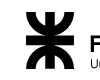
Universidad Tecnológica Nacional

[](http://www.frd.utn.edu.ar/?opc=Home)

Facultad Regional General Pacheco

Tecnicatura Universitaria en Sistemas Informáticos

|  |
| --- |
| Laboratorio de computación V |

Trabajo practico Nro. 1.

Programación Orientada a Objetos

DOCENTES

Prof.: Tamara Herrera

GRUPO N°2

Bellavitis Juan Ignacio

Casmuz Sofía

Martinez David

Moschen Fernando

Savino Christian

|  |
| --- |
| *Fecha de entrega: 02/04/2022* |
|  |

Contenido

[Constructores 3](#_Toc99553985)

[Metodo toString() 5](#_Toc99553986)

[Variables Constantes 5](#_Toc99553987)

[Variables Estáticas 6](#_Toc99553988)

[Herencia 6](#_Toc99553989)

[Clase Abstracta 6](#_Toc99553990)

[Interfaces 7](#_Toc99553991)

[MainClass 8](#_Toc99553992)

[Polimorfismo 11](#_Toc99553993)

# Constructores

Paquete Entidades

Dentro del paquete Entidades podemos encontrar el constructor de la clase Entrada.

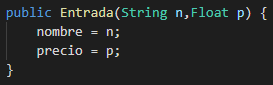


Ilustración 1 Constructor de la clase com.Entidades.Entrada

Este constructor inicializa las variables nombre y precio de dicha clase.

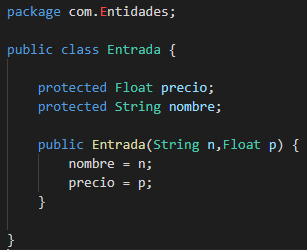


Ilustración 2 Clase com.Entidades.Entrada

Paquete Negocio

Dentro del paquete Negocio podemos encontrar constructores presentes en Deporte, Infantil, Recital y Teatro.

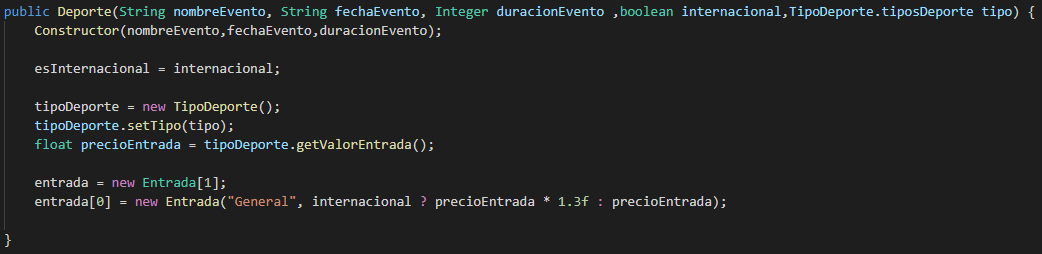


Ilustración 3 Constructor de la clase com.Negocio.Deporte

Aquí como la clase es hija de com.Entidades.Evento inicializa los valores para el nombre del evento indicado en el primer parámetro, fecha del evento como segundo parámetro y duración del evento como tercer parámetro. Los últimos dos parámetros pertenecen a propiedades de la clase Deporte.

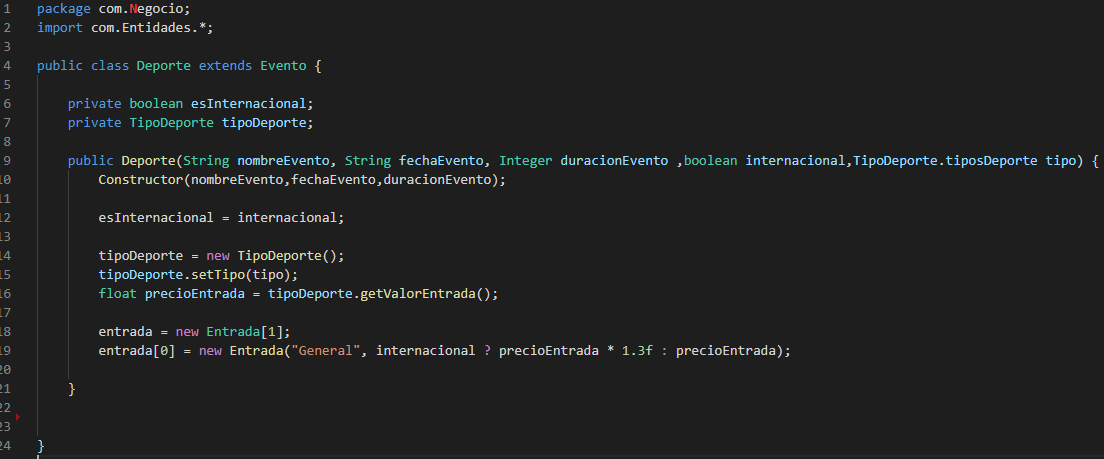


Ilustración 4 Clase com.Negocio.Deporte

En el apartado de clase abstracta se entrará en detalle acerca del método “Constructor”.

**Encapsulamiento**

Paquete Entidades

Texto

Descripción generada automáticamente con confianza baja

Ilustración 8 Encapsulamiento de la clase com.Entidades.Evento

Paquete Negocio

Texto

Descripción generada automáticamente con confianza baja Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ilustración 9 Encapsulamiento de la clase com.Negocio.Deporte

# Metodo toString()

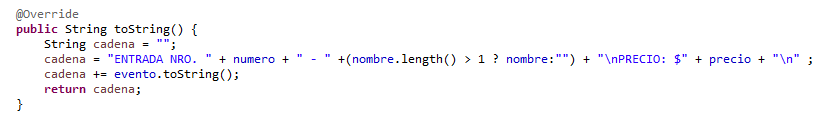


Ilustración 10 Metodo toString de la clase com.Entidades.Entrada

Se concatena con el método toString() de la clase Evento. Este al ser un método abstracto dará como resultado la salida correspondiente a cada tipo de evento que la implementa.

# Variables Constantes

Dentro de las entidades, encontramos el uso de variables constantes en las distintas clases, como por ejemplo en la clase TipoDeporte, donde se utiliza para definir el valor base de la entrada a cada deporte:

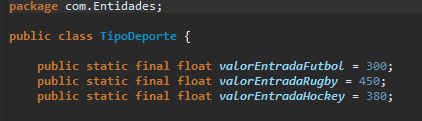


Ilustración 11 Variables contantes de la clase com.Entidades.TipoDeporte

# Variables Estáticas

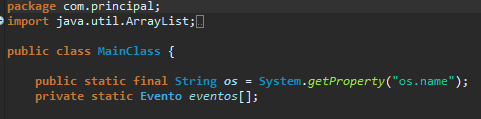


Ilustración 12 Inicialización de Variable estática com.principal.MainClass

# Herencia

En el siguiente ejemplo, la clase Teatro está heredando métodos, atributos y todo lo desarrollado en la clase Evento:

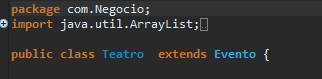


Ilustración 13 Herencia com.Negocio.Teatro

# Clase Abstracta

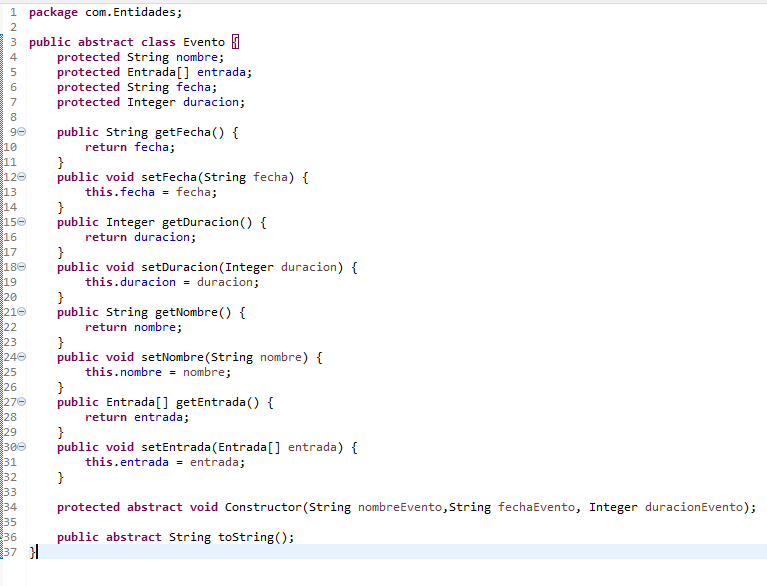


Ilustración 14 Aplicación de clase abstracta en com.Entidades.Evento

# Interfaces



Ilustración 15 Aplicación de inteface en com.interfaces.Menu

# MainClass

Se dividió en 3 partes el Main (MainClass) para mantener un orden en el script.

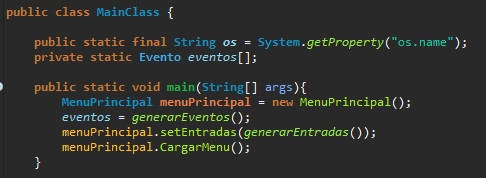


Ilustración 16 MainClass en com.interfaces.Principal

Estas 3 partes son:

1. Generar Eventos:

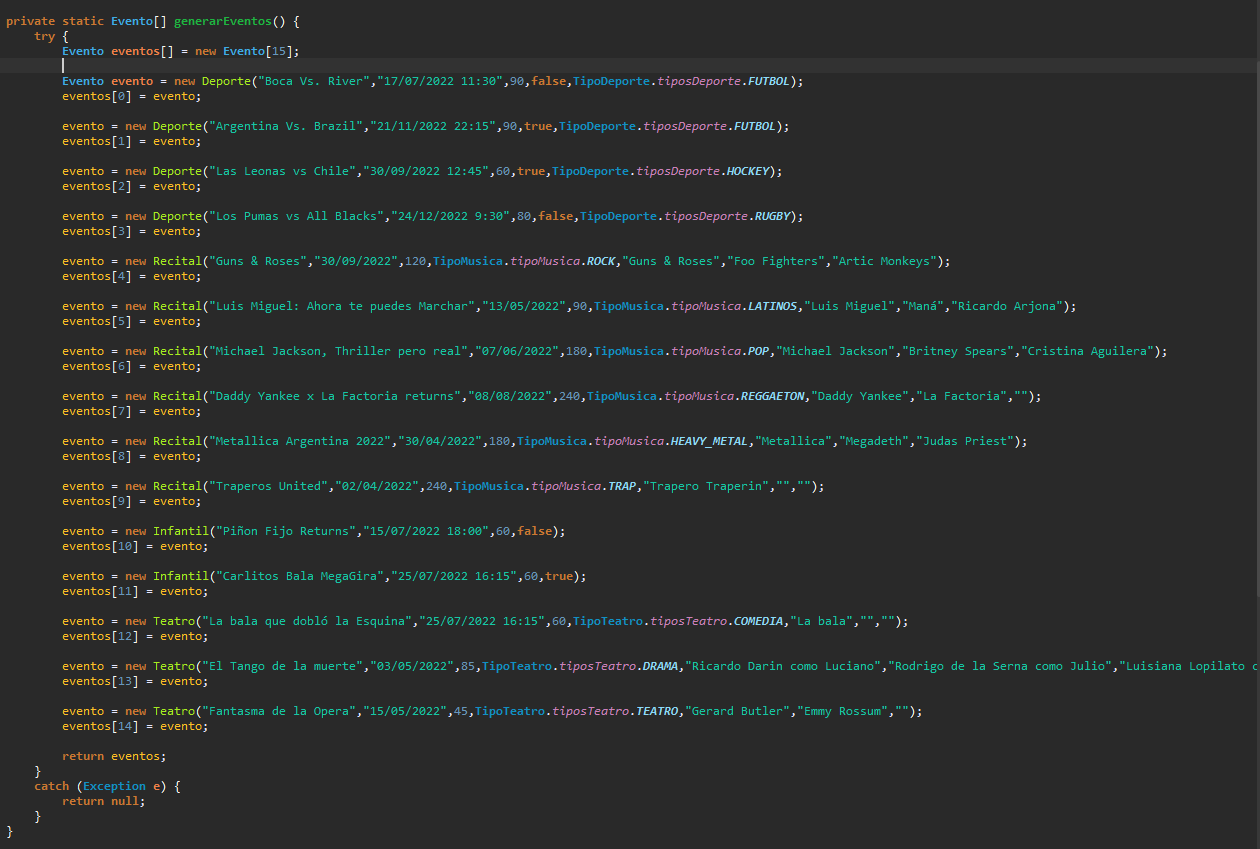


Ilustración 17 MainClass en com.interfaces.Principal

Primero se instancia el objeto MenuPrincipal (heredado de la interface Menu) y antes de hacer la llamada a “Cargar Menu”, se instancian los distintos eventos y sus tipos. Se creó un constructor para cada tipo de evento, donde setea los distintos tipos de entrada que puede llegar a tener (como una referencia). Esto devuelve una array de eventos.

1. Generar Entradas



Ilustración 18 MainClass en com.interfaces.Principal

Este método está encargado de recorrer con un for cada evento creado en el método anterior (GenerarEventos), donde con otro for, se recorre el array de entradas “referencia” para luego copiar el objeto Entrada, setearle el evento actual (del for) y un número único. Por último, la entrada se agrega a una lista, la cual se devuelve.

Al devolver la lista esta se setea en el MenuPrincipal anteriormente instanciado.

1. Imprimir Entradas

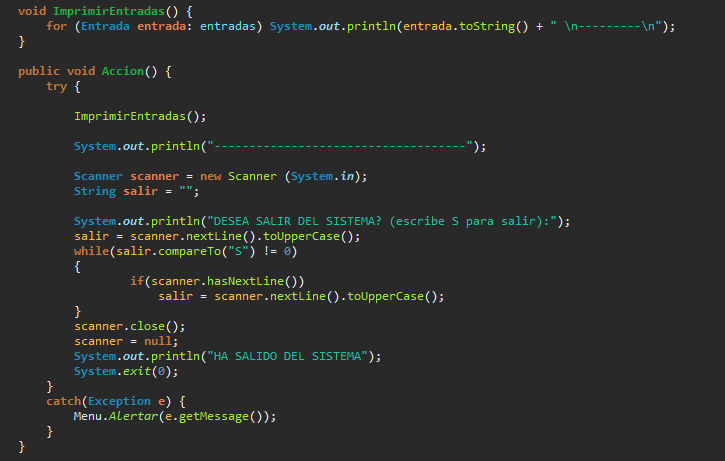


Ilustración 19 MenuPrincipal en com.interfaces.Principal

Al invocar el método “CargarMenu”, este imprime por pantalla una leyenda inicial y luego ejecuta el método “Acción”, donde ejecuta a su vez el método “ImprimirEntradas”. Este método solo contiene un foreach donde realizar un toString() a cada entrada existente en la lista previamente creada.

# Polimorfismo