UNIVERSIDAD DE LIMA

FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS

ASIGNATURA: Lenguajes de Programación
PERÍODO ACADÉMICO: 2024-2
FECHA: 3/12/2024
TIEMPO: 100 minutos

NOTA

Laboratorio 3

CÓDIGO	APELLIDOS Y NOMBRES	SECCIÓN

ANTES DE INICIAR LA EVALUACIÓN DEBE LEER LAS INSTRUCCIONES

Si la evaluación indica cargar algún archivo en la computadora, recuerde que es responsabilidad del estudiante hacerlo en el tiempo establecido y con las instrucciones dadas. Debe indicar el número de la misma en el recuadro siguiente:

INSTRUCCIONES GENERALES:

- La evaluación consta de 2 preguntas. Solo podrá elegir una pregunta 0 o el envío del informe de CIIS. En caso de enviar solución a esta pregunta, se evaluará lo enviado y no el informe del CIIS.
- Debe entregar TODO su código fuente por la plataforma Blackboard.
- Puede utilizar apuntes digitales (no físicos). **No debe de tener internet activado.**
- **Leer detenidamente las situaciones que ocasionarán la anulación de la evaluación, que se encuentran a continuación.**

SITUACIONES QUE OCASIONARÁN LA ANULACIÓN DE LA EVALUACIÓN:

- Mantener prendidos teléfonos celulares, relojes smart, así como cualquier otro medio o dispositivo electrónico de comunicación.
- Utilizar material de consulta no autorizado (apuntes de clase, fotocopias o materiales similares).
- Utilizar calculadora no estando permitido.
- Compartir o intercambiar hojas, tablas, cualquier material impreso, dispositivo electrónico, durante el desarrollo de la evaluación.
- Conversar durante el desarrollo de la evaluación.

Los profesores de la asignatura

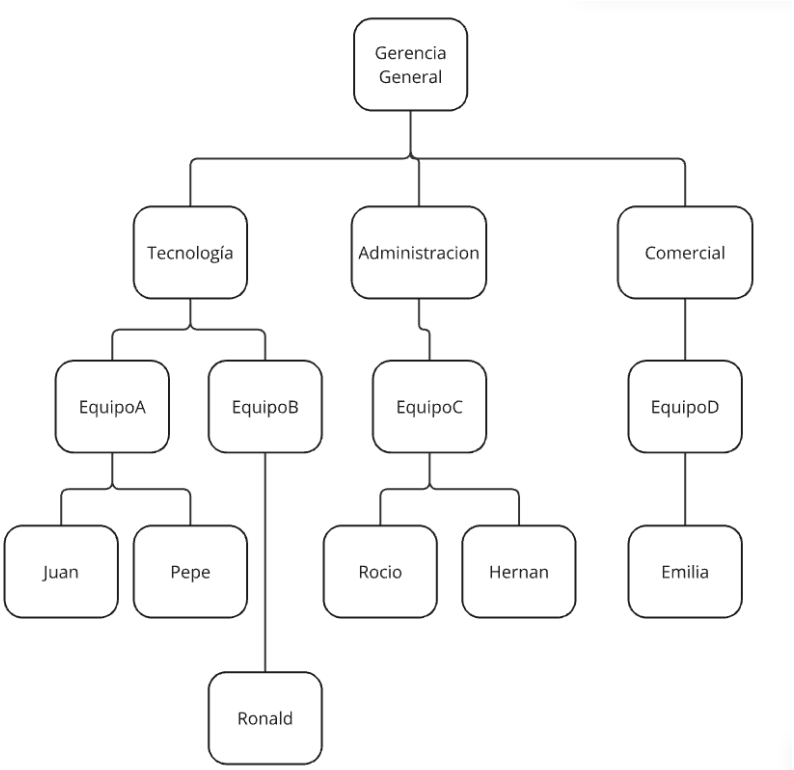
_____

FIRMA DEL ALUMNO
(LEYÓ LAS INSTRUCCIONES)

PREGUNTA 0 (5 puntos)

Nota: Esta pregunta es opcional si ha enviado su informe del CIIS. En caso de enviarla, automáticamente se evalúa su entrega.

Se le pide implementar un programa en **prolog** que permita modelar mediante hechos (facts) la estructura organizaciones de una empresa, según el siguiente gráfico:



Como ven, el segundo nivel son las áreas de la empresa y como tercer nivel los equipos. Al final vendrían a estar las personas que pertenecen a dichos equipos.

Deberá implementar lo siguiente:

- Los hechos (fact) que permitirían esta estructura. Los predicados a implementar serán **pertenece**. (2 puntos)
- Las siguientes reglase:
    - comparten_area(persona1, persona2). Pertenecen a la misma área, pero pueden pertenecer a distinto equipo. (2 puntos)
    - mismo_equipo(persona1, persona2). Pertenecen al mismo equipo. (1 punto)

PREGUNTA 1 (15 puntos)

Papa Noel necesita ayuda. Como sabemos, Papa Noel recibe miles de cartas de los niños para que pueda entregarles un regalo. Por este motivo, necesita un sistema de software que le permita automatizar ciertas actividades, para que durante navidad ningún niño se quede sin regalo.

Para esto, Papa Noel puede recibir 2 tipos de cartas: cartas que piden juguetes físicos (como carritos, muñecas, etc) y cartas que piden juguetes digitales (videojuegos). Para cualquiera de los dos casos, cada carta debe almacenar la siguiente información: nombre del niño (string), edad (int), texto de la carta (string), peso (float) y estaEntregada (bool). Para el caso de la carta de los juguetes físicos, la información que se debe de

almacenar es la siguiente: direccion de entrega (string), mientras que, para la carta de los juguetes digitales, deberá de constar el email del niño (string) y nombre de usuario de steam (string).

Además, se le pide implementar la clase GestorCartas que tenga las siguientes propiedades:
- Atributos (todos privados):
    - listaCartas, que almacene las cartas de todos los niños.
    - indiceCartaActual, entero que nos indica la posición de la carta que será procesada.
    - pesoMaximo, float que nos indica la capacidad máxima del trineo.
- Métodos:
    - Constructor(float), que reciba el peso máximo y lo asigne al atributo correspondiente.
    - AgregarCarta(Carta*)
    - BuscarCarta(string nombre)
    - ObtenerCarta
    - ImprimirReporte

El método **AgregarCarta** deberá recibir un puntero a una carta (superclase) y agregarlo como parte de la lista de cartas. Tomar en cuenta que, al agregar la carta, debe comprobarse de no excederse en el peso (atributo pesoMaximo). Si no puede agregar una carta deberá retornar false. Caso que se agregue una carta correctamente, deberá de devolver true.

El método **BuscarCarta** debe buscar la carta dado el nombre que se recibe como argumento de entrada y devolverla. Debe realizar la mínima cantidad de copias (ser eficiente).

El método **ObtenerCarta** que deberá de devolver la primera carta (puntero) ingresada y marcarla como entregada (campo estaEntregada). Esta carta la deben de obtener con el atributo indiceCartaActual y luego actualizar este indice. **¡Ojo!** No deben de eliminar la carta de la lista. Debe validar que ya no hay más cartas que devolver, en ese caso debe de retornar nullptr.

El método **ImprimirReporte**. Este método deberá de imprimir la lista de cartas así como la cantidad de cartas ya entregadas. Para este método deberán de llamar al método Imprimir de cada carta. Este método es polimórfico por lo que la forma de imprimir cada carta debe ser distinta. A continuación, un ejemplo de cómo debería ser el reporte para una lista de 3 cartas (2 físicas y 1 una digital).

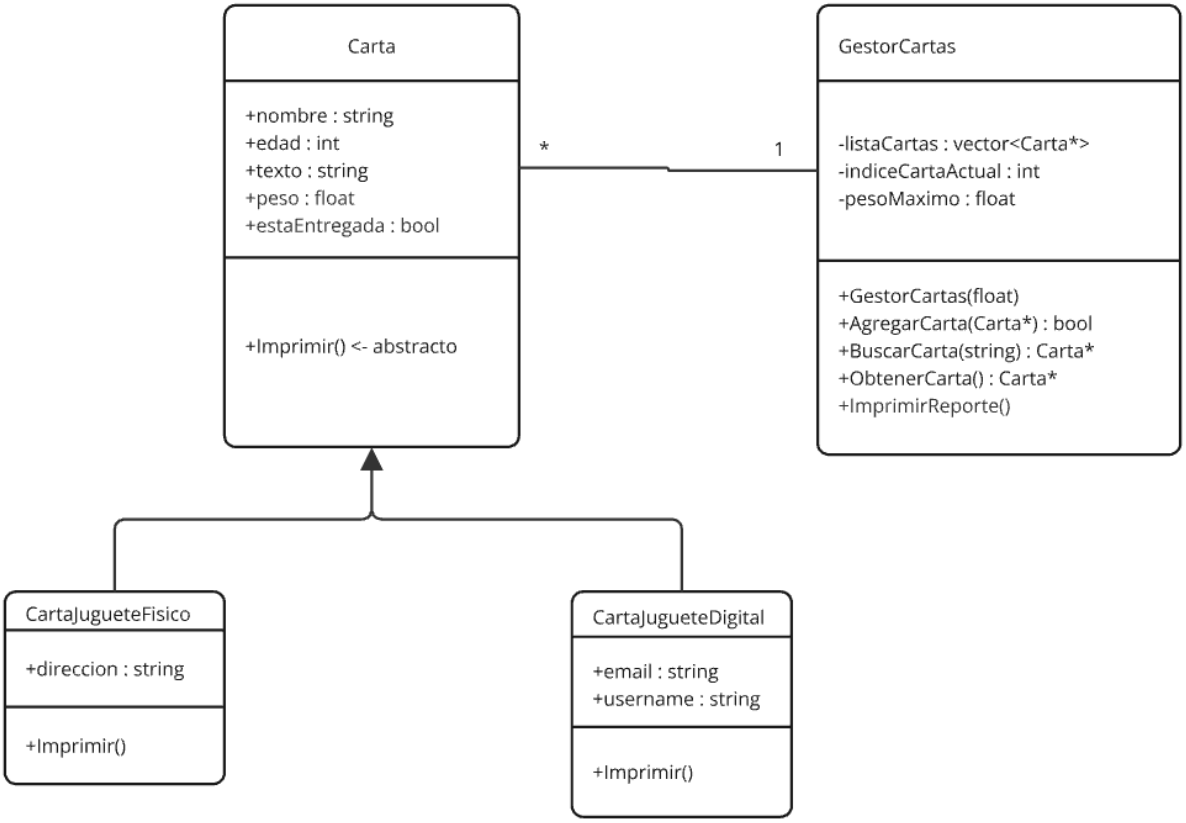
```
============================================================
REPORTE DE CARTAS DE PAPA NOEL
============================================================
- Cantidad de cartas entregadas: 1

Carta 1
Nombre: Pepito
Edad: 10
Texto: Papa Noel, quiero que me regales un auto de carrera
Peso: 0.4
Direccion: "Los Nazgules 123 San Benito"
ENTREGADA

Carta 2
Nombre: Luchita
Edad: 12
Texto: Papa Noel, quiero que me regales el GTA VI
Peso: 0.2
Email: luchita@mail.com
Username: luxita
NO ENTREGADA
```

```
Carta 3
Nombre: Flor
Edad: 5
Texto: Papa Noel, quiero que me regales a Bluey
Peso: 1
Direccion: "San Norpepe 987 Villa Rosa"
NO ENTREGADA
```

Además, se le entrega el siguiente diagrama UML con la información dada anteriormente.



Para finalizar, debe crear 2 unidades de compilación:

- carta : donde irán las clases Carta, CartaJugueteFisico y CartaJugueteDigital.
- gestor: donde irá la clase GestorCartas.

Rúbrica

Definición de clase Carta,sus subclases con sus métodos.	2
Método polimórfico Imprimir().	2
Definición de clase GestorCartas.	1
Implementación correcta de constructor de GestorCartas.	1
Implementación de método AgregarCarta.	3
Implementación de método BuscarCarta.	3
Implementación de método ObtenerCarta.	3
Implementación de método ImprimirReporte.	3
Archivo Makefile	2