



## Documento de diseño

CONTROL DE VERSIONES					
Autor(es)	Fecha de modificación	Versión	Descripción del cambio	Revisó	Estado
AFFJ	28/04/2024	1.0	Creación del Documento	VSSL	APROBADO
VSSL	31/05/2024	1.1	Corrección del Documento		PENDIENTE

### Propósito

Detallar el cómo los requisitos de software deben ser implementados y determinar si se han abordado todos los requisitos necesarios, también proporciona al equipo de desarrollo una orientación general sobre la arquitectura del proyecto de software y los modelos a seguir, facilita a identificar los elementos de diseño que no están alineados con un requerimiento, para determinar si el elemento de diseño es necesario; se convierte en la base para limitar los cambios en el alcance de un proyecto, ya que sirve como una guía durante toda la vida del sistema para los miembros del proyecto. El documento debe ser tan detallado como sea posible, a fin de mantener el equipo de desarrollo de software centrado y alineado al objetivo del sistema.

### Arquitectura del sistema

Se mostrará la arquitectura del proyecto de trabajo terminal I, para lo cual se tomará en cuenta la definición de arquitectura del software que la IEEE std. 1471-2000, y que a la letra dice: *La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente, y los principios que orientan su diseño y evolución.*

La arquitectura de software se refiere a la “estructura general del software y las formas en la que la estructura proporciona una integridad conceptual para un sistema”. En su forma más simple, la arquitectura es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan; así como, la estructura de los datos que utilizan los componentes. La especificación de la arquitectura del software es importante porque:

- Las representaciones de la arquitectura del software permiten la comunicación entre todas las partes integrantes o participantes interesadas en el desarrollo de un sistema de cómputo.
- Destaca las decisiones iniciales relacionadas con el diseño que tendrán un impacto profundo en todo el trabajo de la Ingeniería de Software.
- Constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes.



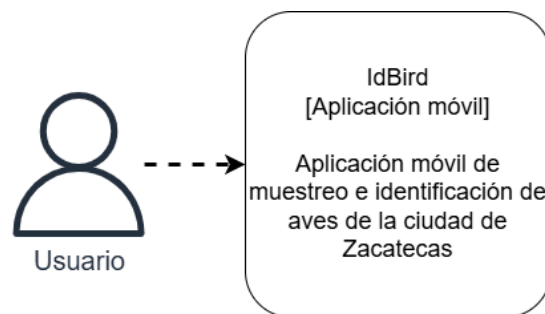
## Documento de diseño

La arquitectura 4C (Contexto, Contenedores, Componentes, Clases) se utiliza en este proyecto debido a su capacidad para ofrecer una visión clara y comprensible del sistema en diferentes niveles de detalle. Cada uno de los niveles de la arquitectura 4C proporciona una perspectiva distinta y esencial del sistema, lo que facilita su comprensión, desarrollo y mantenimiento.

El uso de la arquitectura 4C permite una comprensión progresiva y detallada del sistema, desde una vista general del contexto hasta la implementación específica de clases. Esto no solo mejora la comunicación entre los diferentes miembros del equipo, sino que también asegura que el sistema esté bien organizado y sea mantenible a largo plazo. La clara separación y definición de responsabilidades en cada nivel de la arquitectura 4C también ayuda a identificar y resolver problemas de manera más eficiente, mejorando la calidad y la fiabilidad del sistema.

- **Diagrama de Contexto:**

El diagrama de contexto nos permite identificar y definir las interacciones del sistema con los actores externos. Esto es crucial para entender cómo los usuarios y otros sistemas interactúan con IdBird, proporcionando una visión global del entorno del sistema.



El diagrama presentado es un diagrama de contexto que describe la relación entre el usuario y la aplicación móvil IdBird, enfocada en el muestreo e identificación de aves en la ciudad de Zacatecas.

### Descripción del Diagrama de Contexto

#### Usuario:

Representado por un ícono de persona, el usuario es cualquier individuo que interactúa con la aplicación móvil IdBird. Este puede ser un observador de aves, investigador, estudiante o cualquier persona interesada en registrar y conocer más sobre las aves de Zacatecas.

#### IdBird (Aplicación móvil):

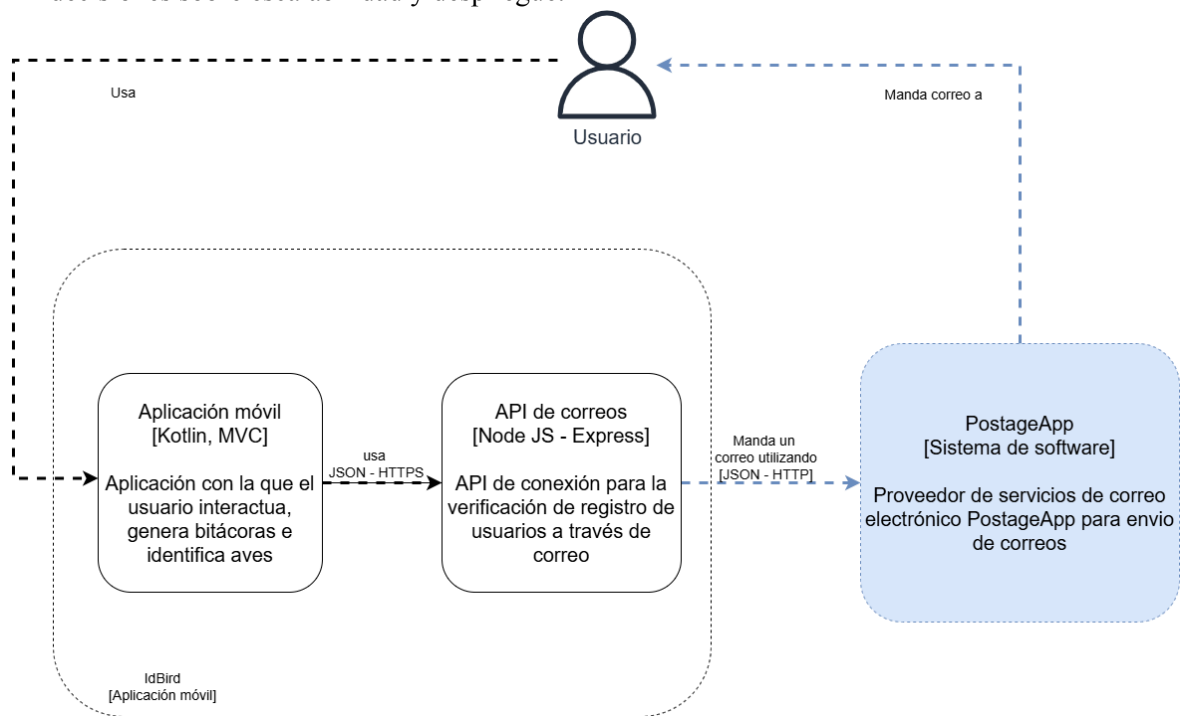


## Documento de diseño

Representada como un sistema dentro de un recuadro, IdBird es una aplicación móvil desarrollada específicamente para facilitar el muestreo e identificación de aves en Zacatecas. La aplicación proporciona herramientas para registrar avistamientos de aves, tomar fotografías, ingresar datos de observación y obtener identificaciones precisas de las especies observadas.

- **Diagrama de contenedores:**

El diagrama de contenedores descompone el sistema en sus principales bloques de construcción, como aplicaciones web, bases de datos, y servicios backend. Esto ayuda a visualizar la infraestructura del sistema y cómo los diferentes contenedores se comunican entre sí, permitiendo una gestión efectiva de la arquitectura del sistema y facilitando decisiones sobre escalabilidad y despliegue.



El diagrama presentado es un diagrama de contenedor que ilustra la arquitectura de alto nivel de la aplicación móvil IdBird y sus interacciones con otros sistemas. Este diagrama muestra cómo diferentes componentes del sistema están organizados e interactúan entre sí y con los usuarios.

### Descripción del Diagrama de Contenedor



## Documento de diseño

---

### Usuario:

Representado por un ícono de persona, el usuario interactúa con la aplicación móvil IdBird. Los usuarios pueden ser observadores de aves, investigadores, estudiantes, o cualquier persona interesada en el muestreo y la identificación de aves.

### Aplicación móvil IdBird (Kotlin, MVC):

Este contenedor representa la aplicación móvil desarrollada en Kotlin utilizando el patrón de arquitectura MVC (Modelo-Vista-Controlador). Es la interfaz principal con la que el usuario interactúa para generar bitácoras e identificar aves. La aplicación permite a los usuarios registrar observaciones, tomar fotos de aves y obtener identificaciones.

### API de correos (Node JS - Express):

Este contenedor es una API desarrollada en Node.js con el framework Express. Su función principal es gestionar la verificación del registro de usuarios a través del envío de correos electrónicos. La aplicación móvil IdBird se comunica con esta API para enviar solicitudes de verificación de registro cuando un usuario se registra en la aplicación.

La aplicación móvil IdBird se comunica con la API de correos mediante solicitudes JSON a través de HTTPS. Esta interacción es crucial para las funcionalidades de registro y verificación de usuarios, donde la aplicación necesita enviar correos electrónicos de confirmación a los nuevos usuarios.

### PostageApp (Sistema de software):

PostageApp es un proveedor de servicios de correo electrónico que la API de correos utiliza para enviar correos electrónicos. La API de correos envía las solicitudes de envío de correos a PostageApp utilizando el formato JSON a través del protocolo HTTPS. PostageApp luego maneja el envío real de los correos electrónicos a los usuarios.

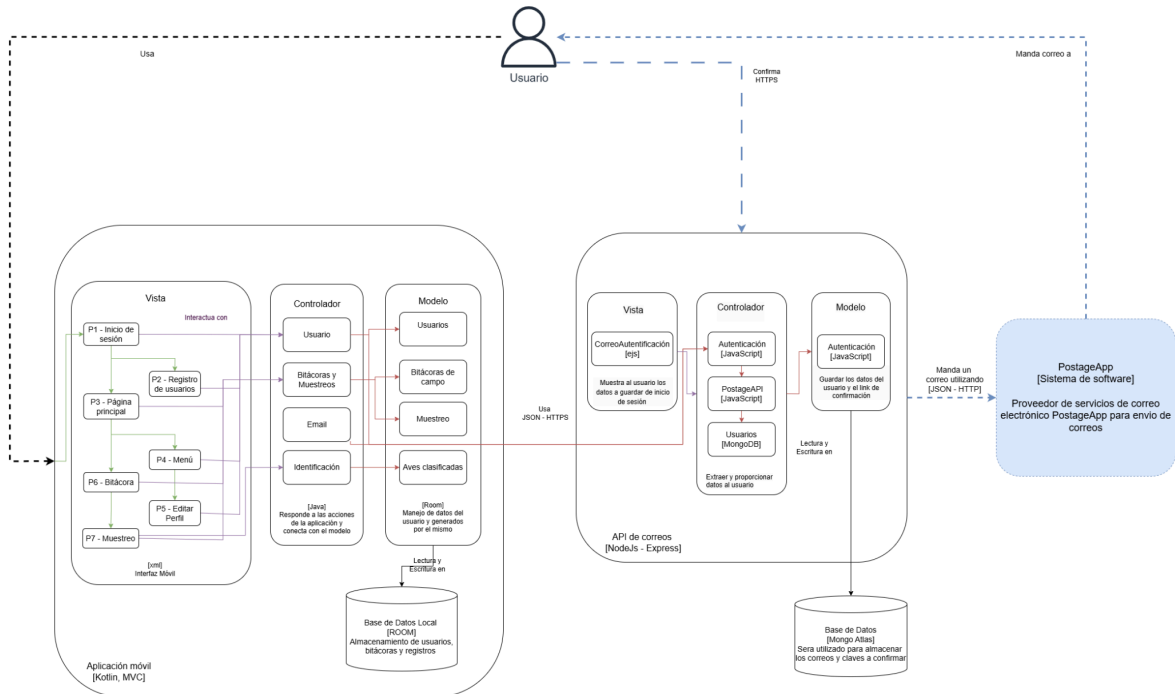
La API de correos envía solicitudes de envío de correos a PostageApp utilizando JSON a través de HTTPS. PostageApp, como proveedor de servicios de correo electrónico, procesa estas solicitudes y envía los correos electrónicos a los usuarios, completando así el proceso de verificación de registro.

- **Diagrama de componentes:**

Dentro de cada contenedor, el diagrama de componentes desglosa el sistema en componentes más pequeños y específicos que realizan funciones particulares. Esta perspectiva es esencial para el desarrollo y la asignación de responsabilidades a diferentes partes del sistema, asegurando que cada componente esté bien definido y que su función esté claramente entendida.



## Documento de diseño



El diagrama de componentes del proyecto IdBird muestra la estructura interna del sistema, dividiendo el sistema en componentes específicos que realizan funciones particulares. A continuación se describe cada parte del diagrama y sus interacciones:

### Aplicación Móvil (Kotlin, MVC):

#### 1. Vista:

- P1 - Inicio de sesión: Pantalla donde el usuario ingresa sus credenciales para acceder a la aplicación.
- P2 - Registro de usuarios: Pantalla donde el usuario puede registrarse proporcionando la información requerida.
- P3 - Página principal: Pantalla principal que muestra las opciones disponibles para el usuario.
- P4 - Menú: Menú de navegación que permite al usuario acceder a diferentes secciones de la aplicación.
- P5 - Editar Perfil: Pantalla donde el usuario puede actualizar su información personal.
- P6 - Bitácora: Pantalla que muestra las bitácoras registradas por el usuario.
- P7 - Muestreo: Pantalla que muestra los muestreos realizados y permite agregar nuevos muestreos.

#### 2. Controlador:

- Usuario: Gestiona las acciones relacionadas con el perfil del usuario.



## Documento de diseño

---

- Bitácoras y Muestreos: Gestiona las acciones relacionadas con la creación, actualización y eliminación de bitácoras y muestreos.
- Email: Maneja el envío de correos electrónicos para confirmaciones y notificaciones.
- Identificación: Gestiona el proceso de identificación de aves a partir de imágenes.
- Aves Clasificadas: Maneja la información de las aves que han sido clasificadas.

### 3. Modelo:

- Usuarios: Gestiona los datos del usuario, como credenciales y perfil.
- Bitácoras de Campo: Gestiona la información de las bitácoras creadas por el usuario.
- Muestreo: Gestiona los datos relacionados con los muestreos realizados por el usuario.
- Base de Datos Local (ROOM): Almacena localmente la información de usuarios, bitácoras y muestreos.

### API de Correos (NodeJS - Express):

#### 1. Vista:

- CorreoAutenticación (ejs): Muestra datos relacionados con la confirmación de correo electrónico y autenticación de usuario.

#### 2. Controlador:

- Autenticación (JavaScript): Gestiona la autenticación de usuarios, incluyendo registro e inicio de sesión.
- PostageAPI (JavaScript): Maneja el envío de correos electrónicos utilizando el servicio PostageApp.
- Usuarios (MongoDB): Gestiona la información de los usuarios en la base de datos global.

#### 3. Modelo:

- Autenticación (JavaScript): Gestiona la lógica relacionada con la autenticación de usuarios.
- Usuarios (MongoDB): Almacena la información de los usuarios y las claves de confirmación.

### PostageApp (Sistema de software):

1. Proveedores de servicios de correo electrónico: Maneja el envío de correos electrónicos utilizando JSON - HTTP para la comunicación.

### Flujo de Interacción

#### 1. Usuario y Aplicación Móvil:

- El usuario interactúa con las pantallas de la aplicación móvil para iniciar sesión, registrar usuarios, ver y editar bitácoras y muestreos, y actualizar su perfil.



## Documento de diseño

- La vista de la aplicación maneja estas interacciones y envía las solicitudes correspondientes a los controladores.

### 2. Controladores de la Aplicación Móvil:

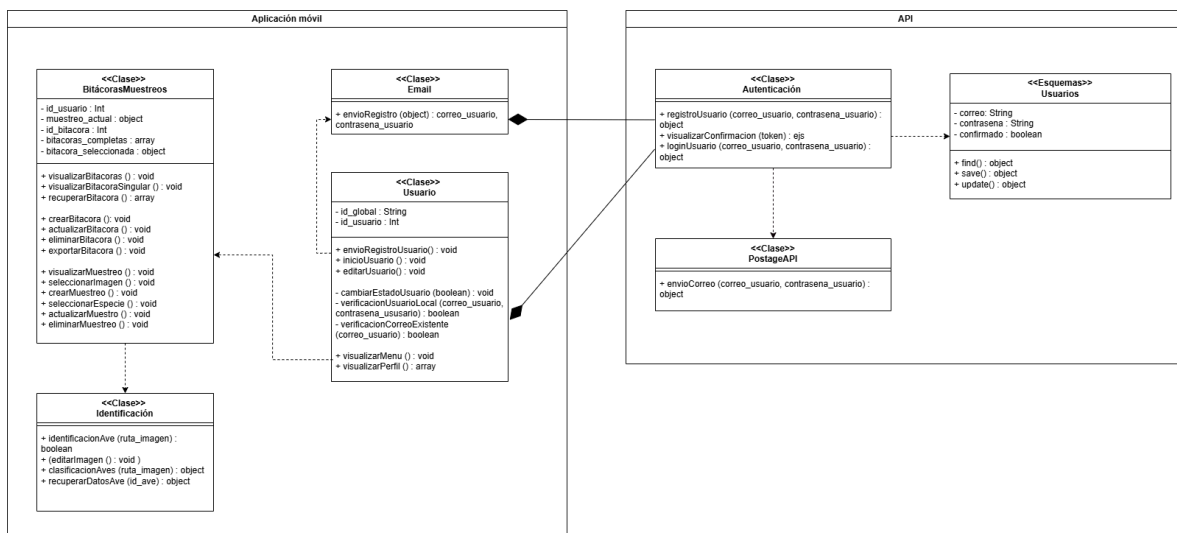
- Los controladores responden a las acciones del usuario y actualizan el modelo según sea necesario.
- Los controladores también interactúan con el API de correos para enviar correos de confirmación y notificaciones.

### API de Correos y PostageApp:

- El API de correos gestiona las solicitudes de autenticación y el envío de correos electrónicos.
- El PostageApp recibe las solicitudes del API de correos y envía los correos electrónicos a los usuarios.

### • Diagrama de clases:

El diagrama de clases ofrece un nivel detallado, mostrando la estructura interna de los componentes en términos de clases y sus relaciones. Este detalle es fundamental para los desarrolladores, ya que proporciona un mapa claro de la implementación del código, facilitando el desarrollo y el mantenimiento del software.



El diagrama de arquitectura de clases ilustra la estructura y las interacciones entre las clases en la aplicación móvil y su API para la detección de aves. Aquí se proporciona una descripción resumida y detallada de las principales clases y sus relaciones.



## Documento de diseño

---

### Aplicación Móvil

La aplicación móvil está compuesta por varias clases que gestionan diferentes aspectos de la funcionalidad de la aplicación.

La clase BitácorasMuestreos es fundamental, ya que maneja la visualización, creación, actualización y eliminación de bitácoras y muestreos. Esta clase contiene atributos que incluyen el identificador del usuario, el muestreo actual, el identificador de la bitácora y arrays para almacenar bitácoras completas y la bitácora seleccionada. Sus métodos permiten al usuario visualizar todas las bitácoras, crear nuevas, actualizar las existentes, y exportarlas en formato PDF. Además, facilita la gestión de los muestreos al permitir seleccionar imágenes y especies, así como actualizar y eliminar muestreos.

La clase Usuario maneja la información y las acciones relacionadas con los usuarios. Posee atributos como el identificador global y el identificador del usuario, y métodos para registrar usuarios, iniciar sesión, editar información, verificar la existencia de correos y cambiar el estado del usuario. También incluye métodos para mostrar el menú y el perfil del usuario.

Para la gestión del correo electrónico, la clase Email envía registros de usuario, conectando con la API para completar el proceso de autenticación. En cuanto a la identificación de aves, la clase Identificación proporciona métodos para identificar aves a partir de imágenes, editar imágenes y clasificar aves, además de recuperar datos específicos de aves identificadas.

Finalmente, la clase Aves Clasificadas almacena información sobre las aves, incluyendo su identificador, nombre, color, dimensiones, nivel de endemismo y nivel de peligro de extinción.

### API

La API se compone principalmente de las clases Autenticación y PostageAPI, junto con un esquema para los usuarios. La clase Autenticación gestiona el registro y el inicio de sesión de los usuarios, y muestra la confirmación del registro mediante un token. La clase PostageAPI es responsable del envío de correos electrónicos durante el proceso de registro y autenticación.

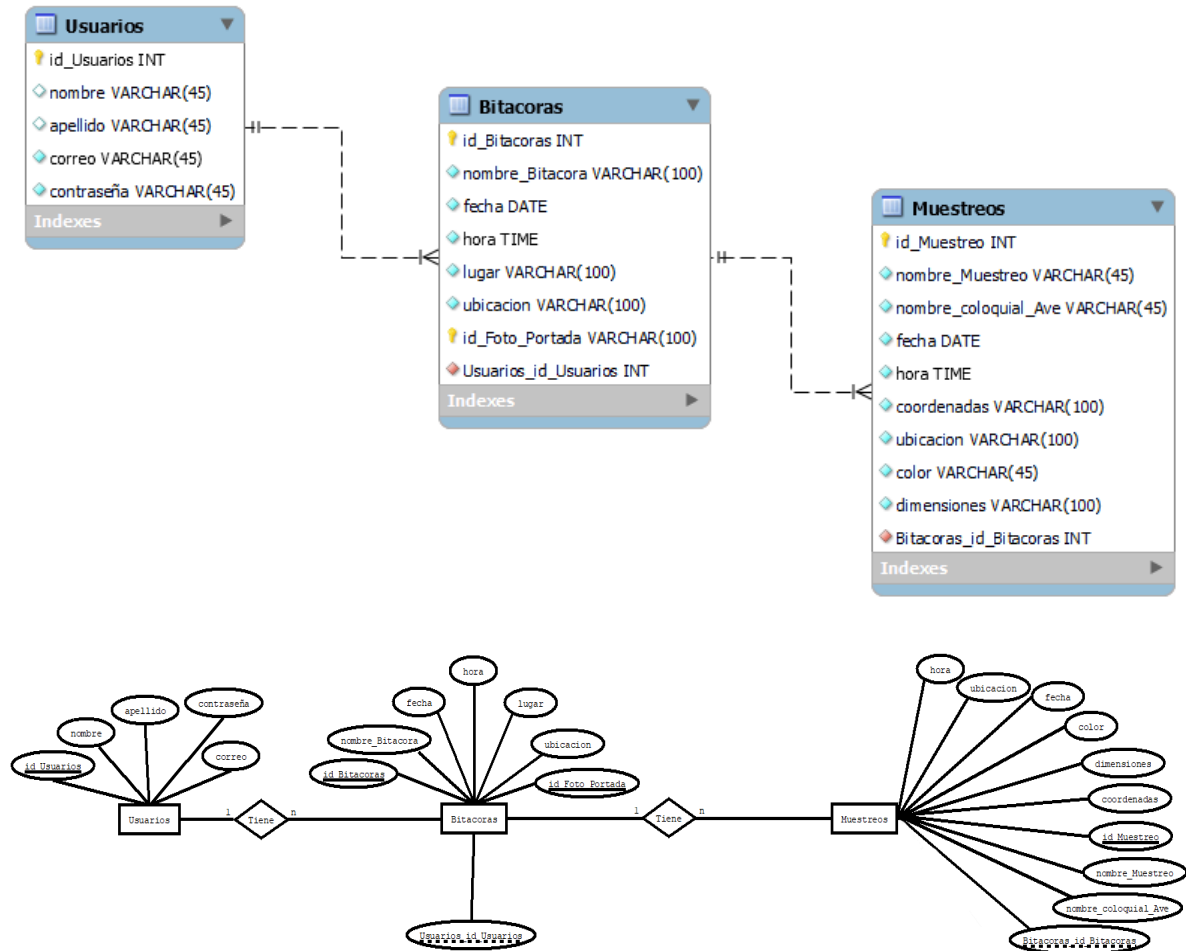
El Esquema de Usuarios define la estructura de los datos de usuario, incluyendo el correo, contraseña y el estado de confirmación. Los métodos en este esquema permiten encontrar, guardar y actualizar los datos de los usuarios.





## Documento de diseño

### Diseño de la base de datos.



El diagrama presentado es un modelo de base de datos relacional que estructura la información necesaria para una aplicación de detección de aves. Este modelo incluye tres tablas principales: Usuarios, Bitacoras y Muestreos, y define las relaciones entre ellas para asegurar la integridad y la eficiencia en la gestión de datos.

La tabla Usuarios contiene información básica sobre los usuarios de la aplicación, incluyendo su identificador único (id\_Usuarios), nombre, apellido, correo electrónico y contraseña. Cada usuario puede crear múltiples bitácoras, lo que se refleja en la relación uno a muchos entre las tablas Usuarios y Bitacoras.



## Documento de diseño

La tabla Bitacoras almacena registros de las observaciones de aves. Cada bitácora tiene un identificador único (id\_Bitacoras) y está vinculada a un usuario específico mediante la clave foránea Usuarios\_id\_Usuarios. Además, la tabla Bitacoras incluye detalles como el nombre de la bitácora, la fecha y hora de la observación, el lugar y la ubicación específica de la misma, y un identificador para la foto de portada. La relación entre Usuarios y Bitacoras asegura que cada bitácora esté vinculada a un solo usuario, permitiendo que un usuario tenga múltiples bitácoras.

La tabla Muestreos guarda información detallada sobre cada muestreo individual realizado dentro de una bitácora. Cada muestreo tiene su propio identificador único (id\_Muestreo) y está asociado con una bitácora específica a través de la clave foránea Bitacoras\_id\_Bitacoras. Los campos en la tabla Muestreos incluyen el nombre del muestreo, el nombre coloquial del ave identificada, la fecha y hora del muestreo, las coordenadas geográficas del lugar, la ubicación específica, el color y las dimensiones del ave observada. La relación entre Bitácoras y Muestreos garantiza que cada muestreo pertenezca a una sola bitácora, permitiendo que una bitácora contenga múltiples muestreos.

### Manejo de archivos.



El diagrama representa la estructura de directorios y el manejo de archivos de la aplicación IdBird, que organiza las bitácoras de campo y sus respectivos contenidos de manera eficiente. A continuación, se detalla la estructura y la funcionalidad de cada uno de los directorios:

#### Estructura de Directorios

##### 1. Directorio principal: IdBird

Este es el directorio raíz que contiene toda la información relacionada con la aplicación IdBird.



## Documento de diseño

---

### 2. Subdirectorio: Bitácoras de campo

Dentro de este subdirectorio se encuentran todas las bitácoras creadas por los usuarios.

### 3. Sub-subdirectorio: Bitácoras

Ruta: ./IdBird/Bitacoras de Campo/Bitacoras/

Contiene todas las bitácoras creadas por los usuarios, organizadas de manera que se pueda acceder fácilmente a cada registro.

### 4. Sub-subdirectorio: Muestreos

Ruta: ./IdBird/Bitacoras de Campo/Bitacoras\_N/Muestreos/

Este directorio almacena las imágenes subidas o tomadas por el usuario para un muestreo específico. Cada bitácora puede tener múltiples muestreos asociados, y este directorio ayuda a mantener organizadas las imágenes relacionadas con cada uno.

### 5. Sub-subdirectorio: Bitácora PDF

Ruta: ./IdBird/Bitacoras de Campo/Bitacoras\_N/Bitacora PDF/

Almacena los archivos PDF de las bitácoras generados por la aplicación. Estos PDFs son versiones exportables de las bitácoras, facilitando su revisión y compartición. Almacenar los PDFs en un directorio separado asegura que los documentos sean fáciles de localizar y gestionar.

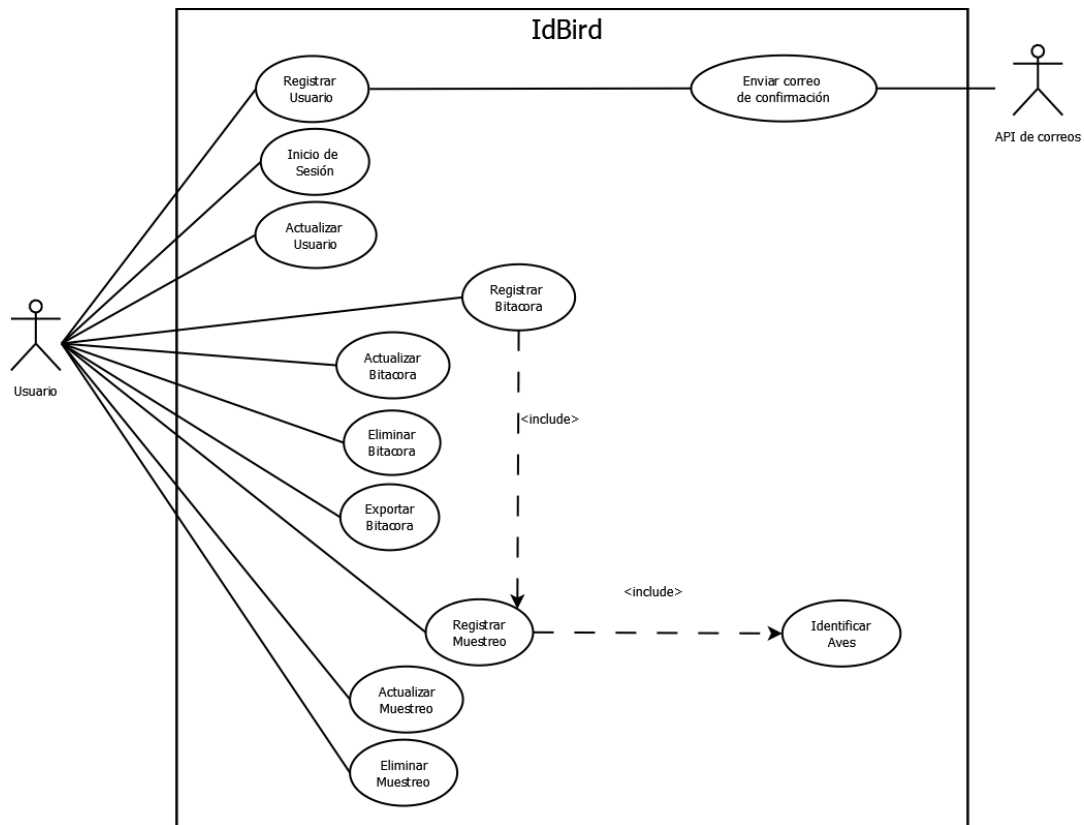


## Documento de diseño

### Diagramas UML

#### Diagramas UML de comportamiento

- Diagrama de caso de uso:



El diagrama presentado es un diagrama de casos de uso que ilustra las interacciones entre el usuario y la aplicación móvil IdBird, así como la integración con el servicio PostageApp para el envío de correos de confirmación. Este diagrama es útil para comprender las funcionalidades clave de la aplicación desde la perspectiva del usuario.

#### Descripción del Diagrama de Casos de Uso

##### Actores:

- Usuario: Representado por un ícono de persona, el usuario interactúa con la aplicación móvil IdBird para realizar diversas tareas relacionadas con el registro, gestión de bitácoras y muestreos, e identificación de aves.



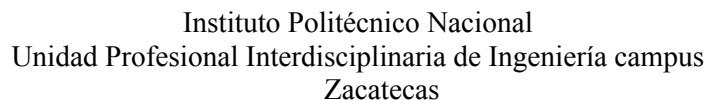
## Documento de diseño

---

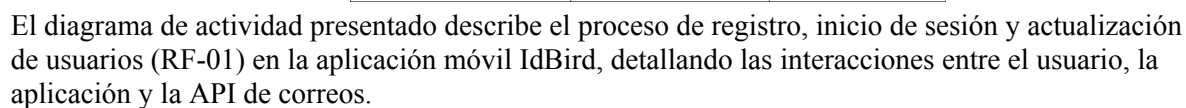
- PostageApp: Representado por otro ícono de persona, PostageApp es el sistema que se encarga de enviar correos electrónicos de confirmación. Este actor interactúa con la aplicación IdBird para completar el proceso de registro del usuario.

### Casos de Uso:

- Registrar Usuario: El usuario se registra en la aplicación IdBird. Este caso de uso incluye el envío de un correo de confirmación a través de PostageApp.
- Inicio de Sesión: El usuario inicia sesión en la aplicación.
- Actualizar Usuario: El usuario actualiza su información personal en la aplicación.
- Registrar Bitácora: El usuario crea una nueva bitácora para registrar observaciones de aves. Este caso de uso incluye la posibilidad de registrar muestreos y, eventualmente, identificar aves.
- Actualizar Bitácora: El usuario actualiza la información de una bitácora existente.
- Eliminar Bitácora: El usuario elimina una bitácora existente.
- Exportar Bitácora: El usuario exporta una bitácora en formato PDF.
- Registrar Muestreo: El usuario registra un nuevo muestreo dentro de una bitácora. Este caso de uso incluye la identificación de aves, lo que permite al usuario registrar detalles específicos sobre las aves observadas.
- Actualizar Muestreo: El usuario actualiza la información de un muestreo existente.
- Eliminar Muestreo: El usuario elimina un muestreo existente.
- Identificar Aves: El usuario utiliza la funcionalidad de la aplicación para identificar aves a partir de imágenes y otros datos.



## RF-01 Registro y Actualizacion de Usuarios





## Documento de diseño

---

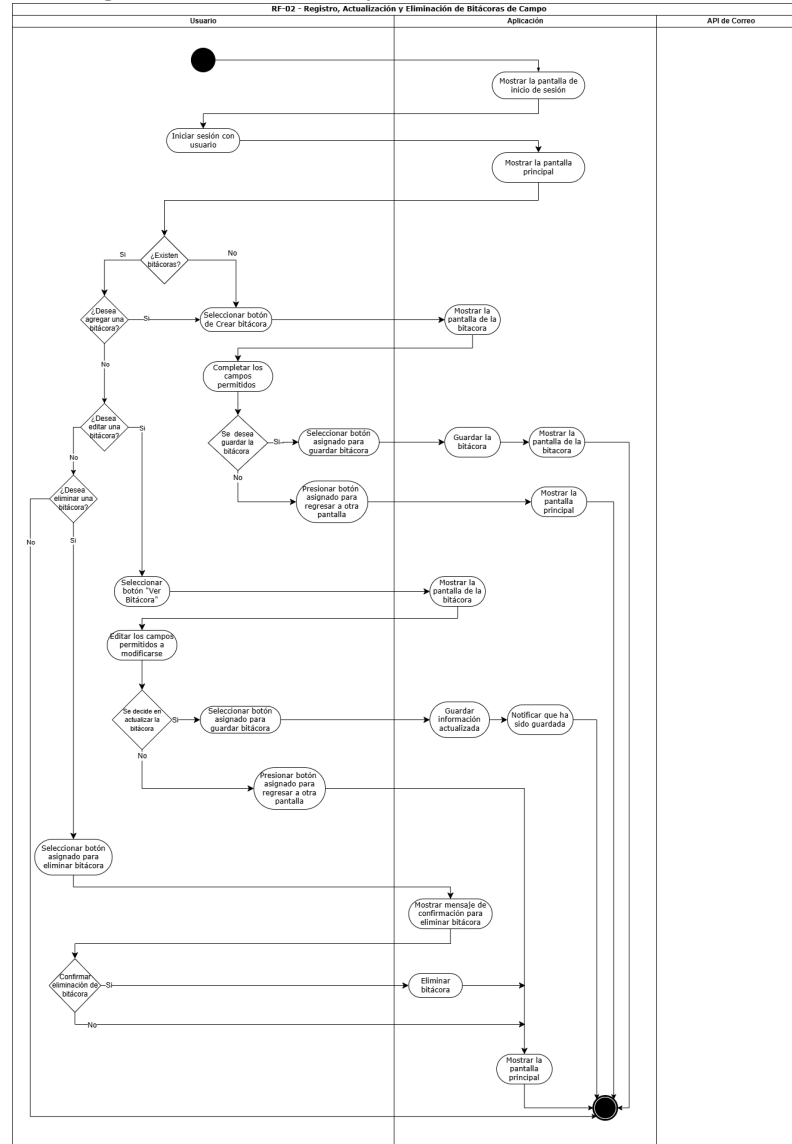
### Descripción del Diagrama de Actividad

1. Inicio del Proceso:
  - El proceso comienza con el usuario accediendo a la pantalla de inicio de sesión de la aplicación.
2. Verificación de Usuario Existente:
  - La aplicación verifica si el usuario ya existe. Si el usuario no existe, se muestra la pantalla de registro.
  - El usuario ingresa su nombre, apellidos, correo y contraseña, y selecciona el botón "Crear Cuenta".
3. Conexión con la API de Correos:
  - La aplicación se conecta con la API de correos para guardar el correo y la clave a confirmar.
  - La API de correos envía una solicitud de registro al correo ingresado por el usuario.
  - El usuario debe confirmar su correo de registro.
4. Guardar Usuario en la Base de Datos:
  - Una vez confirmado el correo, el usuario se guarda en la base de datos y se muestra nuevamente la pantalla de inicio de sesión.
5. Inicio de Sesión:
  - El usuario ingresa su correo y contraseña, y selecciona el botón "Iniciar Sesión".
  - Si es el primer inicio de sesión, la aplicación se conecta con la API de correos para comprobar el correo y contraseña registrada.
  - El usuario se guarda en la base de datos local.
6. Verificación de Usuario y Credenciales:
  - La aplicación comprueba en la base de datos si el usuario está registrado.
  - Si el usuario no está registrado o las credenciales son incorrectas, se notifica el error al usuario y se muestra nuevamente la pantalla de inicio de sesión.
  - Si las credenciales son correctas, se muestra la pantalla principal de la aplicación.
7. Actualización de Usuario:
  - Desde la pantalla principal, el usuario puede optar por actualizar su perfil seleccionando el botón de Menú y luego el botón de "Editar perfil".
  - El usuario edita su nombre y apellidos, y decide si desea actualizar su perfil.
  - Si se decide actualizar, el usuario selecciona el botón asignado para guardar la información actualizada.
8. Guardar Información Actualizada:
  - La información actualizada del usuario se guarda en la base de datos, completando así el proceso de actualización.



## Documento de diseño

### RF-02 Registro, Actualización y Eliminación de Bitácoras de Campo



El diagrama de actividad presentado describe el proceso de registro, actualización y eliminación de bitácoras de campo (RF-02) en la aplicación móvil IdBird. Este diagrama detalla los pasos que siguen los usuarios para gestionar sus bitácoras de campo y las interacciones entre el usuario y la aplicación.

#### Descripción del Diagrama de Actividad

##### 1. Inicio del Proceso:

- El proceso comienza con el usuario iniciando sesión en la aplicación.





## Documento de diseño

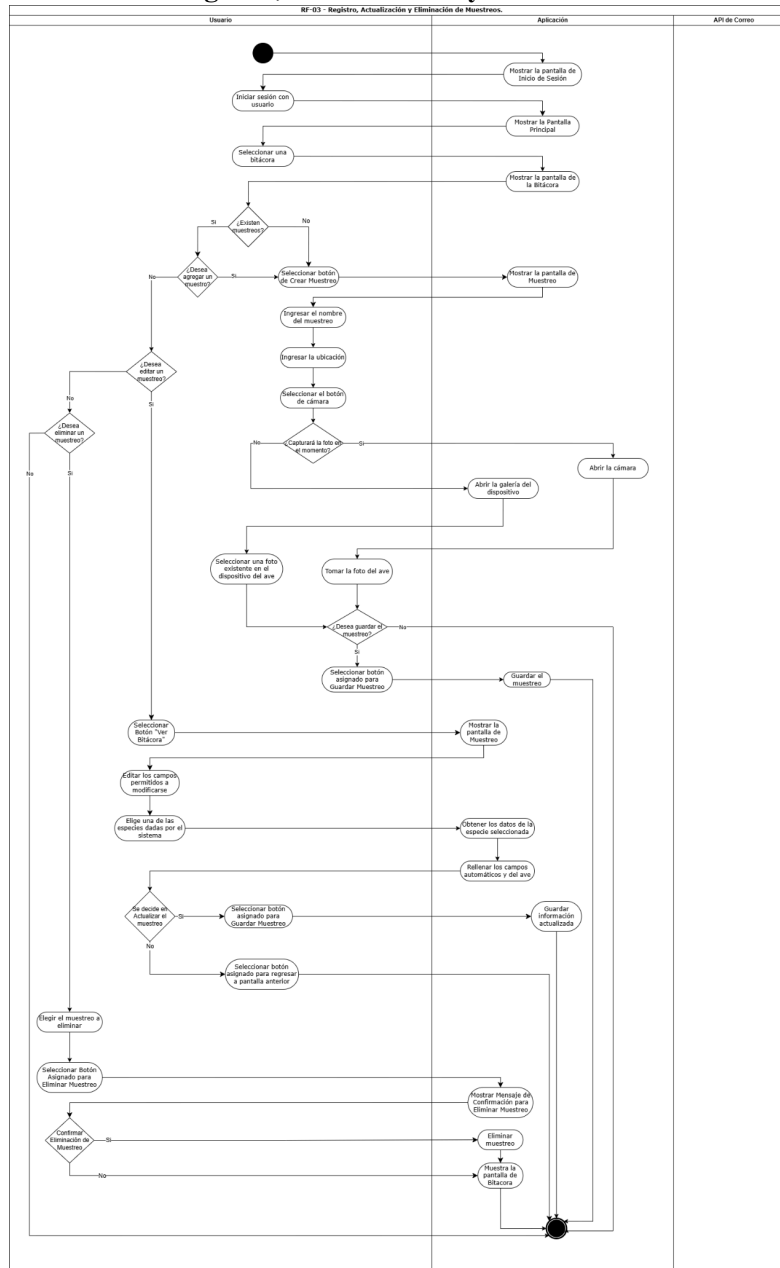
---

2. Verificación de Bitácoras Existentes:
  - Una vez que el usuario ha iniciado sesión, la aplicación muestra la pantalla principal.
  - Se verifica si el usuario tiene bitácoras existentes.
3. Registro de Bitácora:
  - Si el usuario desea agregar una nueva bitácora, selecciona el botón de "Crear Bitácora".
  - La aplicación muestra la pantalla de la bitácora, donde el usuario completa los campos permitidos.
  - Si el usuario decide guardar la bitácora, selecciona el botón asignado para guardar.
  - La bitácora se guarda y la aplicación muestra nuevamente la pantalla de la bitácora y luego la pantalla principal.
4. Actualización de Bitácora:
  - Si el usuario desea editar una bitácora existente, selecciona el botón de "Ver Bitácora".
  - La aplicación muestra la pantalla de la bitácora, donde el usuario puede editar los campos permitidos.
  - Si el usuario decide actualizar la bitácora, selecciona el botón asignado para guardar.
  - La información actualizada de la bitácora se guarda y se notifica al usuario que ha sido guardada.
5. Eliminación de Bitácora:
  - Si el usuario desea eliminar una bitácora existente, selecciona el botón asignado para eliminar.
  - La aplicación muestra un mensaje de confirmación para la eliminación de la bitácora.
  - Si el usuario confirma la eliminación, la bitácora se elimina y se muestra nuevamente la pantalla principal.



## Documento de diseño

### RF-03 Registro, Actualización y Eliminación de Muestreos



El diagrama de actividad presentado describe el proceso de registro, actualización y eliminación de muestreos (RF-03) en la aplicación móvil IdBird. Este diagrama detalla los pasos que siguen los usuarios para gestionar los muestreos asociados a las bitácoras de campo y las interacciones entre el usuario y la aplicación.



## Documento de diseño

---

### Descripción del Diagrama de Actividad

1. Inicio del Proceso:
  - El proceso comienza con el usuario iniciando sesión en la aplicación.
2. Selección de Bitácora:
  - Después de iniciar sesión, el usuario selecciona una bitácora desde la pantalla principal.
  - La aplicación muestra la pantalla de la bitácora seleccionada.
3. Verificación de Muestreos Existentes:
  - La aplicación verifica si existen muestreos asociados a la bitácora seleccionada.
4. Registro de Muestreo:
  - Si el usuario desea agregar un nuevo muestreo, selecciona el botón de "Crear Muestreo".
  - La aplicación muestra la pantalla de muestreo, donde el usuario ingresa el nombre del muestreo y la ubicación.
  - El usuario selecciona el botón de cámara para capturar una foto del ave. Puede elegir entre abrir la cámara o seleccionar una foto existente de la galería del dispositivo.
  - Si el usuario toma una nueva foto, puede optar por guardarla o repetir la toma.
  - Una vez satisfecho con la foto, el usuario selecciona el botón para guardar el muestreo.
  - La aplicación guarda el muestreo y muestra nuevamente la pantalla de muestreo, luego la pantalla de la bitácora.
5. Actualización de Muestreo:
  - Si el usuario desea editar un muestreo existente, selecciona el botón de "Ver Bitácora" y luego el muestreo que desea editar.
  - La aplicación muestra la pantalla del muestreo, donde el usuario puede editar los campos permitidos.
  - El usuario elige una de las especies dadas por el sistema, y la aplicación obtiene los datos de la especie seleccionada, rellenando automáticamente los campos del ave.
  - Si el usuario decide actualizar el muestreo, selecciona el botón para guardar los cambios.
  - La información actualizada se guarda y se notifica al usuario que ha sido guardada.
6. Eliminación de Muestreo:
  - Si el usuario desea eliminar un muestreo existente, selecciona el muestreo a eliminar y luego el botón de "Eliminar Muestreo".



Instituto Politécnico Nacional  
Unidad Profesional Interdisciplinaria de Ingeniería campus  
Zacatecas



## Documento de diseño

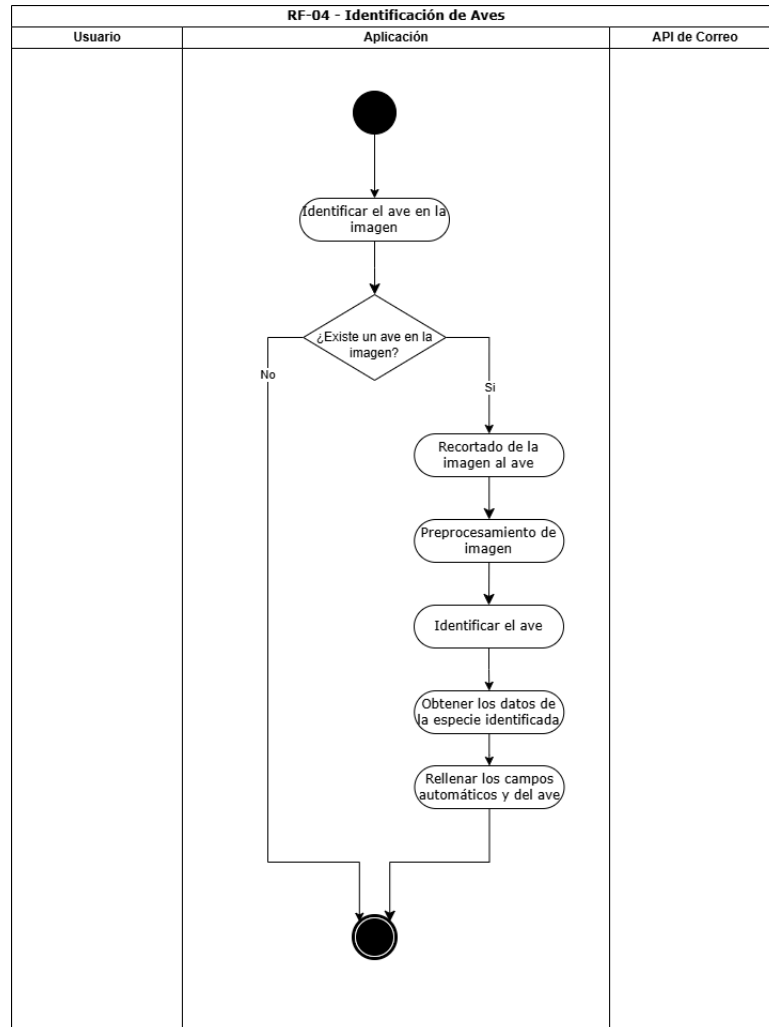
---

- La aplicación muestra un mensaje de confirmación para la eliminación del muestreo.
- Si el usuario confirma la eliminación, el muestreo se elimina y la aplicación muestra nuevamente la pantalla de la bitácora.



## Documento de diseño

### RF-04 Identificación de Aves



El diagrama de actividad presentado describe el proceso de identificación de aves (RF-04) en la aplicación móvil IdBird. Este diagrama detalla los pasos que sigue la aplicación para identificar un ave a partir de una imagen proporcionada por el usuario.

#### Descripción del Diagrama de Actividad

1. Inicio del Proceso:
  - El proceso comienza cuando el usuario inicia la función de identificación de aves en la aplicación.
2. Identificar el Ave en la Imagen:
  - El usuario sube una imagen del ave que desea identificar.
  - La aplicación pregunta si existe un ave en la imagen.



## Documento de diseño

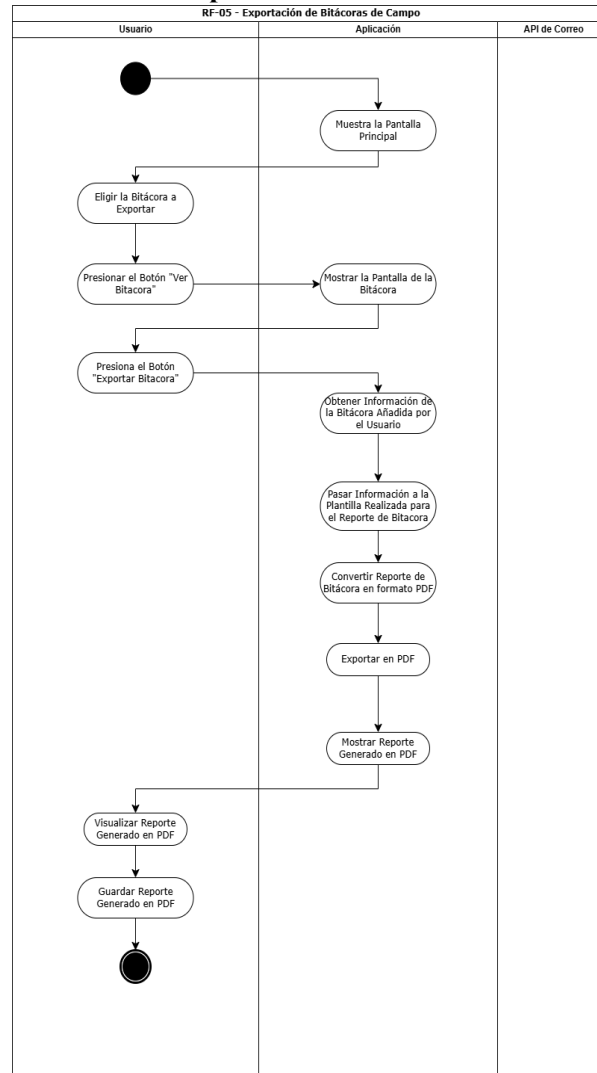
---

3. Verificación de Presencia de Ave:
  - Si no se detecta un ave en la imagen, el proceso termina en este punto.
  - Si se detecta un ave en la imagen, la aplicación procede con los siguientes pasos.
4. Recortado de la Imagen al Ave:
  - La aplicación recorta la imagen para enfocar al ave, eliminando cualquier fondo innecesario.
5. Preprocesamiento de Imagen:
  - La aplicación realiza un preprocesamiento de la imagen para mejorar la calidad y asegurar que los detalles relevantes del ave sean claros para el sistema de identificación.
6. Identificación del Ave:
  - La aplicación utiliza técnicas de reconocimiento de imágenes para identificar el ave en la imagen.
7. Obtener los Datos de la Especie Identificada:
  - Una vez identificada el ave, la aplicación obtiene datos específicos sobre la especie, como el nombre científico, características, y cualquier otra información relevante.
8. Rellenar los Campos Automáticos y del Ave:
  - La aplicación rellena automáticamente los campos con los datos obtenidos de la especie identificada.



## Documento de diseño

### RF-05 Exportación de Bitácoras de Campo



El diagrama de actividad presentado describe el proceso de exportación de bitácoras de campo (RF-05) en la aplicación móvil IdBird. Este diagrama detalla los pasos que sigue el usuario y la aplicación para convertir una bitácora en un archivo PDF y permitir su descarga.

#### Descripción del Diagrama de Actividad

1. Inicio del Proceso:
  - El proceso comienza cuando el usuario accede a la aplicación móvil IdBird.
2. Selección de Bitácora:
  - El usuario elige la bitácora que desea exportar desde la pantalla principal.



## Documento de diseño

---

- El usuario presiona el botón "Ver Bitácora" para visualizar los detalles de la bitácora seleccionada.
3. Exportación de Bitácora:
    - El usuario presiona el botón "Exportar Bitácora" para iniciar el proceso de exportación.
    - La aplicación obtiene la información de la bitácora añadida por el usuario.
  4. Generación del Reporte en PDF:
    - La aplicación pasa la información de la bitácora a una plantilla predefinida para generar el reporte de la bitácora.
    - El reporte de la bitácora se convierte en formato PDF.
    - La aplicación exporta el reporte en formato PDF.
  5. Visualización y Guardado del PDF:
    - La aplicación muestra el reporte generado en formato PDF.
    - El usuario puede visualizar el reporte generado y guardarlo en su dispositivo.

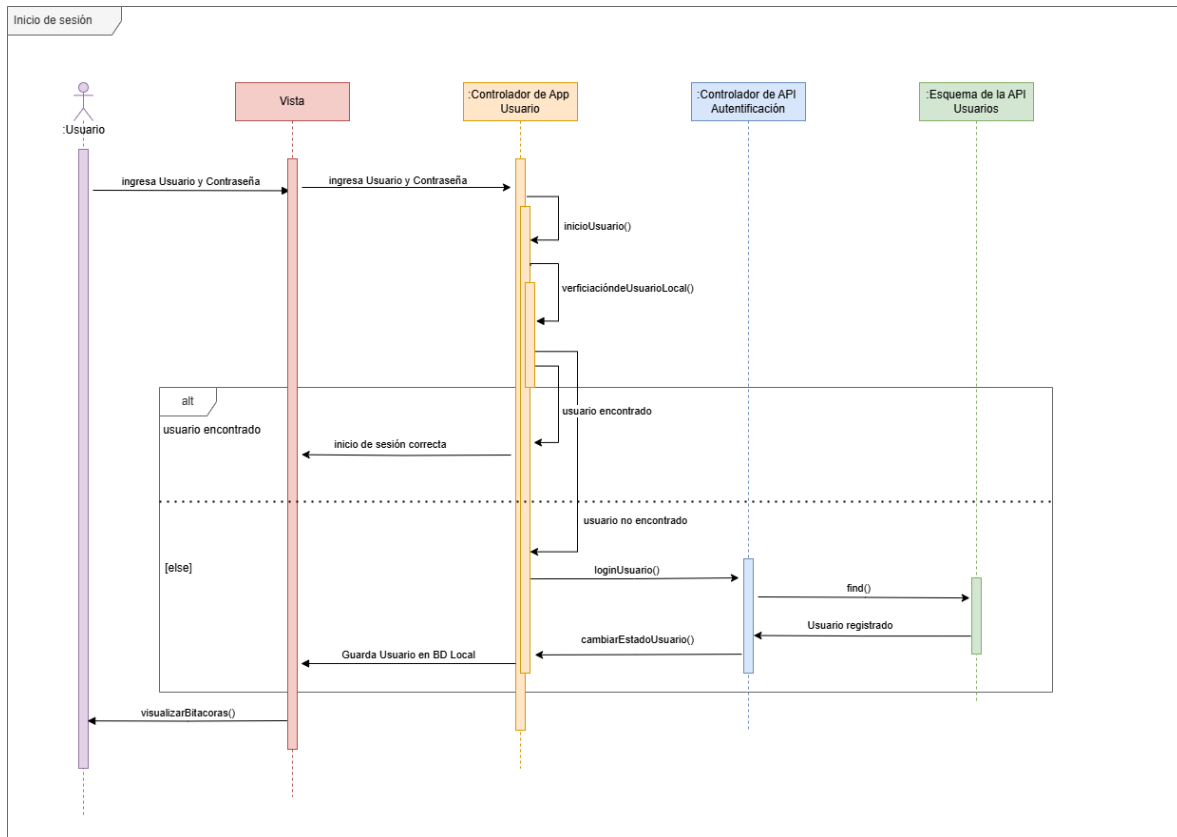




## Documento de diseño

- **Diagrama de secuencia:**

### DS-01 Inicio de Sesión



El diagrama de secuencia presentado describe el proceso de inicio de sesión (DS01) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación, el controlador de la aplicación, el controlador de la API de autenticación y el esquema de la API de usuarios durante el proceso de inicio de sesión.

### Descripción del Diagrama de Secuencia

1. Ingreso de Credenciales por el Usuario.
  - El usuario ingresa su correo y contraseña en la vista de la aplicación.
  - La vista de la aplicación envía las credenciales al controlador de la aplicación (Usuario).
2. Verificación Local del Usuario:
  - El controlador de la aplicación llama al método inicioUsuario() para iniciar el proceso de verificación.



## Documento de diseño

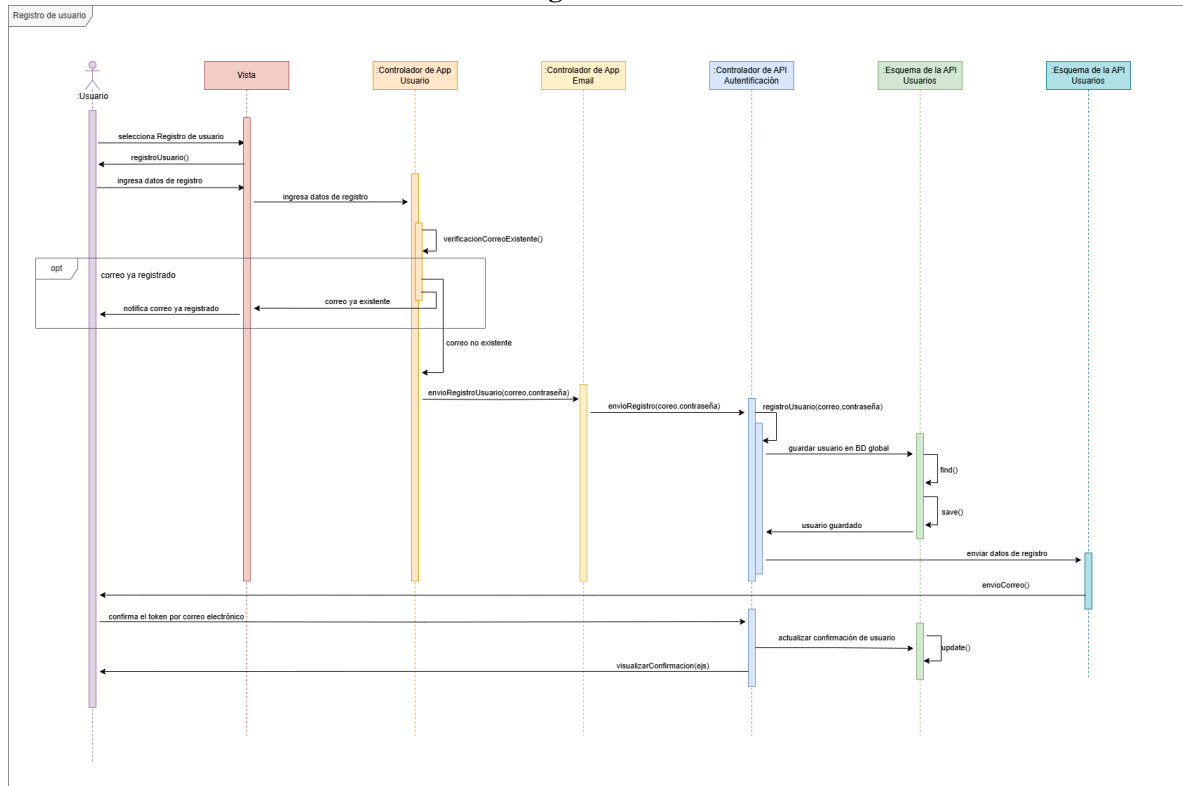
---

- Luego, el controlador de la aplicación utiliza el método `verificacionUsuarioLocal()` para verificar si el usuario está registrado localmente.
3. Confirmación de Usuario Encontrado Localmente:
    - Si el usuario es encontrado localmente, se confirma la verificación.
    - El controlador de la aplicación responde con un mensaje de "inicio de sesión correcta" y llama al método `visualizarBitacoras()` para mostrar las bitácoras del usuario.
    - La vista de la aplicación muestra la pantalla principal.
  4. Verificación Externa del Usuario:
    - Si el usuario no es encontrado localmente, el controlador de la aplicación llama al método `loginUsuario()` en el controlador de la API de autenticación.
    - El controlador de la API de autenticación utiliza el método `find()` del esquema de la API de usuarios para buscar el usuario en la base de datos global.
  5. Confirmación de Usuario Encontrado Externamente:
    - Si el usuario es encontrado en la base de datos global, se confirma la verificación.
    - El controlador de la API de autenticación responde con el estado del usuario y llama al método `cambiarEstadoUsuario()` para actualizar el estado del usuario.
    - El controlador de la aplicación responde con un mensaje de "inicio de sesión correcta" y llama al método `visualizarBitacoras()` para mostrar las bitácoras del usuario.
    - La vista de la aplicación muestra la pantalla principal.



## Documento de diseño

### DS-02 Registro de usuario



El diagrama de secuencia presentado describe el proceso de registro de usuario (DS-02) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación, el controlador de la aplicación, el controlador de la API de autenticación, el esquema de la API de usuarios y el controlador de la API de correos durante el proceso de registro de un nuevo usuario.

#### Descripción del Diagrama de Secuencia

- Selección de Registro de Usuario:
  - El usuario selecciona la opción de "Registro de usuario" en la vista de la aplicación.
  - La vista de la aplicación muestra la pantalla de registro.
- Ingreso de Datos de Registro:
  - El usuario ingresa los datos de registro, como nombre, correo y contraseña.
  - La vista de la aplicación envía estos datos al controlador de la aplicación (Usuario).
- Verificación de Correo Existente:
  - El controlador de la aplicación verifica si el correo ya está registrado usando el método `verificacionCorreoExistente()`.



## Documento de diseño

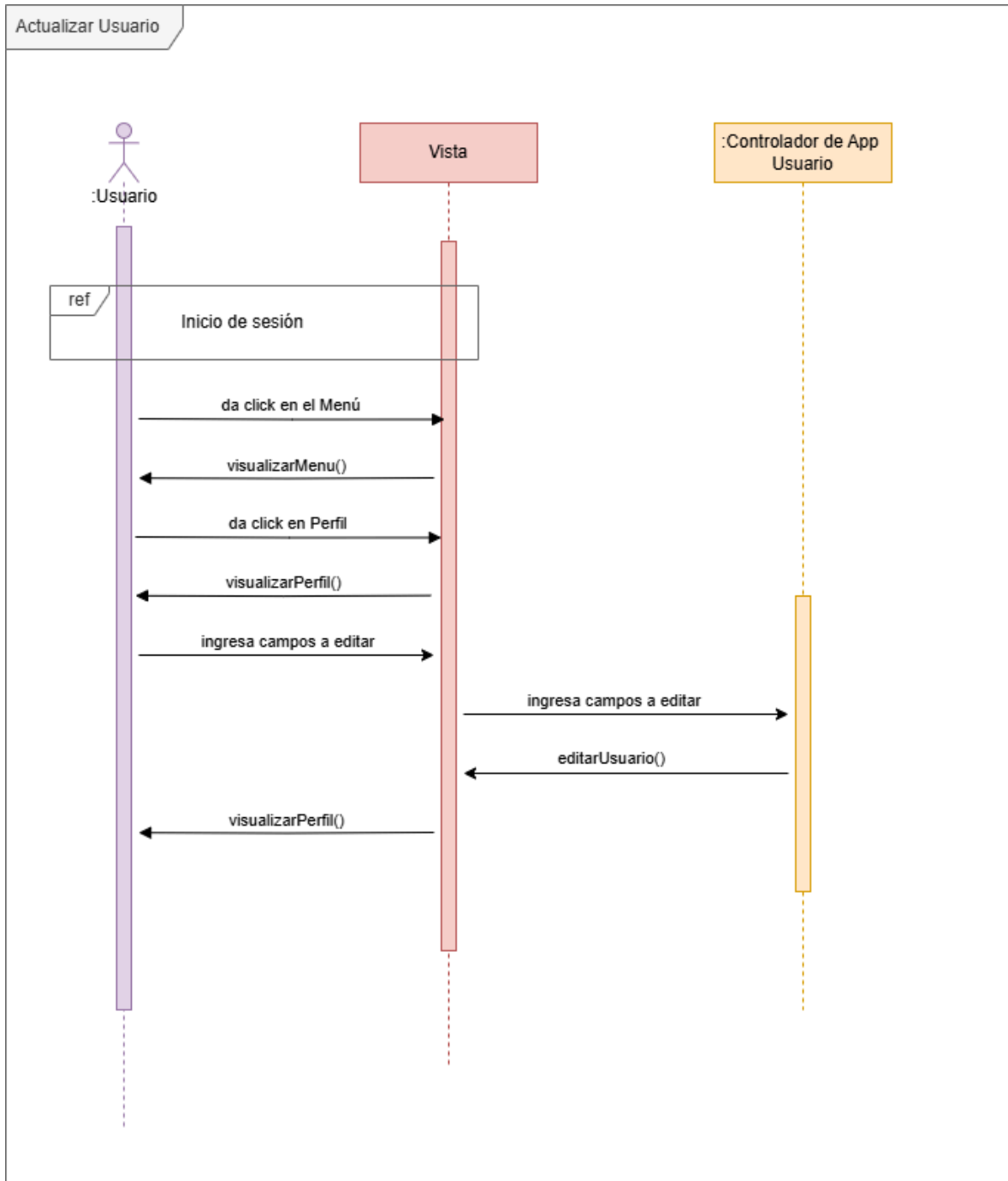
---

- Si el correo ya está registrado, la aplicación notifica al usuario y el proceso termina aquí.
  - Si el correo no está registrado, el controlador de la aplicación envía los datos de registro al controlador de la aplicación de correo (Email).
4. Envío de Registro a la API de Autenticación:
- El controlador de la aplicación de correo envía los datos de registro (correo y contraseña) a la API de autenticación.
  - La API de autenticación llama al método `registroUsuario(correo, contraseña)` para registrar al usuario en la base de datos global.
5. Guardar Usuario en la Base de Datos Global:
- La API de autenticación utiliza el esquema de la API de usuarios para guardar el nuevo usuario en la base de datos global.
  - El esquema de la API de usuarios llama al método `save()` para almacenar los datos del usuario.
  - Una vez guardado el usuario, la API de autenticación envía los datos de registro al controlador de la API de correos.
6. Envío de Correo de Confirmación:
- El controlador de la API de correos envía un correo de confirmación al nuevo usuario utilizando el método `envioCorreo()`.
  - El usuario recibe un correo con un token de confirmación y confirma su registro.
7. Confirmación de Registro:
- El usuario confirma el token recibido por correo electrónico.
  - La aplicación visualiza la página de confirmación de correo.
  - La API de autenticación actualiza la confirmación del usuario en la base de datos global utilizando el método `update()`.



## Documento de diseño

### DS-03 Actualizar Usuario





## Documento de diseño

---

El diagrama de secuencia presentado describe el proceso de actualización de usuario (DS-03) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de actualización de la información del perfil del usuario.

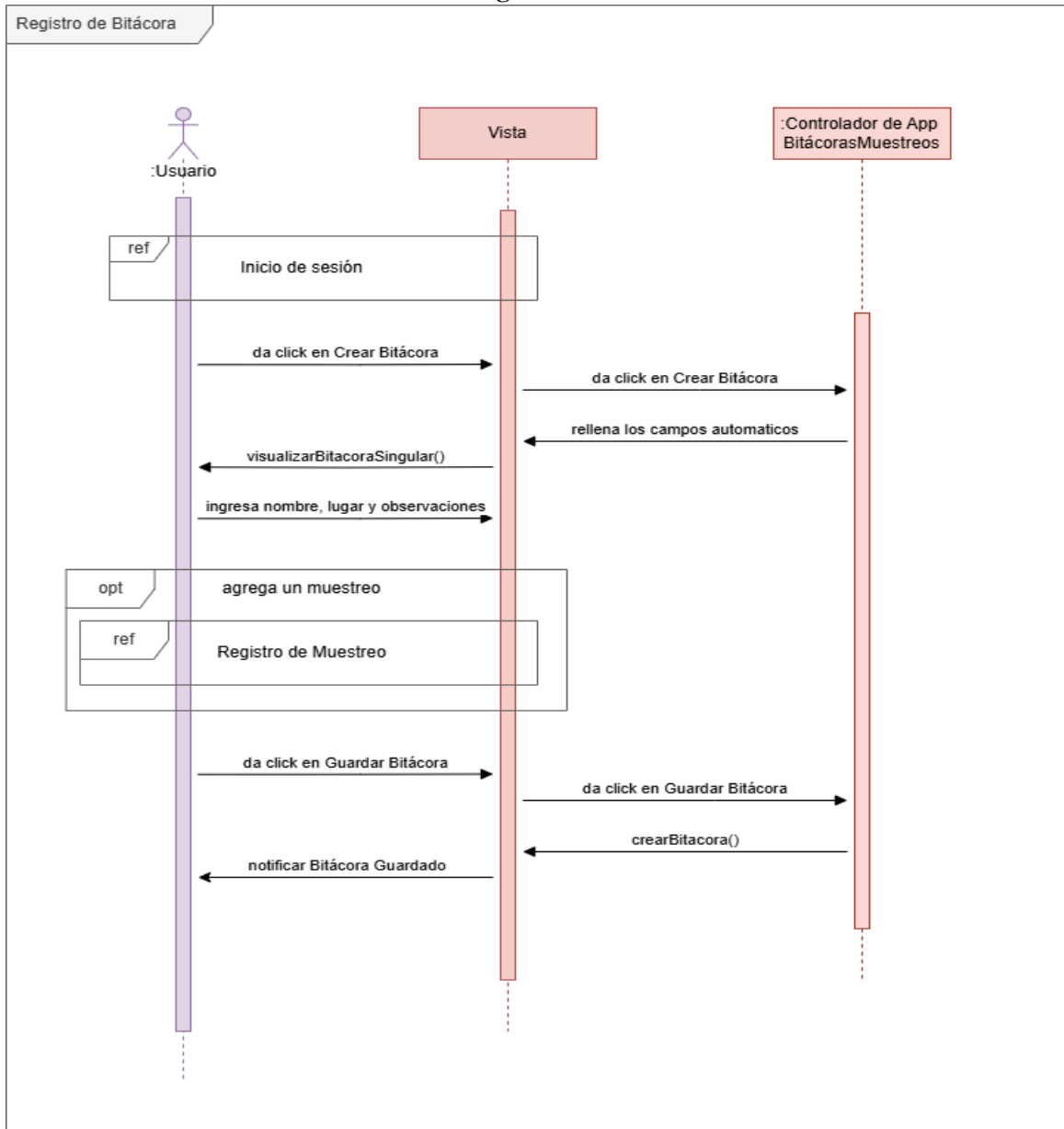
### Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
  - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Navegación al Menú de Perfil:
  - El usuario da clic en el menú de la aplicación.
  - La vista de la aplicación muestra el menú.
  - El usuario da clic en la opción de perfil dentro del menú.
  - La vista de la aplicación muestra la pantalla del perfil del usuario.
3. Acceso a la Pantalla de Editar Perfil:
  - El usuario selecciona la opción para editar el perfil.
  - La vista de la aplicación muestra la pantalla de editar perfil.
4. Ingreso de Datos para Editar:
  - El usuario ingresa los campos que desea editar en su perfil.
  - La vista de la aplicación envía estos datos al controlador de la aplicación (Usuario).
5. Actualización del Perfil del Usuario:
  - El controlador de la aplicación recibe los datos y llama al método `editarUsuario()` para actualizar la información del usuario en la base de datos.
  - Una vez actualizada la información, la vista de la aplicación muestra nuevamente la pantalla de editar perfil con los datos actualizados.



## Documento de diseño

### DS-04 Registro de Bitácora



El diagrama de secuencia presentado describe el proceso de registro de una bitácora (DS-04) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de creación de una nueva bitácora.

#### Descripción del Diagrama de Secuencia



## Documento de diseño

---

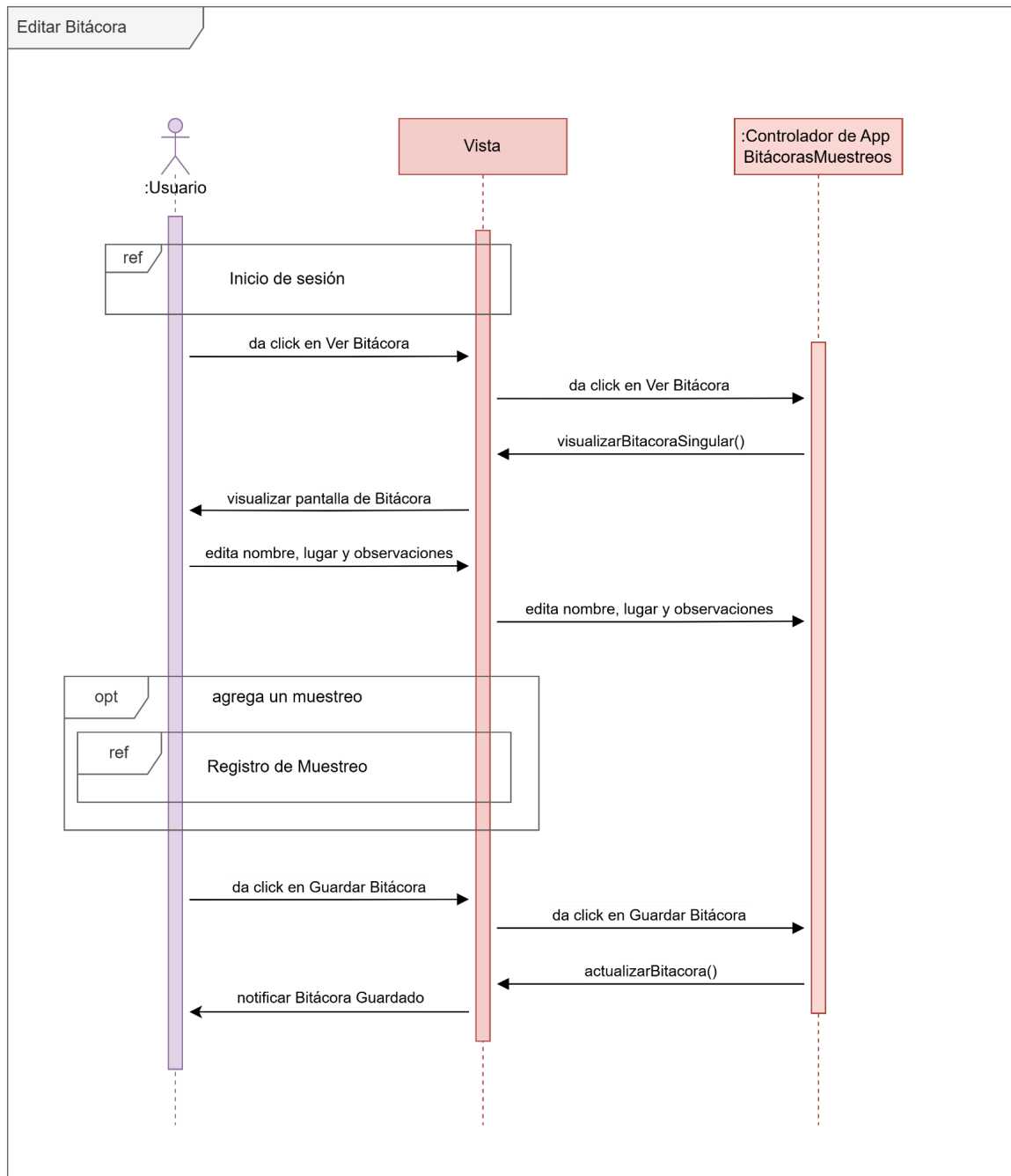
1. Inicio de Sesión:
  - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Navegación a Crear Bitácora:
  - El usuario da clic en la opción "Crear Bitácora".
  - La vista de la aplicación recibe la acción y solicita al controlador de la aplicación (BitácorasMuestreos) que inicialice el proceso de creación de una nueva bitácora.
  - La vista muestra la pantalla de creación de bitácora, con campos para ingresar ubicación, fecha y cantidad.
3. Ingreso de Datos de la Bitácora:
  - El usuario ingresa el nombre, lugar y observaciones para la bitácora.
  - La vista de la aplicación envía estos datos al controlador de la aplicación.
4. Creación de la Bitácora:
  - El controlador de la aplicación recibe los datos y llama al método crearBitacora() para guardar la nueva bitácora en la base de datos.
  - Una vez creada la bitácora, el controlador de la aplicación notifica a la vista que la bitácora ha sido guardada exitosamente.
5. Opción de Agregar Muestreo:
  - El usuario tiene la opción de agregar un muestreo a la bitácora recién creada.
  - Si el usuario elige agregar un muestreo, se sigue el proceso de registro de muestreo (referenciado como "Registro de Muestreo" en el diagrama).





## Documento de diseño

### DS-05 Editar Bitácora





## Documento de diseño

---

El diagrama de secuencia presentado describe el proceso de edición de una bitácora (DS-05) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de modificación de una bitácora existente.

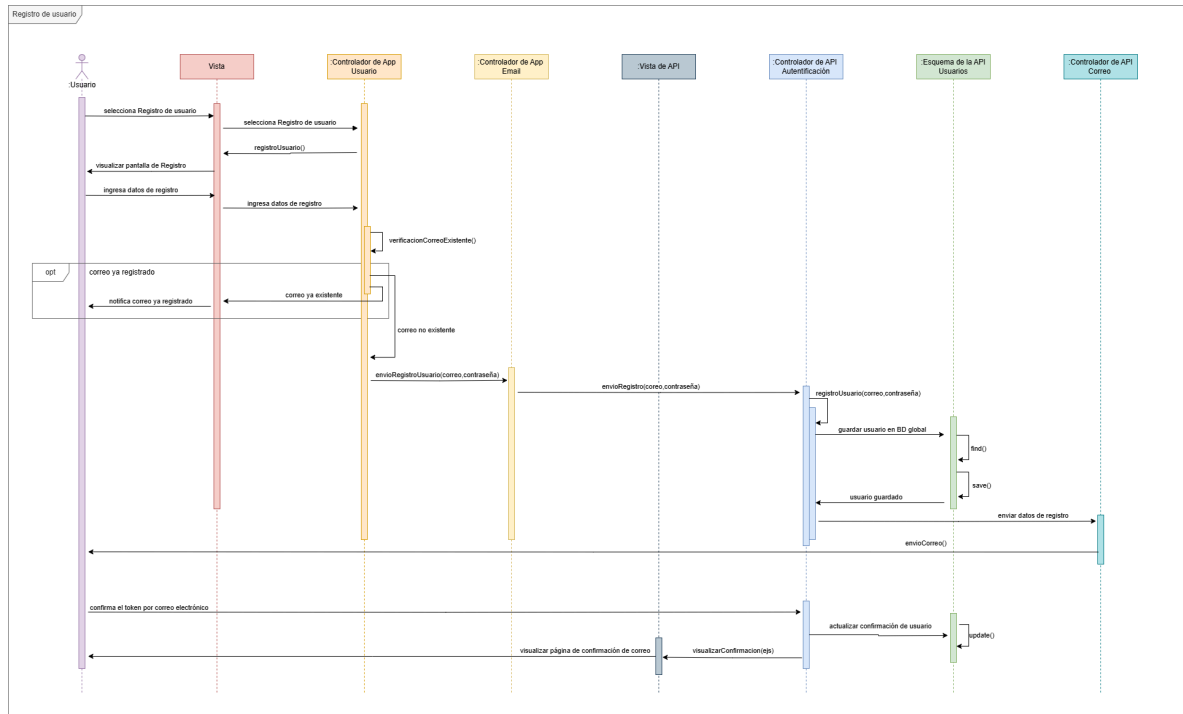
### Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
  - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Navegación a Ver Bitácora:
  - El usuario da clic en la opción "Ver Bitácora".
  - La vista de la aplicación recibe la acción y solicita al controlador de la aplicación (BitácorasMuestreos) que muestre la bitácora seleccionada.
  - La vista muestra la pantalla de la bitácora seleccionada.
3. Edición de Datos de la Bitácora:
  - El usuario edita el nombre, lugar y observaciones de la bitácora.
  - La vista de la aplicación envía estos datos editados al controlador de la aplicación.
4. Opción de Agregar Muestreo:
  - El usuario tiene la opción de agregar un muestreo a la bitácora.
  - Si el usuario elige agregar un muestreo, se sigue el proceso de registro de muestreo (referenciado como "Registro de Muestreo" en el diagrama).
5. Guardado de la Bitácora:
  - El usuario da clic en el botón "Guardar Bitácora" para confirmar los cambios.
  - La vista de la aplicación envía la solicitud de guardado al controlador de la aplicación.
  - El controlador de la aplicación recibe los datos editados y llama al método crearBitacora() para guardar los cambios en la base de datos.
  - Una vez guardada la bitácora, el controlador de la aplicación notifica a la vista que la bitácora ha sido guardada exitosamente.



## Documento de diseño

### DS-06 Eliminar Bitácora



El diagrama de secuencia presentado describe el proceso de eliminación de una bitácora (DS-06) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de eliminación de una bitácora existente.

#### Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
  - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Navegación a Eliminar Bitácora:
  - El usuario da clic en la opción "Eliminar Bitácora".
  - La vista de la aplicación recibe la acción y notifica al usuario con una confirmación de eliminación.
3. Confirmación de Eliminación:
  - El usuario tiene dos opciones: cancelar la eliminación o confirmar la eliminación.
4. Cancelación de Eliminación:
  - Si el usuario decide cancelar la eliminación, da clic en el botón "Cancelar".



## Documento de diseño

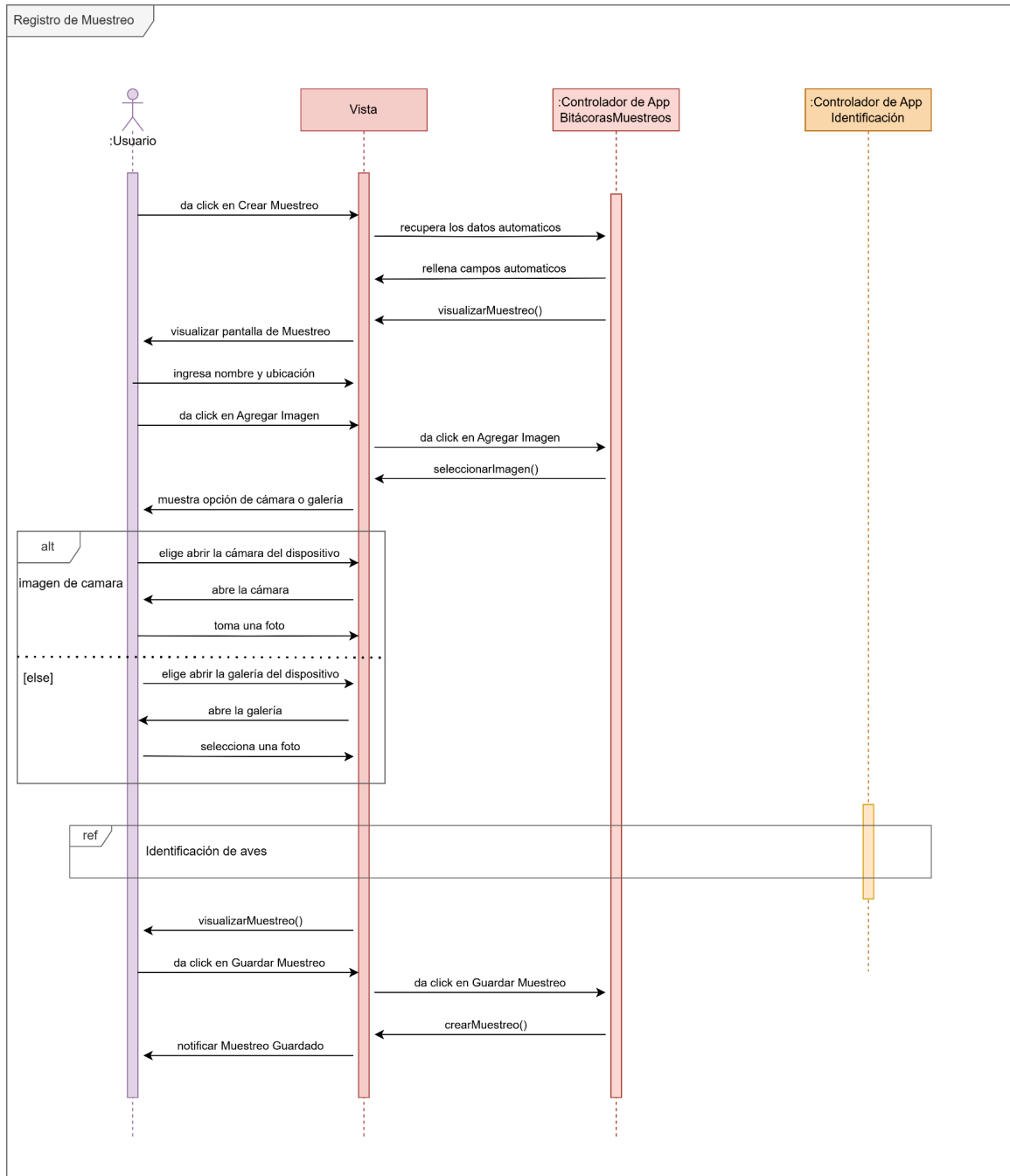
---

- La vista de la aplicación cancela el proceso y muestra nuevamente las bitácoras disponibles utilizando el método `visualizarBitacoras()`.
5. Confirmación de Eliminación:
- Si el usuario decide confirmar la eliminación, da clic en el botón "Confirmar".
  - La vista de la aplicación envía esta acción al controlador de la aplicación (`BitacorasMuestreos`).
  - El controlador de la aplicación llama al método `eliminarBitacora()` para eliminar la bitácora seleccionada de la base de datos.
  - Una vez eliminada la bitácora, la vista de la aplicación muestra nuevamente las bitácoras disponibles.



## Documento de diseño

### DS-07 Registro de Muestreo





## Documento de diseño

---

El diagrama de secuencia presentado describe el proceso de registro de un muestreo (DS-07) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y los controladores de la aplicación durante el proceso de creación de un nuevo muestreo y la identificación de aves.

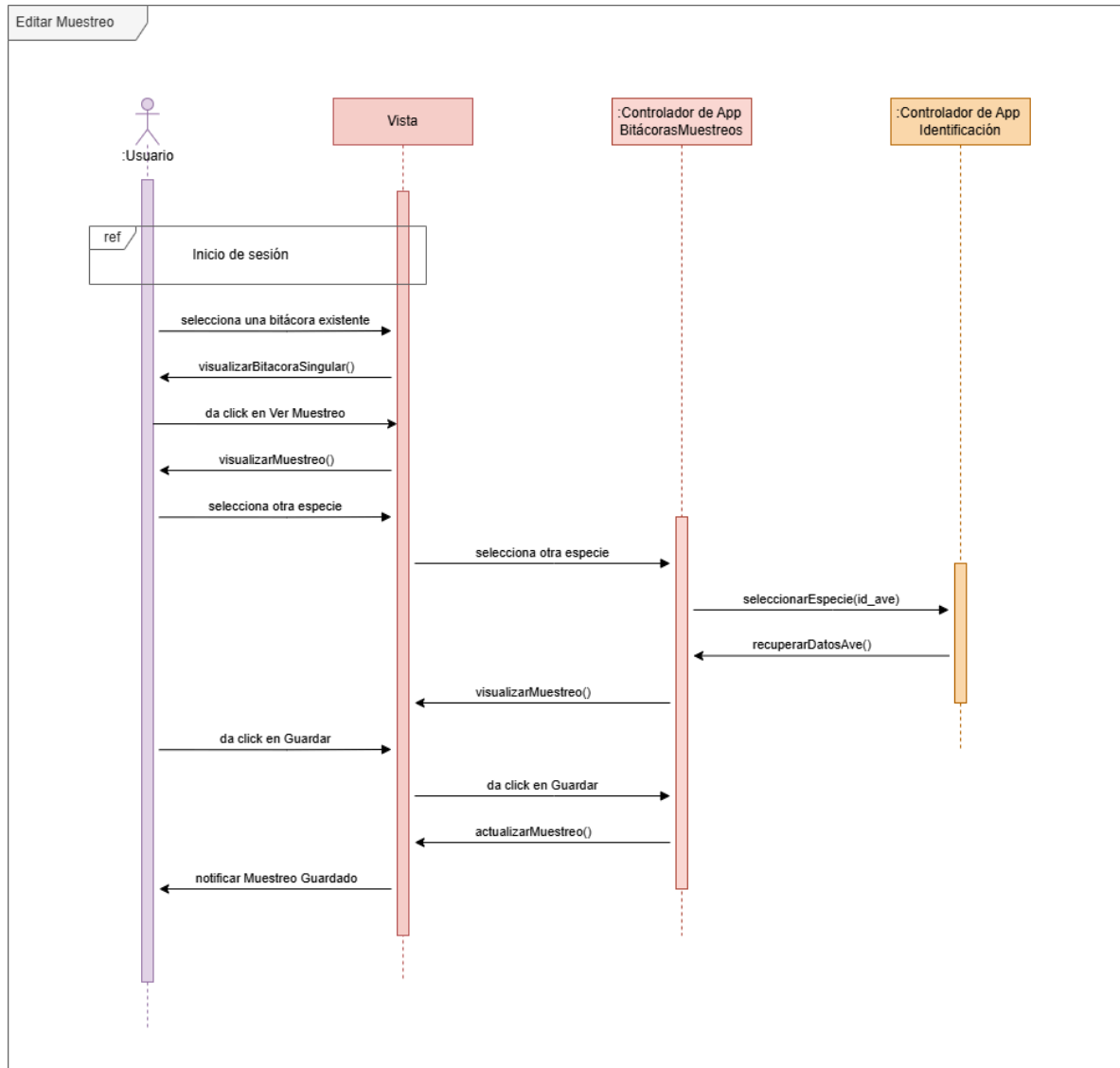
### Descripción del Diagrama de Secuencia

1. Inicio del Proceso:
  - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación y está en la vista de una bitácora específica.
2. Creación de Muestreo:
  - El usuario da clic en la opción "Crear Muestreo".
  - La vista de la aplicación recibe la acción y solicita al controlador de la aplicación (BitácorasMuestreos) que recupere los datos automáticos necesarios para el muestreo.
  - La vista de la aplicación rellena los campos automáticos y muestra la pantalla de muestreo.
3. Agregar Imagen:
  - El usuario da clic en "Agregar Imagen" para añadir una foto del ave.
  - La vista de la aplicación solicita al controlador de la aplicación (BitácorasMuestreos) que inicie el proceso de selección de imagen.
  - El controlador de la aplicación gestiona la selección de la imagen y la vista muestra la pantalla de agregar imagen.
4. Identificación de Aves (Referencia):
  - El proceso de identificación de aves se inicia como una referencia dentro del registro de muestreo. Esto incluye el flujo completo de identificación de aves, desde la selección de imagen hasta la obtención de datos de la especie identificada.
5. Finalización del Muestreo:
  - Una vez que el usuario ha agregado la imagen y se ha realizado la identificación de aves, la vista de la aplicación muestra nuevamente la pantalla de muestreo con los datos actualizados.
  - El usuario puede regresar a la vista de la bitácora singular, completando el proceso de registro de muestreo.



## Documento de diseño

### DS-08 Editar Muestreo



El diagrama de secuencia presentado describe el proceso de edición de un muestreo (DS-08) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y los controladores de la aplicación durante el proceso de modificación de un muestreo existente.

#### Descripción del Diagrama de Secuencia

##### 1. Inicio de Sesión:

- El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.



## Documento de diseño

---

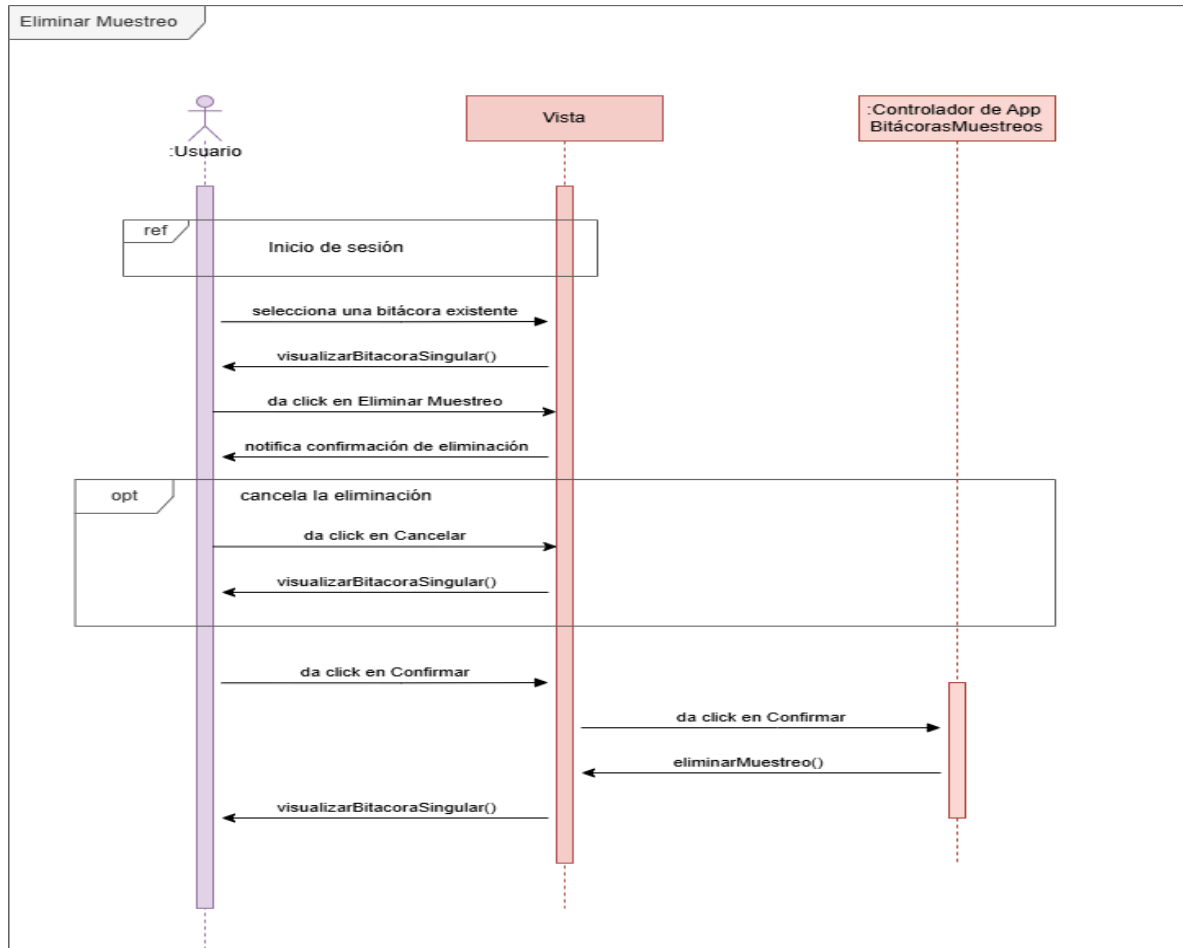
2. Selección de Bitácora y Muestreo:
  - El usuario selecciona una bitácora existente desde la vista principal.
  - La vista de la aplicación muestra la bitácora seleccionada utilizando el método `visualizarBitacoraSingular()`.
  - El usuario da clic en "Ver Muestreo" para seleccionar el muestreo que desea editar.
  - La vista de la aplicación muestra el muestreo utilizando el método `visualizarMuestreo()`.
3. Edición de la Especie del Muestreo:
  - El usuario selecciona otra especie para el muestreo.
  - La vista de la aplicación envía esta acción al controlador de la aplicación (`BitácorasMuestreos`).
  - El controlador de la aplicación llama al método `seleccionarEspecie(id_ave)` del controlador de la aplicación de identificación.
  - El controlador de identificación recupera los datos de la nueva especie seleccionada utilizando el método `recuperaDatosAve()`.
4. Actualización del Muestreo:
  - La vista de la aplicación muestra nuevamente el muestreo con la nueva especie seleccionada utilizando el método `visualizarMuestreo()`.
  - El usuario da clic en "Guardar" para confirmar los cambios
  - La vista de la aplicación envía esta acción al controlador de la aplicación.
  - El controlador de la aplicación llama al método `actualizarMuestreo()` para guardar los cambios en la base de datos.
  - Una vez guardado el muestreo, el controlador de la aplicación notifica a la vista que el muestreo ha sido guardado exitosamente.





## Documento de diseño

### DS-09 Eliminar Muestreo



El diagrama de secuencia presentado describe el proceso de eliminación de un muestreo (DS-09) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de eliminación de un muestreo existente.

#### Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
  - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Selección de Bitácora y Muestreo:
  - El usuario selecciona una bitácora existente desde la vista principal.
  - La vista de la aplicación muestra la bitácora seleccionada utilizando el método `visualizarBitacoraSingular()`.



## Documento de diseño

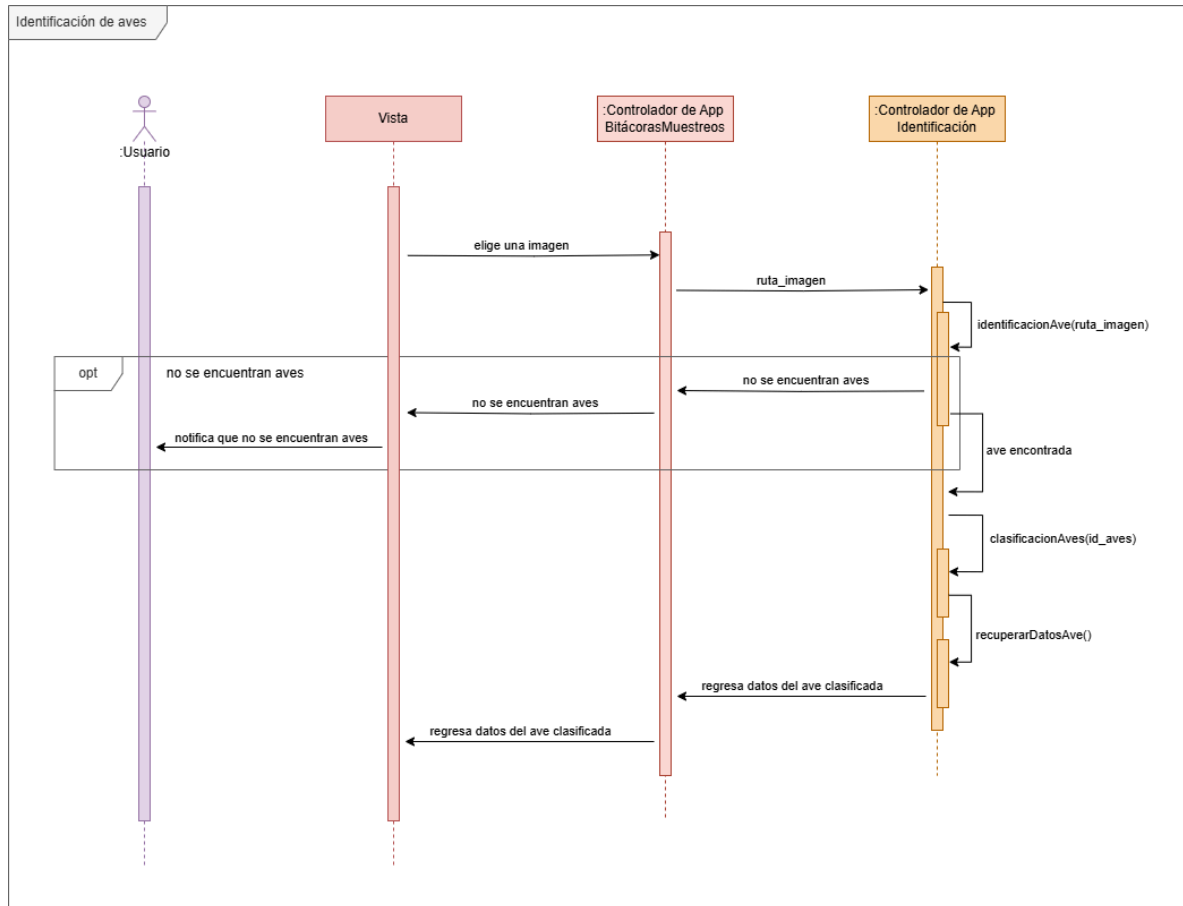
---

- El usuario da clic en "Eliminar Muestreo" para seleccionar el muestreo que desea eliminar.
  - La vista de la aplicación notifica al usuario con una confirmación de eliminación.
3. Confirmación o Cancelación de Eliminación:
    - El usuario tiene dos opciones: cancelar la eliminación o confirmar la eliminación.
  4. Cancelación de Eliminación:
    - Si el usuario decide cancelar la eliminación, da clic en el botón "Cancelar".
    - La vista de la aplicación cancela el proceso y muestra nuevamente la bitácora utilizando el método `visualizarBitacoraSingular()`.
  5. Confirmación de Eliminación:
    - Si el usuario decide confirmar la eliminación, da clic en el botón "Confirmar".
    - La vista de la aplicación envía esta acción al controlador de la aplicación (`BitácorasMuestreos`).
    - El controlador de la aplicación llama al método `eliminarMuestreo()` para eliminar el muestreo seleccionado de la base de datos.
    - Una vez eliminado el muestreo, la vista de la aplicación muestra nuevamente la bitácora utilizando el método `visualizarBitacoraSingular()`.



## Documento de diseño

### DS-10 Identificación de Aves



El diagrama de secuencia presentado describe el proceso de identificación de aves (DS-10) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el usuario, la vista de la aplicación y los controladores de la aplicación durante el proceso de identificación de un ave a partir de una imagen proporcionada por el usuario.

#### Descripción del Diagrama de Secuencia

- Selección de Imagen:
  - El usuario elige una imagen desde la vista de la aplicación.
  - La vista de la aplicación recibe la imagen seleccionada y envía la ruta de la imagen al controlador de la aplicación (BitácorasMuestreos).
- Proceso de Identificación de Aves:
  - El controlador de la aplicación (BitácorasMuestreos) envía la ruta de la imagen al controlador de la aplicación de identificación.



## Documento de diseño

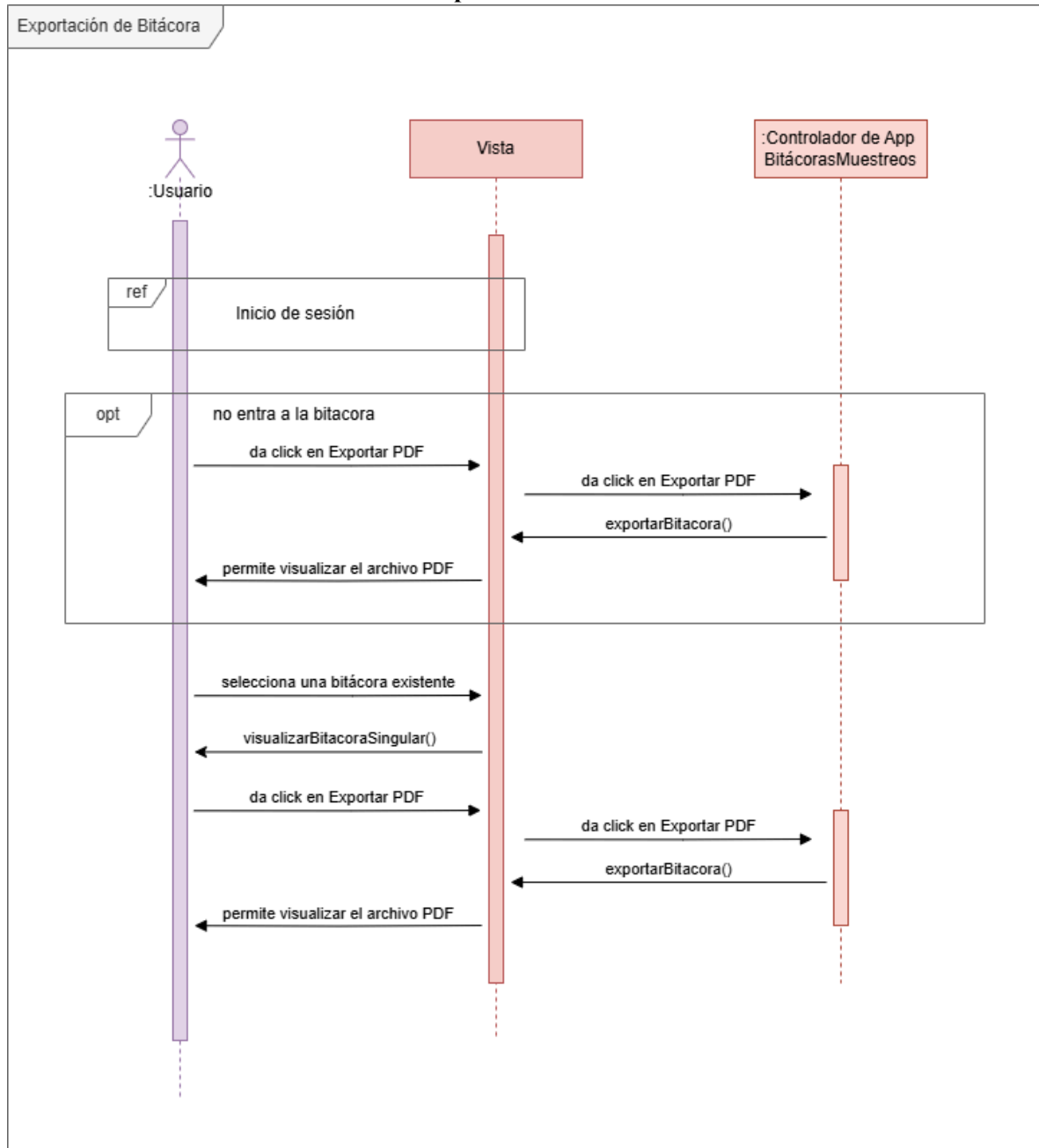
---

- El controlador de la aplicación de identificación utiliza el método `identificacionAve(ruta_imagen)` para procesar la imagen y buscar aves en ella.
3. Resultado de la Identificación:
    - Si no se encuentran aves en la imagen, el controlador de la aplicación de identificación notifica al controlador de la aplicación (`BitácorasMuestreos`) que no se encontraron aves.
    - El controlador de la aplicación (`BitácorasMuestreos`) notifica a la vista que no se encontraron aves, y la vista informa al usuario.
  4. Clasificación y Recuperación de Datos del Ave:
    - Si se encuentran aves en la imagen, el controlador de la aplicación de identificación utiliza el método `clasificacionAves(id_aves)` para clasificar las aves encontradas.
    - Luego, el controlador de la aplicación de identificación utiliza el método `recuperarDatosAve()` para obtener los datos de las aves clasificadas.
  5. Retorno de Datos:
    - Los datos del ave clasificada se regresan al controlador de la aplicación (`BitácorasMuestreos`).
    - El controlador de la aplicación (`BitácorasMuestreos`) envía los datos a la vista de la aplicación.
    - La vista de la aplicación muestra los datos del ave clasificada al usuario.



## Documento de diseño

### DS-11 Exportación de Bitácora



El diagrama de secuencia presentado describe el proceso de exportación de una bitácora (DS-11) en la aplicación móvil IdBird. Este diagrama muestra la interacción entre el



## Documento de diseño

---

usuario, la vista de la aplicación y el controlador de la aplicación durante el proceso de generación y descarga de un archivo PDF de una bitácora.

### Descripción del Diagrama de Secuencia

1. Inicio de Sesión:
  - El diagrama asume que el usuario ya ha iniciado sesión en la aplicación.
2. Opción de Exportar PDF sin Entrar a la Bitácora:
  - El usuario puede optar por exportar una bitácora directamente sin entrar a verla.
  - El usuario da clic en "Exportar PDF" desde la vista principal.
  - La vista de la aplicación envía esta solicitud al controlador de la aplicación (BitácorasMuestreos).
  - El controlador de la aplicación llama al método exportarBitacora() para generar el archivo PDF de la bitácora.
  - Una vez generado el archivo PDF, la vista de la aplicación permite al usuario descargar el archivo.
3. Opción de Exportar PDF desde una Bitácora Existente:
  - Alternativamente, el usuario puede seleccionar una bitácora existente para verla antes de exportarla.
  - El usuario selecciona una bitácora existente desde la vista principal.
  - La vista de la aplicación muestra la bitácora seleccionada utilizando el método visualizarBitacoraSingular().
  - El usuario da clic en "Exportar PDF" desde la vista de la bitácora.
  - La vista de la aplicación envía esta solicitud al controlador de la aplicación (BitácorasMuestreos).
  - El controlador de la aplicación llama al método exportarBitacora() para generar el archivo PDF de la bitácora.
  - Una vez generado el archivo PDF, la vista de la aplicación permite al usuario descargar el archivo.



## Documento de diseño

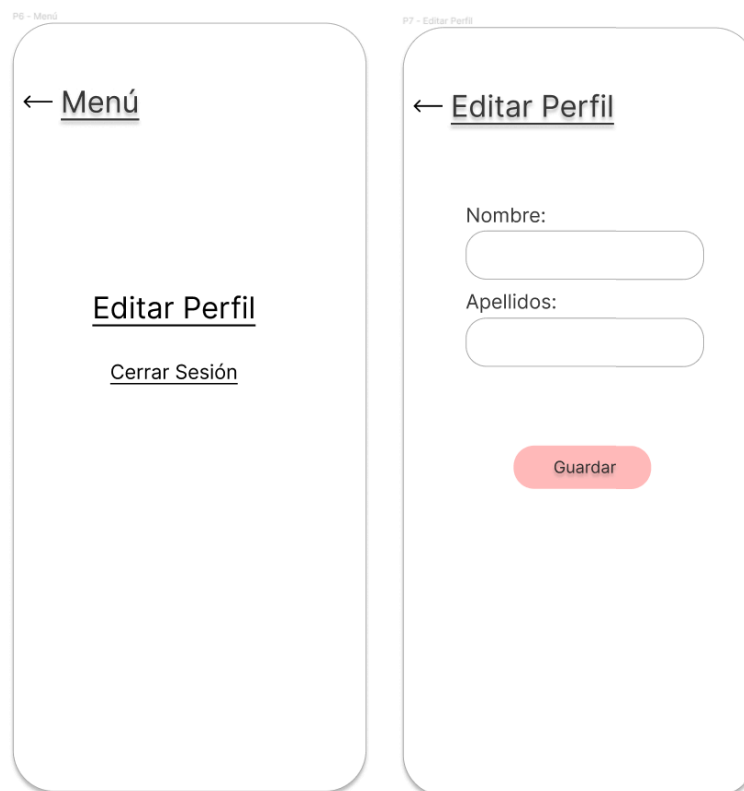
### Diseño de prototipos

#### Logo de aplicación



#### Menú:

Contiene las configuraciones necesarias para acceder a las funciones como Editar Perfil o Cerrar Sesión






## Documento de diseño

### Pantalla de Inicio de Sesión y Registro de Usuarios:

En estas secciones contiene los espacios requeridos para acceder a la aplicación, así mismo como el registro para poder acceder a las funcionalidades de la aplicación,

P1 - Inicio de sesión



## Inicio de Sesión


Correo:

Contraseña:

[Iniciar Sesión](#)

[Crear Cuenta](#)

P2 - Pantalla De Registro



## Registro de Usuarios

Nombre:

Apellidos:

E-mail:

Confirmar e-mail

Contraseña:

[Crear Cuenta](#)

[Cancelar](#)





## Documento de diseño

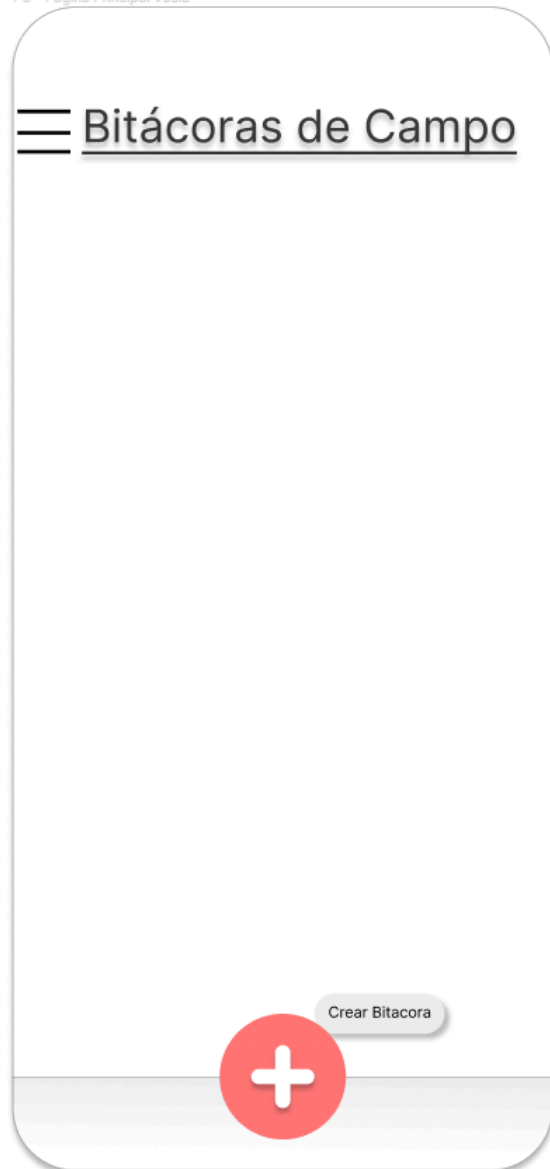
### Pantalla de Bitácoras de Campo:

Esta pantalla contiene la previsualización de la Bitácora realizada por el usuario. Así mismo contiene botones que le permiten al usuario crear, ver y exportar la Bitácora.

Página Principal



P3 - Página Principal Vacía





## Documento de diseño

### Pantalla de Bitácora:

En esta sección contiene los muestreos realizados por el usuario en las Bitácoras creadas, permitiéndole acceder a los muestreos realizados así como también crear un muestreo. Así mismo se le permite al usuario exportar en PDF su bitácora.

Bitacoras Completas

Bitacora

**Nombre:** Bitacora 1  
**Fecha:** 09 abril de 2024  
**Lugar:** UPIIZ  
**Ubicación:** 22.7840495,-102.617548,18.91z  
**Cantidad de Muestreos:** 7  
**Observaciones:** Pocas aves



**Nombre:** Cernícola  
**Fecha:** 09 de abril del 2024  
**Ubicación:** Zacatecas

Ver Muestreo



**Nombre:** Papamoscas Negro  
**Fecha:** 09 de abril del 2024  
**Ubicación:** Zacatecas

Ver Muestreo









P4 - Bitacoras Vacías


Bitacora

**Nombre:** \_\_\_\_\_  
**Fecha:** \_\_\_\_\_  
**Lugar:** \_\_\_\_\_  
**Ubicación:** \_\_\_\_\_  
**Cantidad de Muestreos:** \_\_\_\_\_  
**Observaciones:** \_\_\_\_\_

Crear Muestreo









## Documento de diseño

### Pantalla de Muestreo:

En esta pantalla el usuario podrá realizar o capturar una fotografía para realizar la identificación del ave y así mostrarle al usuario la información relacionada con la misma ave fotografiada.

Muestreo

← Muestreo



**Nombre del muestreo:** Falco Sparverius  
**Nombre coloquial del ave:** Cernícalo Americano  
**Fecha:** 09 de abril de 2024  
**Hora:** 14:00 hrs  
**Coordenadas:** 22.7840495,-102.617548,18.91z  
**Ubicación:** UPIIZ  
**Color:** Gris, Naranja, Blanco  
**Dimensiones:** 12 cm x 8 cm



P5 - Muestreo Vacío

← Muestreo



**Nombre del muestreo:** \_\_\_\_\_  
**Nombre coloquial del ave:** \_\_\_\_\_  
**Fecha:** \_\_\_\_\_  
**Hora:** \_\_\_\_\_  
**Coordenadas:** \_\_\_\_\_  
**Ubicación:** \_\_\_\_\_  
**Color:** \_\_\_\_\_  
**Dimensiones:** \_\_\_\_\_





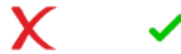
## Documento de diseño

### Alerts:

Estos elementos fueron realizados para mejorar la experiencia en el sistema del usuario y notificarle sobre algunos cambios o acciones realizadas en el sistema hechas por el mismo usuario.

PA1 - Alerta de Eliminar

¿Estas seguro de  
borrar esta bitacora?



PA2 - Alerta de Duplicidad de Cuenta

Cuenta existente  
Inicia sesión o regístrate con  
información diferente.

PA3 - Notificación Usuario Confirmación de Correo

¡Registro exitoso!  
Por favor, verifica tu correo  
electrónico para confirmar tu cuenta.

PA4 - Alerta Inicio de Sesión Incorrecto

El correo electrónico no existe o aún  
no se ha confirmado la dirección de  
correo electrónico para la verificación

PA5 - Alerta Registro Fallido

Registro fallido. El correo de  
confirmación no se pudo enviar.

PA6 - Guardar Bitacora

Bitacora Guardada

PA7 - Guardar Muestreo

Muestreo Guardado

# Documento de diseño

## Formato para la exportaciones la bitácora en PDF

## PE1 - Formato Reporte Bitacora

Formato para el registro de aves mediante el método de conteo por puntos.

Nombre de la bitácora: \_\_\_\_\_ Lugar: \_\_\_\_\_ Fecha: \_\_\_\_\_

Ubicación: \_\_\_\_\_ Cantidad de muestreos: \_\_\_\_\_

Observaciones: \_\_\_\_\_

[illegible]



Instituto Politécnico Nacional  
Unidad Profesional Interdisciplinaria de Ingeniería campus  
Zacatecas



## Documento de diseño

### Pantalla web Confirmacion de Usuario:

Una vez que el usuario confirme su correo se le guiará hasta esta página donde se le dará a conocer que ya está registrado en el sistema y ya está listo para que él lo use.

PE2 - Pantalla De Confirmar Correo Object



¡Usuario  
Confirmado!