

# Programación 4

EXAMEN DICIEMBRE 2022

Por favor siga las siguientes indicaciones:

- Escriba con lápiz y de un solo lado de las hojas
- Escriba su nombre y número de documento en todas las hojas que entregue
- Numere las hojas e indique el total de hojas en la primera de ellas
- Recuerde entregar su número de parcial junto al parcial
- Está prohibido el uso de computadoras, tabletas o teléfonos durante el parcial

## **Problema 1 (30 puntos)**

Se desea modelar un sistema para un portal de innovación en el cual los usuarios pueden cargar proyectos y tratar de conseguir financiamiento para llevarlos a cabo. Cada proyecto puede ser de investigación o de emprendimiento. Todos los proyectos tienen un nombre (que lo identifica), una descripción, y la URL de un video explicativo. Los proyectos de emprendimiento tienen además un monto total a financiar. Los usuarios, de los cuales se sabe sólo su nombre (que los identifica), podrán ser investigadores o colaboradores. Cada proyecto tiene un investigador responsable y hasta 4 investigadores asociados. De los investigadores se conoce, además, su página web y una descripción de su área de experiencia. Los colaboradores podrán seguir proyectos. Un colaborador que sigue un proyecto de emprendimiento podrá realizar una donación de dinero por el monto que desee para poder realizarlo. Un proyecto no puede recibir más donaciones que su monto a financiar. Además, se quiere registrar también cuánto es el total que ha donado cada colaborador hasta el momento (o sea, la suma de todas sus donaciones a diferentes proyectos).

Se cuenta también con la descripción del siguiente Caso de Uso:

Caso de Uso	Crear proyecto de emprendimiento
Actor	Investigador
Descripción	El caso de uso comienza cuando un investigador indica que desea crear un proyecto. Para esto ingresa su nombre, el nombre del proyecto, e indica que se trata de proyecto de emprendimiento. El investigador indica posteriormente el monto a financiar. El sistema muestra la lista de todos los investigadores. Primero, el investigador registra el nombre del investigador que será responsable del proyecto. Luego, registra como investigador del proyecto a un investigador a la vez en el sistema. Cuando el investigador termina de registrar investigadores, el sistema solicita que confirme o cancele el registro. En caso de que el investigador confirme el sistema almacena la información y lista los nombres de los investigadores, indicando también el nombre del proyecto y su descripción. En caso de que el investigador cancele, se elimina el registro de lo realizado.

**Se pide:**

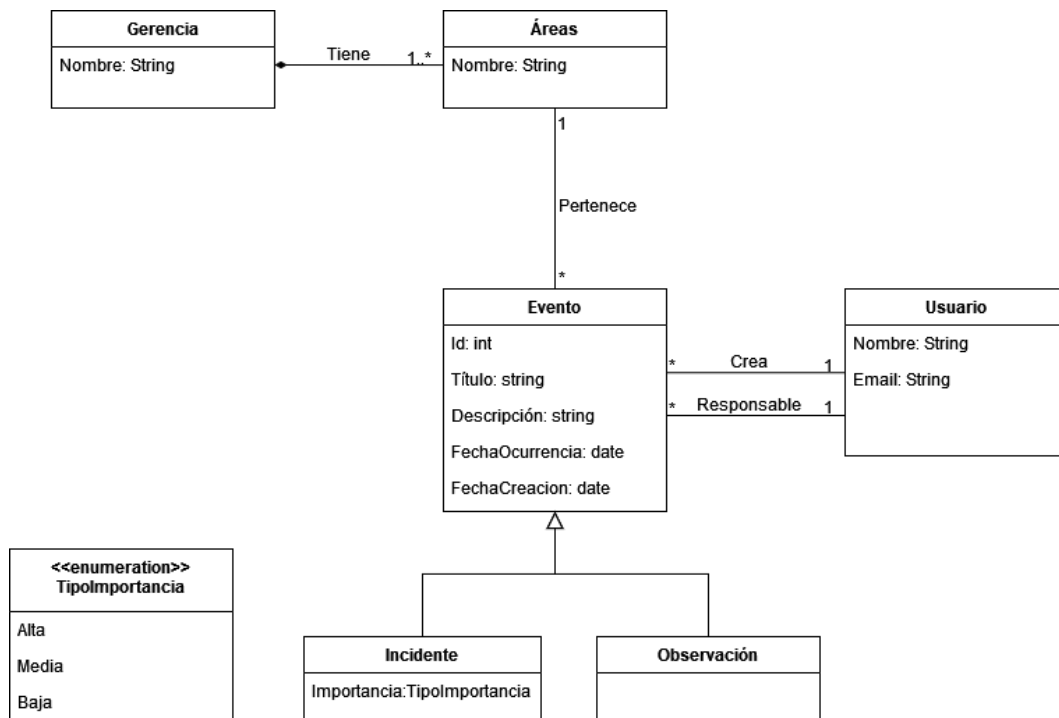
- ¿Cuál es el propósito de la actividad de análisis?
- Realizar el Modelo de Dominio de la realidad planteada, incluyendo restricciones en lenguaje natural.
- Realizar un Diagrama de Secuencia del Sistema (DSS) para el Caso de Uso “Crear proyecto de emprendimiento” indicando el uso de memoria del Sistema y de datatypes, si corresponde. Utilice memoria para resolver este DSS.

**Problema 2 (35 puntos)**

Se desea modelar el funcionamiento de una plataforma que permita la digitalización de eventos físicos sucedidos en una organización.

La organización está compuesta por varias gerencias que poseen áreas asociadas en las cuales se registran eventos. De los eventos interesa registrar un id provisto por el sistema, un título, una descripción, la fecha de ocurrencia (ya que puede ser reportado a posteriori de sucedido) y la fecha de creación asignada por el sistema, así como el usuario que lo creó en el sistema y que usuario es responsable de darle seguimiento de lo sucedido. Un evento puede ser una observación o un incidente, en cuyo caso se registra la importancia de este.

Según los requerimientos planteados se realizó el siguiente Modelo de Dominio.

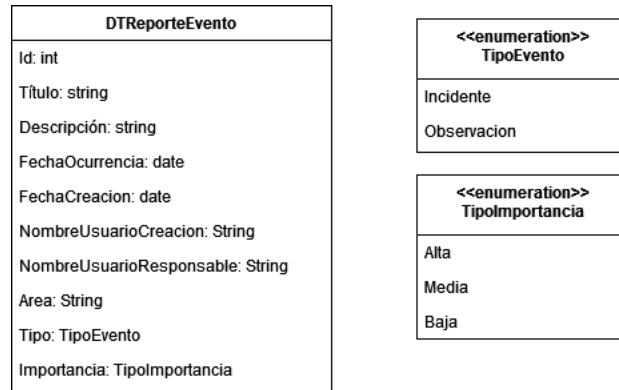


Restricciones:

- El nombre identifica la Gerencia
- El nombre identifica el Área
- La FechaOcurrencia es menor que la FechaCreacion
- El id identifica al Evento
- El nombre identifica al Usuario

Debido a que los gerentes les interesa extraer la información del sistema para analizar qué está sucediendo en las áreas que gestionan, se definió la siguiente operación del sistema:

<code>reporteEventos(String gerencia, Fecha desde, Fecha hasta, String area, TipoEvento tipo): Set(DTReporteEvento)</code>	
Descripción	Retorna el listado de los eventos sucedidos entre dos fechas para la gerencia, en una de sus áreas y para un tipo de evento.
Precondiciones	<ul style="list-style-type: none"> <li>• Existe una gerencia con nombre gerencia</li> <li>• Existe un área con nombre area dentro de dicha gerencia</li> <li>• desde es menor que hasta</li> </ul>
Postcondiciones	Devuelve un DTReporteEvento por cada evento tal que su fecha de ocurrencia esta entre desde y hasta, filtrando aquellos eventos que estén relacionados al área y sean del tipo indicado.

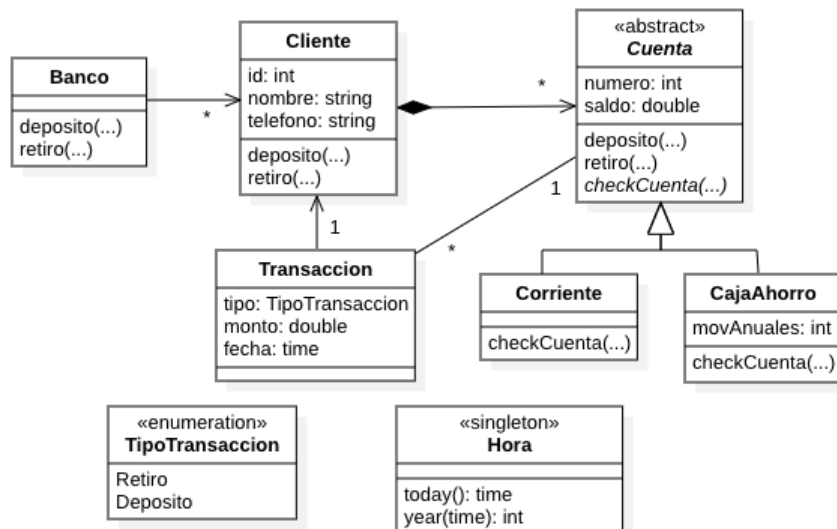


Se pide:

- Realizar el Diagrama de Comunicación correspondiente a la operación especificada. Las fechas pueden ser comparadas con los operadores básico (<, >, =, >=, <=, etc.) como cualquier tipo primitivo
- Realizar el Diagrama de Clases de Diseño (DCD) resultante del diagrama anterior.

**Problema 3 (35 puntos)**

El Diagrama de Clases de Diseño de la figura muestra el diseño parcial de un sistema bancario. Un banco tiene un conjunto de clientes que poseen cuentas, las cuales pueden ser cuentas corrientes o cajas de ahorro. El comportamiento refleja lo que sucede cuando los clientes realizan depósitos o retiros de sus cuentas, los cuales generan un registro de transacción y actualizan el saldo de la cuenta correspondiente. Las transacciones de retiro tienen restricciones dependiendo del tipo de cuenta, en particular: no se puede hacer un retiro si no hay saldo suficiente y, en caso de las cajas de ahorro, tienen un número máximo de transacciones anuales a realizar (de cualquier tipo). Para la manipulación de fechas, se definió un Singleton que manipula un tipo `time` (hora y fecha).



A continuación, se describen el comportamiento de las principales operaciones definidas.

**double Cuenta::deposito(double monto, Cliente\* cliente)**

Realiza un depósito en la cuenta, aumentando el saldo según el `monto` depositado. Además, crea la transacción correspondiente en su lista de transacciones con la fecha actual del sistema y la vincula con el `cliente` propietario de la cuenta. Retorna el saldo de la cuenta.

**double Cuenta::retiro(double monto, Cliente\* c)**

Realiza un retiro de la cuenta, disminuyendo el saldo según el **monto** retirado. Además, crea la transacción correspondiente en su lista de transacciones con la fecha actual del sistema y la vincula con el **cliente** propietario de la cuenta. Retorna el saldo de la cuenta. Utiliza la operación **checkCuenta** para verificar las restricciones específicas de cada tipo de cuenta. En caso de que no se cumpla alguna restricción, no se realiza el retiro y se lanza una excepción **std::runtime::error** según el tipo de restricción que se haya violado.

**bool Cuenta::checkCuenta()**

Verifica que se cumplan las restricciones específicas para cada tipo de cuenta. Retorna **true** si se cumplen las restricciones, o **false** si se violan.

**time Hora::today()**

Retorna la fecha actual del sistema.

**int Hora::year(time fecha)**

Dada una **fecha**, retorna el año correspondiente (ej.: “24/11/2022 14:55” retorna 2022).

#### Se pide:

- Implementar en C++ el .h del enumerado `TipoTransacción`.
- Implementar en C++ el .h de la clase `Hora` incluyendo constructor y destructor.
- Implementar en C++ el .h de las clases `Cuenta` y `CajaAhorro` incluyendo constructor y destructor.
- Implementar en C++ los métodos de las operaciones `Cuenta::deposito` y `Cuenta::retiro`.
- Implementar en C++ el método de la operación `CajaAhorro::checkCuenta`.

#### Observaciones:

- NO se pueden agregar más operaciones de las definidas en el diseño anterior.
- Puede utilizar colecciones genéricas (realizaciones de `IDictionary` e `ICollection`) o paramétricas (contenedores STL).
- No incluir directivas al precompilador (`#include`, etc).
- No es necesario implementar setters y getters adicionales de las clases, pero puede asumir su existencia si se requieren en algún método.
- La implementación debe hacer un correcto manejo de memoria.