

UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA



< Sistema para la gestión del monitoreo y manejo sostenible de playas
mediante técnicas de minerías de datos >

Trabajo de Diploma para optar por el título de Ingeniero Informático

Autor: < Yaisel González Hernández >

Tutores: < Ing. Eduardo Javier Berrio Turiño

Dra. Liz Pérez Martínez >

Matanzas, 2022

PENSAMIENTOS

“Ser Culto, es el único modo de ser libre”

José Martí

“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; [...]”

Fidel Castro Ruz

“Cuando quieres algo, todo el universo conspira para que realices tu deseo.”

Paulo Coelho

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

Steve Jobs

DEDICATORIA

A mis padres por sus sabios consejos, su amor incondicional y por guiarme siempre por el camino correcto.

Gracias por creer en mí cuando me sentí derrotado y darme fuerzas para seguir adelante, gracias por ayudarme a luchar por mis sueños de ser un profesional, por haber estado apoyándome en cada paso de mi vida. Simplemente les doy las gracias por ser los mejores padres del mundo, los quiero mucho.

También se lo dedico a mi hermano por ayudarme durante toda mi carrera y a mi familia en general, a esa que siempre creyó y confió que yo sería el ingeniero de la casa y además a mi tío y abuela que aunque ya no estén en vida fueron grandes pilares en mi educación. A mi sobrina se lo dedico para motivarla a que estudie y sea una gran profesional en la vida.

A mis vecinos y amigos.

A todos los que confiaron en que me convertiría en un profesional.

A mis compañeros de aula por elegirme como su jefe de brigada y representarlos en la universidad.

A mis tutores, porque este proyecto es parte de su gran dedicación.

AGRADECIMIENTOS

A mi madre Iraida y padre Israel, estos dos grandes pilares en mi educación y motores que me han impulsado en mi vida. Mi madre por estar días de desvelo rezo a rezo por mis pruebas, por estar cada lunes pendiente de todo, a ella que le ha dedicado su vida para que yo fuera un hombre de bien, con educación y un gran profesional. Mi padre que por su trabajo día a día era mi sustento en estos cinco años de carrera, ese que cuando más obstinado me sentía me daba fuerzas para seguir adelante, ese que presume de mi ante todos de tal forma que dice que yo soy el ingeniero de la casa. A ellos dos, mis padres, mis tesoros, les agradezco toda su dedicación en mi vida. A mi hermano Osiel, mi ídolo por ayudarme y apoyarme en todo y darme buenos consejos para la vida y a su esposa Mirita también.

Le agradezco a mi familia por apoyarme en todo y por siempre creer en mí, a mi prima Yakelin y Yaima y sus maridos por correr conmigo para todos lados y apoyarme en todo, sin ustedes no hubiera podido matricularme en la universidad. A mi tía Dalia por su apoyo. A mi tío Julio que aunque ya no esté entre nosotros fue un motor impulsor en mi carrera porque siempre quiso verme graduado.

A mis vecinos y en especial a Margarita y a su hermano José Alberto (Mi tío postizo) los que me ayudaron incondicionalmente en todos los sentidos y en la infraestructura tecnológica. A Yosly mi vecina del frente que me ayudó mucho con su caudal de conocimientos y me dio muchos consejos para la vida universitaria. A todos en general.

A mis compañeros de aula por compartir estos cinco años conmigo, a Roxana que fuimos dos servidores mutuos de constante estudio y estrés, a mis compañeros del cuarto por tantos momentos, Adrian, Luis, Daniel y el mejor de todos, mi amigo y hermano Yadian q estuvo en las buenas y en las malas a mi lado. A mis amigos, a Melissa Jiménez que ha estado pendiente a mí en todo, a Jorgito, a Niurka, Melisa Arguelles, Melisa Mariam y todos en general. A Leonardo que fue mi gran apoyo y su ayuda incondicional fue la base del desarrollo del software, muy agradecido por siempre mi amigo, porque cada vez que te necesité para ayuda nunca te negaste y eso vale oro.

A mis TUTORES Eduardo y Liz, les agradezco infinitamente por su ayuda, preocupación y dedicación, por acogernos como sus niños estrellas y por su apoyo en todo momento.

A mi novia por apoyarme en los momentos más difíciles y creer en mí y tener paciencia, a mis suegros y su familia que me acogieron como uno más entre sus miembros.

A todos mis profesores que me ayudaron a formarme como profesional y a todos los que me ayudaron y apoyaron de manera en general, un millón de gracias.

DECLARACIÓN DE AUTORÍA

Yo, Yaisel González Hernández, declaro que soy el único autor del trabajo sistema para la gestión del monitoreo y manejo sostenible de playas mediante técnicas de minerías de datos y autorizo a la Universidad de Matanzas, y en especial, a la Facultad de Ciencias Técnicas, a que hagan el uso que estimen pertinente de estos. Para que así conste, firmo la presente a los 28 días del mes de noviembre del 2022.



Firma del autor

Yaisel González Hernández



Firma de los tutores

Ing. Eduardo Javier Berrio Turiño

Dra. Liz Pérez Martínez

OPINIÓN DE LOS TUTORES

DATOS PERSONALES DEL TUTOR

Nombre y apellidos: Eduardo Javier Berrio Turiño.

Centro de trabajo: Universidad de Matanzas.

Organismo a que pertenece: Ministerio de Educación Superior – MES.

Cargo que ocupa: Profesor e Investigador.

Especialidad de la que es graduado: Ingeniero Informática. Universidad de Matanzas, 2018.

Categoría docente o investigativa: Asistente.

DATOS DE LA TESIS Y EL DIPLOMANTE

Nombre y apellidos: Yaisel González Hernández.

Centro de estudio: Universidad de Matanzas sede “Camilo Cienfuegos”.

Título de la Tesis: Sistema para la gestión del monitoreo y manejo sostenible de playas mediante técnicas de minerías de datos

OPINION SOBRE EL TRABAJO

La tesis presentada posee gran actualidad, pues intenta resolver un problema real presente en la provincia de Matanzas, la cual es de vital importancia por las características del territorio, y además contribuye a la informatización de nuestra sociedad.

El tutor de este trabajo de diploma considera que, durante su ejecución, el estudiante mostró las cualidades que a continuación se detallan:

Independencia y capacidad de investigación. Fueron jornadas adentrándose en temas complejos y nuevos pues se apartan en gran medida de los contenidos recibidos durante su formación, y logró captar con rapidez y profesionalidad el conocimiento necesario para enfrentar el problema planteado. Aportó soluciones importantes que demuestran su madurez como investigador. Fue consecuente con los aspectos tanto metodológicos como de la investigación científica propiamente. Esto le permitió una feliz culminación del método desarrollado, de la documentación y de las pruebas realizadas.

En el trabajo se aprecia rigor, manifestado desde el tratamiento de los conceptos estudiados y referenciados en la bibliografía, hasta las conclusiones, lo que ha contribuido a la correcta solución de los problemas encontrados.

Una gran cantidad de clases y métodos, un producto bien concebido y validaciones correctamente realizadas para culminar su investigación, unido a una excelente planificación de tiempo y recursos, dieron una gran calidad al trabajo obtenido.

También fueron horas de revisión, discusión y consenso en las que demostró notables cualidades para la investigación. El trabajo que hoy presenta y que sintetiza un periodo de aprendizaje no solo académico.

Como resultado se derivó en la obtención de un producto de software al nivel de las exigencias y expectativas de la Universidad respectivamente. Por todo lo anteriormente señalado, considero que el estudiante Yaisel González Hernández reúne los requisitos para el título de Ingeniero Informático y espero le sea otorgada la mejor calificación de este Tribunal. Espero que su labor investigativa no se detenga hasta alcanzar la cima.



Ing. Eduardo J. Berrio Turiño

Dpto. Informática

Universidad de Matanzas

Noviembre/2022

RESUMEN

En medios de la informatización de la sociedad, la modernización de las playas para hacerlas inteligentes y la inexistencia de una aplicación informática de gestión de playas, pues el trabajo era extremadamente manual y con pérdida e alteración de la información, surgió la tarea de crear un software que gestionara dicho litoral y la administrara ambientalmente de forma que creara un histórico seguro de todos los datos relacionados con ésta e hiciera predicciones del comportamiento de sus parámetros más significativos, así construir el marco teórico referencial de la investigación sobre la base de la gerencia de la información, monitoreo y manejo de las mismas. Matanzas al contar con una de las mejores playas del mundo, Varadero, fue necesario ingeniar ideas para garantizar un medio ambiente, una costa y un turismo seguro y sostenible. A partir de esta situación se decide desarrollar una aplicación web para la gestión, la cual permite a los especialistas realizar su trabajo de manera segura y en menos tiempo del que se requiere en la actualidad, donde proporciona también confidencialidad a los datos con los que se trabaja. Se empleó la metodología ágil de desarrollo de software **Scrum**, además de apoyarse para una mejor evolución de software en algunos de los diagramas de la metodología pesadas, el lenguaje de programación Python con su framework Django y como gestor de base de datos PostgreSQL. Se obtuvo un software funcional, el cual se le realizaron una serie de pruebas que permitieron constatar su funcionalidad y verificar que cumpliera con las necesidades y exigencias del cliente.

SUMMARY

In the midst of the computerization of society, the modernization of beaches to make them intelligent and the non-existence of a computer application for beach management, since the work was extremely manual and with loss and alteration of information, the task of creating a software to manage this coastline and administer it environmentally, thus creating a safe history of all the data related to it and making predictions of the behavior of its most significant parameters, thus building the theoretical reference framework of the research on the basis of information management, monitoring and management of the same, arose. Matanzas, having one of the best beaches in the world, Varadero, it was necessary to come up with ideas to guarantee a safe and sustainable environment, coast and tourism. Based on this situation, it was decided to develop a web application for management, which allows specialists to perform their work safely and in less time than is currently required, also providing confidentiality to the data they work with. The agile methodology of software development Scrum was used, in addition to support for a better software evolution in some of the diagrams of the heavy methodology, the programming language Python with its framework Django and as database manager PostgreSQL. A functional software was obtained, which was subjected to a series of tests that allowed to verify its functionality and verify that it met the needs and requirements of the client.

1 ÍNDICE GENERAL

Introducción.....	1
Capítulo I: Marco teórico-referencial	6
1.1 Introducción.	6
1.2 Objeto de estudio.....	6
1.3 Antecedentes.....	6
1.4 Justificación	7
1.5 Metodología de desarrollo	9
1.6 Herramientas y tecnologías.....	13
1.6.1 Arquitectura cliente-servidor:	13
1.6.2 Software Libre	13
1.6.3 Lenguaje de modelado.....	14
1.6.4 Lenguaje de programación.....	14
1.6.5 Marco de Trabajo Framework	16
1.6.6 Herramientas de desarrollo	17
1.6.7 Herramientas CASE: Visual Paradigm-UML.....	19
1.6.8 Patrones de diseño.....	20
1.7 Conclusiones del capítulo.	21
2 Capítulo 2. Análisis, Diseño y Desarrollo de la Solución Propuesta.	22
2.1 Introducción.	22
2.2 Descripción de la solución	22
2.3 Requisitos No Funcionales.	22
2.4 Etapa de Planificación	24
2.4.1 Pila del Producto (SPRINT BACKLOG).....	24
2.4.2 Pila de Sprint (SPRINT BACKLOG).....	25
2.4.3 Planificación de Sprint Del Proyecto	27
2.5 Historia de Usuarios.....	40
2.6 Modelado	44
2.6.1 Diseño de la base de datos.....	44
2.6.2 Modelo Conceptual de la Base de Datos	45
2.7 Descripción del Negocio.....	45
2.7.1 BPMN (Modelo de Proceso de Negocio y Notación).....	46
2.8 Seguridad del módulo.....	46
2.9 Tratamiento de errores.	50
2.10 Análisis de factibilidad.	50
2.11 Conclusiones del capítulo.	53

3	Capítulo 3. Validación de la Solución Propuesta.	54
3.1	Introducción	54
3.2	Descripción de la Propuesta de Solución	54
3.3	Pruebas al Software	58
3.3.1	Tipos de Pruebas de Software Generales	59
3.3.2	Plan de Pruebas	61
3.3.3	Pruebas de Aceptación	64
3.3.4	Pruebas de Caja Negra	70
3.4	Análisis de los resultados obtenidos	74
3.5	Conclusiones del Capítulo	75
4	Conclusiones Generales	76
5	Recomendaciones	77
6	Bibliografía	78

ÍNDICE DE TABLAS

1.	<i>Tabla #1: Comparación entre las Metodología Ágil y Tradicional</i>	9
2.	<i>Tabla 2: Pila de Producto (PRODUCT BACKLOG)</i>	24
3.	<i>Tabla 3: Pila de Sprint (SPRINT BACKLOG)</i>	26
4.	<i>Tabla 4: Pila de Sprint (SPRINT BACKLOG)</i>	27
5.	<i>Tabla 5: Planificación del SPRINT 2</i>	29
6.	<i>Tabla 6: Planificación del SPRINT 3</i>	31
7.	<i>Tabla 7: Planificación del SPRINT 4</i>	33
8.	<i>Tabla 8: Planificación del SPRINT 5</i>	37
9.	<i>Tabla 9: Planificación del SPRINT 6</i>	38
10.	<i>Tabla 10: Historia de Usuario 1</i>	40
11.	<i>Tabla 11: Historia de Usuario 2</i>	41
12.	<i>Tabla 12: Historia de Usuario 3</i>	43
13.	<i>Tabla #13: Caso de uso gestionar usuario (Seguridad del Módulo)</i>	46
14.	<i>Tabla #14: Caso de uso gestionar roles (Seguridad del Módulo)</i>	47
15.	<i>Tabla #15: Caso de uso autenticarse (Seguridad del Módulo)</i>	48
16.	<i>Tabla #16: Caso de uso cerrar sesión (Seguridad del Módulo)</i>	49
17.	<i>Tabla # 17: Factor de Ajuste</i>	50
18.	<i>Tabla # 18: Tabla 1 de Componente para medición de proyectos</i>	52
19.	<i>Tabla # 19: Tabla 2 de Componente para medición de proyectos</i>	53
20.	<i>Tabla #20: Plan de Pruebas (Pruebas al Software)</i>	61
21.	<i>Tabla #21: Prueba de Aceptación 1 (Pruebas al Software)</i>	65
22.	<i>Tabla #22: Prueba de Aceptación 2 (Pruebas al Software)</i>	66
23.	<i>Tabla #23: Prueba de Aceptación 3 (Pruebas al Software)</i>	66
24.	<i>Tabla #24: Prueba de Aceptación 4 (Pruebas al Software)</i>	67
25.	<i>Tabla #25: Prueba de Aceptación 5 (Pruebas al Software)</i>	68
26.	<i>Tabla #26: Prueba de Aceptación 6 (Pruebas al Software)</i>	68
27.	<i>Tabla #27: Prueba de Aceptación 7 (Pruebas al Software)</i>	69
28.	<i>Tabla #28: Prueba de Caja Negra 1 (Pruebas al Software)</i>	71
29.	<i>Tabla #29: Prueba de Caja Negra 2 (Pruebas al Software)</i>	72
30.	<i>Tabla #30: Prueba de Caja Negra 3 (Pruebas al Software)</i>	73

ÍNDICE DE FIGURAS

Figura 1. Modelo Conceptual de la Base de Datos.....	45
Figura 2. Modelo BPMN.....	46
Figura 3. Primera vista del sistema que muestra el administrador Django.	55
Figura 4. Segunda vista del sistema que muestra las zonas.....	56
Figura 5. Tercera vista general del sistema que muestra uno de los elemento de la sección o sub-módulo caracterización.	56
Figura 6. Cuarta vista general del sistema que muestra la sección o sub-módulo estado ecológico-ambiental.....	57
Figura 7. Quinta vista general del sistema que muestra la sección o sub-módulo del análisis de la gestión.....	57
Figura 8. Sexta vista general del sistema que muestra la sección o sub-módulo del monitoreo.....	58
Figura 9. Séptima vista general del sistema que muestra la acción del botón de predicciones.....	58

Introducción

La zona costera es el lugar donde ocurren actualmente, los cambios más intensos, tanto naturales como de desarrollo de actividades humanas, y por lo tanto presenta mayor vulnerabilidad (Cendero Uceda, Sánchez-Arcilla Conejo, & Zazo Cardeñas). La importancia de esta zona radica en la alta diversidad de bienes y servicios ambientales que brinda para el desarrollo de obras y acciones para aprovechar su riqueza. Lo anterior ha dado como resultado zonas de alto dinamismo económico y social derivado de actividades productivas como las industrias petroleras y portuarias, acuícolas, pesqueras, mineras urbanas y turísticas (Quijano Poumián & Rodríguez Aragón, 2004).

Es necesario considerar que las zonas costeras son diferentes tanto en las actividades que se llevan a cabo como en la forma y ubicación de las mismas. Conocer las características, las condiciones y los problemas tanto ambientales y sociales específicos de cada zona costera permite dirigir la acción política para responder, atender y transformarlos en objetivos para proporcionar soluciones. Igualmente, el considerar un panorama más amplio que integre a los actores y usuarios de estas zonas permitiría conocer la relación entre el estado del medio ambiente y la presión que ejercen los diversos sectores involucrados. Un aspecto clave de la Gestión Integral de Zonas Costeras es su complemento con la gestión estratégica, pues fundamenta el papel de los actores para orientar sus acciones con base en la visión, liderazgo, coordinación y dirección hacia la sustentabilidad de las costas. Y tan así se complementan que el propósito de estos modelos es tratar de cambiar la estructura de gestión y tener una visión a largo plazo.

En Cuba un hecho que tuvo gran importancia en materia ambiental fue la creación en el país del Ministerio de Ciencia, Tecnología y Medio ambiente, en 1994, así como la aprobación de la Ley 81 de Medio Ambiente y un conjunto de Decretos y Resoluciones complementarias. Otro hecho trascendental en materia de legislación ambiental en el país, es la aprobación del Decreto-Ley 212, de Gestión de la zona costera, que marca un comienzo en el ordenamiento y gestión de todas las zonas costeras de Cuba, lo que tuvo gran repercusión en el litoral norte de la provincia de Matanzas, dadas sus características físico-geográficas y la diversidad e intensidad de las actividades económicas y de las transformaciones socio-ambientales en proceso. (Hernández Cabrera, Alfonso Martínez, Sánchez Roque, & Pérez Martínez, 2019) .

La gestión integrada y ecosistémica de playas en Cuba se hace basada en indicadores parciales para cada uno de los elementos de este espacio del litoral, realizándose la gestión de forma fragmentada, por tipos de playas, donde la mayoría de los esfuerzos

están centrados en playas turísticas, a pesar de esto en los últimos años ha existido un cambio visión destinándose recursos y esfuerzos para llevar a cabo un desarrollo más amigable con el medio ambiente, sin embargo la generación de iniciativas son insuficientes, por lo que es momento de que los actores locales participen para regular, a corto plazo, las acciones que afectan las playas.

Por otra parte, el empleo de las nuevas tecnologías de la información y las comunicaciones, debido a su acelerado avance, se han comenzado a incorporar en la protección y gestión de estos sistemas costeros, donde (Yepes, 2002-2018), plantea un cambio de paradigma en la gestión, dado por el tránsito de las playas certificadas a las playas inteligentes. Una playa con sistema de gestión certificado no implica una playa inteligente, la adopción de nuevas tecnologías que permitan la interacción del usuario con el gestor o con otros usuarios, conocer en tiempo real información de la playa no es suficiente para considerar que una playa es inteligente, el cambio de paradigma requiere algo más complejo y de mayor trascendencia, supone usar toda la información disponible gracias a las nuevas tecnologías en la toma de decisiones, estos sistemas inteligentes que interactúan con la playa no solo requieren la disposición de una base de datos de información actualizada, sino por el aprendizaje y por la toma inteligente de decisiones en base al análisis científico de dicha información, actualmente se ha avanzado en los aspectos relacionados con la inteligencia artificial y con la minería de datos, estas herramientas facilitan la toma de decisiones basándose en la información disponible y su procesamiento mediante la simulación de escenarios y técnicas de decisión multiobjetivos y multicriterios. No se podrá decir que una playa es inteligente hasta que se incorporen tecnologías que permitan la toma de decisiones basándose en la información disponible y la retroalimentación, lo cual supone un aprendizaje y un proceso de mejora continua.

De momento es un área donde se están dando los primeros pasos a nivel global y se están creando condiciones necesarias para la puesta en marcha, para ello no solo es necesario las nuevas tecnologías, sino la elaboración de un sistema que permita la participación de todas las partes implicadas alimentando de información de calidad que soporte la toma de decisiones mediante la inteligencia artificial, pero siempre recaerá sobre el decisor la responsabilidad última de sus acciones. En nuestro país es un terreno que no ha sido explorado y no existen investigaciones de este tipo, pero es muy necesario para estar acorde con las políticas de desarrollo del país y el proceso de informatización de nuestra sociedad.

La situación presentada anteriormente evidencia una gestión deficiente de las playas, como consecuencia de la carencia de una visión preventiva que permita reducir efectos

previsibles, insuficiencia en la recogida de indicadores e información, un nivel de gestión débil, centrado básicamente en la limpieza y los primeros auxilios a las playas y la ausencia de competencias informacionales que se apoye en el uso de las tecnologías como herramienta de apoyo a la toma de decisiones, lo que deriva como **situación problemática** la insuficiente aplicación de las nuevas tecnologías y herramientas informáticas para la recopilación y el análisis de información, lo que dificulta disponer de información clave, pertinente, actualizada, centraliza y relevante, para apoyar los procesos de toma de decisiones y eliminar subjetividad de los mecanismos actuales.

A partir de lo antes expuesto se deriva en apretada síntesis la situación problemática con la que se vincula la investigación, se formuló como **problema científico** ¿es posible elevar la efectividad en la gestión integral de playas a partir de una herramienta automatizada para este proceso?

Para la solución del problema se trazó la **hipótesis** que rige la investigación, si se desarrolla un Sistema para la Gestión Integral de Playas que contribuirá a elevar la efectividad de este proceso.

Objeto de estudio: Proceso de manejo de los datos resultantes de la gestión integral de playa.

Se define como **objetivo general** de la investigación: desarrollar un Sistema para la Gestión Integral de Playas que contribuya a la elevación de la eficacia y la eficiencia del proceso de gestión en estas zonas costeras.

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

1. Analizar la bibliografía que permita sentar las bases de la investigación y conformar el marco teórico referencial de la misma.
2. Diseñar el Sistema para la Gestión Integral de Playas utilizando la metodología de desarrollo **SCRUM**.
3. Validar el Sistema para la Gestión Integral de Playas a partir de la aplicación de pruebas de software.

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

Dentro de los métodos teóricos:

- Método de análisis histórico – lógico: permitió estudiar la trayectoria y desarrollo de los sistemas de gestión ambiental empresarial existentes, así como la realización de este proceso en nuestro país.

- Método de análisis y síntesis: este se precisó durante la revisión bibliográfica y el análisis de los resultados, permite descomponer lo complejo en sus partes y cualidades, la división del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.
- Método inductivo - deductivo: su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permite arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

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

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

Entre los aportes de la investigación se destacan:

- El teórico-investigativo, al sentar bases para futuras investigaciones.
- El práctico, al desarrollar una herramienta automatizada que asista a la manipulación de la información referente a la gestión integral de playas. El resultado esperado de este trabajo es contar con una herramienta desarrollada en ambiente web fácil de manipular y administrar, la que permitirá

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

- Una Introducción, donde se caracteriza la situación problemática y se fundamenta el problema científico a resolver.
- Capítulo I “Marco teórico referencial”. En el que se presenta una exposición detallada de los referentes teóricos que argumentan la propuesta y permiten un acercamiento al objeto de estudio. Además, se comenta sobre las tendencias y tecnologías actuales que serán usadas para el desarrollo del sistema.
- Capítulo II “Diseño de la solución propuesta”. En el que se argumenta la solución que se propone al problema de investigación mediante su descripción. Se presenta la planificación inicial del proyecto, con el empleo de la metodología de

desarrollo de software SCRUM, y se realiza un estudio tanto de factibilidad, como de los beneficios tangibles e intangibles para la realización del sistema.

- Capítulo III “Validación de la solución propuesta”. En el que se describe el software y se le realizan las pruebas con el objetivo de entregarle al cliente un producto totalmente funcional, que cumple con todos los requisitos demandados por el mismo y que satisfaga sus necesidades. Un apartado de conclusiones donde se verifica el cumplimiento de los objetivos trazados al inicio de la investigación.
- Las recomendaciones en la cual se plasman una serie de propuestas encaminadas a la continuidad de esta investigación.
- Y las referencias de la bibliografía citada.

Capítulo I: Marco teórico-referencial

1.1 Introducción.

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

1.2 Objeto de estudio.

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

El proceso de toma de decisiones también se ve afectado, ya que la información no llega a los decisores en tiempo, ni con la veracidad requerida para que el proceso fluya adecuadamente. El monitoreo ambiental de las empresas, la determinación de los problemas ambientales y sus causas, así como las áreas más afectadas, dependen de la subjetividad de los evaluadores y no de herramientas confiables. La realización de pronósticos y determinación de tendencias a partir de los datos históricos no es posible ya que en numerosas ocasiones esta información no existe.

1.3 Antecedentes

✓ **Aplicación de Gestión de Playas del 2014**

La aplicación acabada es capaz de gestionar todas las incidencias ocurridas en las playas, así como llevar un control del material prestado. También es capaz de llevar un seguimiento del estado meteorológico de ellas y de tener un control sobre los trabajadores y zonas costeras. La empresa Inttegrum Consultora y Gestora del Mediterráneo S.L. es una empresa dedicada al desarrollo, mantenimiento y venta de diferentes soluciones software y hardware. Su cliente Mediterránea Grupo vino con un problema; necesitaba mejorar sus métodos de gestión del socorrista, meteorología, incidentes, salida de embarcaciones y el material prestado de las playas que tenía asignadas. También debía servir como ventaja a la hora de

concurrir por las playas en los ayuntamientos así como toda la preservación de las zonas costeras y datos de estas. (Goold, 2014)

✓ **App para control de aforo y reserva de espacio en las playas llamada WinPlus**

Es una aplicación de gestión de playas desarrollada por la Empresa “Informática del Este”, que es una empresa especializada en sistemas informáticos de control de Accesos, Gestión de tiempos y Seguridad Integral ubicada en Valencia, España. Al llegar a la playa, desde la APP WinPlus Playas se lee el código QR de la parcela. Si la reserva es válida el usuario podrá hacer uso de ella y ver todos los datos de interés de ésta. Dicha app está diseñada entre otros elementos, dar respuesta al control de aforo a las playas para garantizar el distanciamiento social requerido; los usuarios podrán descargarse en el Smart phone la app y realizar la reserva de las parcelas existentes para su uso y disfrute y además ver las condiciones costera en que se encuentra esta; la aplicación se auto configurará para cada playa según las necesidades de cada ayuntamiento y así poder realizar una gestión efectiva. (Informática del Este, 2021)

✓ **Sistema de gestión de incidencias para usuarios de playas v1.0, ClaimBeachApp 2021**

ClaimBeachApp es un sistema para la gestión de incidencias y quejas orientada al ciudadano sobre diferentes aspectos de la playa y su entorno. El sistema ofrece una plataforma completamente accesible y usable mediante una aplicación móvil que permite a los ciudadanos enviar incidencias sobre diferentes recursos y situaciones en la playa, recoger y clasificar dichas incidencias y mostrar al ciudadano información sobre el estado de sus peticiones. Desarrollada Universidad de Alicante. Departamento de Tecnología Informática y Computación, Universidad de Alicante. Departamento de Ingeniería Civil. (Universidad de Alicante, 2022) (Marcos-Jorquera, Gilart, Aragónés, López, & Ignacio, 2019)

1.4 Justificación

Como justificación a la elaboración de este sistema, primeramente se partió de un equipo multidisciplinario que trabaja en el Observatorio Costa Atenas en Matanzas, donde a partir de investigaciones y datos que se fueron necesitando para el trabajo diario y la experiencia en demás proyectos, pues surgió la brecha o inconveniente de que a la hora de buscar datos no existían, el seguimiento de los indicadores no era constante, había pérdida de información e inseguridad en los datos por parte de los especialistas que atienden el monitoreo ambiental de playas, los datos podían ser falseados y modificados por cualquier personal, otros eran de difícil acceso y estaban

incompletos y todo eso sumado a que el proceso era extremadamente manual. Existen otros software a nivel mundial relacionados con este proceso, pero en primera instancia no satisfacía lo que se quería, eran software de pago que incumplía con las políticas de nuestro país y el ministerio de informática y las comunicaciones de utilizar software libres y además no cumplían con las normas cubanas establecidas por el CITMA (Ministerio de Ciencia Tecnología y Medio Ambiente) que es quien rige Medio Ambiente en nuestro país. La adopción de nuevas tecnologías que permitan la interacción del usuario con el gestor o con otros usuarios, conocer en tiempo real información de la playa no es suficiente para considerar que una playa es inteligente, el cambio de paradigma requiere algo más complejo y de mayor trascendencia, supone usar toda la información disponible gracias a las nuevas tecnologías en la toma de decisiones, estos sistemas inteligentes que interactúan con la playa no solo requieren la disposición de una base de datos de información actualizada, sino por el aprendizaje y por la toma inteligente de decisiones en base al análisis científico de dicha información, actualmente se ha avanzado en los aspectos relacionados con la inteligencia artificial y con la minería de datos, estas herramientas facilitan la toma de decisiones basándose en la información disponible y su procesamiento mediante la simulación de escenarios y técnicas de decisión multiobjetivos y multicriterios. No se podrá decir que una playa es inteligente hasta que se incorporen tecnologías que permitan la toma de decisiones basándose en la información disponible y la retroalimentación, lo cual supone un aprendizaje y un proceso de mejora continua.

La alimentación de la información disponible puede realizarse en parte de forma automática la que recorre a las actuales tecnologías de la información y la comunicación (sistemas de visión artificial para la monitorización del uso de las playas, interacción en tiempo real de usuarios y gestores, etc.) todos estos datos son valiosos pero son insuficientes y requieren de un procesamiento en tiempo real para transformarlos en información valiosa en la toma de decisiones, sin embargo, requiere de otro tipo de información de elaboración más compleja, pero de elevado valor en la toma de decisiones como son los datos actualizado del monitoreo físico de la playa, paneles de expertos y de las partes interesadas, pues no solo es favorecer la participación ciudadana, pues también es necesario tener herramientas que permitan calibrar estos nuevos procesos.

Un paso importante previo a la toma de decisiones es el tratamiento o explotación de los datos para su análisis, este aspecto se encuentra poco desarrollado en las playas e incluso en el ámbito de la investigación científica, requiere de la estadística y las ciencias de la computación para descubrir patrones y para inferir modelos predictivos, que

permitan tomar decisiones, donde los modelos actuales basados en la inteligencia artificial y el aprendizaje automático son los más apropiados.

1.5 Metodología de desarrollo

Las metodologías de desarrollo de software son un conjunto de pasos y procedimientos que deben seguirse para llevar a cabo el desarrollo de software con calidad. Estas brindan un conjunto de detalles organizativos añaden rigurosidad y normas de forma que permita a los integrantes de un equipo de desarrollo puedan seguir un criterio común a la hora de realizar las tareas durante el desarrollo de un software. La constante innovación tecnológica hace que cada vez sea más necesaria la aplicación de nuevas metodologías adaptadas a los nuevos tiempos. (Mesa, 2015)

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

En la siguiente tabla se muestra una comparación entre metodologías ágiles y no ágiles. (Sánchez & Letelier, 2003)

1. Tabla #1: Comparación entre las Metodología Ágil y Tradicional

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

Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.

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

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

SCRUM como metodología de desarrollo

Con esta metodología vamos a crear manuales básicos, que nos permite presentar y aclarar todos los procesos y procedimientos necesarios que se van a seguir, y en los que el equipo SCRUM tiene que regirse para terminar con éxito esta solución en los tiempos estimados. Además, podemos revisar y aclarar todos los escenarios en donde se va a utilizar la aplicación que controla repuesto a la medida y generará pedidos remotos, identificando las listas priorizadas y las historias de los usuarios que especifiquen el alcance del proyecto. Justificación para su uso en el proyecto:

Lo ágil de esta metodología, nos permite ir presentando avance para revisar, probar, lo que no se definió al inicio y confirmar los controles con los usuarios finales para que queden aprobados y firmados. Esto evita retrasos, ya que de otra forma recién al final del desarrollo de la solución planteada empezamos a revisar y probar a ver cuáles son los problemas y lo que no se definió al inicio. (Library, 2022)

Según (Sánchez & Letelier, 2003), SCRUM se recomienda ser aplicado donde el "caos" es una constante. Lo que favorece a los desarrolladores noveles o de poca experiencia.

Según (Kniberg, 2007) las ventajas del uso de Scrum son:

- ✓ Scrum puede ser adoptado para manejar las prácticas de ingeniería que se utilizan en una organización, ya que no requiere ninguna práctica de ingeniería específica.
- ✓ Scrum puede mejorar la comunicación entre todos los equipos.

- ✓ Los problemas pueden ser transparentes es decir se hace fácil la detección de errores gracias a la entrega de los nuevos módulos o complementos entregados en cada sprint.
- ✓ Durante el sprint el equipo de Scrum se mantiene centrado y creativo sin interferencias. Lo cual es bueno para la productividad, ya que en este proceso se desarrolla lo que se ha tomado en cuenta en reuniones previas y se busca satisfacer las necesidades del cliente.

Según (Kniberg, 2007) los principios de SCRUM son:

- ✓ Eliminar el desperdicio: No perder tiempo a la hora de hacer cosas sin valor para el cliente.
- ✓ Construir la calidad con el producto: Inyectar la calidad directamente desde el principio.
- ✓ Crear conocimiento: No se puede tener conocimiento desde antes de empezar el proyecto.
- ✓ Diferir decisiones: Dejar las decisiones para el momento que se necesiten tomarlas, puesto que se puede tener más información en ese momento.
- ✓ Entregar rápido: Da competencia.
- ✓ Respetar a las personas: Respetar a todo el equipo, permite motivar.
- ✓ Optimizar el todo: Optimizar cada proceso para mejorar continuamente.

Según (Kniberg, 2007) los artefactos del SCRUM son:

- ✓ Product Backlog: es la lista maestra que contiene toda la funcionalidad deseada en el producto. La característica más importante es que la funcionalidad se encuentra ordenada por un orden de prioridad.
- ✓ Sprint Backlog: es la lista que contiene toda la funcionalidad que el equipo se comprometió a desarrollar durante un Sprint determinado.
- ✓ Burndown Chart: muestra un acumulativo del trabajo hecho, día-a-día.

En el desarrollo de la solución, se hará uso de una metodología ágil denominada SCRUM (Schwaber & Sutherland, 2013), la cual aplica sus premisas conceptuales en resolver el desarrollo evolutivo de aplicaciones, para SCRUM el cliente es pieza fundamental en el desarrollo de software, se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo, esta metodología va a permitir al cliente en cualquier momento realinear el software con los objetivos de aprendizaje que se tienen para él, ya que puede introducir cambios funcionales o de

prioridad en el inicio de cada nueva iteración de la aplicación. Además de usar esta tecnología se usarán algunos artefactos de las metodologías pesadas para dar una mayor explicación del negocio, por ejemplo: BPMN, diagrama de caso de uso del sistema y diagrama de despliegue.

La totalidad de los requerimientos a desarrollar, denominados historias de usuario son divididos en grupos en función de su prioridad relativa para luego ser implementados en ciclos de esfuerzos relativamente cortos llamados "sprints"; en el desarrollo de la aplicación se hará uso de los siguientes Sprint, con el fin de dividir las tareas que se pretenden llevar a cabo:

Sprint 1(máximo 4 horas): El equipo define de manera clara lo que desean ofrecer con la aplicación, se identifican las necesidades que tiene la comunidad para la que se pretende desarrollar.

Sprint 2(máximo 4 horas): Se realizan la planificación de la iteración, se elaboran la lista de tareas de la iteración para desarrollar la solución con los requisitos definidos.

Sprint 3: Ejecución de la iteración, el equipo realiza una reunión de sincronización (15 minutos máximos). En donde se inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con lo estipulado.

Sprint 4: Inspección y adaptación, se realiza la reunión de la revisión de la interacción, esta se divide en dos partes: Demostración (máximo 4 horas) se presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado. En función de los resultados mostrados y de los cambios que haya habido en el contexto de la solución, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re planificando el proyecto. Retrospectiva (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad.

La ejecución rigurosa satisface a la mayoría de los objetivos necesarios que sirven para obtener estos niveles; las pocas áreas del proceso no cubiertas directamente por no ser requeridos por SCRUM son en la práctica un requisito para el correcto desempeño de una organización dedicada a la construcción de software.

1.6 Herramientas y tecnologías.

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

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

1.6.1 Arquitectura cliente-servidor:

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

N-capas:

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

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

1.6.2 Software Libre

Las características del *software* libre, y las implicaciones que tiene su uso, son en gran medida consecuencias de estas libertades que ofrece. La libertad de redistribución proporciona un canal de distribución de gran eficiencia económica, y de muy bajo coste

para el productor. Las posibilidades de modificación y de redistribución de las modificaciones facilitan la evolución y mejora técnica de los programas.

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

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

1.6.3 Lenguaje de modelado UML (Unified Modeling Language)

El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). (Visual Paradigm, 2022)

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir, es el lenguaje en el que está descrito el modelo.

1.6.4 Lenguaje de programación

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

Python: es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma, posee una licencia de código abierto.

Su principal objetivo es la automatización de procesos, en este sentido, Python crea un código con gran legibilidad, que ahorra tiempo y recursos. Uno de sus puntos fuertes es que “comprueba los errores sobre la marcha” para solucionarlos cuando afectan a la memoria, lo que mantiene la integridad de la matriz y evita las complicaciones a la hora de escribir el código.

Marcado de Hipertexto (HTML): es el lenguaje que permite la generación de hipertextos en la World Wide Web. Los lenguajes de marcado de hipertexto construyen un conjunto de reglas que definen todo aquello que es parte de un documento digital, pero que no pertenece al texto del mismo. Definen la estructura y la semántica de un documento (Consortium, 2018). Hoy en día HTML 5 está tan de moda por las ventajas que ofrece que es imposible dejar de hacer uso de él por las nuevas etiquetas que incluye para video, audio y canvas. (J. D., 2012) (Albers, 2010) (Cantón, 2013)

Hojas de Estilo en Cascada (CSS): (del inglés Cascading Style Sheets) es un lenguaje formal usado para definir cómo se va a mostrar un documento en la pantalla, a imprimir o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. (Cristian, 2018)

Java Script: es el lenguaje de programación web del lado del cliente más extendido, permite crear funcionalidades específicas, con él se pueden generar páginas dinámicamente en función de las preferencias del usuario, validar datos en un formulario o modificar dinámicamente el contenido de la página, puede ser ejecutado sin la necesidad de instalar otro programa para ser visualizado, es un lenguaje más orientado a objetos. (Duarte, 2012) (Flanagan, 2002)

1.6.5 Marco de Trabajo Framework

Marco de Trabajo (Framework). Para facilitar la interacción del cliente con los datos se crea una aplicación Web. Esta se puede realizar con diferentes herramientas entre ellas se encuentran los Framework. "...nos referimos a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un Framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta."

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

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

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

1.6.5.1 Django

Se elige Django por ser multiplataforma. Está escrito en lenguaje Python, con código abierto y además proporciona una serie de características que facilitan el desarrollo rápido de páginas webs.

Algunas de sus características principales es que proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de

cualquier página realizada en Django y puede administrar varias hechas con él a partir de una misma instalación. La aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos). Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas. Interfaces de administración generadas automáticamente. (Holovaty & Kaplan-Moss, 2009)

En Django un proyecto es un sitio Web completo que consta de una o varias aplicaciones. El comando que crea un proyecto es `django-admin.py`. Simplemente, con `django-admin.py startproject transport` se crea un directorio `transport` que contiene varios ficheros `.py`:

- `__init__.py`: Define el directorio como un módulo Python válido.
- `manage.py`: Utilidad para gestionar el proyecto: arrancar servidor de pruebas, sincronizar modelos, etc.
- `settings.py`: Configuración del proyecto.
- `urls.py`: Gestión de las urls. Este fichero sería el controlador de la aplicación. Mapea las url entrantes a funciones Python definidas en módulos.

1.6.5.2 jQuery

Es un framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, permite manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX, además de ser software libre y de código abierto. JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Otra gran ventaja de JQuery es que permite cambiar el contenido de la página web sin necesidad de recargarla, utilizando DOM y AJAX de manera extremadamente sencilla gracias a su sintaxis. (Chaffer & Swedberg, 2009)

1.6.6 Herramientas de desarrollo

1.6.6.1 Sistemas gestores de bases de datos (SGBD)

Se desea almacenar la información de la unidad en un SGBD porque estos son capaces de detectar los datos repetidos. Además, cuentan con un subsistema de seguridad y autorización capaz de crear cuentas y especificar sus restricciones. Permiten la extracción, manipulación y modificación de la información según los diferentes tipos de permisos y el mantenimiento de esquemas. Hay tres características inherentes al enfoque de las bases de datos, la separación de los programas y los datos, el manejo de múltiples vistas de usuario y el empleo de un catálogo para almacenar la descripción

de los datos. El objetivo de la arquitectura de tres esquemas (capas) consiste en formar una separación entre las aplicaciones del usuario y la base de datos física.

PostgreSQL

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

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

- Es una base de datos 100% ACID.
- Integridad referencial.
- Completa documentación.
- Licencia BSD.
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
- Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl.
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.
- Herencia de tablas (Inheritance). (Martínez, 2010)

1.6.6.2 VisualStudio Code

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP y R, al igual que entornos de desarrollo web, como ASP.NET MVC y Django. Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros. Se incluye un

diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y pruebas de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga. Las ediciones Express se han diseñado para principiantes, aficionados y pequeños negocios, todas disponibles gratuitamente a través de la página de Microsoft. Se incluye una edición independiente para cada lenguaje: Visual Basic, Visual C++, Visual C#, Visual J# para programación .NET en Windows y Visual Web Developer para la creación de sitios web ASP.NET. Las ediciones Express carecen de algunas herramientas avanzadas de programación, así como de opciones de extensibilidad.

1.6.6.3 Anaconda

Anaconda es una distribución libre y abierta de los lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático (machine learning). Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software.

Las diferentes versiones de los paquetes se administran mediante el sistema de gestión de paquetes conda, el cual lo hace bastante sencillo de instalar, correr, y actualizar software de ciencia de datos y aprendizaje automático como puede ser Scikit-team, TensorFlow y SciPy.

La distribución Anaconda es utilizada por 6 millones de usuarios e incluye más de 250 paquetes de ciencia de datos válidos para Windows, Linux y MacOS. (Anaconda, 2022)

1.6.7 Herramientas CASE: Visual Paradigm-UML.

Visual Paradigm-UML es una herramienta Computer Aided Software Engineering o Ingeniería de Software Asistida por Computadora (CASE) destinada a aumentar la productividad en el desarrollo de software reduce el coste de la misma en términos de tiempo y de dinero. Puede ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas.

1.6.8 Patrones de diseño.

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

Patrón de Integración de Componentes (Contenedor de IoC): su intención es disminuir el acoplamiento entre componentes del software, evitando la instanciación directa de clases y facilitando la reutilización y la integración, localizando las interfaces o servicios que estos brindan a través de contratos bien definidos.

Modelo Vista Controlador (MVC): separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

WebSSO (Web Single Sign On): la intención principal de emplear este patrón es solucionar que los usuarios de cualquier organización solo tengan que autenticarse una única vez para acceder a los diferentes servicios de la misma evitándoles memorizar varias cuentas de usuarios y contraseñas.

Patrones Grasp: nos indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Singleton: se asegura de que una clase tiene una sola instancia y proporcionar un punto de acceso global a ella.

Arquitectura por capas: tiene como objetivo principal la separación de las partes que componen el sistema es decir separa la lógica de negocios, la capa de presentación y la capa de datos. De esta forma permite crear diferentes interfaces sobre el sistema sin tener que cambiar la capa de datos o lógica.

Escape de las salidas: evita que, caracteres o información de control presentes en los datos, tengan un significado especial para el intérprete hacia el que van dirigidos.

Punto único acceso: define un punto único de acceso hacia la aplicación donde sea posible aplicar diferentes medidas de seguridad con el objetivo de ejercer un mejor control de las comunicaciones hacia y desde la misma.

1.7 Conclusiones del capítulo.

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

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

2 Capítulo 2. Análisis, Diseño y Desarrollo de la Solución Propuesta.

2.1 Introducción.

El desarrollo de un sistema informático requiere una gran labor de análisis y diseño para lograr un producto final con la calidad requerida y que a su vez cumpla con las expectativas del cliente. Este proceso tiene como finalidad transformar los requisitos en un diseño del sistema en creación y adaptar el diseño para que se ajuste al entorno de implementación, pensado para el rendimiento.

En el presente capítulo se hará una descripción general de las características del sistema a realizar en correspondencia con la metodología de desarrollo de software expuesta en el capítulo anterior y con la ayuda de los procedimientos, técnicas y herramientas que esta nos brinda.

2.2 Descripción de la solución

Se propone el desarrollo de una aplicación web que les permita a los trabajadores expertos del Ministerio de Ciencia, Tecnología y Medio Ambiente (CITMA), Centro de Servicios Ambientales (CSAM) u otro organismo de esta rama almacenar y gestionar mejor la información de la gestión ambiental y general de playas, su estado ecológico ambiental, su gestión y monitoreo. Además, de almacenarán todos los parámetros de una playa con su fecha correspondiente y partir de esto se podrán hacer predicciones de su comportamiento en el futuro. De esta manera facilitará el acceso a la información, la seguridad y eficiencia de los datos.

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

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

Súper Gestor: Podrá crear una zona y asignarle a esta un usuario Gestor y además hacer reportes y eliminar una zona si está vacía.

Gestor: Es quien podrá crear, editar la última entrada y eliminar las playas dentro de la zona que se le asigne en caso de que ésta se encuentre vacía.

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

2.3 Requisitos No Funcionales.

1. El módulo del sistema debe ajustarse a los colores y diseño del sistema en general.
2. El módulo del sistema de contener colores acordes al diseño original del sistema.

3. El módulo del sistema debe contener el logotipo del Observatorio de Costa Atenas.
4. Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
5. El sistema debe ser capaz de operar adecuadamente con hasta 10.000 usuarios con sesiones concurrentes.
6. Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.
7. Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador general del sistema.
8. El nuevo sistema debe desarrollarse aplicando patrones y recomendaciones de programación que incrementen la seguridad de datos.
9. El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
10. El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
11. La aplicación web debe poseer un diseño "Responsive" a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
12. El sistema debe poseer interfaces gráficas bien formadas.
13. El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.
14. El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.
15. La tasa de tiempos de falla del sistema no podrá ser mayor al 0,5% del tiempo de operación total.
16. La aplicación debe ser compatible con todas las versiones de navegadores web.
17. La aplicación deberá consumir menos de 500 Mb de memoria RAM al ser operada por un navegador.
18. El sistema deberá ser multiplataforma y ser compatible con los sistemas operativos Windows y Linux. Deberá funcionar con los siguientes requisitos mínimos de software: Sistema operativo: Windows XP o superior. Linux. Gestor de base de datos: Postgresql v8.0 o superior.
19. El sistema deberá proteger la información que se maneje, de acceso no autorizado y divulgación, a partir de los diferentes roles de los usuarios que empleen el sistema, es decir tener un control de usuarios.
20. El sistema deberá encontrarse disponible las 24 horas de todos los días para aquellos usuarios autorizados a acceder al sistema.

2.4 Etapa de Planificación

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

2.4.1 Pila del Producto (SPRINT BACKLOG)

La pila de producto es el **corazón de Scrum**. Es donde empieza todo. La Pila de Producto es, básicamente, una lista priorizada de requisitos, o historias, o funcionalidades, o lo que sea. Cosas que el cliente quiere, descritas usando la terminología del cliente. Llamamos a esto historias, o a veces simplemente elementos de la Pila.

Explicación de cada uno de los aspectos de la tabla:

Código: Muestra un código elegido por el equipo de trabajo para identificar cada pila del producto.

Prioridad: Muestra en una escala de “baja, media y alta” la prioridad del desarrollo de la pila del producto por el equipo de desarrollo.

Descripción: Muestra los requisitos o funcionalidades para la realización del producto.

Estimado(hrs): Muestra el estimado en horas que necesita el equipo de trabajo para realizar cada requisito de la pila de producto.

Debido a que la Pila de Producto es la única fuente de requisitos y funcionalidades para la realización del producto; que está en constante evolución y es un artefacto vivo y dinámico; ha pasado por un proceso de cambios constantes para identificar lo que el producto necesita para ser adecuado, competitivo y útil. A continuación se presenta la Pila de Producto final, elaborada y ordenada por el Dueño de Producto la que atiende a las necesidades de dicho producto.

2. Tabla 2: Pila de Producto (PRODUCT BACKLOG)

PILA DE PRODUCTO			
Código	Prioridad	Descripción	Estimado (hrs)
p01	Alta	Investigar la plataforma tecnológica	24

p02	Alta	Analizar y diseñar de la Base de Datos	120
p03	Alta	Diseñar de la Interfaz de Usuario	56
p04	Media	Crear la vista general estática	24
p05	Media	Crear la vista de la autenticación	24
p06	Media	Crear la vista de caracterización	30
p07	Media	Crear la vista de Estado-Ecológico Ambiental	24
p08	Media	Crear la vista de Análisis de la Gestión	24
p09	Media	Crear Vista de Monitoreo	35
p10	Alta	Implementar sistema de autenticación	72
p11	Media	Gestionar Usuarios	48
p12	Medio	Gestionar Roles o grupos de usuarios	16
p13	Alta	Gestionar Playas	56
p14	Media	Gestionar Aspectos Morfológicos	48
p15	Media	Gestionar Sedimentología	48
p16	Media	Gestionar Vegetación y Fauna	48
p17	Media	Gestionar Calidad del Agua	48
p18	Media	Gestionar Antropogénesis	48
p19	Media	Gestionar Zonas	16
p20	Media	Gestionar Estado Ecológico-Ambiental	50
p21	Alta	Gestionar Análisis de la Gestión	12
p22	Alta	Gestionar Monitoreo	60
p23	Alta	Crear Mecanismo de Predicciones	30
p24	Alta	Crear Mecanismo de Seguridad	15
p25	Alta	Crear el Mecanismo del Reporte	12
p26	Baja	Crear Mecanismo de Búsqueda	10

2.4.2 Pila de Sprint (SPRINT BACKLOG)

A partir de la Pila de Producto anteriormente elaborada se crea la Pila de Sprint que descompone las funcionalidades de la Pila de Producto en las tareas necesarias para construir un incremento: una parte completa y operativa del producto. Además también se planificarán de forma independiente y detallada cada sprint, descompone el trabajo

en unidades de tamaño adecuado para monitorear el avance a diario, e identificar riesgos y problemas sin necesidad de procesos de gestión complejos.

Son los elementos de la pila del producto que pueden ser incorporados a un sprint se denominan “**preparados**” o “**accionables**” y son los que pueden seleccionarse en la reunión de planificación del sprint.

3. Tabla 3: Pila de Sprint (SPRINT BACKLOG)

PILA DE SPRINT					
Sprint	Pila de Producto	Encargado	Fecha Inicial	Fecha Final	Valor
1	Investigar la plataforma tecnológica	Yaisel González Hernández Eduardo J Berrio Turiño	26/08/2022	21/09/2022 (26 días)	200
	Analizar y diseñar la base de Datos				
	Diseñar la Interfaz de usuario				
2	Crear la vista general estática	Yaisel González Hernández	22/09/2022	04/10/2022 (12 días)	161 (Sumado: 361)
	Crear la vista de la autenticación				
	Crear la vista de caracterización				
	Crear la vista de Estado-Ecológico Ambiental				
	Crear la vista de Análisis de la Gestión				
Crear Vista de Monitoreo					
3	Implementar sistema de autenticación	Yaisel González Hernández	05/10/2022	14/10/2022 (9 días)	136 (Sumado: 497)
	Gestionar Usuarios				

	Gestionar Roles o grupos de usuarios	Eduardo J Berrio Turiño			
4	Gestionar Playas	Yaisel González Hernández	15/10/2022	07/11/2022 (23 días)	296 (Sumado: 793)
	Gestionar Aspectos Morfológicos				
	Gestionar Sedimentología				
	Gestionar Vegetación y Fauna				
	Gestionar Calidad del Agua				
	Gestionar Antropogénesis				
5	Gestionar Zonas	Yaisel González Hernández	08/11/2022	19/11/2022 (11 días)	138 (Sumado: 931)
	Gestionar Estado Ecológico-Ambiental				
	Gestionar Análisis de la Gestión				
	Gestionar Monitoreo				
6	Crear Mecanismo de Predicciones	Yaisel González Hernández	20/11/2022	26/11/2022 (6 días)	67 (Sumado: 998)
	Crear Mecanismo de Seguridad				
	Crear el Mecanismo del Reporte				
	Crear Mecanismo de Búsqueda				

2.4.3 Planificación de Sprint Del Proyecto

4. Tabla 4: Pila de Sprint (SPRINT BACKLOG)

SPRINT 1

Pila del Producto	Tareas	Encargado	Est. Inicial	Est. Total
Investigar la plataforma tecnológica	<ul style="list-style-type: none"> - Definir las características mínimas del hardware - Seleccionar el gestor de base de datos - Seleccionar los lenguajes de programación y herramientas de desarrollo - Realizar pruebas rendimiento al hardware instalado - Realizar pruebas de rendimiento al gestor de base de datos 	Yaisel González Hernández	20	200
	<ul style="list-style-type: none"> - Comprobar que los lenguajes de programación y las herramientas seleccionadas cumplen las exigencias del proyecto 	Yaisel González Hernández	4	
Analizar y diseñar de la Base de Datos	<ul style="list-style-type: none"> - Analizar y definir los tipos de datos - Normalizar la base de datos - Crear las clases de la base de datos - Generar la base de datos 	Yaisel González Hernández	110	
	<ul style="list-style-type: none"> - Cargar datos de prueba 	Yaisel González Hernández	8	
Diseñar de la Interface de Usuario	<ul style="list-style-type: none"> - Seleccionar colores del diseño - Diseñar la interface de usuario - Realizar la maquetación del diseño - Crear la lógica de JavaScript del diseño 	Yaisel González Hernández	48	

	- Comprobar la funcionalidad y adaptabilidad del diseño en los diferentes dispositivos y navegadores	Yaisel González Hernández	8	
--	--	---------------------------------	---	--

5. Tabla 5: Planificación del SPRINT 2

SPRINT 2				
Pila del Producto	Tareas	Encargado	Est. Inicial	Est. Total
Crear la vista general estática	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Generar el menú de negocios	Yaisel González Hernández	20	161
	- Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Yaisel González Hernández	4	
Crear la vista de la autenticación	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Crear la lógica de python para la comunicación	Yaisel González Hernández	20	
	- Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño. - Comprobar nombre de usuario y contraseña sean correcto y dar acceso al sistema según el rol	Yaisel González Hernández	4	

Crear la vista de caracterización	<ul style="list-style-type: none"> - Crear y maquetar la vista general - Crear y maquetar la vista detalles - Crear la lógica de JavaScript para las vistas - Recuperar la información de la base de datos y mostrarla en la vista correspondiente - Generar el menú de los aspectos de caracterización 	Yaisel González Hernández	25	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño 	Yaisel González Hernández	5	
Crear la vista de Estado-Ecológico Ambiental	<ul style="list-style-type: none"> - Crear y maquetar la vista - Crear la lógica de JavaScript para las vistas - Recuperar la información de la base de datos y mostrarla en la vista - Generar el menú de ofertas - Generar el motor de búsqueda 	Yaisel González Hernández	20	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño 	Yaisel González Hernández	4	
Crear la vista de Análisis de la Gestión	<ul style="list-style-type: none"> - Crear y maquetar la vista - Crear la lógica de JavaScript para la vista - Crear el motor de búsqueda - Recuperar la información de la base de datos según criterio de búsqueda y mostrarla en la vista 	Yaisel González Hernández	16	

	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño 		4	
Crear Vista de Monitoreo	<ul style="list-style-type: none"> - Crear y maquetar la vista - Generar el formulario de contacto - Crear lógica de envío de información - Crear la lógica de JavaScript para la validación del formulario 	Yaisel González Hernández	22	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario 		8	

6. Tabla 6: Planificación del SPRINT 3

SPRINT 3				
Pila del Producto	Tareas	Encargado	Est. Inicial	Est. Total
Implementar sistema de autenticación	<ul style="list-style-type: none"> - Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Generar los formularios de la vista 	Yaisel González Hernández	45	136

	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño -Comprobar la validación de los datos del formulario 	Yaisel González Hernández	15	
Gestionar Usuarios	<ul style="list-style-type: none"> - Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista -Crear la lógica de python para la comunicación Crear lógica de cambiar información del usuario - Crear lógica de cambiar imagen del perfil - Crear lógica de cambiar contraseña - Generar los formularios de la vista - Crear la lógica de Django de validación de formulario 	Yaisel González Hernández	40	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario 	Yaisel González Hernández	8	
Gestionar Roles o grupos de usuarios	<ul style="list-style-type: none"> - Crear y maquetar la vista general - Crear y maquetar la vista detalles 	Yaisel González Hernández	10	

	<ul style="list-style-type: none"> - Recuperar la información de la base de datos y mostrarla en la vista correspondiente - Crear mecanismo de asignación de roles 			
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño 	Yaisel González Hernández	5	

7. Tabla 7: Planificación del SPRINT 4

SPRINT 4				
Pila del Producto	Tareas	Encargado	Est. Inicial	Est. Total
Gestionar Playas	<ul style="list-style-type: none"> - Crear y maquetar la vista de cambiar información de la empresa, imágenes de la interface y datos de la oferta - Crear lógica de cambiar información de la empresa - Crear lógica de cambiar imágenes de la interface - Crear lógica de cambiar datos de la oferta - Generar los formularios de la vista 	Yaisel González Hernández	40	296
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario 	Yaisel González Hernández	6	

Gestionar Aspectos Morfológicos	<ul style="list-style-type: none"> - Crear y maquetar las vistas de agregar, editar y eliminar aspectos - Crear lógica de agregar aspectos - Crear lógica de editar aspectos - Crear lógica de eliminar aspectos - Generar los formularios de las correspondientes vistas 	Yaisel González Hernández	40	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño. - Comprobar la validación de los datos del formulario 	Yaisel González Hernández	8	
Gestionar Sedimentología	<ul style="list-style-type: none"> - Crear y maquetar la vista general - Crear y maquetar la vista detalles - Recuperar la información de la base de datos y mostrarla en la vista correspondiente. - Crear lógica de agregar sedimentología - Crear lógica de editar sedimentología - Crear lógica de eliminar sedimentología 	Yaisel González Hernández	40	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño 	Yaisel González Hernández	8	

	- Comprobar la validación de los datos del formulario			
Gestionar Vegetación y Fauna	<ul style="list-style-type: none"> - Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Crear y maquetar las vistas de agregar, editar y eliminar vegetación y fauna - Crear lógica de agregar vegetación y fauna - Crear lógica de editar vegetación y fauna - Crear lógica de eliminar vegetación y fauna 	Yaisel González Hernández	35	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario 	Yaisel González Hernández	10	
Gestionar Calidad del Agua	<ul style="list-style-type: none"> - Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Crear y maquetar las vistas de agregar, editar y eliminar calidad del agua - Crear lógica de agregar calidad del agua - Crear lógica de editar calidad del agua 	Yaisel González Hernández	35	

	- Crear lógica de eliminar calidad del agua			
	- Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario		10	
Gestionar Antropogénesis	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Crear y maquetar las vistas de agregar, editar y eliminar Antropogénesis - Crear lógica de agregar Antropogénesis - Crear lógica de editar Antropogénesis - Crear lógica de eliminar Antropogénesis	Yaisel González Hernández	40	
	- Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario	Yaisel González Hernández	8	

8. Tabla 8: Planificación del SPRINT 5

SPRINT 5				
Pila del Producto	Tareas	Encargado	Est. Inicial	Est. Total
Gestionar Zonas	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Generar los formularios de la vista	Yaisel González Hernández	10	138
	- Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño -Comprobar la validación de los datos del formulario	Yaisel González Hernández	6	
Gestionar Estado Ecológico-Ambiental	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Generar los formularios de la vista	Yaisel González Hernández	40	
	- Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño -Comprobar la validación de los datos del formulario	Yaisel González Hernández	8	
Gestionar Análisis de la Gestión	- Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista -Crear la lógica de python para la comunicación	Yaisel González Hernández	10	

	<p>Crear lógica de cambiar información del usuario</p> <ul style="list-style-type: none"> - Crear lógica de cambiar imagen del perfil - Crear lógica de cambiar contraseña - Generar los formularios de la vista - Crear la lógica de Django de validación de formulario 			
	<ul style="list-style-type: none"> - Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario 	Yaisel González Hernández	2	
Gestionar Monitoreos	<ul style="list-style-type: none"> - Crear y maquetar la vista general - Crear y maquetar la vista detalles - Recuperar la información de la base de datos y mostrarla en la vista correspondiente - Crear mecanismo de asignación de roles 	Yaisel González Hernández	45	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionabilidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño 	Yaisel González Hernández	10	

9. Tabla 9: Planificación del SPRINT 6

SPRINT 6				
Pila del Producto	Tareas	Encargado	Est. Inicial	Est. Total

Crear Mecanismo de Predicciones	<ul style="list-style-type: none"> - Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Generar los formularios de la vista 	Yaisel González Hernández	25	67
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño 	Yaisel González Hernández	5	
Crear Mecanismo de Seguridad	<ul style="list-style-type: none"> - Crear y maquetar la vista - Recuperar la información de la base de datos y mostrarla en la vista - Crear la lógica de django para la comunicación 	Yaisel González Hernández	10	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño - Comprobar la validación de los datos del formulario 	Yaisel González Hernández	5	
Crear el Mecanismo del Reporte	<ul style="list-style-type: none"> - Crear y maquetar la vista general - Crear y maquetar la vista detalles - Recuperar la información de la base de datos y mostrarla en la vista correspondiente - Crear mecanismo de asignación de roles 	Yaisel González Hernández	10	
	<ul style="list-style-type: none"> - Comprobar la correcta funcionalidad de la vista 	Yaisel González Hernández	2	

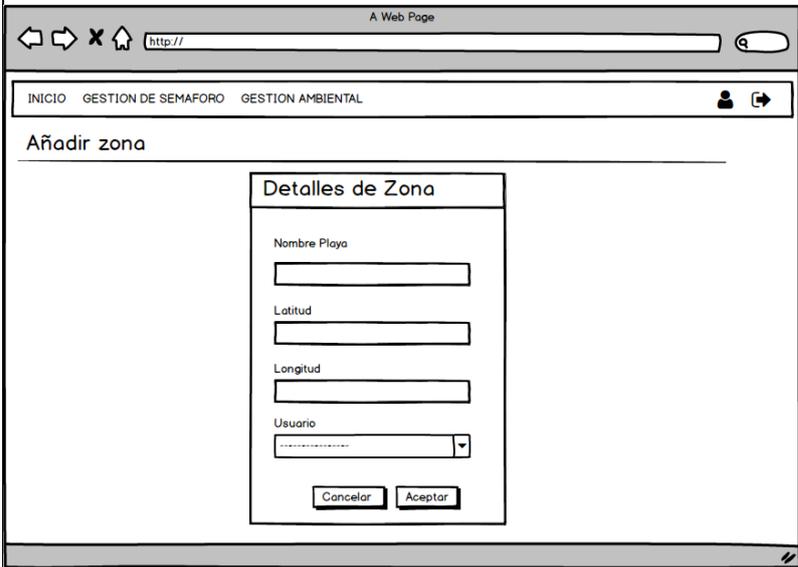
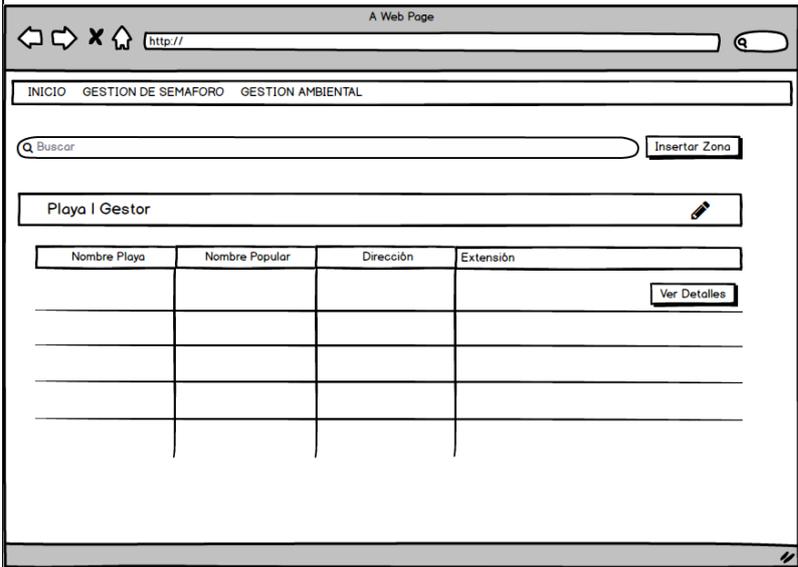
	- Comprobar que la información sea correcta y con el formato adecuado según el diseño		
Crear Mecanismo de Búsqueda	- Crear y maquetar la vista general - Crear y maquetar la vista detalles - Recuperar la información de la base de datos y mostrarla en la vista correspondiente - Crear mecanismo de asignación de roles	Yaisel González Hernández	8
	- Comprobar la correcta funcionalidad de la vista - Comprobar que la información sea correcta y con el formato adecuado según el diseño	Yaisel González Hernández	2

2.5 Historia de Usuarios.

10. Tabla 10: Historia de Usuario 1

HISTORIA DE USUARIO	
Código:hu01	Nombre: Gestionar Zonas
Tipo de HU: Funcional	Complejidad: Alta
Actor: Súper Gestor	HU Relacionadas: No
Descripción: El usuario podrá insertar una nueva zona y asignarle un usuario a esta , modificar la última entrada y eliminarla sino tiene datos guardados	

Prototipo



Resultado:

Se crea una nueva zona playera con los datos suministrados por el usuario.

11. Tabla 11: Historia de Usuario 2

HISTORIA DE USUARIO	
Código:hu02	Nombre: Gestionar Playa
Tipo de HU: Funcional	Complejidad: Alta
Actor: Gestor	HU Relacionadas: hu01

Descripción:

El usuario podrá insertar una nueva playa en la zona que se le fue asignada, modificar la última entrada y eliminarla sino tiene datos guardados

Prototipo

A Web Page

INICIO GESTION DE SEMAFORO GESTION AMBIENTAL

Playa

Nombre Playa	Nombre Popular	Dirección	Extensión

Resultado:

Se crea una nueva playa con los datos suministrados por el usuario para comenzar el proceso de caracterización de la misma.

A Web Page

INICIO GESTION DE SEMAFORO GESTION AMBIENTAL

Añadir Playa

Detalles generales

Nombre Playa

Nombre de Playas Populares

Dirección de la Playa

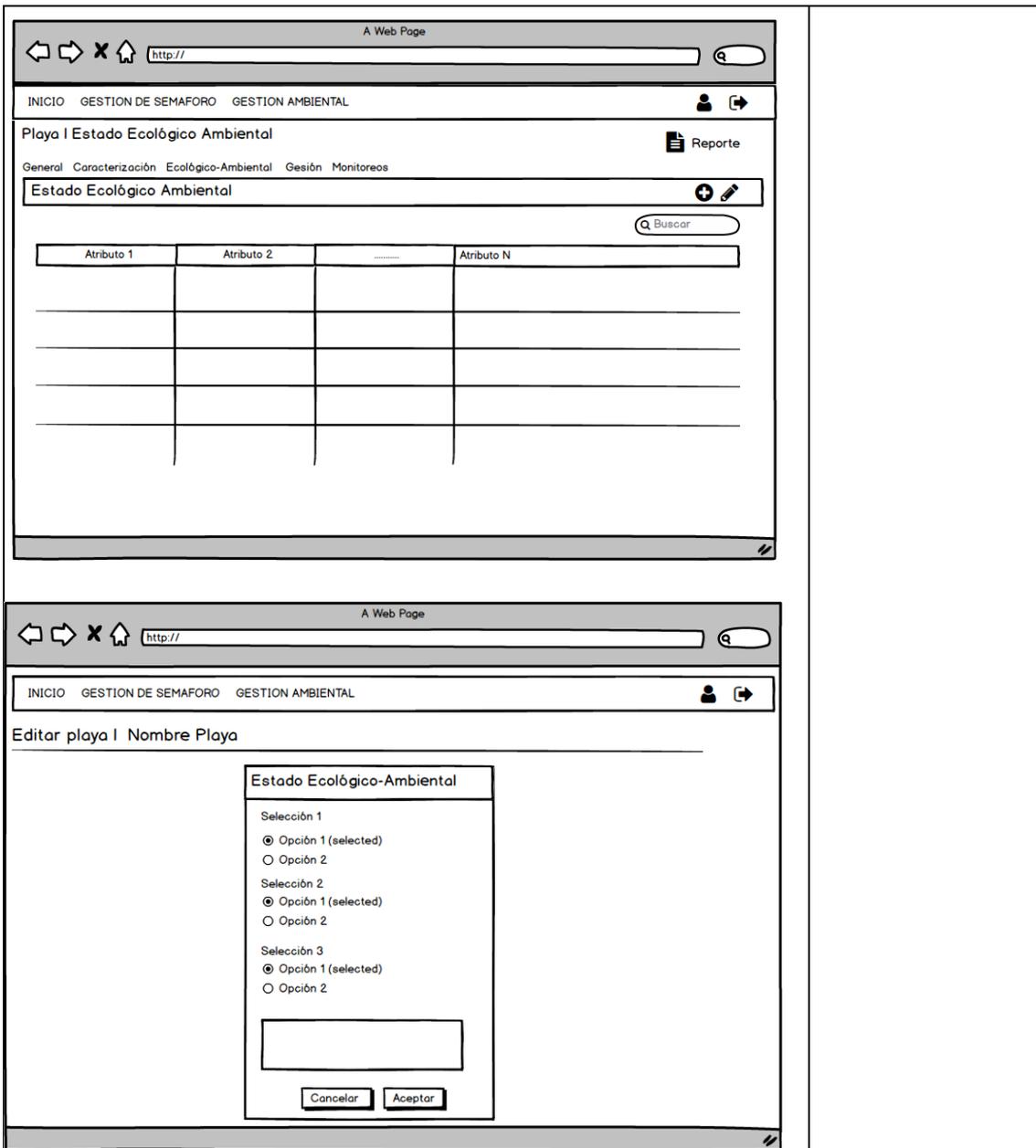
Extensión de la Playa

Latitud

Longitud

12. Tabla 12: Historia de Usuario 3

HISTORIA DE USUARIO	
Código:hu03	Nombre: Gestionar Estado Ecológico-Ambiental
Tipo de HU: Funcional	Complejidad: Alta
Actor: Gestor	HU Relacionadas: hu02
Descripción: El usuario podrá insertar un nuevo estado ecológico-ambiental , modificar la última entrada y eliminarla sino tiene datos guardados	
Prototipo	Resultado: Se crea un nuevo estado ecológico ambiental con los datos suministrados por el usuario para su caracterización.



2.6 Modelado

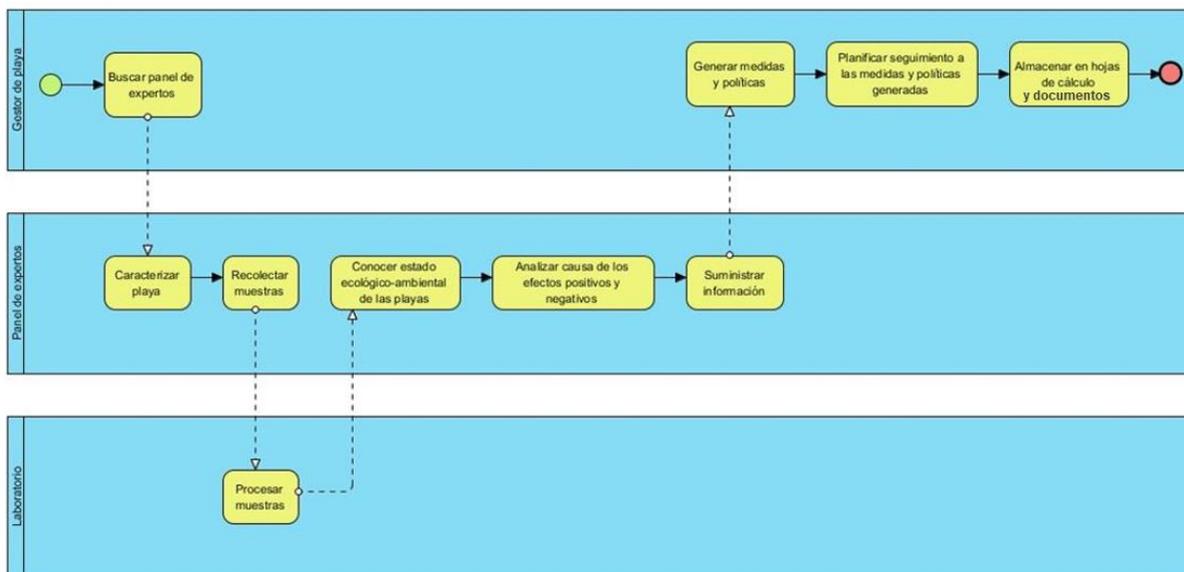
2.6.1 Diseño de la base de datos.

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

en hojas de cálculo y documentos, lo que le da poca integridad y seguridad a los datos, los cuales no están siempre disponibles, es decir inaccesible y en muchas ocasiones no existen por pérdidas de documentos o roturas de equipos.

2.7.1 BPMN (Modelo de Proceso de Negocio y Notación).

Figura 2. Modelo BPMN



2.8 Seguridad del módulo.

13. Tabla #13: Caso de uso gestionar usuario (Seguridad del Módulo)

UC-0001	Gestionar usuarios	
Versión	1.0 (23/11/2022)	
Autor	Yaisel González Hernández	
Fuentes	Alfredo Cabrera Hernández y Eduardo Berrio Turiño	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el súper usuario desea adicionar, modificar, eliminar o buscar un usuario.	
Precondición	Si se desea adicionar un usuario debe existir un rol. Si se desea modificar, eliminar o buscar un usuario debe existir un usuario previamente creado.	
Secuencia normal	Paso	Acción
	1	El actor Súper Usuario (ACT-0001) ve los usuarios existentes
	2	Si se desea adicionar usuario, el actor Súper Usuario (ACT-0001) adiciona usuario
	3	Si se desea asignar un rol a un usuario, se realiza el caso de uso Gestionar roles (UC-0002)
	4	Si se desea editar un usuario, el actor Súper Usuario (ACT-0001) modifica un usuario existente
5	Si se desea eliminar un usuario, el actor Súper Usuario (ACT-0001) elimina un usuario previamente existente	

Postcondición	Se ha adicionado, modificado, eliminado o mostrado un usuario.	
Excepciones	Paso	Acción
	1	Si no existe un usuario previamente creado, el actor Súper Usuario (ACT-0001) ve una lista vacía, a continuación este caso de uso continúa
	2	Si no existe un rol previamente adicionado, el actor Súper Usuario (ACT-0001) no puede adicionar el usuario, a continuación este caso de uso continúa
	3	Si no existe un rol previamente creado, el actor Súper Usuario (ACT-0001) no puede asignar un rol a un usuario, a continuación este caso de uso continúa
	4	Si no existe un usuario previamente creado, el actor Súper Usuario (ACT-0001) no puede modificar un usuario, a continuación este caso de uso continúa
	5	Si no existe un usuario previamente creado, el actor Súper Usuario (ACT-0001) no puede eliminar un usuario, a continuación este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	PD	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

14. Tabla #14: Caso de uso gestionar roles (Seguridad del Módulo)

UC-0001	Gestionar roles	
Versión	1.0 (23/11/2022)	
Autor	Yaisel González Hernández	
Fuentes	Alfredo Cabrera Hernández y Eduardo Berrio Turiño	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el súper usuario desea adicionar, modificar, eliminar o ver un rol o durante la realización de los siguientes casos de uso: [UC-0001] Gestionar usuarios	
Precondición	Si se desea adicionar un rol debe existir un súper usuario. Si se desea ver, modificar o eliminar un rol debe existir un rol previamente creado.	
Secuencia normal	Paso	Acción
	1	El actor Súper Usuario (ACT-0001) puede ver una lista de roles
	2	Si se desea añadir un nuevo rol, el actor Súper Usuario (ACT-0001) adiciona el rol
	3	Si se desea editar un rol, el actor Súper Usuario (ACT-0001) modifica un rol
	4	Si se desea eliminar un rol, el actor Súper Usuario (ACT-0001) elimina el rol
Postcondición	Se ha mostrado, adicionado, modificado o eliminado un rol.	
Excepciones	Paso	Acción

	1	Si no existe un rol creado, el actor Súper Usuario (ACT-0001) no puede ver la lista de roles, a continuación este caso de uso continúa
	2	Si no existe súper usuario, el sistema no permite la gestión de roles, a continuación este caso de uso queda sin efecto
	3	Si no existe un rol creado, el actor Súper Usuario (ACT-0001) no puede modificar un rol, a continuación este caso de uso continúa
	4	Si no existe un rol previamente creado, el actor Súper Usuario (ACT-0001) no puede eliminar un rol, a continuación este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	PD	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

15. Tabla #15: Caso de uso autenticarse (Seguridad del Módulo)

UC-0001	Autenticarse	
Versión	1.0 (23/11/2022)	
Autor	Yaisel González Hernández	
Fuentes	Alfredo Cabrera Hernández y Eduardo Berrio Turiño	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando algún usuario del sistema desea autenticarse	
Precondición	Para poder llevar a cabo la autenticación, el sistema debe tener al menos un súper usuario	
Secuencia normal	Paso	Acción
	1	Si algún usuario desea autenticarse, el sistema comprueba sus datos de inicio de sesión y lo autentica
Postcondición	El usuario queda autenticado.	
Excepciones	Paso	Acción
	1	Si los datos de inicio de sesión son incorrectos, el sistema no autentica al usuario, a continuación este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	PD	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

16. Tabla #16: Caso de uso cerrar sesión (Seguridad del Módulo)

UC-0001	Cerrar Sesión	
Versión	1.0 (23/11/2022)	
Autor	Yaisel González Hernández	
Fuentes	Alfredo Cabrera Hernández y Eduardo Berrio Turiño	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desea cerrar sesión	
Precondición	El usuario inició sesión previamente	
Secuencia normal	Paso	Acción
	1	Si el usuario desea cerrar sesión, el sistema cierra su sesión y lo redirige a la pantalla inicial
Postcondición	El usuario salió de la sesión, por lo que ahora es un usuario general más.	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	PD	
Importancia	vital	
Urgencia	PD	
Estado	validado	
Estabilidad	alta	
Comentarios	Ninguno	

Seguridad del módulo a partir de las vistas e interacción con los usuarios que acceden al sistema:

Súper Usuario (Administrador Total): Puede gestionar usuarios, crearlos, modificarlos y eliminarlos. Además puede asignarles uno o varios roles (grupos) (Súper Gestor o Gestor). Tiene autoridad para añadir, modificar y eliminar los nomencladores y los modelados del sistema, puede realizar la gestión de las zonas, playas y todas sus características, así como es el único que puede modificarlos y eliminarlos sin ninguna restricción aunque estén llenos.

Súper Gestor: Puede gestionar zonas (añadir, modificar, eliminar) en caso de no estar vacías. Puede ver las playas y todos sus detalles, generar reportes, pero no puede modificar nada. No puede gestionar usuarios, pero es autorizado asignarle un gestor a una zona.

Gestor: Puede ver las zonas asignadas a él y sus playas. Además puede gestionar playas, añadir, eliminar (si está vacía) y modificar playas en las zonas que le

corresponda. Puede añadir los parámetros de caracterización y detalles de la playa (morfología, calidad del agua, sedimentología, antropogénesis, vegetación y fauna, estado ecológico ambiental, gestión de los aspectos positivos y negativos, monitoreos) y editar la última entrada de cada uno. No puede gestionar usuarios.

2.9 Tratamiento de errores.

Las excepciones son situaciones que pueden provocar fallos en la ejecución normal de un programa. El sistema propuesto presenta una interfaz diseñada, implementada y dirigida a evitar tales situaciones y errores, al verificar incongruencias en la entrada de datos y en el establecimiento de conexiones. Siempre que se detecten problemas se generan mensajes de error a través de ventanas de alerta.

Se validará que el usuario no deje campos vacíos o incurra en el error de repetir mal la contraseña, utilizando expresiones regulares, así el tiempo de respuesta es mucho menor, pues esta validación se realiza directamente del lado del cliente.

2.10 Análisis de factibilidad.

Factor de Ajuste	Puntaje
Comunicación de Datos	3
Proceso Distribuido	4
Objetivos de Rendimiento	2
Configuración de Explotación Compartida	2
Taza de Transacciones	3
Entrada de datos en línea	4
Eficiencia en el usuario final	5
Actualizaciones en línea	3
Lógica de Procesos Interno Compleja	2
Reusabilidad del Código	4
Conversión e Instalación Contempladas	1
Facilidad de Operación	4
Instalaciones múltiples	2
Facilidad de cambios	5

Componentes:

EI: Procesos en los que se introducen datos y que suponen la actualización de cualquier archivo interno.

EO: Procesos en los que se envía datos al exterior de la aplicación.

EQ: Procesos consistentes en la combinación de una entrada y una salida, en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.

ILF: Grupos de datos relacionados entre sí internos al sistema.

EIF: Grupos de datos que se mantienen externamente.

17. Tabla # 17: Factor de Ajuste

Componente	Tipo de Componente	Prioridad	Nivel de Complejidad	Puntos de Función
Seguridad y Autenticación	Entrada Externa	Alta	Baja	3
Gestionar Zona				
Adicionar zona	Entrada Externa	Media	Media	4
Modificar zona	Entrada Externa	Media	Media	4
Eliminar zona	Entrada Externa	Media	Baja	3
Buscar zona	Consulta Externa	Baja	Baja	3
Listar zona	Consulta Externa	Media	Baja	3
GENERAL				
Gestionar Playas				
Adicionar Playa	Entrada Externa	Media	Baja	3
Modificar Playa	Entrada Externa	Media	Baja	3
Eliminar Playa	Entrada Externa	Media	Baja	3
Buscar Playa	Consulta Externa	Baja	Baja	3
Listar Playa	Consulta Externa	Media	Baja	3
CARACTERIZACIÓN				
Gestionar Aspectos Morfológicos de la Playa				
Adicionar Aspectos M.	Entrada Externa	Media	Media	4
Modificar Aspectos M.	Entrada Externa	Media	Media	4
Eliminar Aspectos M.	Entrada Externa	Media	Baja	3
Buscar Aspectos M.	Consulta Externa	Baja	Baja	3
Listar Aspectos M.	Consulta Externa	Media	Baja	3
Gestionar Vegetación y Fauna				
Adicionar Vegetación y F.	Entrada Externa	Media	Media	4
Modificar Vegetación y F.	Entrada Externa	Media	Media	4
Eliminar Vegetación y F.	Entrada Externa	Media	Baja	3
Buscar Vegetación y F.	Consulta Externa	Baja	Baja	3
Listar Vegetación y F.	Consulta Externa	Media	Baja	3
Gestionar Calidad del Agua				
Adicionar Calidad del Agua	Entrada Externa	Media	Media	4
Modificar Calidad del Agua	Entrada Externa	Media	Media	4
Eliminar Calidad del Agua	Entrada Externa	Media	Baja	3
Buscar Calidad del Agua	Consulta Externa	Baja	Baja	3
Listar Calidad del Agua	Consulta Externa	Media	Baja	3
Gestionar ocupación y usos del sistema playa-dunas				
Adicionar ocupación y uso	Entrada Externa	Media	Baja	3
Modificar ocupación y uso	Entrada Externa	Media	Baja	3
Eliminar ocupación y uso	Entrada Externa	Media	Baja	3
Buscar ocupación y uso	Consulta Externa	Baja	Baja	3
Listar ocupación y uso	Consulta Externa	Media	Baja	3
GESTIONAR ESTADO ECOLÓGICO-AMBIENTAL				
Adicionar Estado Eco.-Amb.	Entrada Externa	Media	Media	4

Modificar Estado Eco.-Amb.	Entrada Externa	Media	Media	4
Eliminar Estado Eco.-Amb.	Entrada Externa	Media	Baja	3
Buscar Estado Eco.-Amb.	Consulta Externa	Baja	Baja	3
Listar Estado Eco.-Amb.	Consulta Externa	Media	Baja	3
Generar Informe	Salida Externa	Alta	Media	5
GESTIONAR MONITOREO				
Adicionar Monitoreo	Entrada Externa	Media	Media	4
Modificar Monitoreo	Entrada Externa	Media	Media	4
Eliminar Monitoreo	Entrada Externa	Media	Baja	3
Buscar Monitoreo	Consulta Externa	Baja	Baja	3
Listar Monitoreo	Consulta Externa	Media	Baja	3
Generar Informe del Monitoreo	Salida Externa	Alta	Media	5
GESTIONAR ANÁLISIS DE LA GESTIÓN DE PLAYA				
Adicionar Análisis de la Gestión de Playa	Entrada Externa	Media	Media	4
Modificar Análisis de la Gestión de Playa	Entrada Externa	Media	Media	4
Eliminar Análisis de la Gestión de Playa	Entrada Externa	Media	Baja	3
Listar Análisis de la Gestión de Playa	Consulta Externa	Media	Baja	3
Generar Informe Análisis de la Gestión de Playa	Salida Externa	Alta	Media	5
Generar Caracterización	Salida Externa	Alta	Media	5
Total Puntos de Función (TPF)				172

18. Tabla # 18: Tabla 1 de Componente para medición de proyectos

19. Tabla # 19: Tabla 2 de Componente para medición de proyectos

Componente	Baja	Media	Alto	Total
EI	$14*3=42$	$14*4=56$	$0*6=0$	98
EO	$0*4=0$	$4*5=20$	$0*7=0$	20
EQ	$17*3=18$	$0*4=0$	$0*6=0$	51
ILF	$0*7=0$	$4*10=40$	$0*15=0$	40
EIF	$0*5=0$	$0*7=0$	$0*10=0$	0
				209

Puntos de Función Ajustado (PFA)

$$PFA = 209 * (0.65 + [0.01] * 44)$$

$$PFA = 209 * (1.09)$$

$$PFA = 227.81$$

Costo Total de Módulo (CT)

$$CT = 700 * 2 * 3$$

$$CT = 4200$$

Líneas de código por Puntos de Función (PF)

$$PF = 227.81 * 20$$

$$PF = 4966.26$$

Esfuerzo horas/persona

$$E = PFA / (1/8 \text{ personas/horas})$$

$$E = 227.81 / 0.125$$

$$E = 1822.48$$

DURACIÓN DEL PROYECTO EN HORAS (DPH)

$$DPH = 1822.48 \text{ horas/persona} / 2 \text{ personas} = 911.24 \text{ horas por miembro}$$

$$\text{DURACIÓN EN MESES} = 911.24 \text{ horas} / 300 \text{ horas/mes} \approx 3 \text{ meses}$$

2.11 Conclusiones del capítulo.

En este capítulo se da una muestra de cómo fue el desarrollo del software mediante la ingeniería de este y se puede arribar a las siguientes conclusiones:

- Con las diferentes tablas se pudo describir el proceso, para un mejor entendimiento de las funcionalidades del mismo.
- También se estimó un costo aproximado, el cual pudo haberle costado a la empresa si se lo desarrolla un equipo profesional. Por lo que es más eficiente la manera en que se desarrolló.
- Los elementos tratados en este capítulo sirvieron para llegar a un acuerdo entre las partes interesadas en el diseño y la estructura de la aplicación a través de la implementación de las funcionalidades q permite la gestión ambiental de playa.

3 Capítulo 3. Validación de la Solución Propuesta.

3.1 Introducción

El desarrollo de todo sistema informático consta de varias etapas como se ha explicado anteriormente. En este capítulo se muestran los resultados de la última etapa de desarrollo en la cual se presentan los resultados de las pruebas aplicadas al software para evaluar su funcionamiento. El proceso de validación que en este capítulo se describe está elaborado con el objetivo de encontrar la mayor cantidad de errores posibles en el sistema para de esta forma erradicarlos. A partir de los errores y su posterior solución, la calidad de la herramienta a obtener es mucho mayor, y su despliegue definitivo más sólido y robusto ante errores de los usuarios.

A continuación, se realizan las pruebas al software que permiten comprobar la calidad de este producto, lo que constituye uno de los pasos más importantes en el diseño e implementación de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de mostrar un error no descubierto hasta entonces y con el fin de verificar la fiabilidad y calidad de la aplicación como un todo. Obviamente, la mejor forma de que las pruebas estén correctas es incluir al cliente en el diseño de las mismas.

3.2 Descripción de la Propuesta de Solución

Se desarrolló un sistema para la gestión de playas basándose en los requisitos propuestos por el cliente, el sistema cuenta con una vista diferente desde del administrador del framework Django en el cual se puede Gestionar los Usuarios (Agregar, Eliminar, Editar y asignarle un rol), aquí solamente puede acceder el súper administrador,

En una primera vista (para el rol de Súper Gestor) encontramos el botón llamado Insertar Zona, en el que se podrá agregar una zona playera, editarla una vez agregada y eliminarla si se encuentra vacía, además de asignarle un usuario para su gestión general de los sub-módulos. Una vez dentro vemos una vista del sistema que cuenta con cinco sub-módulos generales los cuales son detalles generales de la playa, caracterización, estado ecológico-ambiental, análisis de la gestión y sus efectos y monitoreos. El primer sub-módulo se va a observar los detalles generales de la playa de este se puede adicionar, modificar y eliminar dicha playa en caso de que no se encuentre vacía.

También tenemos el sub-módulo de caracterización en este podemos encontrar las distintas características como son aspectos morfológicos, sedimentología, vegetación y fauna, calidad del agua y antropogénesis, de estas descritas anteriormente se puede adicionar, eliminar, modificar y realizar búsquedas, en caso de que existan muchos

datos por cada tabla, pues se encuentra un registro que divide en secciones de diez elementos en cada vista.

El otro sub-módulo es el estado ecológico-ambiental, del cual se encuentran las mismas funcionalidades de adicionar, modificar, eliminar y buscar, en caso de que existan muchos datos por cada tabla, pues se encuentra un registro que divide en secciones de diez elementos en cada vista.

El sistema además cuenta con su cuarto sub-módulo que es el nombrado Gestión, en el cual se realizan análisis y sus efectos, este apartado cuenta con un botón de descargar en formato PDF el último elemento agregado, un botón de modificarlo, un botón de agregar nuevos análisis de gestión y un botón de despliegue que muestra el historial de datos agregados, una vez dentro del historial el personal con el rol adecuado podrá eliminarlo. Cuando le damos editar o agregar un nuevo elemento de dato, nos muestra una interfaz amigable de edición de textos muy parecida a Microsoft Word.

Por último, tenemos el quinto sub-módulo que es el de monitoreo en cual tenemos el botón de agregar, el de mostrar el historial de datos guardados, el botón de editar y un botón para mostrar las predicciones siempre y cuando existan más de cuatro datos históricos del parámetro a predecir, dentro de la sección de predecir encontramos un botón llamado Imprimir que genera un reporte en formato PDF de los gráficos predictivos. En caso de que existan muchos datos por cada tabla, pues se encuentra un registro que divide en secciones de diez elementos en cada vista.

En cada vista antes descrita podemos encontrar un botón en la parte superior derecha llamado Reporte, el que es general de las tres fases, excluyendo el Análisis de la Gestión y los gráficos de predicción que tienen su reporte de manera apartada.

Figura 3. Primera vista del sistema que muestra el administrador Django.

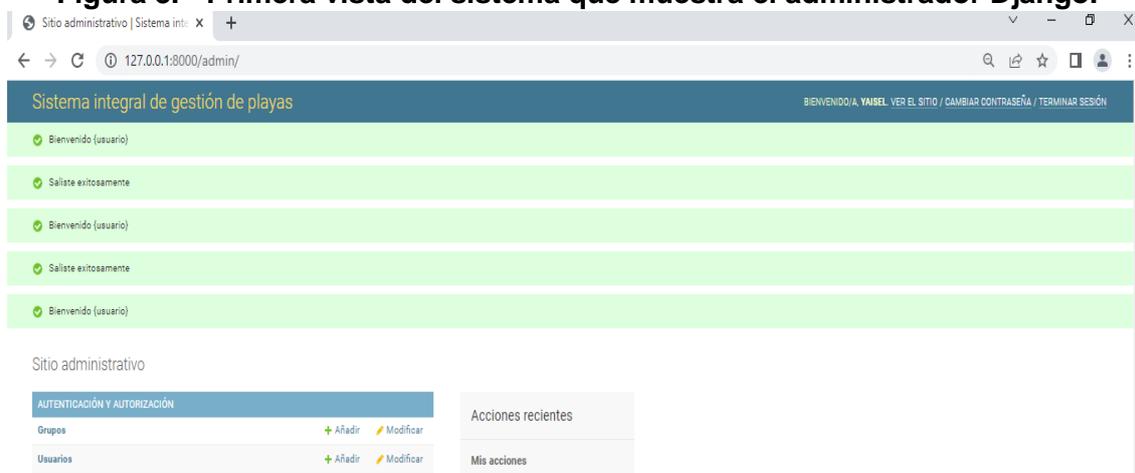


Figura 4. Segunda vista del sistema que muestra las zonas.

SEMPLA x +

127.0.0.1:8000/gestion-ambiental

INICIO GESTIÓN DE SEMÁFORO GESTIÓN AMBIENTAL yadian Salir

Buscar playa o zona INSERTAR ZONA

Varadero, Gestor: alfredo

Nombre de playa	Nombre popular de la playa	Dirección de la playa	Extensión de la playa
Calle 13	Calle 13	Avenida Playa Calle 13	89,98
Calle 54	Calle 54	Avenida Playa Calle 54	89,00

Pinar del Río, Gestor: eleidris

Nombre de playa	Nombre popular de la playa	Dirección de la playa	Extensión de la playa
SanBuco	SanBuco del Bravo	Pedregal entre San Luis y San Alberto, esquina 28	158,00
Arbol Colorado	La Arboleada	Carretera a San Luis ,kilometro 129 bajndo Loma	205,00

Figura 5. Tercera vista general del sistema que muestra uno de los elementos de la sección o sub-módulo caracterización.

SEMPLA x +

127.0.0.1:8000/playa/1/#

INICIO GESTIÓN DE SEMÁFORO GESTIÓN AMBIENTAL alfredo Salir

Calle 13 | Fase I - Caracterización

REPORTE

General Caracterización Ecológico-Ambiental Gestión Monitoreos

Aspectos morfológicos

Mostrar 10 registros Buscar:

Fecha de adición	Barreras arenosas (Cant)	Barreras arrecifales (Dist. de la costa)	Pastizales marinos (Dist. de la costa)	Tipo de Playa (Forma)	Tipo de Perfil de Playa	Tipo de fondo predominante (Parte sumergida)
12 de Octubre de 2022 a las 14:57	1	0,01 m	0,01 m	Lineal	Completo	Arenoso
17 de Octubre de 2022 a las 15:21	40	50,00 m	50,00 m	Lineal	Incompleto	Arenoso
8 de Noviembre de 2022 a las 12:32	5	6,00 m	12,00 m	Encajada	Incompleto	Arenoso

Mostrando registros del 1 al 3 de un total de 3 registros

Anterior 1 Siguiente

Figura 6. Cuarta vista general del sistema que muestra la sección o submódulo estado ecológico-ambiental.

The screenshot shows a web browser window with the URL 127.0.0.1:8000/playa/1/#. The navigation bar includes 'INICIO', 'GESTIÓN DE SEMÁFORO', and 'GESTIÓN AMBIENTAL'. The user 'alfredo' is logged in. The main header is 'Calle 13 | Fase II - Estado Ecológico-Ambiental'. Below it are tabs for 'General', 'Caracterización', 'Ecológico-Ambiental', 'Gestión', and 'Monitoreos'. The 'Estado Ecológico Ambiental' section is active, displaying a table of records.

Fecha de adición	Indicios de erosión por efecto del oleaje	Causas de la erosión	Tendencia erosiva	Procesos asociados al efecto eólico	Procesos asociados al drenaje pluvial	Afectaciones a la vegetación	Afectaciones a la fauna costera y marina	Indicios de acumulación
13 de Octubre de 2022 a las 15:46	Escarpes	Natural	Suave	Ver descripción	No hay descripción	No hay descripción	Ver descripción	Barra adosada
17 de Octubre de 2022 a las 20:48	Escarpes	Antrópica	Intensa	Ver descripción	Ver descripción	Ver descripción	Ver descripción	Existencia de más de una berna

Mostrando registros del 1 al 2 de un total de 2 registros. Botones: Anterior, 1, Siguiente.

Figura 7. Quinta vista general del sistema que muestra la sección o submódulo del análisis de la gestión.

The screenshot shows the same browser window with the URL 127.0.0.1:8000/playa/1/#. The navigation bar is the same. The main header is 'Calle 13 | Fase III - Análisis de la gestión y sus efectos'. Below it are tabs for 'General', 'Caracterización', 'Ecológico-Ambiental', 'Gestión', and 'Monitoreos'. The 'Análisis de la gestión' section is active, displaying a document titled 'Análisis de la gestión 25 de Noviembre de 2022 a las 18:58'.

Gestión Ambiental
Análisis y sus Efectos

Se denomina **gestión ambiental** o **gestión del medio ambiente** al conjunto de diligencias conducentes al manejo integral del sistema ambiental. Dicho de otro modo e incluyendo el concepto de desarrollo sostenible, es la estrategia mediante la cual se organizan las actividades humanas que afectan al **medio ambiente**, con el fin de lograr una adecuada calidad de vida, previniendo o mitigando problemas ambientales, potenciales o actuales.

La **gestión ambiental** responde al "cómo hay que hacer" para conseguir lo planteado por el **desarrollo sostenible**, es decir, para conseguir un equilibrio adecuado para el desarrollo económico, crecimiento de la población, uso racional de los recursos y protección y conservación del ambiente. Abarca un concepto integrador superior al del manejo ambiental: de esta forma no solo están las acciones a ejecutarse por la parte operativa, sino también las directrices, lineamientos y políticas formuladas desde los entes rectores, que terminan mediando la implementación.

Figura 8. Sexta vista general del sistema que muestra la sección o submódulo del monitoreo.

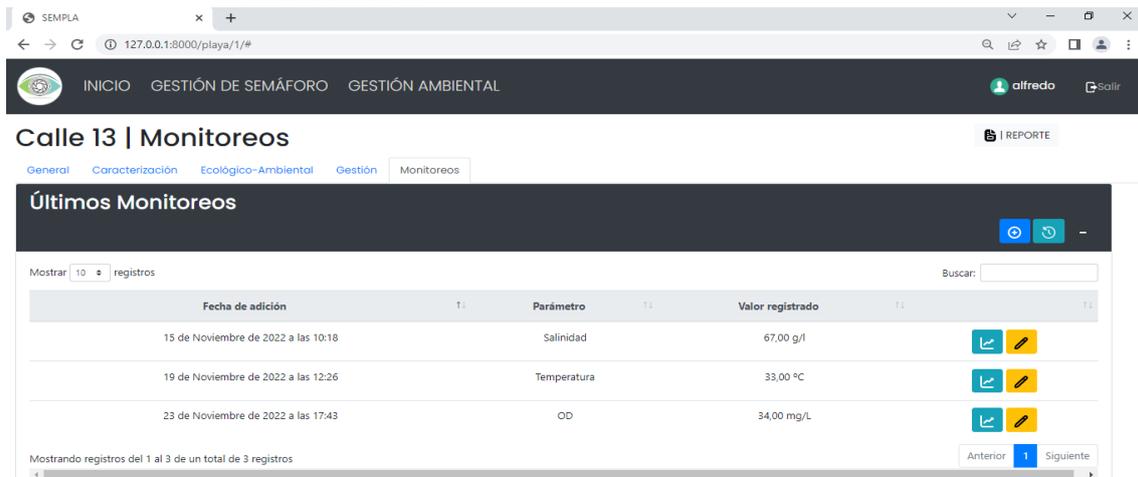


Figura 9. Séptima vista general del sistema que muestra la acción del botón de predicciones.



3.3 Pruebas al Software

La prueba de software es el **proceso de evaluación y verificación de un producto o aplicación de software para saber si hace lo que se supone que debe hacer**. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento. Las pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. A partir de la dependencia del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo. Existen distintos modelos de desarrollo de software, así como modelos de pruebas. A cada uno corresponde un nivel distinto de involucramiento en las actividades de desarrollo.

El proceso de pruebas es el instrumento más adecuado para determinar el status de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes

del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos o si es el software que se quería desarrollar. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba. (González Ramírez, 2017)

Una estrategia de pruebas integra los métodos de diseño de los casos de prueba para lograr un software eficaz. La prueba es un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática. Una estrategia de pruebas debe incluir tanto pruebas de alto como de bajo nivel. Son parte de la Verificación y Validación incluidas en el aseguramiento de la calidad del software.

Verificación: Comprobar que el software está de acuerdo con su especificación, donde se debe comprobar que satisface tanto los requerimientos funcionales como los no funcionales.

Validación: El objetivo es asegurar que el software satisface las expectativas del cliente.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.

3.3.1 Tipos de Pruebas de Software Generales

A continuación, se argumentarán algunas pruebas de software que se le aplicaron al sistema para una mejor aceptación por parte del cliente y para así evitar fallos en su funcionamiento operacional.

Las pruebas de software pueden ser de gran utilidad cuando se realizan en el momento adecuado, ya que mejoran la eficiencia de los productos que tienen en cuenta al usuario final. (IBM, 2022)

Prueba de Unidad: verifican que el componente funciona correctamente a partir del ingreso de diferentes casos de prueba. Los errores más comunes son detectados en estas pruebas.

- ✓ Se examinan las estructuras de datos locales
- ✓ Se prueban condiciones límite.
- ✓ Se ejercitan todos los caminos independientes. Se utiliza un controlador independiente para cada caso. Este es un programa que recibe las pruebas, las envía al módulo y muestra el resultado.

Las pruebas unitarias se centran en probar piezas/unidades individuales de una aplicación de software al principio del SDLC. Cualquier función, procedimiento, método

o módulo puede ser una unidad que se someta a pruebas unitarias para determinar su corrección y comportamiento esperado. Las pruebas unitarias son las primeras pruebas que los desarrolladores realizan durante la fase de desarrollo.

Prueba de Integración: Verifican que los componentes trabajan correctamente en forma conjunta.

- ✓ Se toman los componentes que han pasado las pruebas de unidad y se los combina.
- ✓ Estos test sirven ya que los datos podrían perderse en alguna interfaz.
- ✓ La combinación de los mismos podría traer efectos que no son los esperados.
- ✓ La integración puede ser:
 - Descendente: Inician por el programa principal. En profundidad: Integra todos los módulos de un camino de control principal de la estructura. En anchura: Incorpora todos los módulos directamente subordinados a cada nivel.
 - Ascendente: Se empieza la prueba con los módulos atómicos. Datos que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados siempre está disponible, pero no así los conductores.

Prueba de Sistema: verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total.

- ✓ Pruebas de recuperación: Se controla que el software se recupere ante fallas. Generalmente se fuerza el fallo.
- ✓ Pruebas de seguridad: Se comprueban los mecanismos de protección integrados.
- ✓ Pruebas de resistencia: Se diseñan para enfrentar a los programas a situaciones anormales.
- ✓ Pruebas de rendimiento: Se prueba el sistema en tiempo de ejecución. A veces va emparejada con la prueba de resistencia.

Prueba de Aceptación: Proporcionan una seguridad final de que el software satisface los requerimientos.

- ✓ Revisión de la configuración: Asegurar que todos los elementos de la configuración del software se hallan desarrollado apropiadamente.
- ✓ Pruebas de aceptación (ALFA y BETA): Las realiza el usuario final en lugar del responsable del desarrollo.
- Pruebas ALFA: Desarrolladores con clientes antes de liberar el producto.

- Pruebas BETA: Selecciona los clientes que efectuarán la prueba. El desarrollador no se encuentra presente.

Prueba de Seguridad: Verifica que el software, no presente ninguna vulnerabilidad

- ✓ Pruebas de Autenticación: Se realizaron pruebas de acceso de los distintos roles de usuarios para así permitir realizar las operaciones asignadas por rol sobre el sistema.

Pruebas de caja negra: Las pruebas de caja negra, conocidas también como black box testing, pueden definirse como una técnica donde se busca la verificación de las funcionalidades del software o aplicación analizada, sin tomar como referente la estructura del código interno, las rutas de tipo internas ni la información referente a la implementación. Esto quiere decir que la prueba se lleva a cabo con desconocimiento del funcionamiento del sistema interno, debido a que se enfoca en la entrada y salida de un software, toma como base sus especificaciones y requisitos.

De manera que se puede asegurar que el objetivo de las pruebas de caja negra está relacionado con la validación de los recursos funcionales del software o aplicación que se busca examinar.

3.3.2 Plan de Pruebas

Según (Loaiza, 2010) el plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento (Pérez, 2018) .Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y se asegura la calidad del producto que se está a punto de entregar al cliente. (González Ramírez, 2017)

En la siguiente tabla se resumen las pruebas que se realizan a la aplicación web:

20. Tabla #20: Plan de Pruebas (Pruebas al Software)

No.	Funcionalidades del sistema o Pila del Producto(EPP)	Pruebas a Realizar
1	Diseño y Creación de la base de datos.	Test de Base de Datos

2	Diseño de la interfaz de usuario.	Test diseño y adaptabilidad de la interfaz de usuario(UI)
3	Autenticarse	Test de Autenticación de Usuario.
4	Gestionar Usuarios	Test de Insertar Usuario. Test de Modificar Usuario. Test de Eliminar Usuario. Test de Listar Usuario. Test de asignar Rol
5	Gestionar Zonas	Test de Insertar Zona. Test de Modificar Zona. Test de Eliminar Zona. Test de Listar Zona. Test de Buscar Zona.
6	Gestionar Playa	Test de Insertar Playa. Test de Modificar Playa. Test de Eliminar Playa. Test de Listar Playa. Test de Buscar Playa.
7	Gestionar Aspectos Morfológicos	Test de Insertar Aspectos Morfológicos. Test de Modificar Aspectos Morfológicos. Test de Eliminar Aspectos Morfológicos. Test de Listar Aspectos Morfológicos. Test de Buscar Aspectos Morfológicos.
8	Gestionar Sedimentología	Test de Insertar Sedimentología. Test de Modificar Sedimentología. Test de Eliminar Sedimentología. Test de Listar Sedimentología. Test de Buscar Sedimentología.
9	Gestionar <u>Vegetación y Fauna</u>	Test de Insertar <u>Vegetación y Fauna</u> .

		<p>Test de Modificar <u>Vegetación y Fauna</u>.</p> <p>Test de Eliminar <u>Vegetación y Fauna</u>.</p> <p>Test de Listar <u>Vegetación y Fauna</u>.</p> <p>Test de Buscar <u>Vegetación y Fauna</u>.</p>
10	Gestionar Calidad del agua	<p>Test de Insertar Calidad del agua.</p> <p>Test de Modificar Calidad del agua.</p> <p>Test de Eliminar Calidad del agua.</p> <p>Test de Listar Calidad del agua.</p> <p>Test de Buscar Calidad del agua.</p>
11	Gestionar Antropogénesis	<p>Test de Insertar Antropogénesis.</p> <p>Test de Modificar Antropogénesis.</p> <p>Test de Eliminar Antropogénesis.</p> <p>Test de Listar Antropogénesis.</p> <p>Test de Buscar Antropogénesis.</p>
12	Gestionar Estado Ecológico-Ambiental	<p>Test de Insertar Estado Ecológico-Ambiental.</p> <p>Test de Modificar Estado Ecológico-Ambiental.</p> <p>Test de Eliminar Estado Ecológico-Ambiental.</p> <p>Test de Listar Estado Ecológico-Ambiental.</p> <p>Test de Buscar Estado Ecológico-Ambiental.</p>
13	Gestionar Análisis de la gestión	<p>Test de Insertar Análisis de la gestión.</p> <p>Test de Modificar Análisis de la gestión.</p> <p>Test de Listar Estado Ecológico-Ambiental.</p> <p>Test de Buscar Análisis de la gestión.</p>
14	Gestionar Monitoreo	<p>Test de Insertar Monitoreo.</p>

		Test de Modificar Monitoreo. Test de Eliminar Monitoreo. Test de Listar Monitoreo. Test de Buscar Monitoreo. Test de Predecir Monitoreo. Test de Imprimir Monitoreo
15	Generar Reporte	Test de Realizar Reporte.

3.3.3 Pruebas de Aceptación

De manera general se explicó en el epígrafe anterior cada una de las pruebas, ahora se pretende aplicar alguna de ellas al sistema de gestión integral de playas.

Con la aplicación de las pruebas de aceptación se pretende comprobar que el software cumple las expectativas que el cliente espera. Para llevar a cabo la validación del sistema se decidió aplicar pruebas de aceptación a los Elementos de la Pila de Producto (EPP). Las pruebas de aceptación son destinadas a evaluar si al terminar un Sprint se consiguió la funcionalidad requerida por el cliente. Estas pruebas aseguran el comportamiento del sistema y especifican los aspectos a probar cuando un Sprint ha sido correctamente implementado. Como resultado de las pruebas de aceptación se obtendrán artefactos descritos en tablas, estas contarán con los siguientes campos:

- ✓ Código: servirá como identificador de la prueba realizada, a su vez será sugerente al nombre de la prueba a la que hace referencia.
- ✓ EPP: tendrá el nombre del elemento de la pila de producto al que hace referencia la prueba a realizar (Caso de Uso).
- ✓ Nombre: nombre que se le da a la prueba a realizar.
- ✓ Descripción: se describe la funcionalidad que se desea probar.
- ✓ Condiciones de Ejecución: mostrará las condiciones que deben cumplirse para poder llevar a cabo el caso de prueba, estas condiciones deben ser satisfechas antes de la ejecución del caso de prueba para que se puedan obtener los resultados esperados.
- ✓ Entradas / Pasos de Ejecución: se hará la descripción de cada uno de los pasos seguidos durante el desarrollo de la prueba, se tendrá en cuenta cada una de las entradas que hace el usuario con el objetivo de ver si se obtiene el resultado esperado.
- ✓ Resultado esperado: se hará una breve descripción del resultado que se espera obtener con la prueba realizada.

- ✓ Evaluación de la prueba: acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:
 - Satisfactoria: cuando el resultado de la prueba es exactamente el esperado por el usuario.
 - Parcialmente satisfactoria: cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.
 - No satisfactoria: cuando el resultado de la prueba realizada genera un error de codificación en la aplicación o muestra como resultado elementos no deseados o fuera de contexto, trayendo como consecuencia que la funcionalidad requerida por el cliente no tenga resultado, lo que invalida el EPP.

Se ha tomado una muestra al azar de las pruebas de aceptación.

21. Tabla #21: Prueba de Aceptación 1 (Pruebas al Software)

Pruebas de Aceptación	
Código o Número de Caso de Prueba: 01	Nombre EPP: Diseño y creación de la base de datos
Nombre Caso de Prueba: Test de integridad a la base de datos	
Descripción: Verificar que la inserción, edición, eliminación de datos y las relaciones entre los mismo sean correctas	
Condiciones de ejecución: Servidor PostgreSQL 14 ejecutándose como administrador de base de datos	
Entradas: Datos de Prueba	
Resultado esperado: Todos los conjuntos de datos probados han sido correctamente procesados	
Evaluación: Prueba satisfactoria	

22. Tabla #22: Prueba de Aceptación 2 (Pruebas al Software)

Pruebas de Aceptación	
Código o Número de Caso de Prueba: 02	Nombre EPP: Diseño de la interfaz de usuario.
Nombre Caso de Prueba: Test de adaptabilidad a la Interface de Usuario (UI).	
Descripción: Verificar que la UI se adapte de forma correcta y óptima según la concepción del diseño para las diferentes resoluciones de los dispositivos	
Condiciones de ejecución: Tener acceso a la red con cualquier dispositivo que permita la navegación Web.	
Entradas: Interfaces de la aplicación	
Resultado esperado: Se muestren las interfaces visuales de la aplicación con las características que se diseñaron según la resolución del dispositivo en el cual se va a ejecutar.	
Evaluación: Prueba satisfactoria	

23. Tabla #23: Prueba de Aceptación 3 (Pruebas al Software)

Pruebas de Aceptación	
Código o Número de Caso de Prueba: 03	Nombre EPP: Autenticarse
Nombre Caso de Prueba: Test de autenticación de usuario	
Descripción: Autenticar un usuario	
Condiciones de ejecución: El usuario tiene que estar agregado en la base de datos y activo	
Entradas:	

Nombre de usuario y contraseña
<p>Resultado esperado:</p> <p>El usuario es autenticado y según el rol que desempeña la aplicación debe mostrar la vista de admin de Django para el Súper Usuario administrador total del sistema, la vista con el insertar zona si es un Súper Gestor y la vista de las playas que le corresponden a una zona determinada si es un Gestor.</p>
<p>Evaluación:</p> <p>Prueba satisfactoria</p>

24. Tabla #24: Prueba de Aceptación 4 (Pruebas al Software)

Pruebas de Aceptación	
Código o Número de Caso de Prueba: 04	Nombre EPP: Gestionar Playa
Nombre Caso de Prueba: Test de eliminar Playa	
<p>Descripción:</p> <p>Eliminar una Playa de la lista de playas</p>	
<p>Condiciones de ejecución:</p> <p>El usuario tiene que estar autenticado con el rol de Gestor y la playa no puede contener datos almacenados.</p>	
<p>Entradas:</p> <p>Seleccionar playa a eliminar</p>	
<p>Resultado esperado:</p> <p>El sistema muestra una ventana modal con un texto donde advierte al usuario que está a punto de eliminar una playa y dos botones de acción: uno para cancelar la operación y otro para ejecutarla.</p>	
<p>Evaluación:</p> <p>Prueba satisfactoria</p>	

25. Tabla #25: Prueba de Aceptación 5 (Pruebas al Software)

Pruebas de Aceptación	
Código o Número de Caso de Prueba: 05	Nombre EPP: Gestionar Zonas
Nombre Caso de Prueba: Test de insertar Zona	
Descripción: Insertar una nueva zona al sistema	
Condiciones de ejecución: El usuario tiene que estar autenticado con el rol de Súper Gestor y debe tener un usuario Gestor para su selección y poder operarla	
Entradas: Seleccionar insertar zona	
Resultado esperado: El sistema agrega la zona y le asigna el gestor escogido para su operación de gestión.	
Evaluación: Prueba satisfactoria	

26. Tabla #26: Prueba de Aceptación 6 (Pruebas al Software)

Pruebas de Aceptación	
Código o Número de Caso de Prueba: 06	Nombre EPP: Gestionar Calidad del Agua
Nombre Caso de Prueba: Test modificar calidad del agua	
Descripción: Modificar calidad del agua	
Condiciones de ejecución: El usuario tiene que estar autenticado con el rol de Gestor de la zona que se le esté asignada y solo lo podrá realizar con la última entrada de dato que exista en el sistema.	
Entradas:	

Seleccionar modificar
<p>Resultado esperado:</p> <p>El sistema habilita el formulario de modificación para que se le pueda hacer el cambio y permite guardar con la fecha y hora actual en el momento de dicho cambio.</p>
<p>Evaluación:</p> <p>Prueba satisfactoria</p>

27. Tabla #27: Prueba de Aceptación 7 (Pruebas al Software)

Pruebas de Aceptación	
Código o Número de Caso de Prueba: 07	Nombre EPP: Gestionar Monitoreo
Nombre Caso de Prueba: Test predecir Monitoreo	
<p>Descripción:</p> <p>Acceder al apartado del Monitoreo de los parámetros guardados de una playa</p>	
<p>Condiciones de ejecución:</p> <p>El usuario tiene que estar autenticado en el sistema con el rol de Súper Gestor o Gestor solamente, además tiene que existir al menos cuatro datos históricos del parámetro de playa a predecir.</p>	
<p>Entradas:</p> <p>Seleccionar predecir parámetro</p>	
<p>Resultado esperado:</p> <p>El sistema nos lleva a una nueva vista donde muestra el gráfico de los datos recolectados y el gráfico de las predicciones del parámetro a medir con el valor de su error cuadrático medio.</p>	
<p>Evaluación:</p> <p>Prueba satisfactoria</p>	

3.3.4 Pruebas de Caja Negra

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

Muchos autores consideran que estas pruebas permiten encontrar:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores en estructuras de datos o en accesos a las Bases de Datos externas.
4. Errores de rendimiento.
5. Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen a partir de que se atiende a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

1. Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
2. Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
3. Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduce así en número de clases de prueba que hay que desarrollar.

Los campos que comprenden estas pruebas son los siguientes:

- 1 Funcionalidad: es el nombre del caso de uso a probar
- 2 Código: Constituye un indicador de la prueba realizada.
- 3 Pre requisito: se consideran los elementos fundamentales para darle inicio a la funcionalidad definida.
- 4 Número: número de cada una de las actividades que se realizan para hacer las pruebas.
- 5 Nombre: Contiene el nombre de la actividad que se va a ejecutar para hacer las pruebas.
- 6 Descripción: se describe el funcionamiento de cada una de las actividades a las que se le van a realizar la prueba.
- 7 Respuesta esperada: Representa la respuesta que se espera al ejecutar la actividad definida.
- 8 Respuesta obtenida: Se clasifica en OK si la respuesta que se obtiene es la correcta.

A continuación, se muestran las tablas relacionadas con las pruebas de caja negra realizadas a la aplicación web:

28. Tabla #28: Prueba de Caja Negra 1 (Pruebas al Software)

Funcionalidad		Gestionar Usuarios		
Código		C01		
Pre requisito		Estar autenticado en el sistema con el rol de Súper Usuario administrador total		
No	Nombre	Descripción	Respuesta Esperada	Respuesta Obtenida
1	Cambiar contraseña a un usuario	Se accede al apartado de usuarios y se escoge al que se le va a realizar el cambio y después se guarda el cambio	El sistema coloca al nombre de usuario en modo formulario y le pide la nueva contraseña y confirmación de la	OK

			misma con los requisitos especiales y la cambia.	
2	Cambiar de Rol o Grupo	Se accede al apartado de grupos de usuarios, se elige y se escoge uno entre los roles establecidos previamente y se pasan a la sección de establecer rol para el usuario seleccionado y se guarda la operación.	Hacer el cambio del rol seleccionado desde el apartado de grupos de roles para el apartado de rol seleccionado y lo asigna.	OK

29. Tabla #29: Prueba de Caja Negra 2 (Pruebas al Software)

Funcionalidad		Gestionar Zonas		
Código		C02		
Pre requisito		Estar autenticado en el sistema con el rol de Súper Gestor		
No	Nombre	Descripción	Respuesta Esperada	Respuesta Obtenida
1	Insertar una zona	Se accede al módulo de Gestión Ambiental, se oprime el botón de Insertar Zona en la vista.	El sistema muestra un formulario que pide los datos correspondientes a la nueva zona agregar así como el Usuario (Gestor) que va a operar dicha zona.	OK
2	Modificar zonas	Una vez agregada la zona se accede al	El sistema muestra un formulario en una nueva	OK

		botón modificar (símbolo de lápiz editor).	vista, dónde deja cambiar los datos y al Gestor asignado en su lista.	
3	Eliminar	Una vez agregado la zona, se accede al botón de eliminar.	El sistema mostrará el botón de eliminar zona siempre y cuando esté vacía, es decir no tenga datos la zona. En el caso de que esté vacía la zona, mostrará un modal que va alertar la eliminación con dos botones, uno de cancelar y otro de aceptar la operación.	OK

30. Tabla #30: Prueba de Caja Negra 3 (Pruebas al Software)

Funcionalidad		Gestionar Estado Ecológico-Ambiental		
Código		C03		
Pre requisito		Estar autenticado en el sistema con el rol de Gestor del sistema		
No	Nombre	Descripción	Respuesta Esperada	Respuesta Obtenida
1	Insertar una estado ecológico-ambiental	Acceder al módulo de Gestión Ambiental, después al sub-módulo de Estado Ecológico-Ambiental y presionar el botón de insertar	El sistema muestra una nueva vista con un formulario para su correcto llenado y guardado.	OK

2	Modificar	Acceder al módulo de Gestión Ambiental, después al sub-módulo de Estado Ecológico-Ambiental y presionar el botón de modificar(Lápiz editor)	El sistema mostrará una nueva vista con el formulario llenado en espera de modificación.	OK
3	Eliminar	Acceder al módulo de Gestión Ambiental, después al sub-módulo de Estado Ecológico-Ambiental y presionar el botón de eliminar	El sistema no mostrará el botón de eliminar porque una vez se tengan los datos almacenados solo pueden ser modificados por el Gestor, el súper administrador es el único que puede borrar.	OK
4	Ver el estado ecológico ambiental de otra zona	Acceder al módulo de Gestión Ambiental y presionar ver detalles y después sobre el sub-módulo de Estado Ecológico-Ambiental	El sistema no permitirá ver el estado ecológico ambiental de otra zona que fue asignada a un gestor diferente, solo se pueden ver dicho estado de las zonas que administre el mismo Gestor.	OK

3.4 Análisis de los resultados obtenidos

Después de desarrollar todo un proceso de pruebas con un nivel medio de sencillez se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las realizadas que impedían el óptimo funcionamiento de la aplicación, puede hacerle entrega al cliente en el tiempo establecido, un producto funcional y que satisface todas sus necesidades se solucionaron varios problemas que impedían el cumplimiento de los requisitos fundamentales del sistema en cuestión. Las primeras pruebas fueron planeadas y ejecutadas en módulos individuales del programa, así como en funcionalidades específicas y a medida que se avanzó se desplazaron a módulos integrados, hasta que finalmente llegaron al sistema completo y se logró obtener un software cuyas funciones se encuentran en correspondencia con las

especificaciones acordadas y que además cumple con los requerimientos de rendimiento y seguridad. El desarrollo de la aplicación cumple las expectativas trazadas al inicio del proyecto y satisface al cliente en su totalidad.

Este sistema es de gran beneficio debido a que brinda muchas ventajas tanto en tiempo como en costo. Permite una manipulación segura de datos confidenciales, así como hacer consultas dinámicas sobre la información que deseen. También se podrá exportar datos a documentos del formato deseado para ser impresos o enviados por correo. Está realizado para lograr mayor seguridad y confiabilidad en los datos, así como mayor exactitud y organización lo cual estimula el cumplimiento de nuevas estrategias de mejoras.

3.5 Conclusiones del Capítulo

Las pruebas realizadas a medida que se utiliza las técnicas anteriormente planteadas fueron de gran importancia para demostrar el buen funcionamiento del software y el cumplimiento de las peticiones del cliente. Como resultado final se obtiene una aplicación Web con una apariencia atractiva y fácil de usar, adaptable a cualquier tipo de dispositivos, con todas las funcionalidades requeridas que cumple y satisface todas las expectativas del cliente

Una vez construida la propuesta de solución al problema científico de esta investigación a partir de la metodología SCRUM, y analizado los resultados de la misma a partir de las pruebas realizadas, se concluye que:

- ✓ La planificación fue acertada, pues permitió el desarrollo del sistema según el cronograma y cumple con los objetivos del cliente al iniciar esta investigación.
- ✓ El uso de pruebas durante el desarrollo del sistema permitió tanto al cliente como al desarrollador encontrar defectos y errores en el sistema y su rápida corrección.
- ✓ A partir del diseño de la propuesta y con el uso de las tecnologías adecuadas para su implementación se obtuvo un sistema que cumple a cabalidad con la seguridad y confiabilidad que esperaba el cliente.

4 Conclusiones Generales

Como resultado de esta investigación quedaron satisfechos los objetivos trazados se arriba a las siguientes conclusiones:

- ✓ El estudio realizado sobre los antecedentes, el estado actual de la temática, la bibliografía y documentos relacionados con el objeto de estudio, permitió contar con los elementos necesarios para dar solución a la problemática planteada.
- ✓ No se encontró ningún sistema automatizado, vinculado al tema, que le diera solución al problema planteado.
- ✓ Se utilizaron las herramientas de software más factibles para la construcción de la solución.
- ✓ Se pretende implantar el sistema informático para la gestión de la información en el Observatorio Científico Costa Atenas y el CITMA, se cumple con el cronograma de desarrollo planteado al cliente.
- ✓ La realización de las pruebas funcionales a este sistema informático permitió detectar errores en el sistema y la rápida corrección de los mismos.
- ✓ La implementación del sistema y la aplicación de las pruebas de validación resultados satisfactorios demostraron que el software elaborado cumple con los requerimientos especificados constatándose, mediante avales, su aporte práctico a la entidad.

El software desarrollado es una herramienta confiable y fácil de utilizar, que enriquece el caudal informativo a la búsqueda de datos para investigaciones, reduce los errores a cometer y aumenta un mayor cuidado a las zonas costeras y playas al tener un mayor control sobre éstas, facilita y perfecciona el proceso y la búsqueda de información, así como una mayor organización. Lo planteado deriva en la elevación de la eficiencia en la productividad, un mayor acceso a la información que constituye la vía principal para asegurar el desarrollo económico de nuestra sociedad y así un paso más a la informatización de nuestro país.

5 Recomendaciones

Después de realizar un análisis exhaustivo del presente trabajo en los capítulos referidos anteriormente y haber llegado a conclusiones, se pueden plantear las siguientes recomendaciones:

- ✓ Divulgar los resultados de la presente investigación.
- ✓ Propiciar nuevos estudios en aras de lograr un mayor nivel de informatización de los procesos.
- ✓ Generalización de los resultados de la presente investigación en el resto de los centros homólogos del país.
- ✓ Añadir otras funcionalidades al sistema que permita una mejor gestión de la información a partir de nuevas situaciones que puedan presentarse en otras investigaciones y así hacer el sistema más robusto e integral.
- ✓ Incorporar las trazas al sistema.

6 Bibliografía

1. Albers, B. P. (2010). Pro HTML5 Programming.
2. Anaconda. (26 de Noviembre de 2022). Obtenido de <https://www.anaconda.com/about-us>
3. Cantón, A. C. (2013). Manual de HTML5 en español.
4. Cendero Uceda, A., Sánchez-Arcilla Conejo, A., & Zazo Cardeñas, C. (s.f.). Impactos sobre las Zonas Costeras. Impactos del Cambio Climático en España, 469-475.
5. Chaffer, J., & Swedberg, K. (2009). Learning JQuery 1.3: Better Interaction and Web Development with 21 Simple JavaScript Techniques. Packt Publishing.
6. Consortium, W. W. (2018). Borrador actual de especificaciones de HTML5. Obtenido de <http://dev.w3.org/html5/spec/Overview.html>
7. Cristian. (2018). Ventajas de usar CSS. Obtenido de <http://www.stardustxs.com/2008/03/05/ventajas-de-usar-css/>
8. de la Torre Llorente, C., Zorrila Castro, U., Ramos Barroso, M., & Calvarro Nelson, J. (2010). Guía de Arquitectura N-Capas orientada al Dominio con .Net 4.0. España: España-Printed in Spain.
9. Duarte, E. (2012). Javascript lenguaje de programación: ventajas y desventajas. Obtenido de blog.capacityacademy.com/2012/10/19/
10. Flanagan, D. (2002). JavaScript: The Definitive Guide.
11. González Ramírez, Y. (2017). Sistema informático para la gestión trimestral de los trabajadores en el área de Recursos Humanos en la Empresa Integral Turística Ciénaga de Zapata en Matanzas.
12. Goold, D. H. (2014). Aplicación de gestión de playas.
13. Hernández Cabrera, J., Alfonso Martínez, A., Sánchez Roque, O., & Pérez Martínez, L. (2019). Ordenamiento ambiental y gestión costera integrada: experiencias y retos en el norte de la provincia de Matanzas(Cuba). Matanzas.
14. Holovaty, A., & Kaplan-Moss, J. (2009). The Definitive Guide to Django: Web Development Done Right 2nd ed. Edición . Apress.
15. IBM. (19 de Noviembre de 2022). Obtenido de www.ibm.com/ar-es/topics/software-testing
16. Informática del Este. (15 de Noviembre de 2021). iest. Obtenido de <https://www.iest.com/winplus-playas/>
17. J. D., G. (2012). El gran libro de HTML5, CSS3 y Javascript.
18. Kniberg, H. (2007). SCRUM y XP desde las Trincheras, Como hacemos Scrum. Impreso en EEUU.
19. Library. (21 de Noviembre de 2022). Obtenido de <https://1library.co/article/metodolog%C3%ADas-desarrollo-%C3%A1gil-marco-te%C3%B3rico-tecnol%C3%B3gico.zgwmn3vy>

20. Loaiza. (2010). Plan de Pruebas de Software.
21. Marcos-Jorquera, D., Gilart, V., Aragonés, L., López, I., & Ignacio, J. (2019). Sistema de gestión de incidencias para usuarios de playas (ClaimBeachApp). San Vicente de Raspeig, España.
22. Mesa, S. A. (2015). Metodología de desarrollo de software basada en la experiencia de usuario. Medellin, Colombia.
23. Peña Álvarez, M. (24 de Noviembre de 2007). Obtenido de Arquitectura cliente-servidor: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>
24. Pérez, I. A. (2018). Aplicación web para la gestión de la información asociada a los Recursos Humanos en la División de la Empresa de Atención a Equipos Matanzas. . Matanzas, Cuba.
25. Quijano Poumián, M., & Rodríguez Aragón, B. (2004). El Marco Legal de la zona costera. En G. V. Zapata, El Manejo Costero en México (págs. 69-84). Universidad Autónoma de Campeche, Mexico: CETYS-Universidad.
26. Sánchez, E., & Letelier, P. (2003). Metodologías ágiles en el Desarrollo de Software. Ingeniería del Software y Sistemas de Información. España: Alicante.
27. Sanz. (2010). Pruebas de aceptación en Sistemas Navegables. Revista Española.
28. Schwaber, K., & Sutherland, J. (2013). La Guía de Scrum:Las Reglas del Juego.
29. Universidad de Alicante. (6 de noviembre de 2022). Repositorio Institucional de la universidad de Alicante. Obtenido de <http://rua.ua.es/dspace/handle/10045/112681>
30. Visual Paradigm. (25 de Noviembre de 2022). Obtenido de <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>
31. Yepes, V. (2002-2018). Ordenación y gestión del territorio turístico, Las Playas, Blanquer. Valencia: Tirant lo Blanch.