



SEP

SES

TecNM

INSTITUTO TECNOLÓGICO DE TOLUCA

“PR3 Manejando arreglos mediante ArrayList”

PROGRAMACION AVANZADA

INGENIERIA EN MECATRONICA

PRESENTA:

ABRAHAM SÁNCHEZ JIMÉNEZ	22280836
EDUARDO HERNÁNDEZ SALVADOR	22280890
DANIEL DE LA CRUZ VELAZQUEZ	22280933
SANTIAGO ROJAS SAAVEDRA	22280856

EQUIPO “NEURONA CIBERNÉTICA”

PROFESOR:

JESUS AURELIO CASTRO MAGAÑA

GRUPO: 190002

```
Misc.java    DeerPelt.java    BearPelt.java    DeerPelt.java    BearPelt.java

1  package rpg.enums;
2
3  public enum ItemType { AbrahamDell
4      WEAPON, 11 usages
5      ARMOR, 11 usages
6      MISC 1 usage
7  }
8
```

```
package rpg.interfaces;

import rpg.entities.Player;

public interface Equipable { 2 usages
    void equip(Player player); no us
    void unequip(Player player); no
}
```

```
BearPelt.java    SteelSword.java

package rpg.interfaces;

public interface Sellable { no us
    void sell(); no usages AbrahamDell
}
```

```
SteelSword.java    ItemType.java    Equipable.java    Sellable.java    Armor.java

package rpg.items.armors;

import rpg.enums.ItemType;
import rpg.items.Equipment;

/**
 * Clase Armor que representa una armadura en el juego. Typo: In word 'Clase'. Typo: In word 'Clase'.
 */
public abstract class Armor extends Equipment { 19 usages 5 inheritors AbrahamDell

    protected ItemType itemType;

    /**
     * Constructor de la clase Armor. Typo: In word 'clase'.
     *
     * @param name El nombre de la armadura. Typo: In word 'nombre'. Typo: In word 'nombre'.
     * @param description La descripción de la armadura. Typo: In word 'descripc
     * @param price El precio de la armadura. Typo: In word 'precio'. Typo: In word 'precio'.
     * @param itemType El tipo de armadura. Typo: In word 'tipo'. Typo: In word 'tipo'.
     */
    public Armor(String name, String description, int price, ItemType itemType) {
        super();
        this.name = name;
    }
}
```

```

    */
    public Armor(String name, String description, int price, ItemType itemType) {
        super();
        this.name = name;
        this.description = description;
        this.price = price;
        this.type = ItemType.ARMOR;
    }

    /**
     * Getter para el tipo de armadura. Typo: In word 'tipo'. Typo: In word 'arm
     *
     * @return El tipo de armadura. Typo: In word 'tipo'. Typo: In word 'armadur
     */
    public ItemType getArmorType() { no usages AbrahamDell Method 'getArm
        return itemType;
    }
}

```

Item.java Equipable.java Sellable.java Armor.java DiamondArmor.java x Inventory.java Item.java

```

package rpg.items.armors;

import rpg.items.armors.Armor; Unused import statement.
import rpg.enums.ItemType;

/**
 * Clase DiamondArmor que representa una armadura de diamante en el juego. Typo: In word 'Clase'. Typo: In word 'representa'
 */
public class DiamondArmor extends Armor { no usages AbrahamDell Class 'DiamondArmor' is never used.

    /**
     * Constructor de la clase DiamondArmor. Typo: In word 'clase'.
     */
    public DiamondArmor() { no usages AbrahamDell Constructor 'DiamondArmor()' is never used.
        super( name: "Armadura de diamante", description: "Una armadura hecha de diamante genuino.", price: 250, ItemType.ARMOR);
    }

    @Override AbrahamDell
    protected void initItem() {
        this.name = "Armadura de diamante"; Typo: In word 'Armadura'.
        this.description = "Una armadura hecha de diamante genuino"; Typo: In word 'armadura'. Typo: In word 'hecha'. Typo
        this.price = 250;
        this.itemType = ItemType.ARMOR;
    }
}

```

```

package rpg.items.armors;
import rpg.items.armors.Armor; Unused import statement.
import rpg.enums.ItemType;

/**
 * Clase GoldArmor que representa una armadura de diamante en el juego. Typo: In word 'Clase'. Typo: In word 'repre
 */
public class GoldArmor extends Armor { no usages 1 AbrahamDell * Class 'GoldArmor' is never used.

    /**
     * Constructor de la clase DiamondArmor. Typo: In word 'clase'.
     */
    public GoldArmor() { no usages 1 AbrahamDell Constructor 'GoldArmor()' is never used.
        super( name: "Armadura de oro", description: "Una armadura hecha de oro mitico.", price: 200, ItemType.ARMOR);
    }

    @Override 1 AbrahamDell
    protected void initItem() {
        this.name = "Armadura de oro"; Typo: In word 'Armadura'.
        this.description = "Una armadura hecha de oro mitico."; Typo: In word 'armadura'. Typo: In word 'hecha'.
        this.price = 200;
        this.itemType = ItemType.ARMOR;
    }
}

```

```

package rpg.items.armors;
import rpg.items.armors.Armor; Unused import statement.
import rpg.enums.ItemType;

/**
 * Clase IronArmor que representa una armadura de hierro en el juego. Typo: In word 'Clase'. Typo: In word 'representa'.
 */
public class IronArmor extends Armor { 2 usages 1 AbrahamDell

    /**
     * Constructor de la clase IronArmor. Typo: In word 'clase'.
     */
    public IronArmor() { 1 usage 1 AbrahamDell
        super( name: "Armadura de hierro", description: "Una armadura hecha de hierro forjado.", price: 150, ItemType.ARMOR);
    }

    @Override 1 AbrahamDell
    protected void initItem() {
        this.name = "Armadura de hierro"; Typo: In word 'Armadura'. Typo: In word 'hierro'.
        this.description = "Una armadura hecha de hierro forjado."; Typo: In word 'armadura'. Typo: In word 'hecha'. Typ
        this.price = 150;
        this.itemType = ItemType.ARMOR;
    }
}

```

```

Sellable.java  Armor.java  DiamondArmor.java  GoldArmor.java  IronArmor.java  SteelArmor.java  x  Invent
package rpg.items.armors;
import rpg.items.armors.Armor;  Unused import statement.
import rpg.enums.ItemType;

/**
 * Clase SteelArmor que representa una armadura de acero en el juego.  Typo: In word 'Clase'.  Typo: In word 'representa'.
 */
public class SteelArmor extends Armor {  no usages  ± AbrahamDell  *  Class 'SteelArmor' is never used.

    /**
     * Constructor de la clase SteelArmor.  Typo: In word 'clase'.
     */
    public SteelArmor() {  no usages  ± AbrahamDell  Constructor 'SteelArmor()' is never used.
        super(name: "Armadura de acero", description: "Una armadura hecha de acero templado.", price: 100, ItemType.ARMOR);
    }

    @Override  ± AbrahamDell
    protected void initItem() {
        this.name = "Armadura de acero";  Typo: In word 'Armadura'.  Typo: In word 'acero'.
        this.description = "Una armadura hecha de acero templado.";  Typo: In word 'armadura'.  Typo: In word 'hecha'.  Typo: In word 'templado'.
        this.price = 100;
        this.itemType = ItemType.ARMOR;
    }
}

```

```

Sellable.java  Armor.java  DiamondArmor.java  GoldArmor.java  IronArmor.java  SteelArmor.java  WoodArmor.java
package rpg.items.armors;
import rpg.enums.ItemType;
import rpg.items.armors.Armor;  Unused import statement.

/**
 * Clase WoodArmor que representa una armadura de madera en el juego.  Typo: In word 'Clase'.  Typo: In word 'representa'.
 */
public class WoodArmor extends Armor {  2 usages  ± AbrahamDell  *

    /**
     * Constructor de la clase WoodArmor.  Typo: In word 'clase'.
     */
    public WoodArmor() {  1 usage  ± AbrahamDell
        super(name: "Armadura de madera", description: "Una armadura hecha de madera resistente.", price: 50, ItemType.ARMOR);
    }

    @Override  ± AbrahamDell
    protected void initItem() {
        this.name = "Armadura de madera";  Typo: In word 'Armadura'.
        this.description = "Una armadura hecha de madera resistente.";  Typo: In word 'armadura'.  Typo: In word 'hecha'.  Typo: In word 'resistente'.
        this.price = 50;
        this.itemType = ItemType.ARMOR;
    }
}

```

Armor.java © DiamondArmor.java © GoldArmor.java © IronArmor.java © SteelArmor.java © WoodArmor.java

```

package rpg.items.miscs;

import rpg.items.miscs.Misc; Unused import statement.

/**
 * Clase BearPelt que representa una piel de oso en el juego. Typo: In word 'Clase'. Typo: In word 'r'
 */
public class BearPelt extends Misc { no usages    AbrahamDell    Class 'BearPelt' is never used.

    @Override    AbrahamDell
    protected void initItem() {

    }

    /**
     * Constructor de la clase BearPelt. Typo: In word 'clase'.
     */
    public BearPelt() { no usages    AbrahamDell    Constructor 'BearPelt()' is never used.
        super( name: "Piel de oso", description: "Una piel de oso con habilidad de fuerza.", price: 25);
    }

}

```

DiamondArmor.java © GoldArmor.java © IronArmor.java © SteelArmor.java © WoodArmor.java © BearPelt.java

```

package rpg.items.miscs;

import rpg.items.miscs.Misc; Unused import statement.

/**
 * Clase Deer Pelt que representa una piel de ciervo en el juego. Typo: In word 'Clase'. Typo: In word 'r'
 */
public class DeerPelt extends Misc { no usages    AbrahamDell    Class 'DeerPelt' is never used.

    @Override    AbrahamDell
    protected void initItem() {

    }

    /**
     * Constructor de la clase DeerPelt. Typo: In word 'clase'.
     */
    public DeerPelt() { no usages    AbrahamDell    Constructor 'DeerPelt()' is never used.
        super( name: "Piel de ciervo", description: "Una piel de ciervo con aumento de agilidad.", price: 25);
    }

}

```

```

© GoldArmor.java  © IronArmor.java  © SteelArmor.java

package rpg.items.miscs;
import rpg.enums.ItemType;
import rpg.items.Item;

/**
 * Clase Misc que representa un objeto misceláneo en el juego. Typo: In word 'clase'. Typo: In word 'clase'.
 */
public abstract class Misc extends Item { 19 usages 5 inheritors

    /**
     * Constructor de la clase Misc. Typo: In word 'clase'.
     *
     * @param name El nombre del objeto misceláneo. Typo: In word 'clase'.
     * @param description La descripción del objeto misceláneo.
     * @param price El precio del objeto misceláneo. Typo: In word 'clase'.
     */
    public Misc(String name, String description, int price) { 5
        super();
        this.name = name;
        this.description = description;
        this.price = price;
        this.type = ItemType.MISC;
    }
}

```

```

© IronArmor.java  © SteelArmor.java  © WoodArmor.java  © BearPelt.java  © DeerPelt.java  © Misc.java

package rpg.items.miscs;

import rpg.items.miscs.Misc; Unused import statement.

/**
 * Clase RabbitPelt que representa una piel de conejo en el juego. Typo: In word 'Clase'. Typo: In word 'Clase'.
 */
public class RabbitPelt extends Misc { no usages AbrahamDell Class 'RabbitPelt' is never used.

    @Override AbrahamDell
    protected void initItem() {

    }

    /**
     * Constructor de la clase RabbitPelt. Typo: In word 'clase'.
     */
    public RabbitPelt() { no usages AbrahamDell Constructor 'RabbitPelt()' is never used.
        super( name: "Piel de conejo", description: "Una piel de conejo con habilidad de salto.", price: 25);
    }
}

```

```

SteelArmor.java WoodArmor.java BearPelt.java DeerPelt.java Misc.java RabbitPelt.java
package rpg.items.miscs;

import rpg.items.miscs.Misc; Unused import statement.

/**
 * Clase SquirrelPelt que representa una piel de ardilla en el juego. Typo: In word 'Clase'. Typo: In word 're
 */
public class SquirrelPelt extends Misc { 2 usages AbrahamDell

    @Override AbrahamDell
    protected void initItem() {

    }

    /**
     * Constructor de la clase SquirrelPelt. Typo: In word 'clase'.
     */
    public SquirrelPelt() { 1 usage AbrahamDell
        super( name: "Piel de ardilla", description: "Una piel de ardilla con habilidad de velocidad.", price: 25);
    }
}

```

```

WoodArmor.java BearPelt.java DeerPelt.java Misc.java RabbitPelt.java Squ
package rpg.items.miscs;

import rpg.items.miscs.Misc; Unused import statement.

/**
 * Clase WolfPelt que representa una piel de lobo en el juego. Typo: In word 'Clase'. Typo: In word
 */
public class WolfPelt extends Misc { 2 usages AbrahamDell

    @Override AbrahamDell
    protected void initItem() {

    }

    /**
     * Constructor de la clase WolfPelt. Typo: In word 'clase'.
     */
    public WolfPelt() { 1 usage AbrahamDell
        super( name: "Piel de lobo", description: "Una piel de lobo con aumento de ataque.", price: 25);
    }
}

```



```

BearPelt.java  DeerPelt.java  Misc.java  RabbitPelt.java  SquirrelPelt.java  WolfPelt.java  DiamondSword.java

package rpg.items.weapons;
import rpg.enums.ItemType;
import rpg.items.weapons.Weapon; Unused import statement.

/**
 * Clase DiamondSword que representa una espada de diamante en el juego. Typo: In word 'Clase'. Typo: In word 'representa'
 */
public class DiamondSword extends Weapon { no usages  AbrahamDell * Class 'DiamondSword' is never used.

    /**
     * Constructor de la clase DiamondSword. Typo: In word 'clase'.
     */
    public DiamondSword() { no usages  AbrahamDell Constructor 'DiamondSword()' is never used.
        super( name: "Espada de diamante", description: "Una espada hecha de diamante genuino.", price: 300, ItemType.WEAPON);
    }

    @Override  AbrahamDell
    protected void initItem() {
        this.name = "Espada de diamante"; Typo: In word 'Espada'.
        this.description = "Una espada hecha de diamante genuino."; Typo: In word 'espada'. Typo: In word 'hecha'. Typo:
        this.price = 300;
        this.itemType = ItemType.WEAPON;
    }
}

```

```

DeerPelt.java  Misc.java  RabbitPelt.java  SquirrelPelt.java  WolfPelt.java  DiamondSword.java

package rpg.items.weapons;
import rpg.enums.ItemType;
import rpg.items.weapons.Weapon; Unused import statement.

/**
 * Clase GoldSword que representa una espada de oro en el juego. Typo: In word 'Clase'. Typo: In word 'representa'
 */
public class GoldSword extends Weapon { no usages  AbrahamDell * Class 'GoldSword' is never used.

    /**
     * Constructor de la clase GoldSword. Typo: In word 'clase'.
     */
    public GoldSword() { no usages  AbrahamDell Constructor 'GoldSword()' is never used.
        super( name: "Espada de oro", description: "Una espada hecha de oro mitico.", price: 250, ItemType.WEAPON);
    }

    @Override  AbrahamDell
    protected void initItem() {
        this.name = "Espada de oro"; Typo: In word 'Espada'.
        this.description = "Una espada hecha de oro mitico."; Typo: In word 'espada'. Typo: In word 'hecha'. Typo:
        this.price = 250;
        this.itemType = ItemType.WEAPON;
    }
}

```

```

ac.java  © RabbitPelt.java  © SquirrelPelt.java  © WolfPelt.java  © DiamondSword.java  © GoldSword.java  © IronSword.java
package rpg.items.weapons;
import rpg.enums.ItemType;
import rpg.items.weapons.Weapon;  Unused import statement.

/**
 * Clase IronSword que representa una espada de hierro en el juego. Typo: In word 'Clase'. Typo: In word 'representa'.
 */
public class IronSword extends Weapon { 2 usages  ⚡ AbrahamDell *

    /**
     * Constructor de la clase IronSword. Typo: In word 'clase'.
     */
    public IronSword() { 1 usage  ⚡ AbrahamDell
        super( name: "Espada de hierro", description: "Una espada hecha de hierro forjado.", price: 200, ItemType.WEAPON);
    }

    @Override  ⚡ AbrahamDell
    protected void initItem() {
        this.name = "Espada de hierro"; Typo: In word 'Espada'. Typo: In word 'hierro'.
        this.description = "Una espada hecha de hierro forjado."; Typo: In word 'espada'. Typo: In word 'hecha'. Typo: In word 'forjado'.
        this.price = 200;
        this.itemType = ItemType.WEAPON;
    }
}

```

```

Pelt.java  © SquirrelPelt.java  © WolfPelt.java  © DiamondSword.java  © GoldSword.java  © IronSword.java  © SteelSword.java
package rpg.items.weapons;
import rpg.enums.ItemType;
import rpg.items.weapons.Weapon;  Unused import statement.

/**
 * Clase SteelSword que representa una espada de acero en el juego. Typo: In word 'Clase'. Typo: In word 'representa'.
 */
public class SteelSword extends Weapon { 2 usages  ⚡ AbrahamDell *

    /**
     * Constructor de la clase SteelSword. Typo: In word 'clase'.
     */
    public SteelSword() { 1 usage  ⚡ AbrahamDell
        super( name: "Espada de acero", description: "Una espada hecha de acero fuerte.", price: 150, ItemType.WEAPON);
    }

    @Override  ⚡ AbrahamDell
    protected void initItem() {
        this.name = "Espada de acero"; Typo: In word 'Espada'. Typo: In word 'acero'.
        this.description = "Una espada hecha de acero fuerte."; Typo: In word 'espada'. Typo: In word 'hecha'. Typo: In word 'fuerte'.
        this.price = 150;
        this.itemType = ItemType.WEAPON;
    }
}

```

```

quirlPelt.java    © WolfPelt.java    © DiamondSword.java    © GoldSword.java    © IronSw
package rpg.items.weapons;
import rpg.enums.ItemType;
import rpg.items.Equipment;

/**
 * Clase Weapon que representa un arma en el juego. Typo: In word 'Clase'. Typo:
 */
public abstract class Weapon extends Equipment { 19 usages 5 inheritors AbrahamDell

    protected ItemType itemType;

    /**
     * Constructor de la clase Weapon. Typo: In word 'clase'.
     *
     * @param name El nombre del arma. Typo: In word 'nombre'. Typo: In
     * @param description La descripción del arma. Typo: In word 'descripción'.
     * @param price El precio del arma. Typo: In word 'precio'. Typo: In
     * @param itemType El tipo de arma. Typo: In word 'tipo'. Typo: In word 'a
     */
    public Weapon(String name, String description, int price, ItemType itemType) {
        super();
        this.name = name;
        this.description = description;
        this.price = price;
    }

```

```

quirlPelt.java    © WolfPelt.java    © DiamondSword.java    © GoldSword.java    © IronSw
public abstract class Weapon extends Equipment { 19 usages 5 inheritors AbrahamDell
    * @param itemType El tipo de arma. Typo: In word 'tipo'. Typo: In word 'a
    */
    public Weapon(String name, String description, int price, ItemType itemType) {
        super();
        this.name = name;
        this.description = description;
        this.price = price;
        this.type = ItemType.WEAPON;
    }

    /**
     * Getter para el tipo de arma. Typo: In word 'tipo'. Typo: In word 'arma'.
     *
     * @return El tipo de arma. Typo: In word 'tipo'. Typo: In word 'arma'.
     */
    public ItemType getWeaponType() { return itemType; }
}

```

```

fPelt.java  © DiamondSword.java  © GoldSword.java  © IronSword.java  © SteelSword.java  © Weapon.java  © WoodSword.java x
package rpg.items.weapons;
import rpg.enums.ItemType;
import rpg.items.weapons.Weapon; Unused import statement.

/**
 * Clase WoodSword que representa una espada de madera en el juego. Typo: In word 'Clase'. Typo: In word 'representa'. Typo: In word 'espa
 */
public class WoodSword extends Weapon { no usages  ⚠ AbrahamDell Class 'WoodSword' is never used.

    /**
     * Constructor de la clase WoodSword. Typo: In word 'clase'.
     */
    public WoodSword() { super( name: "Espada de madera", description: "Una espada hecha de madera resistente.", price: 50, ItemType.WEAPON); }

    @Override  ⚠ AbrahamDell
    protected void initItem() {
        this.name = "Espada de madera"; Typo: In word 'Espada'.
        this.description = "Una espada hecha de madera resistente."; Typo: In word 'espada'. Typo: In word 'hecha'. Typo: In word 'resisten
        this.price = 50;
        this.itemType = ItemType.WEAPON;
    }
}

```

```

DiamondSword.java  © GoldSword.java  © IronSword.java  © SteelSword.java
package rpg.items;

import rpg.entities.Player;
import rpg.enums.Stats;
import rpg.interfaces.Equipable;
import java.util.HashMap;

/**
 * Clase Equipment que representa un objeto equipable en el juego.
 */
public abstract class Equipment extends Item implements Equipable {

    protected HashMap<Stats, Integer> stats;

    /**
     * Constructor de la clase Equipment. Typo: In word 'clase'.
     */
    public Equipment() { 2 usages  ⚠ AbrahamDell
        super();
        stats = new HashMap<>();
    }

    /**
     * Método que equipa el objeto en el jugador. Typo: In word 'Mé

```

```

ondSword.java    © GoldSword.java    © IronSword.java    © SteelSword.java

public abstract class Equipment extends Item implements Equipable { 4 usag
    /**
     * Método que equipa el objeto en el jugador. Typo: In word 'Método'.
     *
     * @param player El jugador que equipa el objeto. Typo: In word 'juga
     */
    @Override no usages  ⚠ AbrahamDell
    public void equip(Player player) {
        // Implementar la lógica para equipar el objeto en el jugador. Ty
    }

    /**
     * Método que desequipa el objeto en el jugador. Typo: In word 'Métod
     *
     * @param player El jugador que desequipa el objeto. Typo: In word 'j
     */
    @Override no usages  ⚠ AbrahamDell
    public void unequip(Player player) {
        // Implementar la lógica para desequipar el objeto en el jugador.
    }
}

```

```

© GoldSword.java    © IronSword.java    © SteelSword.java    © Weapon.java    ©

package rpg.items;

import rpg.items.armors.Armor;
import rpg.items.miscs.Misc;
import rpg.items.weapons.Weapon;
import rpg.items.Item; Unused import statement.
import java.util.ArrayList;

/**
 * Clase Inventory que representa el inventario de un personaje en el juego.
 */
public class Inventory { 3 usages  ⚠ AbrahamDell

    private ArrayList<Item> items; 13 usages    Field 'items' may be 'fi
    private int capacity; 5 usages

    /**
     * Constructor de la clase Inventory. Typo: In word 'clase'.
     *
     * @param capacity La capacidad del inventario. Typo: In word 'capacida
     */
    public Inventory(int capacity) { 1 usage  ⚠ AbrahamDell
        this.capacity = capacity;
        items = new ArrayList<>();
    }
}

```

```

GoldSword.java    © IronSword.java    © SteelSword.java

public class Inventory { 3 usages  ⤴ AbrahamDell *
    public Inventory(int capacity) { 1 usage  ⤴ Abrah
        this.capacity = capacity;
        items = new ArrayList<>();
    }

    /**
     * Método que agrega un objeto al inventario. Typ
     *
     * @param item El objeto a agregar. Typo: In word
     */
    public void addItem(Item item) { 6 usages  ⤴ Abra
        if (items.size() < capacity) {
            items.add(item);
        } else {
            System.out.println("Inventario lleno");
        }
    }

    /**
     * Método que elimina un objeto del inventario.
     *
     * @param item El objeto a eliminar. Typo: In word
     */

```

```

GoldSword.java    © IronSword.java    © SteelSword.java    © Weapon.java

public class Inventory { 3 usages  ⤴ AbrahamDell *
    public void removeItem(Item item) { no usages  ⤴ AbrahamDell
        items.remove(item);
    }

    /**
     * Método que devuelve un objeto del inventario por índice. Typ
     *
     * @param index El índice del objeto. Typo: In word 'índice'.
     * @return El objeto en el índice especificado. Typo: In word
     */
    public Item getItem(int index) { no usages  ⤴ AbrahamDell
        return items.get(index);
    }

    /**
     * Método que devuelve la cantidad de objetos en el inventario.
     *
     * @return La cantidad de objetos en el inventario. Typo: In word
     */
    public int getItemCount() { no usages  ⤴ AbrahamDell  Method
        return items.size();
    }

```

GoldSword.java © IronSword.java © SteelSword.java © Weapon.java

```

public class Inventory { 3 usages  ⚡ AbrahamDell *
    /**
     * Método que verifica si el inventario está lleno. Typo: In word 'Método'
     *
     * @return True si el inventario está lleno, false en caso contrario.
     */
    public boolean isFull() { no usages  ⚡ AbrahamDell      Method 'isFull()'
        return items.size() == capacity;
    }

    /**
     * Método que verifica si el inventario está vacío. Typo: In word 'Método'
     *
     * @return True si el inventario está vacío, false en caso contrario.
     */
    public boolean isEmpty() { no usages  ⚡ AbrahamDell      Method 'isEmpty()'
        return items.isEmpty();
    }

    /**
     * Método que limpia el inventario. Typo: In word 'Método'. Typo: In word 'Método'
     */
    public void clear() { no usages  ⚡ AbrahamDell      Method 'clear()'
        items.clear();
    }

```

GoldSword.java © IronSword.java © SteelSword.java © Weapon.java

```

public class Inventory { 3 usages  ⚡ AbrahamDell *
    /**
     * Método que aumenta la capacidad del inventario. Typo: In word 'Método'
     *
     * @param amount La cantidad de espacio adicional. Typo: In word 'Método'
     */
    public void increaseCapacity(int amount) { no usages  ⚡ AbrahamDell
        capacity += amount;
        items.ensureCapacity(capacity);
    }

    /**
     * Método que devuelve una lista de armaduras en el inventario.
     *
     * @return La lista de armaduras en el inventario. Typo: In word 'Método'
     */
    public ArrayList<Armor> getArmors() { 1 usage  ⚡ AbrahamDell
        ArrayList<Armor> armors = new ArrayList<>();
        for (Item item : items) {
            if (item instanceof Armor) {
                armors.add((Armor) item);
            }
        }
        return armors;
    }

```

```

GoldSword.java  © IronSword.java  © SteelSword.java  © Weapon.java  ©
public class Inventory { 3 usages  AbrahamDell *
    /**
     * Método que devuelve una lista de objetos misceláneos en el inventario.
     *
     * @return La lista de objetos misceláneos en el inventario. Typo: In word 'misceláneos'.
     */
    public ArrayList<Misc> getMiscs() { 1 usage  AbrahamDell  Typo: In word 'miscs'.
        ArrayList<Misc> miscs = new ArrayList<>();
        for (Item item : items) {
            if (item instanceof Misc) {
                miscs.add((Misc) item);
            }
        }
        return miscs;
    }

    public ArrayList<Weapon> getWeapons() { 1 usage  AbrahamDell
        ArrayList<Weapon> weapons = new ArrayList<>();
        for (Item item : items) {
            if (item instanceof Weapon) {
                weapons.add((Weapon) item);
            }
        }
        return weapons;
    }
}

```

```

© IronSword.java  © SteelSword.java  © Weapon.java  ©
package rpg.items;

import rpg.enums.ItemType;

/**
 * Clase Item que representa un objeto generico en el juego. Typo: In word 'clase'.
 */
public abstract class Item { 11 usages  19 inheritors  AbrahamDell

    protected String name;
    protected String description;
    protected int price;
    protected ItemType type;

    /**
     * Constructor de la clase Item. Typo: In word 'clase'.
     */
    public Item() { 2 usages  AbrahamDell
        initItem();
    }

    /**
     * Método abstracto que inicializa los atributos del Item.
     */
}

```



```

© IronSword.java © SteelSword.java
public abstract class Item { 11 usages 19 inhe
    protected abstract void initItem(); 15 in

    /**
     * Getter para el nombre del Item. Tipo
     *
     * @return El nombre del Item. Tipo: In
     */
    public String getName() { 1 AbrahamDell
        return name;
    }

    /**
     * Getter para la descripción del Item.
     *
     * @return La descripción del Item. Tipo
     */
    public String getDescription() { 1 Abraham
        return description;
    }

    /**
     * Getter para el precio del Item. Tipo
     *
     */
}

public abstract class Item { 11 usages
    return description;
}

    /**
     * Getter para el precio del Item.
     *
     * @return El precio del Item. Tipo
     */
    public int getPrice() { 1 AbrahamD
        return price;
    }

    /**
     * Getter para el tipo del Item. T
     *
     * @return El tipo del Item. Tipo:
     */
    public ItemType getType() { 1 Abrah
        return type;
    }
}

```

```

va © SteelSword.java © Weapon.java © WoodSword.java © SteelSword.java © Weapon.java © WoodSword.java

package rpg;

import javax.swing.JOptionPane;
import rpg.entities.Player; Unused import statement.
import rpg.items.Inventory;
import rpg.items.armors.Armor;
import rpg.items.armors.IronArmor;
import rpg.items.armors.WoodArmor;
import rpg.items.miscs.Misc;
import rpg.items.miscs.SquirrelPelt;
import rpg.items.miscs.WolfPelt;
import rpg.items.weapons.IronSword;
import rpg.items.weapons.SteelSword;
import rpg.items.weapons.Weapon;

public class Main { 1 AbrahamDell *

    public static void main(String[] args) { 1 AbrahamDe

        Inventory inventory = new Inventory( capacity: 10);

        Armor armor = new WoodArmor();
        inventory.addItem(armor);

        public class Main { 1 AbrahamDell *
            public static void main(String[] args) { 1 AbrahamDell *

                Inventory inventory = new Inventory( capacity: 10);

                Armor armor = new WoodArmor();
                inventory.addItem(armor);

                Armor armor1 = new IronArmor();
                inventory.addItem(armor1);

                Misc misc = new WolfPelt();
                inventory.addItem(misc);

                Misc misc2 = new SquirrelPelt();
                inventory.addItem(misc2);

                Weapon weapon = new IronSword();
                inventory.addItem(weapon);

                Weapon weapon1 = new SteelSword();
                inventory.addItem(weapon1);

                String armorMessage = "Armaduras en inventario:\n";
                for (Armor a : inventory.getArmors()) {

```

```

a      © SteelSword.java      © Weapon.java      © WoodSword.java      © Equipment.java      © Inventory.java      © Item.java      © Main.java
public class Main {
    public static void main(String[] args) {
        String armorMessage = "Armaduras en inventario:\n";
        for (Armor a : inventory.getArmors()) {
            armorMessage += "Nombre: " + a.getName() + "\n Descripción: " + a.getDescription() + "\n Precio: " + a.getPrice() +
                "\n Tipo: " + a.getType() + "\n";
        }
        JOptionPane.showMessageDialog( null, armorMessage, "Armaduras", JOptionPane.INFORMATION_MESSAGE);

        String miscMessage = "Objetos misceláneos en inventario:\n";
        for (Misc m : inventory.getMiscs()) {
            miscMessage += "Nombre: " + m.getName() + "\n Descripción: " + m.getDescription() + "\n Precio: " + m.getPrice() +
                "\n Tipo: " + m.getType() + "\n";
        }
        JOptionPane.showMessageDialog( null, miscMessage, "Misceláneos", JOptionPane.INFORMATION_MESSAGE);

        String weaponMessage = "Armas en inventario:\n";
        for (Weapon m : inventory.getWeapons()) {
            weaponMessage += "Nombre: " + m.getName() + "\n Descripción: " + m.getDescription() + "\n Precio: " + m.getPrice() +
                "\n Tipo: " + m.getType() + "\n";
        }
        JOptionPane.showMessageDialog( null, weaponMessage, "Armas", JOptionPane.INFORMATION_MESSAGE);
    }
}

```

