

**UNIVERSIDAD DE MATANZAS**  
**FACULTAD DE CIENCIAS TÉCNICAS**  
**DEPARTAMENTO DE INFORMÁTICA**



**Aplicación Web para la gestión del flujo de información del Jardín  
Botánico de Matanzas**  
**Trabajo de Diploma por el Título de Ingeniero Informático**

**Autor:** Adrián Almeida García

**Tutor:** Ing. Eduardo Javier Berrio Turiño

**Matanzas, Diciembre 2022**

# Pensamiento

Los árboles son el pulmón del mundo. (Frase popular)

# Dedicatoria

# Agradecimientos

- A mis padres por nunca permitirme rendirme
- A mis amigos por no dejarme dudar
- A mi esposa por no permitirme errores

### **Declaración de autoría**

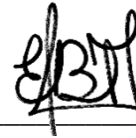
Yo, Adrián Almeida García declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas Camilo Cienfuegos a que hagan el uso que estimen pertinente de la aplicación web para la gestión del flujo de información del Jardín Botánico de Matanzas, para la finalidad que estimen conveniente.

Y para que así conste, firmo la presente a los 30 días del mes de noviembre del año 2022.



---

Firma del Autor



---

Firma del Tutor

**Opinión del Tutor.**

#### **DATOS PERSONALES DEL TUTOR**

**Nombre y apellidos:** Eduardo Javier Berrio Turiño.

**Centro de trabajo:** Universidad de Matanzas.

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

**Cargo que ocupa:** Profesor e Investigador.

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

**Categoría docente o investigativa:** Asistente.

#### **DATOS DE LA TESIS Y EL DIPLOMANTE**

**Nombre y apellidos:** Eduardo Javier Berrio Turiño.

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

**Título de la Tesis:** Aplicación web para gestión del flujo de información del Jardín Botánico de Matanzas

#### **OPINION SOBRE EL TRABAJO**

La tesis presentada posee gran actualidad, pues intenta resolver un problema real presente en la Universidad de Matanzas, y además contribuye a la informatización de nuestra sociedad.

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

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

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

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

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

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



---

Ing. Eduardo J. Berrio Turiño

Dpto. Informática

Universidad de Matanzas

Noviembre/2022

## **RESUMEN:**

Una aplicación web brinda la oportunidad de implementar una gran cantidad de servicios, además de centralizar en un mismo sitio la información que se requiere dentro de una institución para sus actividades. Es considerado como una herramienta que contribuye a la difusión de la misma, teniendo a los usuarios actualizados acerca de la realidad de la misma. La razón fundamental de implementar un sitio web en una organización es que el internet es hoy en día el medio de comunicación más eficiente y económica para impulsar una relación comercial entre las empresas que ofertan sus productos y servicios y los mercados consumidores. El presente trabajo tiene por objetivo la elaboración del sitio web para el Jardín Botánico de la Universidad de Matanzas que permitirá a usuarios beneficiarse con los servicios que éste brinda, pues en los tiempos actuales el uso de las tecnologías de la informática y las comunicaciones se ha vuelto mucho más común ya que permite la realización de disímiles actividades, en menor tiempo y con mayor comodidad. Con el desarrollo de esta aplicación web se pretende agilizar la labor de los especialistas del Jardín Botánico Matanzas y así contribuir a la correcta conservación y control de los individuos que residen en el campus universitario, así como divulgar la labor de tales especialistas, aportando a la cultura general de los estudiantes de la Universidad de Matanzas

Palabras claves: Jardín Botánico, Aplicación Web, Universidad de Matanzas, Conservación de la flora



**Abstract:**

A web application offers the opportunity to implement a large number of services, in addition to centralizing in the same place the information that is required within an institution for its activities. It is considered as a tool that contributes to its dissemination, keeping users updated about its reality. The fundamental reason for implementing a website in an organization is that the internet is today the most efficient and economical means of communication to promote a commercial relationship between the companies that offer their products and services and the consumer markets. The objective of this work is the development of the website for the Botanical Garden of the University of Matanzas that will allow users to benefit from the services it provides, since in current times the use of information and communication technologies has become much more common since it allows the performance of dissimilar activities, in less time and with greater comfort. The development of this web application is intended to streamline the work of specialists from the Matanzas Botanical Garden and thus contribute to the correct conservation and control of the individuals residing on the university campus, as well as disseminate the work of such specialists, contributing to the general culture of the students of the University of Matanzas

Keywords: Botanical Garden, Web Application, and Matanzas University

## TABLA DE CONTENIDO

Introducción .....	13
Capítulo 1: Marco teórico de la investigación .....	17
1.1: Introducción .....	17
1.2: Objeto de estudio.....	17
1.3: Antecedentes.....	17
1.4: Metodologías de desarrollo de software .....	18
1.4.1: Metodologías ágiles de desarrollo de software .....	18
1.4.2: Metodología XP: .....	19
1.5: Arquitectura n capas .....	21
1.6: Herramientas, lenguajes de programación y tecnologías empleadas .....	23
Conclusiones parciales.....	27
Capítulo 2: Solución teórica del problema científico.....	29
2.1: Introduccion .....	29
2.1: Descripción del negocio.....	29
2.2: Levantamiento de Requerimientos .....	31
2.2.1: Requisitos funcionales.....	31
2.2.2: Requisitos no funcionales .....	34
2.3: Modelado del negocio .....	35
2.4: Diagrama de casos de uso del sistema .....	36
2.5: Historias de usuario.....	37
2.6: Diseño conceptual de la base de datos .....	39
2.7: Conclusiones parciales del capítulo.....	40
Capítulo 3: Propuesta de solución práctica al problema científico.....	41
3.1: Introducción .....	41
3.2: Descripción de la solución.....	41
3.3: Pruebas de software .....	45
3.3.1: Métodos empleados para la realización de las pruebas. ....	46
3.3.2: Estrategia de pruebas para HU.....	47
3.4: Resultados de las pruebas realizadas.....	50
Conclusiones parciales.....	50
Conclusiones .....	51
Recomendaciones .....	52
Referencias.....	53
Glosario de términos .....	54

## Índice de imágenes:

Ilustración 1: Estructura por capas de la aplicación .....	23
Ilustración 2: Modelo BPMN del proceso de trabajo del Jardín Botánico de Matanzas .....	36
Ilustración 3: Diagrama de casos de uso del sistema .....	37
Ilustración 4: Modelo conceptual de la base de datos .....	40
Ilustración 5: Vista de la aplicación en modo iPhone12 Pro Max.....	43
Ilustración 6: Vista de la integración con la función de búsqueda del navegador .....	44
Ilustración 7: Validación asíncrona .....	45
Ilustración 8: Validación síncrona.....	45

## Índice de tablas

Tabla 1: Historias de usuario .....	37
Tabla 2: Plan de pruebas por historia de usuario .....	47
Tabla 3: Pruebas realizadas y resultados .....	48

## INTRODUCCIÓN

Un jardín botánico tiene la misión de ser un centro científico, educativo y cultural que al mismo tiempo sirve como área recreativa. Para el cumplimiento de estos objetivos se ve la necesidad de optimizar la manipulación de la información generada en el jardín botánico y a la vez divulgar esta información al público en general, fortaleciendo así el sentido de identidad nacional de la sociedad con respecto a su patrimonio natural.

El Jardín Botánico de Matanzas es un proyecto en desarrollo que data de la última década del pasado siglo. Durante sus años de existencia ha atesorado ejemplares de la flora cubana, así como de otros países. Su misión consiste en preservar las especies de la flora que residen en la provincia de Matanzas y garantizar la armonía con la fauna, así como concientizar en los hábitos de una correcta cultura ambientalista, como premisa de la necesidad de la conservación de la flora por su papel crucial en la permanencia de la vida en el planeta.

Para la realización de su labor, los especialistas del Jardín Botánico de Matanzas se valen de herramientas y plataformas informáticas no especializadas, las cuales no son capaces de satisfacer la totalidad de sus necesidades y exigencias. La información que se maneja no es actualizada, puesto que no da a conocer al público en general de los servicios y labores que realiza el mismo. Una solución a esta problemática existente son las Tecnologías de Información y Comunicación (TICs), que han venido a revolucionar la forma y la habilidad con que los negocios y grandes empresas solucionan sus actividades cotidianas, optimizando así la forma de procesar, almacenar y presentar la información.

Donde el proceso de actualización y modernización de la infraestructura tecnológica, tanto a nivel hardware como software, se ha consolidado en los últimos años y ha permitido el mejoramiento de la capacidad de procesamiento de información de propiedad de la entidad; lo anterior conlleva contar con nuevas y mejores formas de comunicación, nuevos desarrollos de aplicaciones y sistemas de información y en general estar a la vanguardia de la tecnología, que permita el logro de la misionalidad institucional de manera efectiva. Para el sostenimiento de esta infraestructura, se requiere de personal idóneo en capacidad de administrar, monitorear y hacer seguimiento al correcto y adecuado funcionamiento de cada uno de los componentes de hardware, software y servicios puestos a disposición tanto de los funcionarios y colaboradores internos.

Por lo que se decide el desarrollo de una aplicación web tomando en cuenta las facilidades que esta plataforma nos provee, tales como la multiplataforma, la disponibilidad rápida y la seguridad en la información, además también se tomó en cuenta las limitaciones tecnológicas producto de las sanciones impuestas a Cuba.

Como **situación problemática** se plantea la insuficiente aplicación de las nuevas tecnologías y herramientas informáticas para la recopilación y el análisis de información, lo que dificulta disponer de información clave, pertinente, actualizada, centraliza y relevante.

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

Para la solución del problema se trazó la **hipótesis** que rige la investigación, si se desarrolla un Sistema para el Gestión del flujo de la información del Jardín Botánico de Matanzas que contribuirá a elevar la efectividad de este proceso.

Se define como **objetivo general** de la investigación: desarrollar una aplicación web para la gestión del flujo de información del Jardín Botánico Matanzas que contribuya a la elevación de la eficacia y la eficiencia de este proceso.

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

1. Analizar la bibliografía que permita sentar las bases de la investigación y conformar el marco teórico referencial de la misma.
2. Diseñar el Sistema para el Gestión del flujo de la información del Jardín Botánico de Matanzas utilizando la metodología de desarrollo XP.
3. Validar el Sistema para el Gestión del flujo de la información del Jardín Botánico de Matanzas a partir de la aplicación de pruebas de software.

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

Dentro de los métodos teóricos:

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

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

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

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

Entre los aportes de la investigación se destacan:

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

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

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

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



## CAPÍTULO 1: MARCO TEÓRICO DE LA INVESTIGACIÓN

### 1.1: INTRODUCCIÓN

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

### 1.2: OBJETO DE ESTUDIO

El Jardín Botánico de Matanzas tiene como principal objetivo coleccionar especies de la flora para su conservación y estudio científico. Actualmente para la realización de sus tareas se auxilian de hojas de cálculo de Microsoft Excel. Esta cuenta además con un equipo de especialistas y un equipo de campo. Los especialistas se encargan de recopilar y registrar datos que los equipos de campo obtienen del terreno. Los equipos de campo pueden estar integrados por especialistas. El procesamiento de la información se realiza de forma que para cada colección existe una hoja de cálculo de Microsoft Excel donde se registran los datos de cada individuo, esto les permite llevar un control parcial de la información. El avance del proyecto de creación del Jardín Botánico de Matanzas, su postura como una institución científica y el proceso de informatización que vive la sociedad cubana conllevan a que la institución requiera de una herramienta de software propia que permita no solo la gestión del flujo de información sino también la conservación de la información en sí para el estudio preciso de la evolución de los individuos, así como de su interacción con el entorno

### 1.3: ANTECEDENTES

**Aplicación de Android para el “Jardín Botánico de la UCLV”** (Cuba): Trabajo de diploma desarrollado por José Alberto Bisbal Martín en 2019.

**Software móvil para reconocimiento, seguimiento y control de las plantas del vivero inteligente de la Universidad Francisco de Paula Santander Ocaña** (Colombia): Desarrollado por los MS.c Luis Anderson Coronel-Rojas, MS.c Dewar Rico-Batista, Ms.c Fabián Ranulfo Cuesta-Quintero, MS.c Edwin Barrientos-Avedaños, Est. Eimar Alveiro Pedraza Villadiego

## 1.4: METODOLOGÍAS DE DESARROLLO DE SOFTWARE

La mente humana es una maquina biológica metódica. Nuestros antepasados aprendieron que, si sembrabas una semilla, con el tiempo nacía un nuevo árbol y ese conocimiento pasó de generación en generación. Nuestro propio desarrollo como especie y el auge de civilizaciones avanzadas llevaron al ser humano a pensar en grande, a construir artilugios que le permitieran facilitar su día a día, así como el de sus semejantes. Existieron quienes documentaron estas creaciones, detallándolas paso a paso para que los que vinieran después pudieran replicar el proceso con precisión y lograr el mismo resultado, sin saberlo, los humanos de la antigüedad crearon lo que hoy se conoce como metodología

Tomando esto en cuenta podemos definir una metodología como un conjunto de pasos detallados que se deben realizar para lograr un resultado concreto. Estas no deben ser monolíticas puesto que el humano antiguo también descubriría algo esencial, nada es para siempre y así como el mundo cambia también lo hacen las formas del hacer humano, las metodologías de trabajo.

---

### 1.4.1: METODOLOGÍAS ÁGILES DE DESARROLLO DE SOFTWARE

Según (Grupo ISSI, 2003) el desarrollo de software no es tarea fácil y existen numerosas propuestas metodológicas, las más tradicionales se enfocan en lograr un dominio total del proceso y han probado ser efectivas en muchos proyectos, pero en otros han sido un problema más que una solución.

Estas metodologías conocidas peyorativamente como “pesadas” requieren que el equipo de desarrollo domine cada aspecto del proceso a desarrollar apoyándose en establecer con total detalle todos los requerimientos del sistema así como la definición de roles y grupos de trabajo altamente especializados en el equipo de desarrollo

Estas metodologías, erróneamente vistas como monolíticas, han probado su efectividad en proyectos de gran envergadura, pero no así en proyectos más pequeños, un enfoque diferente para esta problemática son las metodologías ágiles

Las metodologías ágiles han ganado auge en los últimos 25 años gracias al auge de Internet y el negocio online, muchas empresas pequeñas y medianas se han beneficiado gracias a ello. Estas metodologías asumen que los cambios en los requisitos del software son algo imposible de evitar y toman un enfoque más personal entre cliente y desarrollador, llegando

al punto en que el cliente se convierte en un miembro más del equipo de desarrollo.

---

#### 1.4.2: METODOLOGÍA XP:

Programación extrema es una metodología de desarrollo en la ingeniería de software formulada por Kent Beck y autor del primer libro sobre esta metodología, esta pone más énfasis en la adaptabilidad y en la previsión, dando por hecho que los cambios en los requisitos durante la marcha del proceso de desarrollo son algo inevitable e incluso deseable. Defiende que ser capaz de adaptarse a los cambios en los requisitos en cualquier punto del proceso de desarrollo del proyecto es un mejor enfoque que intentar definir todos los requisitos desde un primer punto y luego invertir más tiempo y recursos en controlar los cambios que puedan presentar los requisitos

##### **Características**

- Se aplica de manera dinámica en el proceso de desarrollo del software
- Los individuos e interacciones son más importantes que los procesos y herramientas
- Capaz de adaptarse a los cambios en los requisitos
- Un software que funcione es más importante que una documentación exhaustiva de forma que no genera documentos a no ser que sean necesarios para la toma inmediata de decisiones
- Propone que exista una interacción constante entre el cliente y el equipo

##### **Fases**

La metodología XP define 4 fases esenciales

##### **Planificación**

###### ❖ **Historias de usuario(HU)**

Tienen la misma finalidad que los casos de uso solo estos constan de 3 a 4 líneas escritas por el cliente en un lenguaje lo más cercano al coloquial que sea posible, estas deben verificarse en la fase de pruebas para comprobar que el software cumple con los requisitos planteados

###### ❖ **Planificación de entregas**

Luego de definir las historias de usuario se debe crear un plan de entregas que defina los tiempos de implementación aproximados de las historias de usuario así como que historias serán implementadas en cada versión del software

#### ❖ Iteraciones

Plantea periodos de tiempo en los que se implementaran las funcionalidades de cada plan de entrega

### Diseño

#### ❖ Diseños simples

Conseguir diseños simples facilitara la comprensión e implementación, que a su vez costara menos tiempo y esfuerzo desarrollar

### Codificación

Durante esta fase se comienza la implementación, cabe recalcar que el cliente es un factor importante de esta fase, ya que es quien escribe las historias de usuario así como es el que comprueba junto con el equipo que dichas historias fueron implementadas correctamente mediante pruebas de funcionalidad

### Pruebas

Para la correcta realización de esta fase se prefiere que las pruebas sean creadas antes del código en sí, que dichas pruebas no dependen de código que se desarrollara en un futuro, de ser así se deben omitir

### Roles XP:

Los roles dentro de la metodología XP se definen para asignar responsabilidades al equipo de trabajo, así como para definir de forma más precisa los actores del proceso de desarrollo.

Estos son los roles de acuerdo con la propuesta original de Kent Beck:

**-Programador:** Produce el código del sistema y de sus pruebas unitarias

**-Cliente:** Escribe las historias de usuario, asigna prioridad a estas y realiza las pruebas de validación de cada funcionalidad

**-Encargado de pruebas (tester):** Auxilia al usuario en las pruebas funcionales, las ejecuta regularmente y difunde los resultados con el equipo de desarrollo. Esta función muchas veces es ocupada por el programador

**-Encargado de seguimiento (tracker):** Es el encargado de estimar el avance del proyecto tomando como referencia los plazos de entrega de las versiones del sistema

**-Entrenador:** Se encarga de velar por que el equipo siga correctamente las practicas XP y proporciona guía a los miembros

**-Gestor:** Es quien está en contacto directo con el cliente, su labor esencial es coordinar el trabajo del equipo con las necesidades del cliente.

### **Proceso XP:**

El proceso XP es un ciclo que consiste, de forma general en 5 pasos:

1. El cliente define la prioridad del negocio a implementar
2. El programador estima el tiempo necesario para su implementación
3. El cliente selecciona que implementar teniendo en cuenta sus prioridades y las restricciones de tiempo
4. El programador implementa la funcionalidad
5. Vuelve al paso 1

## **1.5: ARQUITECTURA N CAPAS**

Una arquitectura de n capas divide una aplicación en capas lógicas y niveles físicos. Las capas son una forma de separar responsabilidades y administrar dependencias. Cada capa tiene una responsabilidad específica. Una capa superior puede utilizar los servicios de una capa inferior, pero no al revés. (1)

Esta arquitectura tiene dos enfoques:

- Capa cerrada: Cada capa solo puede llamar a la capa inmediata inferior a ella
- Capa abierta: Una capa puede llamar a cualquiera de las capas que se encuentran debajo de esta

Las arquitecturas de n capas se emplean normalmente en aplicaciones donde sus elementos se ejecutan en contenedores aplicativos diferentes, segregando las dependencias y

requerimientos de cada capa de manera que el sistema se articula como las piezas de un rompecabezas, piezas que se comunican e intercambian información mediante mensajes

### **Ventajas:**

**Alta portabilidad:** Cada capa es un elemento separado que puede integrarse a otros sistemas mediante el uso de componentes mediadores de ser necesario

**Menor curva de aprendizaje:** Es más sencillo comprender un sistema mediante el estudio de cada componente como un ente único, este enfoque reduce el tiempo necesario para la preparación del personal y por tanto el tiempo de desarrollo requerido.

**Abierta a entornos heterogéneos:** Segregar dependencias tiene como resultado la abstracción del entorno de desarrollo, por lo tanto, se pueden tener capas que se ejecutan en el entorno de Windows que se comuniquen con otras en un entorno de Linux y el sistema funcionara

### **Desventajas:**

**La comunicación como talón de Aquiles:** Se debe asegurar la confidencialidad de la información contenida en los mensajes que transmite cada capa y su correcta recepción puesto que puede generar vulnerabilidades en el sistema

### **Malas prácticas comunes:**

Es sencillo cometer estos errores si no se desarrolla la arquitectura correctamente, a continuación, se citan:

- **Capas intermedias innecesarias:** Es fácil terminar implementando capas que solos realiza operaciones CRUD con la base de datos añadiendo latencia extra al sistema
- **Diseño monolítico:** En ocasiones se ignora la posibilidad de cambiar una capa en el futuro, se recomienda implementar reduciendo al mínimo el acoplamiento de forma que permita la extensión o reemplazo de cualquiera de los componentes de las capas

Para el desarrollo de se escogió este patrón arquitectónico usando un enfoque de capa cerrada pero permitiendo que cada capa pueda comunicarse con la inmediata superior para responder a las peticiones realizadas, los mensajes se realizan en forma de peticiones HTTP entre el cliente y el servidor, luego el servidor realiza la operación asociada a la petición, en caso de requerir información de la base de datos este realiza una consulta SQL y recibe de la base de datos la información referente a dicha consulta. La respuesta a las peticiones HTTP siempre se realiza usando documentos en formato JSON

A continuación, se presenta un diagrama de la estructura de la aplicación

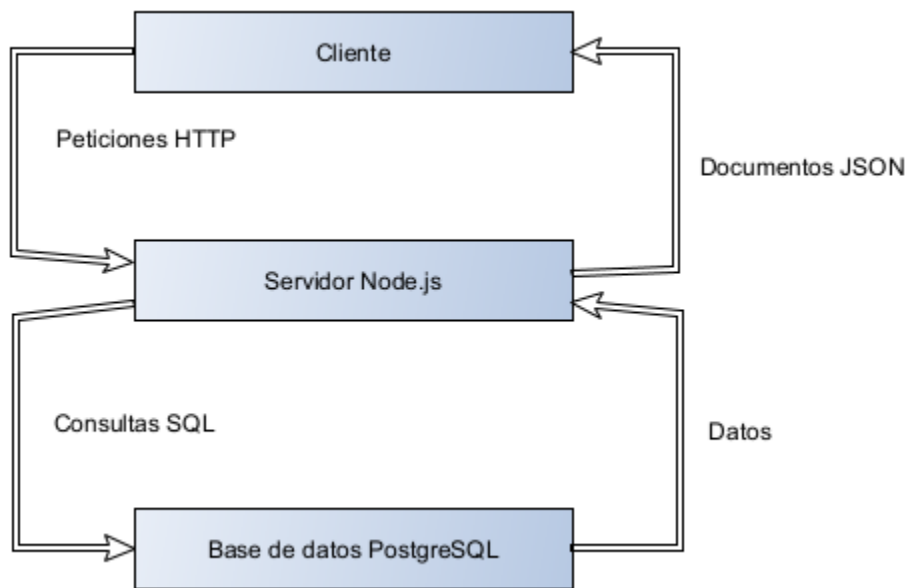


Ilustración 1: Estructura por capas de la aplicación

## 1.6: HERRAMIENTAS, LENGUAJES DE PROGRAMACIÓN Y TECNOLOGÍAS EMPLEADAS

Para la realización de esta investigación se emplearon disímiles herramientas, así como lenguajes de programación, dichas están citadas a continuación

**HTML:** Hypertext Markup Language, o Lenguaje de Marcado de Hipertexto es un lenguaje de marcado utilizado en la programación web. Permite estructurar los elementos de un sitio web a través de la declaración de etiquetas por lo que es llamado comúnmente “el esqueleto de la web”.

**CSS:** Cascading Style Sheets es un lenguaje de diseño gráfico para definir y crear una presentación en un documento creado con un lenguaje de marcado, es el segundo elemento más importante en una web después de HTML (Tomado de la Guía de desarrolladores de Mozilla Firefox 3 de junio del 2022). CSS permite añadir desde personalizar la apariencia de los elementos HTML en una página web mediante el uso de clases, cada una con un grupo de propiedades acordes a las intenciones del desarrollador. Su función complementa la usabilidad de la web, permitiendo crear elementos amenos al ojo humano. CSS3 es la última versión de CSS que añade nuevas características que ayudan a resolver problemas comunes del desarrollo sin necesidad de usar jerarquías complejas, marcadores semánticos o scripts complejos.

**JavaScript:** JavaScript es un lenguaje interpretado basado en objetos comúnmente confundido con Java, pero del que solo comparte la sintaxis, es quien provee la funcionalidad a una aplicación web, tanto del lado del cliente como del lado del servidor. Este lenguaje se ejecuta en el navegador web y gracias a ello puede ejecutarse en cualquier dispositivo que posea esta herramienta, esta ventaja le abre un universo de posibilidades a los desarrolladores que pretendan crear aplicaciones altamente usables y que puedan estar disponibles en múltiples plataformas de software. JavaScript llega mucho más lejos, pues permite a los desarrolladores abstraerse completamente de elementos como sistemas operativos o requerimientos complejos de hardware o software

El uso de marcos de trabajo facilita el desarrollo de aplicaciones para los desarrolladores ya que automatiza muchos patrones utilizados para resolver tareas comunes. Estos proporcionan, en la mayoría de los casos, un código fuente que fuerza al desarrollador a crear un código más legible y más fácil de mantener y extender

Definidas las tecnologías básicas, a continuación se presentan las herramientas que se pretende emplear para el diseño, modelado, codificación y pruebas del sistema que se desea desarrollar.

**Bootstrap:** Es una librería de CSS, gratis y de código abierto que contiene una galería de elementos ya creados para la personalización de una página web, su diseño minimalista crea interfaces intuitivas y ágiles capaces de desplegarse en cualquier dispositivo adaptándose automáticamente a la pantalla que acceda a la web, una técnica conocida como “responsive desing”. Su uso es polémico ya que si bien permite al desarrollador abstraerse del aspecto visual, las propiedades y la jerarquía de las clases CSS esto conlleva que soporte muy poca modificación por parte del usuario haciéndola altamente monolítica y en muchos casos impráctica. Se decide su uso para esta investigación debido al énfasis de este en la funcionalidad simple sin muchos elementos visuales que constituyan una distracción o conlleven a la sobrecarga de elementos en una vista. A modo de resumen Bootstrap es una librería para aplicaciones pequeñas desarrolladas por equipo pequeños, por lo general de tres personas o menos en las que no siempre se cuenta con un diseñador de interfaces web

**React:** Es una plataforma libre basada en JavaScript creada y mantenida por Facebook para el desarrollo de aplicaciones del lado del cliente. Se basa en un diseño modular donde cada elemento de la web se desarrolla como un componente y luego estos se articulan en función



de lo que se pretenda desarrollar. Esto permite el desarrollo de aplicaciones fáciles de mantener y extender. Esta permite crear aplicaciones web usando solo JavaScript a través de elementos propios de esta plataforma como JSX. Se integra con otras plataformas tanto de JavaScript como de otros lenguajes como PHP, Python, C# o Ruby.

Esta plataforma destaca creando un tipo de aplicación web conocida como “aplicación de una sola página”, (Single Page App o SPA por sus siglas en inglés) las cuales despliegan un solo documento HTML el cual modifican en tiempo de ejecución siguiendo las indicaciones del usuario.

Esto permite resolver muchas funcionalidades del lado del cliente, al punto que la aplicación puede funcionar sin el lado del servidor y solo contacta con el lado del servidor para intercambiar información: como insertar u obtener elementos en la base de datos.

Para el desarrollo de la aplicación del lado del cliente se emplearon diferentes librerías de terceros compatibles con React, a continuación se exponen las más importantes, el resto se encuentra en el epígrafe 2.3 de este documento:

- **Yup:** Se emplea para la validación de los formularios, trabaja de manera asíncrona de forma que espera a que el usuario libere el control para aplicar una validación, posee múltiples funciones de validación ya implementadas que solucionan la mayoría de los problemas comunes en este ámbito y soporta la extensión mediante la creación de nuevas funciones que permitan al desarrollador definir sus propios validadores. Se escogió gracias a su facilidad de uso y legibilidad
- **Formik:** Se emplea para el diseño de los formularios. Ofrece componentes ya creados para la elaboración de los controles de los formularios, el despliegue de mensajes de errores así como la inicialización y envío de los datos de cada formulario. Soporta integración con Bootstrap y Yup
- **Leaflet:** Librería de control y trabajo con mapas desarrollada para el trabajo con cualquier repositorio de mapas, aunque para esta investigación se empleó OpenMaps
- **Leaflet-Draw:** Librería que permite el marcado y trazado de elementos superpuestos en un mapa como marcadores de superficie, posiciones puntuales así como el despliegue de información interactiva de estos

**JSX:** Es un lenguaje de etiquetado que imita a HTML y es empleado por React para el diseño de las interfaces. Es la principal causa de la alta modularidad en las aplicaciones

desarrolladas con React. Se puede retornar elementos JSX como resultado de cualquier función de JavaScript permitiendo al programador crear estructuras altamente reutilizables y modificables que reducen el tiempo necesario para la codificación de un componente puesto que los elementos comunes entre componentes se codifican una sola vez

**Node.js:** Es una plataforma libre de JavaScript para el desarrollo de aplicaciones del lado del servidor. Permite el desarrollo de aplicaciones del lado del servidor altamente modulares. Se vale de una librería de JavaScript llamada Express que provee la funcionalidad a la plataforma. Las principales características en el uso de Node.js se exponen a continuación.

- ❖ **Baja curva de aprendizaje:** Se requiere poco entrenamiento para dominar los conceptos básicos de Express y existe una vasta documentación provista por sus desarrolladores y la comunidad que permite resolver los problemas más comunes
- ❖ **Bajos requerimientos de hardware:** Al tratarse de JavaScript, las necesidades de hardware de una aplicación son bajas, reduciendo los costes de despliegue y mantenimiento de la aplicación
- ❖ **Personalizable:** No posee un enfoque a una arquitectura de software en específico, por lo que el desarrollador puede adaptar el sistema a la que le sea más eficiente. Para esta investigación se utiliza el Modelo – Vista – Controlador (MVC)
- ❖ **Abstracción del cliente:** La aplicación recibe y responde peticiones HTTP de cualquier cliente, esto permite que el programador pueda definir el comportamiento de los módulos desde un alto nivel de abstracción. Esto conlleva un punto negativo en la seguridad, que obliga al desarrollador a establecer y definir los clientes que pueden realizar peticiones al servidor y la respuesta que pueden recibir

**PostgreSQL:** Es un sistema de gestión de bases de datos relacionales libre publicado bajo una licencia homónima similar a la MIT o la BSD. Esto significa que PostgreSQL no es manejado por una empresa o persona, sino por una comunidad de programadores que trabajan de forma altruista y desinteresada para mantener y actualizar este sistema. Este elemento permite que usando este sistema gestor de bases de datos se puedan crear aplicaciones comercializables a pesar de estar empleando tecnologías libres.

**Sequelize:** Es un herramienta ORM para Node.js basada en promesa (2) que permite a los desarrolladores manipular la base de datos sin insertar código SQL. Posee soporte para múltiples dialectos, así como otras funcionalidades como relacionamientos y transacciones

**Google Chrome:** Navegador Web gratuito desarrollado por Google. Posee múltiples herramientas de desarrollo que facilitan las pequeñas rectificaciones a la estructura final de

la web. También provee de un espacio de almacenamiento temporal para preservar información importante

**Visual Studio Code:** Editor de código desarrollado por Microsoft, gratis y de código abierto favorece el desarrollo ágil de cualquier tipo de aplicaciones gracias a una enorme comunidad y el uso de extensiones que se adecuan a las tareas a realizar. Con una interfaz intuitiva e integración con los servicios y plataformas de desarrollo es una herramienta esencial para la programación en cualquier nivel. Durante el desarrollo de esta investigación se utilizaron múltiples extensiones libres entre las que a continuación se citan las más importantes:

- ❖ **ThunderClient:** Generador de peticiones HTTP que permite realizar pruebas al lado del servidor y observar su comportamiento sin necesidad de codificar una aplicación cliente similar a lo que permite el software Postman.
- ❖ **Git Lens:** Facilita la integración del entorno local de desarrollo con el repositorio de código abierto GitHub. Ofrece funcionalidades como la regresión de versiones y control de cambios permitiendo un eficiente control en las versiones del software
- ❖ **DB Manager:** Permite realizar conexiones con los sistemas gestores de bases de datos para su control parcial, no sustituye herramientas como pgAdmin o phpMyAdmin más bien las complementa permitiendo realizar pequeñas consultas y modificaciones en la base de datos sin salir del entorno de desarrollo
- ❖ **ES7 Linter:** Herramienta de resaltado de código JavaScript. La idea detrás de esta extensión es obligar al programador a crear código en JavaScript que siga con el estándar ES7 de lenguaje
- ❖ **Doxygen:** Extensión para la creación de documentación de software a través de la declaración de descripciones simples en los componentes

**BrModelo:** Herramienta para la elaboración de modelos de bases de datos, en cualquiera de sus niveles tanto conceptual, lógico o físico. Esta herramienta cuenta con una versión de escritorio y una web, ambas totalmente libres

**OpenMaps:** Biblioteca de mapas libre que soporta diferentes funcionalidades de marcado y dibujo.

## CONCLUSIONES PARCIALES

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

1. En el desarrollo del capítulo se consolidó el basamento teórico, sobre el cual se sustenta la propuesta de trabajo, con vistas al desarrollo del software mediante un estudio de las principales herramientas y tecnologías que se proponen para el proyecto, se sientan las bases que dan pie a su inicio y además de forma general se ha contribuido a la mejor comprensión del objeto de estudio.
2. La combinación de herramientas, tecnologías y la metodología de desarrollo de software seleccionada, es la apropiada para la realización del sistema ya que resuelve la situación problémica planteada en la investigación.

## CAPÍTULO 2: SOLUCIÓN TEÓRICA DEL PROBLEMA CIENTÍFICO

### 2.1: INTRODUCCION

En el desarrollo de software es muy importante definir los procesos que intervienen en este para lograr un mejor entendimiento del negocio a informatizar. Este capítulo tiene como objetivo exponer el contexto del problema que se pretende resolver. Se pretende dar una descripción detallada del negocio y definir los requisitos del sistema para con esto modelar el negocio empleando tanto diagramas UML como BPMN y la base de datos, haciendo uso de la herramienta brModelo.

Siguiendo la metodología de desarrollo empleada, Programación Extrema (XP) se han definido historias de usuario que describen con términos del cliente como debe funcionar el sistema

### 2.1: DESCRIPCIÓN DEL NEGOCIO

#### **Colecciones e individuos**

1. Una colección es un conjunto de individuos que interactúan entre sí. Estas se clasifican en colonias, que son las creadas por la mano humana y comunidades que son las formadas naturalmente. Una colección tiene como características su nombre vulgar, nombre científico, nombre de la familia de individuos que forman parte de ella, su posición geográfica y la cantidad total de individuos que la conforman.

Una colección se comienza a estudiar desde el momento que se descubre, esto se realiza por parte de un equipo de campo el cual entrega los datos a los especialistas del Jardín Botánico de Matanzas y estos los registran en el sistema y comienzan su estudio periódico.

2. Los individuos son los organismos individuales que pertenecen a una colección. Un equipo de campo realiza su descubrimiento le coloca un identificador, registra su nombre vulgar, nombre científico, nombre de la familia a la que pertenece, posición geográfica, diámetro y altura. Pueden existir colecciones de un solo individuo
3. Los especialistas utilizan un mapa aéreo para ubicar los individuos, cuando un individuo se registra se añade un marcador en dicho mapa en la posición del individuo.

4. Un individuo puede ser trasladado hacia otra colección, para esto se registra la fecha del traslado y la colección de origen y la colección de destino

### **Seguimiento, estudio y mantenimiento**

1. Se realizan estudios periódicos a las colecciones e individuos para actualizar la información sobre su estado, para ello los especialistas programan una tarea la cual tiene dos fechas entre las cuales deberá completarse la tarea, una descripción breve de dicha tarea y la persona o grupo de personas a las que se le asigna esa tarea

En caso de alteración en la información registrada en el sistema referente a las colecciones e individuos se requiere conservar por tiempo indefinido los estados anteriores de cada colección e individuo por separado para el desarrollo de futuras investigaciones por parte del Jardín Botánico de Matanzas o de terceros.

2. En caso de presentarse situaciones excepcionales que impidan la realización de una tarea los especialistas pueden posponer su realización
3. Se requiere llevar un registro de las tareas cumplidas por un periodo indefinido, al cumplirse los objetivos de una tarea, un especialista puede dar la tarea por cumplida, para esto se marca la tarea como cumplida y se registra la fecha de cumplimiento. Esta información también se utiliza para la evaluación sistemática del personal
4. Un equipo puede tener varias tareas asignadas a una misma jornada, si estas no pueden realizarse en su totalidad se deberán reprogramar teniendo en cuenta que no existan factores que invaliden su realización, o cancelar en caso contrario
5. Si los especialistas consideran que una tarea ya no es necesaria esta debe eliminarse del sistema

### **Gestión del personal**

1. No todo el personal del Jardín Botánico Matanzas tiene acceso al sistema. Los especialistas con acceso al sistema son seleccionados por el director del Jardín Botánico de Matanzas el cual les asigna un usuario en el sistema, para ello se le otorga un nombre de usuario que es único y una contraseña escogida por el usuario.
2. El director del Jardín Botánico de Matanzas puede otorgar y revocar permisos sobre el sistema a los especialistas según sea necesario
3. El director del Jardín Botánico de Matanzas o un especialista autorizado puede registrar un trabajador en el sistema, de estos se registra su número del carnet de

identidad, nombre completo y el cargo que ocupa. De manera opcional se registra su número telefónico, correo electrónico

4. El director del Jardín Botánico puede modificar o eliminar la información de los usuarios del sistema
5. El director del Jardín Botánico puede modificar o eliminar la información del personal que labora en la institución
6. Los directores del Jardín Botánico de Matanzas es el único autorizado a manejar la información de los registros de estados anteriores de individuos y colecciones
7. El administrador del sistema puede restablecer la contraseña de acceso de cualquier usuario que por cualquier motivo la haya perdido siempre y cuando este se presente personalmente en sus dependencias y solicite el servicio
8. Todos los usuarios del sistema pueden cambiar su contraseña en el momento que lo consideren

## 2.2: LEVANTAMIENTO DE REQUERIMIENTOS

A través de la captura de requisitos se recopila la información y se transforma en un conjunto de requerimientos que son los que darán límite al alcance del sistema. Estos requisitos pueden ser funcionales y no funcionales. Los requisitos funcionales especifican los servicios o funciones que proveerá el sistema, mientras que los no funcionales definen las restricciones que tendrá este, las cuales limitan las elecciones para construir una solución

### 2.2.1: REQUISITOS FUNCIONALES

1. Mostrar el mapa con las colecciones e individuos
  - a. Mostrar un mapa del área del Jardín Botánico Matanzas con las colecciones e individuos existentes
2. Mostrar información sobre una colección o individuo
  - a. Mostrar una vista con los datos actuales de la colección o individuo al dar clic en su posición en el mapa o al buscarlo en el sistema
  - b. Mostrar un botón que acceda al historial de cambios en la colección o individuo en seleccionado
3. Añadir una colección
  - a. Mostrar un formulario con la información requerida
  - b. Registrar la colección en la base de datos
4. Añadir un individuo
  - a. Mostrar un formulario con la información requerida

- b. Insertar la posición del individuo a partir de sus coordenadas geográficas o insertar dichas coordenadas en el formulario
  - c. Añadir el individuo a una colección existente, en caso de no existir el usuario debe contar con un control que le permita crear la colección
  - d. Registrar el individuo en la base de datos
5. Modificar la información de un colección o individuo
- a. Mostrar el control que acceda a una vista detallada de la colección o individuo
  - b. Mostrar una vista con los datos actuales de la colección o individuo al dar clic en su posición en el mapa o al buscarlo en el sistema
  - c. Mostrar un formulario con la información actual para su edición
  - d. Registrar el estado anterior en la base de datos junto con la fecha de realización del cambio
  - e. Actualizar el estado actual en la base de datos
6. Eliminar un individuo o colección
- a. Mostrar el control que habilita la eliminación
  - b. Pedir confirmación solicitando al usuario que ingrese algún texto o dato aleatorio
  - c. Se registra como fallecido al individuo en la base de datos
  - d. Se elimina al individuo de la tabla de individuos vivos
7. Consultar estados anteriores de las colecciones o individuos
- a. Buscar la colección o el individuo por cualquiera de sus parámetros
  - b. Mostrar una vista con el estado actual
  - c. Mostrar un control para acceder al historial
  - d. Mostrar un control que permita filtrar los estados anteriores según un criterio dado
8. Mostrar tareas programadas
- a. Mostrar una tabla con las tareas programadas
  - b. Mostrar un control que permita filtrar las tareas dado un criterio
9. Añadir una tarea
- a. Mostrar un formulario con la información requerida
  - b. Notificar a los trabajadores asignados a dicha tarea
  - c. Registrar la tarea en la base de datos
10. Reprogramar una tarea
- a. Mostrar el control que permite acceder al formulario
  - b. Mostrar un formulario con la información actual y habilitar su edición
  - c. Notificar a los trabajadores asignados del cambio en la tarea



- d. Actualizar la tarea en la base de datos
11. Reasignar una tarea
    - a. Mostrar el control que permita acceder al formulario
    - b. Mostrar el formulario con la información actual y habilitar su edición
    - c. Notificar al personal involucrado
    - d. Actualizar la información en la base de datos
  12. Eliminar una tarea
    - a. Mostrar una tabla con las tareas programadas
    - b. Mostrar el control que habilita la eliminación
    - c. Mostrar una confirmación de eliminación
    - d. Notificar a los trabajadores asignados de la cancelación de la tarea
    - e. Eliminar la tarea en la base de datos
  13. Evaluar una tarea como cumplida
    - a. Mostrar un control que permita la acción
    - b. Registrar la fecha actual
    - c. Actualizar la información en la base de datos
  14. Mostrar lista del personal
    - a. Mostrar una tabla con los datos del personal
    - b. Mostrar un control que permita filtrar los datos dado un criterio
  15. Añadir nuevo personal
    - a. Mostrar una tabla con los datos del personal
    - b. Mostrar un formulario con la información requerida
    - c. Registrar la persona en la base de datos
  16. Modificar información del personal
    - a. Mostrar una tabla con los datos del personal
    - b. Mostrar el control que accede a una vista detallada de la información de la persona
    - c. Mostrar un formulario con la información actual para su edición
    - d. Actualizar la información en la base de datos
  17. Eliminar un miembro del personal
    - a. Mostrar una tabla con los datos del personal
    - b. Mostrar el control que habilita la eliminación
    - c. Mostrar una confirmación de eliminación
    - d. Eliminar la persona en la base de datos
  18. Cambiar contraseña de acceso
    - a. Mostrar una vista con los campos requeridos

- b. Realizar una verificación de la contraseña anterior
  - c. Notificar al usuario del resultado de la operación, sea este positivo o negativo
19. Reiniciar la contraseña de acceso
- a. Comprobar la identidad de la persona solicitante del servicio
  - b. Mostrar una vista con los usuarios del sistema
  - c. Mostrar un control que permita restablecer la contraseña de dicho usuario solo si es el administrador del sistema

### 2.2.2: REQUISITOS NO FUNCIONALES

- ❖ Interfaz de usuario
  - Minimalista
  - Simple de usar
- ❖ Usabilidad
  - Facilidad de uso para personas con poca experiencia en el uso de las computadoras
  - Controles simples e intuitivos
- ❖ Rendimiento
  - Tiempo de respuesta inferior a un segundo
  - Disponibilidad en múltiples sistemas operativos
  - Eficiencia en el uso de recursos de hardware
- ❖ Hardware
  - Servidor
    - ✓ Sistema operativo Windows 7 (o superior) o Linux
    - ✓ Procesador de dos núcleos o superior
    - ✓ 2gb de RAM
    - ✓ Conexión a Internet
  - Cliente
    - ✓ Conexión a la red
- ❖ Software
  - Servidor
    - ✓ Node.js v18.2.0
    - ✓ Express v4.18.1
    - ✓ Bcrypt v5.0.1
    - ✓ Cors v2.8.5
    - ✓ JsonWebToken v8.5.1
    - ✓ Sequelize v6.21.4

- ✓ PostgreSQL v14.2.0
- ✓ React v18.2.0
- ✓ Bootstrap v5.2.0
- ✓ React-Bootstrap v2.5.0
- ✓ Axios v0.27.2
- ✓ Leaflet v1.9.2
- ✓ React-Leaflet v4.1.0
- ✓ Formik v2.2.9
- ✓ Yup v0.32.11
- ✓ Moment v2.29.4
- ✓ React-Icons v4.4.0
- ✓ React-Leaflet-Draw v0.20.4
- Cliente
  - ✓ Navegador web Mozilla Firefox 90.0 o Google Chrome 90.0 o versiones superiores

### 2.3: MODELADO DEL NEGOCIO

La ingeniería no es un proceso estático y está altamente condicionado por la persona que la realiza, esto implica comprender el hecho objetivo de que dos personas no piensan igual y que se requiere de un lenguaje universal que permita el entendimiento y vaya más allá de toda barrera comunicacional.

Los diagramas y modelos son la herramienta elegida por la ingeniería para solucionar este problema y son el elemento más importante para asegurar la continuidad de una investigación, así como su desarrollo, pues el conocimiento ha de compartirse en un lenguaje comprensible y útil para todos. Son una representación visual de la realidad que pretende transmitir una idea y debe seguir para ello un estándar que asegure su comprensión.

Durante esta investigación se emplearon diferentes tipos de representaciones y modelos utilizando diagramas BPMN, UML y otras representaciones empíricas para facilitar el entendimiento del proceso de gestión del flujo de información del Jardín Botánico de Matanzas

## Diagrama del negocio (BPMN):

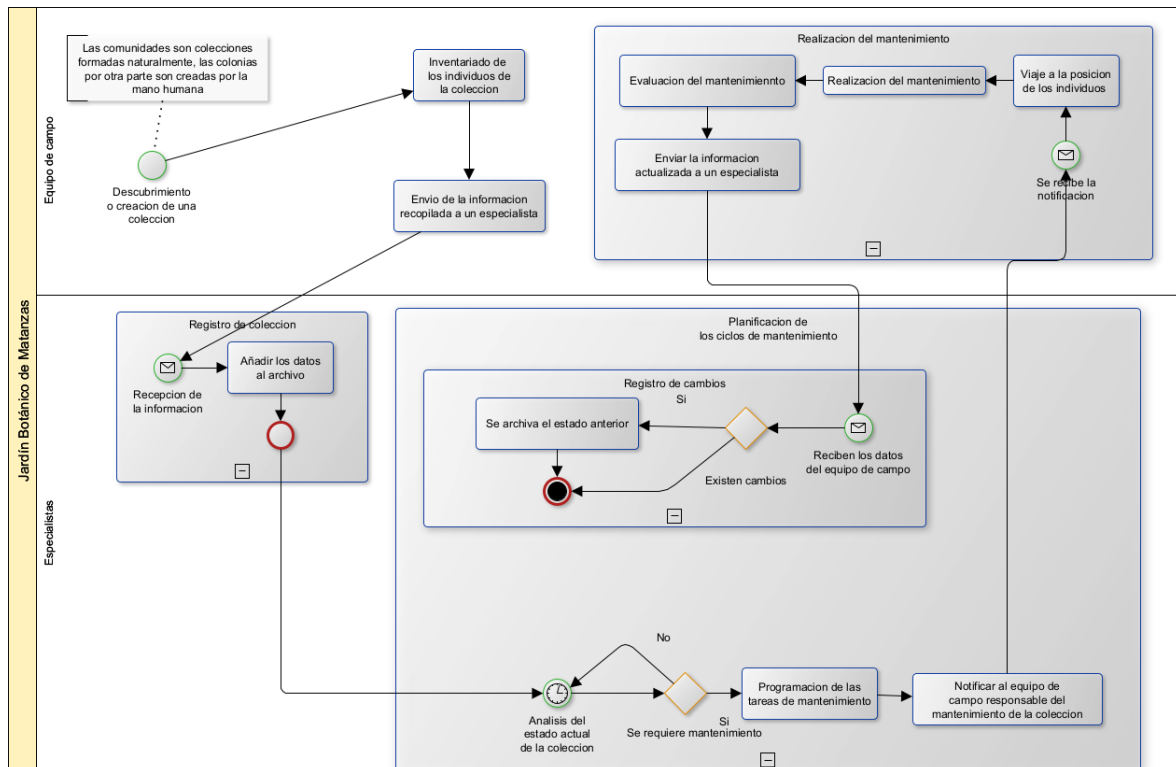


Ilustración 2: Modelo BPMN del proceso de trabajo del Jardín Botánico de Matanzas

### 2.4: DIAGRAMA DE CASOS DE USO DEL SISTEMA

Para una mejor comprensión de las responsabilidades de cada persona que interviene en el negocio y que acciones realiza en el sistema de forma que en un posible caso de mantenimiento o extensión se haga una revisión más detallada, o en caso de cambios en los actores sea posible entender de forma sencilla que responsabilidades poseen los roles actuales se toma la decisión de tomar un artefacto que no se define para una metodología de desarrollo ágil, el diagrama de casos de uso del sistema. A continuación, se expone el mencionado diagrama

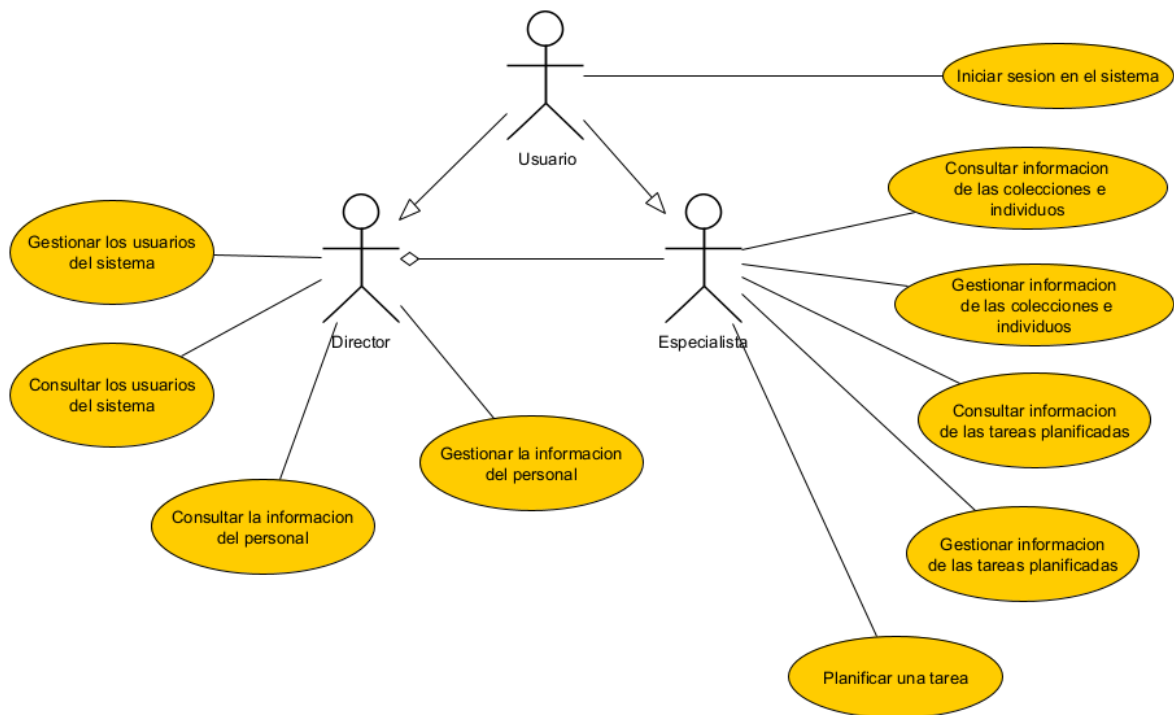


Ilustración 3: Diagrama de casos de uso del sistema

## 2.5: HISTORIAS DE USUARIO

En las historias de usuario (HU) se especifican los requisitos que debe cumplir el software. El cliente da una descripción de lo que debe hacer el software junto con el programador para llegar a un acuerdo de que debe hacerse y como debe hacerse

Tomando esto en cuenta se estableció una escala de prioridad en el negocio:

- Alta: Asignada a aquellas funcionalidades esenciales que el cliente define como imprescindibles
- Media: Funcionalidades de menor importancia para el funcionamiento del sistema las cuales el cliente denomina opcionales o de comodidad
- Baja: Funcionalidades no asociadas al funcionamiento directo del sistema, sino que constituyen una ayuda al control del mismo sin que

A continuación, se muestra una tabla resumen con las historias de usuario y su prioridad

Se define la prioridad (P) de una historia de usuario

Tabla 1: Historias de usuario

No HU	Historia de usuario	P
1	Gestionar colecciones	Alta

2	Gestionar individuos	Alta
3	Gestionar personal	Baja
4	Gestionar tareas de mantenimiento	Alta
5	Gestionar usuarios del sistema	Alta
6	Autenticarse en el sistema	Media
7	Gestionar mapa	Alta

Según las funcionalidades estas son las historias de usuario elaboradas por el cliente designado como de alta prioridad

Gestionar colecciones:

Historia de Usuario	
Número: 1	Usuario: Toni
Nombre historia: Gestionar colecciones	
Programador responsable: Adrián Almeida García	
Descripción: El usuario puede insertar, editar o eliminar colecciones de individuos	
Observaciones: El usuario de estar autenticado.	

Gestionar individuos:

Historia de Usuario	
Número: 1	Usuario: Toni
Nombre historia: Gestionar individuos	
Programador responsable: Adrián Almeida García	
Descripción: El usuario puede insertar, editar o eliminar individuos al sistema	
Observaciones: El usuario de estar autenticado.	

Gestionar tareas de mantenimiento

Historia de Usuario	
Número: 1	Usuario: Toni
Nombre historia: Gestionar tareas de mantenimiento	
Programador responsable: Adrián Almeida García	
Descripción: El usuario puede insertar, editar o eliminar, reprogramar, asignar y reasignar tareas de mantenimiento así como poder darlas por cumplidas	

Observaciones: El usuario de estar autenticado.

Gestionar usuarios del sistema

Historia de Usuario	
Número: 1	Usuario: Toni
Nombre historia: Gestionar usuarios del sistema	
Programador responsable: Adrián Almeida García	
Descripción: Como no todos el personal tiene acceso al sistema, el administrador debe poder insertar, editar y eliminar usuarios del sistema. También debe poder reiniciar las credenciales de acceso a un usuario en caso de que se pierdan	
Observaciones: El usuario de estar autenticado. El usuario debe ser el administrador del sistema	

Gestionar mapa

Historia de Usuario	
Número: 1	Usuario: Toni
Nombre historia: Gestionar mapa	
Programador responsable: Adrián Almeida García	
Descripción: El usuario puede consultar la información del mapa aéreo de forma que le permita mejorar su toma de decisiones	
Observaciones: El usuario de estar autenticado.	

## 2.6: DISEÑO CONCEPTUAL DE LA BASE DE DATOS

La base de datos provee la persistencia de la información y su posterior uso. Para esta investigación se le dio prioridad a la conservación de toda la información útil así como de sus estados anteriores. Se toma en cuenta la necesidad del Jardín Botánico de Matanzas de crear una biblioteca de información sobre el desarrollo y evolución de sus individuos y colecciones durante el paso del tiempo que facilite el estudio de estas por parte del propio Jardín Botánico de Matanzas o de terceros.

Teniendo en cuentas los requisitos y diseño del sistema se llegó a este modelo conceptual de la base de datos que responde a las necesidades del Jardín Botánico de Matanzas y

asegura la persistencia de la información generada así como su disponibilidad en todo momento

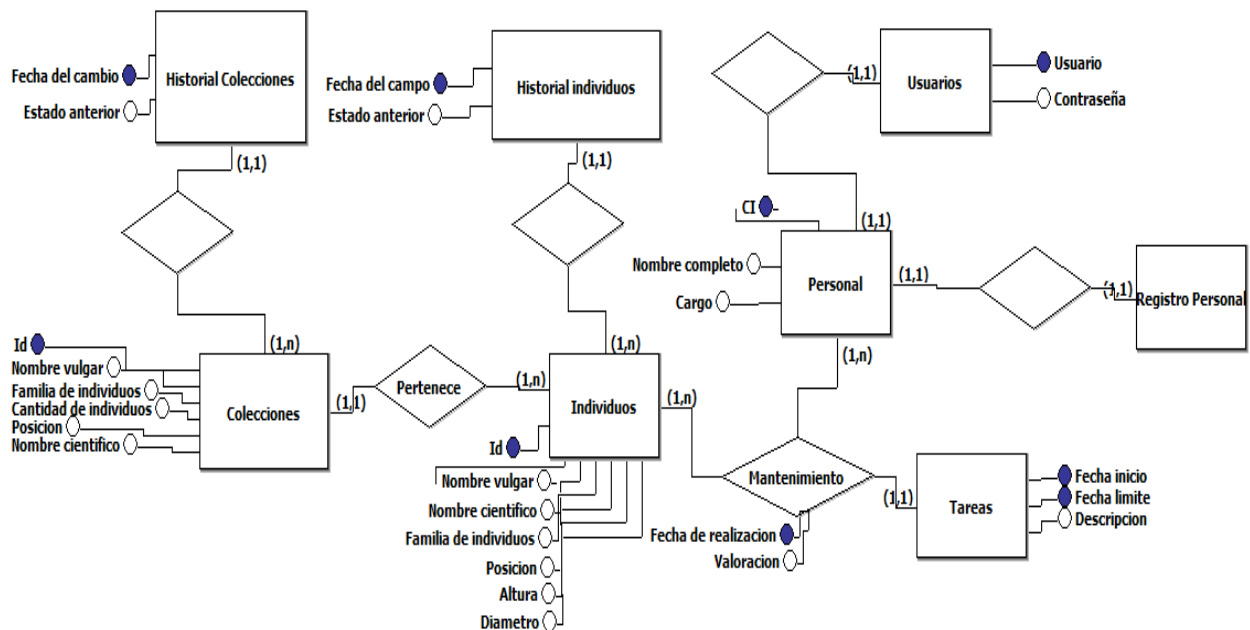


Ilustración 4: Modelo conceptual de la base de datos

## 2.7: CONCLUSIONES PARCIALES DEL CAPÍTULO

Luego de describir el diseño de la solución propuesta al problema científico de esta investigación se concluye que:

1. En este capítulo se especificaron los requerimientos del sistema a desarrollar, así como mediante el uso de los artefactos de la ingeniería de software se obtuvo un modelo preciso del negocio lo que permite al desarrollador tener una visión general de los principales conceptos del objeto de estudio, además fue fundamental para lograr un diseño más apropiado para la realización del proyecto.
2. Se enmarcaron las historias de usuario y su prioridad en el proceso de desarrollo y se obtuvo un modelo conceptual de la base de datos del sistema



## CAPÍTULO 3: PROPUESTA DE SOLUCIÓN PRÁCTICA AL PROBLEMA CIENTÍFICO

### 3.1: INTRODUCCIÓN

Ante la problemática de desarrollar una aplicación web para la gestión del flujo de información del Jardín Botánico de Matanzas se llevó a cabo un proceso de desarrollo de software siguiendo la metodología XP del cual se obtuvo un producto de software usando herramientas modernas pero que a su vez fueran de software libre, teniendo como premisa la soberanía tecnológica del país.

### 3.2: DESCRIPCIÓN DE LA SOLUCIÓN

Seleccionado el tipo de aplicación a desarrollar y las herramientas se procede a la codificación de la aplicación cuyo objetivo es dar solución al problema científico.

Se comenzó por definir las capas que integraran la aplicación, estas permiten segregar cada responsabilidad y definir con mayor precisión qué y quien las realiza. Para ello se concibieron tres capas, la capa de la base de datos, la capa del servidor y la capa del cliente. Las responsabilidades de cada capa se exponen a continuación:

#### **Capa de la base de datos:**

- Dar persistencia a la información almacenada
- Proveer la información almacenada a solicitud
- Organizar los datos

#### **Capa del servidor:**

- Creación de la base de datos
- Conexión con la base de datos
- Intermediario entre el cliente y la base de datos
- Comunicación con la base de datos para la ejecución de operaciones sobre esta
- Comprobaciones de seguridad con respecto a la autenticidad de los usuarios del sistema
- Configuración inicial del sistema

#### **Capa del cliente:**

- Despliegue de la información en un formato legible

- Recopilación de la información necesaria para la creación de nuevas entradas en la base de datos
- Control del acceso a los módulos del sistema a través de la protección de rutas

### **Detalles importantes del proceso de desarrollo**

Con respecto al desarrollo de la aplicación cabe recalcar una serie de detalles que facilitan en entendimiento de dicho proceso así como el uso del producto final. A continuación se clasifican y exponen los más importantes, el resto está contenido en el manual de usuario de la aplicación. Se muestra además capturas de la aplicación que evidencian lo expuesto

#### **Base de datos:**

La conexión con la base de datos se establece usando un usuario con permisos reducidos, este usuario se crea usando una herramienta como pgAdmin o similares. Esto mantiene un nivel de seguridad en el sistema puesto que un atacante no tendría acceso total al sistema de gestión de bases de datos. El usuario solo cuenta con un permiso

- **Iniciar sesión en el sistema gestor de bases de datos:** Se emplea para el acceso a los datos de la base de datos propia del sistema y solo a la base de datos propia del sistema

Bajo ningún concepto se debe otorgar permisos extra al usuario de la base de datos, esto constituye una violación crítica de seguridad y no está cubierta por el sistema.

#### **Servidor:**

Para la comunicación con la base de datos se eligió emplear los métodos de consulta que ofrece Sequelize que permiten la ejecución de consultas SQL sin necesidad de que el desarrollador tenga que insertar código SQL para ello.

El servidor escucha por defecto al puerto 3001 del ordenador aunque cambiara si se especifica uno en las variables de entorno, esto responde a la necesidad de que en algún momento del ciclo de vida del software sea necesario cambiar el puerto de escucha, las peticiones http provenientes del cliente se envían a una dirección que soporta intermediarios proxy que redirijan dicha petición al destino deseado en caso de cambiar el puerto por defecto de la aplicación.

Para la codificación de la aplicación del lado del servidor se eligió usar el Modelo-Vista-Controlador enfocado a Node.js. Se escogió este enfoque buscando crear un código organizado y fácilmente legible que permita un rápido mantenimiento o extensión.

## Ciente:

El diseño de la interfaz de usuario sigue un estilo minimalista, de forma que cada vista pueda acoplarse en la pantalla evitando que sea necesario navegar vertical u horizontalmente en una vista.

Aunque existe un proyecto de aplicación móvil para este sistema que auxilie a los equipos de campo, durante el desarrollo de esta aplicación web se tomó en cuenta su uso en dispositivos móviles, haciendo hincapié en los dispositivos del fabricante Apple para los cuales el desarrollo de aplicaciones es una dificultad producto de restricciones tecnológicas debido a las sanciones impuestas contra Cuba. Esto se logró mediante el uso de la librería de CSS, Bootstrap la cual automatiza la técnica conocida como “responsive design” que consiste en adaptar la vista a la pantalla del dispositivo que la consulta.

Se utilizó la gama de colores propia de Bootstrap que son azul, blanco, gris, amarillo y rojo como colores matices para el estilizado de las vistas y controles.

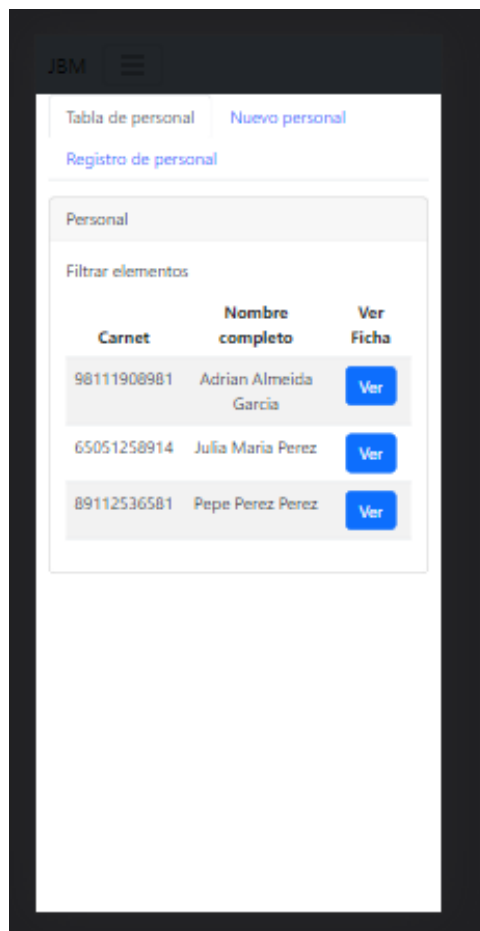
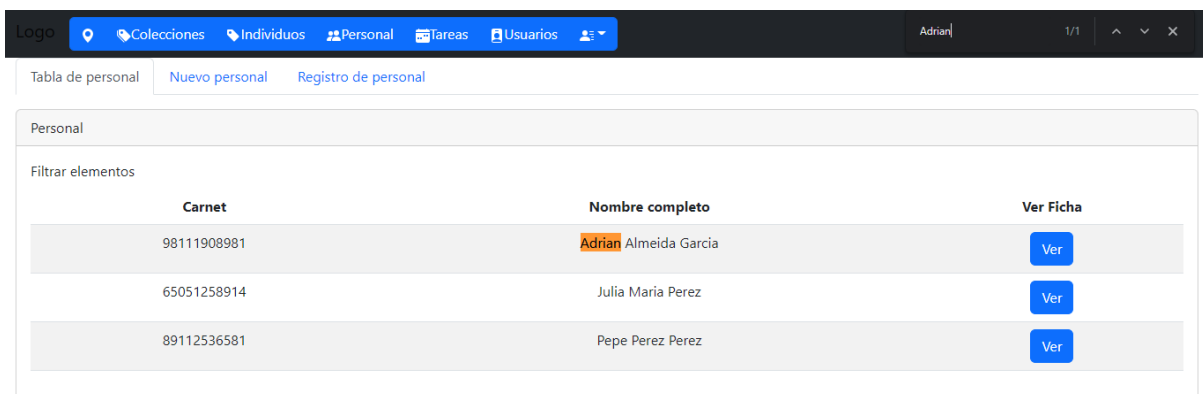


Ilustración 5: Vista de la aplicación en modo iPhone12 Pro Max

Para el despliegue de los datos extraídos de la base de datos se emplean dos enfoques, citados a continuación:

- Tablas para el despliegue de conjuntos de datos
- Perfiles para el despliegue de información detallada de elementos individuales

Las tablas están construidas de forma que soporta el uso de la función de búsqueda nativa del navegador web, la cual es accesible mediante un atajo del teclado (*Ctrl+F en Google Chrome y Mozilla Firefox*) por lo que no fue necesario implementar una búsqueda para la aplicación siguiendo la buena práctica de programación de “no reinventar la rueda”.



The screenshot shows a web browser window with a dark theme. The address bar contains the URL 'logos'. The browser's navigation bar includes tabs for 'Colecciones', 'Individuos', 'Personal', 'Tareas', and 'Usuarios'. The 'Personal' tab is active. Below the navigation bar, there are three sub-tabs: 'Tabla de personal', 'Nuevo personal', and 'Registro de personal'. The main content area displays a table titled 'Personal' with a search filter 'Filtrar elementos'. The table has three columns: 'Carnet', 'Nombre completo', and 'Ver Ficha'. The first row shows '98111908981' in the 'Carnet' column, 'Adrian Almeida Garcia' in the 'Nombre completo' column, and a blue 'Ver' button in the 'Ver Ficha' column. The second row shows '65051258914' in the 'Carnet' column, 'Julia Maria Perez' in the 'Nombre completo' column, and a blue 'Ver' button in the 'Ver Ficha' column. The third row shows '89112536581' in the 'Carnet' column, 'Pepe Perez Perez' in the 'Nombre completo' column, and a blue 'Ver' button in the 'Ver Ficha' column.

Carnet	Nombre completo	Ver Ficha
98111908981	Adrian Almeida Garcia	Ver
65051258914	Julia Maria Perez	Ver
89112536581	Pepe Perez Perez	Ver

Ilustración 6: Vista de la integración con la función de búsqueda del navegador

La validación de la información se ocupa de asegurar que no se envíen datos erróneos a la base de datos, manteniendo la integridad de la información almacenada en esta. Funciona de forma en la que se definen parámetros para los cuales el formulario es válido, de no cumplirse este formulario no se enviara en la petición HTTP y por tanto no se inserta en la base de datos. La validación de la información se realiza mediante el uso de dos técnicas diferentes las cuales se exponen a continuación:

- **Validación asíncrona:** Se emplea en los campos de cada formulario, esta verifica que la información de estos campos sea correcta mediante el uso de la librería Yup, la diferencia de este estilo es que se espera a que el usuario libere el control del formulario para aplicar la validación

Carnet de identidad:

El carnet de identidad debe contener 11 caracteres

Ilustración 7: Validación asíncrona

- **Validación síncrona:** Se emplean una vez el formulario está confeccionado, ejecutan comprobaciones sobre los datos que se pretenden enviar antes de que se realice dicho envío. Las comprobaciones realizadas mediante esta técnica son más complejas y en caso de ser negativas anulan el envío del formulario y reinician su instancia

localhost:3000 says  
El carnet de identidad no es correcto  
OK

**Personal**

Carnet de identidad:  
98023012457

Nombre completo:  
Juan Carnet Falso

Correo electrónico:  
a@a.com

Telefono  
53535353

Cargo  
falsificador de carnet de identidad

Añadir Cancelar

Ilustración 8: Validación síncrona

### 3.3: PRUEBAS DE SOFTWARE

Es el proceso de verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa; probando el comportamiento del mismo con el fin proporcionar información objetiva e independiente sobre la calidad del producto. Es una actividad más en el proceso de control de calidad.

### 3.3.1: MÉTODOS EMPLEADOS PARA LA REALIZACIÓN DE LAS PRUEBAS.

El procedimiento para la liberación de un software en la empresa XETID tiene varios métodos bien definidos, es por ello que las realizaciones de las pruebas se sustentaron sobre las bases de dichos métodos los cuales sirvieron como guía a la hora de realizar este proceso de prueba:

- Pruebas de Aceptación.
- Pruebas Funcionales.
- Pruebas Unitarias.
- Pruebas de Integración.
- Prueba de Caja Negra.

**Pruebas de aceptación:** son aquellas pruebas diseñadas por el propio equipo de desarrollo en base a los requisitos funcionales especificados en la fase de análisis para cubrir todo ese espectro, y ejecutadas por el propio usuario final, no por todos evidentemente, pero sí por una cantidad de usuarios finales significativo que den validez y conformidad al producto que se les está entregado en base a lo que se acordó inicialmente. Las pruebas de aceptación fueron realizadas por los clientes los cuales utilizaron juegos de datos en cada una de las interfaces seleccionadas en este módulo, dado que el módulo funcionó correctamente, con todos los juegos de datos, el cliente demuestra su satisfacción con el producto.

**Pruebas funcionales:** El objetivo de la prueba funcional es validar cuando el comportamiento observado del software probado cumple o no con sus especificaciones. Para esto es necesario contar con las especificaciones del producto y una versión ejecutable del mismo. Se revisan las especificaciones para asegurarse de que se pueden definir los casos de prueba a partir de ellas, en el caso de que no existan especificaciones o que las mismas estén incompletas, se trabaja junto al cliente y los desarrolladores para generarlas o mejorarlas.

**Pruebas unitarias y de integración:** Las pruebas unitarias están basadas en hacer pruebas de pequeños fragmentos de código. Por lo que cada prueba tiene que ser lo más independiente posible de las otras y encargarse de una acción específica, en programación orientada a objetos se puede afirmar que estas unidades son los métodos o las funciones que tenemos definidos. El objetivo de las pruebas unitarias es el aislamiento de partes del código y la demostración de que estas partes no contienen errores. Una vez que se han aprobado las pruebas unitarias se llevan a cabo las pruebas de integración las cuales consisten en realizar pruebas para verificar que un gran conjunto de partes de software funciona junto.

**Pruebas de caja negra:** Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales del programa. En ellas se ignora la estructura de control o sea que no precisa definir ni conocer los detalles internos de su funcionamiento, sino que se concentra en los requisitos funcionales del sistema, pues deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz. Estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

### 3.3.2: ESTRATEGIA DE PRUEBAS PARA HU

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

Tabla 2: Plan de pruebas por historia de usuario

HU	Historia de usuario	Pruebas a realizar
1	Gestionar colecciones	Insertar colección Editar colección Eliminar colección
2	Gestionar individuos	Insertar individuo

		Editar individuo Eliminar individuo
3	Gestionar personal	Insertar personal Editar personal Eliminar personal
4	Gestionar tareas de mantenimiento	Programar tarea Reprogramar tarea Eliminar tarea
5	Gestionar usuarios del sistema	Insertar usuario Reiniciar contraseña de acceso Eliminar usuario
6	Autenticarse en el sistema	Comprobar contraseña Verificar inicio de sesión automático con token de acceso vivo
7	Gestionar mapa	Verificar correcto despliegue del mapa

**Definir casos de pruebas:**

Se define (-) como la prueba que aplica para todas las historias de usuario

Tabla 3: Pruebas realizadas y resultados

HU	Prueba a realizar	Resultado
6	Acceder al sistema con credenciales erróneas	Error: Credenciales incorrectas
6	Acceder al sistema con credenciales correctas	Accede al sistema y obtiene un token de acceso valido por 8 horas
6	Cerrar el navegador e intenta acceder nuevamente con un token valido	Accede al sistema, el token no incrementa su tiempo de vida
6	Acceder al sistema con un token no valido	Redirección a la vista de inicio de sesión



-	Enviar un formulario sin llenar un campo obligatorio	El formulario no se envía al presionar el botón "Añadir"
2	Insertar un valor no numérico en el campo de altura y diámetro	El valor no se despliega puesto que los campos no soportan valores no numéricos
2	Insertar un valor negativo en los campos de altura y diámetro	Error en la validación asincrónica: Los campos altura y diámetro deben ser números positivos  El formulario no se envía al presionar el botón "Añadir individuo"
2	Eliminar un individuo sin especificar motivo	Error en la validación asincrónica: Debe especificarse un motivo  La operación de eliminación no se completa
2	Eliminar un individuo especificando un motivo	Se elimina el individuo y su último estado se añade al registro
3	Insertar un número de carnet de identidad con una fecha de nacimiento inválida	Error de validación: El carnet de identidad no es válido
3	Insertar un valor numérico en el campo "Nombre Completo"	Error de validación asincrónica: El campo no puede contener números
3	Insertar un valor que no cumpla con el formato de un correo electrónico	Error de validación asincrónica: El correo electrónico no es válido
3	Insertar un número de teléfono móvil incorrecto	Error de validación asincrónica: El teléfono no es válido
3	Insertar un valor no numérico en el campo "Teléfono"	Error de validación asincrónica: El teléfono no puede contener letras
4	Insertar una tarea con una fecha de inicio o de	Error de validación asincrónica: Las fechas no deben ser anteriores al día actual

	culminación anterior al día actual	
4	Insertar una fecha de culminación anterior a la fecha de inicio de la tarea	Error de validación : La fecha de culminación no puede ser anterior a la fecha de inicio de una tarea
7	Despliegue del mapa sin conexión a internet	Se despliegan todos los marcadores pero el mapa se despliega parcialmente con los datos que se encuentran en el cache del navegador.

### 3.4: RESULTADOS DE LAS PRUEBAS REALIZADAS

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

### CONCLUSIONES PARCIALES

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

1. Se realizó una descripción detallada de la aplicación especificando cada uno de sus componentes.
2. Para garantizar una mejor calidad del software fue necesario la realización de las pruebas, donde los resultados arrojados por las mismas permitieron conocer algunos errores y erradicarlos para así obtener un software de mayor calidad y entregar al cliente un producto que cumpla con sus expectativas

## CONCLUSIONES

Como resultado de esta investigación se dio cumplimiento a los objetivos trazados arribando a las siguientes conclusiones:

1. El objetivo principal de esta investigación se centró en proveer a los especialistas del Jardín Botánico de Matanzas de una herramienta informática especializada para auxiliarlos en esta noble tarea. Mediante un contacto estrecho con los profesionales que laboran en el Jardín Botánico de Matanzas se pudo establecer un conjunto de investigaciones precedentes, tanto de Cuba como del mundo para servir de bases para el desarrollo de la presente investigación. Se estableció la filosofía de trabajo basándose en la metodología ágil de desarrollo de software Programación Extrema (XP) y posteriormente se seleccionaron las herramientas de software y hardware que se utilizaron para la elaboración del producto final.
2. Posteriormente se definieron los requisitos en presencia de los profesionales que laboran en el Jardín Botánico de Matanzas de forma que pudieran exponer con sus palabras como debería funcionar el sistema y que responsabilidades eran más importantes y requerían una atención priorizada. Con esta información se obtuvo un modelo del negocio que describe los procesos del Jardín Botánico de Matanzas, así como un diseño conceptual de la base de datos.
3. El desarrollo dio como fruto una aplicación web desarrollada con tecnologías libres cuyas licencias permiten el uso comercial teniendo en cuenta la soberanía tecnológica necesaria para la informatización de la sociedad cubana, proceso al que esta aplicación se acoge cumpliendo su objetivo de ser una herramienta especializada creada para satisfacer las necesidades del Jardín Botánico de Matanzas. La aplicación cuenta con un diseño minimalista y puede ser ejecutada en cualquier dispositivo que posea un navegador web, cumpliendo con los requisitos de simplicidad y multiplataforma solicitados por el cliente. Luego se ejecutaron una serie de pruebas a la aplicación web para validar la correcta implementación de las funcionalidades las cuales dieron resultados positivos.

Por lo anteriormente expuesto se considera que se ha dado solución al problema científico de esta investigación puesto que la aplicación desarrollada permite que el flujo de información del Jardín Botánico de Matanzas pueda ser controlado por sus especialistas mediante una herramienta especializada, moderna y multiplataforma.

## 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:

- Implementar una funcionalidad de búsqueda propia para la aplicación de forma que la función nativa del navegador no sea necesaria
- Incluir funcionalidades de manejo de imágenes
- Incluir funcionalidades de dibujo de polígonos en el mapa y el uso de datos espaciales

## REFERENCIAS

1. **Microsoft**. Estilo de arquitectura de n niveles. [En línea] 23 de noviembre de 2022. <https://learn.microsoft.com/es-es/azure/architecture/guide/architecture-styles/n-tier>.
2. **Documentacion de Sequelize**. [En línea] [Citado el: 1 de noviembre de 2022.] <https://sequelize.org/docs/v6/>.
3. **Puerta Fernández, Juan Francisco, Yáñez Sarmiento, Mariana Marisol, & Medina Peña, Rolando**. (2018). Scielo. [En línea] 3 de febrero de 2018. [Citado el: 22 de junio de 2022.] [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S2218-36202018000200209&lng=es&tlng=es..](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202018000200209&lng=es&tlng=es..)
4. **Programación de aplicaciones web: historia, principios básicos y clientes web**. Sergio Lujan-Mora. Alicante : Editorial Club Universitario, 2002.
5. **Lujan-Mora, Sergio**. *Programacion en Internet*. Alicante : Club Universitario, 2001.
6. *Encuesta de percepción pública sobre valores y conservación de la flora cubana*:. L. Granado, J.A. García, A. Palmarola, D. Barrios, L. González-Oliva, L.R. González-Torres, M. 3, La Habana : s.n., 2013, Bissae, Vol. 7.
7. **Cumbre de los pueblos**. CumbrePueblos. [En línea] septiembre de 2017. [Citado el: 22 de junio de 2022.] <https://cumbrepuebloscop20.org/medio-ambiente/proteccion/>.
8. **Bustamante, Dayana y Rodriguez, Jean C**. *Metodología Actual. Metodología XP*. Barinas : s.n., 2014.
9. **Beck, Kent**. *Extreme Programming Explained: Embrace Change*. 1999.
10. **Pressman, Roger S**. *Ingeniería de software. Un enfoque practico Ed.7*.
11. **Martínez, M.C Saúl González Campos y M.C Luis Felipe Fernadez**. *Programación Extrema: Prácticas, aceptación y controversia*. Juarez : Universidad Autonoma de Cd. Juarez.
12. **Grupo ISSI**. *Metodologías agiles en el desarrollo de software*. Alicante : s.n., 2003.
13. **Documentacion de React-Bootstrap**. [En línea] [Citado el: 12 de septiembre de 2022.] <https://react-bootstrap.github.io/>.
14. **Documentacion de NodeJS**. [En línea] [Citado el: 1 de noviembre de 2022.] <https://nodejs.org/en/docs/>.
15. **Documentacion de ReactJS**. [En línea] [Citado el: 23 de octubre de 2022.] <https://es.reactjs.org/docs/getting-started.html>.

## GLOSARIO DE TÉRMINOS

**CRUD:** Acrónimo de Create, Read, Update, Delete. Resume las operaciones básicas que se realizan en un sistema de gestión de la información: insertar, leer, actualizar y borrar.

**JSON:** JavaScript Object Notation es el formato de nombrado de los objetos en JavaScript de forma que cada objeto tiene una propiedad y un valor