

Orientación a Objetos I

Explicación de práctica
Semana del 2 de Octubre



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Anuncios

- Última clase de explicación de práctica... presencial
- A partir del lunes 9/10, las explicaciones de práctica serán virtuales. (Comunicaremos detalles en el foro).



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Actividades de esta semana

Los ejercicios de la semana:

- Ejercicio 13: Cliente de correo con adjuntos
- Ejercicio 14: Intervalo de tiempo



FACULTAD DE INFORMATICA



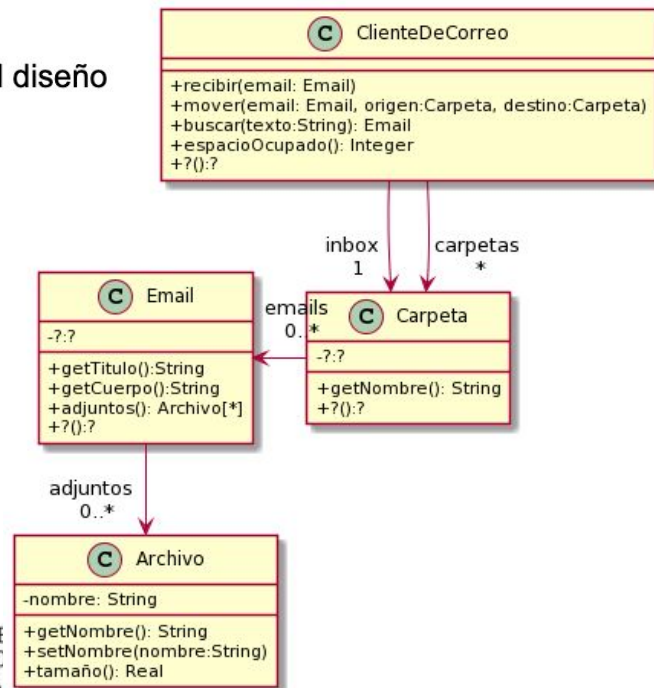
UNIVERSIDAD
NACIONAL
DE LA PLATA

Cliente de correo

Ejercicio 13. Cliente de correo con adjuntos

El diagrama de clases de UML que se muestra a continuación documenta parte del diseño simplificado de un cliente de correo electrónico.

- En respuesta al mensaje #recibir, almacena en el inbox (una de las carpetas) el email que recibe como parámetro.
- En respuesta al mensaje #mover, mueve el email que viene como parámetro de la carpeta origen a la carpeta destino (asuma que el email está en la carpeta origen cuando se recibe este mensaje).
- En respuesta al mensaje #buscar retorna el primer email que encuentra cuyo título o cuerpo contienen el texto indicado como parámetro. Busca en todas las carpetas.
- En respuesta al mensaje #espacioOcupado, retorna la suma del espacio ocupado por todos los emails de todas las carpetas.
- El tamaño de un email es la suma del largo del título, el largo del cuerpo, y del tamaño de sus adjuntos.
- Para simplificar, asuma que el tamaño de un archivo es el largo de su nombre.



FACULTAD DE INFORMÁTICA



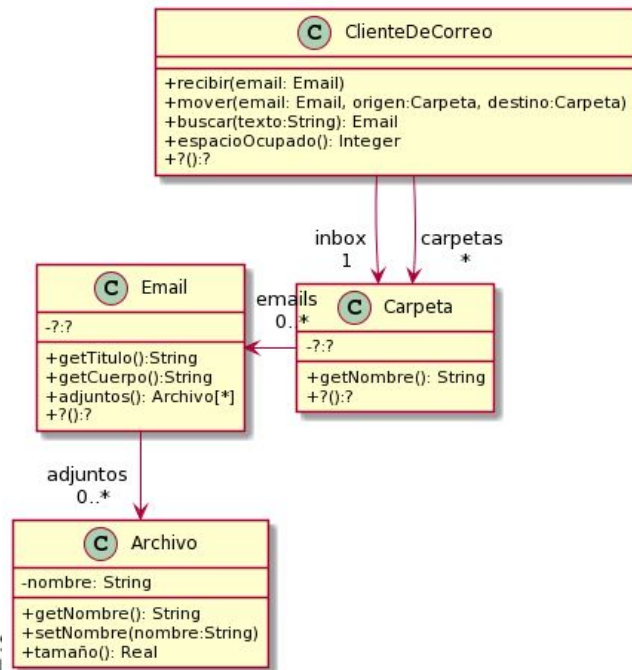
UNIVE
NACIO
DE LA

Cliente de correo

Delegar en la carpeta!!!

```
recibir(Email email){  
    this.inbox.agregarCorreo(email)  
}
```

```
mover (email, origen, destino){  
    origen.eliminarCorreo(email)  
    destino.agregarCorreo(email)  
}
```



FACULTAD DE INFORMATICA

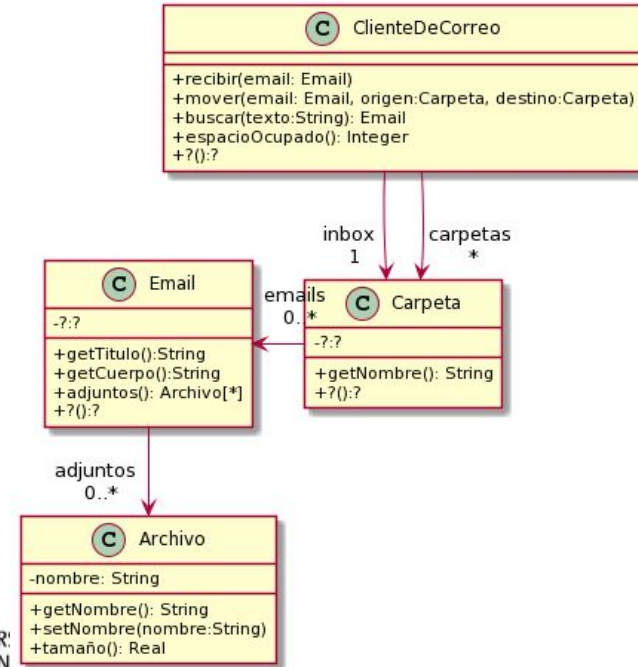


UNIVERSIDAD
NACIONAL
DE LA PLATA

Ciente de correo

Buscar: cómo buscamos en todas las carpetas?

- Podemos buscar en inbox y después buscamos en carpetas.
- Si tuviéramos inbox, spam, trash y carpetas que pasaría?
- Ideas para simplificar la solución?



FACULTAD DE INFORMÁTICA



UNIVERSIDAD NACIONAL DE LA PLATA

Cliente de correo

API Stream **SUM**

Recursos

¿Cómo preg

Trabajando

Criterios y h

Apunte sobr

Cheatsheet

Ejercicios

Cuadernillo

Retorna la suma de los elementos de un stream de números (DoubleStream, IntStream...). Ej: calcular cuántos exámenes se tomaron en total a todos los alumnos, en un año determinado.

```
public int totalExamenesTomadosEn(int anio){  
    return alumnos.stream()  
        .mapToInt(alumno->alumno.cantidadExamenesRendidos(anio))  
        .sum();  
}
```

los elementos de un stream de ream, IntStream...). Ej: calcular cuántos on en total a todos los alumnos, en un

```
enesTomadosEn(int anio){  
    am()  
    alumno.cantidadExamenesRendidos(anio))
```

lio de los elementos de un stream de ream, IntStream...).
ino, obtener el promedio de un alumno
promedio(){
 i.stream()
 ble(examen->examen.getNota())
 }.orElse(0);

os de un stream de acuerdo a la
e como parámetro.
ino, obtener los exámenes ordenados
ascendente
1> examensOrdenadosPorFechaAsc(){
 itream()
 . ex2) ->
 cha().compareTo(ex2.getFecha())
 cors.toList();

ANY
r alumno cuyo nombre comience con
si no existe ninguno, obtiene null.
serAlumnoCon(String x){
 itream()
 +alumno.getNombre().startsWith(x))
 ise(null);



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Cliente de correo

Espacio ocupado: No reinventamos la rueda

```
cliente.espacioOcupado(){  
    return carpetas.stream().mapToInt(carpetas -> carpetas.espacioOcupado()).sum();  
}
```

```
carpetas.espacioOcupado(){  
    return emails.stream().mapToInt(email -> email.espacioOcupado()).sum();  
}
```



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Intervalo de tiempo

Ejercicio 14. Intervalo de tiempo

En Java, las fechas se representan normalmente con instancias de la clase `java.time.LocalDate` (<https://docs.oracle.com/javase/8/docs/api/java/time/LocalDate.html>). Se pueden crear con varios métodos "static" como por ejemplo `LocalDate.now()` .

- Investigue cómo hacer para crear una fecha determinada, por ejemplo 15/09/1972.
- Investigue cómo hacer para determinar si la fecha de hoy se encuentra entre las fechas 15/12/1972 y 15/12/2032. Sugerencia: vea los métodos permiten comparar `LocalDates` y que retornan `booleans`.
- Investigue cómo hacer para calcular el número de días entre dos fechas. Lo mismo para el número de meses y de años Sugerencia: vea el método `until`.

Tenga en cuenta que los métodos de `LocalDate` colaboran con otros objetos que están definidos a partir de enums, clases e interfaces de `java.time`; por ejemplo `java.time.temporal.ChronoUnit.DAYS`



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Intervalo de tiempo

static LocalDate

of(int year, int month, int dayOfMonth)

Obtains an instance of `LocalDate` from a year, month and day.

LocalDate

plusDays(long daysToAdd)

Returns a copy of this `LocalDate` with the specified number of days added.

boolean

isAfter(**ChronoLocalDate** other)

Checks if this date is after the specified date.

boolean

isBefore(**ChronoLocalDate** other)

Checks if this date is before the specified date.

boolean

isEqual(**ChronoLocalDate** other)

Checks if this date is equal to the specified date.



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Intervalo de tiempo

Tarea 1

Implemente la clase DateLapse (Lapso de tiempo). Un objeto DateLapse representa el lapso de tiempo entre dos fechas determinadas. La primera fecha se conoce como "from" y la segunda como "to". Una instancia de esta clase entiende los mensajes:

```
public LocalDate getFrom()
```

"Retorna la fecha de inicio del rango"

```
public LocalDate getTo()
```

"Retorna la fecha de fin del rango"

```
public int sizeInDays()
```

"retorna la cantidad de días entre la fecha 'from' y la fecha 'to'"

```
public boolean includesDate(other: LocalDate)
```

"recibe un objeto LocalDate y retorna true si la fecha está entre el from y el to del receptor y false en caso contrario".



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Intervalo de tiempo

Tarea 2

Escriba tests de unidad para asegurarse de que los métodos implementados funcionan adecuadamente.

Tarea 3

Asumiendo que implementó la clase `DateLapse` con dos variables de instancia “from” y “to”, realice otra implementación de la clase para que su representación sea a través de los atributos “from” y “sizeInDays” y coloquela en otro paquete. Es decir, debe basar su nueva implementación en estas variables de instancia solamente. Intente definir una interfaz java para que ambas soluciones la implementen.

Los cambios en la estructura interna de un objeto sólo deben afectar a la implementación de sus métodos. Estos cambios deben ser transparentes para quien le envía mensajes, no debe notar ningún cambio y seguir usándolo de la misma forma. Por lo tanto, los tests que implementó en la tarea 2 deberían pasar sin problemas.



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA

Foros de consulta

Cómo preguntar en el foro

Breve guía para poder sacar el mejor provecho al foro y a la convivencia a través de las preguntas y respuestas.

Cómo preguntar en el foro

Antes de Preguntar: Busca una respuesta por tus propios medios

Elegí el foro específico

Elegí un título apropiado para la pregunta

No envíes una solución para que la corrijan

Describir qué estás intentando hacer

Describir el problema y lo que has intentado para resolverlo

Escribir claro

No solicites respuestas a tu correo

Si no entendés la respuesta

Terminá con una breve nota de conclusión.

Evitá el “Me sumo al pedido”

Foros



Ejercicio 13 - Cliente de correos con adjuntos



Ejercicio 14 - Intervalo de Tiempo



FACULTAD DE INFORMATICA



UNIVERSIDAD
NACIONAL
DE LA PLATA