

# Programación 4

EXAMEN JULIO 2021

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

Se desea modelar un sistema de gestión de cambios en el contexto de desarrollo de software. Este cuenta con usuarios de los cuales se conoce su nick (que los identifica), nombre y estado (activo o inactivo). Los usuarios se clasifican en dos categorías: cliente y desarrollador, no mutuamente excluyentes, es decir que un usuario puede utilizar el sistema tanto como cliente como desarrollador. Los clientes solicitan dos tipos distintos de tickets que pueden ser bugs o tareas para ser atendidas por desarrolladores. De los tickets se registra un código que los identifica, fecha y hora de su creación. De los bugs además se registra su fecha de resolución y para las tareas la descripción de requerimientos. Los bugs pueden ser resueltos por varios usuarios desarrolladores mientras que las tareas solo son asignadas a un desarrollador; el sistema registra quién es asignado a cada ticket. Además, todos los bugs resueltos deben tener un diagnóstico al que los desarrolladores que trabajen en él deberán asociar al menos un cambio. Como parte del diagnóstico interesa conocer el orden en que fueron realizados dichos cambios.

Además, se cuenta con la descripción del siguiente Caso de Uso:

Caso de Uso	Listar información de bugs resueltos por un desarrollador
Actor	Desarrollador
Descripción	El caso de uso comienza cuando un desarrollador desea obtener información de los bugs que resolvió. Para eso, ingresa su nick y a continuación el sistema lista los bugs que resolvió. Luego, el desarrollador selecciona los bugs de los cuales desea obtener información y finalmente el sistema devuelve el listado correspondiente. Para cada bug se lista su diagnóstico y el conjunto de cambios realizados.

**Se pide:**

- 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 “Listar información de bugs resueltos por un desarrollador”, indicando el uso de memoria del Sistema y de datatypes si corresponde.

# Programación 4

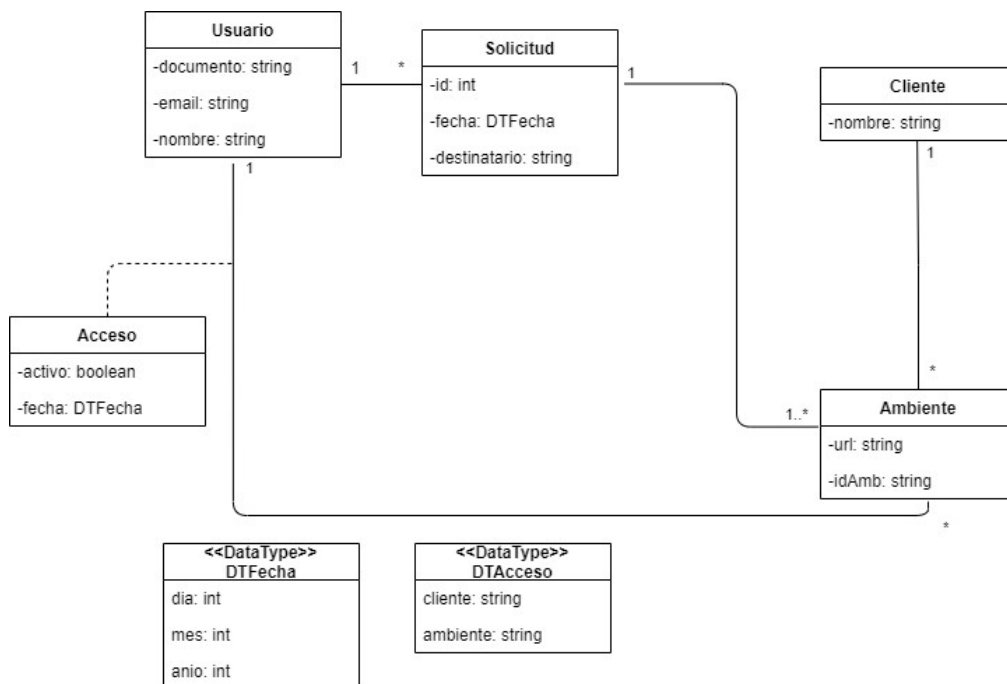
EXAMEN JULIO 2021

## Problema 2 (35 puntos)

En una empresa de desarrollo de software, un sistema de gestión de ambientes permite al equipo tener control sobre el acceso a los ambientes. Un ambiente tiene su propia url, su propia base de datos y normalmente la versión de la aplicación que ejecuta en ese ambiente es diferente a la de otros ambientes. Para un mismo cliente pueden existir varios ambientes, por ejemplo, el ambiente de desarrollo, de testing, de pruebas de usuario y de producción, entre otros. Distintos tipos de usuarios suelen tener distintos tipos de accesos, por ejemplo, los testers pueden acceder y modificar el ambiente de testing, pero solo pueden acceder a consultar el ambiente de pruebas de usuario. Para acceder a los ambientes los usuarios deben generar una solicitud de acceso que será evaluada por el equipo de operaciones.

### Parte I:

Considere el siguiente modelo de dominio para un sistema de gestión de acceso a ambientes, donde documento identifica al Usuario, nombre identifica al Cliente, id identifica a la Solicitud e idAmb identifica el Ambiente.



Por otra parte, se cuenta con el contrato de la siguiente operación del sistema:

listarAccesos(doc : string): Set (DTAcceso)	
Descripción:	Lista los ambientes a los que un usuario dado tiene acceso activo.
Parámetros:	doc: Identificador del usuario.
Precondiciones:	Existe en el sistema un Usuario con documento doc.
Poscondiciones:	Se crea un conjunto de instancias DTAcceso, donde cada elemento corresponde a una instancia de Acceso asociada al Usuario, e incluye el nombre del Cliente al que pertenece el ambiente y la url del Ambiente.

**Se pide:**

- a) Realizar el Diagrama de Comunicación correspondiente a la operación `listarAccesos`, **incluyendo visibilidades**.
- b) Realizar el Diagrama de Clases de Diseño, incluyendo controladores y datatypes si corresponde. No incorporar setters ni getters. Incorporar sólo los constructores y destructores que se utilicen en los diagramas de comunicación.

Parte II:

Las solicitudes de acceso pueden estar pendientes, aprobadas o caducadas. Inicialmente una solicitud está pendiente y una vez evaluada pasa a estar aprobada. Luego de pasado un periodo de tiempo, una solicitud aprobada pasa a estar caducada. Por otra parte, la operación `enviarNotificacion()` envía un correo electrónico con el mensaje “Tiene una solicitud pendiente” al destinatario de la solicitud si esta está pendiente, envía un correo electrónico con el mensaje “Su solicitud fue aprobada” al usuario que solicitó el acceso si esta está aprobada, mientras que si está caducada no hace nada.

**Se pide:**

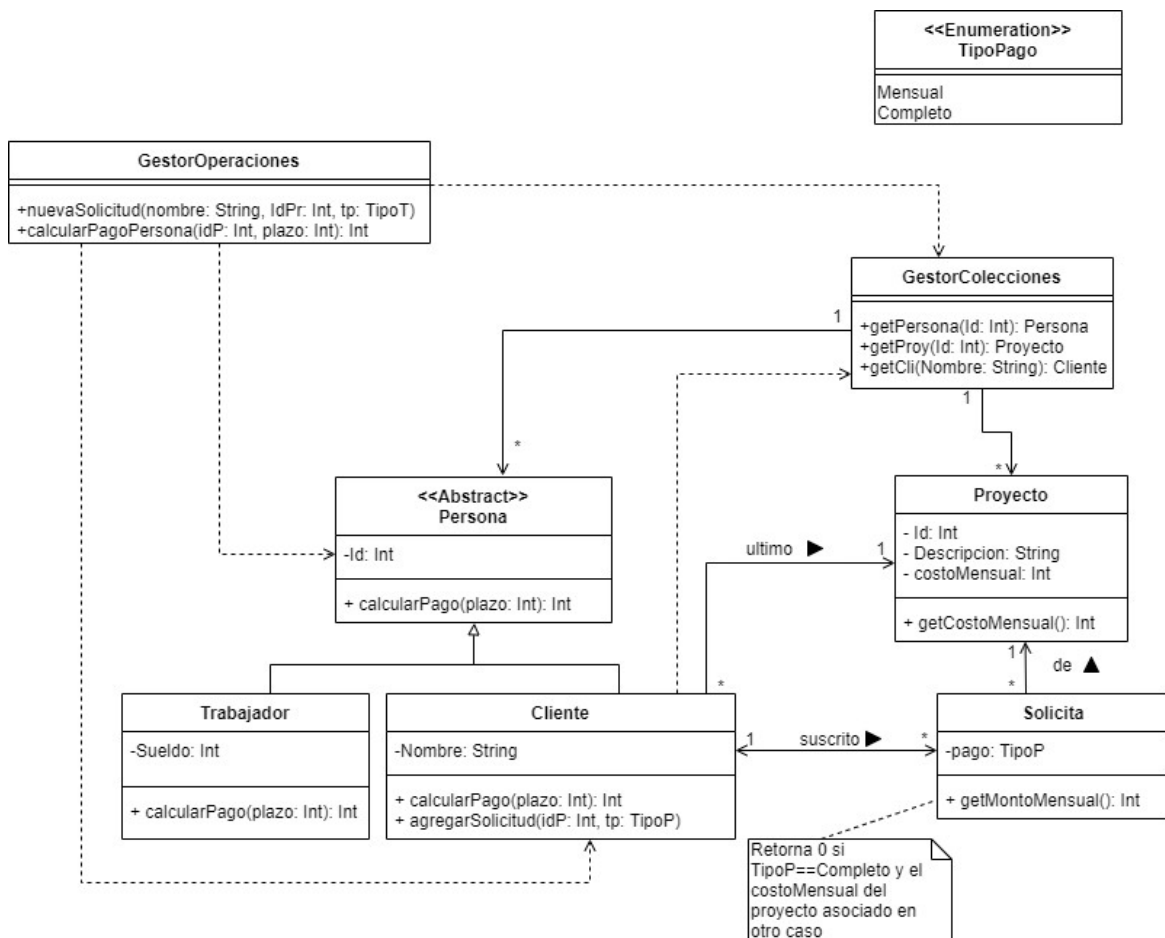
- c) Realizar el Diagrama de Clases de Diseño para modelar la situación descrita anteriormente, utilizando un patrón de diseño. Indicar el nombre del patrón, los roles de las clases participantes y los seudocódigos de las operaciones del patrón. Asuma que dispone de la siguiente operación:  
`enviarMail(string destinatario, string texto).`

# Programación 4

EXAMEN JULIO 2021

## Problema 3 (35 puntos)

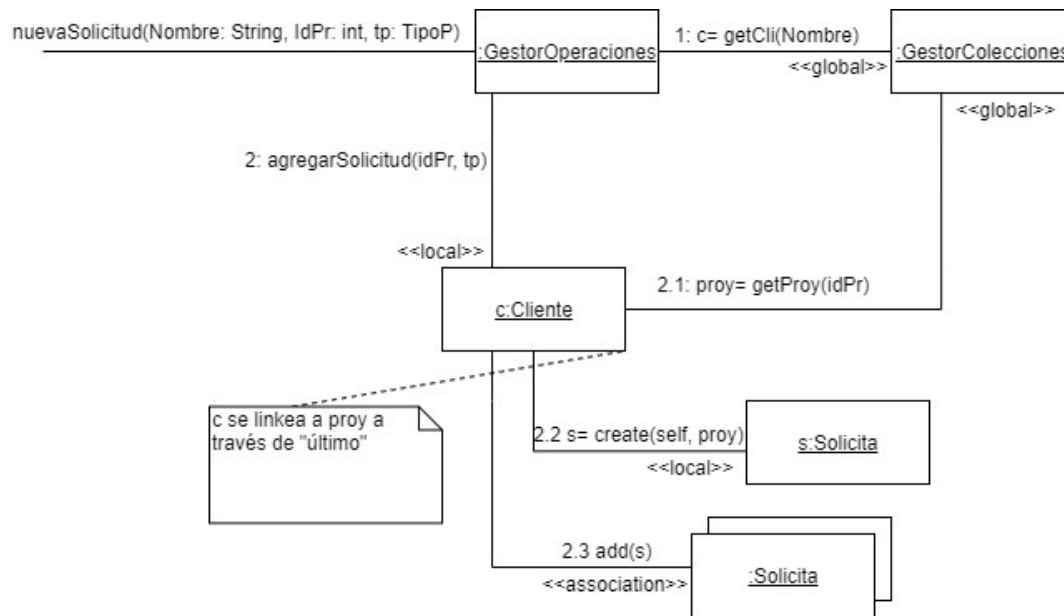
Se desea desarrollar un sistema para gestionar los proyectos de la empresa Altamira S.A. que se dedica a gestionar proyectos. La empresa cuenta con un grupo predefinido de proyectos “modelo” (identificados por una id) y los clientes (que se identifican por nombre) solicitan alguno de ellos. Los precios del servicio de gestión de proyectos dependen del formato de pago. En la actualidad se brindan dos métodos: un pago mensual en el que cada mes se computa un monto fijo (dependiendo del proyecto) y otro total en el que se cobra un valor estimado al inicio. El estimado se calcula como el equivalente a ocho meses. Por otro lado, hay un segundo grupo de personas, los trabajadores de la empresa, de los que interesa saber su sueldo. Las personas en general (tanto trabajadores como clientes) se identifican por una id interna del sistema. En la siguiente figura se muestra el Diagrama de Clases de Diseño (DCD) elaborado por el ex-responsable de Diseño, quien fue despedido recientemente.



Se pide:

- Los responsables del equipo de desarrollo afirman que el DCD es incorrecto porque las clases `GestorOperaciones` y `GestorColecciones` deben ser Singleton ¿Es correcta esa afirmación? Explique **brevemente** cuál(es) de ellas cree que debe(n) serlo, cuál(es) no y por qué.

En la siguiente figura se muestra un Diagrama de Comunicación de la operación nuevaSolicitud.



### Se pide:

- b) De acuerdo a dicho diagrama implemente la(s) operación(es) de la clase Cliente que aparece(n) allí.
- c) Implemente los módulos GestorOperaciones.h, Persona.h y Trabajador.h. Implemente además la operación calcularPago(int plazo) en los .cpp de Trabajador y Cliente. Todas sus implementaciones deben ser coherentes con lo hecho en las partes anteriores.

calcularPago retorna, en caso de ser un trabajador, el dinero que se le debe pagar por el plazo que se da como parámetro (en meses) y en caso de que sea cliente el monto que este debe. El dinero a pagar al trabajador se calcula como el sueldo multiplicado por la cantidad de meses. La deuda del cliente se calcula como lo que debe pagar mensualmente por cada proyecto solicitado por el plazo en meses; los proyectos pagados completos retornan como costo mensual 0.

### Observaciones:

- Puede utilizar colecciones genéricas (realizaciones de IDictionary e ICollection) o paramétricas (contenedores STL).
- No incluir directivas al precompilador (#include, etc).