Grupo 24:

Joaquín Alejandro Godoy Vergara.

Gabriela Isidora Zambrano Novoa.

Nuestro grupo eligió el desarrollo de un simulador interactivo de gestión de zoológicos virtuales. El programa permitirá a los usuarios personalizar hábitats y gestionar diversas especies de animales, atendiendo a las necesidades específicas de cada una, como alimentación, temperatura y relaciones grupales. Los usuarios dispondrán de hábitats y animales desde un Menu Bar (botones donde podrá cambiar al hábitat que desee y seleccionar a los animales que desee). El sistema impedirá la ubicación de animales en hábitats inadecuados.

La siguiente lista muestra los patrones utilizados durante el código.

1. Factory Method.

   El patrón Factory Method se utiliza principalmente en las clases que extienden de la clase base “Animal”. El constructor de Animal sirve como un método que podría considerarse un Factory Method indirectamente, ya que es responsable de crear instancias de diferentes tipos de animales según los parámetros que recibe. Por otro lado, las subclases concretas utilizan el constructor de la clase base “Animal” para crear instancias específicas de esos animales.

2. MVC (Modelo-Vista-Controlador)

   En el código el patrón se implementa de la siguiente manera:

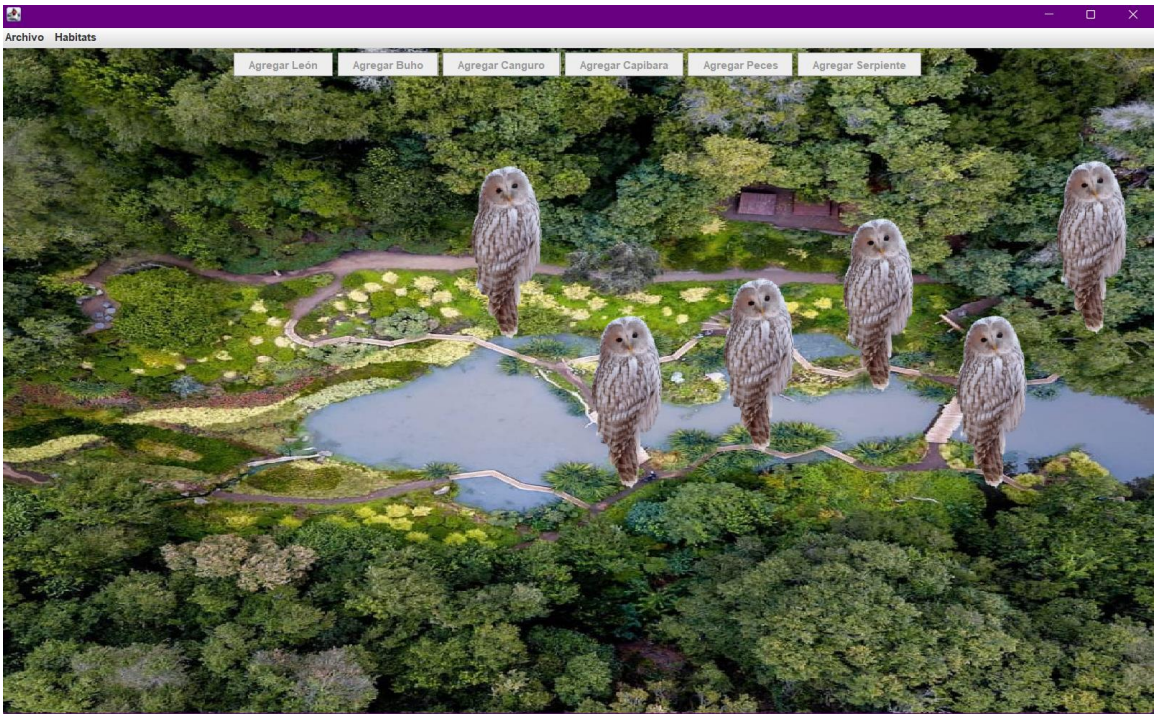
a) Modelo: El modelo se representa principalmente por la clase “Habitat1”. La clase maneja la lógica relacionada con el hábitat y la gestión de los animales. También hace uso de excepciones para manejar situaciones específicas relacionadas con el modelo.
b) Vista: Las clases que extienden a ‘JPanel’ representan la vista. Cada una de estas clases se encarga de mostrar un hábitat específico con sus respectivos elementos gráficos, se utilizan botones y se establecen acciones para manejar las interacciones del usuario en cada vista.
c) Controlador: Los ‘ActionListener’ asociados a los botones actúan como controladores. Por ejemplos, en ‘PanelAcuario’, cada botón tiene un ‘ActionListener’ que responde a la interación del usuario y realiza acciones como agregar imágenes al hábitat o mostrar mensajes de error en función de la lógica definida en el modelo.

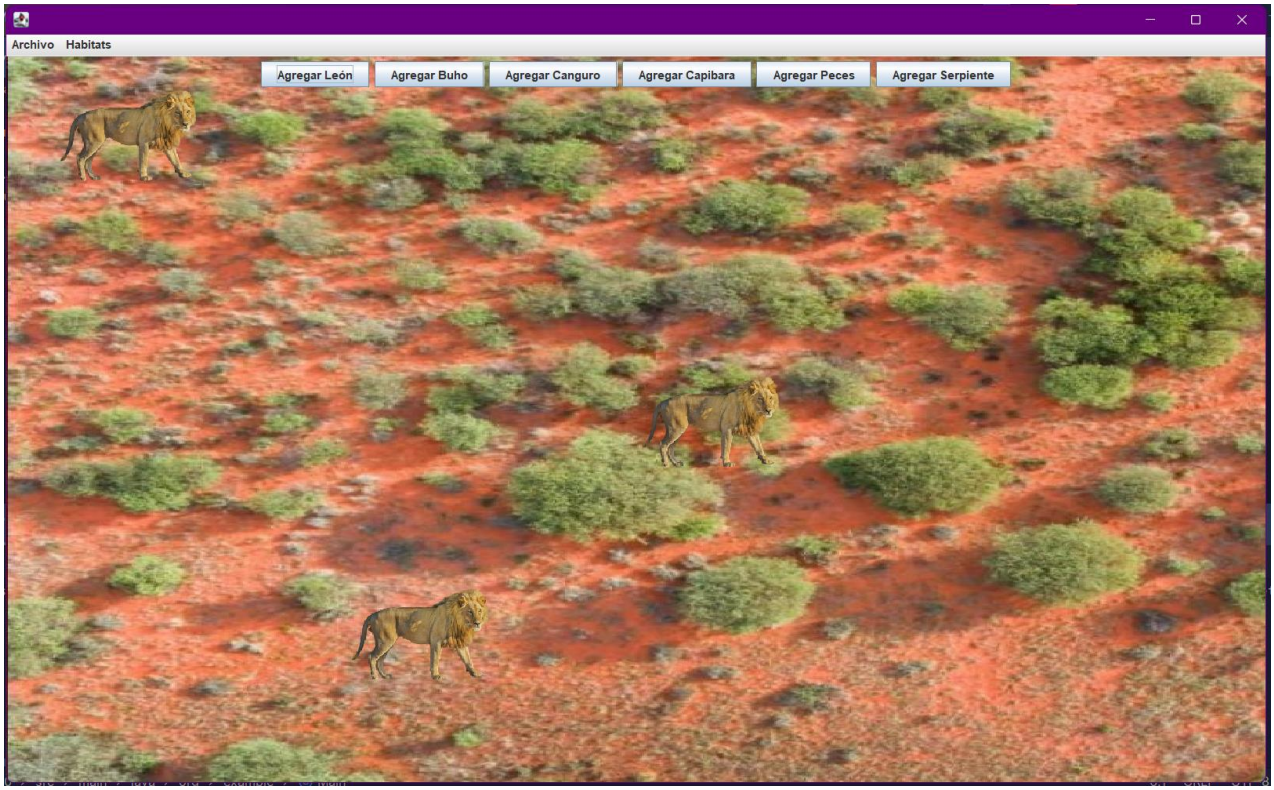
Durante el desarrollo del proyecto, tomamos decisiones importantes para mejorar la estructura y la gestión del código. Una de las decisiones clave fue migrar de tener múltiples JFrame a consolidar la interfaz en un solo JFrame, dividiendo su contenido en varios JPanles. Esta modificación nos permitió mejorar la organización del código y facilitar su mantenimiento, además de brindar una experiencia de usuario más cohesiva y eficiente.

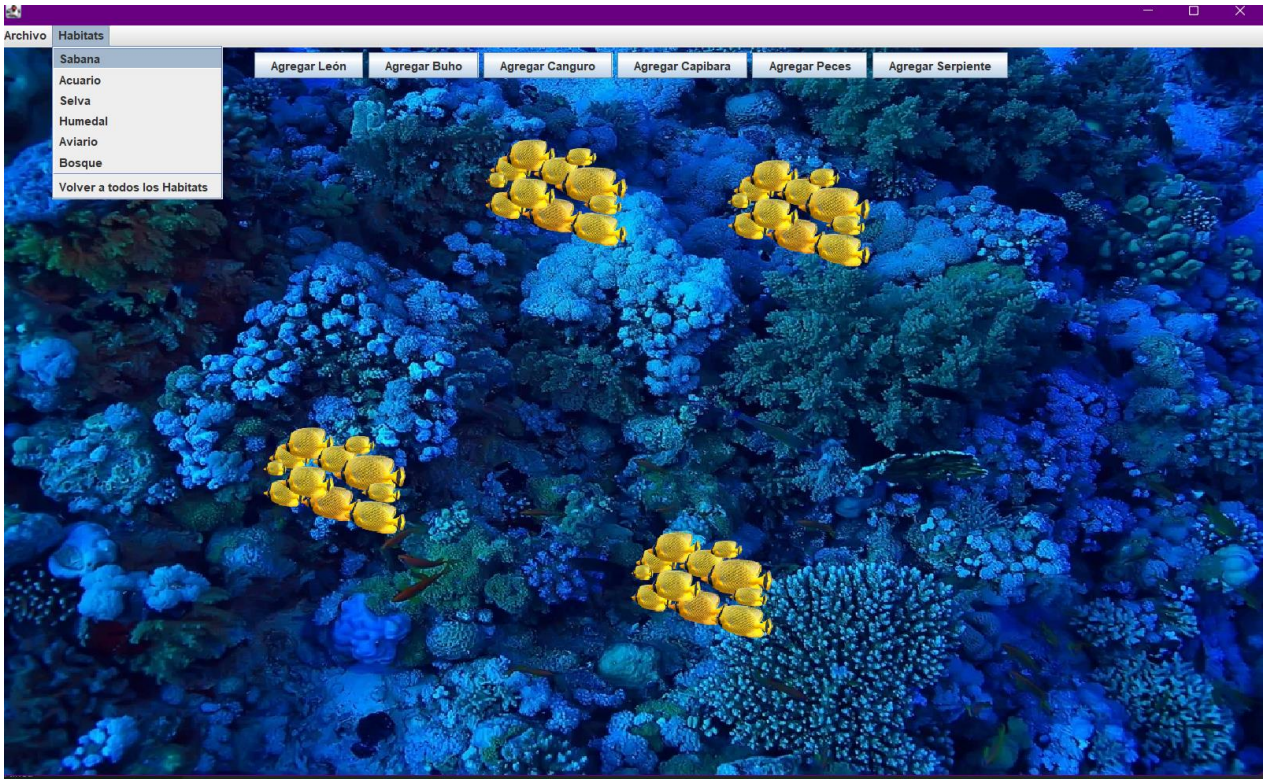
Otra decisión relevante que tomamos fue la creación de un nuevo repositorio en GitHUb para el proyecto. Optamos por esta acción para empezar desde cero, evitando dependencias o limitaciones del antiguo repositorio y garantizando un inicio limpio y estructurado para el desarrollo futuro. Tenemos en cuenta que esta decisión nos podría costar puntos en la rubrica del trabajo, pero independiente a eso, quedamos satisfechos con el resultado.

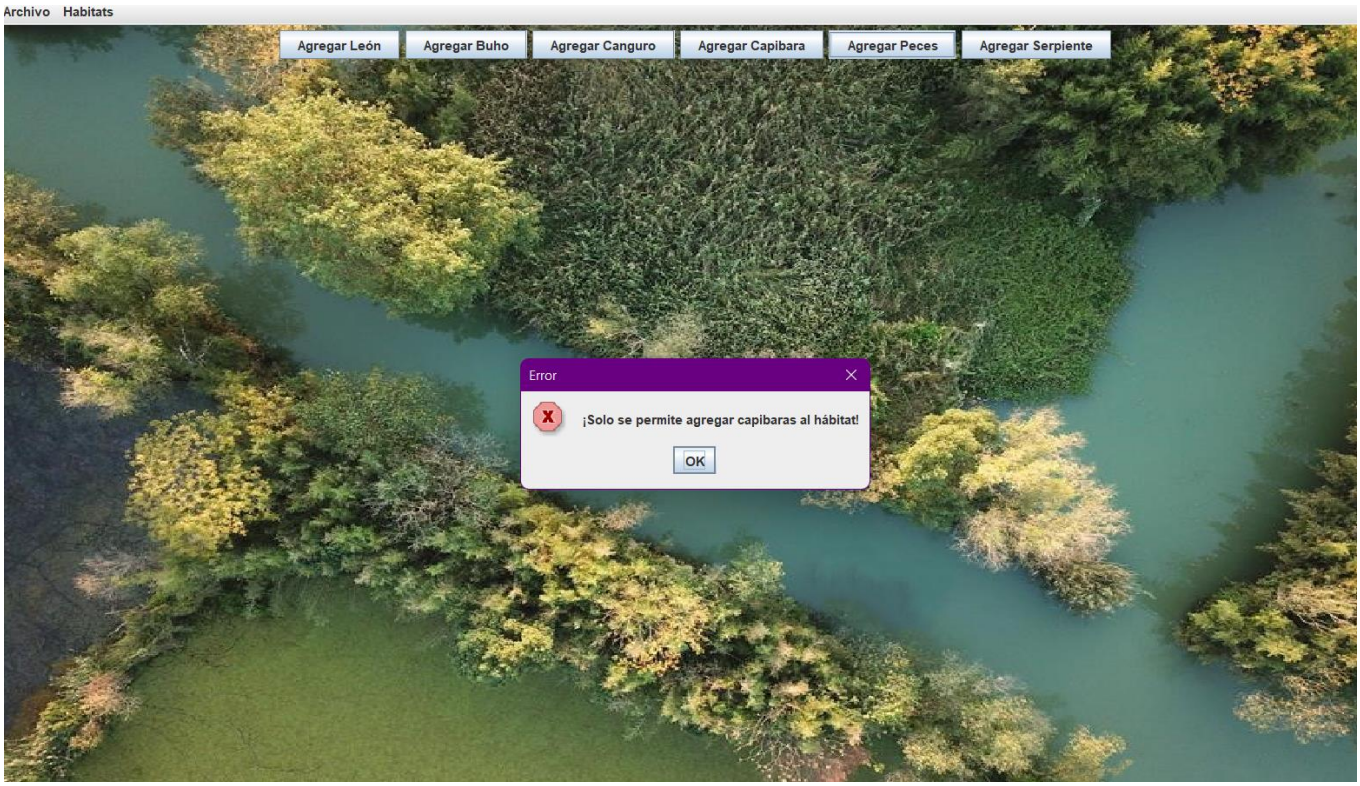
Durante la fase inicial del proyecto, nos enfrentamos a desafíos significativos al intentar integrar imágenes de animales en cada panel correspondiente a los hábitats. Lamentablemente, la complejidad técnica de esta tarea en el contexto del antiguo repositorio dificultó en gran medida su implementación exitosa. Los obstáculos técnicos relacionados con la carga y visualización de las imágenes en los diferentes paneles resultaron más complejos de lo anticipado, lo que generó retrasos y limitaciones en el progreso del proyecto. Por ende, tomamos la decisión de crear el nuevo repositorio.

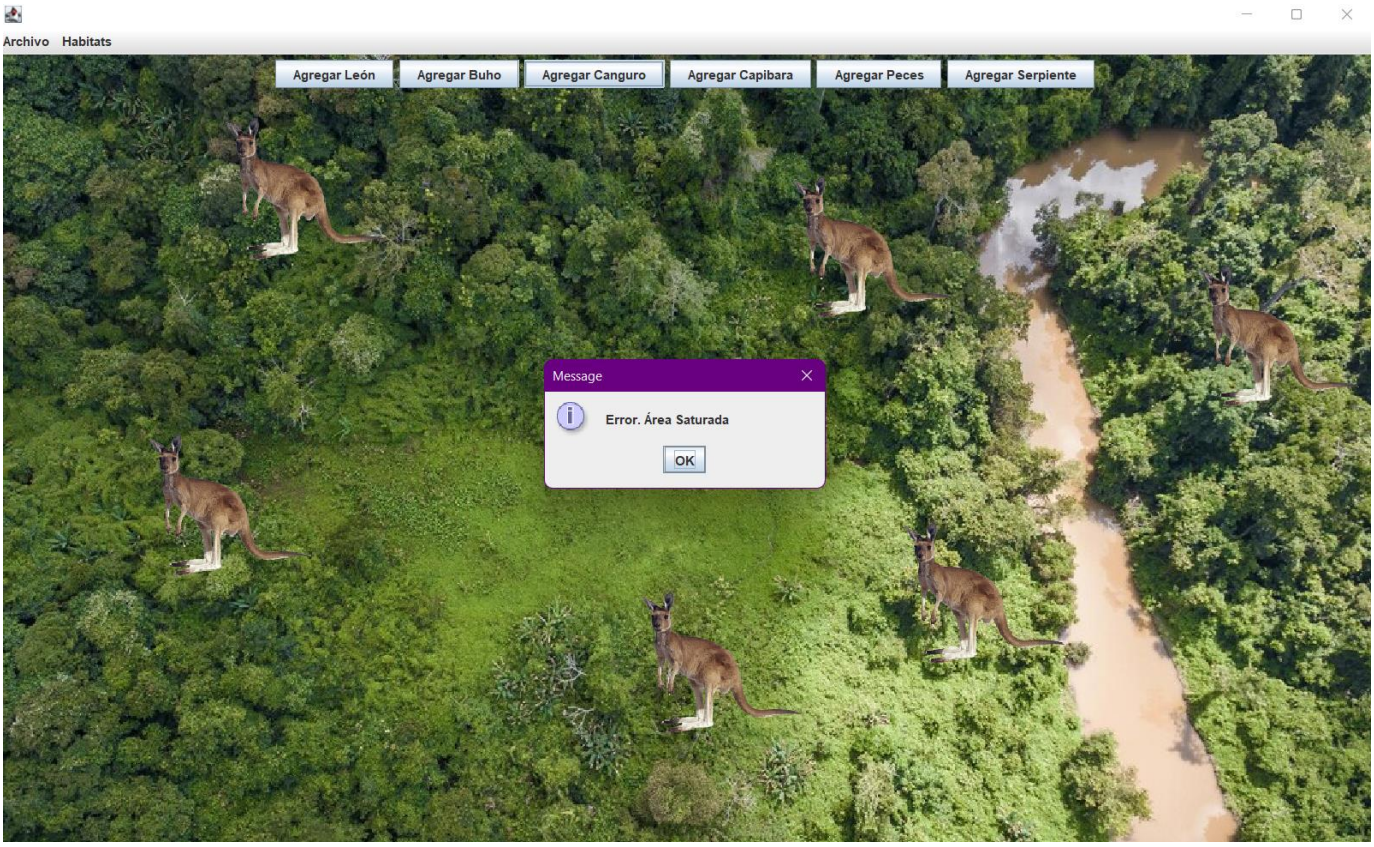
Adjunto capturas de pantalla de la interfaz.




Archivo
Habitats
Agregar León
Agregar Buho
Agregar Canguro
Agregar Capibara
Agregar Peces
Agregar Serpiente


Archivo
Habitats
Sabana
Acuario
Selva
Humedal
Aviario
Bosque
Volver a todos los Habitats
Agregar León
Agregar Buho
Agregar Canguro
Agregar Capibara
Agregar Peces
Agregar Serpiente


Archivo Habitats
Agregar León
Agregar Buho
Agregar Canguro
Agregar Capibara
Agregar Peces
Agregar Serpiente
Error
¡Solo se permite agregar capibaras al hábitat!
OK


Archivo Habitats
Agregar León
Agregar Buho
Agregar Canguro
Agregar Capibara
Agregar Peces
Agregar Serpiente
Message
Error. Área Saturada
OK

Adjunto diagrama de casos de uso y diagrama UML, respectivamente.