



## PRIMER PARCIAL

**Asignatura:** Programación Orientada a Objetos

### Ejercicio 1: UML y Conceptos generales

Se solicita modelar una clase **CuentaUsuario**, cuyos atributos deben incluir (al menos) información de nombre de usuario (string), email (string), contraseña (entero), estado de sesión (boolean, valor por defecto False) y un registro de operaciones (lista, cada operación se guarda como un string). Todas las operaciones realizadas en una cuenta deben ser registradas en la lista.

Implementar la clase teniendo en cuenta los siguientes requisitos y métodos:

- a) Constructor: debe recibir los parámetros correspondientes al nombre de usuario, email y contraseña. Si el email no contiene el símbolo '@', se permite crear la instancia pero se considera inválido y no es posible realizar operaciones que requieran su uso.
- b) cambiarContraseña(contraseñaActual: entero, nuevaContraseña: entero): permite actualizar la contraseña, para lo cual se requiere ingresar la contraseña actual y comprobar si es correcta.
- c) consultarDatos(): retorna un diccionario con el nombre de usuario y el correo electrónico del usuario. No debe retornar la contraseña.
- d) mostrarRegistroOperaciones(): muestra por pantalla cada una de las operaciones realizadas.
- e) enviarMensaje(emailDestino: str, mensaje: str): simula la operación de enviar un mensaje a una cuenta de correo electrónico. No es posible enviar un mensaje si la dirección de correo del usuario es inválida.
- f) iniciarSesion(<parametros>): debe permitir el inicio de sesión del usuario. Incluir los parámetros que se consideren necesarios.
- g) cerrarSesion(<parametros>): debe permitir el cierre de sesión del usuario. Incluir los parámetros que se consideren necesarios.
- h) Proponer e implementar un método adicional, a elección, para modelar otra operación extra.

Completar la implementación con los métodos de consulta y modificación necesarios, teniendo en cuenta que no afecten la funcionalidad de los métodos anteriores.

En caso de considerarlo necesario, está permitido agregar nuevos atributos o métodos adicionales.

Realizar el diagrama UML de la clase completa (usar el nombre de archivo *PrimerParcial\_Ejercicio1* e incluir un elemento de Nota con el nombre y apellido; puede realizarse en drawio o en papel)

*Ejemplo de salida de mostrarRegistroOperaciones:*

*Se creó una cuenta de usuario para Pedro*

*El email ingresado no es válido (pedro\*gmail.com)*

*Sesión iniciada*

*No se puede enviar un mensaje porque el email no es válido*

*Se modificó el email a pedro@gmail.com*

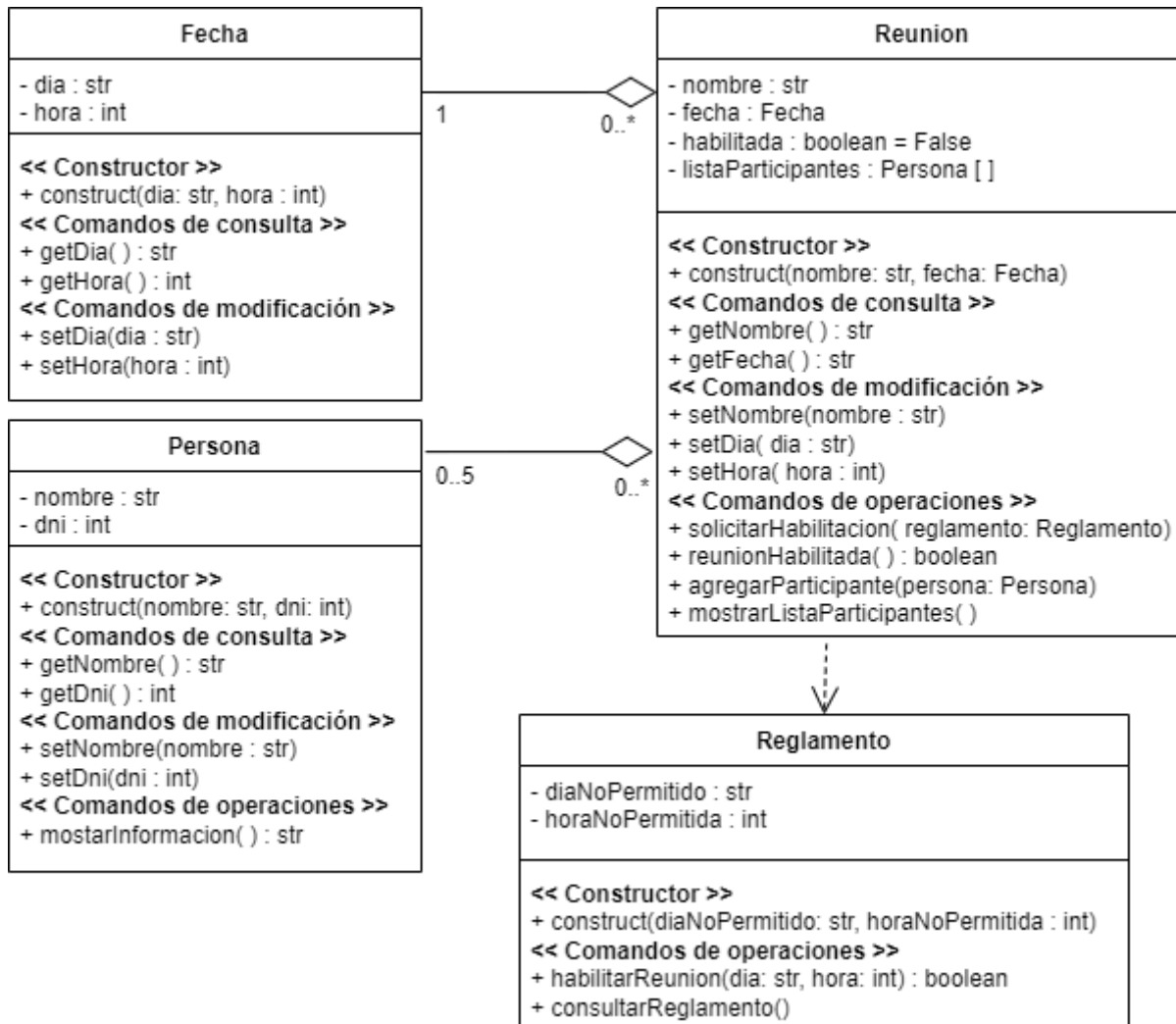
*Se envió un mensaje a ana@mail.com*

*No se pudo modificar la contraseña porque no coincide la contraseña actual*

*Se modificó la contraseña correctamente*

## Ejercicio 2: Relaciones entre clases

Se tiene un modelo de clases representado en el diagrama UML.



Realizar la implementación de acuerdo al diagrama y teniendo en cuenta que:

- El sistema modelado es un software para agendar reuniones dentro de una oficina.
- La fecha se modela con el nombre del día y la hora (sin minutos).
- El método mostrarInformacion de la clase Persona devuelve una cadena con el nombre y dni de la persona.
- En una reunión no se pueden agregar participantes que tengan el mismo dni.
- El método consultarReglamento imprime en pantalla un mensaje con el siguiente formato de ejemplo: "No se permiten reuniones los días lunes, ni reuniones que comiencen a las 5 horas"
- El método reunionHabilitada devuelve un valor booleano dependiendo si la reunión está o no habilitada.

**IMPORTANTE:** no es obligatoria la implementación de los métodos de consulta y modificación que no estén indicados en el diagrama, a menos que sean necesarios para completar la implementación de otras funcionalidades.