BodyTic.

**Conocido**

1. Código de la aplicación
   1. Dentro de aplicación no tiene paquetes.
   2. Tiene cinco clases en total.
   3. La clase ejecutiva es Deportista.
2. Aplicación
   1. Funcionalidades
      1. Deportista
         1. inicie ()
         2. mensaje ()
         3. pare ()
      2. Salón
         1. demeSalon ()
         2. cambieSalon (Salon)
         3. adicione (EnSalon)
         4. decisión ()
         5. deme (int)
         6. entrada ()
         7. inicio ()
         8. numeroEnSalon ()
         9. parada ()
         10. salida ()
      3. Persona
         1. getColor ()
         2. getPosicionBrazo (char)
         3. getPosicionPierna (char)
         4. getPosicionX ()
         5. getPosicionY ()
         6. muevaBrazo (char, char)
         7. muevaPierna (char, char)
         8. muévase (char)
   2. Actualmente
      1. Deportista
         1. inicie ()
         2. mensaje ()
         3. pare ()

La interface tiene los tres métodos anteriores por lo cual la clase tiene que implementarlas

* + 1. Salón
       1. demeSalon ()
       2. cambieSalon (Salon)
       3. adicione ()
       4. deme (int)
       5. numeroEnSalon ()

están implementados en la clase

* + 1. Persona
       1. getColor ()
       2. getPosicionBrazo (char)
       3. getPosicionPierna (char)
       4. getPosicionX ()
       5. getPosicionY ()
       6. muevaBrazo (char, char)
       7. muevaPierna (char, char)
       8. muévase (char)

están implementados en el código, además que la mayoría de métodos son finales por lo que si no están implementados en la herencia no puede reescribirlos.

**Arquitectura general**

1. En este proyecto tiene dos paquetes el de presentación y el de aplicación, y en su implementación el de presentación importa todo el paquete de aplicación.
2. Package: un agrupamiento de elementos del proyecto que permite la organización y la separación de funcionalidades del proyecto
3. Los paquetes y los directorios son los mismos así que todos sus archivos coinciden

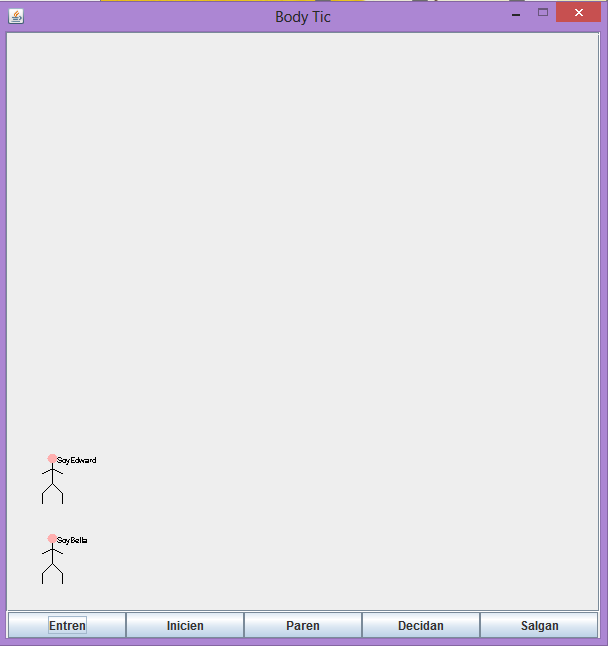
**Arquitectura detallada.**

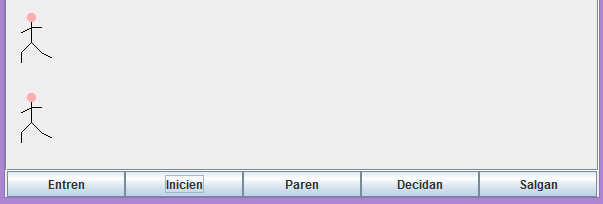
En el diagrama realizado en astah, podemos observar que:

~Los atributos que tienen protected se simbolizan con “#”, los privados con “-“ y los públicos con “+”  
~Los métodos que son estáticos se ven subrayados, los privados con “-“ y los públicos con “+”  
~Para la Interfaz se utiliza la letra itálica, sus métodos que son también abstractos van en letra itálica   
~Las flechas para la herencia es una flecha con la punta triangular sin relleno, para el uso es una flecha punteada y para una relación normal va una flecha sin la punta triangular; Todas siempre indicando la dirección de cómo funcionan.

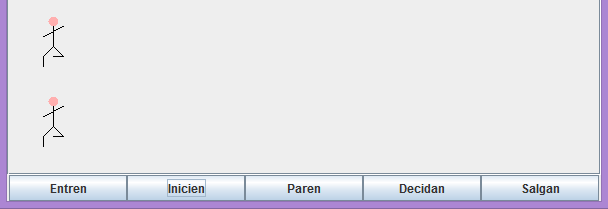
**Ciclo 1. Iniciando con los deportistas normales.**

1. Un ArrayList, No porque nunca tiene una interacción directa a la clase deportista
2. \*Es de color negro  
   \*Dice las palabras: "Soy"+nombre de la persona, "¡Uff!" y nada (“”).  
   \*Inician: sin palabras, aumenta el número de pasos, los brazos pueden estar abajo, arriba o alfrente.  
   \*Paran: baja ambos brazos y los pies los deja en el piso, adicionalmente sus palabras cambian a “¡Uff!”  
   \*Deciden: Si se refiere a la clase de “puedeMoverse” inicia diciendo que es falso, obtiene sus posiciones en x y en y, de acuerdo a la dirección mira si la posición futura esta entre 0 y el máximo del salón y puede moverse, sino no se mueve.
3. Método entren()

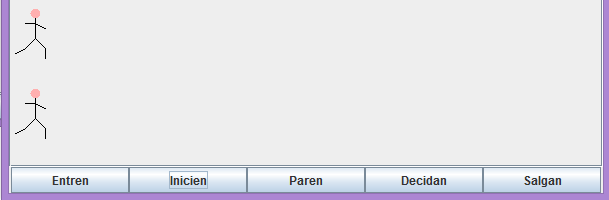


1. Inicien Primer clic  
   

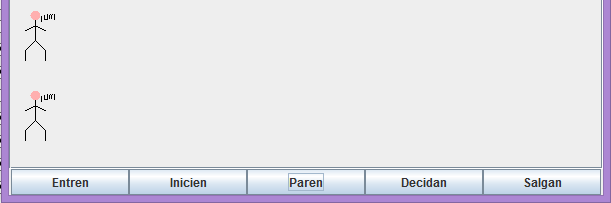
Segundo clic

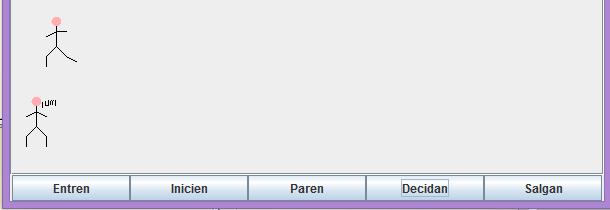
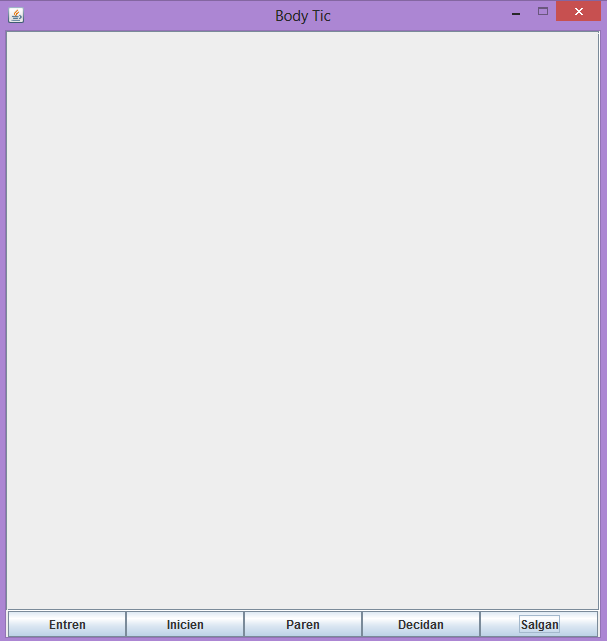


Tercer clic



1. Paren

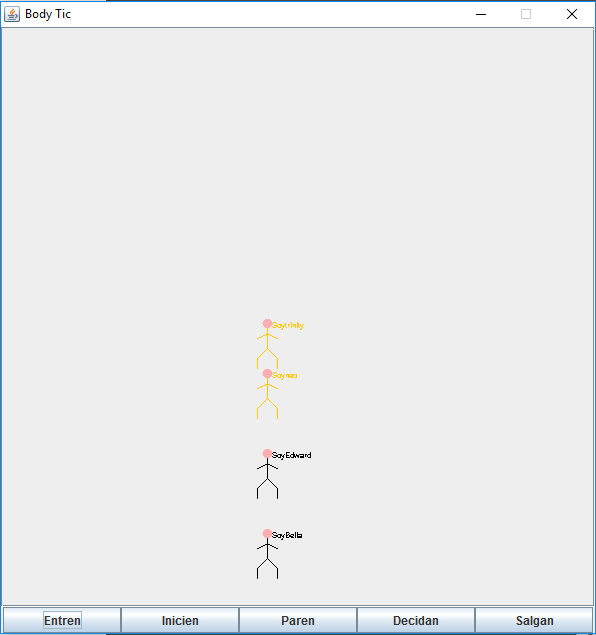


1. 
2. 

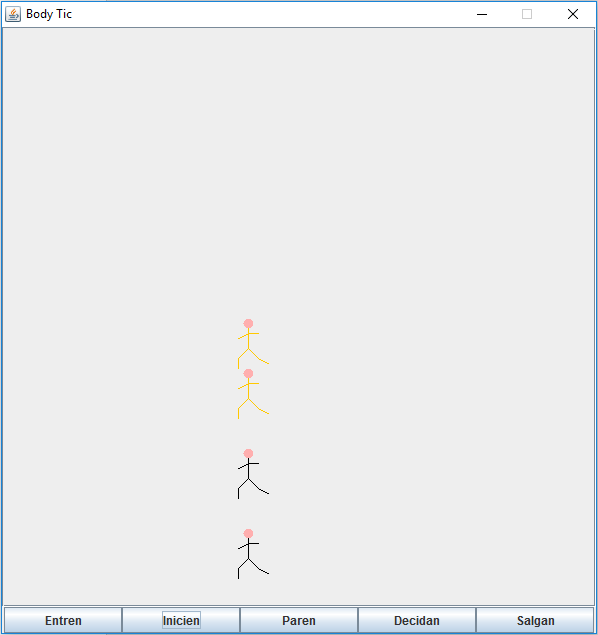
**CICLO 2**

1. Sobrescribimos los métodos:
   1. inicie()
   2. pare()
   3. decida()

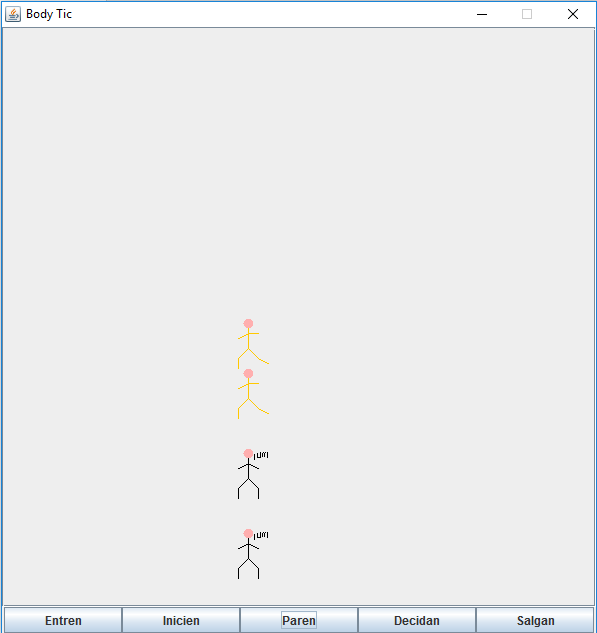
Entren



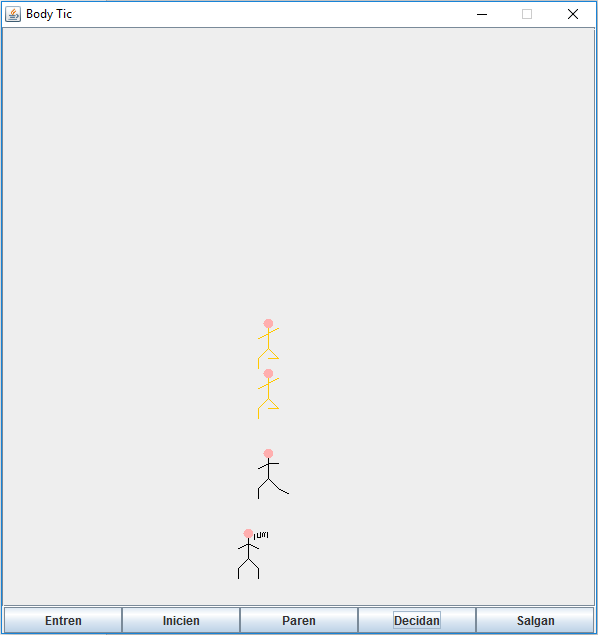
Inicien (un solo clic)



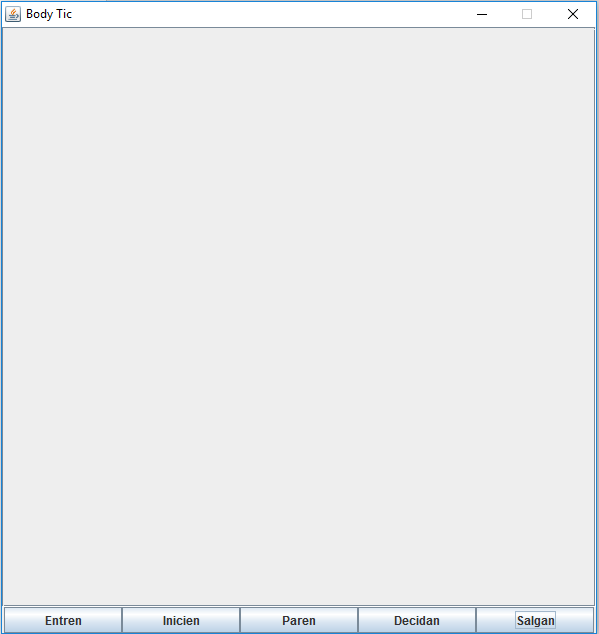
Paren



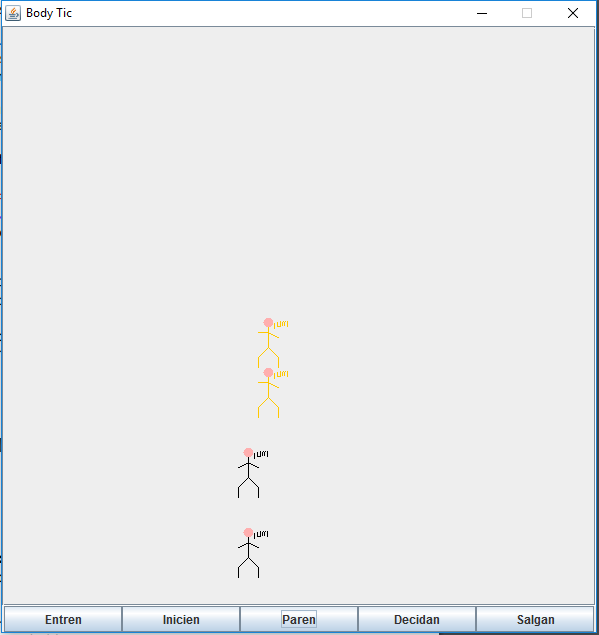
Decidan



Salgan



1. Modificaríamos el método pare() para que no realicen la acción cada dos órdenes sino cada mas ordenes
2. Acción Pare() realizada después de 10 ordenes.



**Ciclo 3**