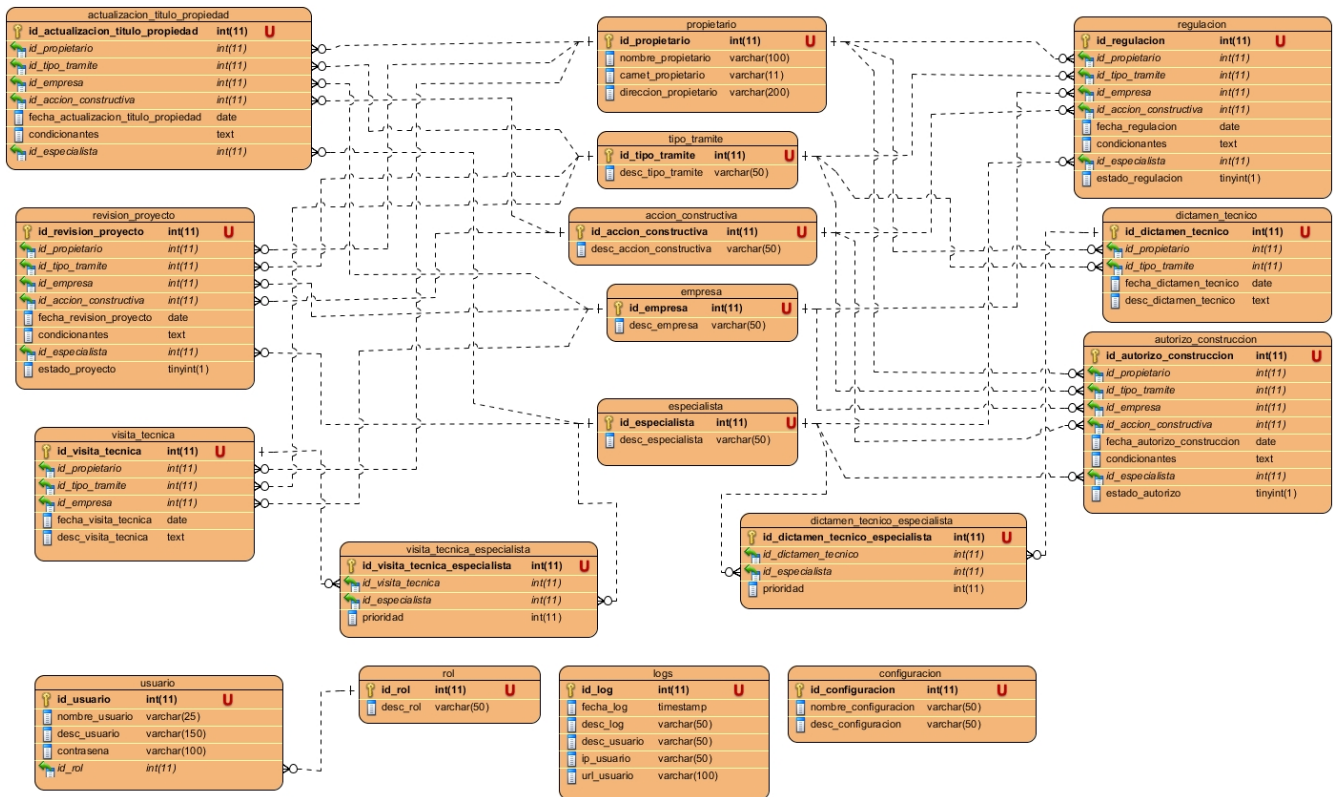


Fig: 8 Modelo de Datos

2.7.1 Implementación

Para el diseño de esta aplicación Web se emplearon algunos principios y técnicas de diseño visual en las páginas que la conforman. El diseño visual define la apariencia del sistema y es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace. Los colores de la aplicación se encuentran en la gama de los azules claros, los cuales son suaves y refrescantes, a la vez que brindan una sensación de paz y tranquilidad al usuario. A continuación se mostrarán y explicarán algunas pantallas de la aplicación, donde se puede apreciar el diseño visual de la misma.

El sistema muestra una interfaz donde el usuario introduce su nombre de usuario y su contraseña (Fig: 10) para autenticarse y da un clic en el botón “Iniciar Sesión”. Todos los usuarios tienen un rol de usuario único y una contraseña.



Fig: 9 Interfaz de Autenticar Usuario

Luego de autenticarse correctamente como usuario este es direccionado a la ventana correspondiente como se muestra en la Fig: 11, en dependencia del tipo de usuario que intenta acceder a nuestro sistema, le permite el acceso a las operaciones que brinda la aplicación.

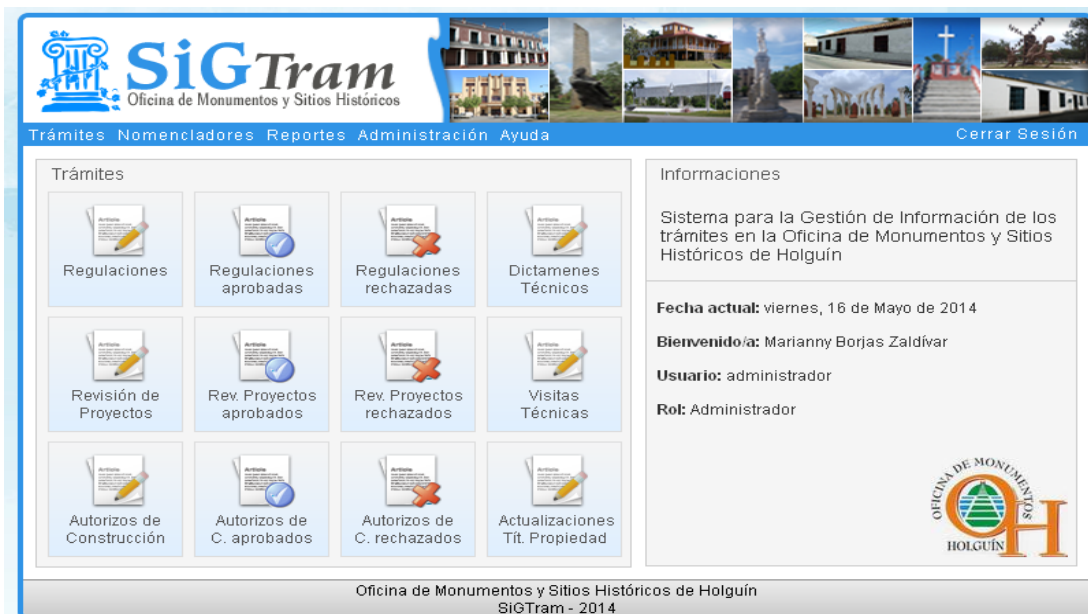


Fig: 10 Interfaz Principal del Sistema

El diseño del Sistema en general se basa en la utilización del menú principal para acceder de forma íntegra a las diferentes opciones que brinda el mismo. Este menú principal aparece en la parte superior izquierda de todas las páginas dentro del sistema, se encuentra integrado por las opciones que serán explicadas a continuación:



Fig: 11 Menú Principal del Sistema

Trámites:

Los trámites correspondientes a realizar se muestran al entrar al sistema que con sólo dar clic encima de alguno de ellos se pueden llenar los datos del mismo, con diferentes opciones a realizar, editar, mostrar, exportar a Pdf, imprimir, eliminar, etc.

Nomencladores:

En este menú se pueden agregar los nomencladores del sistema, para agilizar el proceso de llenado de los trámites una vez comenzado el llenado de los mismos, una vez insertados se pueden realizar modificaciones en los mismos, mostrar, editar o eliminar según se desee y el rol de usuario te lo permita. Ver Fig: 13.

Reportes:

En este menú se especifican en forma de texto y gráfico la cantidad de trámites hasta la fecha ingresados en el sistema. Ver Fig: 14, 15, 16 y 17.

Administración:

En este menú se realiza la configuración del sistema, así como el registro de usuarios y la cantidad de veces que se accede a la aplicación.

Ayuda:

En este menú se pueden ver los créditos del sistema, así como su autor. No presentando en sí una ayuda al usuario debido a que este sistema tiene un Manual de Usuario adicional para consultar las posibles dudas que pueda tener el usuario en cuestión.

2.8 Pruebas realizadas al Sistema

La metodología AUP propone un desarrollo dirigido por pruebas, que permiten evacuar de forma rápida y con el menor número de modificaciones posibles cualquier error detectado. Como una técnica para garantizar que los requerimientos hayan sido cumplidos y que la aplicación es realmente lo que el cliente necesita, además de asegurar su correcto funcionamiento son realizadas las pruebas de aceptación. Estas pruebas funcionan como una caja negra, pues cada una de ellas representa una salida esperada del sistema, donde es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

El objetivo final de las mismas es lograr que los requerimientos sean cumplidos y que el sistema sea aceptable. Una vez que todas las historias de usuario hayan pasado sus pruebas de aceptación se considera entonces terminada la aplicación.

Existen varias premisas que pueden servir acertadamente como objetivos de las pruebas:

- La prueba es 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 un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.[22]

2.8.1 Pruebas de aceptación (PA).

Las pruebas de aceptación son las pruebas formales realizadas por (o en nombre de) el usuario, para determinar si el sistema cumple con los requerimientos especificados. Una

prueba de aceptación (PA) tiene como propósito, demostrar al usuario el cumplimiento de un requerimiento del *software*. [23]

Características de una PA:

- Describe un escenario (secuencia de pasos) de ejecución o uso del sistema desde la perspectiva del usuario.
- Puede estar asociada a un requerimiento funcional o uno no funcional.
- Un requerimiento tiene una o más PA asociadas.

Las PA permiten validar el producto, en cambio las pruebas unitarias y de integración solo permiten verificar el producto. Con las PA sólo se evalúan las entradas que se realizan sobre el *software* y las salidas que produce, sin preocuparse por su comportamiento interno, por lo que se dice que son pruebas de caja negra. La validación del *software* se desarrolla mediante una serie de pruebas de caja negra que demuestran la conformidad con los requerimientos. [24]

Caso de uso	Agregar Propietario RF 7
Caso de prueba	Agregar Propietario ya existente
Entrada	El administrador hace clic en el Menú Nomencladores para Agregar Propietario en la pestaña Propietarios. El sistema muestra el formulario “Agregar Propietario” para la entrada de los datos del Propietario. El administrador o técnico especifica datos del mismo y hace clic en el botón Guardar. El sistema comprueba que no existan campos vacíos.
Resultado	El sistema no agrega el propietario, muestra un mensaje donde especifica que el campo carnet es único y el especificado ya existe en el sistema, así que no se puede volver a crear. El sistema muestra opciones para volver a la lista de propietarios del sistema.
Condiciones	El administrador y técnico deben haber iniciado sesión.

Tabla 3 Caso de Prueba Agregar Propietario ya existente

- El caso de prueba Agregar Propietario ya existente, arrojó un resultado positivo ya que no se puede agregar más de un propietario con el mismo nombre y carnet de identidad ya que este es un identificativo de cada propietario.

Ver Anexo II para otros casos de prueba:

- El caso de prueba Agregar Regulaciones ya existente, arrojó un resultado positivo ya que no se puede agregar más de una regulación con la misma dirección, nombre del propietario y carnet de identidad. Ver Fig: 27
- El caso de prueba Agregar Usuario ya existente, arrojó un resultado positivo ya que no se puede agregar más de un Usuario con la misma configuración ya que el campo usuario tiene valor único. Ver Fig: 28

2.9 Valoración de sostenibilidad de la propuesta

¿Qué es valoración de sostenibilidad de un producto informático?

Se entiende por valoración de sostenibilidad de un producto informático: El proceso de evaluación preventiva de riesgos de impactos de tipo ambiental, socio humanistas, administrativos y tecnológicos de un producto informático, previsibles desde el diseño del proyecto, que favorece su autorregulación, para la satisfacción de la necesidad que resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente. [25]

- Dimensión Socio-Humanista
- Dimensión Ambiental
- Dimensión Administrativa
- Dimensión Tecnológica

Valoración de Sostenibilidad

A continuación se hace una valoración de sostenibilidad del sistema, teniendo en cuenta su impacto socio-humanista, ambiental, tecnológico y administrativo, en el cual se realiza un estudio de factibilidad del producto informático mediante la estimación de esfuerzo y costo.

2.9.1 Impacto Socio-Humanista

Con relación al impacto socio-humanista, se puede asegurar que la implementación del producto mejorará el trabajo con los trámites que se procesan en la entidad, beneficiando al personal que va a trabajar con la aplicación, ya que podrán tener acceso a toda la información de forma segura e íntegra. Mejorando en gran medida las condiciones de trabajo y a la hora de prestar un determinado servicio a los clientes, se realiza de forma más rápida y eficiente, tendiendo este sistema que le permite realizar búsquedas de forma rápida de determinado trámite (los trámites que se llevan a cabo en la institución, pertenecen a todo el casco histórico que abarca desde la Loma de la Cruz (González Valdés) hasta Vidal Pita y José Antonio Cardet hasta Fomento).

Por otro lado, la implementación del sistema no genera ni disminuye el empleo en la organización, lo que hace es favorecer el trabajo del personal existente que trabaja con los trámites.

A partir de lo analizado, se arriba a la conclusión de que el software es sostenible desde la dimensión socio-humanista.

2.9.2 Impacto Ambiental

Este sistema presenta una interfaz agradable a la vista, con colores claros que no molestan para realizar el trabajo de tramitación, interrelacionando así el trabajo con el entorno del lugar donde se utiliza el mismo. Se tuvieron en cuenta las exigencias fisiológicas del ser humano, es decir, tamaño de letra, espaciamiento entre caracteres, tipografía, etc.

Como el sistema brindará seguridad y rapidez a la hora del procesamiento de la información, el tiempo del usuario frente al ordenador se reduce y se minimizan los riesgos de deformación en la columna por mala postura durante horas y el desgaste de la visión. De todas formas se recomienda una postura correcta de usuario en su puesto de trabajo.

Por todo lo antes expuesto se concluye que el software es sostenible en la dimensión ambiental.

2.9.3 Impacto Administrativo

Al desarrollar un sistema para determinada entidad, se debe analizar el impacto que tendrá el mismo, en cuanto a gastos, calidad, servicios, administración de recursos, ahorro, etc. Para el caso de este sistema no provocaría gastos sino ayudaría a realizar la tramitación, de forma más eficiente, rápida y segura, si en un futuro cercano se ampliara la red y las PC del centro, se capacitarían aquellas personas a trabajar directamente con el sistema, de lo contrario no haría falta invertir, debido a que la tecnología utilizada para la creación del sistema es flexible a las características que presentan el hardware y software, no se incrementaría el consumo eléctrico ya que sería con la misma cantidad de máquinas.

En cuanto al ahorro de material de oficina, se continuaría imprimiendo la copia del trámite para ser entregado al cliente y el resto se guardaría de forma digital. Anteriormente se sacaban 2 copias y la de la entidad se almacenaba en un archivo, con posibilidad de extravío, ataque de plagas, etc.

Finalmente se puede decir que este sistema no proporcionará ingresos directos a la entidad, pero mejorará considerablemente el trabajo de los técnicos y especialistas de la OMSH, con los trámites que se procesan de forma diaria en el centro, evitando que se generen pérdidas o deterioro de las copias de archivo.

Por tanto en la dimensión administrativa la realización del sistema es considerada sostenible.

2.9.4 Impacto Tecnológico

Para el empleo del producto no se necesitan grandes conocimientos informáticos, pues cuenta con un Manual de Usuario y una Ayuda que facilitan el trabajo con el mismo. Desde el punto de vista de la infraestructura electrónica, la Oficina de Monumentos y Sitios Históricos (OMSH) cuenta con el equipamiento necesario para la implantación y empleo de este sistema informático. En su fabricación se usaron herramientas que se encontraban al alcance (PHP como lenguaje de programación (Frameworks CodeIgniter), MySQL como gestor de base de datos y el uso de la Metodología AUP con su herramienta Case Visual Paradigm). Este producto está diseñado para admitir cambios tecnológicos, como mejoras en la red y hardware, así como cambios de plataforma.

Por lo antes expuesto, se arriba a la conclusión de que el sistema es sostenible desde el impacto Tecnológico.

2.9.5 Conclusiones de la valoración de sostenibilidad.

Después de realizar las pruebas al sistema se obtuvo como resultado un Producto Informático terminado, que cumple con los requerimientos del cliente y satisface las necesidades del mismo que dieron origen al problema planteado en la investigación.

El producto cumple con las necesidades del cliente, es de fácil uso y acceso, agradable a la vista, resuelve el problema planteado y facilita el trabajo con los trámites que son procesados en la entidad, siendo de gran utilidad para el departamento de proyectistas que trabajarán directamente con él.

2.10 Conclusiones del Capítulo

Una vez llevado a cabo las etapas que componen la metodología AUP permitiendo la construcción de los Diagramas: Procesos del negocio a partir de la captura de los Requerimientos Funcionales (RF), se especificaron los Requerimientos No Funcionales (RNF), se modelaron los diagramas: casos de uso, modelo de despliegue, arquitectura, clases y modelo de datos. Una vez implementado se realizaron casos de pruebas de aceptación para probar las funcionalidades del sistema, dando paso a la valoración de sostenibilidad desde sus cuatro dimensiones.

Conclusiones

Con el desarrollo del sistema propuesto se cumplió con el objetivo planteado en la presente investigación, obteniéndose un producto informático, que permite gestionar la información de los trámites que son procesados de forma diaria en la entidad. Además, la solución garantiza la integridad, disponibilidad y seguridad de la información procesada, con un adecuado rendimiento y usabilidad.

El estudio realizado y los resultados obtenidos permiten concluir que:

- El estudio y diagnóstico realizado en el departamento de proyecto de la OMSH Holguín, permitió detectar las deficiencias existentes en la gestión de la información básica de los trámites, lo que constituyó el punto de partida de la presente investigación.
- La utilización de una adecuada selección de tecnologías, facilitó el desarrollo de una aplicación *Web* para la gestión de la información básica de los trámites procesados en la OMSH Holguín, potenciando la funcionalidad de la misma.
- La metodología de desarrollo de software utilizada para estructurar, planificar y controlar el proceso de desarrollo de la solución, resultó fundamental para maximizar los índices de calidad en los procesos de ingeniería de software implicados.
- La realización de Pruebas al Sistema permitió llegar a la conclusión que el mismo está lo más óptimo posible, siendo de gran ayuda para el trabajo en la entidad y la seguridad y confiabilidad de los datos que se procesan de forma diaria en la entidad.
- El producto obtenido se evaluó preventivamente como sostenible desde las cuatro dimensiones, según los criterios propuestos por el procedimiento aplicado para la valoración de sostenibilidad.
- La aplicación *Web* favoreció la gestión de la información básica de los técnicos de OMSH Holguín, dando cumplimiento al objetivo trazado en la investigación.

Recomendaciones

1. Continuar la generalización de la aplicación, incorporándose dentro de las demás oficinas del país que realizan estos tipos de trámites, como una herramienta que favorezca el trabajo con los mismos y permita la seguridad de los datos procesados.
2. Agregar al sistema nuevas funcionalidades que enriquezcan el mismo y ayuden aún más a desarrollar el trabajo del departamento de Proyecto y la OMSH en general.

Glosario de Términos

OMSH: Oficina de Monumentos y Sitios Históricos

PHP: Hypertext Preprocessor

MVC: Modelo Vista Controlador. Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón *MVC* se ve frecuentemente en aplicaciones web, donde la vista es la página *HTML* y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y el controlador representa la Lógica de negocio.

SQL: *Structured Query Language* o Lenguaje Estructurado de Consultas

PDF: *Portable Document Format* (formato de documento portable).

Framework: Es una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software* para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

AP: Pruebas de Aceptación

CUS: Casos de Uso del Sistema.

VP: Visual Paradigm, herramienta Case para el modelado de diagramas.

PA: Pruebas de Aceptación

PD: Pruebas Directas

AUP: Proceso Unificado Ágil

Bibliografía

- [1] E. OMSH, *Proyecto Cultural*, 2013.
- [2] E. OMSH, *Descripción de la actividad*, 2012.
- [3] E. OMSH, *Curriculum OMSH*, 2012.
- [4] R. Mir, "Trabajo de diploma: Sistema de Gestión de la Información de los Servicios de Ventas a Turistas en Infotur," 2010.
- [5] Y. Anzardo, "Módulo para la Gestión de Trámites de la Dirección de Relaciones Internacionales de la Universidad de Holguín "Oscar Lucero Moya", " 2013.
- [6] I. Rivas, "Estudiantealdeaunare3irmadj Aplicaciones web," ed, 2011.
- [7] J. Martínez, "APLICACIÓN WEB PARA LA GESTIÓN DE LA INFORMACIÓN BÁSICA DE LOS TRABAJADORES DE EDUCACIÓN EN HOLGUÍN," 2013.
- [8] "Apache, F. About the Apache HTTP Server Project," 2013.
- [9] U. Hernández, "Sistema informático para la gestión de la información de la actividad de cuadros en la Facultad de Enfermería "Arides Estévez Sánchez", " 2013.
- [10] Y. Armide, "SISTEMA DE GESTIÓN DE INFORMACIÓN DE LA PRODUCCIÓN EN LA UEB DE CULTIVOS PROTEGIDOS DE LA EMPRESA AGROPECUARIA LA JÍQUIMA," 2013.
- [11] V. Téllez, "Codeigniter," ed, 2011.
- [12] J. J. Gutiérrez, "¿Qué es un framework web? ," 2006.
- [13] V. T. Lozano, "CodeIgniter. ," 2011.
- [14] (2013, CodeIgniter.maff. Available: <http://www.desarrolloweb.com/articulos/codeigniter.html>
- [15] F. Velo, *CodeIgniter_Guia_Usuario_2.1.0*, 2011.
- [16] J. F. E. Sibley, "Evolution of data-base management systems. A. C. Surveys," 1976.
- [17] M. Velázquez, "NeuroScreening. Sistema Informático para la cuantificación de variables neurológicas en pacientes con Ataxia Espinocerebelosa tipo 2.," 2011.
- [18] A. SW. (2006, The Agile Unified Process (AUP). Available: <http://www.ambysoft.com/unifiedprocess/agileUP.htm>
- [19] F. E, *Metodologías Ágiles. Proceso Unificado Ágil (AUP)*, 2009.
- [20] V. P. I. A. r. reserved, "Ayuda Visual Paradigm," 1999-2010.
- [21] A. S. Díaz, "Trabajo de diploma: Sistema de Gestión de Información de apoyo a la toma de decisiones en la Dirección de Relaciones Internacionales de la Universidad de Holguín "Oscar Lucero Moya", " 2011.
- [22] G. J. Myers, "The Art of Software Testing," 2004.
- [23] S. Rosenberg, "Rosenberg, D. and M. Stephens, «Use Case Driven Object Modeling with UML: Theory and Practice». 2007.," 2007.
- [24] R. Pressman, *Ingeniería de Software. Un enfoque práctico 5ta Edición* 2006.
- [25] R. Concepción, *Procedimiento para valorar la sostenibilidad de un producto informático*, 2006.

ANEXOS

Gestionar Trámites

Caso de Uso	Gestionar Dictamen Técnico
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con EL Dictamen Técnico (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-2

Tabla 4 Descripción Gestionar Dictamen Técnico

Caso de Uso	Gestionar Revisión Proyecto
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con la Revisión de Proyecto (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-3

Tabla 5 Descripción Gestionar Revisión de Proyecto.

Caso de Uso	Gestionar Visita Técnica
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con la Visita Técnica (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-4

Tabla 6 Descripción Gestionar Visita Técnica.

Caso de Uso	Gestionar Autorizo Construcción
Actores	Administrador y Técnico

Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con el Autorizo de Construcción (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-5

Tabla 7 Descripción Autorizo Construcción

Caso de Uso	Gestionar Actualización Título Propiedad
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con la Actualización de Título de Propiedad (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-6

Tabla 8 Descripción Gestionar Actualización Título Propiedad.

Gestionar Nomencladores

Caso de Uso	Gestionar Propietario
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con el Propietario (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-7

Tabla 9 Descripción Gestionar Propietario.

Caso de Uso	Gestionar Empresas
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con la Empresa (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).

Referencia	RF-8
------------	------

Tabla 10 Descripción Gestionar Empresas.

Caso de Uso	Gestionar Tipo de Trámite
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con el Tipo de Trámite (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-9

Tabla 11 Descripción Gestionar Tipo de Trámite.

Caso de Uso	Gestionar Especialistas
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con los Especialistas (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-10

Tabla 12 Descripción Gestionar Especialistas.

Caso de Uso	Gestionar Acciones Constructivas
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes con las Acciones Constructivas (agregar, mostrar, editar, eliminar, buscar, imprimir, exportar).
Referencia	RF-11

Tabla 13 Descripción Gestionar Acciones Constructivas.

Configuración del Sistema

Caso de Uso	Gestionar Usuario
Actores	Administrador
Importancia	Alto
Descripción	El administrador realiza las opciones pertinentes con el usuario(agregar, editar, eliminar, buscar, exportar).
Referencia	RF-12

Tabla 14 Descripción Gestionar Usuario.

Caso de Uso	Gestionar Log del Sistema
Actores	Administrador
Importancia	Alto
Descripción	El administrador realiza las opciones pertinentes con loa Log del sistema(mostrar, buscar).
Referencia	RF-13

Tabla 15 Descripción Gestionar Log del Sistema.

Caso de Uso	Gestionar Configuración del Sistema
Actores	Administrador
Importancia	Alto
Descripción	El administrador realiza las opciones pertinentes para la Configuración del Sistema(agregar, mostrar, editar, buscar).
Referencia	RF-14

Tabla 16 Descripción Gestionar Configuración del Sistema.

Caso de Uso	Cambiar Contraseña
Actores	Administrador, Técnico y Especialista
Importancia	Alto
Descripción	El administrador, técnico o especialista realizan las opciones pertinentes para Cambiar Contraseña.

Referencia	RF-15
------------	-------

Tabla 17 Descripción Gestionar Cambiar Contraseña.

Caso de Uso	Iniciar Sesión
Actores	Administrador, Técnico y Especialista
Importancia	Alto
Descripción	El administrador, técnico o especialista realizan las opciones pertinentes para Iniciar Sesión.
Referencia	RF-16

Tabla 18 Descripción Gestionar Iniciar Sesión.

Caso de Uso	Cerrar Sesión
Actores	Administrador, Técnico y Especialista
Importancia	Alto
Descripción	El administrador, técnico y especialista realizan las opciones pertinentes para cerrar sesión.
Referencia	RF-17

Tabla 19 Descripción Gestionar Cerrar Sesión.

Generar Reportes – Gráficos

Caso de Uso	Gestionar Cantidad de Trámites
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes para consultar el reporte Cantidad de Trámites (imprimir, exportar).
Referencia	RF-18

Tabla 20 Descripción Gestionar Cantidad de Trámites.

Caso de Uso	Gestionar Cantidad de Trámites por Especialista
Actores	Administrador y Técnico
Importancia	Alto

Descripción	El administrador o técnico realizan las opciones pertinentes para consultar el reporte Cantidad de Trámites por Especialista (imprimir, exportar).
Referencia	RF-19

Tabla 21 Descripción Gestionar Cantidad de Trámites por Especialista.

Caso de Uso	Gestionar Cantidad de Trámites por Estado de Ejecución
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes para consultar el reporte Cantidad de Trámites por Estado de Ejecución (imprimir, exportar).
Referencia	RF-20

Tabla 22 Descripción Gestionar Cantidad de Trámites por Estado de Ejecución.

Caso de Uso	Gestionar Cantidad de Trámites por Tipo
Actores	Administrador y Técnico
Importancia	Alto
Descripción	El administrador o técnico realizan las opciones pertinentes para consultar el reporte Cantidad de Trámites por Tipo (imprimir, exportar).
Referencia	RF-21

Tabla 23 Descripción Gestionar Cantidad de Trámites por Tipo.

Caso de Uso	Mostrar Cantidad de Trámites
Actores	Administrador, Técnico y Especialista
Importancia	Alto
Descripción	El administrador, técnico o especialista realizan las opciones pertinentes para Mostrar el gráfico cantidad de trámites.
Referencia	RF-22

Tabla 24 Descripción Mostrar Cantidad de Trámites.

Caso de Uso	Mostrar Cantidad de Trámites por Estado de Ejecución
Actores	Administrador, Técnico y Especialista
Importancia	Alto
Descripción	El administrador, técnico o especialista realizan las opciones pertinentes para Mostrar el gráfico cantidad de trámites por estado de ejecución.
Referencia	RF-23

Tabla 25 Descripción Mostrar Cantidad de Trámites por Estado de Ejecución.

Caso de Uso	Mostrar Cantidad de Trámites por Tipo
Actores	Administrador, Técnico y Especialista
Importancia	Alto
Descripción	El administrador, técnico o especialista realizan las opciones pertinentes para Mostrar el gráfico cantidad de trámites por tipo.
Referencia	RF-24

Tabla 26 Descripción Mostrar Cantidad de Trámites por Tipo.

ANEXO I

Trámites Nomencladores Reportes Administración Ayuda Cerrar Sesión

+ Agregar Dictamen técnico Buscar

Fecha	Tipo de trámite	Nombre propietario	Carnet	Dirección Propietario	Acciones
2014-03-27	Estatal	sin propietario	47012536478	La Murcielaguina	
2014-02-13	Estatal	Casa del Teniente a Gobernador	78102836947	Morales Lemus	
2014-03-11	Estatal	Artes Escénicas	58062832715	Martí entre Libertad y Maceo	
2014-03-12	Estatal	Empresa Forestal Holguín	56052714836	Libertad No. 162 entre Cables y Aricochea	
2014-03-26	Estatal	Dirección Municipal de Cultura. Gibara.	56072931795	Ave. Lenin.	
2014-03-27	Estatal	Local FMC Gibara	54071937482	Gibara	

« « 1 » » »
Mostrar por página: 25 registros
Mostrando 1 a 6 de 6 registros

Oficina de Monumentos y Sitios Históricos de Holguín
SiGTram - 2014

Fig: 12 Interfaz Dictamen Técnico

Trámites Nomencladores Reportes Administración Ayuda Cerrar Sesión

+ Agregar Empresa Exportar Imprimir Buscar

Empresa	Acciones
Arquitecto de la Comunidad. Holguín	
UMV	
Planificación Física Municipal	
Planificación Física Provincial	
Empresa Municipal de Comercio (Inversiones)	
UMBRALES	
OMSH	

« « 1 » » »
Mostrar por página: 25 registros
Mostrando 1 a 7 de 7 registros

Oficina de Monumentos y Sitios Históricos de Holguín
SiGTram - 2014

Fig: 13 Interfaz Nomenclador Empresa



Fig: 14 Interfaz de Reporte

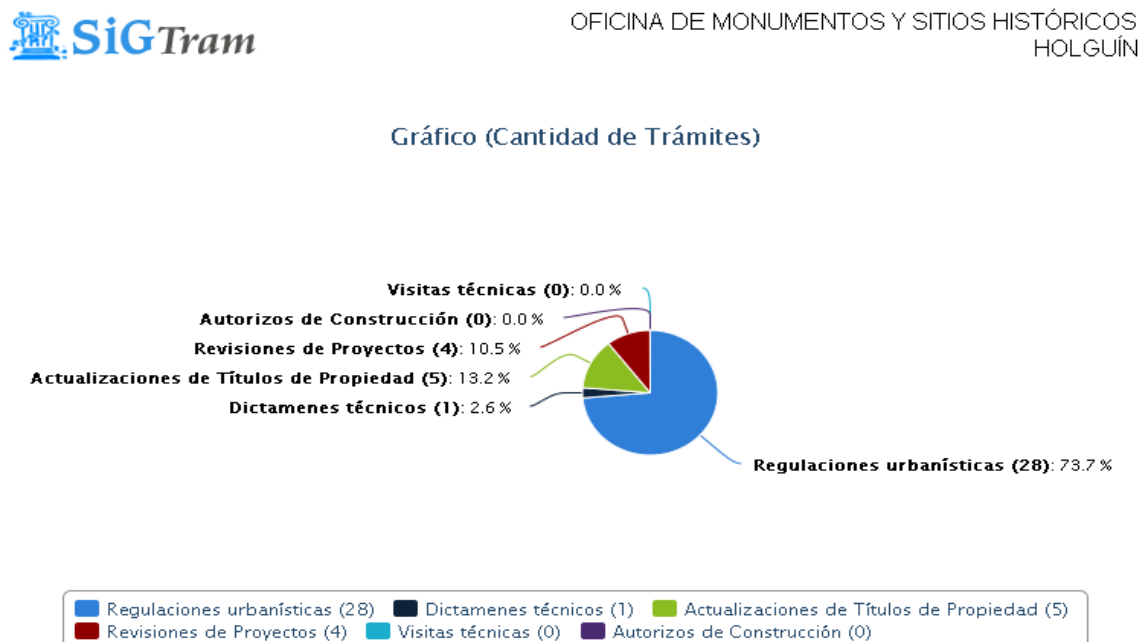


Fig: 15 Interfaz de Gráficos

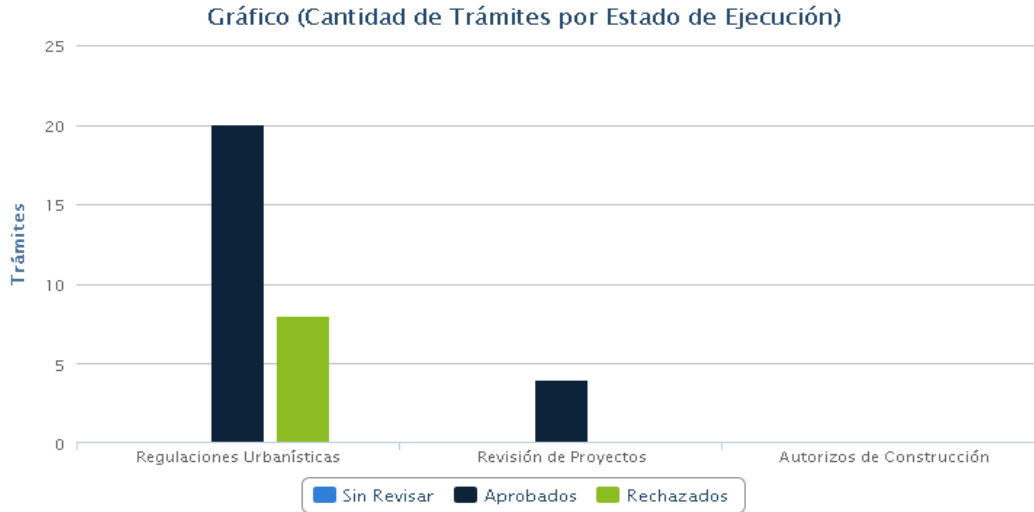


Fig: 16 Interfaz de Gráficos 1

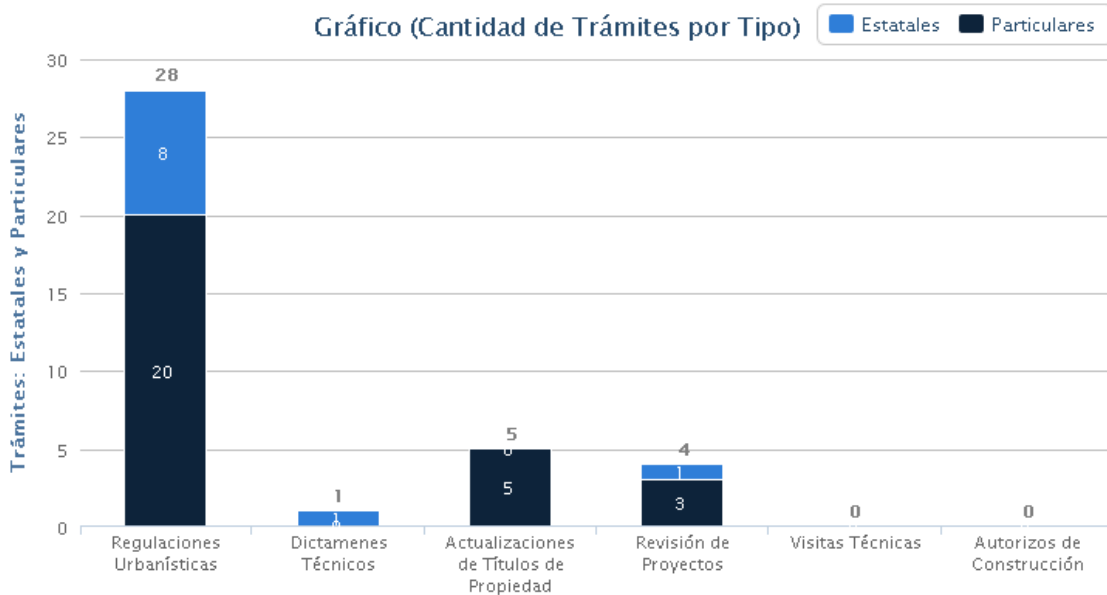


Fig: 17 Interfaz de Gráficos 2



Fig: 18 Interfaz Configuración del Sistema



Fig: 19 Interfaz Créditos del Sistema

ANEXO II

Caso de uso	Agregar Regulaciones RF1
Caso de prueba	Agregar Regulaciones ya existente
Entrada	El administrador hace clic en la pestaña Regulaciones para Agregar Regulación en la interfaz activa. El sistema muestra el formulario “Agregar Regulación” para la entrada de los datos de la Regulación. El administrador o técnico especifica datos de la misma y hace clic en el botón Guardar. El sistema comprueba que no existan campos vacíos.
Resultado	El sistema agrega la regulación según su estado (sin revisar, aprobada o rechazada), muestra un mensaje de confirmación al administrador o técnico y muestra opciones para volver a la lista de regulaciones del sistema.
Condiciones	El administrador y técnico deben haber iniciado sesión.

Tabla 27 Caso de Prueba Agregar Regulación ya existente

Caso de uso	Agregar Usuario RF 12
Caso de prueba	Agregar Usuario ya existente
Entrada	El administrador en el Menú Administración hace clic en Usuario para Agregar Usuario. El sistema muestra el formulario “Agregar Usuario” para la entrada de los datos del nuevo usuario. El administrador especifica datos del mismo y hace clic en el botón Guardar. El sistema comprueba que no existan campos vacíos.
Resultado	El sistema no agrega usuario, muestra un mensaje de donde especifica que el campo usuario tiene un valor único. Muestra opciones para volver a la lista de usuarios del sistema.
Condiciones	El administrador debe haber iniciado sesión.

Tabla 28 Caso de Prueba Agregar Usuario ya existente