



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
Facultad de Ciencias de la Computación

**Aplicación web para catalogar y difundir la
biodiversidad de un parque natural de la Ciudad
de Puebla**

TESIS

Para obtener el título de
Licenciado en Ciencias de la Computación

Presenta

María del Lourdes Camacho Castro

Director y Asesor

M. I. Carlos Armando Ríos Acevedo

Puebla, Pue. Mayo 2021

ÍNDICE

ÍNDICE	2
ÍNDICE DE FIGURAS	5
AGRADECIMIENTOS	6
INTRODUCCIÓN.....	7
CAPÍTULO I. DESCRIPCIÓN DEL PROYECTO.....	8
1.1 Objetivo general.....	8
1.2 Objetivos específicos.....	8
1.3 Metodología	8
1.4 Estado del campo o del arte.....	9
1.5 Resultados esperados	9
CAPÍTULO II. MARCO TEÓRICO	10
2.1 SCRUM	10
2.1.1 Equipo [8]	10
2.1.2 Eventos	11
2.1.3 Artefactos [7].....	12
2.2 BPMN (<i>Business Process Model and Notation</i>).....	12
2.3 UML (<i>Unified Modeling Language</i>)	12
2.3.1 Artefactos UML	13
2.4 Base de datos.....	13
2.4.1 Modelo relacional	13
2.4.2 Modelo Entidad-Relación	13
2.5 SQL.....	14
2.6 MySQL.....	14
2.7 PHP.....	14
2.8 Aplicación web.....	14
2.9 CodeIgniter.....	14
2.10 Grocery CRUD.....	14
CAPÍTULO III. ANÁLISIS Y DISEÑO	15
3.1 Especificación de requerimientos	15
3.1.1 Requerimientos funcionales	15
3.1.2 Requerimientos no funcionales	15

3.1.3	Requerimientos de interfaz.....	16
3.2	Análisis del Sistema	16
3.2.1	Diagrama de Casos de uso.....	17
3.2.2	Especificaciones de Casos de uso.....	18
3.3	Diseño del Sistema	28
3.3.1	Diagrama de clases.....	28
3.3.2	Diagramas de secuencia.	29
3.3.3	Diagrama BPM.....	35
3.3.4	Diagrama de paquetes	36
3.3.5	Diagrama de actividades	37
3.3.6	Diagrama de despliegue.....	38
3.3.7	Diagrama de flujo	38
3.3.8	Diseño de prototipo	40
3.4	Desarrollo Scrum.....	46
CAPÍTULO IV. DISEÑO DE BASES DE DATOS.....		50
4.1	Diseño lógico.....	50
4.1.1	Modelo Entidad-Relación	50
4.1.2	Diccionario de datos.....	52
4.2	Diseño físico.....	55
4.2.1	Diagrama Entidad-Relación Extendido.....	55
4.2.2	Código SQL.....	56
CAPÍTULO V. IMPLEMENTACIÓN Y PRUEBAS.....		57
5.1	Implementación	57
5.1.1	Administrador.....	57
5.1.2	Visitante	61
5.2	Pruebas	63
5.2.1	Pruebas de caja negra	63
CAPÍTULO VI. SEGURIDAD		68
6.1	Seguridad	68
CAPÍTULO VII. CONCLUSIÓN Y TRABAJO A FUTURO		69
7.1	Conclusión	69
7.2	Trabajo futuro.....	69
REFERENCIAS		70

ANEXO.....	71
-------------------	-----------

ÍNDICE DE FIGURAS

Figura 1. Proceso Scrum [7]	10
Figura 2. Diagrama de casos de uso.....	17
Figura 3. Diagrama de clases.....	28
Figura 4. Diagrama de secuencia de CU1.	29
Figura 5. Diagrama de secuencia de CU2.	30
Figura 6. Diagrama de secuencia de CU3	31
Figura 7. Diagrama de secuencia de CU4	32
Figura 8. Diagrama de secuencia de CU5.	33
Figura 9. Diagrama de secuencia de CU6.	34
Figura 10. Diagrama de secuencia de CU7.	34
Figura 11. Diagrama BPM.....	35
Figura 12. Diagrama de paquetes.	36
Figura 13. Diagrama de actividades.....	37
Figura 14. Diagrama de despliegue.	38
Figura 15. Diagrama de Flujo de Codelgniter. [11]	39
Figura 16. Visitante-Interfaz principal.....	40
Figura 17. Visitante-Interfaz de flora o fauna.	41
Figura 18. Visitante-Interfaz de flora o fauna.	41
Figura 19. Visitante-Interfaz del foro.	42
Figura 20. Visitante-Interfaz del contacto.	42
Figura 21. Visitante-Interfaz inicio de sesión para administrador.	43
Figura 22. Administrador-Interfaz gestión del parque.....	43
Figura 23. Administrador-Interfaz gestión de especie.	44
Figura 24. Administrador-Interfaz gestión de flora.	44
Figura 25. Administrador-Interfaz gestión de fauna.....	45
Figura 26. Administrador-Interfaz gestión del foro.	45
Figura 27. <i>Modelo Entidad-Relación.</i>	51
Figura 28. Diagrama Entidad-Relación Extendido.	55
Figura 29. Inicio de sesión (Administrador).....	57
Figura 30. Inicio (Administrador).....	58
Figura 31. Gestionar Parque.	58
Figura 32. Gestionar Especie.	59
Figura 33. Gestionar Flora.	59
Figura 34. Gestionar Fauna.	60
Figura 35. Gestionar Comentarios.....	60
Figura 36. Inicio (Visitante).	61
Figura 37. Consultar Flora.	61
Figura 38. Consultar Fauna.	62
Figura 39. Foro.	63

AGRADECIMIENTOS

Quiero expresar un sincero agradecimiento, en primer lugar, a mis padres y hermanos por todo su amor, comprensión y apoyo, pero sobre todo gracias infinitas por la paciencia que me han tenido y gracias a la vida por brindarme salud, fortaleza y capacidad.

Le agradezco también a mi asesor M. I. Carlos Armando Ríos Acevedo que gracias a su paciencia, consejos y correcciones hoy puedo culminar este trabajo.

Y por supuesto a mi querida Universidad y a todas las autoridades, por permitirme concluir con una etapa de mi vida.

No puedo dejar de agradecerte especialmente a ti Mary Alma, mi compañera fiel de Universidad, de tesina y ahora de corazón y vida.

INTRODUCCIÓN

Los parques de la ciudad de Puebla son de gran importancia ya que nos muestran la riqueza de la biodiversidad que existe en esta ciudad, además, como una conservación ambiental y como espacios públicos para la convivencia social y recreativa.

Las áreas verdes y espacios de esparcimiento son particularmente responsabilidad del municipio y con frecuencia, éste no cuenta con recursos para protegerlos, ocasionando que la ciudad crezca sin dotar a la población de nuevos espacios, ni conservar los existentes (Morales-García de Alba, 2009).

Por ello el cuidado de los parques de la ciudad de Puebla es responsabilidad de los administradores; adicionalmente, obtienen de las visitas recursos económicos ya que no cuentan con subsidios de parte del gobierno, de esta manera, no se ha hecho el esfuerzo para que los habitantes se relacionen con esto, puesto que en contexto muchos visualizan sólo el valor superficial de los parques. En ocasiones, las personas que acuden a estos no conocen los cuidados que deben tener estas áreas verdes, porque carecen de información de las diferentes especies de flora y fauna que existen en ellos.

La mayoría de parques de la ciudad de Puebla lleva un control de la flora y fauna que habita en ellos, pero sólo para uso exclusivo del parque. Por ello se creó una aplicación web que cataloga y difunde la biodiversidad que existe en la Laguna de San Baltazar Campeche de la Ciudad de Puebla, de esta manera más personas pueden conocer la importancia de la conservación de los parques. Es una herramienta de software compartida como código *open source* (código abierto), en los repositorios de GitHub permitiendo que cualquier persona interesada en el proyecto pueda utilizarla y adicionar nuevos módulos.

CAPÍTULO I. DESCRIPCIÓN DEL PROYECTO

1.1 Objetivo general

Desarrollar una aplicación web que permita catalogar y difundir la biodiversidad de un determinado parque de la ciudad de Puebla.

1.2 Objetivos específicos

- Investigar qué flora y fauna existe y así poder especificar cuáles son sus características para dar un mejor control.
- Desarrollar interfaces CRUD (*Create, Read, Update and Delete*) para el control de las especies.
- Desarrollar módulos para mostrar las diferentes especies que habitan en el parque.
- Diseño e implementación del foro para opiniones de los usuarios

1.3 Metodología

La metodología empleada en este proyecto es SCRUM, debido a que es una metodología ágil, se basa en cortos ciclos iterativos que van desde 2 semanas hasta un mes a este periodo se le llama *Sprint*, el resultado de cada *Sprint* deberá ser un producto listo para entregar y por ello cumple con varios aspectos importantes como los son [4]:

- Satisfacción del cliente: el cliente forma parte del equipo y así lo comprometen al resultado final, por ello que el cliente, en conjunto con el equipo de desarrollo, es quién define cuándo se hace y qué se hace.
- Simplicidad: todo lo que se hace en cada uno de los *Sprints* tiene en claro quiénes participan, el objetivo del *Sprint*, el tiempo que debe tomar y cuál es el resultado final.
- Inspección: existen varios tipos de eventos para la inspección, estos son: reunión diaria, revisión del *Sprint* y retrospectiva de este último. Estos eventos ayudan a llevar un mejor control de las tareas que se realizan en cada *Sprint*.
- Adaptación: Los cambios se pueden realizar en cualquier momento, incluso dentro del desarrollo de la ejecución de las diferentes *Sprints*, siempre y cuando no afecte el objetivo principal.

SCRUM cuenta con las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega.

De esta manera en cada *Sprint* se hará entrega de un producto de calidad, si un *Sprint* genera software inestable o con muchos errores se puede retroceder al último hito sin necesidad de volver al inicio.

Se aplicará el modelado de proceso de negocios (BPMN), según White & Miers (2009), en BPMN, los “Procesos de Negocio” involucran la captura de una secuencia de las actividades e información de apoyo, por consecuente, permitirá que se represente lo que se está haciendo en la aplicación para lograr el objetivo principal.

1.4 Estado del campo o del arte

De acuerdo al Programa de las Naciones Unidas para el Medio Ambiente (PNUMA) y a la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) (2015), las actividades humanas, magnificadas por el crecimiento de la población y el cambio climático, están provocando una pérdida considerable de biodiversidad en todo el mundo. Esta pérdida puede acarrear graves consecuencias, en particular para los sistemas de alimentación, la seguridad alimentaria y el bienestar humano. De mantenerse los actuales patrones de consumo y producción insostenibles, corremos el riesgo de enfrentarnos a una pérdida irreversible de biodiversidad.

Por lo tanto, se están creando herramientas tecnológicas para la conservación y cuidado de la biodiversidad, una de esas herramientas es la creación de apps o aplicaciones web que ayuden a identificar y a cuidar las especies de flora y fauna que existen en el mundo.

Existen 2 aplicaciones móviles que se asemejan a la aplicación web que se desarrollará, las cuales son:

- NaturaLista [5]. Es una app que ayuda al reconocimiento de biodiversidad según la ubicación en la que te encuentres. Te muestra fotografías y una pequeña descripción.
- Seek [6]. Es una app que ayuda a identificar las plantas y los animales a tu alrededor. Te muestra fotografías, una pequeña descripción, su taxonomía y especies similares.

El propósito de la aplicación web que se desarrollará permitirá visualizar la información como: su clasificación, sus cuidados, características, sobre las especies de flora y fauna que habitan en un determinado parque de la ciudad de Puebla; las personas que acudan al parque y visiten la aplicación web podrán colocar un comentario en un foro acerca de estas especies, ya sea opiniones para mejorar el parque o datos interesantes. A través, de los formularios que se llenaran por los encargados del parque la información será transferida a la base de datos de la aplicación web.

1.5 Resultados esperados

El resultado que se espera es una aplicación web con una interfaz intuitiva, que permita catalogar y difundir la biodiversidad de la Laguna de San Baltazar en la ciudad de Puebla

CAPÍTULO II. MARCO TEÓRICO

2.1 SCRUM

Schwaber & Sutherland señalan que, Scrum no es un proceso o una técnica para desarrollar productos; en vez de eso, es un marco de trabajo que emplea varios procesos y técnicas. Scrum muestra la capacidad relativa de las prácticas de gestión de producto y las prácticas de desarrollo de modo que podamos mejorar.

En la figura 1, se muestra el proceso de Scrum para entregar incrementos del producto hasta que se cumpla con todos los requerimientos del producto.

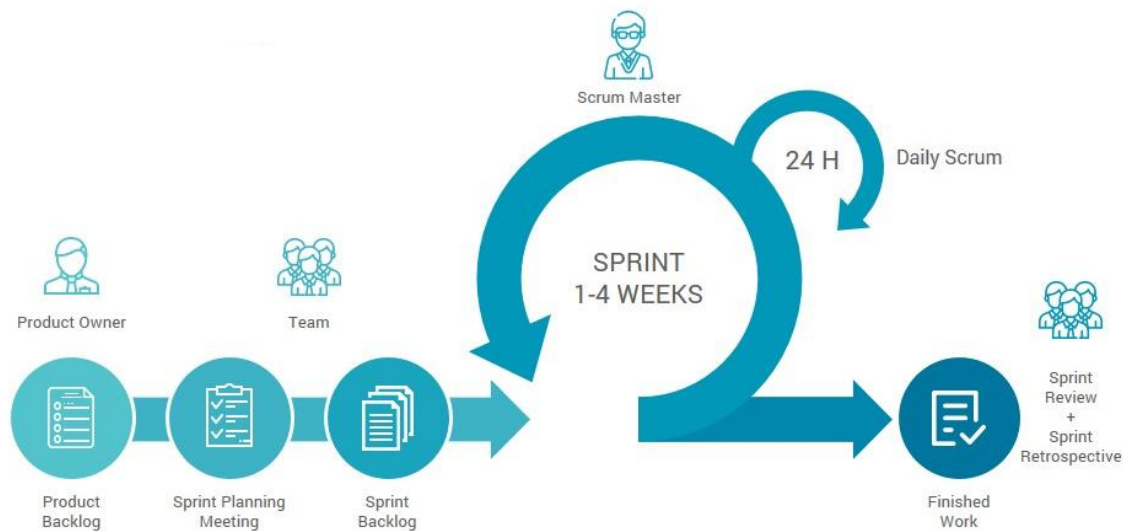


Figura 1. Proceso Scrum [7]

2.1.1 Equipo [8]

El equipo Scrum, es auto organizado y multifuncional debido a esto tienen la facilidad de entregar productos de forma iterativa e incremental teniendo así mayores oportunidades de retroalimentación.

- **El Dueño de Producto (Product Owner)**

Es el responsable del equipo, así como de potenciar el valor del producto. También, es el responsable de gestionar la Lista del Producto (Product Backlog), esto quiere decir que todo el equipo Scrum debe respetar sus decisiones.

- **El Equipo de Desarrollo (Development Team)**

Es un conjunto de profesionales que tienen como tarea entregar un incremento de producto en un estado de “terminado”, este incremento debe ser capaz de ponerse en producción al término de un sprint. Cabe mencionar que la organización es la encargada de asignar el trabajo de cada miembro del equipo.

- **El Scrum Master [7]**

Es el líder del equipo, este debe hacer que las reglas de Scrum se cumplan, así también, debe ser el intermediario del equipo Scrum con personas ajenas a él; ya que su interacción debe aportar valor al equipo.

2.1.2 Eventos

Los eventos de Scrum tienen como finalidad regular y disminuir reuniones no establecidas en el desarrollo de Scrum.

- **Sprint**

Es un lapso de una a cuatro semanas donde se crea un incremento del producto en estado “terminado”. El dueño del producto puede cancelar el Sprint antes de finalice su tiempo en colaboración del Equipo de Desarrollo o por el Scrum Master.

- **Planificación de Sprint (Sprint Planning)**

Para saber qué se tiene que hacer durante el Sprint se hace la planificación de Sprint la cual tiene una duración de hasta 8 horas si el Sprint dura un mes si es más corto la planificación también, el encargado de este evento es el Scrum Master.

- **Scrum Diario (Daily Scrum)**

Es una reunión con un lapso de al menos 15 minutos para que el Equipo de Desarrollo para poder concordar sus actividades y crear un plan para las siguientes 24 horas. El plan depende del último Scrum Diario ya que se necesita saber cómo va el Sprint.

- **Revisión de Sprint (Sprint Review)**

Al finalizar el Sprint se realiza una Revisión de Sprint, esto es para analizar el incremento y hacerle cambios a la Lista de Producto si esto lo requiere.

- **Retrospectiva de Sprint (Sprint Retrospective)**

Se verifica que el desarrollo del Sprint haya sido correcto si hubo errores o retrasos el Equipo Scrum tiene la oportunidad de crear un plan adecuado para el siguiente Sprint.

2.1.3 Artefactos [7]

Es la representación del trabajo o valor en diferentes formas, estos ayudan a dar transparencia y oportunidades para la inspección y adaptación en cada Sprint.

- **Lista de Producto (Product Backlog)**

Es una lista de los requisitos del producto. El responsable de esta lista es el Dueño del Producto el cual debe de ser consiente del contenido, de su disponibilidad y de la forma en la que los requisitos se encuentran.

La Lista de Producto puede evolucionar dependiente del desarrollo de los Sprints y los cambios que se hagan, por ello se considera que es dinámica.

- **Lista de tareas del Sprint (Sprint Backlog)**

Es un listado de tareas suficientemente detallado para que el Sprint se lleve a la práctica de manera correcta y que en los Scrum Diarios se comprenda qué y cómo se va a realizar a lo largo de cada Sprint.

2.2 BPMN (*Business Process Model and Notation*)

En el 2001, se comenzó a desarrollar BPML (Lenguaje de Modelo de Procesos de Negocios) por BPMI.org, de este lenguaje surgió la necesidad de representarlo gráficamente. BPMI.org y el *Notation Working Group* desarrollaron BPMN 1.0 y su especificación fue publicada en mayo de 2004, de ahí en febrero de 2006 esta especificación fue adoptada como un estándar en el OMG (*Object Management Group*). Para febrero de 2008 la OMG publicó la versión final de BPMN 1.1 y para el 2009 salió a la luz la versión BPMN 2.0. [9]

De acuerdo con White y Miers, los “Procesos de Negocio”, en BPM son utilizados para la captura de una secuencia de actividades e información de apoyo. Modelar un Proceso de Negocio involucra representar cómo una empresa realiza sus objetivos centrales.

2.3 UML (*Unified Modeling Language*)

En 1996 la colaboración de las organizaciones Digital Equipment Corporation, Hewlett-Packard, I-Logix, Intellicorp, I3M, ICONComputing, MCI Syserrhouse,uml 1.0, Microsoft, Oracle, Rational, Texas Instruments y Unisys, produjeron un lenguaje de modelado bien definido, expresivo, potente y aplicable de problemas. En 1997 fue adaptada y aceptada la estandarización de UML 1.1 por OMG. En 2005 se adoptó la versión 2.0 que incluye muchas características adicionales. [1]

Booch, Rumbaugh & Jacobson, definen al Lenguaje Unificado de Modelado (*Unified Modeling Language*, UML) como un lenguaje estándar para escribir planos de software, UML puede emplearse para visualizar, especificar, construir y

documentar los artefactos de un sistema que involucre una gran cantidad de software.

2.3.1 Artefactos UML

UML está conformado por artefactos, los cuales ayudan con el proceso de desarrollo del software, enseguida se enlistan algunos de ellos (Booch et al., 2006).

- **Diagrama de clases:** muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones para representar los elementos del sistema.
- **Diagrama de casos de uso:** se utilizan para especificar y visualizar el comportamiento del sistema. Estos diagramas facilitan la comprensión de los sistemas, subsistemas y clases, al presentar una vista externa de cómo pueden interactuar estos elementos en un contexto dado.
- **Especificación de casos de uso:** se describe de manera detallada lo que los actores realizan en cada caso de uso.
- **Diagrama de secuencia:** se emplea para modelar los componentes dinámicos de un sistema. Esto puede implicar modelar instancias (clases), interfaces y relaciones, además, especificar los mensajes enviados entre ellos, todo esto en un escenario establecido.
- **Diagrama de actividades:** son usados para los aspectos dinámicos del sistema.

2.4 Base de datos

Una base de datos es una colección organizada de datos o información, de un mismo contexto, habitualmente almacenados electrónicamente en un sistema de computadora. [10]

2.4.1 Modelo relacional

Con base a las definiciones de la empresa Oracle, una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales utilizan una forma intuitiva y directa de representar o simbolizar los datos persistentes en tablas llamadas relaciones.

2.4.2 Modelo Entidad-Relación

El esquema conceptual de la base de datos reúne un conjunto de especificaciones que serán la base que generará el esquema conceptual, originado por los requisitos de usuario (Marqués, 2011).

2.5 SQL

Las siglas SQL corresponden a *Structured Query Language*, es un lenguaje estándar que permite manejar los datos de una base de datos relacional.

2.6 MySQL

Es un Sistema de Gestión de Bases de Datos (SGBD) relacionales de código abierto con un modelo cliente-servidor.

2.7 PHP

PHP es un acrónimo de (Hypertext Preprocessor) es un lenguaje de código abierto utilizado a nivel mundial y que permite la incrustación del lenguaje en el esqueleto de HTML. Ejecuta el código script del lado del servidor y regresa peticiones al cliente.

2.8 Aplicación web

Es un tipo de software que se programa en un lenguaje soportado por los navegadores web y su ejecución es hecha en un navegador en la Internet o de una intranet.

2.9 CodeIgniter

CodeIgniter, es framework que ejecuta scripts del lenguaje de programación PHP. El núcleo utiliza el modelo vista-controlador (MVC), permitiendo que las aplicaciones se desarrollen en un menor tiempo y con un alto margen de calidad en el código.

2.10 Grocery CRUD

Con pocas líneas simples de código, se puede crear un CRUD (*Create, Read, Update and Delete*), simplificando el desarrollo de las páginas en PHP a través de invocar métodos render.

CAPÍTULO III. ANÁLISIS Y DISEÑO

3.1 Especificación de requerimientos

En esta sección se especifican de forma detallada cada uno de los requerimientos, estos deben ser satisfechos por la aplicación web. Todos los requerimientos aquí expuestos son indispensables, es decir, no sería aceptable un sistema que no satisfaga alguno de los requerimientos aquí mostrados

3.1.1 Requerimientos funcionales

Los requerimientos funcionales deben especificar de manera concreta los servicios que debe proporcionar la aplicación. [2]

- **RF1 Gestionar parque:** El administrador podrá consultar y modificar el parque, para luego poder añadir las diferentes especies de flora y fauna que habiten en él.
- **RF2 Gestionar especies:** La aplicación debe permitir al administrador crear, consultar, modificar y eliminar flora y fauna del parque.
- **RF3 Gestionar flora y fauna:** La aplicación debe permitir al administrador crear, consultar, modificar y eliminar flora o fauna parque.
- **RF4 Gestionar comentarios:** La aplicación debe permitir al administrador consultar y eliminar los comentarios del parque.
- **RF5 Consultar flora y fauna:** La aplicación debe permitir al visitante buscar las diferentes especies de flora y fauna que habitan en el parque para poder visualizar sus características.
- **RF6 Comentar en el foro:** La aplicación debe permitir al usuario colocar comentarios en el foro del parque.
- **RF7 Consultar comentarios:** La aplicación debe permitir al usuario ver los comentarios del parque

3.1.2 Requerimientos no funcionales

Los requerimientos no funcionales especifican restricciones, atributos de calidad o de rendimiento. [2]

- Usabilidad:
 - **RN-1** El sistema está preparado para ser manejado a través de mouse y teclado.
 - **RN-2** Desplegar mensajes de error y advertencia intuitivos.
 - **RN-3** Verificar/validar límites de campos y tipos de datos de las pantallas en relación al modelo de datos.
 - **RN-4** La consulta Web debe ser accedida desde cualquier navegador.

- Confiabilidad:
 - **RN-5** Se debe asegurar la disponibilidad del Sistema 24 X 7 X 365
- Performance:
 - **RN-6** Se espera que el tiempo de respuesta en el momento de presionar un botón para continuar con el flujo de la información que no supere los 20 segundos.
 - **RN-7** El sistema deberá liberar a todos los recursos de memoria al momento de cerrar una ventana y finalizar una funcionalidad.
- Soportabilidad:
 - **RN-8** Los mensajes de error serán capturados por la aplicación y serán visualizados al usuario final.
- Documentación:
 - **RN-9** Correcta redacción y ortografía en las pantallas.
 - **RN-10** Uso estandarizado de pantallas, mensajes y estilos.

3.1.3 Requerimientos de interfaz

- **Interfaces de Hardware**

Para esta aplicación es necesario un ordenador servidor en el cual se instalará el servidor WEB apache, MySQL, PHP y la aplicación se puede descargar del repositorio GitHub

- **Interfaces de Software**

La aplicación se desarrolló mediante software de licencia libre por lo tanto no se deberá pagar por el uso de: servidor WEB (Apache), Sistema de Gestión de base de datos (MySQL) y el Framework CodeIgniter de PHP, por lo tanto, la utilización de estos programas se hará mediante las políticas establecidas por este tipo de licenciamiento.

- **Interfaces de Comunicación**

Debido a que el sistema no interactúa con otros sistemas y es autónomo no se desarrollaran interfaces con otras aplicaciones. Las conexiones requeridas para poder utilizar el servidor web, MySql, PHP y un DNS, se hizo por medio de configuraciones a estos programas.

3.2 Análisis del Sistema

De acuerdo con los requisitos funcionales y no funcionales se analizó el sistema por medio del diagrama de casos de uso y sus especificaciones de cada uno de ellos.

3.2.1 Diagrama de Casos de uso

En la figura 2, se muestra el diagrama de casos de uso del sistema con las siguientes características.

Actores

La aplicación web tiene dos tipos de usuario: administrador y visitante.

- **Administrador:** Usuario el cual manejará todo el sistema.
- **Visitante:** Usuario que utilizará el sistema para obtener información sobre la biodiversidad del parque.

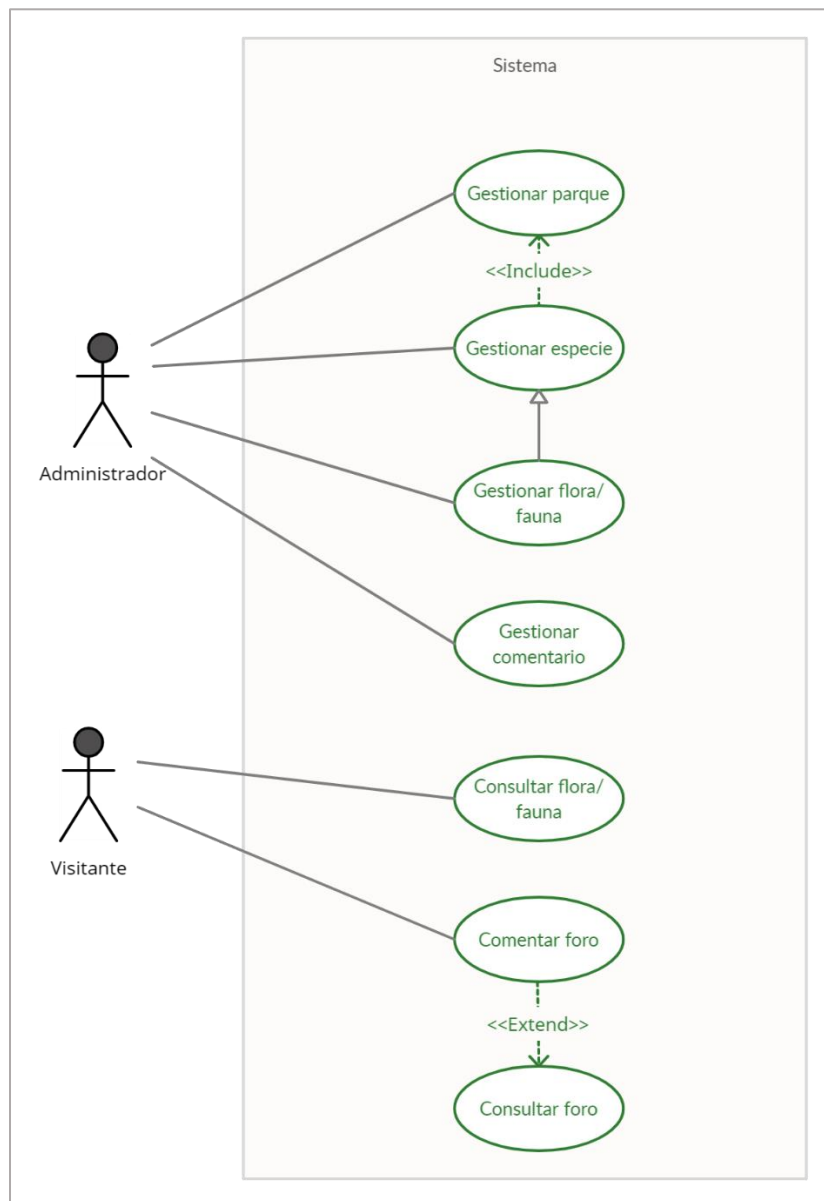


Figura 2. Diagrama de casos de uso

3.2.2 Especificaciones de Casos de uso

A continuación, se detallará los casos de uso, esto quiere decir que se especificará los flujos que deben seguir los actores para cumplir lo que el caso de uso representa.

Caso Uso: Gestionar parques (ECU1)

- **Flujo básico:** (Acceder a la ventana de Parque)
 1. Para **Ver parque**
 - 1.1 Hacer clic en el botón de ver (ícono de lupa) del parque deseado.
 - 1.2 El administrador podrá ver toda la información sobre el parque que selecciono.
 - 1.3 Dar clic en el botón de “Volver a la lista”.
 2. Para **Actualizar parque**
 - 2.1 Hacer clic en el botón Editar (ícono de lápiz) del parque deseado.
 - 2.2 Se habilitan los campos de texto para su edición y con ello tres botones: “Actualizar cambios”, “Actualizar y volver a la lista” y “cancelar”.
 - 2.3 El administrador da clic en “Actualizar cambios”. E1
 - 2.4 Se actualiza el parque en la Base de Datos.
 - 2.5 El administrador da clic en “Volver a la lista”.
- **Flujos Alternativos**
 1. Para Actualizar parque, si el administrador da clic en el botón “Actualizar y volver a la lista” del flujo FB2.2, se muestra la lista con el parque. E1
 2. Para Actualizar parque, si el administrador da clic en el botón “Cancelar” del flujo FB2.2, se muestro un cuadro de dialogo E2. Dar clic en “Aceptar”.
- **Precondición**
 1. El administrador debe haber iniciado sesión como Administrador.
- **Postcondición**
 1. El visitante puede ver el parque.
- **Puntos de extensión**
 1. Sus datos han sido actualizados correctamente
 2. Los datos que intenta actualizar no se han guardado ¿Esta seguro que quiere volver a la lista?

Caso de uso: Gestionar especies (ECU2)

- **Flujo básico:** (Acceder a la ventana de Especie)

1. Para **Ver especie**
 - 1.1 Hacer clic en el botón de ver (ícono de lupa) de la especie deseada.
 - 1.2 El administrador podrá ver toda la información sobre la especie que selecciono.
 - 1.3 Dar clic en el botón de “Volver a la lista”.
2. Para **añadir especie**
 - 2.1 Dar clic en “Añadir registro”.
 - 2.2 Llenar los campos: Nombre especie y Parque.
 - 2.3 Hacer clic en Guardar.
 - 2.4 Se valida que no haya un registro igual. E1
 - 2.5 Se da de alta la especie en la base de datos.
 - 2.6 Dar clic en el botón de “Volver a la lista”.
3. Para **Eliminar especie**,
 - 3.1 Hacer clic en el botón de Eliminar (ícono de menos) de la especie que deseas. E3
 - 3.2 Dar clic en “Aceptar”.
 - 3.3 Se elimina la especie de la Base de Datos.
4. Para **Actualizar especie**
 - 4.1 Hacer clic en el botón Editar (ícono de lápiz) de la especie deseada.
 - 4.2 Se habilitan los campos de texto para su edición y con ello tres botones: “Actualizar cambios”, “Actualizar y volver a la lista” y “cancelar”.
 - 4.3 El administrador da clic en “Actualizar cambios”. E4
 - 4.4 Se actualiza la especie en la Base de Datos.
 - 4.5 El administrador da clic en “Volver a la lista”.

- **Flujos Alternativos**

1. Para agregar una nueva especie, el administrador no realiza la acción del flujo FB2.3 y da clic en “Guardar y volver a la lista”. Se muestra la lista. E1
2. Para agregar una nueva especie, el administrador no realiza la acción del flujo FB2.3 y da clic en “Cancelar”. Se muestra un cuadro de dialogo E2, Dar clic en “Aceptar”
3. Para Eliminar una especie, si el administrador da clic en el botón “Cancelar” la acción del cuadro de diálogo del flujo FB3.1 se interrumpe el flujo y la especie no es eliminada.

4. Para Actualizar especie, si el administrador da clic en el botón “Actualizar y volver a la lista” del flujo FB4.2, se muestra la lista con la especie. E1
5. Para Actualizar especie, si el administrador da clic en el botón “Cancelar” del flujo FB2.2, se muestro un cuadro de dialogo E5. Dar clic en “Aceptar”.

- **Precondición**

1. El administrador debe haber iniciado sesión como Administrador.

- **Postcondición**

1. El visitante puede ver especies.

- **Puntos de extensión**

1. Sus datos han sido guardados correctamente.
2. Los datos que intenta añadir no se han guardado ¿Está seguro de volver a la lista?
3. ¿Estás seguro que quieres eliminar este registro?
4. Sus datos han sido actualizados correctamente.
5. Los datos que intenta actualizar no se han guardado ¿Esta seguro que quiere volver a la lista?

Caso de uso: Gestionar flora y fauna (ECU3)

- **Flujo básico:** (Acceder a la ventana de Flora o Fauna)

1. Para **Ver Flora o Fauna**

- 1.1 Hacer clic en el botón de ver (ícono de lupa) de la especie de flora o fauna deseada.
- 1.2 El administrador podrá ver toda la información sobre la especie de flora y Fauna que selecciono.
- 1.3 Dar clic en el botón de “Volver a la lista”.

2. Para **Añadir flora o fauna**

- 2.1 Dar clic en añadir.
- 2.2 Llenar los campos:
 - Flora: Nombre científico, Nombre común, Clase, Orden, Familia, Cuidado, Imagen 1, Imagen 2, Imagen 3, Especie.
 - Fauna: Nombre científico, Nombre común, Clase, Alimentación, Dieta, Reproducción, Tamaño, Longevidad, Periodo de incubación, Cuidado, Cantidad, Imagen 1, Imagen 2, Imagen 3, Especie.
- 2.3 Hacer clic en Guardar.
- 2.4 Se valida que no haya un registro igual. E1
- 2.5 Se da de alta la especie de flora o fauna en la base de datos.
- 2.6 Dar clic en el botón de “Volver a la lista”.

3. Para **Eliminar flora o fauna,**

- 3.1 Hacer clic en el botón de Eliminar (ícono de menos) de la especie de flora o fauna que deseas. E3
- 3.2 Dar clic en “Aceptar”.
- 3.3 Se elimina la especie de flora o fauna de la Base de Datos.

4. Para **Actualizar flora o fauna**

- 4.1 Hacer clic en el botón Editar (ícono de lápiz) de la especie de flora o fauna deseada.
- 4.2 Se habilitan los campos de texto para su edición y con ello tres botones: “Actualizar cambios”, “Actualizar y volver a la lista” y “cancelar”.
- 4.3 El usuario da clic en “Actualizar cambios”. E4
- 4.4 Se actualiza la especie de flora o fauna en la Base de Datos.
- 4.5 El usuario da clic en “Volver a la lista”.

- **Flujos Alternativos**

1. Para agregar una nueva especie de flora o fauna, el administrador no realiza la acción del flujo FB2.3 y da clic en “Guardar y volver a la lista”. Se muestra la lista. E1

2. Para agregar una nueva especie de flora o fauna, el administrador no realiza la acción del flujo FB2.3 y da clic en “Cancelar”. Se muestra un cuadro de dialogo E2, Dar clic en “Aceptar”
3. Para Eliminar una especie de flora y fauna, si el administrador da clic en el botón “Cancelar” la acción del cuadro de diálogo del flujo FB3.1 se interrumpe el flujo y la especie no es eliminada.
4. Para Actualizar una especie de flora y fauna, si el administrador da clic en el botón “Actualizar y volver a la lista” del flujo FB4.2, se muestra la lista con el parque. E1
5. Para Actualizar una especie de flora y fauna, si el administrador da clic en el botón “Cancelar” del flujo FB2.2, se muestro un cuadro de dialogo E5. Dar clic en “Aceptar”.

- **Precondición**

1. El administrador debe haber iniciado sesión como Administrador.

- **Postcondición**

1. El visitante puede ver las especies de flora y fauna.

- **Puntos de extensión**

1. Sus datos han sido guardados correctamente.
2. Los datos que intenta añadir no se han guardado ¿Está seguro de volver a la lista?
3. ¿Estás seguro que quieres eliminar este registro?
4. Sus datos han sido actualizados correctamente
5. Los datos que intenta actualizar no se han guardado ¿Esta seguro que quiere volver a la lista?

Caso de uso: Gestionar comentario (ECU4)

- **Flujo básico:** (Acceder a la ventana de Foro)
 1. Para **Ver comentario**
 - 1.1 Hacer clic en el botón de ver (ícono de lupa) del comentario deseado.
 - 1.2 El administrador podrá ver toda la información sobre el comentario que selecciono.
 - 1.3 Dar clic en el botón de “Volver a la lista”.
 2. Para **Eliminar comentario**,
 - 2.1 Hacer clic en el botón de Eliminar (ícono de menos) del comentario que deseese. E1
 - 2.2 Dar clic en “Aceptar”.
 - 2.3 Se elimina el comentario de la Base de Datos.
- **Flujos Alternativos**
 1. Para Eliminar comentario, si el administrador da clic en el botón “Cancelar” la acción del cuadro de diálogo del flujo FB2.1 se interrumpe el flujo y la especie no es eliminada.
- **Precondición**
 1. El administrador debe haber iniciado sesión como Administrador.
- **Postcondición**
 1. El visitante puede ver las especies de flora y fauna.
- **Puntos de extensión**
 1. ¿Estás seguro que quieres eliminar este registro?

Caso de uso: Consultar flora o fauna (ECU5)

- **Flujo básico**

1. Clic en Especie-Flora o Especie-Fauna
2. Se visualizará todas las especies de flora o fauna
3. Hacer clic en buscar (ícono de lupa).
4. Llenar los campos por los que desees filtrar la búsqueda.
5. Se mostrarán todas las especies relacionadas a la búsqueda. E1

- **Flujos alternativos**

1. Para el flujo FB2, si la búsqueda no arroja datos. E2

- **Precondición**

1. Tiene que haber registros de especies de flora o fauna.

- **Postcondición**

1. El visitante puede navegar en la ventana para ver las especies de flora y fauna

- **Puntos de extensión**

1. Los resultados de la búsqueda son
2. No hay datos que mostrar

Caso de uso: Consultar comentarios (ECU6)

- **Flujo básico:**

1. Clic en Foro
2. Desplazarse en la ventana foro

- **Flujo alternativo**

1. Para el flujo FB1 si el visitante se quiere salir, dar clic en inicio.

- **Precondición**

- 1.

- **Postcondición**

Ninguna

- **Puntos de extensión**

Ninguno

Caso de uso: Comentar en el foro (ECU7)

- **Flujo básico:**

1. Clic en Foro
2. Llenar datos del formulario: Nombre, Comentario e Imagen (Nombre e Imagen no son obligatorios)
3. Dar clic aceptar. E1

- **Flujo alternativo**

1. Para el flujo FB3 si el usuario quiere cancelar la publicación del comentario vaciar todo el formulario.

- **Precondición**

Ninguna

- **Postcondición**

1. El visitante podrá ver su comentario

- **Puntos de extensión**

1. Haz hecho un comentario.

3.3 Diseño del Sistema

En este apartado se muestran los diagramas que ayudaron a diseñar el sistema y entender cómo sería su comportamiento interno y externo.

3.3.1 Diagrama de clases

En la figura 3 se muestra el diagrama de clases, el cual describe las clases de objetos, sus relaciones, atributos y operaciones del sistema.

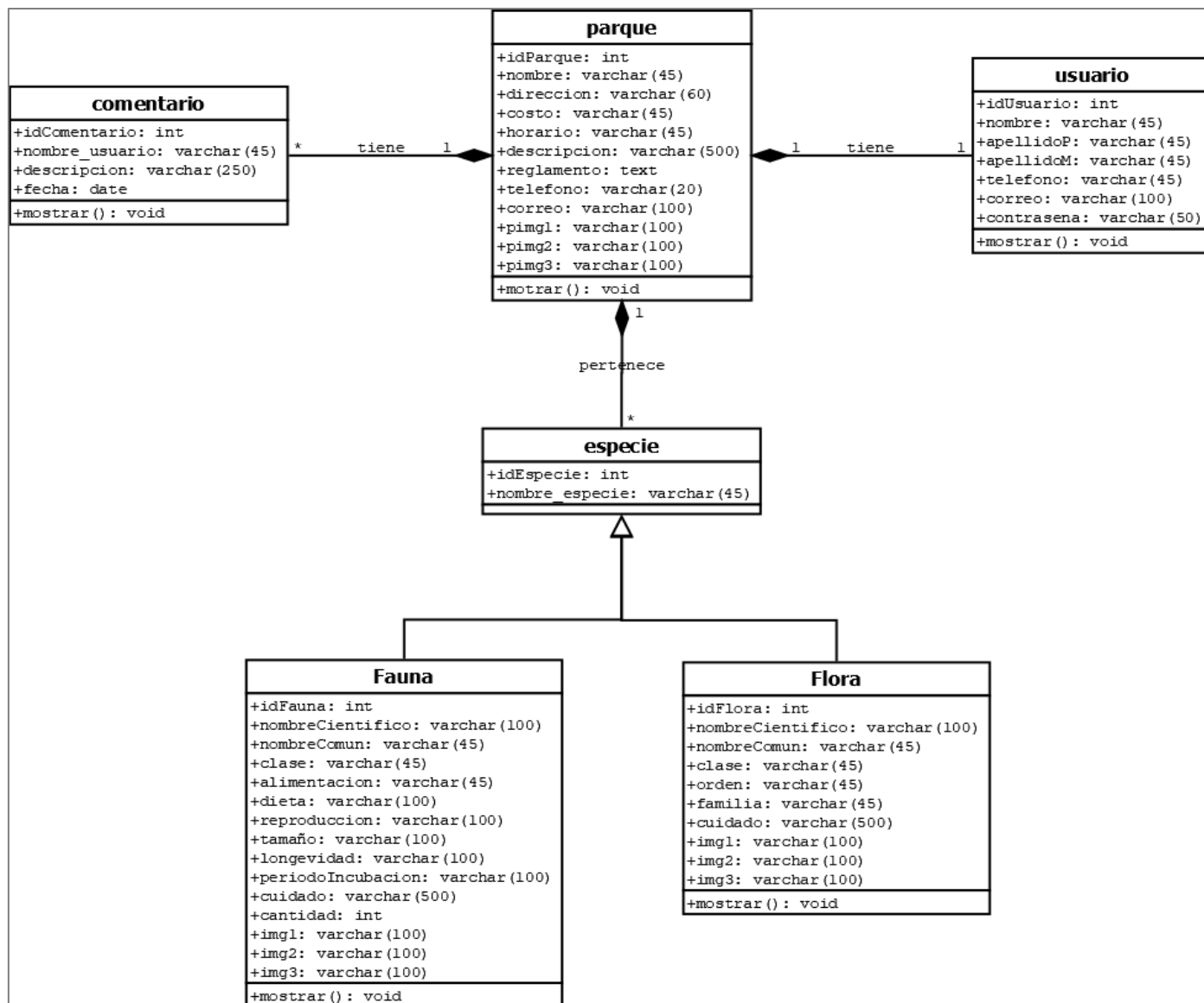


Figura 3. Diagrama de clases

3.3.2 Diagramas de secuencia.

Las siguientes figuras muestran los diagramas de secuencia de cada uno de los casos de uso, estos diagramas ayudaron a modelar la interacción de los objetos con el sistema.

En la figura 4, se muestra el diagrama de secuencia del caso de uso: Gestionar Parque (CU1). Al gestionar parque, se puede ver y modificar parque.

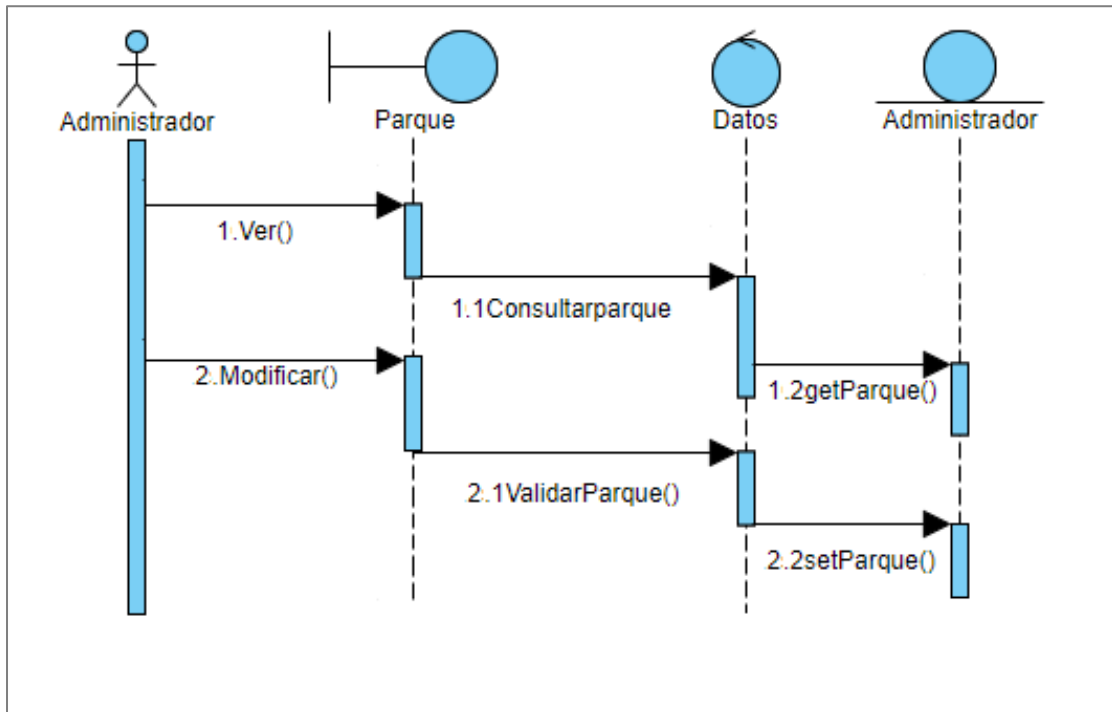


Figura 4. Diagrama de secuencia de CU1.

En la figura 5, se muestra el diagrama de secuencia del caso de uso: Gestionar Especie (CU2). Al gestionar especie, se puede añadir, ver, modificar y eliminar especie.

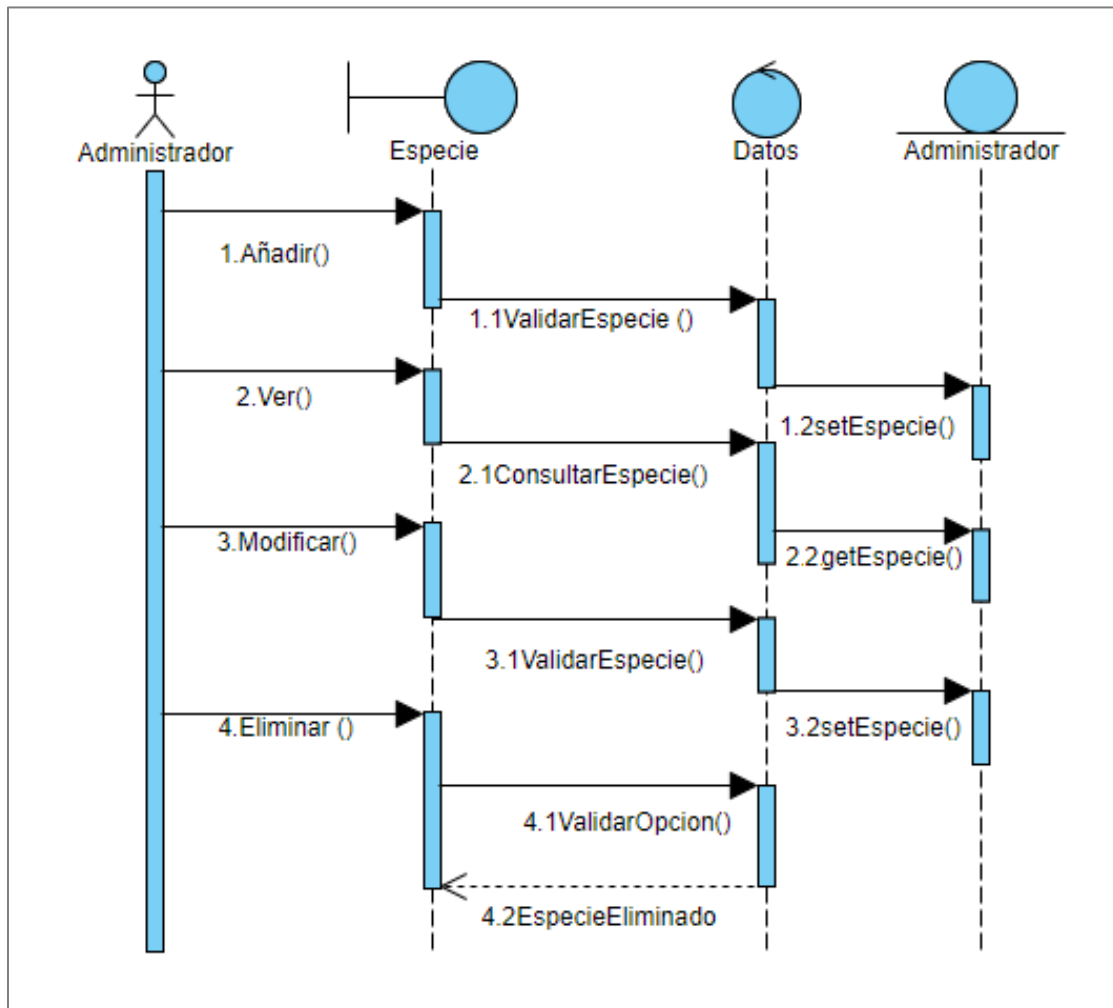


Figura 5. Diagrama de secuencia de CU2.

En la figura 6, se muestra el diagrama de secuencia del caso de uso: Gestionar Flora y Fauna (CU3). Al gestionar flora o fauna, se puede añadir, ver, modificar y eliminar.

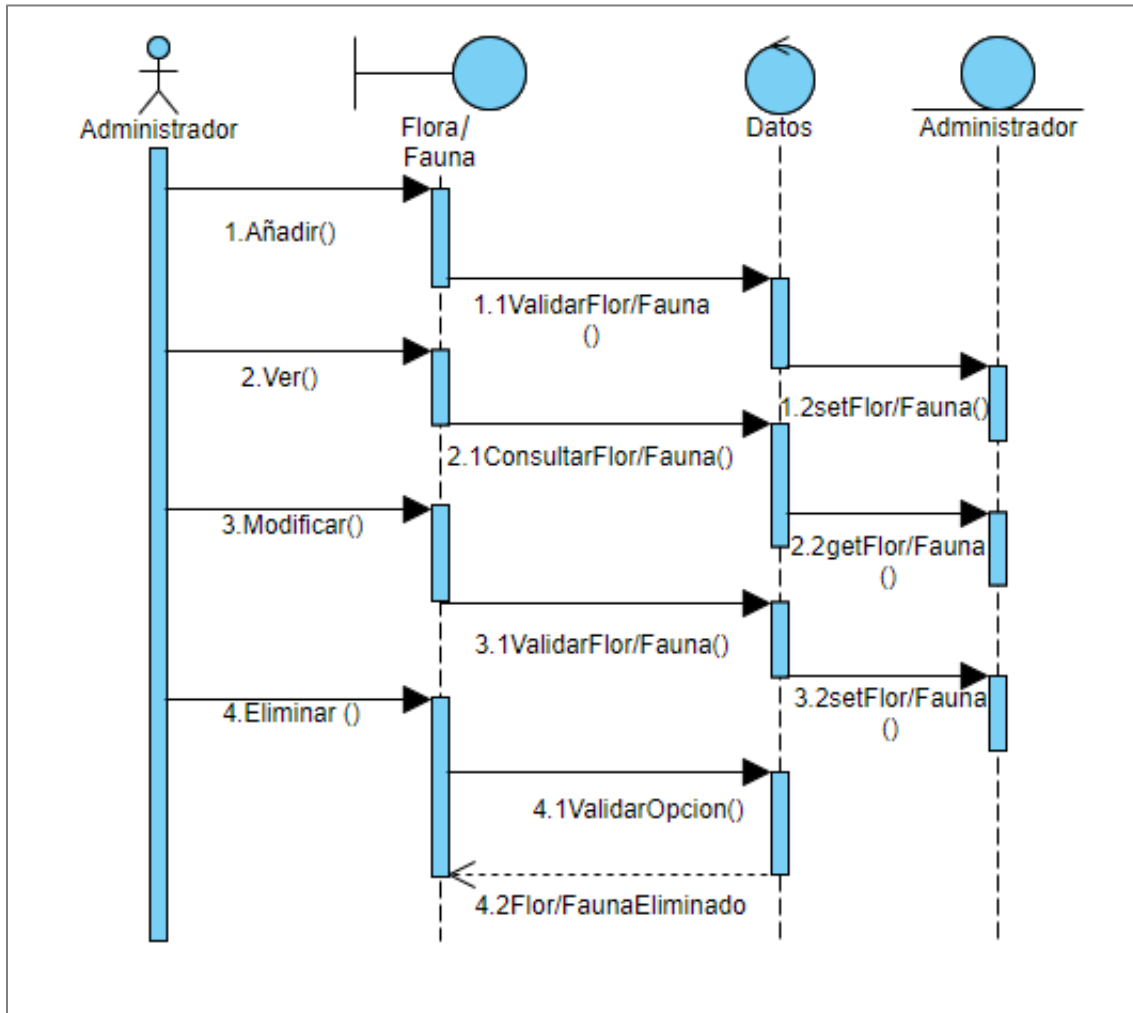


Figura 6. Diagrama de secuencia de CU3

En la figura 7, se muestra el diagrama de secuencia del caso de uso: Gestionar Comentario (CU4). Al gestionar comentario, se puede ver y eliminar comentarios.

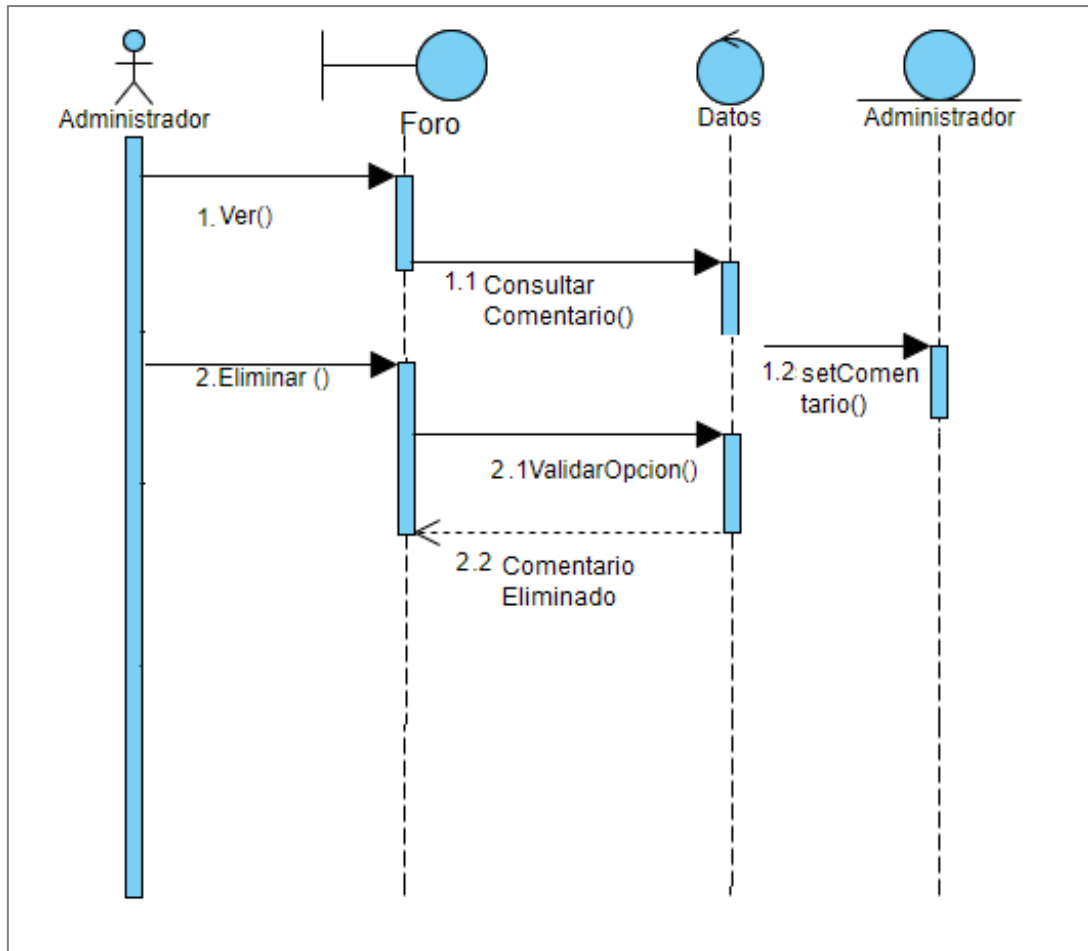


Figura 7. Diagrama de secuencia de CU4

En la figura 8, se muestra el diagrama de secuencia del caso de uso: Consultar flora o fauna (CU5). Al consultar flora o fauna, se puede hacer búsquedas de estas especies y visualizar en pantalla.

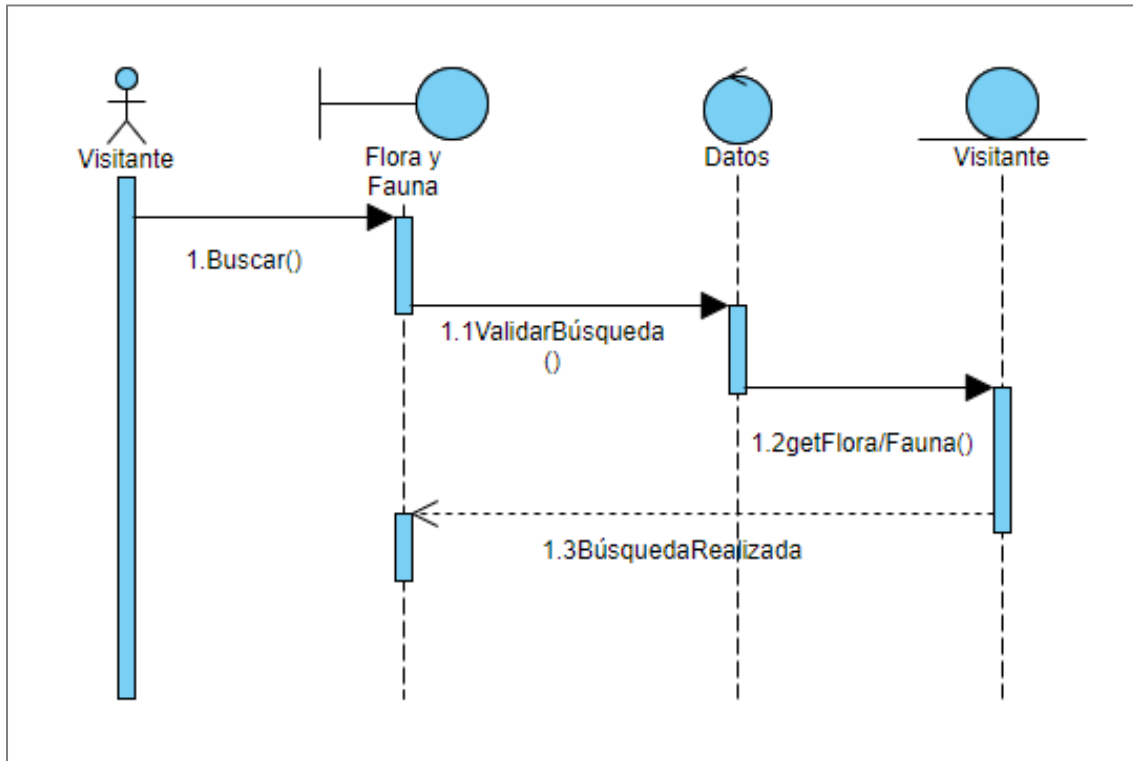


Figura 8. Diagrama de secuencia de CU5.

En la figura 9, se muestra el diagrama de secuencia del caso de uso: Consultar comentarios (CU6). Al consultar comentarios, se puede visualizar todos los comentarios que se hayan hecho del parque.

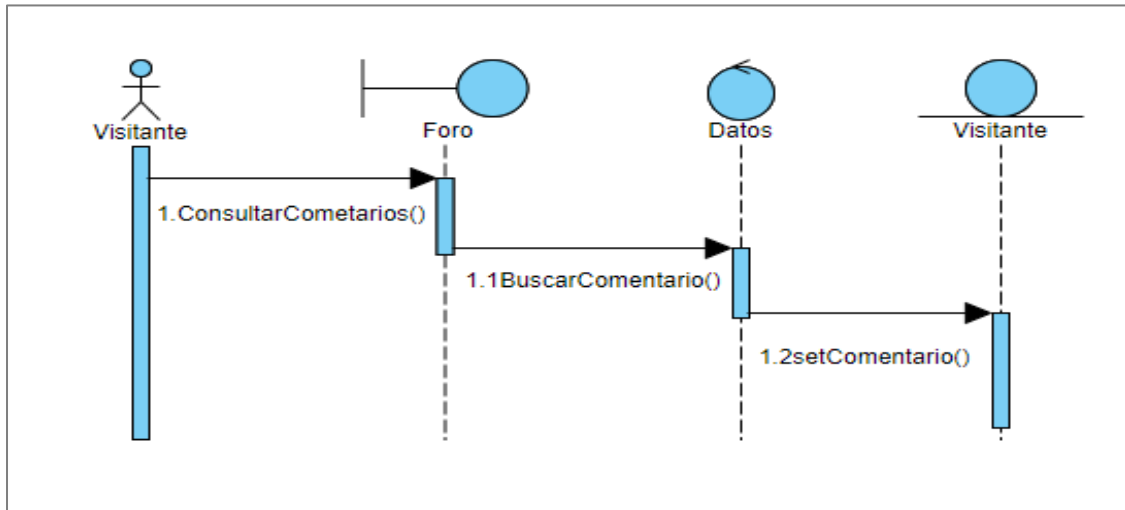


Figura 9. Diagrama de secuencia de CU6.

En la figura 10, se muestra el diagrama de secuencia del caso de uso: Comentar en el foro (CU7). Al comentar en el foro, el comentario puede ser visualizado en pantalla.

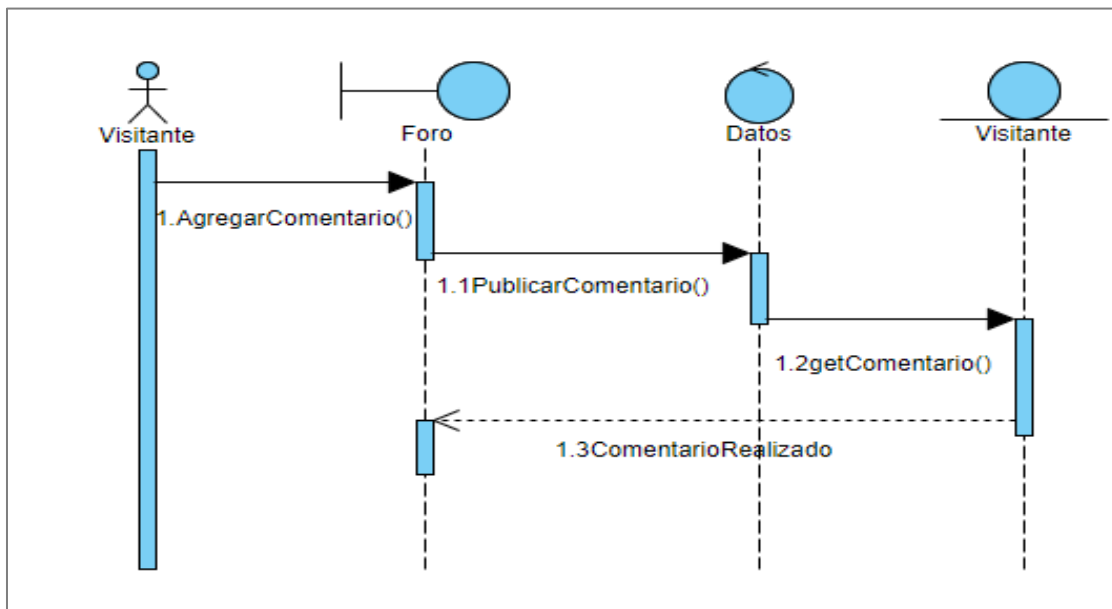


Figura 10. Diagrama de secuencia de CU7.

3.3.3 Diagrama BPM

En la figura 11, se muestra el diagrama BPM el cual ayudó a visualizar los objetivos de la aplicación y el camino que se iba a seguir para crear los módulos de la misma y aumentar la productividad y tener calidad en el sistema.

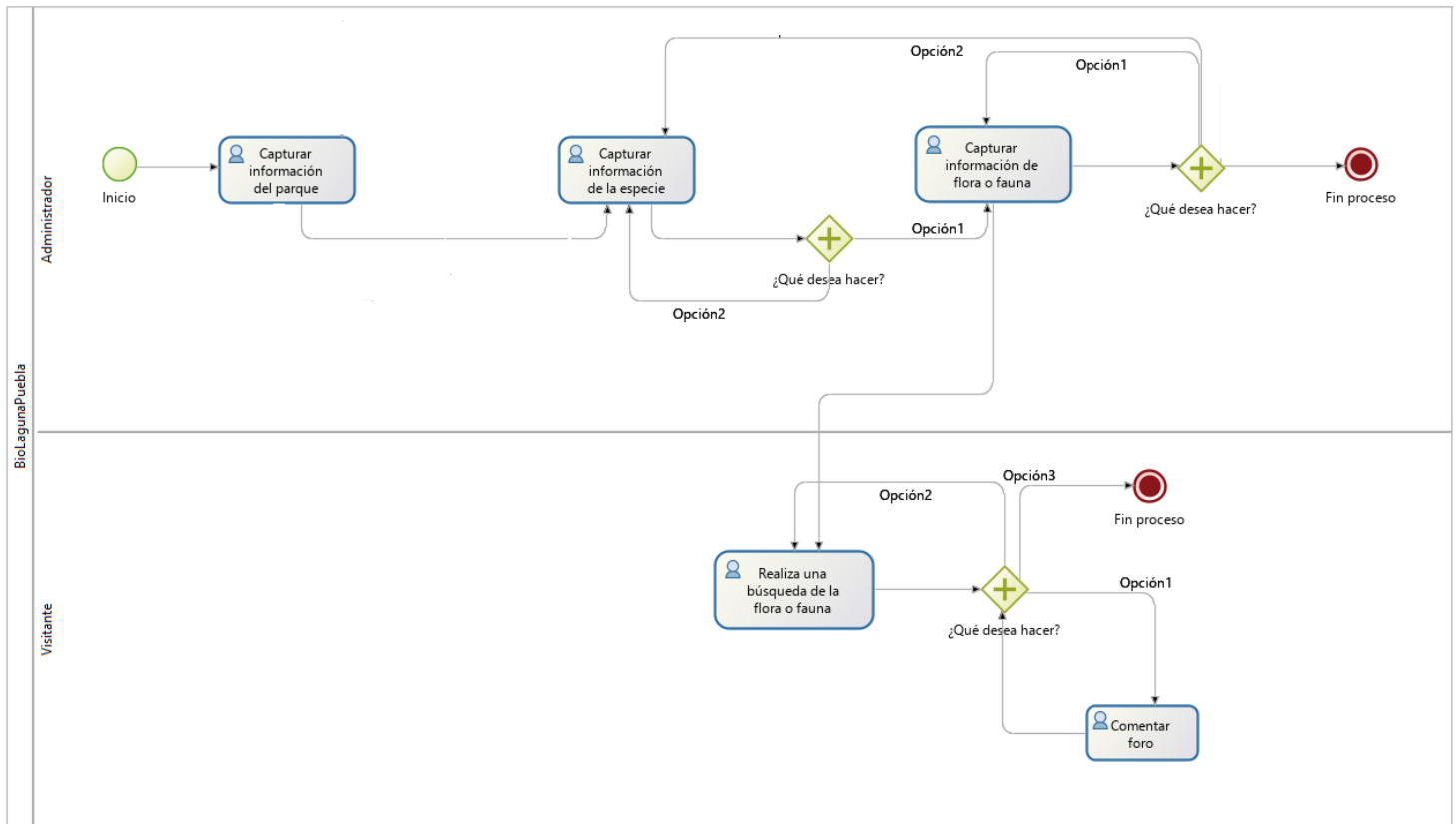


Figura 11. Diagrama BPM.

3.3.4 Diagrama de paquetes

En la figura 12, se muestra el diagrama de paquetes de la aplicación. En este diagrama podemos observar 3 tipos de paquetes, el primero que es la capa de presentación la cual contiene las interfaces que tiene la aplicación, el segundo es la capa lógica de negocios que contiene las acciones que hace la aplicación y la tercera es el paquete de la base de datos que almacena la información de la aplicación.

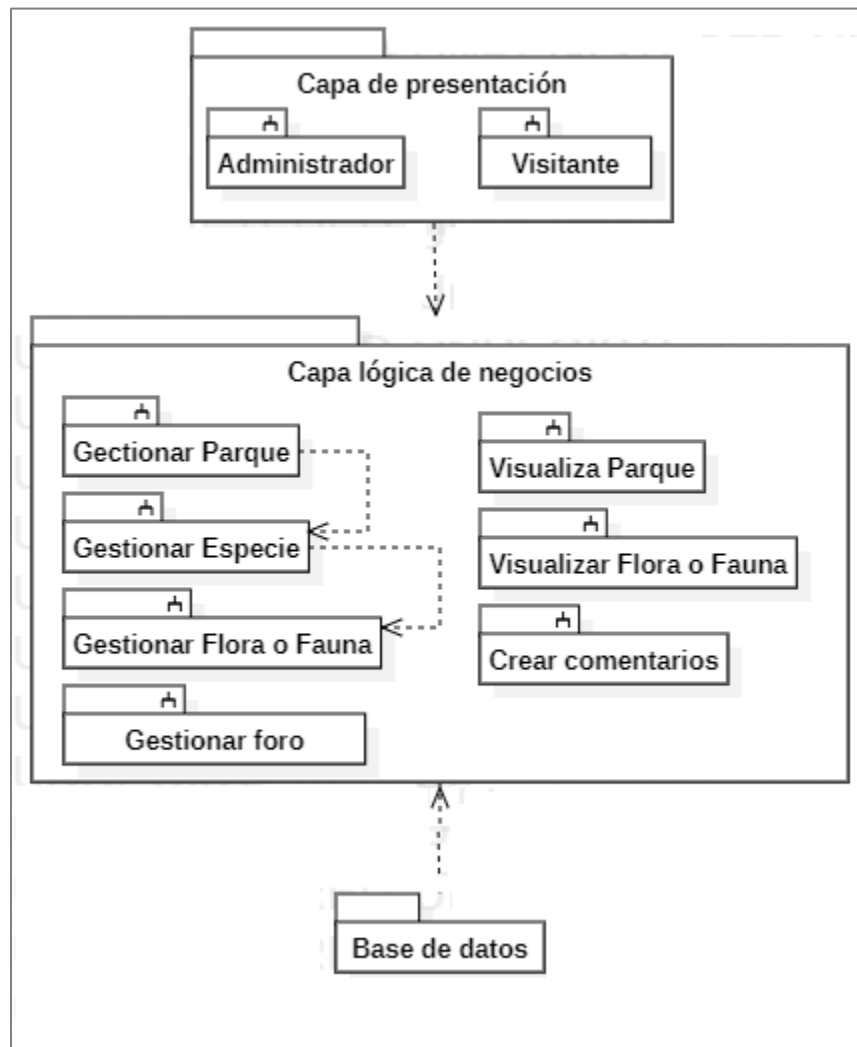


Figura 12. Diagrama de paquetes.

3.3.5 Diagrama de actividades

El diagrama de actividades de la figura 13, nos ayudó a mostrar cuál es el proceso que sigue el usuario para cumplir el objetivo, es decir, muestra el comportamiento del sistema.

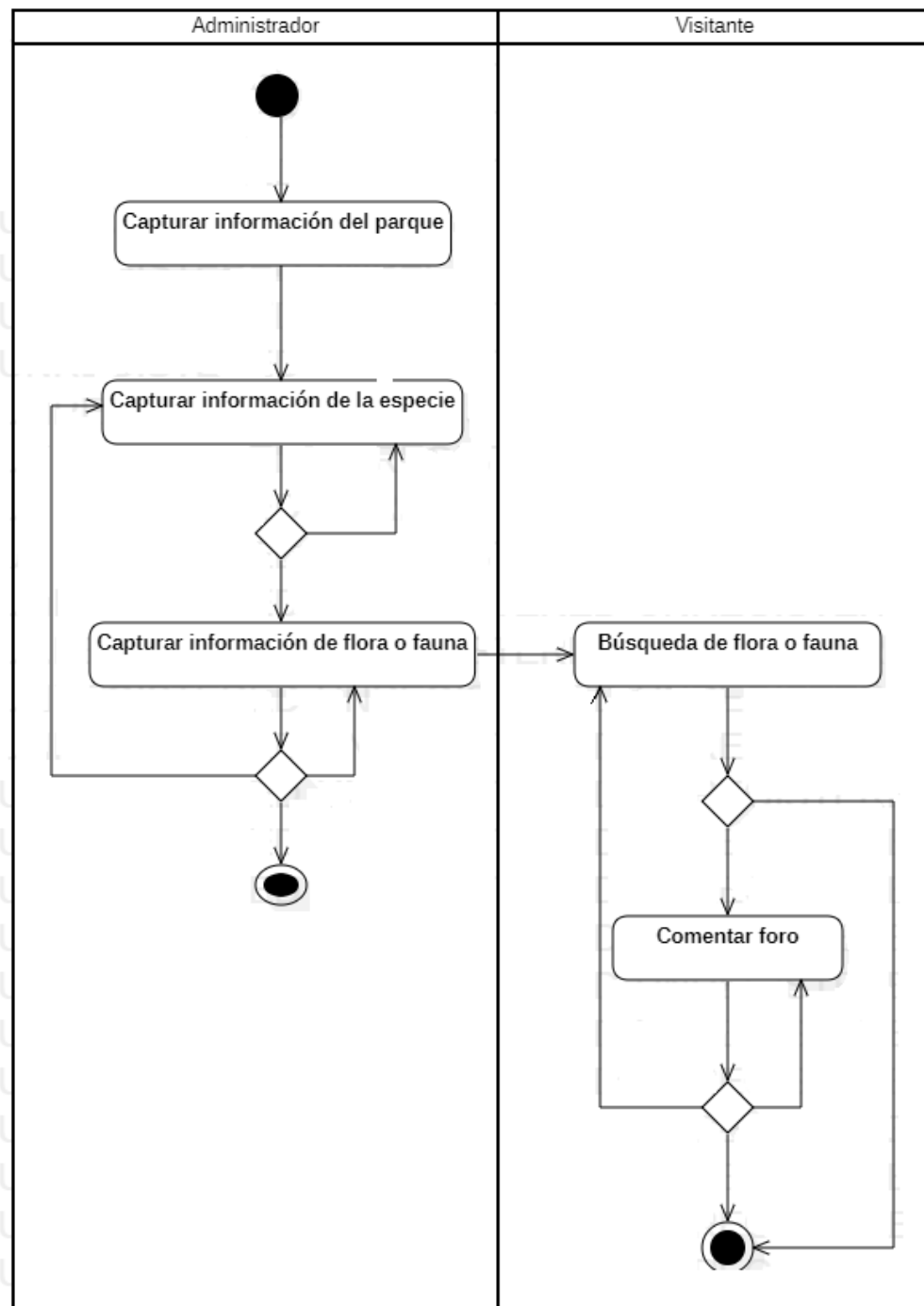


Figura 13. Diagrama de actividades

3.3.6 Diagrama de despliegue

En la figura 14, se muestra el diagrama de despliegue que representa la distribución física (estática) de los componentes software. Se pueden apreciar los siguientes componentes:

- Navegador web: programa que posibilita mostrar el contenido de una página web, ya sea alojada en un servidor local o en Internet.
- Servidor web: programa que emplea el protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol), para almacenar los archivos que forman páginas web.
- CodeIgniter: es el framework que se utilizará el cual se basa en el Modelo-Vista-Controlador (MVC), éste es un estilo de arquitectura de software que funciona de la siguiente manera: el Modelo contiene una representación de los datos, esto quiere decir que maneja el control de la base de datos; la Vista es la interfaz que se le muestra al usuario; el Controlador, es el intermediario entre el Modelo y la Vista, regula el flujo de información y transforma los datos para poder mostrarlos en la vista.
- MySQL: Es un sistema de gestión de bases de datos que sirve como el almacén de información principal de la aplicación, mediante el cual se pueden obtener los datos necesarios para gestionar las peticiones del sistema.

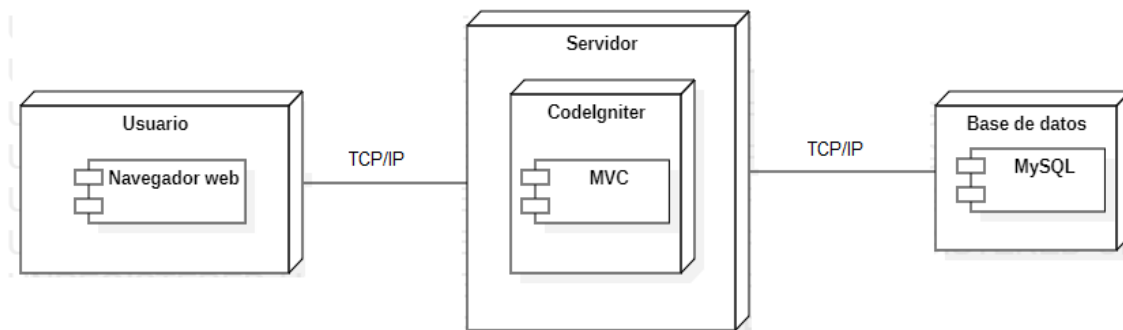


Figura 14. Diagrama de despliegue.

3.3.7 Diagrama de flujo

El framework de CodeIgniter, nos muestra su funcionamiento a través de los pasos siguientes y de acuerdo con la figura 15 que es el diagrama de flujo.

- Index.php inicia los recursos base primordiales para ejecutar CodeIgniter, es decir, ejerce como controlador frontal.

- El enrutador para poder determinar qué hacer con la solicitud HTTP, la examina.
- Si se encuentra un archivo de caché, se envía directamente al navegador, impidiendo la ejecución normal del sistema.
- Seguridad, antes de que se cargue el controlador de la aplicación, la solicitud HTTP y los datos enviados por el usuario se filtran para mayor seguridad.
- Para procesar cualquier solicitud determinada es necesario que el controlador cargue el modelo, las bibliotecas centrales, los helpers, drivers, hooks, lenguajes y cualquier otro recurso,
- La Vista ya finalizada se procesa y luego se envía al navegador web para que se pueda ver, si el almacenamiento en caché está habilitado, la vista se almacena primero en la memoria caché para que se pueda procesar en solicitudes posteriores.

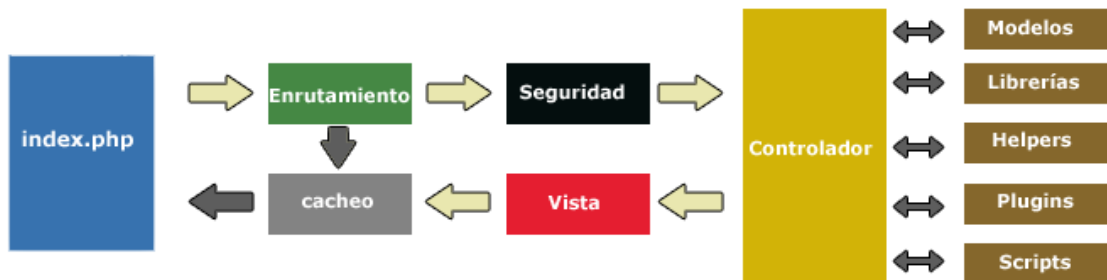


Figura 15. Diagrama de Flujo de CodeIgniter. [11]

3.3.8 Diseño de prototipo

En las figuras de la 16 a la 26, se muestra el prototipo que tendrán las interfaces de la aplicación web. Al ser sólo un prototipo puede que haya cambios pequeños de las interfaces.

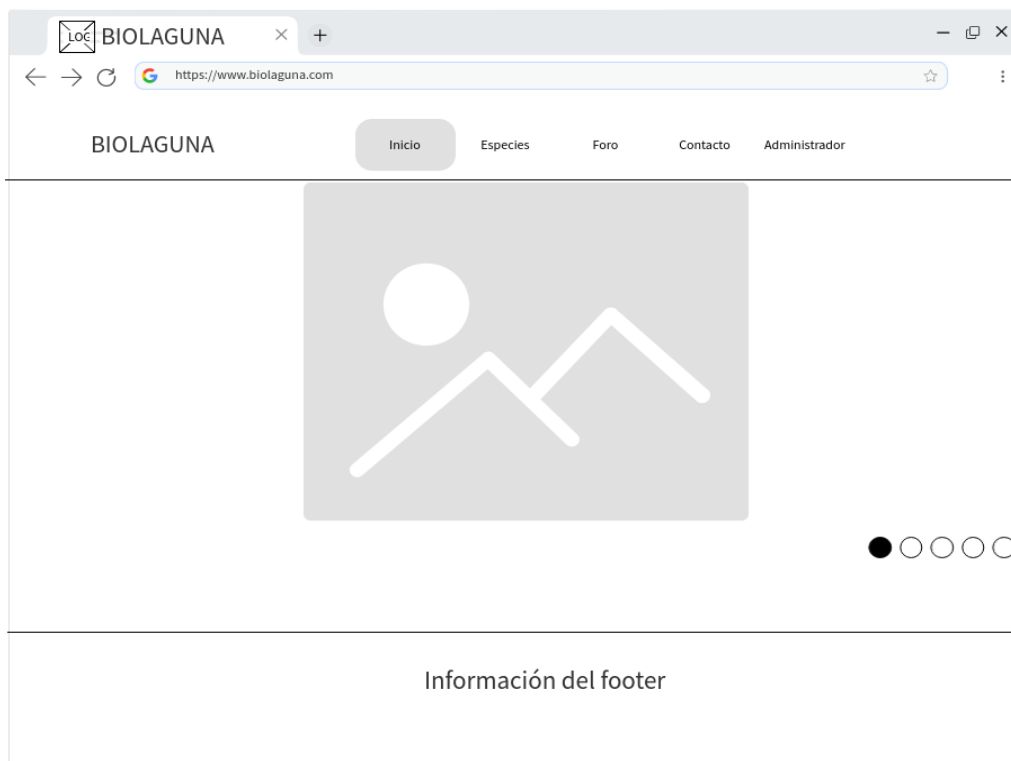


Figura 16. Visitante-Interfaz principal.

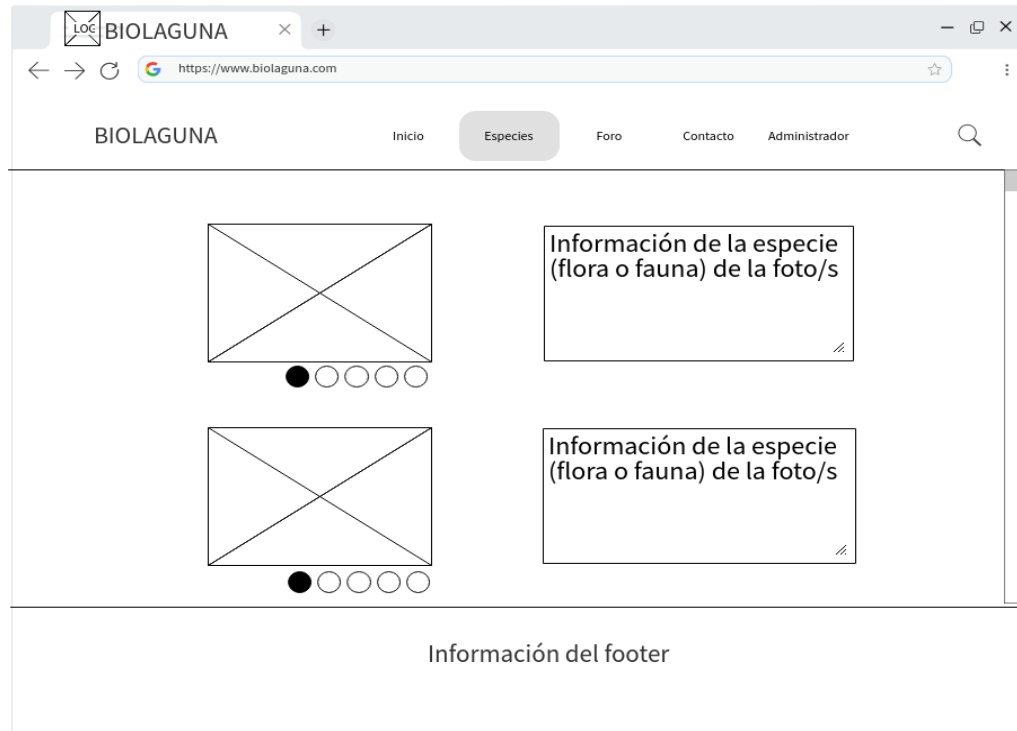


Figura 17. Visitante-Interfaz de flora o fauna.

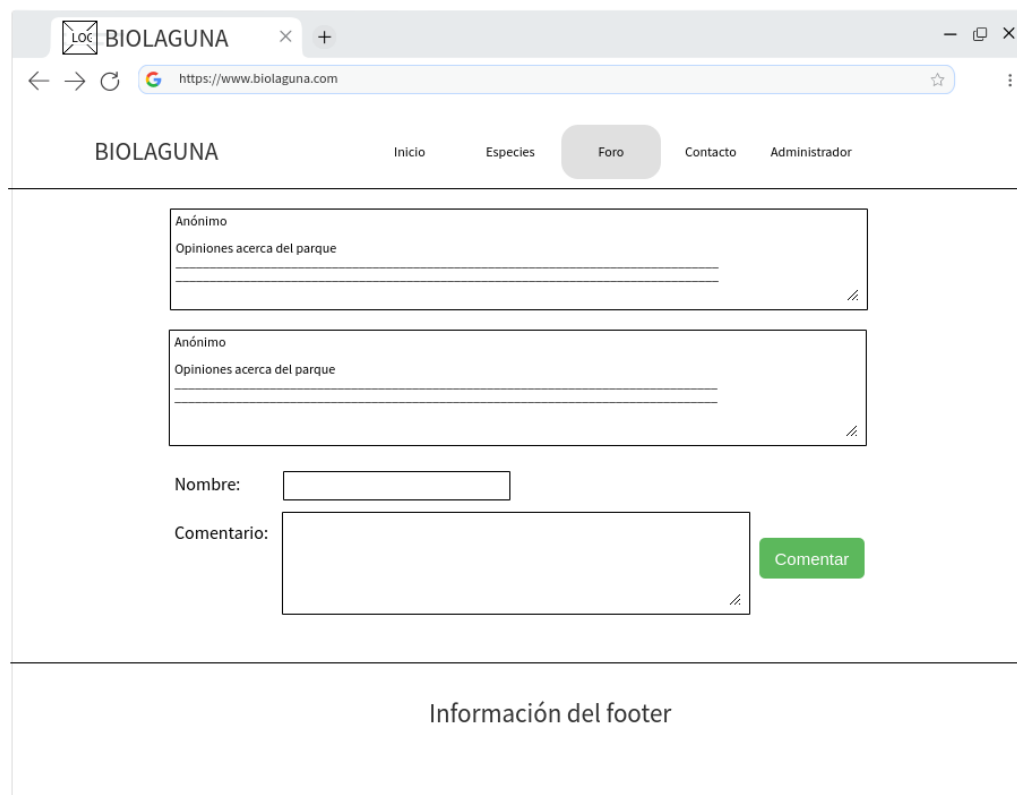


Figura 18. Visitante-Interfaz de flora o fauna.

BIOLAGUNA Inicio Especies **Foro** Contacto Administrador

Anónimo
Opiniones acerca del parque

Anónimo
Opiniones acerca del parque

Nombre:

Comentario:

Comentar

Información del footer

Figura 19. Visitante-Interfaz del foro.

BIOLAGUNA Inicio Especies Foro **Contacto** Administrador

@ contacto@correo.com

Información del footer

Figura 20. Visitante-Interfaz del contacto.

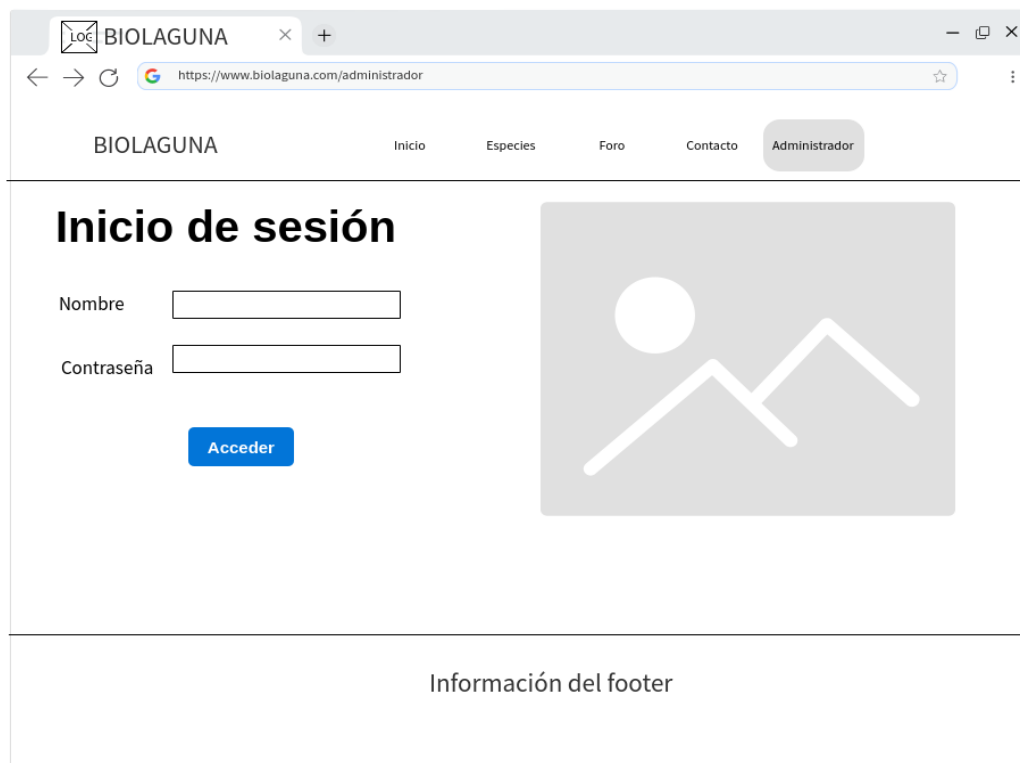


Figura 21. Visitante-Interfaz inicio de sesión para administrador.

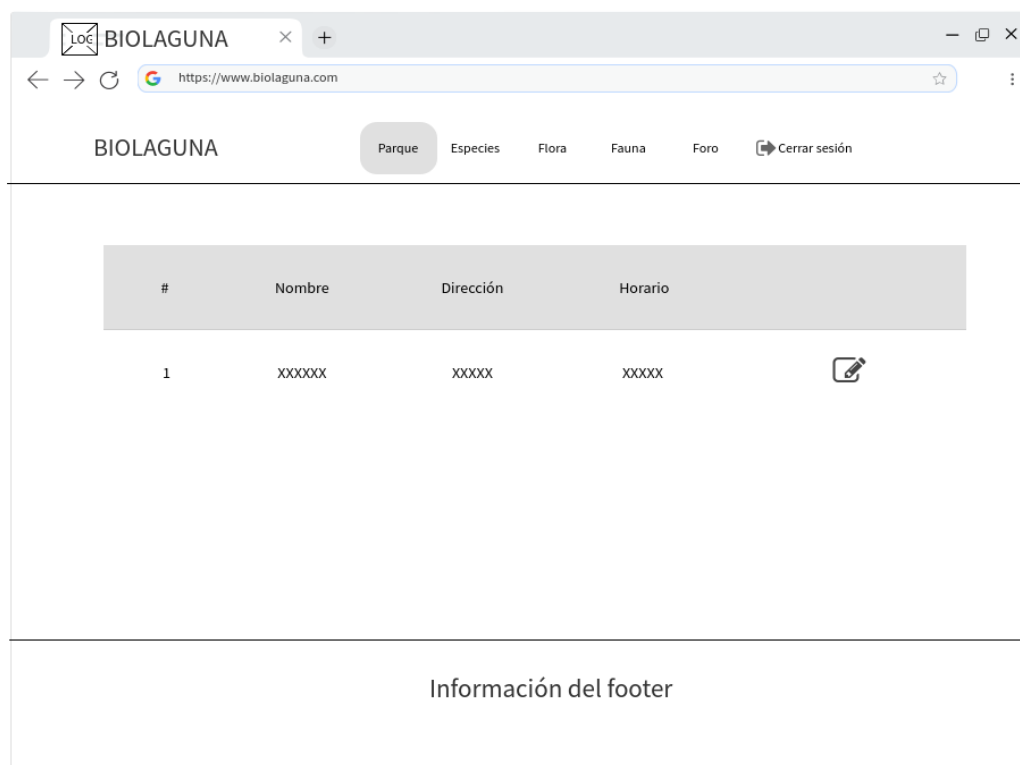


Figura 22. Administrador-Interfaz gestión del parque.



Figura 23. Administrador-Interfaz gestión de especie.

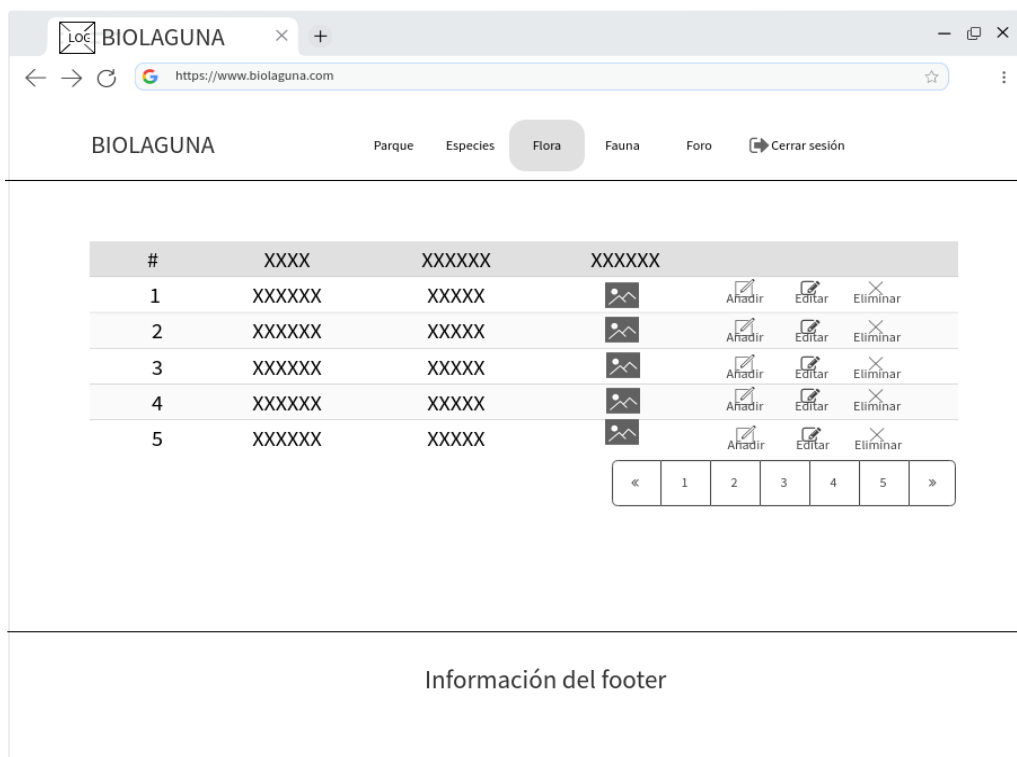


Figura 24. Administrador-Interfaz gestión de flora.



Figura 25. Administrador-Interfaz gestión de fauna.

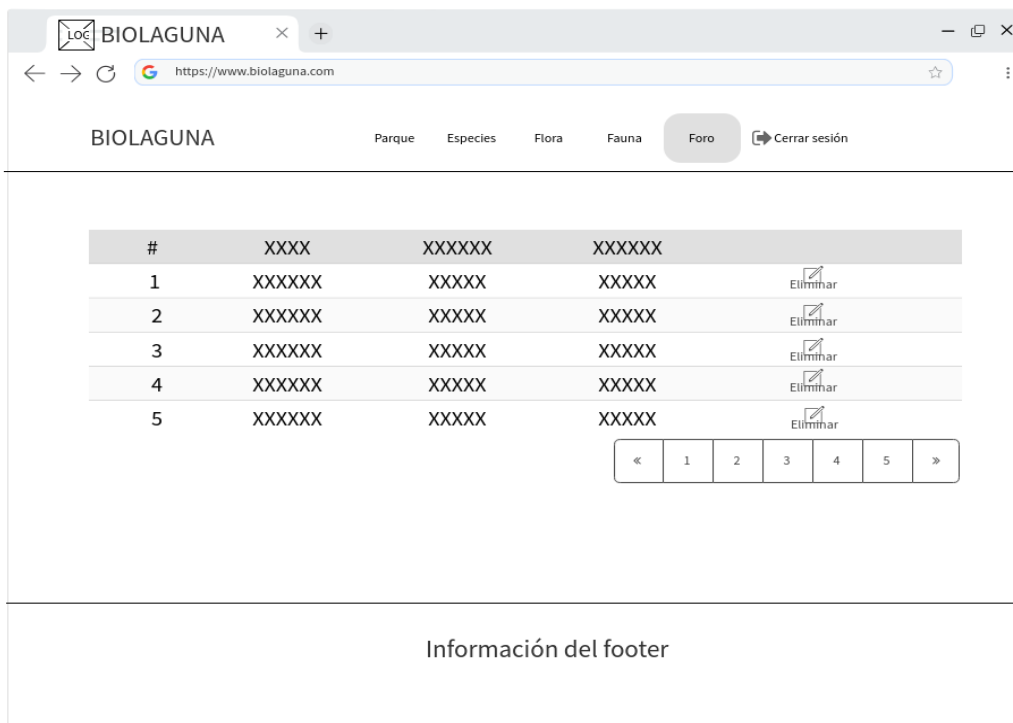


Figura 26. Administrador-Interfaz gestión del foro.

3.4 Desarrollo Scrum

Se establecieron 5 Sprints, después de obtener los requerimientos y el análisis del sistema, cada sprint fue de 15 días trabajando 5 horas diarias. Además, las pruebas tuvieron un lapso para verificar si el sistema cumplía con los requerimientos. En la tabla 1, se observa el cronograma que se tomó en cuenta a lo largo del desarrollo de la aplicación web.

Tabla 1.

Cronograma

		Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	
	21-01-2020- 20-03-2020	23-03-2020 10-04-2020	13-04-2020 01-05-2020	04-05-2020 22-05-2020	25-05-2020 12-06-2020	15-06-2020 03-07-2020	06-07-2020 14-08-2020
Requerimientos							
Análisis							
Diseño							
Desarrollo							
Pruebas							

En la tabla 2, se muestra la lista del producto (*Product Backlog*) que cuenta con 12 ítems que se desarrollaron conforme a la prioridad estimada y a las horas especificadas en ellos, además, cada ítem se desglosó en tareas.

Tabla 2.

Lista del producto (Product Backlog)

ID	Prioridad	Descripción	Horas
1	Alta	Base de datos	45
2	Alta	Instalación de CodeIgniter	15
3	Alta	Instalación de Bootstrap	15
4	Alta	Instalación de Grocery CRUD	15
5	Alta	Inicio de sesión (administrador)	40
6	Alta	Gestión del parque	30
7	Alta	Gestión de especies	30
8	Alta	Gestión de flora y fauna	45
9	Media	Gestión de comentarios	15
10	Media	Inicio de visitante	35
11	Alta	Consultar flora y fauna	50
12	Media	Foro	30
Total			375

En las siguientes tablas, se indica cuáles son las tareas de cada ítem establecido en la lista del producto.

Tabla 3.*Tareas del ítem 1*

1 - Base de datos - 45 horas	
Tarea	Horas
Modelado Entidad-Relación	10
Diccionario de datos	10
Diagrama Entidad-Relación Extendido	10
Código SQL	14
Implementación en MySql	1
Total	45

Tabla 4.*Tareas del ítem 2*

2 – Instalación de CodeIgniter – 15 horas	
Tarea	Horas
Descarga de la última versión	1
Configuración de config	4
Configuración de database	4
Configuración htaccess	6
Total	15

Tabla 5.*Tareas del ítem 3*

3 – Instalación de Bootstrap – 15 horas	
Tarea	Horas
Descarga	1
Instalación de archivos en las carpetas	14
Total	10

Tabla 6.*Tareas del ítem 4*

4 – Instalación de Grocery CRUD – 15 horas	
Tarea	Horas
Descarga de la última versión	1
Instalación de archivos en las carpetas	14
Total	15

Tabla 7.*Tareas del ítem 5*

5 – Inicio de sesión - 40 horas	
Tarea	Horas
Vista inicio de sesión	10
Vista inicio de administrador	10
Método de inicio de sesión	8
Método recaptcha	4
Método de cerrar sesión	8
Total	40

Tabla 8.*Tareas del ítem 6*

6 – Gestión del parque - 35 horas	
Tarea	Horas
Vista	10
Método CRUD	20
Total	30

Tabla 9.*Tareas del ítem 7*

7 – Gestión del especie - 35 horas	
Tarea	Horas
Vista	10
Método CRUD	20
Total	30

Tabla 10.*Tareas del ítem 8*

8 – Gestión de flora y fauna - 45 horas	
Tarea	Horas
Vista flora	2.5
Vista fauna	2.5
Método CRUD flora	20
Método CRUD fauna	20
Total	45

Tabla 11.*Tareas del ítem 9*

9 – Gestión de comentarios- 30 horas	
Tarea	Horas
Vista	5
Método CRUD	10
Total	15

Tabla 12.*Tareas del ítem 10*

10 – Inicio de visitante - 35 horas	
Tarea	Horas
Vista	15
Método para obtener información del parque	25
Total	35

Tabla 13.*Tareas del ítem 11*

11 – Consultar flora y fauna - 60 horas	
Tarea	Horas
Vista de flora	12
Método para obtener información de flora	7
Método de la búsqueda con filtros de flora	11
Vista de fauna	12
Método para obtener información de fauna	7
Método de la búsqueda con filtros de fauna	11
Total	60

Tabla 14.*Tareas del ítem 12*

12 – Foro - 30 horas	
Tarea	Horas
Vista de foro	15
Método para obtener información de comentarios	6
Método para insertar un comentario a la BD	9
Total	30

CAPÍTULO IV. DISEÑO DE BASES DE DATOS

4.1 Diseño lógico

El diseño lógico de la base de datos se basó en el diagrama de clases de la aplicación, obteniendo las clases, sus atributos y sus relaciones. A continuación, se muestra el Modelo Entidad-Relación y el diccionario de datos para detallar mejor cada una de las entidades.

4.1.1 Modelo Entidad-Relación

Utilizamos el Modelo Entidad-Relación (MER) para representar de manera simplificada cómo es que las entidades interactúan entre ellas y cuáles son sus atributos de cada una de ellas.

Entidad: Es un objeto o cosa del mundo real que se diferencia de otros por sus atributos.

Atributo: Son las características que definen a una entidad.

Relación: Es la asociación que permite determinar la interacción que tendrá con otras entidades.

En la figura 27 se muestra el Modelo Entidad-Relación del sistema el cual nos ayudará a obtener el diagrama Entidad-Relación-Extendido.

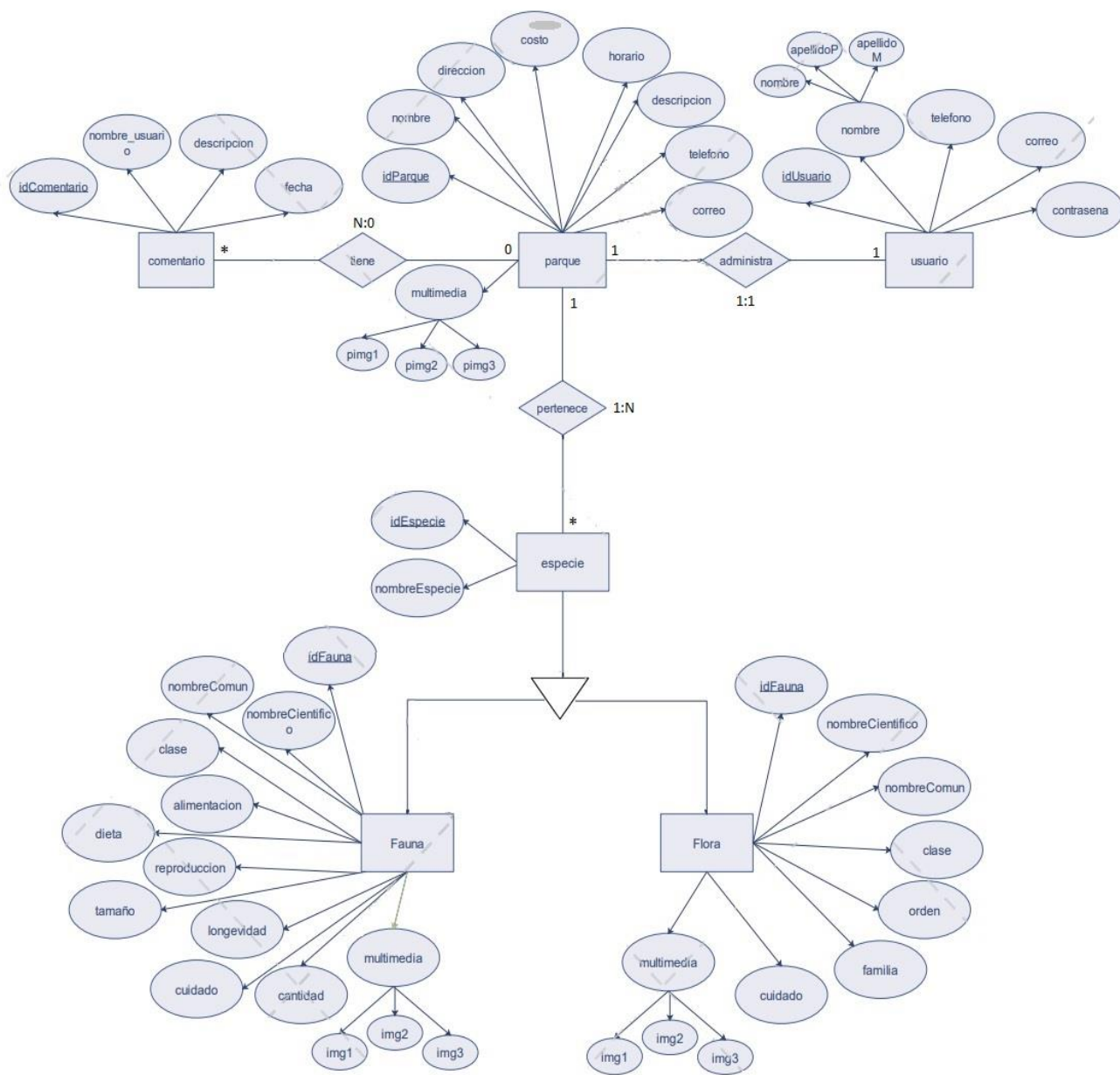


Figura 27. Modelo Entidad-Relación.

4.1.2 Diccionario de datos

Se creó el diccionario de datos (DDD) para enlistar todos los datos que el sistema va a tener y así poder entender cada uno de ellos, ya que se define su significado, su tipo y su flujo.

En las siguientes tablas se muestran los diferentes datos de cada una de las entidades que tendrá el sistema.

Tabla 15.

DDD de la entidad: Parque

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idParque (<i>Primaria</i>)	Identificador del parque	int(11)	No	-
nombre	Nombre del parque	varchar(45)	No	-
direccion	Dirección donde se encuentra el parque	varchar(60)	No	-
costo	El costo de la entrada al parque	varchar(45)	No	-
horario	Horario en el que el parque se encuentra abierto	varchar(45)	No	-
descripcion	Descripción del parque	varchar(500)	No	-
reglamento	El reglamento general del parque	text	No	-
telefono	Teléfono para comunicarse con el responsable del parque	varchar(20)	Sí	-
correo	Correo para comunicarse con el responsable del parque	varchar(100)	No	-
pimg1	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	No	-
pimg2	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-
pimg3	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-

Tabla 16.

DDD de la entidad: Usuario

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idUsuario (<i>Primaria</i>)	Identificador del usuario	varchar(45)	No	-
nombre	Nombre del usuario	varchar(50)	No	-
apellidoP	Apellido paterno del usuario	varchar(45)	No	-
apellidoM	Apellido materno del usuario	varchar(45)	No	-
telefono	Teléfono del usuario	varchar(45)	No	-
correo	Correo del usuario	varchar(100)	No	-
contrasena	Contraseña para ingresar al sistema del usuario	varchar(50)	No	-
idParque	Identificador del parque al que pertenece	int(11)	No	Parque

Tabla 17.*DDD de la entidad: Comentario*

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idComentario (<i>Primaria</i>)	Identificador del comentario	int(11)	No	-
nombre_usuario	Nombre del usuario que realizó el comentario	varchar(45)	Sí	-
descripcion	Comentario del usuario.	varchar(250)	No	-
fecha	Fecha de la realización del comentario	date	No	-
img	Nombre de la imagen que el usuario quiera subir	varchar(100)	No	-
idParque	Identificador del parque al que pertenece	int(11)	No	Parque

Tabla 18.*DDD de la entidad: Especie*

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idEspecie (<i>Primaria</i>)	Identificador de la especie	int(11)	No	-
nombre_especie	Nombre para clasificar las especies.	varchar(45)	No	-
idParque	Identificador del parque al que pertenece	int(11)	No	Parque

Tabla 19.*DDD de la entidad: Flora*

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idFlora (<i>Primaria</i>)	Identificador de la especie de flora	int(11)	No	-
nombreCientifico	Nombre científico de la especie de flora	varchar(100)	No	-
nombreComun	Nombre común de la especie de flora	varchar(45)	No	-
clase	Clase a la que pertenece la especie de flora	varchar(45)	No	-
orden	Orden a la que pertenece la especie de flora	varchar(45)	No	-
familia	Familia a la que pertenece la especie de flora	varchar(45)	No	-
cuidado	Cuidado que debe tener la especie de flora	varchar(500)	No	-
img1	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-
img2	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-

img3	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-
idEspecie	Identificador de la especie a la que pertenece	int(11)	No	Especie

Tabla 20.

DDD de la entidad: Fauna

Nombre	Descripción	Tipo y longitud	Nulo	Relación-Tabla
idFauna (Primaria)	Identificador de la especie de fauna	int(11)	No	-
nombreCientifico	Nombre científico de la especie de fauna	varchar(100)	No	-
nombreComun	Nombre común de la especie de fauna	varchar(45)	No	-
clase	Clase a la que pertenece la especie de fauna	varchar(45)	No	-
alimentacion	Alimentación que tiene la especie de fauna	varchar(45)	No	-
dieta	Dieta que consume la especie de fauna	varchar(100)	No	-
reproduccion	Tipo de reproducción de la especie de fauna	varchar(100)	No	-
tamaño	Tamaño aproximado de la especie de fauna	varchar(100)	No	-
longevidad	Tiempo de vida de la especie de fauna	varchar(100)	No	-
periodoIncubacion	Periodo/Tiempo de la reproducción de la especie de fauna	varchar(100)	No	-
cuidado	Cuidado que debe tener la especie de fauna	varchar(500)	No	-
cantidad	Número de especies de fauna existentes en el parque	int(11)	No	-
img1	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-
img2	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-
img3	Nombre de la imagen que se quiera mostrar en la aplicación	varchar(100)	Sí	-
idEspecie	Identificador de la especie a la que pertenece	int(11)	No	Especie

4.2 Diseño físico

El diseño físico de la base de datos consta de la creación del diagrama Entidad-Relación Extendido con base al DDD y al MER los cuales nos ayudaron a establecer las entidades, los atributos y sus relaciones. Después se creó el código SQL de la base de datos en el SGBD (Sistemas de Gestión de Bases de Datos).

4.2.1 Diagrama Entidad-Relación Extendido

El diagrama Entidad-Relación Extendido se implementó en Workbench, éste es un software que se utiliza como herramienta visual de diseños de base de datos.

En la figura 28, se visualiza el diagrama Entidad-Relación Extendido del sistema.

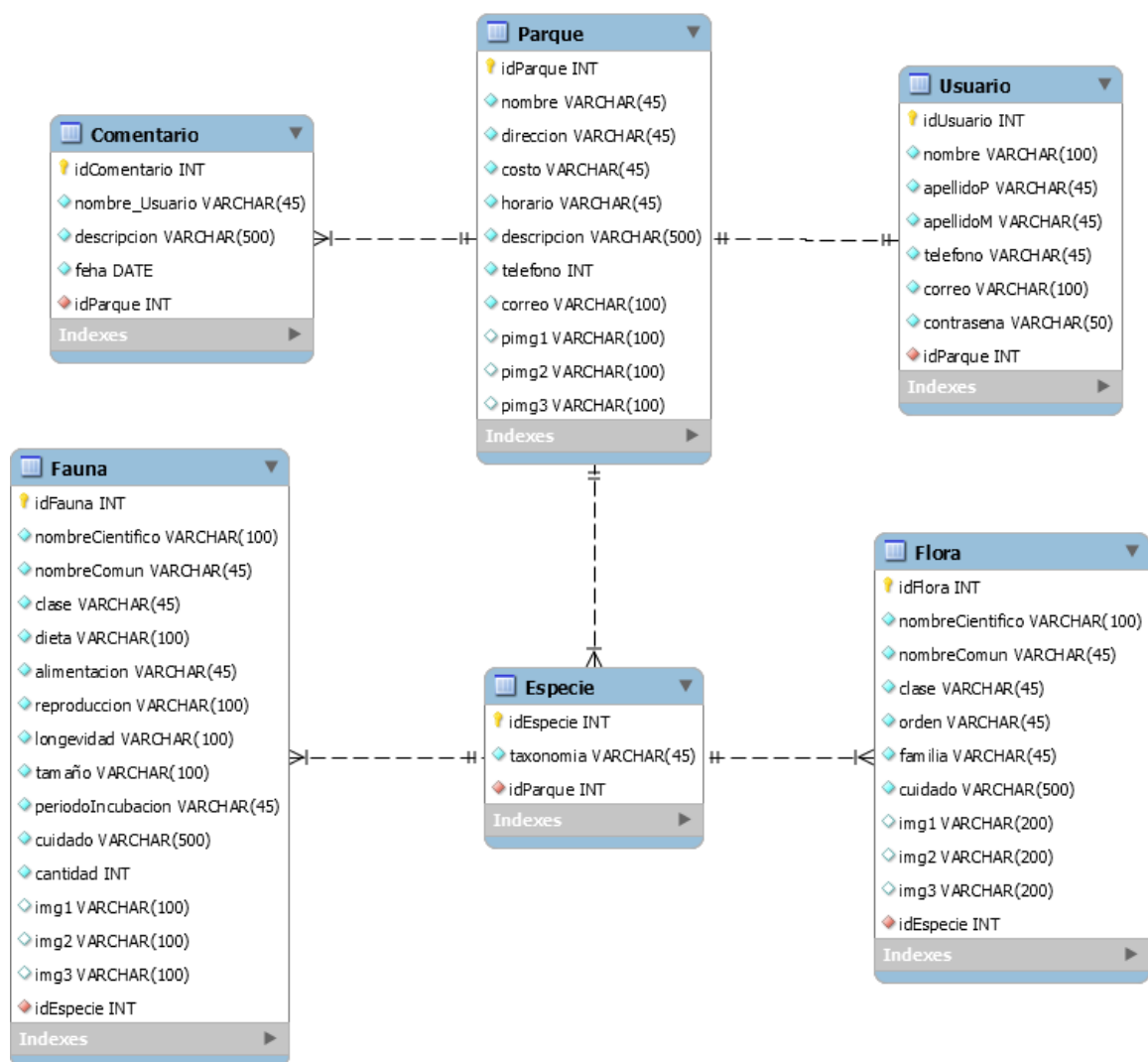


Figura 28. Diagrama Entidad-Relación Extendido.

4.2.2 Código SQL

Con base a lo anterior se implementó el código SQL en MySql que es el SGBD de nuestro sistema.

Además, se creó un procedimiento almacenado que es un programa que se almacena físicamente en una base de datos, este nos servirá de ayuda para las búsquedas de las especies

En el Anexo 1 se muestra el código SQL.

CAPÍTULO V. IMPLEMENTACIÓN Y PRUEBAS

5.1 Implementación

En este apartado se mostrará el resultado de la aplicación web, con base al Capítulo III. Análisis y Diseño.

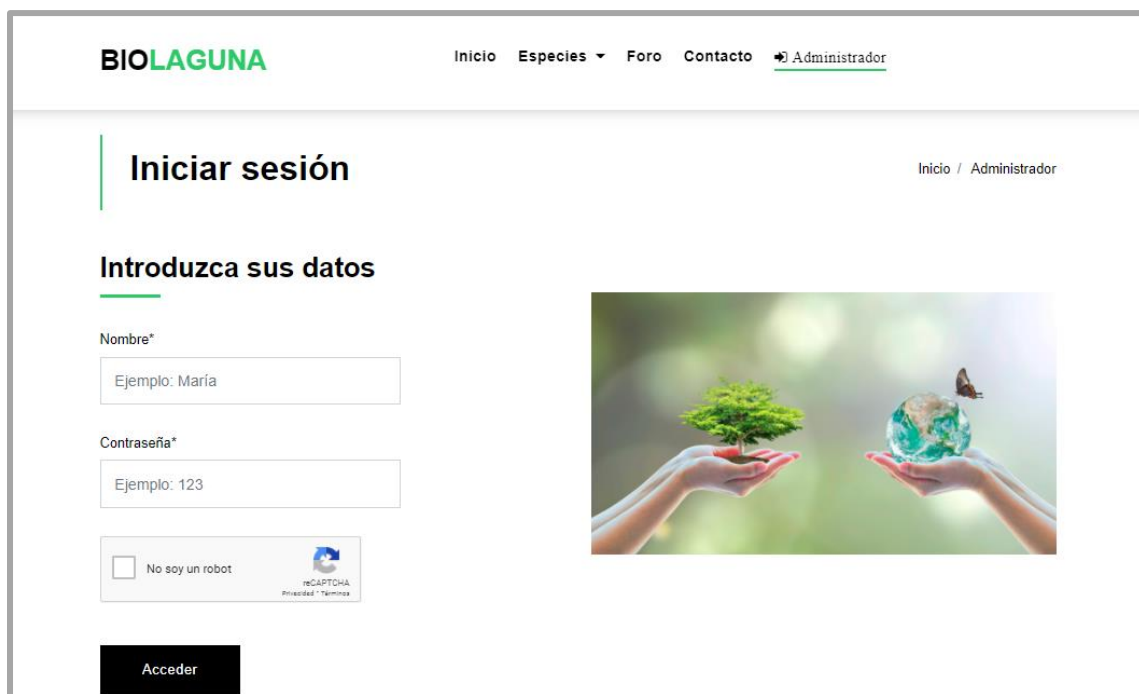
El framework utilizado para crear esta aplicación fue CodeIgniter, se instaló la biblioteca Bootstrap para el diseño de las interfaces las cuales son responsivas y la biblioteca de Grocery Crud para implementar la gestión (añadir, consultar, modificar y eliminar) de los módulos de la aplicación.

Se utilizó una paleta de colores verde, blanco y café, estos colores están relacionados con la naturaleza, el cual es el tema central de la aplicación y donde el color verde es el predominante [3].

A continuación, se puede observar la implementación del sistema de acuerdo al usuario que la utilice.

5.1.1 Administrador

Inicio de sesión: El administrador tiene que colocar su usuario y contraseña para acceder a los módulos de gestión. En la figura 29, se observa la interfaz del Inicio de Sesión del administrador.



The screenshot displays the administrator login interface for the BIOLAGUNA application. At the top, a navigation bar includes the logo and links for Inicio, Especies, Foro, Contacto, and Administrador. The main heading is 'Iniciar sesión', with a breadcrumb trail 'Inicio / Administrador'. Below this, a section titled 'Introduzca sus datos' contains two input fields: 'Nombre*' with the placeholder 'Ejemplo: María' and 'Contraseña*' with the placeholder 'Ejemplo: 123'. A CAPTCHA verification area includes a checkbox labeled 'No soy un robot' and a reCAPTCHA logo. A black 'Acceder' button is positioned at the bottom left. To the right of the form is a decorative image showing two hands holding a small tree and a globe, symbolizing environmental care.

Figura 29. Inicio de sesión (Administrador).

Inicio de Administrador: Al iniciar sesión el Administrador observa la siguiente interfaz mostrada en la figura 30. En el menú tendrá 6 botones que ayudarán a gestionar el parque, las especies de flora y fauna y el foro, y un botón para el cierre de sesión.



Figura 30. Inicio (Administrador).

Gestionar parque: El administrador observa la interfaz de la figura 31, la cual tendrá 2 botones (icono: lupa y lápiz) de lado derecho, los cuales son para consultar y editar la información del parque.

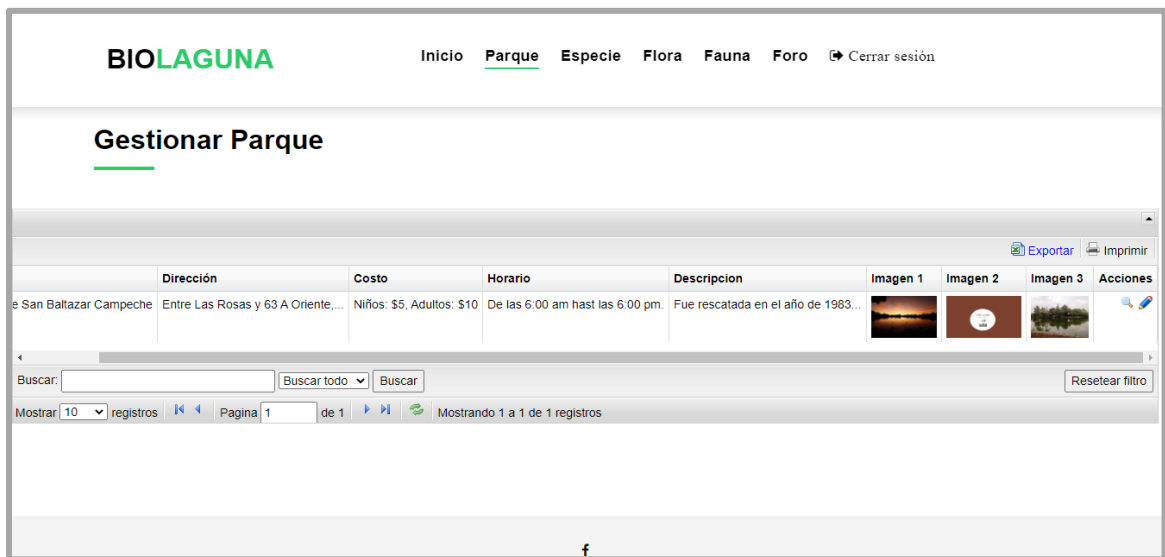
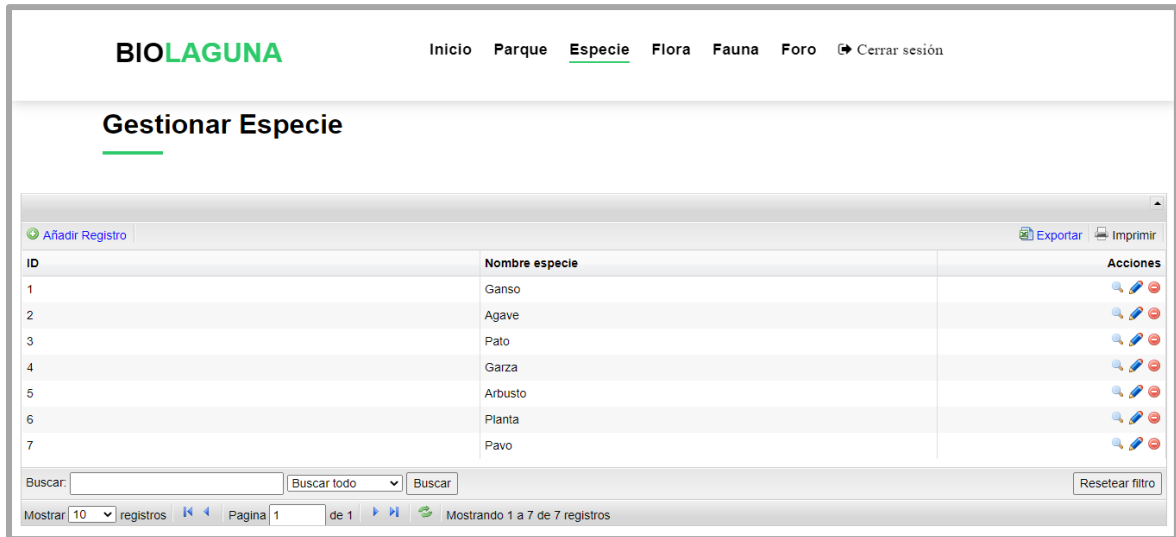


Figura 31. Gestionar Parque.

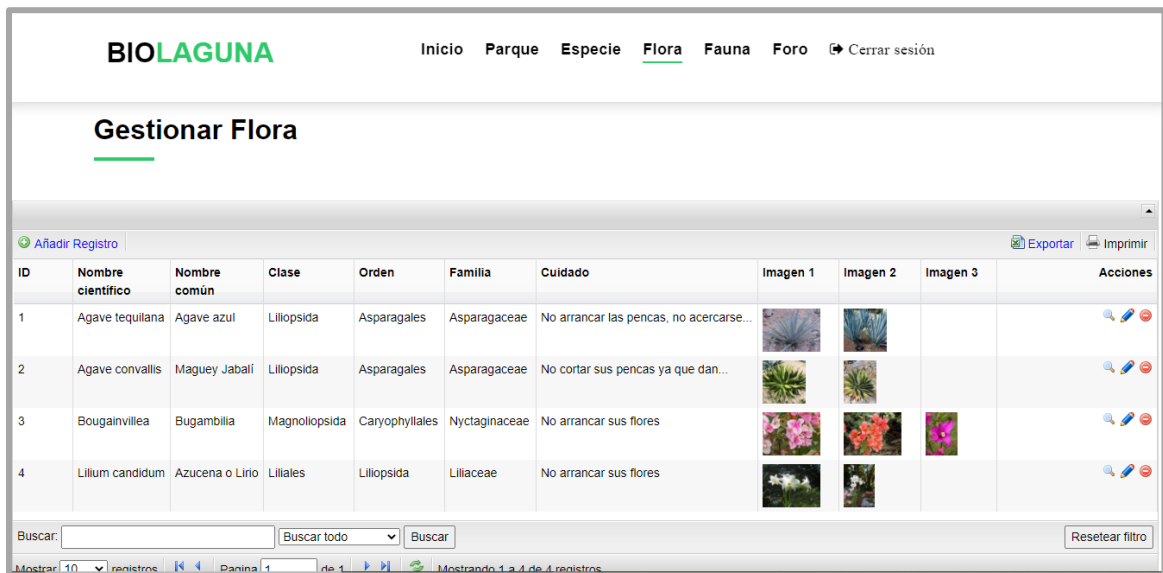
Gestionar especie: El administrador observa la interfaz de la figura 32, la cual tiene 3 botones (icono: lupa, lápiz y menor) del lado derecho los cuales son para consultar, editar y eliminar la información de las especies del parque y de lado izquierdo el botón para Añadir registro.



ID	Nombre especie	Acciones
1	Ganso	
2	Agave	
3	Pato	
4	Garza	
5	Arbusto	
6	Planta	
7	Pavo	

Figura 32. Gestionar Especie.

Gestionar Flora y Fauna: El administrador observa las interfaces de las figuras 33 y 34, las cuales tienen 3 botones (icono: lupa, lápiz y menor) del lado derecho los cuales son para consultar, editar y eliminar la información de las especies de flora o fauna del parque y de lado izquierdo el botón para Añadir registro.



ID	Nombre científico	Nombre común	Clase	Orden	Familia	Cuidado	Imagen 1	Imagen 2	Imagen 3	Acciones
1	Agave tequilana	Agave azul	Liliopsida	Asparagales	Asparagaceae	No arrancar las pencas, no acercarse...				
2	Agave convallis	Maguey Jabalí	Liliopsida	Asparagales	Asparagaceae	No cortar sus pencas ya que dan...				
3	Bougainvillea	Bugambilia	Magnoliopsida	Caryophyllales	Nyctaginaceae	No arrancar sus flores				
4	Lilium candidum	Azucena o Lirio	Liliales	Liliopsida	Liliaceae	No arrancar sus flores				

Figura 33. Gestionar Flora.

BIOLAGUNA Inicio Parque Especie Flora Fauna Foro Cerrar sesión										
Gestionar Fauna										
Añadir Registro Exportar Imprimir										
ID	Nombre científico	Nombre común	Clase	Cuidado	Cantidad (aprox)	Imagen 1	Imagen 2	Imagen 3	Especie	Acciones
1	Anser anser	Ganso común	Aves	No alimentar a los gansos con...	15				Ganso	
2	Anser anser	Ganso de Emden o blanco	Aves	No alimentar a los gansos con...	10				Ganso	
3	Anas platyrhynchos domesticus	Pato cayuga	Aves	No alimentar a los patos con otro...	16				Pato	
4	Ardea alba	Garza blanca	Aves	No aventarles objetos	8				Garza	
Buscar: Buscar todo Buscar Reseteaer filtro										
Mostrar 10 registros Pagina 1 de 1 Mostrando 1 a 4 de 4 registros										

Figura 34. Gestionar Fauna.

Gestionar comentarios: El administrador observa la interfaz de la figura 35, la cual tiene 2 botones (icono: lupa y restar) de lado derecho, los cuales son para consultar y eliminar los comentarios del foro.

BIOLAGUNA Inicio Parque Especie Flora Fauna Foro Cerrar sesión					
Gestionar Foro					
Exportar Imprimir					
ID	Nombre del usuario	Fecha	Comentario	Imagen	Acciones
1	Lulú	13/08/2020	Bonito parque		
2	María	13/08/2020	Bonitos animales		
13	Ricardo	23/08/2020	Bonito parque, muchos animales.		
Buscar: Buscar todo Buscar Reseteaer filtro					
Mostrar 10 registros Pagina 1 de 1 Mostrando 1 a 3 de 3 registros					

Figura 35. Gestionar Comentarios.

5.1.2 Visitante

Inicio de visitante: El visitante al entrar a la aplicación web ve el inicio que se observa en la figura 36, esta pantalla muestra información del parque como el nombre, la dirección, el horario, el reglamento, fotografías y el costo.

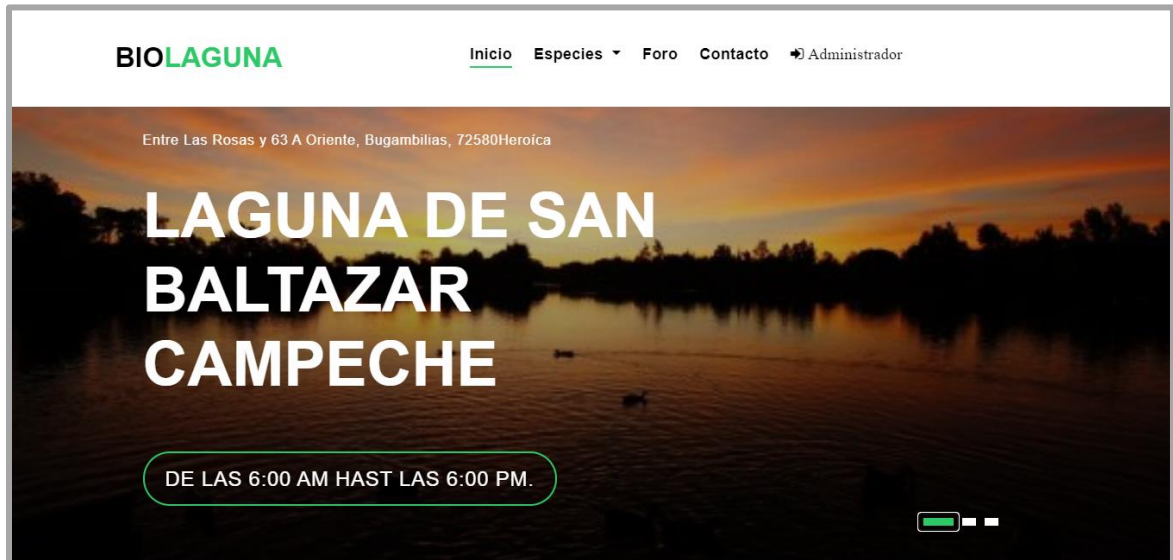


Figura 36. Inicio (Visitante).

Consultar especies de flora y fauna: El visitante observa las especies de flora y fauna registradas en el parque entrando en la interfaz de las figuras 37 y 38, en las cuales hay un botón para hacer una búsqueda con filtros, además, que se puede ver las imágenes de la especie en un carrusel.

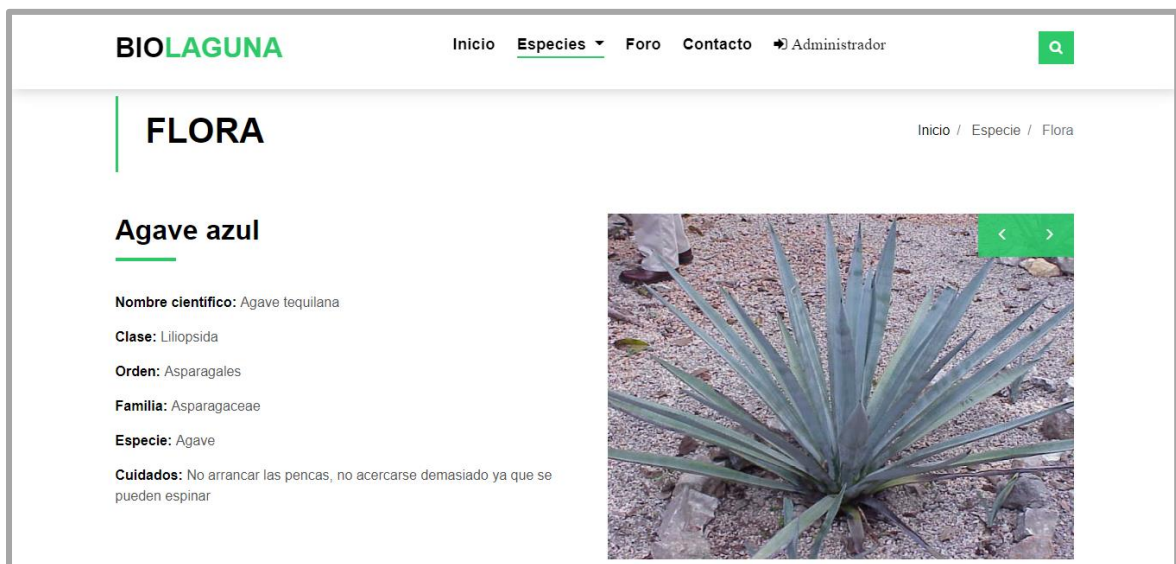


Figura 37. Consultar Flora.

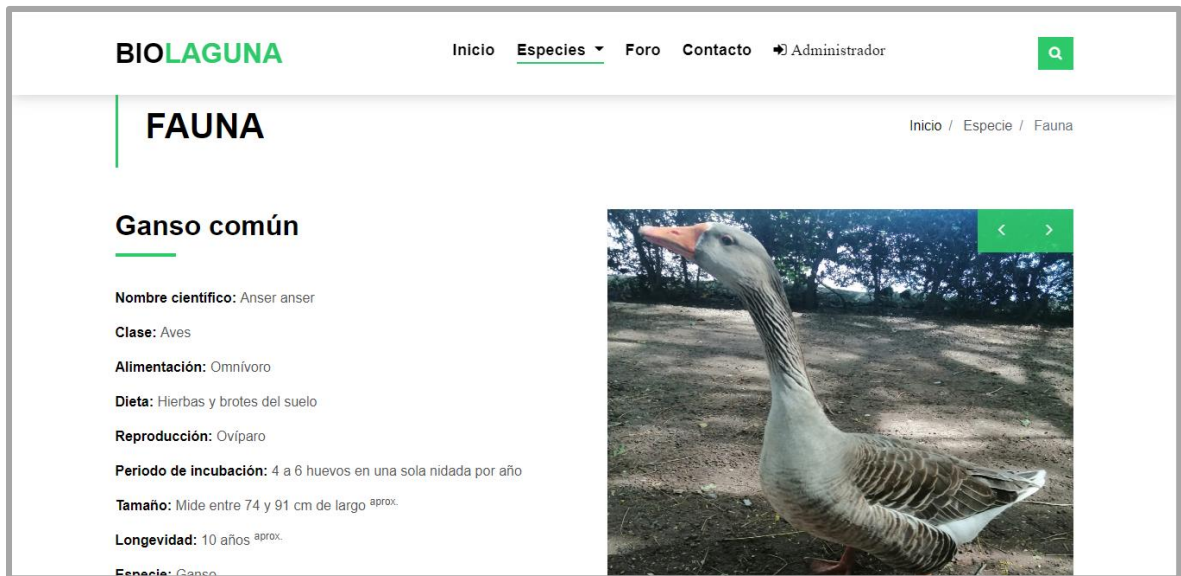


Figura 38. Consultar Fauna.

Foro: El visitante puede consultar los comentarios hechos al parque en el Foro, el cual se muestra en la figura 39, los comentarios tienen el nombre de la persona que lo escribió o podrá ser anónimo, además, pueden tener una fotografía. También, se muestra el formulario para añadir un comentario al foro.

BIOLAGUNA

[Inicio](#)
[Especies](#)
[Foro](#)
[Contacto](#)
[Administrador](#)

FORO

Inicio / Foro

Los comentarios con contenido inapropiado o que no sean referentes al parque serán eliminados.

Hacer un comentario

Nombre

Nombre (máximo 20 caracteres)

Imagen

Seleccionar archivo

Ningún archivo seleccionado

Comentario*

Escribe aquí (máximo 250 caracteres)

Aceptar

Comentarios

Está muy limpio el parque.

04/09/2020

Raúl





Figura 39. Foro.

5.2 Pruebas

En este apartado, se especificaron los tipos de pruebas hechas a la aplicación web, las entradas y los resultados que arrojaron cada una de ellas.

5.2.1 Pruebas de caja negra

En la tabla 21, se muestran las pruebas de caja negra, estas pruebas se hacen sin tomar en cuenta el funcionamiento de código del sistema, sino que se toman en cuenta las entradas y las salidas.

Tabla 21.*Pruebas de caja negra*

Código	Requisitos	Entrada	Descripción	Salida	Resultado esperado
PCN1	Consultar la información del parque.	-ID del parque	El administrador da clic en el botón de consultar la información del parque deseado.	Se muestra la siguiente información: - Nombre - Dirección - Costo - Horario - Descripción - Reglamento - Teléfono - Correo - Imagen 1 - Imagen 2 - Imagen 3	Sí
PCN2	Editar la información del parque.	Puede editar alguno de estas características - Nombre - Dirección - Costo - Horario - Descripción - Reglamento - Teléfono - Correo - Imagen 1 - Imagen 2 - Imagen 3	El administrador edita la información que requiera del parque.	Mensaje: Sus datos han sido actualizados correctamente.	Sí
PCN3	Añadir registro de especie	-Nombre especie -Parque	El administrador da clic en el botón de "Añadir registro" para agregar una especie.	Mensaje: Si el nombre de la especie no se repite "Sus datos han sido guardados correctamente". Si el nombre de la especie se repite "La especie con ese nombre ya existe"	Sí
PCN4	Consultar información de las especies.	-ID de la especie	El administrador da clic en el botón de consultar la información de la especie deseada.	-Nombre especie -Parque	Sí
PCN5	Editar información de las especies.	-Nombre especie -Parque	El administrador edita la información que requiera de la	Mensaje: Si el nombre de la especie no se repite con otro registro "Sus datos	Sí

			especie deseada.	han sido actualizados correctamente". Si el nombre de la especie se repite "La especie con ese nombre ya existe"	
PCN6	Eliminar una especie.	-ID de la especie.	El administrador da clic en el botón de eliminar el registro de la especie deseada.	La tabla de registros se actualiza sin la especie eliminada.	Sí
PCN7	Añadir registro de especie flora.	-Nombre científico -Nombre común -Clase -Orden -Familia -Cuidado -Imagen 1 (opcional) -Imagen 2 (opcional) -Imagen 3 (opcional) -Especie	El administrador da clic en el botón de "Añadir registro" para agregar una especie de flora.	Mensaje: Si el nombre científico de la especie de flora no se repite "Sus datos han sido guardados correctamente". Si el nombre de la especie de flora se repite "La especie con ese nombre ya existe"	Sí
PCN8	Consultar información de las especies de flora.	-ID de la especie de flora.	El administrador da clic en el botón de consultar la información de la especie de flora deseada.	-Nombre científico -Nombre común -Clase -Orden -Familia -Cuidado -Imagen 1 -Imagen 2 -Imagen 3 -Especie	Sí
PCN9	Editar información de las especies de flora.	-Nombre científico -Nombre común -Clase -Orden -Familia -Cuidado -Imagen 1 -Imagen 2 -Imagen 3 -Especie	El administrador edita la información que requiera de la especie de flora deseada.	Mensaje: Si el nombre científico de la especie de flora no se repite con otro registro "Sus datos han sido actualizados correctamente". Si el nombre de la especie de flora se repite "La especie con ese nombre ya existe"	Sí
PCN10	Eliminar una especie de flora.	-ID de la especie.	El administrador da clic en el botón de eliminar el	La tabla de registros se actualiza sin la	Sí

			registro de la especie de flora deseada.	especie de flora eliminada.	
PCN11	Añadir registro de especie fauna.	-Nombre científico -Nombre común -Clase -Alimentación -Dieta -Reproducción -Tamaño -Longevidad -Periodo de incubación -Cuidado -Cantidad -Imagen 1 (opcional) -Imagen 2 (opcional) -Imagen 3 (opcional) -Especie	El administrador da clic en el botón de "Añadir registro" para agregar una especie de fauna.	Mensaje: Si el nombre científico de la especie de fauna no se repite "Sus datos han sido guardados correctamente". Si el nombre de la especie de fauna se repite "La especie con ese nombre ya existe"	Sí
PCN12	Consultar información de las especies de fauna.	-ID de la especie de fauna.	El administrador da clic en el botón de consultar la información de la especie de fauna deseada.	-Nombre científico -Nombre común -Clase -Alimentación -Dieta -Reproducción -Tamaño -Longevidad -Periodo de incubación -Cuidado -Cantidad -Imagen 1 -Imagen 2 -Imagen 3 -Especie	Sí
PCN13	Editar información de las especies de fauna.	-Nombre científico -Nombre común -Clase -Alimentación -Dieta -Reproducción -Tamaño -Longevidad -Periodo de incubación -Cuidado -Cantidad -Imagen 1 -Imagen 2 -Imagen 3 -Especie	El administrador edita la información que requiera de la especie de fauna deseada.	Mensaje: Si el nombre científico de la especie de fauna no se repite con otro registro "Sus datos han sido actualizados correctamente". Si el nombre de la especie de fauna se repite "La especie con ese nombre ya existe"	Sí

PCN14	Eliminar una especie de fauna.	-ID de la especie.	El administrador da clic en el botón de eliminar el registro de la especie de fauna deseada.	La tabla de registros se actualiza sin la especie de fauna eliminada.	Sí
PCN15	Consultar información de los comentarios.	-ID del comentario.	El administrador da clic en el botón de consultar la información del comentario deseado.	-Nombre del usuario -Comentario -Fecha -Imagen -Parque	Sí
PCN16	Eliminar un comentario.	-ID del comentario.	El administrador da clic en el botón de eliminar el registro del comentario deseado.	La tabla de registros se actualiza sin el comentario eliminado.	Sí
PCN17	Consultar especies de flora.	-Flora	El visitante da clic en la opción de flora del menú.	La interfaz muestra todos los registros de flora existentes en el parque.	Sí
PCN18	Consultar especies de fauna.	-Fauna	El visitante da clic en la opción de fauna del menú.	La interfaz muestra todos los registros de fauna existen en el parque.	Sí
PCN19	Búsqueda de flora	-Nombre científico (opcional) -Nombre común (opcional) -Clase (opcional) -Ordenar	El visitante da clic en el botón de buscar y llena los campos que requiera para filtrar la búsqueda de flora.	La interfaz muestra todos los registros dependiendo de los filtros de la búsqueda.	Sí
PCN20	Búsqueda de fauna	-Nombre científico (opcional) -Nombre común (opcional) -Clase (opcional) -Ordenar	El visitante da clic en el botón de buscar y llena los campos que requiera para filtrar la búsqueda de fauna.	La interfaz muestra todos los registros dependiendo de los filtros de la búsqueda.	Sí
PCN21	Comentar foro	-Nombre (opcional) -Comentario -Imagen (opcional)	El visitante llena el formulario para escribir un comentario.	Mensaje: “Haz hecho un comentario en el foro”, además, el comentario será reflejado en la interfaz del foro.	Sí

CAPÍTULO VI. SEGURIDAD

6.1 Seguridad

La seguridad que se implementó en la aplicación web consta de varios puntos, a continuación, se especificará cómo se llevaron a cabo.

Cifrado de contraseña: La contraseña del administrador fue cifrada con la función de MySQL SHA1(), la cual es una función hash para proteger el acceso de intrusos a la aplicación web.

reCAPTCHA: Según Google es un servicio gratuito que protege sitios del spam y el abuso. Utiliza técnicas avanzadas de análisis de riesgos para diferenciar a los humanos de los robots.

Url amigable (semántica): Estas Url son fáciles de entender por el usuario, esto puede ayudar a que los usuarios recuerden la Url de la aplicación.

Inicio de sesión: Para acceder a los módulos de gestión el usuario tendrá que iniciar sesión creándose así, una variable de sesión, si esta variable no inicia no se podrá tener acceso a estos módulos.

Aviso de privacidad: Según el Instituto Nacional de Acceso a la Información (INAI) es un documento físico, electrónico o en cualquier otro formato (por ejemplo: sonoro), a través del cual el responsable informa al titular sobre la existencia y características principales del tratamiento al que serán sometidos sus datos personales. A través del aviso de privacidad se cumple el principio de información que establece la Ley Federal de Protección de Datos Personales en Posesión de los Particulares (en lo sucesivo la Ley) y su Reglamento.

La aplicación web tendrá una interfaz que establezca el aviso de seguridad para que los usuarios puedan acceder a él y estar informados de cómo se utilizaran sus datos personales.

CAPÍTULO VII. CONCLUSIÓN Y TRABAJO A FUTURO

7.1 Conclusión

La conservación del medio ambiente es un interés común que todos los seres humanos tenemos, ya que con ello estamos en equilibrio para sobrevivir y tener bienestar con nuestro entorno. El uso de las herramientas tecnológicas y las aplicaciones web, han ayudado a crear conciencia en las personas para que poco a poco vaya disminuyendo el daño de la biodiversidad con información que ayude a cuidar todo lo que nos rodea.

La aplicación web que se desarrolló, es una herramienta que apoya a la población de la ciudad de Puebla que la visite, a ser conscientes de los cuidados y la existencia de la biodiversidad que hay en La Laguna de San Baltazar Campeche.

En esta primera versión sólo se toma en cuenta este parque, pero en un futuro todos los parques de la Ciudad de Puebla podrían incluirse, para que la población tenga a la mano la información sobre la biodiversidad que habita en ellos.

La aplicación web se encuentra en un repositorio de GitHub, para que ésta pueda ser mejorada por algún programador que le agrade la idea.

7.2 Trabajo futuro

A continuación, se enlistan algunas propuestas de trabajo futuro:

Parques: Dado que esta versión solo contiene un parque, en un futuro se piensa en añadir todos los parques que existan en la Ciudad de Puebla.

Registro: El usuario visitante en esta versión, solamente podrá interactuar con información pública y se espera que en una siguiente versión puedan agregar información de flora y fauna siempre y cuando cuenten con los permisos otorgados.

Módulo de donaciones de especies: Las personas interesadas en donar alguna especie de flora o fauna podrán hacer un registro para facilitar la donación.

REFERENCIAS

Bibliografía

- [1] Booch, G., Rumbaugh, J. & Jacobson, I. (2006). *El Lenguaje Unificado de Modelado*. Madrid: PEARSON EDUCACIÓN
- [2] Braude J. (2003). *Ingeniería de Software. Una perspectiva orientada a objetos*. Alfaomega.
- [3] Heller, E. & Chamorro Mielke, J. (2017). *Psicología del color*. Barcelona: Gustavo Gili.

Mesografía

- [4] Rodríguez, C. & Dorado, R. (2015). ¿Por qué implementar Scrum? Universidad Nacional de Colombia, s. f., 134-136. Recuperado de <https://journal.universidadean.edu.co/index.php/Revistao/article/view/1253/1218>
- [5] NaturaLista. https://play.google.com/store/apps/details?id=org.inaturalist.android&hl=es_MX
- [6] SEEK. https://play.google.com/store/apps/details?id=org.inaturalist.seek&hl=es_MX
- [7] Scrum y metodología de proyectos. <https://xn--zoraidaceballosdemario-4ec.info/scrum/zoraida-ceballos-de-marino-scrum-que-es-y-para-que-sirve-esta-metodologia/>
- [8] Schwaber, K & Sutherland, J. (2016). La Guía Definitiva de Scrum: Las Reglas del Juego. Recuperado de <https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf>
- [9] White, S. & Miers, D. (2009). BPMN GUÍA DE REFERENCIA Y MODELADO. Recuperado de <https://users.dcc.uchile.cl/~nbaloian/DSS-DCC/Software/ModeladoBPMN.pdf>
- [10] Marqués, M. (2011). *Base de datos*. Recuperado de <https://libros.metabiblioteca.org/bitstream/001/353/5/978-84-693-0146-3.pdf>
- [11] CodeIgniter. https://www.codeigniter.es/user_guide/overview/appflow.html

ANEXO

Anexo 1. Código SQL

```
create database BIOLAGUNA;
use BIOLAGUNA;

-- Estructura de tabla para la tabla parque--
create table parque(
    idParque int not null AUTO_INCREMENT,
    nombre varchar(45) not null,
    direccion varchar(60) not null,
    costo varchar(45) not null,
    horario varchar(45) not null,
    descripcion varchar(500) not null,
    reglamento text not null,
    telefono varchar(20) not null,
    pimg1 varchar(100),
    pimg2 varchar(100),
    pimg3 varchar(100),
    primary key(idParque)
);

-- Estructura de tabla para la tabla usuario--
create table usuario(
    idUsuario varchar(45) not null,
    nombre varchar(50) not null,
    apellidoP varchar(45) not null,
    apellidoM varchar(45) not null,
    telefono varchar(45) not null,
```

```
    correo varchar(100) not null,  
    contrasena varchar(50) not null,  
    idParque int not null,  
    foreign key(idParque) references parque(idParque),  
    primary key(idUsuario)  
);
```

```
-- Estructura de tabla para la tabla comentario--  
create table comentario(  
    idComentario int not null AUTO_INCREMENT,  
    nombreUsuario varchar(45),  
    descripcion varchar(250) not null,  
    fecha date not null,  
    img varchar(100) NOT NULL,  
    idParque int not null,  
    foreign key(idParque) references parque(idParque),  
    primary key(idComentario)  
);
```

```
-- Estructura de tabla para la tabla especie--  
create table especie(  
    idEspecie int not null AUTO_INCREMENT,  
    nombre_especie varchar(45) not null,  
    idParque int not null,  
    foreign key(idParque) references parque(idParque),  
    primary key(idEspecie)  
);
```

```
-- Estructura de tabla para la tabla flora--
create table flora(
  idFlora int not null AUTO_INCREMENT,
  nombreCientifico varchar(100) not null,
  nombreComun varchar(45) not null,
  clase varchar(45) not null,
  orden varchar(45) not null,
  familia varchar(45) not null,
  cuidado varchar(500),
  img1 varchar(100),
  img2 varchar(100),
  img3 varchar(100),
  idEspecie int not null,
  foreign key(idEspecie) references especie(idEspecie),
  primary key(idFlora))
);
```

```
-- Estructura de tabla para la tabla fauna--
create table fauna(
  idFauna int not null AUTO_INCREMENT,
  nombreCientifico varchar(100) not null,
  nombreComun varchar(45) not null,
  clase varchar(45) not null,
  alimentacion varchar(45) not null,
  dieta varchar(100) not null,
  reproduccion varchar(100) not null,
  tamaño varchar(100) not null,
  longevidad varchar(100) nor null,
```

```

periodoIncubacion varchar(100) not null,
cuidado varchar(500) not null,
cantidad int not null,
img1 varchar(100),
img2 varchar(100),
img3 varchar(100),
idEspecie int not null,
foreign key(idEspecie) references especie(idEspecie),
primary key(idFauna))
);

-- Procedimientos--

DELIMITER $$

DROP PROCEDURE IF EXISTS 'buscar' $$

CREATE PROCEDURE buscar (IN 'tipoBuscar' INT, IN 'condiciones'
CHAR(200)) Begin

    if tipoBuscar=1 then

        SET @s = CONCAT('Select idFauna, nombreCientifico, nombreComun,
clase, alimentacion, dieta, reproduccion, tamaño, longevidad,
periodoIncubacion, cuidado, cantidad, img1, img2, img3, nombre_especie
FROM fauna AS fa INNER JOIN especie AS es ON fa.idEspecie = es.idEspecie
INNER JOIN parque AS pa ON pa.idParque = es.idParque ',condiciones);

        PREPARE stmt FROM @s;

        EXECUTE stmt;

        DEALLOCATE PREPARE stmt;

    end if;

    if tipoBuscar=2 then

        SET @s = CONCAT('Select idFlora , nombreCientifico , nombreComun ,
clase, orden, familia , cuidado , img1 , img2 , img3, nombre_especie FROM flora
AS fa INNER JOIN especie AS es ON fa.idEspecie = es.idEspecie INNER JOIN
parque AS pa ON pa.idParque = es.idParque ',condiciones);

```

```
    PREPARE stmt FROM @s;  
    EXECUTE stmt;  
    DEALLOCATE PREPARE stmt;  
end if;  
end$$  
DELIMITER ;
```