

# Orientación a Objetos I

Agenda de la semana  
del 7 de octubre



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Actividades de la semana anterior

- Ejercicio 12: Volumen y superficie de sólidos
- Ejercicio 13: Cliente de correo con adjuntos
- Ejercicio 14: Intervalo de tiempo



FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

# Actividades esperadas para esta semana

- Ejercicio 15: Distribuidora Eléctrica
- Ejercicio 16: Filtered Set

## 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 15: Distribuidora Eléctrica

Una distribuidora eléctrica desea gestionar los consumos de sus usuarios para la emisión de facturas de cobro.

De cada usuario se conoce su nombre y domicilio. Se considera que cada usuario sólo puede tener un único domicilio en donde se registran los consumos.

Los consumos de los usuarios se dividen en dos componentes, ambos medidos en kWh (kilowatt/hora):

- **Consumo de energía activa:** tiene un costo asociado para el usuario.
- **Consumo de energía reactiva:** no genera ningún costo para el usuario, es decir, se utiliza solamente para determinar si hay alguna bonificación.

Se cuenta con un **cuadro tarifario** que establece el precio del kWh para calcular el costo del consumo de energía activa. Este cuadro tarifario puede ser ajustado periódicamente según sea necesario (por ejemplo, para reflejar cambios en los costos).

Para emitir la factura de un cliente se tiene en cuenta **solo su último consumo registrado**. Los datos que debe contener la factura son los siguientes:

- El usuario a quien se está cobrando.
- La fecha de emisión.
- La bonificación, si aplica.
- El monto final de la factura: se calcula restando la bonificación al costo del consumo:
  - El costo del consumo se calcula multiplicando el consumo de energía activa por el **precio del kWh** proporcionado por el cuadro tarifario.
  - Se calcula su **factor de potencia** para determinar si hay alguna bonificación aplicable. Si el factor de potencia estimado (fpe) del último consumo del usuario es mayor a 0.8, el usuario recibe una bonificación del 10%.

El factor de potencia estimado se calcula de acuerdo a la siguiente fórmula. Para realizar las operaciones matemáticas, puede ayudarse con la clase [Math](#)

$$fpe = \frac{EnergiaActiva}{\sqrt{EnergiaActiva^2 + EnergiaReactiva^2}}$$

# Ejercicio 15: Distribuidora Eléctrica

Realice el modelo de dominio detallando para cada clase conceptual identificada:

- La categoría correspondiente a la clase
- Los atributos candidatos de la clase
- Las asociaciones entre los conceptos

Repasar material de teoría  
Ejercicio “empresa  
mantenimiento”



FAC

| Categoría de Clase Conceptual   | Ejemplos   |
|---------------------------------|--|
| Objeto físico o tangible        | Libro impreso  |
| Especificación de una cosa      | Especificación del producto, descripción               |
| Lugar                           | Tienda   |
| Transacción                     | Compra, pago, cancelación                              |
| Roles de la gente               | cliente  |
| Contenedor de cosas             | Catálogo de libros, carrito, Avion                     |
| Cosas en un contenedor          | Libro, Artículo, Pasajero                              |
| Otros sistemas                  | Sistema de facturación de AFIP                         |
| Hechos                          | cancelación, venta, pago                               |
| Reglas y políticas              | Política de cancelación                                |
| Registros financieros/laborales | Factura/ Recibo de compra                              |
| Manuales, documentos            | Reglas de cancelación, cambios de categoría de cliente |

# Ejercicio 15: Distribuidora Eléctrica

Realice el modelo de dominio detallando para cada clase conceptual identificada:

- La categoría correspondiente a la clase
- Los atributos candidatos de la clase
- **Las asociaciones entre los conceptos**

**Repasar material de teoría**  
**Ejercicio “empresa**  
**mantenimiento”**



F/

| Categoría   | Ejemplo                          |
|---|----------------------------------|
| A es una parte física de B                              | Ala - Avión                      |
| A es una parte lógica de B                              | Etapa de vuelo - Vuelo           |
| A está físicamente contenido en B                       | Pasajero - Avión                 |
| A está lógicamente contenido en B                       | Libro-Catálogo                   |
| A es una descripción para B                             | EspecificaciónDeProducto - Libro |
| A es un miembro de B                                    | Piloto - Aerolínea               |
| A usa o maneja a B                                      | Cliente Carrito                  |
| A se comunica con B                                     | Cliente- Librería                |
| A está relacionado con la transacción B                 | Cliente- Pago                    |
|   | Cliente- Agregar al carrito      |
| A es una transacción relacionada con otra transacción B | Pago- Compra                     |
|   |                                  |
| A es dueño de B   | Cliente- Carrito                 |



# Ejercicio 16: Filtered Set

En la teoría de colecciones se explicaron algunos tipos de colecciones; en particular, el **Set** (*java.util.Set*) es una colección que no admite duplicados y no tiene índice para sus elementos.

Implemente una clase **EvenNumberSet** (conjunto de números pares). Esta clase se comporta casi exactamente igual a **Set**, con la diferencia que únicamente permite agregar números enteros que sean pares. Por simplicidad, considere únicamente el tipo de datos **Integer** para su solución (ignore el resto de tipos de datos numéricos).

Tenga en cuenta que la clase **EvenNumberSet** debe implementar la interface **Set<E>** de Java. Esto significa que a las variables de tipo *Set<Integer>* se les puede asignar un objeto concreto de tipo **EvenNumberSet** y luego utilizarlo enviando los mensajes que están definidos en el protocolo de **Set<E>**.

```
Set<Integer> numbers  
    = new EvenNumberSet();  
numbers.add(1);  
numbers.add(2);  
numbers.add(4);  
numbers.add(2);
```

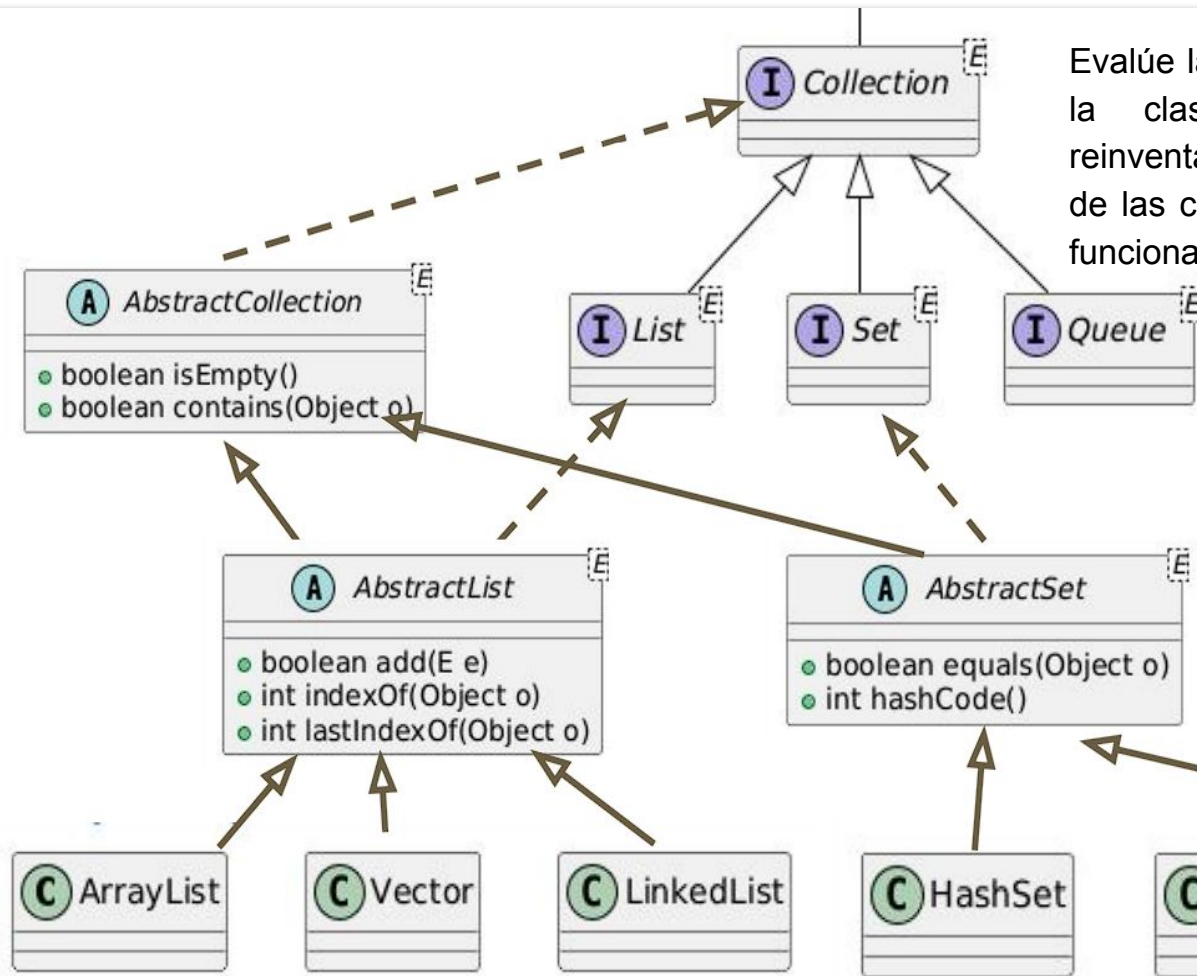


FACULTAD DE INFORMATICA



UNIVERSIDAD  
NACIONAL  
DE LA PLATA





Evalúe las distintas opciones para implementar la clase **EvenNumberSet**. Para evitar reinventar la rueda, considere reutilizar alguna de las clases existentes en Java que ofrezcan funcionalidades similares.

Repasar material de teoría

Colecciones

Module java.base

Package java.util

## Interface Set<E>

Type Parameters:

E - the type of elements maintained by this set

All Superinterfaces:

Collection<E>, Iterable<E>

All Known Subinterfaces:

EventSet, NavigableSet<E>, SortedSet<E>

All Known Implementing Classes:

AbstractSet, ConcurrentHashMap.KeySetView, ConcurrentSkipListSet, CopyOnWriteArraySet, EnumSet, HashSet, JobStateReasons, LinkedHashSet, TreeSet

Evalúe las distintas opciones para implementar la clase **EvenNumberSet**. Para evitar reinventar la rueda, considere reutilizar alguna de las clases existentes en Java que ofrezcan funcionalidades similares.

| All Methods       | Static Methods                       | Instance Methods | Abstract Methods  | Default Methods |
|-------------------|--------------------------------------|------------------|---|-----------------|
| Modifier and Type | Method                               |                  | Description   |                 |
| boolean           | add(E e)                             |                  | Adds the specified element to this set if it is not already present (optional operation).                             |                 |
| boolean           | addAll(Collection<? extends E> c)    |                  | Adds all of the elements in the specified collection to this set if they're not already present (optional operation). |                 |
| void              | clear()                              |                  | Removes all of the elements from this set (optional operation).   |                 |
| boolean           | contains(Object o)                   |                  | Returns true if this set contains the specified element.  |                 |
| boolean           | containsAll(Collection<?> c)         |                  | Returns true if this set contains all of the elements of the specified collection.                                    |                 |
| static <E> Set<E> | copyOf(Collection<? extends E> coll) |                  | Returns an <b>unmodifiable</b> Set containing the elements of the given Collection.                                   |                 |
| boolean           | equals(Object o)                     |                  | Compares the specified object with this set for equality.   |                 |
| int               | hashCode()                           |                  | Returns the hash code value for this set.   |                 |
| boolean           | isEmpty()                            |                  | Returns true if this set contains no elements.  |                 |

# Ejercicio 16: Filtered Set

- a. Investigue qué clases se pueden utilizar para implementar la clase **EvenNumberSet**. Consulte la [documentación de Set](#).
- b. Explique brevemente cómo propone utilizar las clases investigadas anteriormente para implementar su solución.
- c. Implemente en Java las alternativas que haya propuesto.
- d. Implemente tests automatizados utilizando JUnit para verificar sus implementaciones.
- e. Compare las soluciones y liste las ventajas y desventajas de cada una.

**Repasar material de teoría**

**Reuso de Código**  
**Herencia vs. Composición**



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 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