

Objetivos**Unidad 1: Introducción a la solución de problemas utilizando algoritmos**

- OE2.1 Modelar las características de un objeto, utilizando tipos de datos primitivos y la técnica de definir constantes para representar los valores posibles de un atributo.
- OE2.2. Utilizar expresiones aritméticas, lógicas, relacionales y operaciones con cadenas en el cuerpo de un método.
- OE2.3. Declarar y hacer llamados a métodos constructores declarados explícitamente en la misma clase y utilizar null en caso de que un objeto no haya sido inicializado.
- OE2.4. Utilizar las instrucciones condicionales como parte del cuerpo de un algoritmo, para poder considerar distintos casos de la solución de un problema.
- OE2.6. Interpretar los errores producidos en tiempo de ejecución para el caso en que se realicen llamados u operaciones con objetos que no han sido contruidos.
- OE2.8. Valorar la coherencia estricta entre el diseño propuesto en el diagrama de clases y la implementación en el lenguaje de programación.

Enunciado

La primera entrega de avances del desarrollo del software del zoológico dejó a las directivas de esta prestigiosa entidad muy impresionadas. Por lo que ahora quieren que se les entregue un programa funcional que resuelva las necesidades anteriormente planteadas ([link](#)).

Junto a esas funcionalidades el Zoológico quiere que los biólogos que van a operar el software también puedan realizar las siguientes acciones:

- Crear nuevos canguros y agregarlos a una Ambiente.
- Eliminar canguros
- Cambiar canguros de Ambiente
- Encontrar los animales del zoológico cuyos nombres empiezan y terminan en vocal.
- Dar un reporte de las fechas de vacunación del los canguros.

Ten en cuenta que cuando se mueven los canguros de Ambiente, o se crean automáticamente se debe actualizar los valores del área de la zonas y los ambientes. Además, en caso de que no se pueda agregar una canguro al ambiente (Porque está lleno o por la restricción territorial de género en los ambientes) se le debe mostrar un mensaje de error al usuario que indique porque no fue posible agregar/trasladar el canguro.

Un programa funcional implica que:

- Se creen algunos valores iniciales para el programa funcione.
- Existan al menos 2 canguros en cada uno de los ambientes en los valores iniciales existentes en el programa.
- Se calculen de manera correcta todos los RF identificados en la entrega pasada y esta.
- Se despliega un menú que permita al usuario elegir la opción que desea utilizar del programa. Al usuario elegir la opción se realiza la operación solicitada por el usuario y se muestra de nuevo el menú.

Entregables entrega 2.

1. Especificación de Requerimientos Funcionales y no funcionales actualizado .
2. Diagrama de Clases Completo actualizado con todos los métodos, atributos, relaciones, constantes, dependencias, constructores necesarios para la solución (incluye el Modelo y el Main en la interfaz). El modelo debe ser elaborado digitalmente, pero NO generado automáticamente (por ejemplo, no es válido entregar modelos generados por Object Aid o ninguna otra herramienta).
3. Trazabilidad del Análisis al Diseño. Una tabla a dos columnas en la que se relaciona cada requerimiento con el método o métodos que permiten satisfacer dicho requerimiento.
4. Implementación en Java 100% funcional. Incluya en la implementación, los comentarios descriptivos sobre los atributos y métodos de cada clase. Recuerde que todos los artefactos generados de fase de diseño e implementación deben ser en inglés.
5. El código debe estar presente un repositorio de [github](https://github.com), en donde se evidencie los avances del proyecto a través del tiempo.

Recuerde que puede encontrar la Rúbrica laboratorio 2 en el siguiente [enlace](#).

Nota: Usted debe entregar un archivo comprimido en formato zip de un directorio con únicamente 2 archivos: 1 archivo en formato pdf con toda la documentación (análisis, diseño y tabla de trazabilidad) y otro archivo comprimido de un directorio con los archivos de codificación en sus respectivos paquetes.

El nombre del archivo comprimido debe tener el formato: PRIMERAPELLIDO_PRIMERNOMBRE.zip (tenga en cuenta que el separador entre el primer apellido y el primer nombre es un guion al piso).

SOLUCIÓN

1 Requerimientos funcionales

Nombre	R.# 1. Registrar un canguro
Resumen	El usuario debe ingresar la información del canguro al sistema para poder ser utilizada dentro del programa y agregarlo a un ambiente
Entradas	
-Nombre -Peso -Estatura -Sexo -Tipo de sangre -Fecha de nacimiento	
Resultados	
Registrar la información del canguro en el sistema para así agregarlo a un ambiente.	

Nombre	R.# 2. Calcular el área requerida de un ambiente
Resumen	El programa debe calcular el área de un ambiente, multiplicando la estatura de cada canguro del ambiente por 8 y luego sumando los 3 resultados.
Entradas	
-estatura del canguro 1 -estatura del canguro 2 -estatura del canguro 3	
Resultados	
Área de un ambiente	

Nombre	R.# 3. Calcular la cantidad de alimento para cada canguro
Resumen	El programa debe calcular la cantidad de alimento para cada canguro, según su peso.
Entradas	
-peso del canguro	
Resultados	
La cantidad de alimento para un canguro	

Nombre	R.# 4. Determinar si un canguro necesita la vacuna contra la rabia
Resumen	El programa debe determinar si un canguro es menor o mayor de un año de nacido, si es menor, necesita la vacuna de la rabia. Si es mayor, no necesita la vacuna.
Entradas	
-fecha de nacimiento del canguro	
Resultados	
Determinar si un canguro necesita la vacuna contra la rabia, y en caso de que falte, dar el tiempo en semanas y días	

Nombre	R.# 5. Calcular el índice de masa corporal (IMC) de cada animal
Resumen	El programa debe determinar el IMC de cada animal por medio de la formula: $IMC = (\text{peso en kg} / (\text{estatura en metros})^2)$.
Entradas	
-peso del animal -altura del animal	
Resultados	
IMC de un canguro	

Nombre	R.# 6. Determinar el nivel de riesgo cardíaco de cada canguro
Resumen	El programa debe determinar para cada canguro que nivel de riesgo cardíaco tienen con base en su IMC y su tipo de sangre
Entradas	
-Índice de masa corporal del canguro -tipo de sangre del canguro	
Resultados	
El nivel de riesgo cardíaco de cada canguro	

Nombre	R.# 7. Notificar a los biólogos cuando los dragones barbados tengan menos de 5Kg de comida
Resumen	El programa debe notificar a los biólogos cuando los dragones barbados tengan menos de 5Kg de comida
Entradas	
-Cantidad de comida en la zona de dragones	
Resultados	
notificar a los biólogos cuando los dragones barbados tienen menos de 5Kg de comida.	

Nombre	R.# 8. Calcular la cantidad de agua para cada canguro
Resumen	El programa debe calcular la cantidad de agua para los canguros de los 3 ambientes teniendo en cuenta que la cantidad de agua para un canguro es: $IMC * 1.5$ litros
Entradas	
-IMC del canguro	
Resultados	
Consumo óptimo de agua de cada canguros.	

Nombre	R.# 9. calcular la cantidad de agua par cada dragon
Resumen	El programa debe calcular el consumo de agua para los dragones barbados teniendo en cuenta que el consumo de agua de un dragón barbado es el IMC *0,75 litros.
Entradas	
-Indice de masa corporal del dragon	
Resultados	
Consumo óptimo de agua para cada dragones barbados.	

Nombre	R.# 10. calcular el area total de la zona de canguros
Resumen	El programa debe sumar el area de cada ambiente, y asi calcular cuanta area requieren la zona de canguros.
Entradas	
-area por canguro	
Resultados	
Area total de la zona de canguros	

Nombre	R.# 11. Determinar si un canguro necesita la vacuna contra la rabia
Resumen	El programa debe determinar si un canguro es menor o mayor de un año de nacido, si es menor, necesita la vacuna de la rabia. Si es mayor, no necesita la vacuna.
Entradas	
<Ninguna>	
Resultados	
Determinar si un canguro necesita la vacuna contra la rabia, y en caso de que falte, dar el tiempo en semanas y días	

Nombre	R.# 12. Separar canguros machos
Resumen	Cada ambiente tiene que tener únicamente un canguro
Entradas	
-sexo del canguroo1 -sexo del canguroo2 -sexo del canguroo3	
Resultados	
Solo dejara registrar un canguro macho por ambiente	

Nombre	R.# 13. Mantener la temperatura de los dragones barbados
Resumen	La zona de dragones de estar permanentemente en 35-40 grados celsius
Entradas	
-Temperatura de la zona de dragones	
Resultados	
Los dragones barbados están en una zona con temperatura óptima	

Nombre	R.# 14. Eliminar canguros
Resumen	El programa debe permitir eliminar un canguro
Entradas	
-numero del ambiente del canguro	
-numero del canguro	
Resultados	
El canguro sera eliminado	

Nombre	R.# 15. Cambiar canguros de ambiente
Resumen	El programa debe permitir cambiar un canguro de ambiente
Entradas	
-numero del ambiente del canguro del que saldra	
-numero del canguro	
-numero del ambiente al que llegara	
Resultados	
El canguro sera cambiado	

Nombre	R.# 16. Encontrar los nombres de los animales que empiecen y terminen por vocal
Resumen	El programa lanzara los nombres de los animales que empiecen y terminen en vocal
Entradas	
-nombre del animal	
Resultados	
Nombres de canguros que empiecen por vocal	

Nombre	R.# 17. Reporte de fechas de vacunación de los canguros
Resumen	El programa dara las fechas de vacunación de cada canguro
Entradas	
-nombre del canguro	
-fecha de nacimiento	
Resultados	
Fecha en la que el canguro fue o sera vacunado	

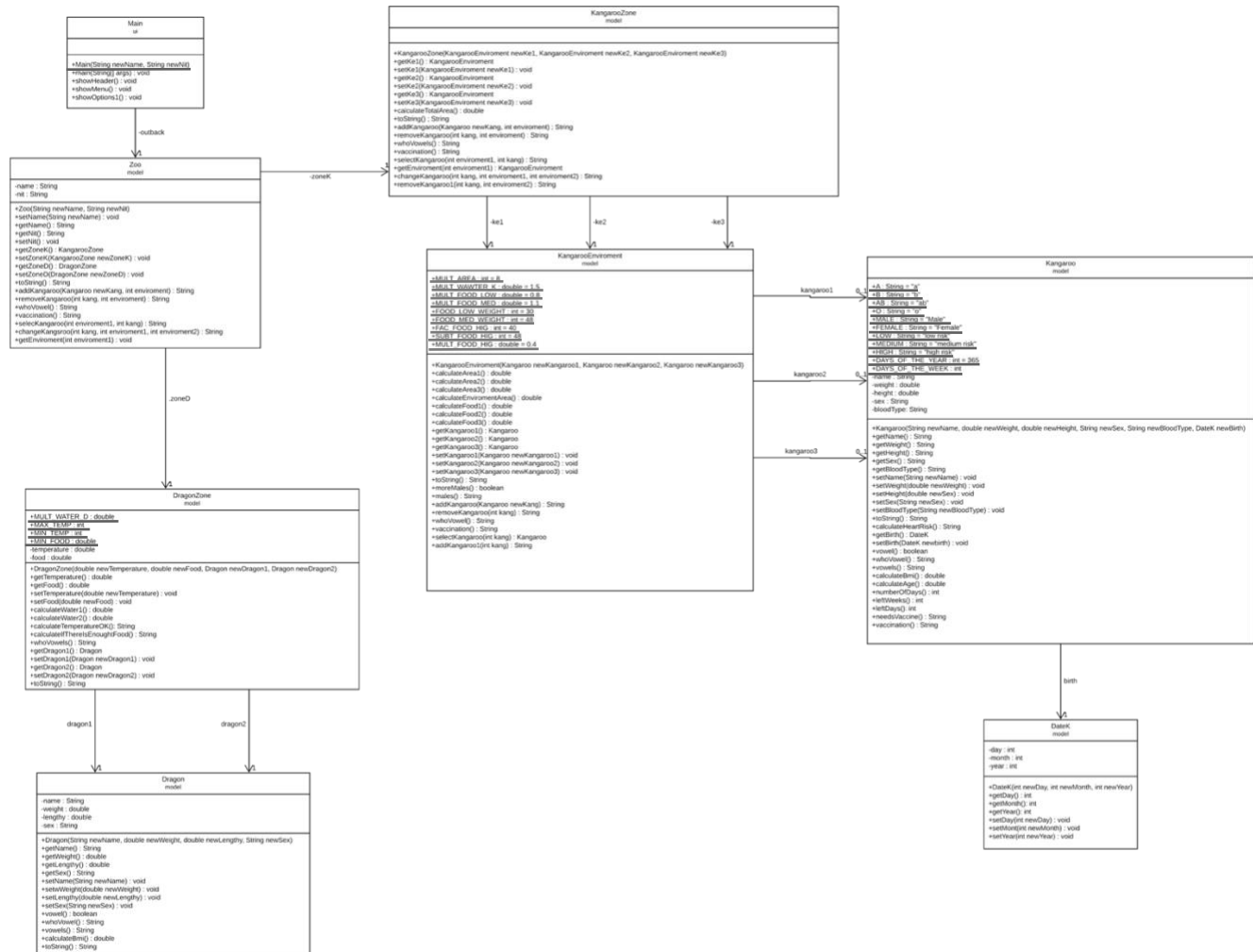
Requerimientos no Funcionales

Nombre	R.# 1. Se creen algunos valores iniciales para el programa funcione.
Resumen	El programa tendra unos valores iniciales por defecto para poder funcionar
Entradas	
<ninguna>	
Resultados	
Programa funcionando	

Nombre	R.# 2. Existan al menos 2 canguros en cada uno de los ambientes en los valores iniciales existentes en el programa.
Resumen	El programa tendra por defecto, al menos, la informaci3n de cada canguro.
Entradas	
<ninguna>	
Resultados	

Nombre	R.# 3. Se despliegue un men3 que permita al usuario elegir la opci3n que desea utilizar del programa. Al usuario elegir la opci3n se realiza la operaci3n solicitada por el usuario y se muestra de nuevo el men3.
Resumen	El programa debera mostrar un menu, cada vez que se termine una operaci3n
Entradas	
<ninguna>	
Resultados	

2.diagrama de clases



3.Trazabilidad del análisis al diseño

Requerimiento Funcional	Clase	Método
R. #1 Registrar un canguro	KangarooEnviroment: KangarooZone:	addKangaroo() addKangaroo()
R. #2 calcular el area requerida de un ambiente	Kangaroo: KangarooEnviroment: KangarooEnviroment: KangarooEnviroment: KangarooEnviroment:	getHeight() calculatearea1() calculatearea2() calculatearea3() calculateEnviromentArea()
R. #3 calcular la cantidad de alimento para cada canguro	Kangaroo: KangarooEnviroment: KangarooEnviroment: KangarooEnviroment:	getWeight() calculateFood1() calculateFood2() calculateFood3()
R. #4 determinar si un canguro necesita la vacuna contra la rabia	Kangaroo: Kangaroo: Kangaroo: Kangaroo: Kangaroo:	numberOfDays() calculateAge() leftWeeks() leftDays() needsVaccine()
R. #5 calcular el indice de masa corporal de cada canguro	Kangaroo: Kangaroo: Kangaroo:	getWeight() getHeight() calculateBmi()
R. #6 determinar el nivel de riesgo cardiaco de cada canguro	Kangaroo: Kangaroo:	getBloodType() calculateHeartRisk()
R. #7 notificar a los biologos cuando la zona de los dragones barbados tengan menos de 5kg de comida en la esquina	DragonZone: DragonZone:	getFood() calculateIfThereIsEnoughFood()
R. #8 calcular la cantidad de agua para cada canguro	Kangaroo: KangarooEnviroment: KangarooEnviroment: KangarooEnviroment:	calculateBmi() calculateWater1() calculateWater2() calculateWater3()
R. #9 calcular la cantidad de agua par cada dragon	Dragon: DragonZone: DragonZone:	calculateBmi() calculaterWater1() calculaterWater2()
R. #10 calcular el area total de la zona de canguros	KangarooZone:	calculateTotalArea()
R. #11 Determinar si un canguro necesita vacuna contra la rabia	Kangaroo:	needsVaccine()
R. #12. Separar canguros machos	Kangaroo KangarooEnviroment: KangarooEnviroment	getSex() moreMales() males()
R. #13. Mantener la temperatura de los dragones barbados	DragonZone:	calculateIfThereIsEnoughFood()
R. #14. Eliminar canguros	KangarooEnviroment:	removeKangaroo()
R. #15. Cambiar canguros de ambiente.	KangarooZone: KangarooZone:	changeKangaroo() selectKangaroo()
R. #16. Encontrar los nombres de los canguros que empiecen y terminan por vocal	Kangaroo: Kangaroo: Kangaroo: Dragon: Dragon: Dragon:	vowel() whoVowel() vowels() vowel() whoVowel() vowels()
R. #17. Reporte de fechas de vacunación de los canguros	Kangaroo:	vaccination()

5.GitHub [link](#) //no hay reporte de actualización ya que se nos enseñó a usarlo, el sabado antes de la entrega.