

LABORATORIO 2

ANDREA NÚÑEZ RODRÍGUEZ

A00355453

ALGORTIMOS Y PROGRAMACIÓN 1

UNIVERSIDAD ICESI

FEBRERO DE 2019

1. **Identificación del problema:** Buscar un método para monitorear el bienestar de los animales de la exhibición del zoológico a través de un equipo de desarrollo de software.
2. **REQUERIMIENTOS FUNCIONALES**

Nombre	R.# 1. Conocer la información del ambiente de canguros y de los dragones
Resumen	El programa permite que el usuario conozca la información de los ambientes de los canguros y de los dragones.
Entradas: <ul style="list-style-type: none"> • Área • Cantidad de comida • Cantidad de agua 	
Resultados: La información del ambiente ha sido cargada.	

Nombre	R.# 2. Crear un canguro
Resumen	El usuario debe ingresar la información del canguro al sistema para poder realizar futuras operaciones con esta.
Entradas: <ul style="list-style-type: none"> • Nombre • Peso • Estatura • Sexo • Tipo de sangre • Fecha de nacimiento • Ambiente al que se desea que el canguro sea ingresado 	
Resultados: Los datos del canguro han sido cargados.	

Nombre	R.# 3. Calcular área del ambiente de canguros
Resumen	El programa debe ser capaz de calcular el área de un ambiente multiplicando la estatura de cada canguro por ocho y luego sumar los resultados del total de canguros de cada zona.
Entradas:	
A cual ambiente se le desea calcular el área	
Resultados: Se ha calculado el área de un ambiente.	

Nombre	R.# 4. Calcular la cantidad de alimento para los canguros
Resumen	El programa debe ser capaz de calcular la cantidad de alimento que se necesita para cada canguro según su peso.
Entradas:	
<Ninguna>	
Resultados: Se ha calculado la cantidad de alimento para cada canguro.	

Nombre	R.# 5. Determinar si un canguro necesita la vacuna contra la rabia
Resumen	El programa debe determinar, a partir de los datos del canguro, si este es apto para la vacuna de la rabia, dependiendo si es menor o mayor a un año de nacido.
Entradas: Condición que permita reconocer cuando es el tiempo indicado en el cual el canguro debe ser vacunado	
Resultados: Determinar si el canguro necesita la vacuna contra la rabia.	

Nombre	R.# 6. Calcular el índice de masa corporal para cada canguro
Resumen	El programa debe calcular el IMC de cada canguro.
Entrada:	
<Ninguna>	
Resultados: Se ha calculado el IMC de cada canguro.	

Nombre	R.# 7. Determinar el nivel de riesgo cardiaco de cada canguro
Resumen	El programa debe ser capaz de determinar para cada canguro el nivel de riesgo cardiaco dependiendo su IMC y tipo de sangre.
Entradas: Condición que permita reconocer, a partir del IMC del canguro, cuando éste tiene hacia enfermedades cardíacas.	
<Ninguna>	
Resultados: Se ha determinado el nivel de riesgo cardiaco para cada canguro.	

Nombre	R.# 8. Registrar un dragón barbado
Resumen	El usuario debe ingresar la información del dragón barbado para poder realizar futuras operaciones con esta.
Entradas:	
<ul style="list-style-type: none"> Nombre 	

<ul style="list-style-type: none"> • Peso • Estatura • Sexo • Tipo de Sangre • Fecha de Nacimiento
<Ninguna>
Resultados: Se ha registrado la información del dragón barbado en el sistema.

Nombre	R.# 9. Notificar a los biólogos cuando los dragones barbados tengan menos de 5kg de comida.
Resumen	El programa debe ser capaz de notificar a los biólogos cuando los dragones barbados tengan menos de 5kg de comida.
Entradas: Condición que indique que la comida de los canguros es la suficiente o no dependiendo de una asignación inicial.	
Resultados: El programa notifica a los biólogos cuando los dragones barbados tienen menos de 5kg de comida.	

Nombre	R.# 10. Calcular el consumo de agua en cada ambiente de canguros.
Resumen	El programa debe ser capaz de calcular el consumo de agua para los canguros teniendo en cuenta su IMC * 1,5 litros.
Entradas:	
<Ninguna>	
Resultados: El programa ha calculado el consumo de agua ideal para los ambientes de los canguros.	
Nombre	R.# 11. Consultar el consumo de agua en la zona de dragones
Resumen	El programa debe calcular el consumo de agua para los dragones barbados tomando en cuenta el IMC de cada dragón barbado * 0,75 litros.
Entradas:	
<Ninguna>	
Resultados: El programa ha calculado el consumo de agua ideal para los ambientes de los dragones barbados.	
Nombre	R.# 12. Eliminar un canguro.
Resumen	El programa debe ser capaz de permitir al usuario eliminar un canguro.
Entradas: El nombre del canguro que se desea eliminar y el ambiente del cual proviene.	
Resultados: El programa ha permitido al usuario eliminar un canguro existente.	

Nombre	R.# 13. Encontrar los animales que tengan una vocal al comienzo y al final de su nombre.
Resumen	El programa debe ser capaz de mostrar que canguros de los ambientes contienen una vocal en el principio y al final de su nombre.
Entradas: Nombres de los canguros de todos los ambientes.	
Resultados: Los nombres de todos los canguros.	

Nombre	R.# 14. Cambiar un canguro de ambiente.
Resumen	El programa debe ser capaz de permitir al usuario cambiar un canguro de un ambiente a otro.
Entradas:	
Resultados: El canguro se ha cambiado de ambiente	

REQUERIMIENTOS NO FUNCIONALES

Nombre	R.# 1. Registrar un canguro macho en cada ambiente
Resumen	El programa debe registrar únicamente un canguro macho por cada ambiente.
Entradas:	
<Ninguna>	
Resultados: El programa ha registrado un canguro macho por cada ambiente.	

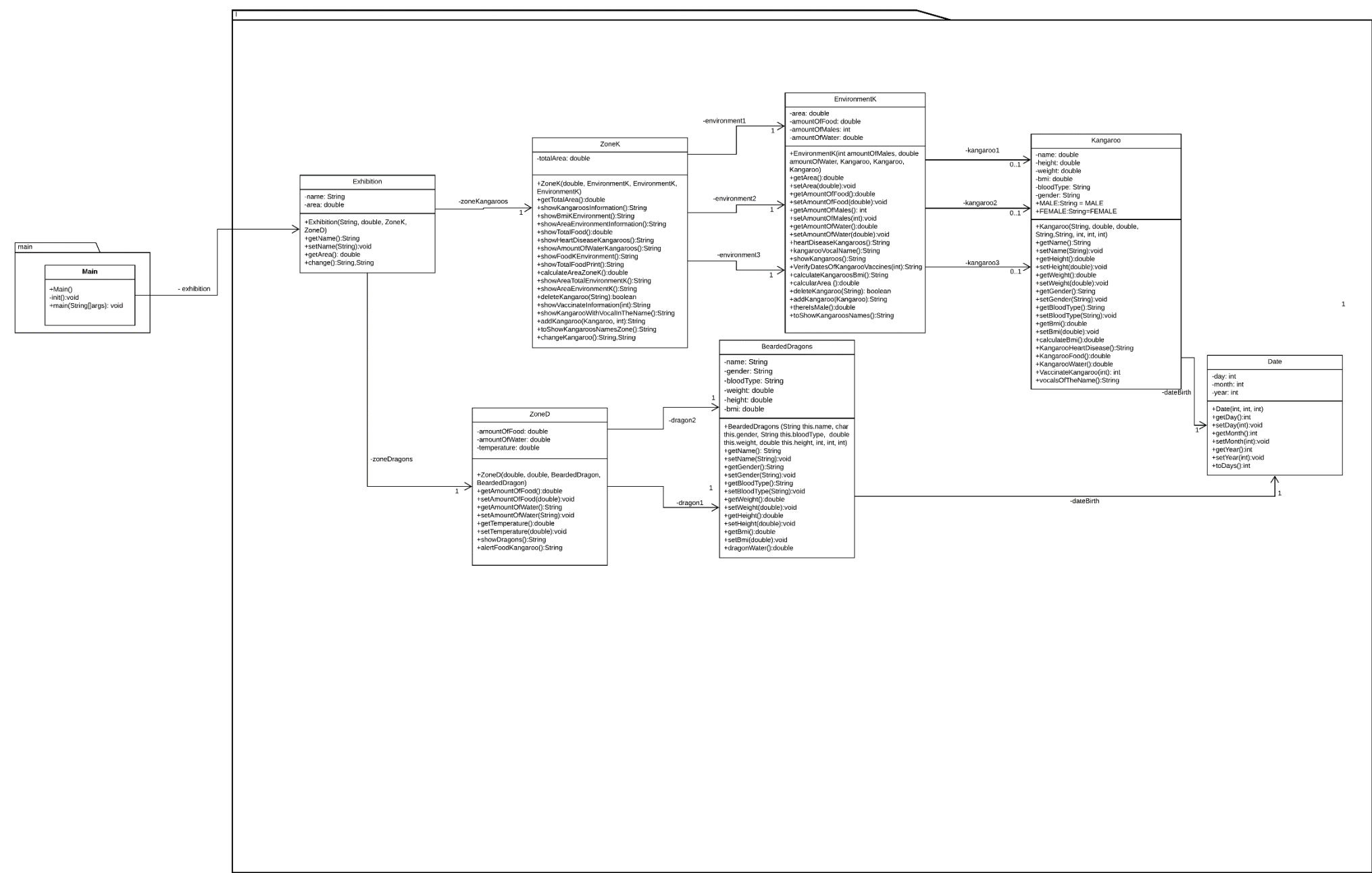
Nombre	R.# 2. Tener registrados al menos dos canguros en cada ambiente y que si los hay, estos no sean machos en el mismo ambiente.
Resumen	El programa debe ser capaz de tener previamente registrado al menos a dos canguros por ambiente con la condición de que solo haya uno por ambiente.
Entradas: Datos de los canguros.	
Resultados: El programa tiene el registro de los canguros existentes en los ambientes.	

Trazabilidad

REQUERIMIENTO	MÉTODOS	CLASES
R1: Conocer la información del ambiente de canguros y de los dragones	+ getAmountOfFood():double + getAmountOfWater():String + getAmountOfMales():int + showKangaroos():String + calcularArea():double + <u>thereIsMale():boolean</u>	ZoneK ZoneD
R.# 2. Crear un canguro	+ addKangaroo(Kangaroo):String	EnvironmentK
R.# 3. Calcular área del ambiente de canguros	+ calcularArea():double +getHeight():double	Kangaroo EnvironmentK
R.# 4. Calcular la cantidad de alimento para los canguros	+KangarooFood()	Kangaroo
R.# 5. Determinar si un canguro necesita la vacuna contra la rabia	+VaccinateKangaroo(int):int + toDays():int +VerifyDatesOfKangarooVaccines(int):String	EnvironmentK Kangaroo
R6: Calcular el índice de masa corporal para cada canguro	+ calculateBmi():double + getWeight():double + getHeight()	Kangaroo
R.# 7. Determinar el nivel de riesgo cardiaco de cada canguro	+ KangarooHeartDisease()	Kangaroo
R.# 8. Registrar un dragón barbado	+BeardedDragon (String name, char gender, String bloodType, double weight, double height, int, int, int)	BeardedDragon
R.# 9. Notificar a los biólogos cuando los dragones barbados tengan menos de 5kg de comida.	+getAmountOfFood():double +alertFoodKangaroo()	ZoneD
R.# 10. Calcular el consumo de agua en cada ambiente de canguros.	+calculateAmountOfWater():double +KangarooWater()	ZoneK Kangaroo
R.# 11. Consultar el consumo de agua en la zona de dragones	+getAmountOfWater():double +dragonWater():double	ZoneD BeardedDragon
R.# 12. Eliminar un canguro.	+ deleteKangaroo(String): Boolean +getName():String	EnvironmentK Kangaroo

R.# 13. Encontrar los animales que tengan una vocal al comienzo y al final de su nombre.	+vocalsOfTheName():String +kangarooVocalName():String	Kangaroo EnvironmentK
R.#14. Cambiar un canguro de ambiente	+changeKangaroo():String, String +change():String,String	ZoneK <u>Exhibition</u>

DIAGRAMA DE CLASES



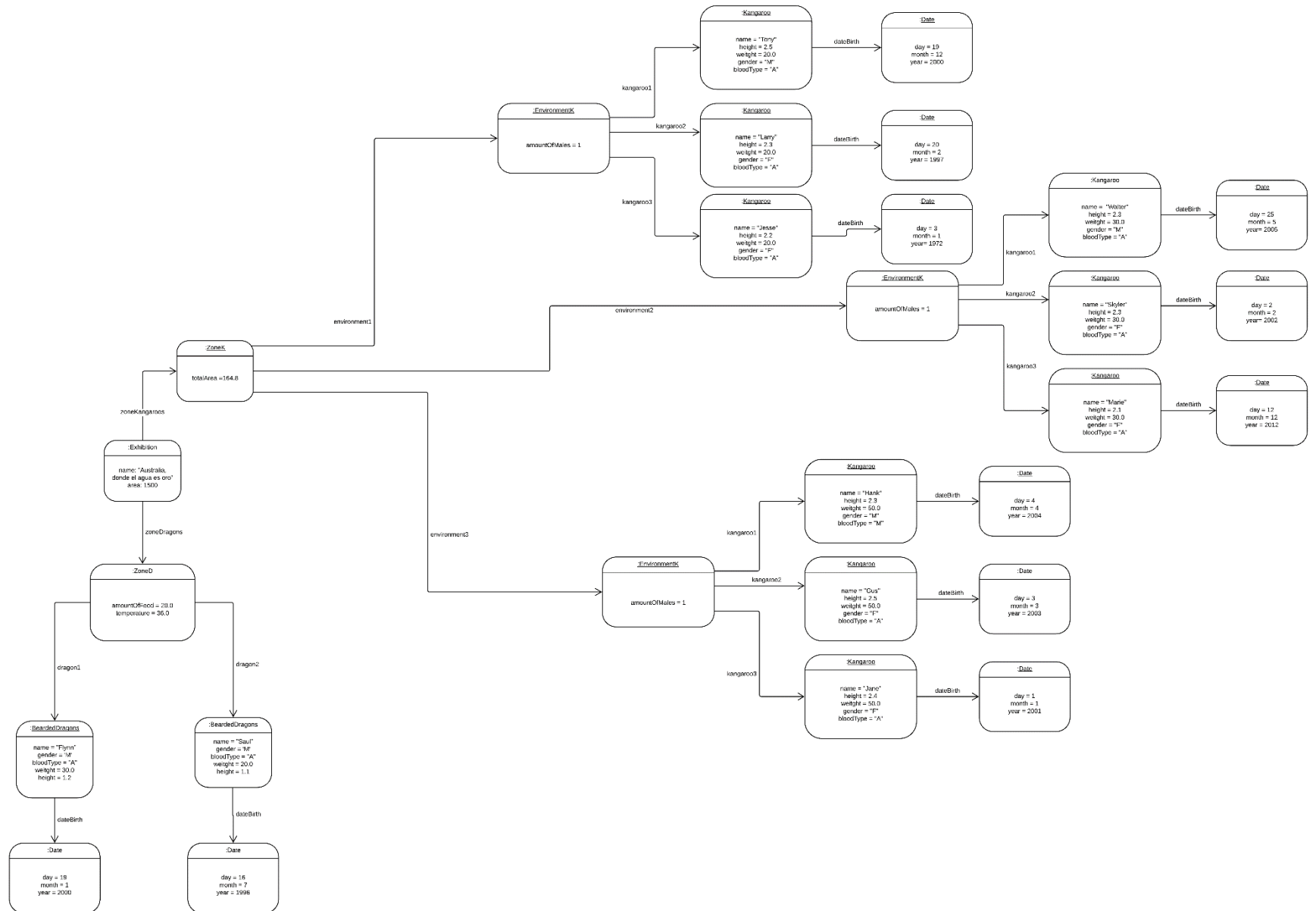


DIAGRAMA DE OBJETOS