

Todos los actos estarán regidos por **la buena fe, la transparencia y la responsabilidad** por parte de las y los estudiantes, docentes, nodocentes y del equipo de gestión.

IDE: Eclipse

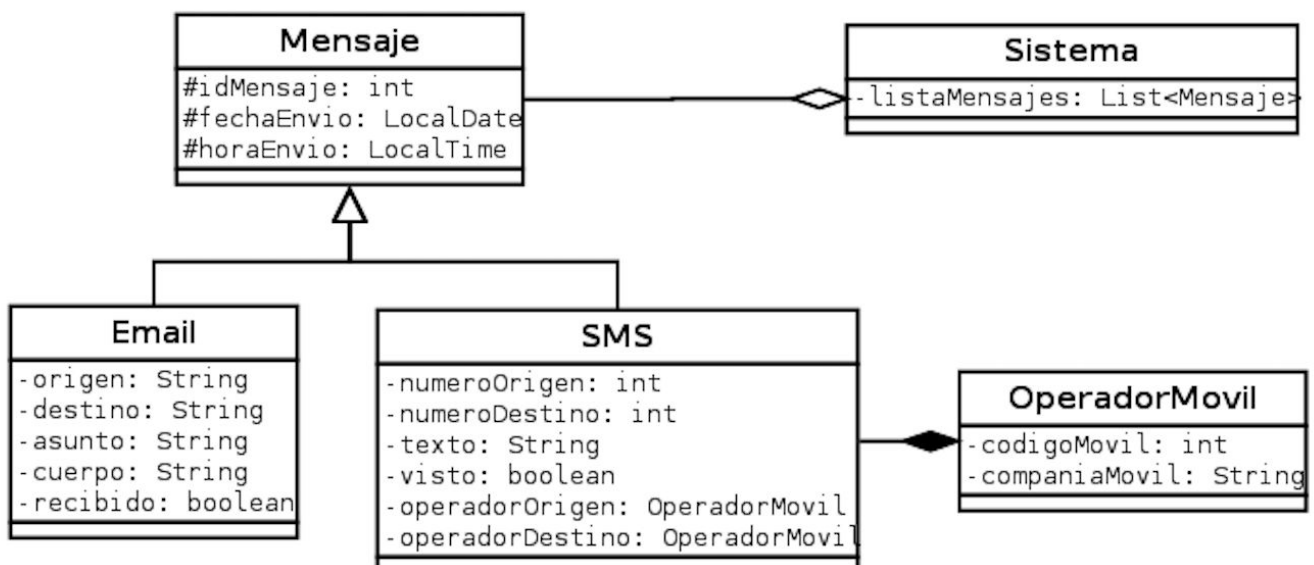
Lenguaje de programación: Java

Proyecto: Crear una carpeta /tuApellidoNombre/tuNroDni (el nombre del proyecto es tu DNI)

- **Entrega:** Aula Virtual subir la carpeta comprimida con tuApellidoNombre

Sistema de Mensajería

Nota: El examen acredita puntos por la resolución del modelo y test de cada CU.



Capa modelo:

Clase Mensaje:

Superclase abstracta

Atributos:

-int idMensaje //Identificador único autonumérico, se tiene que calcular de forma automática cada vez que se agrega un elemento a lista de mensajes

-LocalDate fechaEnvio

-LocalTime horaEnvio

Metodos:

~ cambiarDeEstado(boolean estadoNuevo) //método abstracto

Clase SMS:

Subclase de Mensaje

Atributos:

-int numeroOrigen

-int numeroDestino //Se tiene que lanzar una excepción en caso de que no tenga una longitud de 10 números. Para saber la longitud en dígitos de un int se puede usar el siguiente código:

String.valueOf(numeroEntero).length()

-String texto

- boolean visto
- OperadorMovil operadorOrigen
- OperadorMovil operadorDestino

Metodos:

+cambiarDeEstado(boolean estadoNuevo) //Guardar el valor recibido en estadoNuevo en el atributo visto.

Clase Email:

Subclase de Mensaje

Atributos:

- String origen
- String destino //Se tiene que lanzar una excepción en caso de que no contenga un arroba. Para saber si un String contiene dentro un char se puede usar el siguiente código:
unString.indexOf(unChar) //devuelve la posición del char en el String o un -1 en caso de no encontrarlo
- String asunto
- String cuerpo
- boolean recibido

Metodos:

+cambiarDeEstado(boolean estadoNuevo) //Guardar el valor recibido en estadoNuevo en el atributo recibido.

Clase OperadorMovil:

Atributos:

- int codigoMovil
- String companiaMovil

Clase Sistema

Atributo: List<Mensaje> listaMensajes

Metodos:

- + agregarEmail(LocalDate fechaEnvio, LocalTime horaEnvio, String origen, String destino, String asunto, String cuerpo) : boolean //El atributo recibido se inicializará por defecto en falso.
- + agregarSMS(LocalDate fechaEnvio, LocalTime horaEnvio, int numeroOrigen, int numeroDestino, String texto, OperadorMovil operadorOrigen, OperadorMovil operadorDestino) : boolean //El atributo visto se inicializará por defecto en falso.
- + traerMensaje(int idMensaje) : Mensaje
- + traerMensaje(String nombreCompania) : Mensaje //retorna una lista del tipo Mensaje con aquellos que tengan como operadorOrigen el nombre de la compañía igual al que ingresa como parámetro.
- + traerMensaje(LocalDate fecha, LocalTime horaDesde, LocalTime horaHasta) : Mensaje //retorna una lista del tipo Mensaje con aquellos que tengan fecha y hora dentro de los parámetros ingresados.
- + traerMensaje(LocalDate fecha, LocalTime horaDesde, LocalTime horaHasta, String codigoCompania) : Mensaje //retorna una lista del tipo Mensaje con aquellos que tengan fecha y hora dentro de los parámetros ingresados y además contentan operador origen de código compañía con la compañía que ingresa como parámetro.

Capa test:

Nota: Al comenzar cada test indicar el número a resolver ej: `System.out.println("1-3")`; y luego la implementación del mismo.

Test.java

1-1) Intentar agregar a la lista el objeto SMS: fechaEnvio=11/12/2020, horaEnvio=17:37, numeroOrigen=1523456780, numeroDestino=156543210, texto="Voy a llegar más tarde", operadorOrigen=[OperadorMovil: codigoMovil: 1, companiaMovil="Personal"], operadorDestino=[OperadorMovil: codigoMovil: 2, companiaMovil="Claro"]

1-2) Agregar a la lista e imprimir el objeto SMS: fechaEnvio=11/12/2020, horaEnvio=17:37, numeroOrigen=1523456780, numeroDestino=1565432109, texto="Voy a llegar más tarde", operadorOrigen=[OperadorMovil: codigoMovil: 1, companiaMovil="Personal"], operadorDestino=[OperadorMovil: codigoMovil: 2, companiaMovil="Claro"]

1-3) Intentar agregar a la lista el objeto Email: fechaEnvio=13/12/2020, horaEnvio=15:16, origen="informacion@unla.gob.ar", destino="alumno.com", asunto="Información - UNLa", cuerpo="Estimado alumno, le enviamos el protocolo para rendir el examen final virtual"

1-4) Agregar a la lista e imprimir el objeto Email: fechaEnvio=13/12/2020, horaEnvio=15:16, origen="informacion@unla.gob.ar", destino="alumno@gmail.com", asunto="Información - UNLa", cuerpo="Estimado alumno, le enviamos el protocolo para rendir el examen final virtual"

2-1) Traer e imprimir el SMS con IdMensaje = 1 y cambiar el estado a Visto=true

2-2) Traer e imprimir el Email con IdMensaje = 2 y cambiar el estado a Recibido=true

3) Agregar los siguientes objetos a la lista, al finalizar la carga, imprimirla:

SMS: fechaEnvio=15/11/2020, horaEnvio=12:03, numeroOrigen=1523456780, numeroDestino=1598760883, texto="Recibido OK",

operadorOrigen=[OperadorMovil: codigoMovil: 3, companiaMovil="Movistar"],

operadorDestino=[OperadorMovil: codigoMovil: 2, companiaMovil="Claro"]

SMS: fechaEnvio=10/12/2020, horaEnvio=13:16, numeroOrigen=1523456780, numeroDestino=1598768250, texto="Gracias"

operadorOrigen=[OperadorMovil: codigoMovil: 1, companiaMovil="Personal"],

operadorDestino=[OperadorMovil: codigoMovil: 3, companiaMovil="Movistar"]

Email: fechaEnvio=08/11/2020, horaEnvio=17:15, origen="oo1@unla.gob.ar",

destino="alumno@gmail.com.ar", asunto="Orientación a Objetos 1 - UNLa", cuerpo="Se adjunta la grilla de inscripciones al final de Diciembre"

Email: fechaEnvio=10/12/2020, horaEnvio=08:19, origen="informacion@gmail.com.ar",

destino="cliente@gmail.com.ar", asunto="Ofertas", cuerpo="Aproveche nuestras Ofertas"

4-1) Traer e imprimir los mensajes que tengan como operadorOrigen la compañía "Personal"

4-2) Traer e imprimir los mensajes que se hayan enviado el día 10/12/2020 entre las 8 y las 14Hs.

4-3) Traer e imprimir los mensajes que se hayan enviado el día 15/12/2020 entre las 12 y las 13Hs y que tengan como compañía origen "Movistar"