



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 1**

Aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas.

**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: Alejandro David Santana Rodríguez

Tutores: Dr. C. Liz Pérez Martínez
Ing. Javier Anias Santos

**La Habana, noviembre de 2022
Año 64 del Triunfo de la Revolución**

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “**Aplicación web para la realización de recorrido virtual por el Jardín Botánico de Matanzas**” concede a la Universidad de las Ciencias Informáticas y a la Universidad de Matanzas los derechos patrimoniales de la investigación, con carácter exclusivo así como yo **Alejandro David Santana Rodríguez**, con carné de identidad **96040910584** soy el autor principal del trabajo y autorizo a la Universidad de las Ciencias Informáticas y a la Universidad de Matanzas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo. De forma similar me declaro como único autor de su contenido. Para que así conste se firma la presente a los 16 días del mes de noviembre del año 2022.

Alejandro David Santana Rodríguez

Firma del Autor

Dr. C. Liz Pérez Martínez



Firma del Tutor

Ing. Javier Anias Santos

Firma del Tutor

DATOS DE CONTACTO

L. Pérez, recibió sus títulos como Ingeniero Informático, MSc. en Administración de Empresas y Doctor en Ciencias Técnicas, en la Universidad de Matanzas, Matanzas, Cuba en 2012, 2015 y 2022 respectivamente. En la actualidad es Profesor Auxiliar colaborador del Departamento de Informática y Decana de la Facultad de Ciencias Técnicas de la Universidad de Matanzas, Matanzas, Cuba. Sus principales intereses de investigación incluyen aplicaciones de la Inteligencia Artificial a la gestión de negocios y a la gestión ambiental.

J. Anias, recibió sus títulos como Ingeniero Informático en la Universidad de Ciencias Informáticas, La Habana, Cuba. En la actualidad es Profesor Auxiliar en la Facultad 1 de la Universidad de Ciencias Informáticas, La Habana, Cuba.

RESUMEN

El Jardín Botánico de Matanzas, perteneciente a la Universidad de Matanzas, presenta ciertas deficiencias asociadas a su ubicación física, lo que atenta contra su visibilidad de cara a científicos, investigadores y público en general interesado en abundar sobre las especies que allí se conservan. La presente investigación da solución a esta problemática, a través del desarrollo de una aplicación web que actúa como recorrido virtual y que acerca a los interesados a dicha institución de una manera no presencial. La solución fue realizada a partir de los elementos teóricos y metodológicos relacionados con este proceso y la disciplina de ingeniería de software. La investigación contiene todas las fases por la cual pasó la solución, acorde a la metodología aplicada con cada una de las descripciones y artefactos necesarios para el desarrollo y pruebas. La aplicación propuesta servirá de apoyo a la comunidad universitaria, científica y botánica, así como a los visitantes, al dotarlos de una herramienta de percepción genérica para el recorrido virtual, extensible que sirva como guía informatizada para transitar por el Jardín Botánico de Matanzas.

PALABRAS CLAVE

Desarrollo web, informatización, Jardín botánico, recorrido virtual.

ABSTRACT

The Matanzas Botanical Garden, which belongs to the University of Matanzas, has certain deficiencies associated with its physical location, which hinders its visibility to scientists, researchers and the general public interested in learning more about the species preserved there. The object of this research is the virtual tour of the Matanzas Botanical Garden. It is proposed as a software solution the development of a web application for the realization of a virtual tour based on web services programming. The solution was developed based on the theoretical and methodological elements related to this process and the discipline of software engineering. The research contains all the phases through which the solution passed, according to the methodology applied with each of the descriptions and artifacts necessary for development and testing. The proposed application will support the university, scientific and botanical community, as well as visitors, by providing them with a generic perception tool for the virtual tour, extensible to serve as a computerized guide to transit through the Matanzas Botanical Garden.

KEYWORDS

Botanical, computerization, virtual tours, web development.

ÍNDICE

| | |
|--|----|
| Introducción | 9 |
| Capítulo 1: Fundamentos teóricos de una aplicación web para la realización del recorrido virtual por el jardín botánico de matanzas..... | 14 |
| 1.1 Definición de conceptos..... | 14 |
| 1.2 Análisis comparativo de aplicaciones informáticas para la realización del recorrido virtual..... | 17 |
| 1.3 Metodología de desarrollo de software..... | 24 |
| 1.4 Herramientas y tecnologías informáticas para la realización del recorrido virtual | 28 |
| 1.4.1 Interfaz:..... | 28 |
| 1.4.2 Back-end:..... | 30 |
| 1.4.3 Diseño y modelado: | 31 |
| 1.4.4 Entorno de desarrollo integrado..... | 32 |
| 1.5 Conclusiones del capítulo..... | 32 |
| Capítulo 2: análisis y diseño de la propuesta de solución de una aplicación web para la realización del recorrido virtual por el jardín botánico de matanzas..... | 34 |
| 2.1 Modelado del negocio..... | 34 |
| 2.1.1 Descripción del proceso de realización del recorrido virtual por el jardín botánico de matanzas | 35 |
| 2.2 Requisitos del recorrido virtual | 36 |
| 2.2.1 Requisitos funcionales | 36 |
| 2.2.2 Requisitos no funcionales | 38 |
| 2.2.3 Historias de usuario | 39 |
| 2.3 Análisis y diseño..... | 44 |
| 2.3.1 Modelo de diseño..... | 46 |
| 2.4 Patrones de diseño..... | 46 |
| 2.4.1 Patrones grasp:..... | 46 |
| 2.4.2 Patrones gof:..... | 48 |
| 2.5 Conclusiones del capítulo..... | 49 |
| Capítulo 3: Implementación y pruebas de la solución propuesta para una aplicación web para la realización del recorrido virtual por el jardín botánico de matanzas. | 50 |
| 3.1 Implementación de la propuesta de solución..... | 50 |
| 3.1.1 Diagrama de despliegue | 50 |

| | |
|--|----|
| 3.1.2 Interfaz gráfica de usuario | 51 |
| 3.2 Validación de requisitos..... | 54 |
| 3.2.1 Técnicas de validación de requisitos | 54 |
| 3.2.2 Resultados al aplicar técnicas de revisión | 55 |
| 3.3 Pruebas | 55 |
| 3.3.1 Pruebas funcionales | 56 |
| 3.3.2 Pruebas de rendimiento..... | 58 |
| 3.3.3 Pruebas de seguridad..... | 60 |
| 3.4 Conclusiones del capítulo..... | 62 |
| Conclusiones | 63 |
| Recomendaciones | 64 |
| Referencias bibliográficas..... | 65 |
| Anexos..... | 68 |

TABLAS

| | |
|--|----|
| Tabla 1.1 Valoración de los criterios a seguir en el desarrollo de aplicaciones web para recorridos virtuales en el contexto nacional. | 22 |
| Tabla 2.1 Roles del negocio. | 36 |
| Tabla 2.2 Requisitos funcionales. | 36 |
| Tabla 2.3 Historia de usuario. Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. | 41 |
| Tabla 2.4 Historia de usuario. Observar las imágenes panorámicas cilíndricas referentes a las salas de un recorrido virtual. | 42 |
| Tabla 2.5 Plan de iteraciones. | 43 |
| Tabla 2.6 Tarjeta CRC para la Clase Controladora. | 46 |
| Tabla 3.1 Caso de prueba. Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. | 56 |
| Tabla 3.2 Simulaciones | 59 |
| Tabla 3.3 Riesgos | 60 |
| Tabla 3.4 Vulnerabilidades. | 61 |
| Tabla A 2.1 Historia de usuario. Mostrar la descripción correspondiente a la sala. | 69 |
| Tabla A 2.2 Historia de usuario. Ampliar y disminuir un área de la imagen panorámica perteneciente a la sala visualizada. | 69 |
| Tabla A 2.3 Historia de usuario. Girar en una imagen panorámica de forma horizontal. | 70 |
| Tabla A 2.4 Historia de usuario. Girar en una imagen panorámica de forma vertical. | 70 |
| Tabla A 2.5 Tarjeta CRC para Vista. | 71 |

FIGURAS

| | |
|---|-----|
| <i>Figura 1.1 Criterios a seguir en el desarrollo de aplicaciones web para recorridos virtuales</i> | 189 |
| <i>Figura 1.2 Aplicaciones web para la realización de recorridos virtuales</i> | 20 |
| <i>Figura 1.3 Evaluación de los criterios a seguir por las aplicaciones web para recorridos virtuales</i> | 21 |
| <i>Figura 2.1 Organización del patrón arquitectónico MVC</i> | 456 |
| <i>Figura 3.1 Diagrama de Despliegue</i> | 501 |
| <i>Figura 3.2 Pantalla de inicio</i> | 54 |
| <i>Figura 3.3 Pantalla de recorrido virtual</i> | 54 |
| <i>Figura 3.4 No conformidades</i> | 59 |

INTRODUCCIÓN

El siglo XX y el comienzo del XXI, se encuentra plagado de palabras que han ocupado su protagonismo, “medio ambiente” y “tecnologías de la información” son algunas de ellas, que albergan gran importancia. El cuidado y respeto por el medio ambiente se ha convertido en un valor significativo en todas las sociedades occidentales, se ha incorporado a los sistemas educativos y ha alcanzado tanto a las esferas públicas como privadas, posiblemente como consecuencia del aumento demográfico, del imparable consumismo, del agotamiento de recursos que parecían inagotables, del aumento exponencial de los residuos, de la imparable contaminación, y de las transformaciones en los valores que se lleva a cabo en la esfera humana (Cabero Almenara & Llorente Cejudo, 2005).

Ya en la carta de Belgrado formulada en 1975, se hablaba que el individuo debería ser formado en una mínima de competencias y capacidades medio ambientales. Por otra parte, no podemos dejar de reconocer que las Tecnologías de la Información y las Comunicaciones (TICs) se han convertido en un elemento de impulso y desarrollo de las sociedades occidentales, alcanzado a todos los sectores, desde los políticos y económicos, hasta los culturales y educativos, de forma que su no utilización está influyendo para la creación de zonas marginales y de pobreza y marginalidad tecnológica, es decir de lo que se comienza a conocer como brecha digital (Cabero Almenara, 2005).

Como parte del desarrollo tecnológico que vivimos en nuestros tiempos, la virtualidad ha incursionado en diversos aspectos del quehacer humano. Una de esas áreas ha sido la creación de recorridos virtuales, con fines tan variados como la publicidad (promoción de sitios e instalaciones turísticas), la cultura (representaciones de acervos museográficos o edificios históricos) y la educación (recorridos por instalaciones universitarias, bibliotecas, colecciones científicas), cuyo propósito es acercar a las personas a diversos lugares para conocerlos sin la necesidad de una presencia física, sino a través del uso de interfaces tecnológicas (García, 2021). Afortunadamente, la disponibilidad de medios y dispositivos tecnológicos para digitalizar a través de imágenes o videos, ha permitido la creación y publicación en la web de recorridos virtuales 360

(RV360). Un RV360 facilita, mediante panoramas equirrectangulares, crear un efecto de visualización que permite al usuario situarse en el centro de la imagen y poder observar en múltiples direcciones (básicamente arriba, abajo, izquierda y derecha) lo cual da una sensación de presencia en el lugar (García, 2021)

El incremento en los últimos tiempos de los espacios virtuales dedicados a recorridos de centros históricos, zonas de turismo, negocios de diferentes índoles y espacios educativos en los límites de nuestra nación, han sido significativos. Espacios como habana-360.com, cubaen360.com memorialgranma.cult.cu, casanatal.cult.cu y sadhea.uho.edu.cu son solo algunos de estos ejemplos donde se puede certificar el alcance e importancia que este avance tecnológico ha alcanzado en nuestro país.

Los impactos del proyecto se manifestarán directamente en los beneficios que puede aportar en cuanto promoción y divulgación del Jardín Botánico de Matanzas, pues se persigue un acercamiento y mejor aprovechamiento de los resultados de las investigaciones científicas desarrolladas en este espacio, a través del desarrollo e implementación de herramientas informáticas que ayuden a la medición, adquisición de datos. Contribuirán también en la elevación de la cultura de profesionales sobre esta temática, al contar con información confiable.

El jardín Botánico de Matanzas, en su estrecha relación con la Bahía, tiene poca divulgación a nivel nacional e internacional como centro de ciencias, esto es un obstáculo para obtener una mayor colaboración de profesionales de esta especialidad para su crecimiento científico. La pobre promoción que posee como zona de turismo impide que cuente con un aporte monetario tanto para su propio sustento como para apoyar en este sentido a la provincia de Matanzas y el país. Su presencia como centro de cuidado y protección ambiental pasa de ser percibida debido al desconocimiento de las persona sobre su ubicación y existencia, lo cual imposibilita que cumpla su objetivo de llevar la educación ambiental y botánica a la mayor cantidad de personas posibles. La distancia a la que se encuentra de otros sitios de interés, especialistas de la rama e interesados y entusiastas de la botánica, es una limitación que frena su desarrollo científico, turístico y como foco de resguardo medio ambiental.

Para dar solución a la problemática descrita anteriormente se presenta como **problema de investigación**: La ubicación física del Jardín Botánico y su falta de un espacio virtual afectan a la accesibilidad de la información científica por parte de los interesados

Para dar solución al problema planteado se establece como **objetivo general** desarrollar una aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas mediante la contribución a la elaboración de la aplicación.

El **objeto de estudio** lo constituye el desarrollo de recorridos virtuales y el **campo de acción** será el desarrollo de recorridos virtuales a través de aplicaciones web.

En aras de encaminar la investigación y lograr el objetivo, se formularon las siguientes preguntas científicas:

1. ¿Cuáles son los fundamentos teóricos que permiten caracterizar el estado del arte de la problemática planteada, así como las tecnologías para diseñar e implementar la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas?
2. ¿Qué elementos y componentes de software se deben considerar en el diseño de la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas?
3. ¿Qué elementos y componentes de software se deben implementar en la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas?
4. ¿Qué pruebas de software se deben aplicar en la validación de la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas?

Para dar cumplimiento al objetivo propuesto se formularon las siguientes **tareas de investigación**:

1. Análisis los fundamentos teóricos que sustentan el desarrollo de recorridos virtuales.
2. Análisis del estado del arte la problemática planteada.
3. Estudio y selección de las tecnologías para el desarrollo de recorridos virtuales.
4. Análisis, desde el punto de vista ingenieril, de la aplicación para el desarrollo de recorridos virtuales.
5. Diseño, desde el punto de vista ingenieril, de la aplicación para el desarrollo de recorridos virtuales.
6. Implementación de la aplicación para el desarrollo de recorridos virtuales.

7. Validación, a partir de una estrategia de prueba de la aplicación para el desarrollo de recorridos virtuales.

Métodos de investigación científica:

Se definen en esta sección los métodos de investigación que ayudan a identificar la estrategia para el diseño de los experimentos. El método científico se puede clasificar en teóricos y empíricos, los cuales están dialécticamente relacionados (Sampieri, 2018).

Métodos teóricos

Analítico-sintético: con este método se realizó el análisis de la problemática, las fuentes bibliográficas existentes referentes al proceso de realización del recorrido virtual por el Jardín Botánico de Matanzas y las alternativas de solución. Se identifica de esta manera conceptos, definiciones y otros elementos de utilidad para la investigación.

Inductivo-deductivo: a través de este método se pudo arribar a conclusiones generales sobre la realización del recorrido virtual por el Jardín Botánico de Matanzas y permitió definir la estructura de la realización de recorrido virtual.

Métodos empíricos

Entrevista: Aportó la descripción del proceso de realización de recorrido virtual por el Jardín Botánico de Matanzas así como los conceptos asociados que más se ajustan a este trabajo.

Observación: a través la de este método se tuvo una mayor percepción directa del objeto de investigación y permitió conocer, de forma efectiva, el objeto de estudio para luego describir y analizar situaciones sobre la realidad estudiada.

Entre los **aportes** de la investigación se destacan: el **teórico-investigativo**, al sentar las bases para futuras investigaciones, e integrar trabajos anteriores relacionados al tema; y el **práctico**, al desarrollar un sistema informático que asista a la realización del recorrido virtual por el Jardín Botánico de Matanzas.

El contenido del trabajo se encuentra distribuido de la siguiente forma:

- **CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DEL DESARROLLO DE RECORRIDOS VIRTUALES A TRAVÉS DE APLICACIONES WEB.** Aborda los conceptos principales en los que se basa la investigación. Se realiza un análisis del estado del arte, tanto a nivel internacional

como nacional. Puntualiza los lenguajes de programación, tecnologías y herramientas de desarrollo utilizadas para llevar a cabo la propuesta de solución.

- **CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN DEL DESARROLLO DE RECORRIDOS VIRTUALES A TRAVÉS DE APLICACIONES WEB.** Describe la propuesta de solución a desarrollar basada en la metodología seleccionada, para la disciplina de requisitos. El cual está comprendido por la descripción del negocio y modelo conceptual obteniendo así la descripción de requisitos. Además, contiene algunos de los artefactos ingenieriles que responden a la etapa de diseño.
- **CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA PARA EL DESARROLLO DE RECORRIDOS VIRTUALES A TRAVÉS DE APLICACIONES WEB.** Se muestra el resultado obtenido, así como los criterios y métodos utilizados para escribir código de solución, se especifica tipos de pruebas, modelado de diseño de caso de prueba y la estrategia de validación.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DEL DESARROLLO DE RECORRIDOS VIRTUALES A TRAVÉS DE APLICACIONES WEB.

Este capítulo contiene las referencias a los conceptos fundamentales sobre los cuales gira el desarrollo de recorridos virtuales. Se realiza un estudio de los sistemas homólogos a este tema, por último, se estudian las herramientas y tecnologías de desarrollo de software a emplear.

1.1 Definición de conceptos

Para el estudio y desarrollo de presente investigación se tomaron en cuenta una serie de conceptos elegidos según las necesidades, especificidades y requerimientos de esta investigación basada en el método empírico entrevista.

Virtualizar

Virtualizar ha sido considerado históricamente y de manera general como tomar algo en cierto estado y hacer parecer que se encuentra en otro estado diferente. Es una estrategia o metodología para dividir los recursos del ordenador en entornos de ejecución múltiples aplicando uno o más conceptos o tecnologías, tales como, particionamiento de software y hardware, tiempo compartido, simulación parcial o completa de ordenadores, emulación, calidad de servicio y muchos otros (Fernández Romero, 2011).

Se entiende por virtualización, como la abstracción de los recursos de TI físicos, tanto de las personas como de las aplicaciones que los utilizan. La virtualización permite a los servidores, elementos de almacenamiento y otros elementos hardware ser tratados como una pila de recursos, estos recursos pueden ser distribuidos según la demanda (Cebrián García, 2018).

La virtualización crea un entorno informático simulado, o virtual, en lugar de un entorno físico. A menudo, incluye versiones de hardware, sistemas operativos, dispositivos de almacenamiento, etc., generadas por un equipo. Esto permite a las organizaciones particionar un equipo o servidor físico en varias máquinas virtuales. Cada máquina virtual puede interactuar de forma independiente y ejecutar sistemas operativos o aplicaciones diferentes mientras comparten los recursos de una sola máquina host (Redmond, 2022).

En esta investigación se asume la virtualización como una tecnología que permite crear servicios de TI útiles, con recursos que están tradicionalmente limitados al hardware, gracias a que distribuye las funciones de una máquina física entre varios usuarios o entornos (Redmond, 2022).

Recorrido virtual

Un recorrido virtual es un entorno tridimensional que puede ser empleado como herramienta de apoyo, difusión y entrenamiento en diferentes áreas, como medicina, educación, turismo, milicia, entre otros. La calidad del entorno y ambientación contribuye, en gran medida, a la sensación inmersiva que el usuario percibe. De tal forma que se definen el entorno o recorrido virtual como la representación de objetos del mundo real a través de medios electrónicos (Ortega, 2014).

El recorrido virtual o *tour* virtual es la solución ideal para mostrar a los clientes de una forma fácil e interactiva todos los inmuebles o instalaciones de una sede. Por sus características naturales, puede ajustarse perfectamente a cualquier sector mercantil del país, entre otros, el inmobiliario, turístico y hotelero. Es adaptable a todos los estándares tecnológicos (Acevedo et al., 2016).

Un recorrido virtual permite situar al visitante en el centro de una imagen panorámica y para poder apreciar cualquier ángulo del sitio que virtualmente lo rodea. En este sentido, un recorrido es una herramienta que puede ser empleada con fines didácticos, o utilizada para acercar espacios a personas con capacidades diferentes, o aprovechada para concientizar sobre la diversidad de ecosistemas distantes o requeridos para minimizar riesgos y poder recorrer instalaciones industriales peligrosas (García, 2021).

Aunque en la literatura consultada se emplean varios términos para definir recorrido virtual, como visita virtual o paseo virtual, en la presente investigación se acogerá el primero. Por tanto, se asume recorrido virtual como una recreación de un entorno completamente virtual sobre el que se puede desplazar e interactuar para conocer y recorrer diferentes espacios libremente. En este caso, un recorrido virtual es la simulación de una ubicación existente (Jardín Botánico de Matanzas), con la ayuda de imágenes secuenciales. El recorrido virtual ayuda a presentar vistas a áreas inaccesibles y proporciona una alternativa que reduce gastos, tiempo y logística (García, 2021).

Foto equirrectangular

Las imágenes equirectangulares, son panoramas de 360x180 con vistas oblicuas en distintas direcciones abarcando todos los ángulos (Bedia Palacios, 2019).

Una foto equirectangular no es otra cosa que una representación plana de una vista esférica, que cubre un ángulo de 360 x 180 (Takeshita Caro, 2020).

Una imagen equirectangular es una imagen panorámica que permite poder apreciar cualquier ángulo del sitio que virtualmente lo rodea (García, 2021).

En la presente investigación el autor considera el formato equirectangular como la visión de los 360 grados de una imagen en una proyección rectangular, lo que brinda una perspectiva mucho más realista de un espacio, ya que un panorama completo de 360° abarca todos los ángulos y direcciones (García, 2021).

Jardín Botánico

Son espacios que se centran en la protección, exposición y estudio de las plantas o especies vegetales. Un jardín botánico se dedica a realizar investigaciones científicas sobre la biología centrada en las plantas y se comprometen a exhibir y educar sobre estas. Un jardín botánico no es solamente un espacio donde se alberga una inmensa colección de especies, este tipo de instituciones deben clasificar y tratar de informar y educar con rigor de las plantas. A parte de centrarse en los aspectos científicos los botánicos también son lugares increíbles donde se le da un valor especial a la estética donde las plantas cobran importancia y crean un entorno inspirador y realmente bonito (Forero, 1989).

El jardín botánico es un área bien cuidada que muestra una amplia gama de plantas etiquetadas con sus nombres botánicos. Puede contener colecciones especializadas de plantas de diversas partes del mundo; puede haber invernaderos, colecciones especiales, plantas tropicales, plantas alpinas y otras plantas exóticas (Castellano, 2003).

Lugar donde se cultivan plantas para realizar estudios botánicos y para ser mostradas al público con un grado razonable de permanencia, una base científica para las colecciones, documentación apropiada de las colecciones, incluyendo el origen silvestre, monitoreo de los individuos coleccionados, etiquetado adecuado de los ejemplares, abierto al público, comunicación e información hacia y desde otros Jardines, otras instituciones y el público, intercambio de semillas

u otros materiales con otros Jardines, arboreta u organismos de investigación, emprendimiento de investigación científica o técnica sobre los ejemplares de la colección, mantenimiento de programas de investigación en taxonomía, con herbario asociado, compromiso a largo plazo con el mantenimiento de las colecciones y promoción de la conservación a través de la extensión y las actividades de educación ambiental (Smith & Harvey-Brown, 2017).

En concordancia con lo anteriormente planteado, se considera en la presente investigación el concepto de jardín botánico como un tipo de jardín destinado a la conservación, investigación y divulgación de la diversidad de las especies vegetales. En este caso, coincide con estar enclavado en una institución habilitada por un organismo público, la Universidad de Matanzas (Smith & Harvey-Brown, 2017).

1.2 Análisis comparativo de aplicaciones informáticas para la realización del recorrido virtual.

A partir de la literatura consultada referidas a la creación de recorridos virtuales, como (Barrera et al., 2006) y (Leyva et al., 2017) y de la visita virtual a sitios de recorridos existentes y descritos por (Fleitas, 2018), (Toranzo, 2020) y (Ojeda, 2022), se resumen una serie de criterios que deben cumplir las aplicaciones con fines de recorridos virtuales (Figura 1.1).

Figura 4.1 Criterios a seguir en el desarrollo de aplicaciones web para recorridos virtuales

| Características | Descripción | Valores posibles |
|-----------------|--|--|
| Compartir | Posibilidad de compartir el recorrido en redes sociales. | Facebook, Twitter, Linkeid, Google, Whatsapp, Correo. |
| Controles de UI | Se refiere a la interfaz de usuario gráfica que permite navegar por los panoramas. | Izquierda, derecha, arriba, abajo, zoom in, zoom out, ocultar/mostrar controles. |
| Ejemplos | Recorridos virtuales de ejemplo incluidos. | No/Si |
| Hotspots | Vínculos que funcionan como ligas hacia otros panoramas o contenidos. | URL, imágenes, videos, textos, documentos, audios y vínculos a panoramas. |
| Idioma | Instrucciones en español . | No/Si |
| Logotipo | Inclusión de un logotipo personalizado en el recorrido. | No/Si |
| Mapa | Despliegue de un mapa 2D que permite al usuario ubicarse dentro del recorrido. | No/Si |
| Orientación | Capacidad de seleccionar el ángulo de visualización inicial en cada panorama. | No/Si |
| Publicación | Posibilidad de una vez creado el recorrido, publicarlo. | Publicación no permitida, con costo, gratuita por tiempo limitado, gratuita por tiempo ilimitado |
| Registro | Necesidad de registrarse con un usuario y contraseña para utilizar la aplicación. | No necesario, correo, google, facebook, twitter, microsoft, apple. |
| Sonido | Capacidad de incluir un audio de fondo. | No/Si |
| Títulos | Personalización de títulos en los panoramas del recorrido. | No/Si |
| Visor 360 | Adaptación del RV360 para su visualización en un visor. | No/Si |

Fuente: (García, 2021)

En el estudio realizado de las aplicaciones existentes destinadas a la realización de recorridos virtuales (Figura 1.2), se constató que ninguna de ellas responde a las características requeridas por el objeto de estudio de la presente investigación.

Figura 1.5 Aplicaciones web para la realización de recorridos virtuales

| | | | |
|----|----------------------|--|---|
| 1 | Cloud Pano | Plataforma para recorridos virtuales 360 que ofrece crear, hospedar y compartir éstos recorridos | https://www.cloudpano.com |
| 2 | Divein.studio | Plataforma para crear recorridos virtuales inmersivos. | https://www.divein.studio |
| 3 | Kuula | Software para cargar y compartir, a través de una comunidad, panoramas en 360 | https://kuula.co |
| 4 | Lapentor | Aplicación web para crear recorridos virtuales basados en la nube. | https://lapentor.com |
| 5 | Makevt | Editor visual basado en Krpano para crear y compartir recorridos con panoramas | https://makevt.com |
| 6 | Marzipano | Herramienta en línea que permite obtener un aplicación web para visualizar RV360. | https://www.marzipano.net |
| 7 | My360 | Software para la creación de experiencias visuales e interactivas. | https://my360.io |
| 8 | Orbix | Aplicacion web para la creación y publicación de RV360. | https://orbix360.com |
| 9 | Panoskin | Plataforma web para la creación de recorridos virtuales usando la tecnología de google | https://www.panoskin.com |
| 10 | Panaroo | Creación de mágicos recorridos virtuales en minutos. | https://panaroo.com |

Fuente: (García, 2021)

Se verificó que las aplicaciones para recorridos virtuales existentes a nivel mundial, no cumplen con los criterios valorativos definidos anteriormente en la Figura 1.1, tal y como refleja la (Figura 1.3).

Se aprecia que ninguna de las aplicaciones destinadas a la creación de recorridos virtuales cumple a cabalidad los criterios acordados para la presente investigación, y que fueron obtenidos a partir de las entrevistas realizadas al cliente (Anexo 1). Por lo que se evidencia la necesidad de desarrollar una aplicación propia, que satisfaga las necesidades planteadas por el cliente y que a su vez esté acorde a la política del país del empleo de software libre.

Figura 1.6 Evaluación de los criterios a seguir por las aplicaciones web para recorridos virtuales

| Plataforma Característica | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------------------------|---|---|--------------------------------|--|-------------------------------------|---|----------------------------------|--------------------------------|--------------------------------------|---|
| Compartir | Facebook, Twitter, Correo | Facebook, Whatsapp, Twitter, LinkedIn, Correo | No | No | No | No | Correo | Facebook, Twitter, Correo | Facebook, LinkedIn, Twitter | Facebook, Twitter, Whatsapp |
| Controles | Ocultar/ mostrar | Ocultar/ mostrar | Ocultar/ mostrar | Zoom in, zoom out, izquierda, derecha, arriba, abajo, ocultar/ mostrar | Zoom in, zoom out, ocultar/ mostrar | Zoom in, zoom out, izquierda, derecha, arriba, abajo. | Ninguno | Ocultar/ mostrar | Izquierda, derecha, Ocultar/ mostrar | Ocultar/ mostrar |
| Ejemplos | No | Si | No | No | No | Si | Si | No | No | Si |
| Hotspots | URL, texto, imagen, información, vínculo. | Video, imagen, texto, vínculo. | Ninguno | URL, video, imagen, documentos, infomacion, vínculo, | URL, video, texto, imagen,vínculo | Texto, vínculo | Video, texto, imagen, vínculo, | Texto, imagen, video vínculo | vínculo | URL, texto, video, audio, imagen, vínculo |
| Idioma | Español | Inglés | Inglés | Inglés | Inglés | Inglés | Inglés | Inglés | Inglés | Inglés |
| Logotipo | Si | Si | No | No | No | No | No | Si | No | Si |
| Mapa | Si | Si | Si | Si | No | No | No | No | No | No |
| Orientación | Si | Si | Si | Si | Si | Si | Si | No | No | Si |
| Publicación | Con costo | Sin costo por tiempo ilimitado | Sin costo por tiempo ilimitado | Sin costo por tiempo ilimitado | Sin costo por tiempo ilimitado | No permitida | Sin costo por tiempo ilimitado | Sin costo por tiempo ilimitado | Sin costo por tiempo limitado | Sin costo por tiempo limitado |
| Registro | Correo, facebook, google | Correo, facebook, google | Correo, facebook | Correo, facebook, google | Correo | No requerido | Correo, google, microsoft, apple | Correo, facebook, google | Correo | Correo, facebook |
| Sonido | Si | No | Si | No | No | No | Si | Si | No | Si |
| Títulos | Si | No | No | Si | No | Si | Si | Si | Si | Si |
| Visor 360 | Si | No | No | Si | No | No | No | Si | Si | Si |

Fuente: (García, 2021)

1.2.1 Análisis de antecedentes

En Cuba, desde hace varios años se sigue una política de empleo de tecnologías de software libre para el desarrollo de aplicaciones y productos informáticos. En este sentido, se realizó un estudio de los antecedentes afines a la investigación desarrollada en aras de determinar pautas y buenas prácticas a seguir durante el desarrollo de la solución propuesta. A continuación se muestran algunas de las aplicaciones para la realización de recorridos virtuales encontrados:

SAdHEA-Web es un Sistema de Autor para el desarrollo de Hiperentornos de Aprendizajes para la web, desarrollado por el Centro de Estudios de Software y sus Aplicaciones Docentes CESOF-TAD, de la Universidad de Holguín, sede “José de la Luz y Caballero”, con la colaboración del Centro de Estudio de Software Educativos Genus, de la Universidad de Ciencias Pedagógicas “Pepito Tey”, de Las Tunas. Este Sistema de Autor se soporta sobre la base del concepto de Hiperentorno de Aprendizaje empleado por el Ministerio de Educación de la República de Cuba para dotar a todas las educaciones de colecciones de software educativo con un carácter curricular extensivo (Fleitas, 2018).

Sobre este concepto se han desarrollado casi un centenar de productos de esta naturaleza para CD-ROM pero con SAdHEA-WEB se brinda la posibilidad de producir software educativo *on-line*, con herramientas de software libre y multiplataforma.

A partir de cuatro vistas panorámicas exteriores podemos, con esta multimedia, entrar al **Memorial Granma** donde se encuentra, preservado en una gran urna de cristal, el yate Granma, que fue utilizado en 1956 por el grupo de revolucionarios cubanos encabezados por Fidel Castro para realizar el viaje desde el puerto mexicano de Tuxpan, hasta la zona de Los Cayuelos, en la parte suroriental de Cuba. El interior del yate Granma cuenta con siete vistas panorámicas que incluyen la popa y la proa del yate, la cabina del timonel, la cocina y los tres dormitorios. Esta visita virtual cuenta con una selección de entrevistas a expedicionarios del Granma relata episodios de la travesía. Contiene además una serie de animaciones del yate que nos permiten estudiarlo a partir de tres vistas diferentes y hacerlo girar a 360° (Toranzo, 2020).

Habana360 cuenta con un mapa en 360 grados: las zonas famosas de la capital de Cuba, La Habana. Un vídeo panorámico 360 mediante la colocación de una cámara en diferentes vehículos,

incluyendo un clásico almendrón, que viaja entre todas las calles de La Habana y vídeos panorámicos de 360 grados en todos los distritos y en frente de los principales puntos turísticos de la capital cubana (Ojeda, 2022).

cubaen360 son un equipo de profesionales con amplia experiencia en la actividad turística y la promoción del producto. Realizan tour virtuales 360, fotos de alta resolución, fotos de productos, videos promocionales. Para ello cuentan con avanzada tecnología y un dominio de la actividad audiovisual. Están inscrito en el registro del Creador Audiovisual y Cinematográfico de la República de Cuba. Este contiene dentro de sus opciones, visitas a zonas extra hoteleras, hoteles, restaurantes y bares, tiendas y casa de habanos. Todas con visitas panorámicas llevadas a cabo con fotografías de alta resolución y calidad, haciendo de esta una correcta representación de visitas virtuales.

Pese a que todas las aplicaciones revisadas fueron desarrolladas con fines específicos, se les realizó una valoración del cumplimiento de los criterios planteados en la Figura 1.1, con el objetivo de valorar el nivel de completamiento de las mismas. El resultado obtenido se refleja en la Tabla 1.1.

Tabla 1.0.1 Valoración de los criterios a seguir en el desarrollo de aplicaciones web para recorridos virtuales en el contexto nacional. **Fuente:** elaboración propia

| | <i>habana-360.com</i> | <i>cubaen360.com</i> | <i>Memo- rialgranma.cult. cu</i> | <i>casana- tal.cult.cu</i> | <i>sad- hea .uho.edu.cu</i> |
|-----------|-----------------------------|--------------------------------------|--|--------------------------------|---------------------------------|
| Compartir | Facebook, Tweeter y Correo. | Facebook, Whatsapp Tweeter y Correo. | No | No | No |
| Controles | Ocultar/mostrar | Ocultar/mostrar | Zoom in/ zoom out | Ninguno | Ninguno |

| | | | | | |
|-------------|--|-------------------------------|--------------|-------------------------------|--------------|
| Ejemplos | Si | No | No | No | No |
| Hotspot | Url, video, imagen, texto, vínculo e información | Video, imagen, texto, vínculo | Ninguno | Video, imagen, texto, vínculo | Ninguno |
| Idioma | Español | Español | Español | Español | Español |
| Logotipo | Si | Si | Si | Si | No |
| Mapa | Si | Si | No | No | Si |
| Orientación | Si | No | No | No | No |
| Publicación | Con costo | No permitida | No permitida | No permitida | No permitida |
| Registro | Facebook, Google y Correo | Facebook, Google y Correo | Correo | Correo | Correo |
| Sonido | Si | Si | No | No | Si |
| Títulos | Si | No | Si | Si | Si |
| Visor 360 | Si | No | No | No | No |

A modo de resumen, no es posible utilizar ninguna de las aplicaciones web descritas como solución a la problemática de esta investigación; además, los sistemas analizados cumplen con funcionalidades semejantes a las del problema en cuestión por lo que pueden ser utilizados como guía para la implementación de la solución propuesta.

1.3 Metodología de desarrollo de software

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental para desarrollar un software. Para ello, se hace necesario llevar a cabo un estricto control de los flujos de trabajo, de no ser así, es muy posible que una vez finalizado el proyecto aún existan inconformidades o salidas con resultados inesperados. Para evitar este tipo de problemas, es de vital importancia escoger la metodología de desarrollo de software que se ajuste a las condiciones de trabajo de los desarrolladores y a las necesidades del cliente. Las Metodologías de Desarrollo de Software se clasifican en dos grandes grupos (Figuroa, 2010):

Metodologías tradicionales (pesadas)

Las metodologías tradicionales están orientadas al control de los procesos, estableciendo rigurosamente las actividades a desarrollar, herramientas a utilizar y notaciones que se usarán (Figuroa, Roberth, Solís, Cabrera, 2010). Este tipo de metodología, puede manifestarse en diversas modalidades, tales como:

Proceso Racional Unificado (RUP): Tiene como objetivo ordenar y estructurar el desarrollo de software, en la cual se tienen un conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema. Es un proceso basado en los modelos en Cascada y por Componentes, el cual presenta las siguientes características: es dirigido por los casos de uso, es centrado en la arquitectura, así como iterativo e incremental, lo cual es fundamental para el proceso de desarrollo de software (Figuroa, 2010).

Marco de Soluciones de Microsoft (MSF 2): Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. Se centra en los modelos de proceso y de equipo, deja en un segundo plano las elecciones tecnológicas. Compuesta de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño

de Proceso y finalmente e modelo de Aplicación. La Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología (Figuroa, Roberth, Solís, Cabrera, 2010).

Metodologías ligeras (ágiles)

Este tipo de metodologías van orientadas a la interacción con el cliente y el desarrollo incremental del software, la cual muestra versiones parcialmente funcionales del software al cliente en intervalos cortos de tiempo, para que pueda evaluar y sugerir cambios en el producto según se desarrolla (Callejas, Baquero, 2007). Entre los ejemplos representativos de este grupo se encuentran: la Programación Extrema, SCRUM y el Proceso Unificado Ágil (AUP).

Proceso Unificado Ágil (AUP3): Es una versión simplificada del RUP. Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio a través de conceptos que aún se mantienen válidos en RUP y técnicas ágiles. Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva: inicio, elaboración, construcción y transición. En el proceso AUP las disciplinas de: Modelado de Negocio, Requisitos y Análisis y Diseño, se agrupan en una sola, mientras que las restantes (Implementación, Pruebas, Despliegue, Gestión de Configuración, Gestión y Entorno) coinciden con las de RUP (Callejas, Baquero, 2007).

SCRUM: Es una metodología ágil útil para empresas de desarrollo de software orientadas a varios clientes. Consiste en un modelo de asignación de tareas diarias basado en reuniones rápidas y control de la evolución de los procesos. Es muy bueno para llevar un seguimiento de las tareas que se llevan a cabo y saber en qué puntos se ha atascado el equipo. Además, la profundidad de las tareas que se asignan en SCRUM tiende a ser incremental, y esto coincide exactamente con el devenir normal de un proceso de desarrollo (Callejas, Baquero, 2007).

Metodología de desarrollo para la Actividad productiva de la UCI (AUP-UCI): Como parte de un proceso de estandarización llevado a cabo en los centros productivos de la UCI, se aprobó este tipo de metodología. Al no existir una metodología de software universal, se decide hacer una variación de la metodología y las disciplinas de AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI, que queda de la siguiente forma: gestión de la configuración, la planeación y el control del proyecto (Sánchez, 2015).

A partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos (Casos de Uso del Negocio (CUN), Descripción de Procesos del Negocio (DPN) o Modelo Conceptual (MC)) y existen tres formas de encapsular los requisitos (Casos de Uso del Sistema (CUS), Historias de Usuario (HU) o Descripción de Requisitos por Proceso (DRP)), surgen cuatro escenarios en la metodología AUP en su variante UCI para modelar el sistema en los proyectos, manteniendo en dos de ellos el MC, que queda de la siguiente forma(Santana,2015):

- Escenario No 1: proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario No 2: proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No 3: proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No 4: proyectos que no modelen negocio solo pueden modelar el sistema con HU.

Programación Extrema o *Extreme Programming(XP)*: Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y estableciendo una fuerte retroalimentación con el cliente. Es adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Esta metodología, evidencia principios tales como el desarrollo incremental, la participación activa del cliente, el interés en las personas y no en los procesos como elemento principal, y aceptar el cambio y la simplicidad. Tomó su nombre por las prácticas reconocidas en el desarrollo de software y por la participación del cliente en niveles extremos. XP incluye las fases de: planeación, diseño, codificación y pruebas. Fases del ciclo XP (Pressman, 2010).

Planeación o Planificación del proyecto: Entre las acciones que se ejecutan en esta fase se encuentra definir las historias de usuario con el cliente y crear el plan de iteraciones donde se indica en qué iteración serán implementadas las historias de usuarios previamente definidas.

Diseño: Sugiere utilizar diseños sencillos y simples pues de esta manera costarán menos tiempo y esfuerzo desarrollarlos posteriormente.

Codificación: Se definen los estándares de codificación que se utilizarán para facilitar la comprensión del código.

Pruebas: El funcionamiento del sistema se lleva a cabo a través de las historias de usuario definidas en la primera fase, el cual sugiere que se realicen pruebas funcionales que sirven para evaluar las distintas tareas realizadas.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, puesto que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

Justificación de selección de la metodología

Luego de analizar en profundidad las características y particularidades, tanto de los enfoques ágiles como de los tradicionales, se decidió que para el desarrollo de la presente solución se adoptará una metodología ágil, en base a la necesidad de realizar entregas parciales de la solución en el menor tiempo posible, dándole prioridad a las funcionalidades esenciales y se aprovecha la flexibilidad de las tareas asignadas.

Como metodología de desarrollo, fue seleccionada la Programación Extrema (XP). Para ello, se tuvo en cuenta que:

- El equipo de desarrollo es pequeño (2 personas), el cual se encuentra en constante comunicación con el cliente.
- Los requisitos de software pueden cambiar durante el proceso de implementación.
- El software puede ser implementado en un tiempo relativamente corto.
- La comunicación es muy importante, que exista un ambiente de colaboración y comunicación al interior del equipo de desarrollo, así como en la interacción de éste con el cliente. En XP la interacción con el cliente es tan estrecha, que es considerado parte del equipo de desarrollo.

- La simplicidad, desde diseños muy sencillos donde lo relevante es la funcionalidad necesaria que requiere el cliente, hasta la simplificación del código mediante la refactorización del mismo. La programación XP no utiliza sus recursos para la realización de actividades complejas, sólo se desarrolla lo que el cliente demanda, de la una forma sencilla.
- La retroalimentación, ayuda a encaminarlo y darle forma. Ésta se presenta en los dos sentidos, por parte del equipo de trabajo hacia el cliente, con el fin de brindarle información sobre la evolución del sistema, y desde el cliente hacia el equipo en los aportes a la construcción del proyecto.

1.4 Herramientas y tecnologías informáticas para la realización del recorrido virtual

Apoyándose en el método empírico entrevista se conoció que herramientas tecnológicas informáticas se utilizarían para la realización del recorrido virtual por el Jardín Botánico de Matanzas gracias a las demandas específicas realizadas por el cliente.

1.4.1 Interfaz:

HTML, o *HyperText Markup Language*, es el lenguaje utilizado para los documentos web. No es un lenguaje de programación, sino un lenguaje que identifica el significado, el propósito y la estructura del texto dentro de un documento. HTML 5 es la última especificación del lenguaje HTML y representó una gran ruptura con las prácticas de marcado anteriores. El propósito de los profundos cambios en el lenguaje era estandarizar las muchas formas nuevas en que los desarrolladores lo usaban, así como fomentar un conjunto único de mejores prácticas con respecto al desarrollo web. La mayoría de los cambios individuales son el resultado de objetivos amplios en el diseño del lenguaje. Estos objetivos incluyen principalmente:

- Fomentar el marcado semántico (significativo)
- Separar el diseño del contenido
- Promoción de la accesibilidad y la capacidad de respuesta del diseño
- Reducir la superposición entre HTML, CSS y JavaScript
- Compatibilidad con experiencias multimedia enriquecidas y eliminación de la necesidad de complementos como Flash o Java (Schwarz, 2021).

CSS es el lenguaje que define la presentación de una página web. Se utiliza para agregar color, imágenes de fondo y texturas, y para organizar elementos en la página. Sin embargo, CSS hace mucho, además que pintar una imagen bonita. También se utiliza para mejorar la usabilidad de un sitio web (Schwarz, 2021).

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo, no solo sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales (HTML y CSS) hemos cubierto con mucho detalle en otras partes del Área de aprendizaje. (Eich, 2020)

Bootstrap5 es un *framework* CSS de código abierto que favorece el desarrollo web de un modo sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegables, etc. También existe la posibilidad de utilizar extensiones de Javascript adicionales. Fue desarrollado inicialmente por *Twitter* en 2011 y permite crear interfaces de usuario limpias y compatibles con todo tipo de dispositivos. Entre las ventajas que tiene *Bootstrap* es que favorece el diseño *responsive*, el cual se utiliza para mejorar la experiencia de los usuarios en el sitio web y en consecuencia el posicionamiento(Bootstrap, 2020)

React es una biblioteca para construir interfaces de usuario. *React* no es un *framework* — ni siquiera se limita a la web. *React* es utilizado con otras bibliotecas para renderizar en ciertos entornos. Por ejemplo, *React Native* puede usarse para desarrollar aplicaciones móviles; *React 360* permite crear aplicaciones de realidad virtual; además de otras posibilidades. Al desarrollar para la web, los desarrolladores usan *React* en conjunto con *ReactDOM*. *React* y *ReactDOM* son, a menudo, considerados al igual que —y utilizados para resolver los mismo problemas que— otros verdaderos *frameworks* de desarrollo web (Walke, 2022).

Cuando se refiere a *React* como un *framework*, se emplea este significado de manera coloquial. El objetivo principal de *React* es minimizar los errores que ocurren si los desarrolladores construyen

interfaces de usuario. Esto lo hace mediante el uso de componentes — piezas de código lógicas y auto-contenidas que describen una parte de la interfaz del usuario. Estos componentes se pueden juntar para crear una interfaz de usuario completa, y *React* abstrae la mayor parte del trabajo de renderizado, lo que permite al desarrollador enfocarse en el diseño de la interfaz (Walke, 2022)

1.4.2 Back-end:

Node.JS es un entorno que trabaja en tiempo de ejecución, de código abierto, multiplataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript. La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor). Como tal, el entorno omite las APIs de JavaScript específicas del explorador web y añade soporte para APIs de sistema operativo tradicionales que incluyen HTTP y bibliotecas de sistemas de ficheros (Hermina, 2022).

Desde una perspectiva de desarrollo de servidor web, *Node* tiene un gran número de ventajas:

- Gran rendimiento, *Node* ha sido diseñado para optimizar el rendimiento y la escalabilidad en aplicaciones web y es un muy buen complemento para muchos problemas comunes de desarrollo web (ej, aplicaciones web en tiempo real).
- El código está escrito en simple JavaScript, lo que significa que se consume menos tiempo ocupándose de las conmutaciones de contexto entre lenguajes mientras estás escribiendo tanto el código del explorador web como del servidor.
- JavaScript es un lenguaje de programación relativamente nuevo y se beneficia de los avances en diseño de lenguajes si se compara con otros lenguajes de servidor web tradicionales (ej, *Python*, *PHP*, etc.) Muchos otros lenguajes nuevos y populares se compilan/convierten a JavaScript de manera que puedes también usar *CoffeeScript*, *ClosureScript*, *Scala*, *LiveScript*, etc.
- El gestor de paquetes de *Node* (NPM del inglés: *Node Packet Manager*) proporciona acceso a cientos o miles de paquetes reutilizables. Tiene además la mejor en su clase resolución de

dependencias y puede usarse para automatizar la mayor parte de la cadena de herramientas de compilación.

- Es portable, con versiones que funcionan en *Microsoft Windows*, OS X, Linux, Solaris, *FreeBSD*, *OpenBSD*, *WebOS*, y *NonStop OS*. Además, está bien soportado por muchos de los proveedores de hospedaje web, que proporcionan infraestructura específica y documentación para hospedaje de sitios *Node*.
- Tiene un ecosistema y comunidad de desarrolladores de terceros muy activa, con cantidad de desarrolladores deseosos de ayudar. (Hermina, 2022)

Express.js es un *framework* web transigente, escrito en JavaScript y alojado dentro del entorno de ejecución *NodeJS*, es un *framework* web de *Node*, y es la librería subyacente para un gran número de otros *frameworks* web de *Node* populares. Proporciona mecanismos para: (Hermina, 2022)

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Integración con motores de renderización de vistas para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones *middleware* adicional en cualquier punto dentro de la tubería de manejo de la petición.

1.4.3 Diseño y Modelado:

Lenguaje unificado de modelado UML 2.0 (*Unified Modeling Language*) es el estándar industrial de notación de modelado para sistemas orientados a objetos, constituye la plataforma inicial para el desarrollo rápido de aplicaciones. Es un lenguaje gráfico para visualizar, especificar, construir y documentar artefactos de un software. Se utiliza UML para modelar el sistema, ya que se compone de elementos que permiten la representación del software mediante diagramas. Además, sirve de guía a la mayoría de los procesos orientados a objetos, capta la información sobre la estructura estática y el comportamiento dinámico del sistema. (Roger S. Pressman, 2005)

Visual Paradigm 8.0 Constituye una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadoras) que utiliza UML. Soporta el ciclo de vida del desarrollo de software, además de tener la ventaja de ser multiplataforma. Propicia un conjunto de funcionalidades para el desarrollo de programas informáticos, desde la planificación, análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.(Pressman, 2005)

De acuerdo a Roger S. Pressman (2005), las herramientas **CASE** son un complemento de la caja de herramientas del ingeniero del software que le proporciona, la posibilidad de automatizar actividades manuales y de mejorar su visión general de la ingeniería. Al igual que las herramientas de ingeniería y diseño asistidos por computadora que utilizan de otras disciplinas, las herramientas CASE ayudan a asegurar que la calidad sea algo diseñado antes de llegar a construir el producto. Estas herramientas ayudan a los gestores y practicantes de la ingeniería del software en todas las actividades asociadas a los procesos de software. Automatizan las actividades de gestión de proyectos, gestionan todos los productos de los trabajos elaborados a través del proceso y ayudan a los ingenieros en el trabajo de análisis, diseño y codificación. Las herramientas CASE se pueden integrar dentro de un entorno sofisticado (Leyva et al., 2017).

1.4.4 Entorno de Desarrollo Integrado

IDE Visual Studio Code versión 1.64.2 es un editor de código fuente ligera pero potente que se ejecuta en escritorio y está disponible para *Windows*, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y 14 Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity)(Pasero, 2022)

1.5 Conclusiones del capítulo

Luego de estudiar los principales conceptos relacionados con el objeto de estudio, así como los homólogos y antecedentes del mismo; y de realizar una descripción de la metodología y tecnologías de desarrollo de software a considerar, se concluye que:

1. Los sistemas existentes no resuelven la situación problemática planteada. El estudio realizado del sistema de homólogos, evidenció la necesidad de implementar un Sistema informático para la realización del recorrido virtual realización del recorrido virtual por el Jardín Botánico de Matanzas, como una propuesta de solución acorde a las necesidades del cliente.
 2. El empleo de métodos y técnicas de metodología de la investigación permitió adquirir y fomentar los conocimientos necesarios para entender los procesos de realización del recorrido virtual.
 3. La combinación de las tecnologías antes mencionadas y la metodología XP, es la apropiada para desarrollar la aplicación para la realización de recorrido virtual por el Jardín Botánico de Matanzas acorde a las condiciones de infraestructura tecnológica de la Universidad de Matanzas, donde se encuentra enclavado dicho jardín..
 4. El uso de últimas versiones de herramientas componentes y estándares le brinda a la aplicación un mayor tiempo de vida útil.
 5. Se demuestra la necesidad de la utilización de software libre como una tendencia a asumir.
- En sentido general se ha contribuido a la mejor comprensión del objeto de estudio y se han establecido las bases para las siguientes fases de la investigación.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN DEL DESARROLLO DE RECORRIDOS VIRTUALES A TRAVÉS DE APLICACIONES WEB.

En el actual capítulo se describe el procedimiento y los elementos que se tuvieron en cuenta para el desarrollo del recorrido virtual a través de una aplicación web. Se presentan los conceptos asociados al modelo conceptual y la representación formal del mismo. Se identifican los requisitos no funcionales y funcionales. Además, se emplea la técnica de descripción de requisitos, acorde con la metodología de desarrollo de software seleccionada.

2.1 Modelado del negocio

Todo proceso de negocio es legislativamente dependiente de las reglas de negocio predefinidas para el diseño del proceso. Su ejecución se caracteriza por el empleo de datos manipulados por un conjunto de actividades en las que participan agentes o actores acorde a un flujo de trabajo (Pressman, 2007).

El propósito principal de esta fase en el proceso de desarrollo de software es realizar una exploración del dominio del problema, con énfasis en los conceptos y las relaciones que existen entre estos. A continuación, se detalla una breve descripción del negocio y el modelo de dominio generado para mayor entendimiento del mismo.

En el modelo de negocio se identifica como proceso de negocio mostrar en el correcto orden y secuencia las imágenes preprocesadas de la base de datos

Como se ha explicado anteriormente el proceso de creación de un recorrido virtual comienza con la obtención de los panoramas (fotografías equirrectangulares), continua con la edición y corrección de estas imágenes y concluye con el ordenamiento y secuenciación de los panoramas para crear el recorrido. Es importante destacar que las dos primeras etapas; obtención de información y preprocesamiento de imágenes, quedan fuera del alcance de este documento.

El proceso de mostrar las imágenes de la base de datos comienza en el momento que un usuario entra al tour virtual y comienzan a mostrarse las imágenes correspondientes en el orden preciso de tal manera que sea una correcta representación de la localidad que se muestra en este caso el

Jardín Botánico de Matanzas. La secuencia en que se mostraran estas fotos dependerá de la ruta seguida por el usuario.

2.1.1 Descripción del proceso de realización del recorrido virtual por el Jardín botánico de Matanzas

El sistema tendrá la capacidad de almacenar toda la información espacial referente al Jardín, que permitirán la realización del recorrido virtual que brinda además la posibilidad de visualizarlos en las diferentes capas contextuales. Se espera poner a disposición de la comunidad botánica, científica y universitaria, así como de visitantes externos, un sistema de percepción genérico para el recorrido virtual extensible. Se concibe el aporte social en la contribución de la estrategia y el sistema informático para obtener mejoras en la atención a los usuarios e incrementar su satisfacción con el sistema desarrollado.

Deberá poseer una arquitectura adaptable para estandarizarla a otros entornos, puesto que estará hospedado en servidor web de la Universidad de Matanzas. Además, requerirá solo de la actualización de la información que se muestra, proceso totalmente visible desde las interfaces del propio sistema, a menos que se desee a futuro incrementar sus funcionalidades. El sistema permitirá:

- Observar las imágenes panorámicas cilíndricas referentes a los espacios del recorrido virtual.
- Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido.
- Ampliar y disminuir un área de la imagen panorámica perteneciente al espacio visualizado. Con ello, se consigue mayor exactitud en la visualización de los objetos pequeños que se encuentran en la escena.
- Mostrar la descripción correspondiente al espacio seleccionado, lo que le facilita a los usuarios conocer en detalle la instalación.
- Desplazarse en una imagen panorámica de forma horizontal y vertical, lo que proporciona a los usuarios efecto de realismo.
- Mostrar una imagen ampliada de la zona seleccionada correspondiente a un espacio del recorrido virtual.

Roles del negocio:

Tabla 2.0.1 Roles del negocio. **Fuente:** elaboración propia

| Nombre | Descripción |
|----------|--|
| Usuarios | Es quien realiza el recorrido en la página |

2.2 Requisitos del recorrido virtual

Para dar solución a la problemática planteada se propone el desarrollo de un recorrido virtual. Su implementación proporciona una herramienta capaz de garantizar el recorrido virtual por el Jardín Botánico de Matanzas.

2.2.1 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, la forma en que debe reaccionar ante ciertas entradas y cómo se debe comportar en situaciones particulares (Sommerville, 2011).

Un requisito funcional es aquel que define una funcionalidad del sistema software o sus componentes. Una funcionalidad es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales describen cómo se debe comportar el sistema si se cumplen ciertas condiciones. A continuación se detallan cada uno de los requerimientos que deben ser implementados en la solución.

En el análisis realizado, se identificaron los siguientes requisitos funcionales (Tabla 2.2):

Tabla 2.0.2 Requisitos funcionales. **Fuente:** elaboración propia

| Requisito funcional | Complejidad | Prioridad | Descripción |
|---------------------|-------------|-----------|---|
| RF 1 | Media | Baja | Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. |

| | | | |
|------|-------|------|---|
| RF 2 | Alta | Alta | Observar las imágenes panorámicas cilíndricas referentes a las salas de un recorrido virtual. |
| RF 3 | Media | Baja | Mostrar la descripción correspondiente a la sala, lo que le facilita a los usuarios conocer en detalle la instalación. |
| RF 4 | Alta | Alta | Ampliar y disminuir un área de la imagen panorámica perteneciente a la sala visualizada. |
| RF 5 | Alta | Alta | Girar en una imagen panorámica de forma horizontal en 360 grados, brindándoles a los usuarios mayor efecto de realismo. |
| RF 6 | Alta | Alta | Girar en una imagen panorámica de forma vertical 180 grados, brindándoles a los |

| | | | |
|--|--|--|------------------------------------|
| | | | usuarios mayor efecto de realismo. |
|--|--|--|------------------------------------|

2.2.2 Requisitos no funcionales

Los requisitos no funcionales constituyen restricciones de los servicios que brindan el sistema o el componente. No se refiere a lo que debe realizar el componente, sino a propiedades físicas. Incluyen aspectos del diseño, del proceso de desarrollo, del rendimiento de la solución informática, entre otras. Comprenden las características del componente que el usuario o cliente puede valorar como fiabilidad, seguridad, portabilidad, diseño de interfaces, capacidad de almacenamiento, entre otras (Sommerville, 2011).

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que conoce y especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Son restricciones o condiciones que impone el cliente al software que necesita.

Usabilidad:

- RNF 1: Los mensajes para interactuar con los usuarios y los de error deben ser lo suficientemente informativos, en idioma español y no deben revelar información interna. El sistema propuesto podrá ser usado por usuarios sin conocimiento de informática, ya que estará estructurado de forma sencilla.
- RNF 2. Tanto los textos de las interfaces de usuario, como los mensajes de información, error o alerta, deben encontrarse en un lenguaje claro, de fácil entendimiento para el usuario.

Rendimiento:

- RNF 3: El tiempo de respuesta del sistema no debe exceder de tres segundos.

Interfaz:

- RNF 4: El diseño será sencillo, amigable e intuitivo; que combine correctamente los colores, tipo y tamaño de letra.

Seguridad:

- RNF 5: La información solo puede ser modificados por aquellos usuarios autorizados.
- RNF 6: La seguridad estará regida por un mecanismo de control de acceso basado en roles, convenientemente asignados a los usuarios del sistema al momento de su creación.

Software:

- RNF 7. El dispositivo móvil debe tener sistema operativo Android con versión 5.0 o superior o en caso de ser una computadora personal los sistemas operativos: GNU/Linux o Microsoft *Windows*.
- RNF 8. El dispositivo móvil o la computadora personal debe tener una versión de navegador actualizada; *Mozilla Firefox 57.0.3*, *Netscape 9.0.0.6*, *OmniWeb 5.11.2*, *Microsoft Edge 44.19592.1001.0*, *Opera 50.0*, *Chrome 41.0.2272.101* o superiores.

Hardware:

- RNF 9. El sistema deberá desplegarse en una PC, que cuente con los requerimientos mínimos de un microprocesador Intel Core i3-2100 3.10 GHz, memoria RAM: 4 GB y un disco duro con al menos 50 GB de capacidad disponible. Los dispositivos móviles necesitan tener implementado el estándar IEEE 802.11. En la red donde se despliegue el producto se debe disponer de un punto de acceso inalámbrico.

Escalabilidad:

- RNF 10 Añadir nuevas mejoras al sistema a partir de la implementación de nuevas funcionalidades

Fiabilidad:

- RNF 11. La aplicación debe ejecutarse durante el periodo de tiempo determinado por los usuarios sin que se produzcan fallos.

2.2.3 Historias de usuario

La especificación de requisitos establece la base para el acuerdo entre usuarios y desarrolladores de software, lo cual deja definido el comportamiento deseado del producto (IEEE, 2004).

En el entorno de XP, las historias de usuario (HU) constituyen el artefacto utilizado para describir las funcionalidades del sistema. Contienen una breve descripción del comportamiento del sistema

desde la perspectiva del usuario y representan un medio de comunicación entre el mismo y el equipo de desarrollo. A continuación, se presentan las historias de usuario correspondientes a dos de los requisitos funcionales identificados. Las HU se clasifican según:

La prioridad en el negocio

- Alta: se le otorga a las HU que resultan funcionalidades fundamentales en el desarrollo del sistema, a las que el cliente define como principales para el control integral del sistema.
- Media: se le otorga a las HU que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se desarrolla.
- Baja: se le otorga a las HU que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el sistema en desarrollo.

El riesgo en su desarrollo

- Alta: En la implementación de las HU se considera la posible existencia de errores que conlleven a la inoperatividad del código
- Media: Se pueden aparecer errores en la implementación de la HU que puedan retrasar la entrega de la versión.
- Baja: Se pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

Las HU son representadas mediante tablas divididas por las siguientes secciones:

- Número: esta sección representa el número, incremental en el tiempo, de la HU que se describe.
- Nombre de HU: identifica la HU que se describe entre los desarrolladores y el cliente.
- Usuario: los programadores responsables de la HU.
- Iteración asignada: número de la iteración donde va a desarrollarse la HU.
- Prioridad en negocio: se le otorga una prioridad (Alta, Media, Baja) a las HU de acuerdo a la necesidad de desarrollo.

- **Riesgo en desarrollo:** se le otorga una medida de (Alto, Medio, Bajo), a la ocurrencia de errores en el proceso de desarrollo de la HU.
- **Puntos estimados:** es el tiempo estimado en semanas que se demorará el desarrollo de la HU.
- **Descripción:** breve descripción de la HU.
- **Observaciones:** señalamiento o advertencia del sistema.

En una entrega se puede desarrollar una o más HU, esto depende solo del tiempo que demore la implementación de cada una de las mismas que son utilizadas para estimar el tiempo que el equipo de desarrollo tomará para realizar las entregas, deben poder ser programadas en un tiempo entre una y dos semanas. Si la estimación es superior a dos semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.

Tabla 2.0.3 Historia de usuario. Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. Fuente: elaboración propia

| HISTORIA DE USUARIO | |
|---|-----------------------------------|
| Numero: HU-01 | Usuario: Usuario |
| Nombre de la historia: Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Programador responsable: Alejandro David Santana Rodríguez | |
| Descripción: El software debe permitir mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. | |
| Observaciones: | |

Tabla 2.0.4 Historia de usuario. Observar las imágenes panorámicas cilíndricas referentes a las salas de un recorrido virtual. **Fuente:** elaboración propia

| HISTORIA DE USUARIO | |
|---|-----------------------------------|
| Numero: HU-02 | Usuario: Usuario |
| Nombre de la historia: Observar las imágenes panorámicas cilíndricas referentes a las salas de un recorrido virtual. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 3 | Iteración asignada: 3 |
| Programador responsable: Alejandro David Santana Rodríguez | |
| Descripción: El software debe permitir observar las imágenes panorámicas cilíndricas referentes a las salas de un recorrido virtual. | |
| Observaciones: | |

Plan de iteraciones

Una vez identificados los requisitos y descritas las historias de usuario correspondientes, el equipo de desarrollo realiza una estimación del esfuerzo que supone requerirá la implementación de cada una, mientras el cliente define la prioridad en función del valor para el negocio. Finalmente, se llega a un acuerdo sobre el orden de implementación y el contenido de las entregas y se apuesta por enfrentar las historias de gran valor y riesgo lo antes posible (Letelier y Penadé, 2003).

Las metodologías ágiles han cambiado muchos de los conceptos asociados al desarrollo de software, entre ellos la forma de realizar estimaciones. Uno de los métodos utilizados para la estimación, al trabajar con metodologías ágiles, son los puntos de historia, concebidos para dimensionar y relacionar la complejidad de las historias de usuario con respecto a otras.

La técnica de Planeación *Poker* (*Planning Poker*) permitió evaluar de forma rápida y eficaz la dificultad relativa de las historias de usuario especificadas. Para la aplicación de esta técnica cada miembro del equipo de desarrollo propone un valor de estimación para cada historia de usuario, si

estos no coinciden se exponen los argumentos de las estimaciones alejadas y se repite el proceso hasta lograr un consenso (Elssamadisy, 2008).

El ciclo de desarrollo de software guiado por XP se caracteriza por ser iterativo e incremental. Los niveles de prioridad para el negocio y el esfuerzo estimado de implementación para cada una de las historias de usuario especificadas, hicieron posible la planificación de la implementación del sistema, de esta manera quedan definidas cuáles historias de usuario serían implementadas en cada iteración. A continuación, se muestra el plan de iteraciones:

Tabla 2.0.5 Plan de iteraciones. **Fuente:** elaboración propia

| No. | Nombre de historia de usuario | Prioridad | Esfuerzo estimado | |
|-----|---|-----------|-------------------|------|
| 1 | Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. | Alta | 2 | 24cc |
| | Observar las imágenes panorámicas cilíndricas referentes a las salas de un recorrido virtual. | Alta | 4 | |
| | Mostrar la descripción correspondiente a la sala | Alta | 3 | |
| | Ampliar y disminuir un área de la imagen panorámica perteneciente a la sala visualizada. | Alta | 5 | |

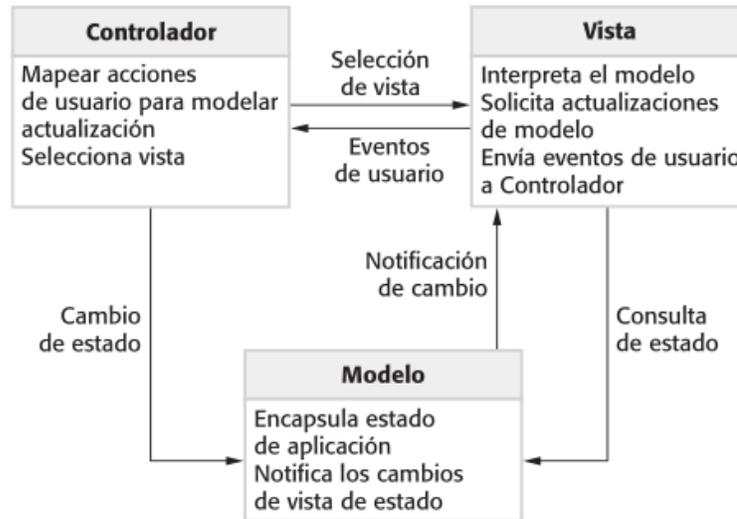
| | | | | |
|--|--|------|---|--|
| | Girar en una imagen panorámica de forma horizontal | Alta | 5 | |
| | Girar en una imagen panorámica de forma horizontal | Alta | 5 | |

2.3 Análisis y diseño.

Para que un software sea desarrollado con calidad es necesario establecer desde el inicio una arquitectura que describa sus principios fundamentales, se garantiza así robustez y escalabilidad. La arquitectura de software define la estructura del sistema, constituida por componentes con funciones específicas que interactúan entre sí

Para el diseño arquitectónico del sistema se hará uso de una versión del patrón Modelo Vista Controlador (MVC) (Figura 2.1), el cual garantiza la organización del código fuente de la aplicación, dividiéndola en tres componentes fundamentales: el modelado del dominio, la presentación y las clases controladoras. En este patrón, el "Modelo" hace referencia al acceso a la capa de datos, la "Vista" se refiere a la parte del sistema que selecciona qué mostrar y cómo mostrarlo, y el "Controlador" implica la parte del sistema que decide qué vista usar, dependiendo de la entrada del usuario, accediendo al modelo si es necesario.

Figura 2.1 Organización del patrón arquitectónico MVC



Fuente: (González & Romero, 2012)

Express.js sigue el patrón MVC tan al pie de la letra que puede ser llamado un *framework* MVC. Someramente, la M, V y C se separan en *Express.js* de la siguiente manera:

- M, la porción de acceso a los datos, es manejada por la capa de datos de *Express*, la cual describiremos en este capítulo. Maneja las imágenes panorámicas (Clase: *Data*)
- V, la porción que selecciona qué datos mostrar y cómo mostrarlos, es manejada por la vista y las plantillas donde se visualizan las imágenes procesadas en la clase controladora (Clase: *View*)
- C, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el *framework* mismo siguiendo tu *URLconf* y se llama a la función apropiada de *React* para la URL obtenida. Determina las secuencias de imágenes a seguir y el orden en que deben mostrarse procesando estas imágenes (Clase: *Controller*)

2.3.1 Modelo de diseño

Uno de los principios fundamentales de la metodología XP es la simplicidad, en la etapa de diseño se propone el empleo de tarjetas CRC (Clase Responsabilidad Colaboración), en lugar de diagramas de clases para la descripción del sistema en notación UML.

Las características sobresalientes de las tarjetas CRC son su sencillez y adaptabilidad (Casas y H, 2012). Las tarjetas CRC son fichas que representan cada una de las clases del sistema, en ellas se describen brevemente las responsabilidades de la clase, donde se muestra un listado de las clases con las que colabora para ejecutar dichas responsabilidades. Su utilización en el diseño potencia el uso de patrones de asignación de responsabilidades (GRASP)(*Larman, 2000*).

A continuación, se muestran las tarjetas CRC correspondientes a las clases presentes en la propuesta de solución (Tabla 2.6).

Tabla 2.0.6 Tarjeta CRC para la Clase Controladora. **Fuente:** Elaboración propia

| Tarjeta CRC | |
|--|------------------------|
| Clase: <i>Controller</i> | |
| Responsabilidades: | Colaboraciones: |
| Es la clase encargada de: <ul style="list-style-type: none">• Generar las imágenes.• Determina la secuencia y a seguir para mostrar las imágenes. | <i>View</i> |

2.4 Patrones de Diseño

2.4.1 Patrones GRASP:

Como patrones generales de asignación de responsabilidades (*General Responsibility Assignment Software Patterns, GRASP*). Los patrones GRASP describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones. Su uso posibilita una correcta selección, para cada funcionalidad, de la clase en que debe implementarse. Se utilizarán los siguientes patrones de asignación de responsabilidades:

Experto: con el fin de asignar las responsabilidades a la clase experta en la información (Larman, 2000). El patrón resulta de utilidad en las clases del modelo, las cuales contienen toda la información relacionada con los objetos persistentes que representan.

Este patrón es evidenciado en la definición de las clases de acuerdo a las funcionalidades que deben realizar a partir de la información manejada dentro del componente, como por ejemplo las clases controladoras.

Creador: guía la asignación de responsabilidades relacionadas con la creación de los objetos (Larman, 2000). El *framework* instancia las clases encargadas de cada una de las funcionalidades accesibles desde la interfaz principal.

Una de las consecuencias de usar este patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

Controlador: Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto permite aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en un mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento.

Este patrón es evidenciado en la especificación de la clase controladora. El método registrar se encarga de cargar previamente los datos que serán mostrados en la vista del formulario una vez que el mismo la llame.

Bajo acoplamiento: garantiza la asignación de responsabilidades de manera que el acoplamiento entre las clases permanezca bajo (Larman, 2000). El acoplamiento mide la fuerza con que una clase está conectada a otra. La arquitectura del sistema es la máxima evidencia de la utilización de este patrón.

Este patrón es evidenciado claramente en la utilización del método “*getImage*” de la clase entidad “*Image*”. Para obtener las solicitudes es necesario acceder al *endpoint* del API.

Alta cohesión: con el propósito de asignar las responsabilidades a las clases de manera que permanezcan lo más cohesionadas posible, se asegura de esta forma que ninguna realice un trabajo excesivo gracias a la colaboración con otras clases (Larman, 2000). Su utilización puede verse, de conjunto con el bajo acoplamiento, en la arquitectura del sistema.

Este patrón es evidenciado en la estrecha relación que guardan los nombre de las funcionalidades y variables.

La utilización de estos patrones resultó en un diseño sencillo, pensado para la creación de un repositorio robusto, fácil de entender, mantener y ampliar; se aumenta por lo tanto la capacidad de reutilización de sus componentes y lo cual permite conservar el encapsulamiento de la información.

2.4.2 Patrones GOF:

Los patrones GOF describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases y la formación de estructuras de mayor complejidad. Permiten crear grupos de objetos para facilitar la realización de tareas complejas. Existen tres tipos de patrones GOF: de creación, estructurales y de comportamiento. Los de creación se encargan de la creación de instancias de las clases. Los estructurales son los que plantean las relaciones y combinaciones entre clases, mientras que los de comportamiento tratan la interacción y cooperación entre las mismas (Sutherland, 2018). Los patrones a utilizar serían:

Sencillo (*Singleton*): Es un patrón de creación. Asegura que solo se cree una instancia de la clase y provee un punto global para acceder a ella. Este patrón es muy útil en el establecimiento de la conexión con la base de datos. Para ello se crea una propiedad compartida para todas las clases de la aplicación y, antes de asignársele la instancia que da lugar a la conexión, se comprueba que la propiedad no haya sido asignada antes (que tenga valor *null*). La propuesta de solución utiliza este patrón al cargar las siguientes imágenes.

Plantilla (*Template*): define el esqueleto de programa de un algoritmo en un método, llamado método de plantilla, el cual difiere algunos pasos a las subclases. Se utilizó vistas genéricas

basadas en clases donde los pasos de un algoritmo se pueden redefinir mediante subclases sin cambiar la estructura del algoritmo.

Comando (*Command*): También es un patrón de comportamiento que encapsula una operación en un objeto. De esta manera, la vista puede ejecutar los comandos independientemente de su implementación, de la cual solo tiene conocimiento el modelo de vista. Muchos de los métodos implementados en la propuesta de solución utilizan el patrón Acción para reducir el número de funciones que son necesarias para su funcionamiento, pasando la operación como un parámetro, junto con los argumentos. De hecho, el propio sistema de utiliza este patrón, por lo que los módulos no tienen que definir cada uno, sino más bien sólo los que se necesiten implementar.

2.5 Conclusiones del capítulo

Después de lo expuesto en los epígrafes anteriores puede concluirse que:

1. Se dispone de un equipo de trabajo capaz de solucionar el problema existente.
2. Se profundizó en la lógica del sistema y el funcionamiento de cada uno de sus procesos, necesario para una visión general del problema que se enfrenta y pensar desde el inicio en soluciones eficaces.
3. La planificación se realizó en consecuencia con los casos de uso teniendo en cuenta la prioridad y el riesgo de desarrollo.
4. La construcción conjunta con el cliente de las HU permitieron describir los aspectos principales a tener en cuenta para el desarrollo de la solución.
5. La obtención de las tarjetas CRC permitió definir la responsabilidad y colaboración entre las clases, lo que provee mayor adaptabilidad a la solución propuesta.
6. La implementación de patrones de diseño y arquitectura contribuyó a la obtención de un producto final de mayor calidad.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA SOLUCIÓN PROPUESTA PARA EL DESARROLLO DE RECORRIDOS VIRTUALES A TRAVÉS DE APLICACIONES WEB.

En este capítulo se describe la validación de las variables de la investigación, la fase de implementación y posteriormente las pruebas al sistema desarrollado.

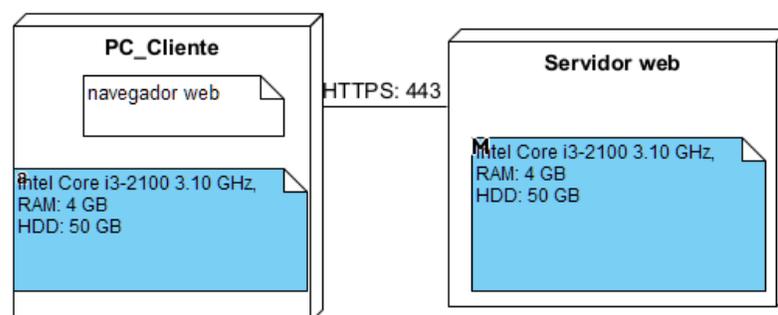
3.1 Implementación de la propuesta de solución

El modelo de implementación identifica los componentes físicos de la implementación para que puedan comprenderse y gestionarse mejor y el modelo de implementación define las principales unidades de desarrollo alrededor de las cuales se organizan los equipos.

3.1.1 Diagrama de Despliegue

Un diagrama de despliegue de UML es un diagrama que muestra, entre otras cosas, el hardware que se utilizará para implementar el sistema y los vínculos entre diferentes elementos del mismo, así como las rutas de comunicación entre ellos. Los diagramas de implementación son un tipo de diagrama de estructura que se utiliza para modelar los aspectos físicos de un sistema orientado a objetos. A menudo se utilizan para modelar la vista de implementación estática de un sistema (topología del hardware)(Pressman, 2005)

Figura 3.1 Diagrama de Despliegue



Fuente: Elaboración propia

3.1.2 Interfaz gráfica de usuario

Para el diseño de las interfaces gráficas de usuarios se aplicaron las Reglas de Oro (Pressman, 2005):

- Lograr que la interfaz sea consistente: Permitir que el usuario incluya la tarea actual en el contexto que tenga algún significado.
- Reducir la carga en memoria del usuario: Reducir la demanda de memoria a corto plazo, definir valores por defecto que tengan significado, definir accesos directos intuitivos, el formato visual de la interfaz debe basarse en la metáfora tomada de la realidad, desglosar la información de manera progresiva.
- Dar control al usuario: Definir los modos de interacción de forma que el usuario no realice acciones innecesarias o indeseables, proporcionar una interfaz comprensible, incluir opciones de interrumpir y deshacer la interacción del usuario, ocultar al usuario ocasional los elementos técnicos internos, diseñar interacción directa con los objetos que aparecen en pantalla

Las pantallas principales de la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas son el resultado visual donde se define la apariencia de cada una de las páginas del portal web, lo que ubica en ellas todos los elementos que las componen.

Figura 3.2 Pantalla de inicio



Fuente: www.umcc.cu/jardinbotanico

Figura 3.3 Pantalla de recorrido virtual



Fuente: www.umcc.cu/jardinbotanico

Los estándares y estilos de codificación empleados en la solución propuesta, se describen a continuación:

- **Métodos:** Los métodos fueron agrupados por funcionalidad, más que por visión o accesibilidad. Por ejemplo, un método de clase privado puede estar entre dos métodos públicos de instancia. El objetivo es hacer el código más legible y comprensible.
- **Indentación:** Se emplearon cuatro espacios como unidad de indentación. La construcción exacta de la indentación (espacios en blanco contra tabuladores) no se especifica. Los tabuladores deben ser exactamente cada ocho espacios (no cuatro).
- **Longitud de línea:** Se evitaron líneas de código de más de 80 caracteres, ya que no son manejadas bien por muchas terminales y herramientas.
- **Comentarios:** Se emplearon comentarios para dar descripciones de código y facilitar información adicional que no es legible en el código mismo. Los comentarios contienen sólo información que es relevante para la lectura y entendimiento del código.
- **Declaraciones:** Se programó siguiendo la estructura de una declaración por línea, ya que facilita los comentarios.
- **Inicialización:** Las variables locales fueron inicializadas donde se declaran.
- **Espacios en blanco:** se emplearon espacios en blanco entre los operadores para lograr una mayor legibilidad en el código.
- **Líneas en blanco:** se dejó una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.
- **Constantes:** se declararon las constantes con todas sus letras en mayúscula.
- **Sentencias:** La llave de apertura aparece al final de la línea que comienza la sentencia compuesta; la llave de cierre empieza una nueva línea y es indentada al mismo nivel que el principio de la sentencia compuesta. Las llaves se usaron en todas las sentencias, incluso las simples, cuando forman parte de una estructura de control, como en las sentencias *if-else* o *for*.

En este trabajo se realizó una revisión cuantitativa de diversas plataformas de software disponibles en internet para la creación de un RV360, considerando que en función de la disponibilidad de dispositivos de diversa gama tanto para la obtención de fotografías panorámicas, como para la visualización de contenido virtual, la publicación de un RV360 puede ser un mecanismo mejor utilizado en el ámbito turístico, cultural, ambiental y educativo, que permita acercar virtualmente a múltiples usuarios a un sitio en particular proporcionándoles una experiencia inmersiva.

3.2 Validación de requisitos

La validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proyecto y el producto (Pressman, 2005)

Esta fase permite mitigar el riesgo de implementar una mala especificación, con el costo que eso conlleva.

Los parámetros a validar en los requisitos son:

Validez: No basta con preguntar a un usuario, todos los potenciales usuarios pueden tener puntos de vista distintos y necesitar otros requisitos.

Consistencia: No debe haber contradicciones entre unos requisitos y otros.

Compleitud: Deben estar todos los requisitos. Esto es imposible en un desarrollo iterativo, pero, al menos, deben estar disponibles todos los requisitos de la iteración en curso.

Realismo: Se pueden implementar con la tecnología actual.

Verificabilidad: Tiene que existir alguna forma de comprobar que cada requisito se cumple.

3.2.1 Técnicas de validación de requisitos

Existen varias técnicas bien definidas para la validación de requisitos, ellas son:

Revisiones de verificabilidad: Consiste en la lectura y corrección de la documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida.

Auditorías: Consiste en un chequeo de los resultados.

Matrices de trazabilidad: Consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo que objetivos cubre cada requisito para detectar inconsistencias u objetivos no cubiertos.

Prototipos: Muestran en imágenes la interfaz esperada de las funcionalidades descritas. (Pressman, 2005)

3.2.2 Resultados al aplicar técnicas de revisión

- 1) Se verificó el cumplimiento de los requisitos obteniéndose los siguientes resultados:
 - Se añadieron dos nuevos requisitos.
 - Se mejoró la descripción de algunos requisitos que se encontraban poco detallados.
 - Se detectaron y corrigieron errores gramáticos y ortográficos en la redacción de los requisitos.
 - Se detectaron y corrigieron errores gramáticos y ortográficos en la redacción en las historias de usuario.
- 2) Se pudo verificar que se cumple el 100% los requisitos.
- 3) Mediante la técnica de auditoria se fueron señalando al final de cada iteración que los requisitos fueron implementados correctamente, permitiendo que se identificara el trabajo restante de la iteración anterior.

3.3 Pruebas

En la fase de desarrollo de software es posible que se cometan errores; se hace necesario realizar una validación constante de la calidad y la satisfacción de las necesidades del cliente final para mitigar estos errores. Con el objetivo de comprobar la seguridad, el funcionamiento y la carga y estrés del portal web se diseñan pruebas que permiten validar el cumplimiento de los requisitos.

Las pruebas en el proceso de desarrollo de software representan un conjunto de actividades que se pueden llevar a cabo sistemáticamente. Estas persiguen como objetivo detectar los errores y fallos que se cometan durante el desarrollo, lo cual contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones.

XP enfatiza en la realización de un sin número de pruebas a lo largo del proyecto, con el fin de asegurar en todo momento la realización de lo planteado en el diseño. En este proceso no sólo participa el equipo de desarrollo, también es importante los aportes del cliente, sobre todo en las pruebas de aceptación (Pressman, 2007).

3.3.1 Pruebas funcionales

Las pruebas funcionales se centran en comprobar el correcto funcionamiento del sistema y en las respuestas del sistema de acuerdo con los datos de entrada y sus resultados en los datos de salida. Estos datos se definen generalmente en los casos de prueba que se crean antes de su inicio. Al conocer las funciones específicas que se le asignaron a una aplicación para su realización, pueden llevarse a cabo casos de pruebas que demuestren que cada función es completamente operativa mientras que al mismo tiempo se buscan errores en cada función (Pressman, 2005) .

Con el objetivo de realizar este tipo de pruebas al sistema, se diseñan y aplican seis casos de prueba que abarcan todos los requisitos funcionales del sistema. A continuación se muestra el caso de prueba: Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido.

Tabla 3.0.1 Caso de prueba. Mostrar la información descriptiva del recorrido virtual perteneciente al lugar escogido. Fuente: Elaboración propia

| Número | Campo | Requerido | Clasificación | Descripción |
|--------|--------|-----------|-----------------|--|
| 1 | Imagen | Sí | Campo de imagen | Se deben observar todos los elementos requeridos del recorrido |

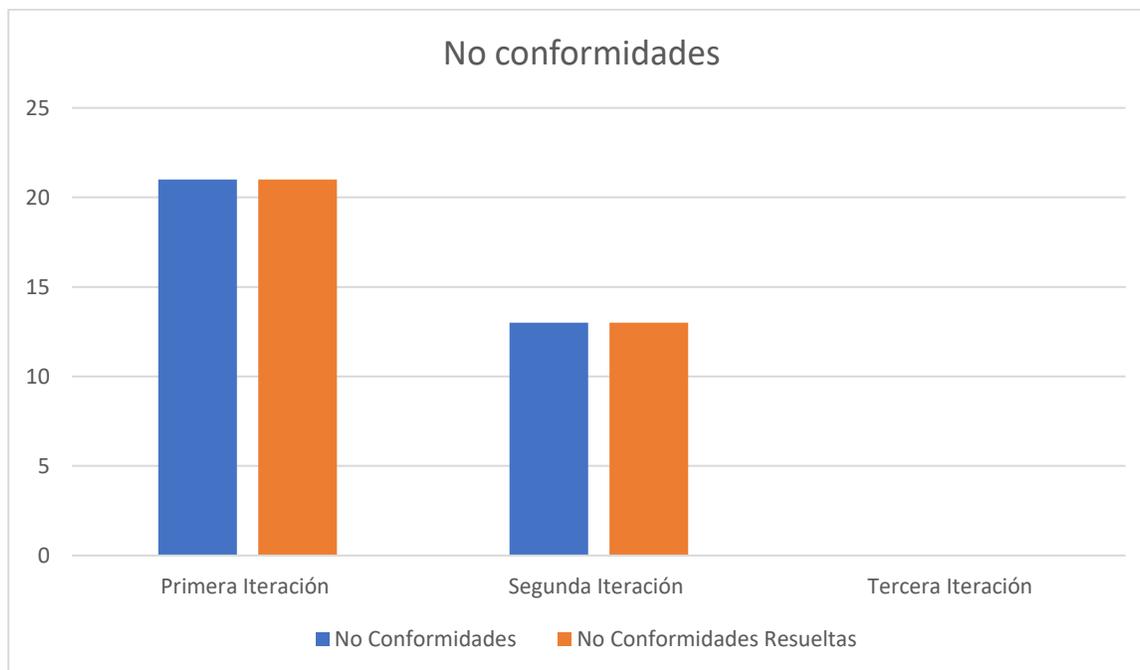
Resultados de las pruebas funcionales

Se realizaron tres iteraciones y se aplicaron los casos de pruebas correspondientes a las funcionalidades del Sistema informático para la realización del recorrido virtual por el Jardín Botánico de Matanzas. En la primera iteración se obtuvieron 21 no conformidades. De ellas, 4 corresponden a opciones que no funcionaban. Se identificaron cuatro no conformidades correspondientes a campos que no se encontraban validados (usuario y contraseña). Otras 7

correspondían al formato de visualización de contenidos del portal. Correspondían 6 no conformidades entre errores de idioma y errores de ortografía en las páginas de administración y en la barra lateral (*sidebar*) del portal. Se corrigieron los problemas referentes a la visualización, los textos en idioma inglés y otros textos con errores ortográficos, además se solucionaron los errores de opciones y los de validación.

Las 21 no conformidades fueron resueltas al culminar la iteración. En la segunda iteración fueron detectadas un total de 13 no conformidades, como son: existían errores en el pie de página, en la página principal existían capas que se sobreponían, no se cargaban todas las imágenes, existían errores de idioma en la página principal y los campos de imagen no estaban validados. Se corrigieron los errores de interfaz, se solucionaron los errores de validación y los que existían en el pie de página. Las 13 no conformidades fueron resueltas al culminar la iteración y las soluciones aportadas no generaron nuevas inconformidades. Se realizó una tercera iteración donde no se detectaron no conformidades.

Figura 3.4 No conformidades



Fuente: Elaboración propia

En la figura se muestra una gráfica que ilustra las 3 iteraciones realizadas con el total de 21, 13 y cero no conformidades encontradas respectivamente. Todas las no conformidades fueron resueltas en las pruebas funcionales por tanto asegura que el sistema está validado y cumple con los requisitos especificados.

3.3.2 Pruebas de rendimiento

Con el objetivo de comprobar el rendimiento del sistema se aplican pruebas de carga y estrés. Mediante estas pruebas se pueden detectar los problemas de rendimiento generó esto situaciones reales. Para lograr este objetivo se utilizó la herramienta *JMeter* en su versión 5.1.1.

Carga

Mediante la ejecución de las pruebas de carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella de la misma. (Sánchez Peño, 2015)

Estrés

Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de Usuarios y/o procesos, una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos que se verá reflejada en una óptima operación de negocio. (Sánchez Peño, 2015)

Resultados de las pruebas de carga y estrés

El sistema sobre el cual se realizaron las pruebas cumple con las siguientes especificaciones:

Procesador: Intel Core i3-2100 3.10 GHz

Memoria RAM: 4 GB

Disco duro: 50 GB

La siguiente tabla muestra el reporte generado por la herramienta *Jmeter*

Resultado de las pruebas de rendimiento

Etiqueta: El nombre de la muestra (conjunto de muestras).

Muestras: El número de muestras para cada URL (peticiones).

Min.: El mínimo tiempo (milisegundos) transcurrido para las muestras de la URL dada.

Max: El máximo tiempo (milisegundos) transcurrido para las muestras de la URL dada.

% Error: Porcentaje de las peticiones con errores.

Rendimiento: Rendimiento medido en base a peticiones respondidas por segundo/minuto/hora.

Usuarios: Cantidad de usuarios concurrentes.

Tabla 3.0.2 Simulaciones. **Fuente:** Elaboración propia

| Etiqueta | #Muestras | Media | Mín | Máx. | Desv.Es-tándar | %Error | Rendimiento | Kb/s | Media de Bytes |
|---|-----------|-------|-----|------|----------------|--------|-------------|--------|----------------|
| Observar las imágenes panorámicas | 50 | 2971 | 9 | 4208 | 1350.46 | 0.00 % | 1.1/s | 315.33 | 17470.1 |
| Ampliar y disminuir un área de la imagen panorámica | 50 | 1270 | 11 | 2619 | 1510.83 | 0.00 % | 1.7/s | 315.33 | 18430.4 |
| Girar en una imagen panorámica | 50 | 4277 | 15 | 2157 | 910.71 | 0.00 % | 2.0/s | 315.33 | 7470.8 |

Se simularon un total de 150 peticiones que es la media de usuarios concurrentes al sistema, obteniendo una respuesta media de 1.5 segundos aproximadamente. La ocurrencia de errores fue nula para cada petición realizada. Los resultados obtenidos se corresponden con los estándares de rendimiento en portales web y se considera que el sistema es capaz de responder

correctamente ante situaciones atípicas en las que se realizan un gran número de peticiones en un período de tiempo corto.

3.3.3 Pruebas de seguridad

Las pruebas de seguridad permiten identificar amenazas y riesgos potenciales de las aplicaciones; además de intentar medir la confidencialidad, integridad y disponibilidad de los datos. Tienen como objetivo hacer un análisis con el fin de encontrar fallos de seguridad tanto en el diseño como en la implementación de la aplicación y se parte de la identificación de amenazas y riesgos en el uso de interfaces de usuarios final. Para ello se decide utilizar la herramienta *Acunetix WVS v8*.

Tabla 3.0.3 Riesgos. **Fuente:** Elaboración propia

| Grado de Severidad | Cantidad de errores |
|--------------------|---------------------|
| Alto | 1 |
| Medio | 3 |
| Bajo | 4 |
| Informacional | 2 |

Los resultados obtenidos con las pruebas aplicadas demuestran que las vulnerabilidades encontradas dependen del servidor web donde está alojado el módulo y no corresponden a problemas críticos de seguridad del propio módulo. Para solucionar estos problemas se realizaron las siguientes acciones.

- Se aseguró de que no se divulgue información confidencial o que pudiera restringir listados de directorios de la configuración del servidor web.

- Se remplazaron los scripts/marcos que se cargan desde dominios que no se resuelven con nuevos enlaces que están apuntando a dominios válidos.
- Restringir el acceso al directorio de proveedores.
- Deshabilito el método *Trace* en el servidor web.
- Se implementó la Política de seguridad de contenido (CSP) en el módulo. La configuración de la política de seguridad de contenido implica agregar la política de seguridad de contenido HTTP encabezado a una página web y darle valores para controlar los recursos que el agente de usuario puede cargar para esa página.
- Se configuró correctamente el servidor web para que no divulgue información sobre el funcionamiento interno de una aplicación. funcionamiento al usuario.

Resultados de las pruebas de seguridad

Se realizaron tres iteraciones. En la primera iteración se detectaron 7 no conformidades, relacionadas riesgos en el uso de interfaces, que dejan evaluada de insatisfactoria la prueba. Se consideran de mayor importancia las no conformidades altas y medias. Las no conformidades referentes al entorno de despliegue de la aplicación son descartadas porque existen entidades encargadas de solucionarlas. Las no conformidades encontradas fueron solucionadas en su totalidad. A continuación se muestra un informe con los tipos de vulnerabilidades detectadas por la aplicación y su posible solución.

Tabla 3.0.4 Vulnerabilidades. **Fuente:** Elaboración propia

| Vulnerabilidades | Nivel | Solución |
|------------------|---------|---|
| Accesos remotos | Crítico | Instalación y configuración del complemento <i>Security – Firewall & Malware Scan</i> |
| Inyección SQL | Crítico | Validación con funciones que sanean las entradas de los campos al eliminar caracteres especiales. |

| | | |
|---|-------------|---|
| Enlaces rotos | Información | Corrección manual de los enlaces. |
| Falsificación de petición | Crítico | Instalación y configuración del complemento <i>Security – Firewall & Malware Scan</i> |
| Revelación de ficheros de configuración | Medio | Instalación y configuración del complemento <i>Security – Firewall & Malware Scan</i> |
| Instrucciones de código entre sitios(<i>Cross Site Scripting</i>) | Crítico | Instalación y configuración del complemento <i>Security – Firewall & Malware Scan</i> |

3.4 Conclusiones del capítulo

1. La planificación fue acertada, pues permitió el desarrollo del sistema según el cronograma y cumpliendo con los objetivos del cliente al iniciar esta investigación.
2. 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 las expectativas del cliente.
3. La utilización de los estándares de codificación permitió obtener una aplicación escalable y fácil de mantener.
4. La aplicación de la estrategia de validación comprobó que la aplicación es funcional, segura y responde en un tiempo medio de dos segundos aproximadamente. La validez de la solución se mostró a partir de las pruebas realizadas en tres iteraciones y reduciendo a cero las no conformidades sobre el resultado. La aplicación de las pruebas de carga y estrés y seguridad permitió identificar las principales deficiencias en el desarrollo, así como la estrategia para solucionar los errores detectados y obtener un producto con un alto valor, pertinencia y utilidad.

CONCLUSIONES

Como resultado de la investigación realizada se cumplieron satisfactoriamente las tareas de investigación planteadas, lo que permitió arribar a las siguientes conclusiones.

1. Se analizaron los fundamentos teóricos que caracterizan el estado del arte de la problemática planteada, así como las tecnologías para el diseño e implementación de la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas.
2. Se diseñó e implementó la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas. Las tecnologías empleadas en el desarrollo del sistema permitieron obtener un producto escalable ante actualizaciones tecnológicas como quedó demostrado en la presente investigación.
3. Se validó la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas, a partir de la aplicación de pruebas de software.
4. Se desarrolló la aplicación web para la realización del recorrido virtual por el Jardín Botánico de Matanzas que permite a los usuarios del mismo visitar la institución desde cualquier dispositivo tecnológico.

RECOMENDACIONES

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el tiempo para el desarrollo del mismo, se proponen las siguientes recomendaciones:

1. Utilizar una mejor tecnología para la recopilación de fotos panorámicas, en busca de mejorar la calidad del sistema.
2. Incrementar el número de fotos y tomas del Jardín Botánico añadiéndolo a la base de datos del recorrido y pre procesándolos para así mejorar la calidad del sistema paulatinamente.
3. Ampliar la aplicación de modo que ofrezca servicios que puedan ser consumidos por terceras aplicaciones, dada la arquitectura y técnicas de programación utilizadas.

REFERENCIAS BIBLIOGRÁFICAS

- Acevedo, Y. V. N., Quintero, J. F. L., & Clavijo, C. C. G. (2016). Recorrido virtual en tercera Dimensión de la Sede principal en una universidad de Bogotá. *Publicaciones e Investigación*, 10, 83-93. <https://doi.org/10.22490/25394088.1589>
- Barrera, C. R., Núñez Amaro, S., & Motola Pedroso, D. (2006). Evaluación de sitios Web en Internet: Propuestas para la evaluación de sitios web de bibliotecas públicas y de salud. *ACI-MED*, 14(4), 0-0.
- Bedia Palacios, M. (2021). *Desarrollo de un sistema de posicionamiento 3D de personas mediante inteligencia artificial empleando imágenes 360º* [Masters, E.T.S.I. Telecomunicación (UPM)]. <https://oa.upm.es/68669/>
- Bootstrap. (2020, enero 28). *Qué es Bootstrap—Definición, significado y ejemplos*. <https://www.arimetrics.com/glosario-digital/bootstrap>
- Cabero Almenara, J. (2005). *Reflexiones sobre la brecha digital y la educación*.
- Cabero Almenara, J., & Llorente Cejudo, M. del C. (2005). *Las TIC y la Educación Ambiental*.
- Castellano, E. (2003). Herbarios y jardines botánicos. *Biodiversidad en Venezuela*, 2, 944-957.
- Cebrián García, S. (2018). *Virtualizar un puesto de usuario*. <https://openaccess.uoc.edu/handle/10609/80567>
- Eich, B. (2020). *JavaScript | MDN*. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Elssamadisy, A. (2008). *Agile Adoption Patterns*.
- Fernández Romero, Y. (2011). *Virtualización*.

- Fleitas. (2018). *Sitio Web de SAdHEA*. https://sadhea.uho.edu.cu/inicio/sobre_sadhea
- Forero, E. (1989). SciELO - Brasil -. *Acta Botanica Brasilica*, 3, 315-322.
<https://doi.org/10.1590/S0102-33061989000300026>
- García, O. S. N. (2021). Comparativa de características de software para la creación de recorridos virtuales 360 en Web. *Revista de Investigación en Tecnologías de la Información: RITI*, 9(19), 109-117.
- Hermina. (2022). *Introducción a Express/Node—Aprende sobre desarrollo web | MDN*. https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction
- Larman—2000—. *.Pdf*. (s. f.).
- Leyva, V. de la C., Pacheco, Y. A. O., & Selema, N. E. E. (2017). Sistema informático para el diseño y gestión de recorridos virtuales para Toudroid. *3 c TIC: cuadernos de desarrollo aplicados a las TIC*, 6(2), 38-51.
- Ojeda, R. (2022). *Habana 360: Visita virtual en carro y almendron*. Habana 360. <https://www.habana-360.com/>
- Oracle. (2022). *What Is MySQL?* <https://www.oracle.com/mysql/what-is-mysql/>
- Ortega, J. A. Z., Reyes, J. de J. A., Bañuelos, C. M., Ramírez, A. M., & Hernández, C. I. M. (2014). Desarrollo de un entorno virtual tridimensional como herramienta de apoyo a la difusión turística de la zona arqueológica de Teotihuacán. *Acta Universitaria*, 24(4), 34-42.
- Pasero, B. (2022). *Documentation for Visual Studio Code*. <https://code.visualstudio.com/docs>
- Pressman, R. S. (2005). *Software Engineering: A Practitioner's Approach*. Palgrave Macmillan.

- Redmond. (2022). *Qué es la virtualización: Definición | Microsoft Azure*. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-virtualization>
- Sampieri, R. H. (2018). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. McGraw Hill México.
- Sánchez Peño, J. M. (2015). *Pruebas de Software. Fundamentos y Técnicas*.
- schwarz, B. (2021). *HTML5: What's New in The Latest Version of HTML?* <https://html.com/html5/>
- Smith, P., & Harvey-Brown, Y. (2017). *BGCI: Revisión Técnica Definición de un jardín botánico y cómo medir el funcionamiento y la gestión*.
- Sommerville, I. (2011). *INGENIERÍA DE SOFTWARE*.
- Sutherland, J. (2018). *Ideal Modeling & Diagramming Tool for Agile Team Collaboration*. Visual Paradigm Internationals. <https://www.visual-paradigm.com/>
- Takeshita Caro, A. (2020). *Bitácora del recorrido virtual*. <http://repository.javeriana.edu.co/handle/10554/50673>
- Toranzo. (2020). *Librería Virtual Cuba | Comprar online | Visita Virtual al Memorial Yate Granma del Museo de la Revolución*. <https://www.libreriavirtualcuba.com/productos.php?producto=408>
- UML_y_Patrones_-_Larman_2_Edicion.pdf*. (s. f.).
- Walke, J. (2022). *Primeros pasos en React—Aprende sobre desarrollo web | MDN*. https://developer.mozilla.org/es/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started

ANEXOS

Anexo 1

Entrevista para el levantamiento de requisitos:

¿Qué problemas debe solucionar el Sistema?

¿Cómo se resuelven ahora? (Para cada problema)

¿Cómo los resolvería? (Para cada problema)

¿Qué metodología debe usarse para su desarrollo?

¿Qué herramientas y tecnologías informáticas deben usarse?

¿Quiénes serán los usuarios del Sistema?

¿Tienen los usuarios experiencia en este tipo de aplicaciones?

¿Hay aplicaciones/sistemas relevantes a tener en cuenta?

¿Cómo valoraría que la solución ha sido un éxito?

¿Cuáles son tus expectativas en cuanto a fiabilidad del sistema?

¿Cuáles son tus expectativas en cuanto a rendimiento del sistema?

¿Dará soporte la institución al producto? ¿O lo hará otro equipo?

¿Existen necesidades especiales para dar el soporte?

¿Cuáles son los requisitos funcionales que deberá tener el sistema?

¿Cuáles son los requisitos no funcionales que deberá tener el sistema?

Atendiendo a los criterios de valoración planteados en la figura 1.1, diga cuáles considera que debe cumplir la aplicación para la realización del recorrido virtual por el Jardín Botánico de Matanzas.

Anexo 2

Tabla A 2.0.1 Historia de usuario. Mostrar la descripción correspondiente a la sala. **Fuente:** Elaboración propia

| HISTORIA DE USUARIO | |
|---|-----------------------------------|
| Numero: HU-03 | Usuario: Usuario |
| Nombre de la historia: Mostrar la descripción correspondiente a la sala. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 4 | Iteración asignada: 3 |
| Programador responsable: Alejandro David Santana Rodríguez | |
| Descripción: El software debe permitir mostrar la descripción correspondiente a la sala, lo que le facilita a los usuarios conocer en detalle la instalación | |
| Observaciones: | |

Tabla A 2.0.2 Historia de usuario. Ampliar y disminuir un área de la imagen panorámica perteneciente a la sala visualizada. **Fuente:** Elaboración propia

| HISTORIA DE USUARIO | |
|--|-----------------------------------|
| Numero: HU-04 | Usuario: Usuario |
| Nombre de la historia: Ampliar y disminuir un área de la imagen panorámica perteneciente a la sala visualizada. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 4 | Iteración asignada: 3 |
| Programador responsable: Alejandro David Santana Rodríguez | |
| Descripción: El software debe permitir Ampliar y disminuir un área de la imagen panorámica perteneciente a la sala visualizada. Con ello, se consigue mayor | |

| |
|--|
| exactitud en la visualización de los objetos pequeños que se encuentran en la escena |
| Observaciones: |

Tabla A 2.0.3 Historia de usuario. Girar en una imagen panorámica de forma horizontal. **Fuente:** Elaboración propia

| | |
|---|-----------------------------------|
| HISTORIA DE USUARIO | |
| Numero: HU-05 | Usuario: Usuario |
| Nombre de la historia: Girar en una imagen panorámica de forma horizontal. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 2 | Iteración asignada: 3 |
| Programador responsable: Alejandro David Santana Rodríguez | |
| Descripción: El software debe permitir Girar en una imagen panorámica de forma horizontal, brindándole a los usuarios mayor efecto de realismo | |
| Observaciones: | |

Tabla 2.0.4 Historia de usuario. Girar en una imagen panorámica de forma vertical. **Fuente:** Elaboración propia

| | |
|---|-----------------------------------|
| HISTORIA DE USUARIO | |
| Numero: HU-06 | Usuario: Usuario |
| Nombre de la historia: Girar en una imagen panorámica de forma vertical. | |
| Prioridad en negocio: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 2 | Iteración asignada: 3 |
| Programador responsable: Alejandro David Santana Rodríguez | |

Descripción: El software debe permitir Girar en una imagen panorámica de forma vertical, brindándoles a los usuarios mayor efecto de realismo.

Observaciones:

Tabla 2.5 Tarjeta CRC para Vista. **Fuente:** Elaboración propia

| Tarjeta CRC | |
|---|------------------------|
| Clase: <i>View</i> | |
| Responsabilidades: | Colaboraciones: |
| Es la clase encargada de: <ul style="list-style-type: none">• Visualiza y muestra las imágenes. | <i>Controller</i> |