Evaluación Formativa N°2

Sigla	Nombre Asignatura	Tiempo Asignado	% Ponderación
DSY	Desarrollo Orientado a Objetos	30 minutos	0%

1. Situación evaluativa

X	Ejecución práctica
	Entrega de encargo
	Presentación

2. Agente evaluativo

	Heteroevaluación
	Coevaluación
X	Autoevaluación

3. Tabla de Especificaciones

Resultado de Aprendizaje	Indicador de Logro (IL)	Ponderación Indicador Logro
RA2 Aplica conceptos avanzados del paradigma orientado a objetos en componentes de software, para dar respuesta a requerimientos de clientes	IL 2.1 Aplica las sentencias de ciclos, para ser representadas en la solución según el requerimiento del usuario en un caso de negocios.	15%
	IL 2.2 Aplica conceptos de encapsulación avanzada permitiendo cambios internos sin afectar otros componentes del sistema del software.	15%
	IL 2.3 Configura una colección para almacenar información temporal en el programa según lo solicitado por el usuario.	30%
	IL 2.4 Aplica los conceptos de herencia en la programación orientada a objetos que permitan dar solución a un problema planteado un caso de negocios.	20%
	IL 2.5 Utiliza abstracción y polimorfismo de manera efectiva, para permitir la creación de interfaces flexibles y genéricas que puedan adaptarse a diferentes situaciones.	20%
	Total	**100%**

4. Instrucciones generales para el/la estudiante

Esta es una evaluación que corresponde a una ejecución práctica del tipo formativa y la cual no tiene asignada un ponderador sobre la nota final de la asignatura.

El tiempo para desarrollar esta evaluación es de 30 minutos y se realiza de manera grupal (máximos 2 alumnos por grupo) en laboratorio PC avanzado.

En esta evaluación el estudiante trabajará en la resolución de un caso para implementar colecciones y aplicar conceptos básicos de herencia. Debe implementar un menú para invocar diferentes funcionalidades según el caso.

Para dejar registro deberá realizar los siguientes puntos:

- Realizar el diagrama de clases
- Programar en el lenguaje JAVA utilizando el IDE Netbeans el diagrama de clases generado
- Deberá dejar registro del trabajo realizado por medio de la plataforma de Blackboad.

5. Evaluación

Implementación de un Sistema de Registro de Mascotas

Contexto: Una veterinaria llamada **AirbnbPet**, ha decidido mejorar su proceso de atención y contratar los servicios de PGY2121 para desarrolle un sistema que permita almacenar la información de las mascotas que se alojan en sus dependencias

Requerimiento Inicial: Las mascotas que son alojadas se identifican con un nombre, peso, edad y días de alojamiento e inicialmente se atenderán solo perros, gatos y conejos.

Los perros se caracterizan por salir hacer ejercicio durante el día. En cambio, los gatos se definen por medio de un pedigrí y los conejos se identifican con tipo de comida que ingieran (dieta).

El desarrollo de la aplicación debe responder a los siguientes requerimientos:

- Cada mascota debe identificarse con un código alfanumérico único.
- Cada mascota puede o no requerir la supervisión de un veterinario durante la noche.
- Debe existir un método para mostrar los datos de mascota que debe ser sobrescrito dependiendo del tipo.
- La clase padre debe ser abstracta.
- Debe considerar generar por lo menos un constructor vacío y uno sobrecargado con todos sus datos en cada clase que se deba instanciar

Considere una clase auxiliar que permita almacenar las mascotas que se alojen en el hotel con las siguientes funcionalidades implementadas:

- Validar que la mascota agregada no exista por su código antes de ingresarla
- Un método que liste todas las mascotas alojadas
- Un método que retorne la cantidad de mascotas

Todas las clases deben tener los siguientes métodos implementados: constructores, accesadores y mutadores.

- En la clase ***principal*** debe:

- Agregar 3 perros, 2 gatos y 2 conejos a la colección
- Listar todas las mascotas y el total de mascotas

6. Pauta de Evaluación

Categoría	% logro	Descripción niveles de logro
Muy buen desempeño	**100%**	Demuestra un desempeño destacado, evidenciando el logro de todos los aspectos evaluados en el indicador.
Buen desempeño	**80%**	Demuestra un alto desempeño del indicador, presentando pequeñas omisiones, dificultades y/o errores.
Desempeño aceptable	**60%**	Demuestra un desempeño competente, evidenciando el logro de los elementos básicos del indicador, pero con omisiones, dificultades o errores.
Desempeño incipiente	**30%**	Presenta importantes omisiones, dificultades o errores en el desempeño, que no permiten evidenciar los elementos básicos del logro del indicador, por lo que no puede ser considerado competente.
Desempeño no logrado	**0%**	Presenta ausencia o incorrecto desempeño.

Indicador de Evaluación	Categorías de Respuesta					Ponderación Indicador de Evaluación
	Muy buen desempeño 100%	**Buen desempeño 80%**	**Desempeño aceptable 60%**	**Desempeño incipiente 30%**	**Desempeño no logrado 0%**	
IL 2.1 Aplica las sentencias de ciclos, para ser representadas en la solución según el requerimiento del usuario en un caso de negocios.	Utiliza las estructuras de repetición sin errores en la aplicación	Utiliza correctamente las estructuras de repetición existiendo un error lógico	Utiliza correctamente las estructuras de repetición existiendo dos errores lógicos	Utiliza correctamente las estructuras de repetición existiendo tres errores lógicos	Utiliza correctamente las estructuras de repetición existiendo más de tres errores lógicos o una de las	15%

					sentencias impide la ejecución del programa	
IL 2.2 Aplica conceptos de encapsulación avanzada permitiendo cambios internos sin afectar otros componentes del sistema del software.	Implementa modificadores y métodos de acceso y paquetes, y clases abstractas para ocultar implementación interna de una clase.	N/A	Implementa modificadores y métodos de acceso y paquetes, pero no codifica clases abstractas	N/A	No codifica modificadores de accesos ni implementa clase abstractas o la aplicación no compila	15%
IL 2.3 Configura una colección para almacenar información temporal en el programa según lo solicitado por el usuario.	Crea TODAS las colecciones necesarias e implementa los métodos que permiten agregar, buscar, modificar, eliminar y listar las colecciones, permitiendo da solución al problema planteado.	Crea TODAS las colecciones necesarias, PERO falta implementar UNO de los métodos requeridos para gestionar la información que permita dar solución al problema planteado.	Crea TODAS las colecciones necesarias, PERO falta implementar DOS de los métodos requeridos para gestionar la información que permita dar solución al problema planteado.	Crea TODAS las colecciones necesarias, PERO falta implementar TRES O MÁS de los métodos requeridos para gestionar la información que permita dar solución al problema planteado.	No implementa colecciones o faltan más de dos métodos por implementar o la aplicación no compila.	30%
IL 2.4 Aplica los conceptos de herencia en la programación orientada a objetos que	Aplica TODOS los conceptos de herencia	Define correctamente las clases	Aplica los conceptos de herencia,	Definir AL MENOS UNA clase hija y	No aplica concepto de herencia o la	20%

permitan dar solución a un problema planteado un caso de negocios.	dentro de la jerarquía de clases para dar solución al problema planteado.	hijas, PERO no aplica la sobrecarga de los constructores (super).	PERO falta una de las clases hijas.	aplica al menos UNA sobrecarga de constructor.	aplicación no compila.	
IL 2.5 Utiliza abstracción y polimorfismo de manera efectiva, para permitir la creación de interfaces flexibles y genéricas que puedan adaptarse a diferentes situaciones.	Implementa interfaces, polimorfismo y clases abstractas que permitan a objetos de diferentes clases compartir métodos y comportamiento, que permita dar solución al problema planteado.	Implementa AL MENOS DOS : -interfaces -polimorfismo -clases abstractas.	Implementa AL MENOS DOS conceptos CON ALGÚN ERROR: -interfaces -polimorfismo -clases abstractas.	Implementa AL MENOS UN concepto: -interfaces -polimorfismo -clases abstractas.	No codifica interfaces, polimorfismo, clase abstractas o la aplicación no compila.	20%
					Total	**100%**