

## Reporte de práctica # 1

**Nombre:** *Roberto Torres Jiménez*

**Matrícula:** *2123007798*

**Fecha:** *30/Septiembre/2015*

## Contenido

1 Introducción.....	3
2 Resultados.....	4
2.1 Paso 1 Compilación y ejecución de "Hola Mundo" en Java.....	4
2.2 Paso 2 Creación de un primer objeto .....	4
2.3 Paso 3 Relación de dependencia y creación de varios objetos. ....	5
3 Conclusiones.....	8

# 1 Introducción

Para alcanzar los objetivos de la practica se realizaron 3 programas cada vez mas completos.

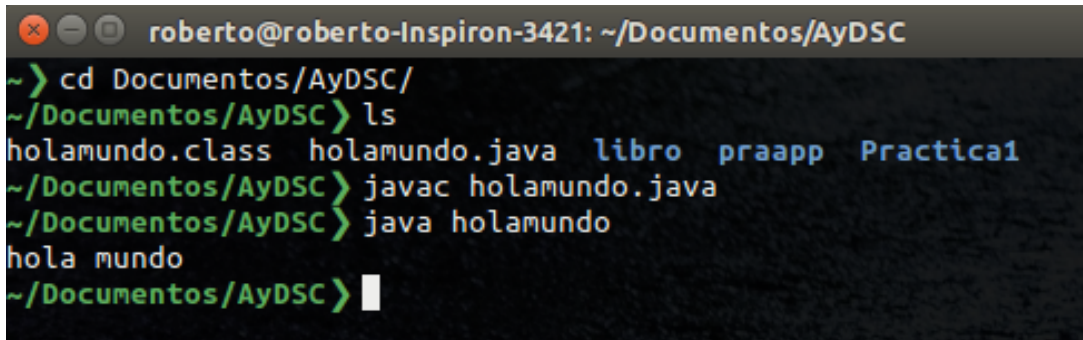
- **Objetivo 1:** Conocerá la manera de compilar y ejecutar un programa en java desde la línea de comando
- **Objetivo 2:** Conocerá el propósito del método main() y la manera en que arrancan los programas en Java
- **Objetivo 3:** Conocerá la manera de crear objetos a partir de una clase.
- **Objetivo 4:** Conocerá la manera de interrogar y modificar el estado de un objeto, y el concepto de relación de dependencia.

## 2 Resultados

### 2.1 Paso 1

Compilación y ejecución de "Hola Mundo" en Java

Como resultado se obtuvo la impresión en la terminal de la frase "Hola mundo"

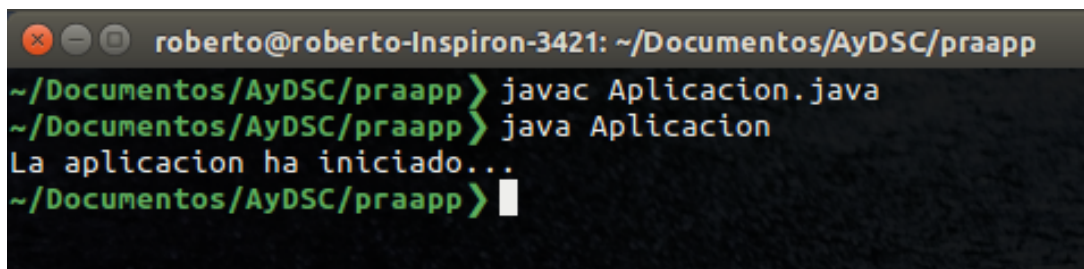


```
roberto@roberto-Inspiron-3421: ~/Documentos/AyDSC
~> cd Documentos/AyDSC/
~/Documentos/AyDSC> ls
holamundo.class holamundo.java libro praapp Practica1
~/Documentos/AyDSC> javac holamundo.java
~/Documentos/AyDSC> java holamundo
hola mundo
~/Documentos/AyDSC> |
```

### 2.2 Paso 2

Creación de un primer objeto

Con la creación de un objeto se mando llamar al método inicia() con el cual se obtuvo la impresión en la terminal del mensaje "La aplicación ah iniciado..."



```
roberto@roberto-Inspiron-3421: ~/Documentos/AyDSC/praapp
~/Documentos/AyDSC/praapp> javac Aplicacion.java
~/Documentos/AyDSC/praapp> java Aplicacion
La aplicacion ha iniciado...
~/Documentos/AyDSC/praapp> |
```

### 2.3 Paso 3

Relación de dependencia y creación de varios objetos

Para este paso se crearon dos clases diferentes, en la clase Book se crearon los siguientes métodos:

Book()  
getName()  
getAuthor()  
setCost()  
getCost()

En clase Aplicacion se creo el método init() el cual se crean tres libros de autores distintos, después se solicitaba mostrar en pantalla la descripción de los libros también se realizo el cambio del precio de uno de los libros llamando al método getCost() y setCost(), por ultimo se creó la condición de que no se pudieran asignar valores negativos al costo de los libros.

```
public class Aplicacion{
    public static void main (String []args){
        Aplicacion app = new Aplicacion();
        // La inicia
        app.init();
    }
}
/**
 * Este metodo es el constructor
 */
public Aplicacion() {
    // No hay nada que inicializar
}
/**
 * Este metodo inicia la aplicación
 */
public void init() {
    //creación de los tres libros
    Book book1 = new Book("Lord of the rings","J.R.R. Tolkien",40.0);
    Book book2 = new Book("Where the wild things are", "Maurice
Sendak", 20.0);
    Book book3 = new Book("Diary of a zombie","Sergi Llauger",30.0);
    //Despligue de todos los libros con su descripción
    System.out.println("Name: " + book1.name + " Author: " +
book1.author + " Cost: " + book1.cost);
    System.out.println("Name: " + book2.name + " Author: " +
book2.author + " Cost: " + book2.cost);
    System.out.println("Name: " + book3.name + " Author: " +
book3.author + " Cost: " + book3.cost);
}
```

```
//Cambio de precio de dos de los libros llamando el precio
actual
    book1.setCost(book1.getCost()+10);
    book2.setCost(book2.getCost()+10);
    //Despligue de todos los libros con precio actualizado
    System.out.println("Name: " + book1.name + " Author: " +
book1.author + " Cost: " + book1.cost);
    System.out.println("Name: " + book2.name + " Author: " +
book2.author + " Cost: " + book2.cost);
    System.out.println("Name: " + book3.name + " Author: " +
book3.author + " Cost: " + book3.cost);
    //Asignación de un precio con valor negativo
    book1.setCost(-50);
}

public class Book{
    //creación de los atributos de la clase Book
    String name;
    String author;
    double cost;

    /**
     * Este metodo es el constructor por defecto
     */
    public Book(){
        //Constructor vacio
    }

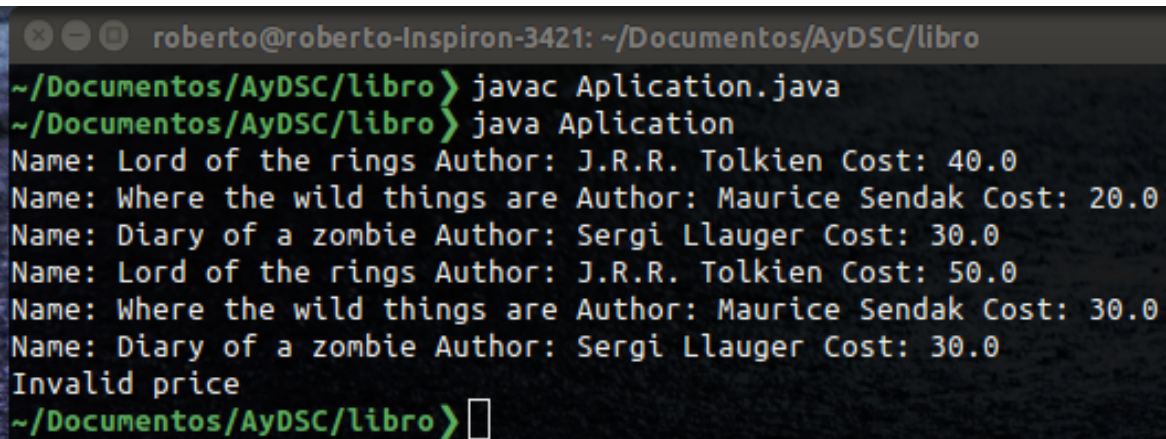
    /**
     * Este metodo es el contructor que recibe todos los parametros para inicializar
     * todos los atributos de la clase
     */
    public Book(String name, String author, double cost){
        this.name = name;
        this.author = author;
        this.cost = cost;
    }

    /**
     * Este metodo devuelve el valor de la variable name
     */
    public String getName(){
        return name;
    }

    /**
     * Este metodo devuelve el valor de la variable author
     */
    public String getAuthor(){
        return author;
    }
}
```

```
/**
 * Este metodo devuelve el valor de la variable cost
 */
    public double getCost(){
        return cost;
    }

/**
 * Este metodo asigna un nuevo valor a la variable cost
 */
    public void setCost(double cost){
        //validación de costos negativos o con valor de 0.0
        if(cost<=0)
            System.out.println("Invalid price");
        else
            this.cost = cost;
    }
}
```



```
roberto@roberto-Inspiron-3421: ~/Documentos/AyDSC/libro
~/Documentos/AyDSC/libro> javac Aplicacion.java
~/Documentos/AyDSC/libro> java Aplicacion
Name: Lord of the rings Author: J.R.R. Tolkien Cost: 40.0
Name: Where the wild things are Author: Maurice Sendak Cost: 20.0
Name: Diary of a zombie Author: Sergi Llauger Cost: 30.0
Name: Lord of the rings Author: J.R.R. Tolkien Cost: 50.0
Name: Where the wild things are Author: Maurice Sendak Cost: 30.0
Name: Diary of a zombie Author: Sergi Llauger Cost: 30.0
Invalid price
~/Documentos/AyDSC/libro> 
```

### 3 Conclusiones

Con la realización del programa del paso 1 se aprendió la instrucción que todo programador debe saber y es la realización del "Hola mundo", a su vez el objetivo 1 es alcanzado con la correcta realización del "Hola mundo".

Se aprendió que en java el método main() es utilizado siempre ya que es por ese método que se invoca a la maquina virtual de Java.

Otro dato importante en la realización de esta practica fue aprender sobre la interacción entre clases y como invocar objetos de una clase dentro de otra clase donde para conseguir esto fue necesario saber como es la creación de un objeto en una clase.

Objetivo	Resultado
Conocer la manera de compilar y ejecutar un programa en java desde la línea de comando	Si se alcanzó,
Conocer el propósito del método Main() y la manera en que arrancan los programas en Java	Si se alcanzó
Conocer la manera de crear objetos a partir de una clase	Si se alcanzó
Conocer la manera de interrogar y modificar el estado de un objeto, y el concepto de relación de dependencia	Si se alcanzó