

# Orientación a Objetos I

Explicación de práctica  
Semana del 18 de septiembre



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Temario

- Actividades de la semana anterior
- Actividades a realizar esta semana



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Actividades de la semana anterior

- Ejercicio 5: Genealogía Salvaje
- Ejercicio 6: Red de Alumbrado
- Ejercicio 7: Method lookup con Empleados

## Cuadernillo Semestral de Actividades

Actualizado: 14 de septiembre de 2023

El presente cuadernillo posee un compilado con todos los ejercicios que se usarán durante el semestre en la asignatura. Los ejercicios están organizados en forma secuencial, siguiendo los contenidos que se van viendo en la materia.

Cada semana les indicaremos cuales son los ejercicios en los que deberían enfocarse para estar al día y algunos de ellos serán discutidos en la explicación de práctica.

### Recomendación importante:

Los contenidos de la materia se incorporan y fijan mejor cuando uno intenta aplicarlos - **no alcanza con ver un ejercicio resuelto por alguien más**. Para sacar el máximo provecho de los ejercicios, es importante que asistan a las consultas de práctica habiendo intentado resolverlos (tanto como les sea posible). De esa manera podrán hacer consultas más enfocadas y el docente podrá darles mejor feedback.



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Actividades esperadas para esta semana

- Ejercicio 8: Distribuidora Eléctrica
- Ejercicio 9: Cuenta con ganchos
- Ejercicio 10: Job Scheduler

## Cuadernillo Semestral de Actividades

Actualizado: 14 de septiembre de 2023

El presente cuadernillo posee un compilado con todos los ejercicios que se usarán durante el semestre en la asignatura. Los ejercicios están organizados en forma secuencial, siguiendo los contenidos que se van viendo en la materia.

Cada semana les indicaremos cuales son los ejercicios en los que deberían enfocarse para estar al día y algunos de ellos serán discutidos en la explicación de práctica.

### Recomendación importante:

Los contenidos de la materia se incorporan y fijan mejor cuando uno intenta aplicarlos - **no alcanza con ver un ejercicio resuelto por alguien más**. Para sacar el máximo provecho de los ejercicios, es importante que asistan a las consultas de práctica habiendo intentado resolverlos (tanto como les sea posible). De esa manera podrán hacer consultas más enfocadas y el docente podrá darles mejor feedback.



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 8: Distribuidora Eléctrica

## Ejercicios de la semana:

- **Ejercicio 8: Distribuidora Eléctrica**
- Ejercicio 9: Cuenta con ganchos
- Ejercicio 10: Job Scheduler

## Ejercicio 8: Distribuidora Eléctrica

Una distribuidora eléctrica desea un sistema para el registro de los consumos de sus usuarios y para la emisión de facturas de cobro.

El sistema permite registrar usuarios, para los cuales se indica nombre y dirección. Por simplificación, un usuario puede estar relacionado con un solo domicilio (para el que se registran los consumos).

El sistema permite registrar los consumos para los usuarios. Los consumos que se registran para los usuarios tienen dos componentes: el consumo de energía activa y el consumo de energía reactiva.

Una vez al mes, la empresa distribuidora realiza el proceso de facturación. Por cada usuario se emite una factura (el proceso completo retorna una colección).



FACULTAD DE INFORMATICA

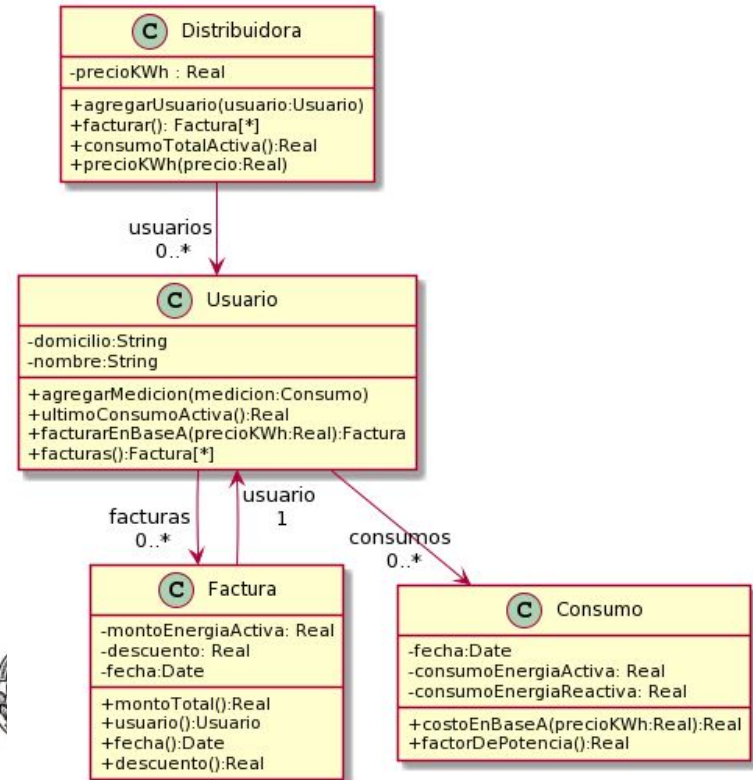


UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 8: Distribuidora Eléctrica

Ejercicios de la semana:

- **Ejercicio 8: Distribuidora Eléctrica**
- Ejercicio 9: Cuenta con ganchos
- Ejercicio 10: Job Scheduler



FACULTAD DE INFORMÁTICA

# Ejercicio 8: Distribuidora Eléctrica

Ejercicios de la semana:

- **Ejercicio 8: Distribuidora Eléctrica**
- Ejercicio 9: Cuenta con ganchos
- Ejercicio 10: Job Scheduler

Recursos adicionales:

- Tests
- Implementación de referencia



FACULTAD DE INFORMATICA



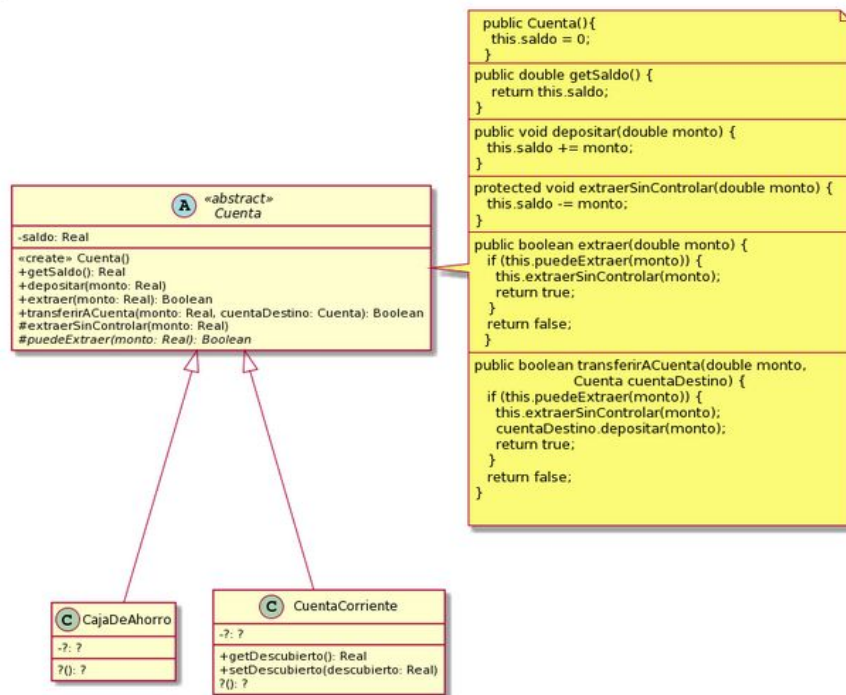
UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 9: Cuenta con ganchos

## Ejercicios de la semana:

- Ejercicio 8: Distribuidora Eléctrica
- **Ejercicio 9: Cuenta con ganchos**
- Ejercicio 10: Job Scheduler

Observe con detenimiento el diseño que se muestra en el siguiente diagrama. La clase cuenta es *abstracta*. El método `puedeExtraer()` es abstracto. Las clases `CajaDeAhorro` y `CuentaCorriente` son concretas y están incompletas.



FACULTAD DE INFORM



# Ejercicio 9: Cuenta con ganchos

## Ejercicios de la semana:

- Ejercicio 8: Distribuidora Eléctric
- **Ejercicio 9: Cuenta con ganchos**
- Ejercicio 10: Job Scheduler

**Tarea A:** Complete la implementación de las clases CajaDeAhorro y CuentaCorriente para que se puedan efectuar depósitos, extracciones y transferencias teniendo en cuenta los siguientes criterios.

- 1) Las **cajas de ahorro** solo pueden extraer y transferir cuando cuentan con fondos suficientes.
- 2) Las extracciones, los depósitos y las transferencias desde **cajas de ahorro** tienen un costo adicional de 2% del monto en cuestión (téngalo en cuenta antes de permitir una extracción o transferencia desde caja de ahorro).
- 3) Las **cuentas corrientes** pueden extraer aún cuando el saldo de la cuenta sea insuficiente. Sin embargo, no deben superar cierto límite por debajo del saldo. Dicho límite se conoce como límite de descubierto (algo así como el máximo saldo negativo permitido). Ese límite es diferente para cada cuenta (lo negocia el cliente con la gente del banco).
- 4) Cuando se abre una **cuenta corriente**, su límite descubierto es 0 (no olvide definir el constructor por default).



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 9: Cuenta con ganchos

Ejercicios de la semana:

Caja Ahorro  
\$ 400

Cuenta Corriente  
\$100  
límite: \$500

- |  |                  |   |   |
|--|------------------|---|---|
| - Ejercicio 8: Distribuidora Eléctrica   |                  |   |   |
| - <b>Ejercicio 9: Cuenta con ganchos</b> | depositar(200) ? | ? | ? |
|  | extraer(\$100) ? | ? | ? |
| - Ejercicio 10: Job Scheduler            | extraer(\$400) ? | ? | ? |



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 9: Cuenta con ganchos

Ejercicios de la semana:

- Ejercicio 8: Distribuidora Eléctrica

- **Ejercicio 9: Cuenta con ganchos**

- Ejercicio 10: Job Scheduler

depositar(200) ?

extraer(\$100) ?

extraer(\$400) ?

Caja Ahorro  
\$ 400

✓ Saldo \$596

✓ Saldo \$298

✗

Cuenta Corriente  
\$100  
límite: \$500

✓ Saldo \$300

✓ Saldo \$0

✓ Saldo \$-300



FACULTAD DE INFORMATICA



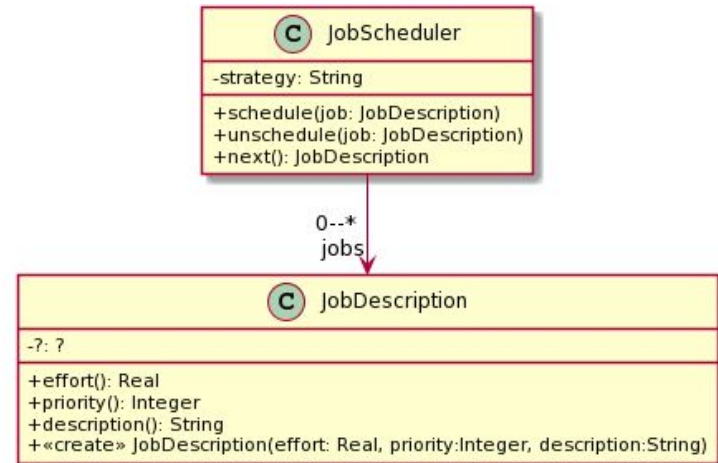
UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Ejercicio 10: Job Scheduler

## Ejercicios de la semana:

- Ejercicio 8: Distribuidora Eléctrica
- Ejercicio 9: Cuenta con ganchos
- **Ejercicio 10: Job Scheduler**

El JobScheduler es un objeto cuya responsabilidad es determinar qué trabajo debe resolverse a continuación. El siguiente diseño ayuda a entender cómo funciona la implementación actual del JobScheduler.

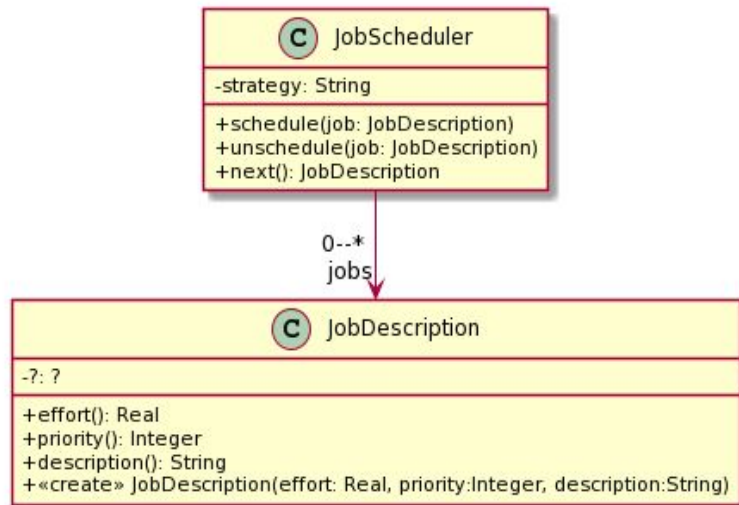


- El mensaje `schedule(job: JobDescription)` recibe un job (trabajo) y lo agrega al final de la colección de trabajos pendientes.
- El mensaje `next()` determina cuál es el siguiente trabajo de la colección que debe ser atendido, lo retorna, y lo quita de la colección.



FACULT

# Ejercicio 10: Job Scheduler



FACULTAD

```
public JobDescription next() {
    JobDescription nextJob = null;

    switch (strategy) {
        case "FIFO":
            nextJob = jobs.get(0);
            this.unschedule(nextJob);
            return nextJob;

        case "LIFO":
            nextJob = jobs.get(jobs.size()-1);
            this.unschedule(nextJob);
            return nextJob;

        case "HighestPriority":
            nextJob = jobs.stream()
                .max((j1,j2) -> Double.compare(j1.getPriority(), j2.getPriority()))
                .orElse(null);
            this.unschedule(nextJob);
            return nextJob;

        case "MostEffort":
            nextJob = jobs.stream()
                .max((j1,j2) -> Double.compare(j1.getEffort(), j2.getEffort()))
                .orElse(null);
            this.unschedule(nextJob);
            return nextJob;
    }
    return null;
}
```

# 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 3 - Presupuestos


### Foros

  Ejercicio 5 - Genealogía Salvaje

### Foros

 Ejercicio 8 - Distribuidora Eléctrica

 Ejercicio 9 - Cuenta con ganchos

 Ejercicio 10 - Job Scheduler



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA