

Algorítmica y Programación por Objetos 1

Ejercicio de Nivel 3

Álbum

Descripción global

Se desea construir una aplicación que permita simular un álbum con equipos históricos de mundiales de fútbol. La aplicación permite agregar equipos, modificar la información de los jugadores, pegar láminas de distintos tipos, ver la ficha técnica de los jugadores y conocer algunas estadísticas.

Objetivos del ejercicio

En el presente ejercicio usted trabajará en los siguientes objetivos:

- Crear una nueva clase completa.
- Declarar e inicializar atributos.
- Declarar e inicializar constantes.
- Declarar e inicializar enumeraciones.
- Crear e implementar métodos.
- Utilizar estructuras condicionales.
- Utilizar estructuras repetitivas haciendo uso de los patrones de recorrido.
- Crear y manipular estructuras contenedoras de tamaño fijo.
- Crear y manipular estructuras contenedoras de tamaño variable.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio.

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

Preparación

Esta sección presenta una lista de chequeo de todas las tareas necesarias para la preparación del ejercicio. Por favor, revise que cada tarea haya sido completada **antes** de pasar a la siguiente sección de esta guía de trabajo.

Nota: En el siguiente enlace se encuentran las instrucciones de instalación de Java y Eclipse: <https://cupitaller.uniandes.edu.co/instaladores/>.

1. Para conocer el funcionamiento esperado de la aplicación, descargue y/o ejecute el archivo demo del ejercicio que se encuentra en el siguiente enlace: <https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo1/ejercicio-n3>.
2. Descargue el esqueleto del ejercicio que se encuentra en el siguiente enlace: <https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo1/ejercicio-n3>.

3. Descomprima este archivo e importe el proyecto llamado **n3_album** a Eclipse.

El siguiente video explica detalladamente la forma en la cual se descarga, descomprime e importa a Eclipse, para Windows (en el enlace: <https://youtu.be/4PDAn55uNTI>) y para MAC (en el siguiente enlace: <https://youtu.be/XIkMktjIN6c>). Es muy importante que consulte este video, porque lo explicado allí se realizará a lo largo de todo el semestre en el curso y usted deberá hacerlo con especial agilidad en todos los laboratorios y parciales prácticos.

No se preocupe si algunas clases le reportan errores (como marcas rojas) al importar el ejercicio por primera vez. Estos desaparecerán cuando termine correctamente los cambios requeridos en el mundo.

Finalmente, limpie el ejercicio, para ello en Eclipse vaya a: **Project > Clean > Ok**, en el siguiente enlace hay un video con un ejemplo de cómo limpiar un proyecto en Eclipse: <https://youtu.be/mbcpY46wXS0>.

4. Lea el enunciado del problema del ejercicio disponible en:

n3_album/docs/specs/Descripcion.pdf

5. Lea el documento de requerimientos funcionales disponible en:

n3_album/docs/specs/RequerimientosFuncionales.pdf

6. Explore el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en:

n3_album/docs/specs/ModeloConceptual.png.

Usted debería estar en capacidad de identificar las clases, relaciones entre clases, los métodos y los atributos.

7. Lea el documento de consideraciones adicionales de diseño disponible en:

n3_album/docs/specs/ConsideracionesAdicionalesDeDiseno.pdf

8. Desde Eclipse identifique los elementos del mundo en Java (paquetes, clases, métodos y atributos).

Se recomienda que consulte los siguientes videos:

- <https://youtu.be/KmhDGxPIGwl>. Este video le explicará cómo acceder al explorador de Eclipse.
- <https://youtu.be/j2AQKcB8pG4> Este video le explicará cómo leer los contratos de un método.
- <https://youtu.be/tYxAI0F6s-c>. Este video le explicará cómo listar y acceder directamente a los métodos y atributos de una clase en forma rápida y eficiente.
- <https://youtu.be/Lnu9udEXTew>. Este video le explicará cómo dirigirse a la implementación (declaración) de una clase, método y atributo (objeto). Este

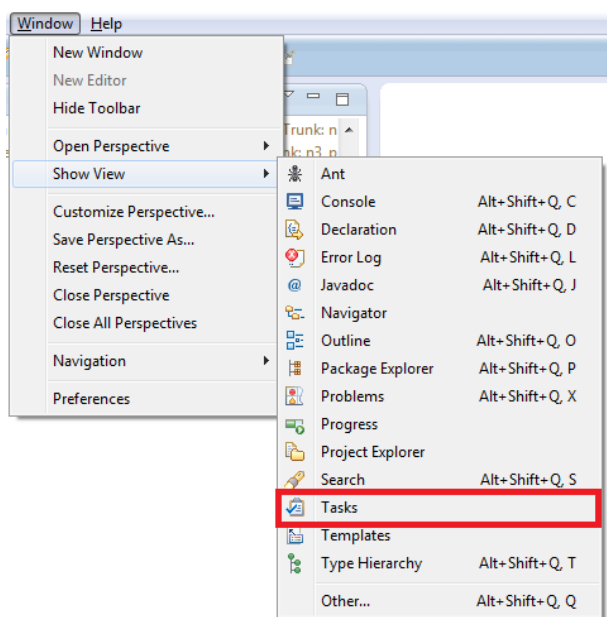
conocimiento le será de gran utilidad para agilizar el desarrollo de su ejercicio y de parciales prácticos.

Desarrollo

En esta sección se explican detalladamente las tareas necesarias para completar el desarrollo del ejercicio. Una vez importado el esqueleto del ejercicio a Eclipse, es necesario identificar los puntos en donde usted deberá realizar alguna modificación puntual (por ejemplo: añadir atributos, completar métodos, etc.). Estos puntos se identifican mediante comentarios especiales denominados: **TODO** (*por hacer*) y tienen la siguiente estructura:

//TODO ParteX PuntoY: Breve explicación de la modificación que se debe realizar.

Donde, **X** hace referencia a una parte de esta guía de trabajo y **Y** hace referencia a una subsección de **X**. Para visualizar y acceder a los TODOs, en Eclipse vaya al menú: **Window > Show View > Tasks**, tal y como se muestra en la siguiente figura.



En el siguiente enlace <https://youtu.be/NnRvintGJAo> encuentra un video que muestra cómo consultar los TODOs de un programa. Esto le permitirá desarrollar el ejercicio en forma ordenada.

Para facilitar el desarrollo de los TODOs, se recomienda ver el siguiente video: <https://youtu.be/xHIRI-yX6HY>, el cual enseña a autocompletar los nombres de clases, métodos y atributos.

Opcionalmente, se recomienda ver el video: https://youtu.be/zYXTlvrOZ_Y, el cual le enseña a renombrar un método o un atributo de forma automática y segura en caso de haber cometido un error en su nombramiento.

Por último, el video disponible en <https://youtu.be/O-dirtGpy50> le enseña cómo determinar en dónde se está usando (se llama) un atributo o un método en un programa.

Por favor recuerde que en este nivel NO debe modificar el paquete de interfaz gráfica (interfaz) ni el paquete de pruebas (test). Así mismo, NO debe modificar ninguna de las firmas de los métodos del mundo, únicamente deberá completar su cuerpo conforme a las instrucciones provistas. En caso de tener dudas sobre el tipo de dato de un atributo o asociación a añadir, por favor revise el modelo conceptual disponible en:

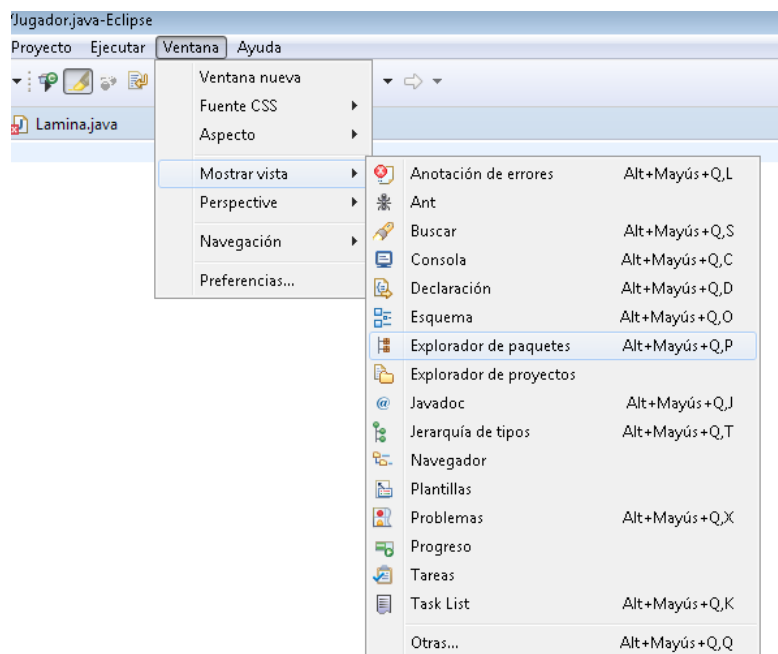
n3_album/docs/specs/ModeloConceptual.png.

Parte 1: Reconocer la clase Lamina

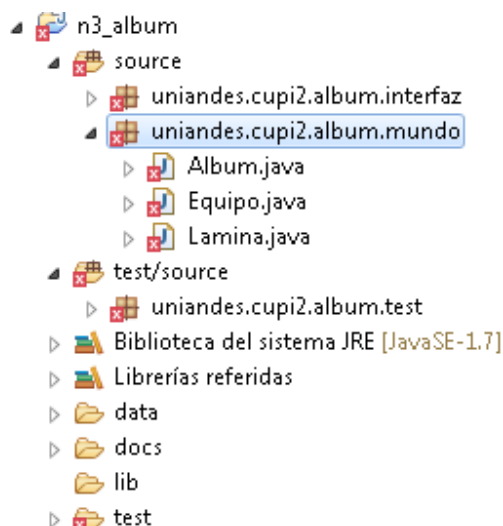
Vaya a la clase Lamina. Revise los métodos existentes y entiéndalos.

Parte 2: Crear la clase Jugador

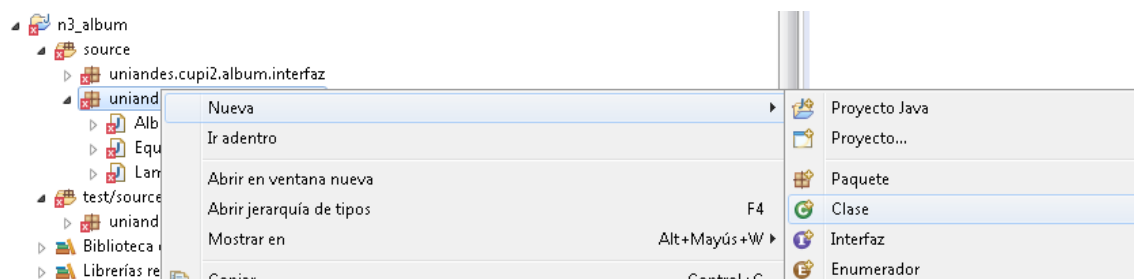
1. Cree la clase **Jugador**. Para crear una nueva clase en Eclipse, tenga en cuenta las siguientes instrucciones:
 - a. Seleccione la vista **Explorador de paquetes** del menú **Ventana**, opción **Mostrar vista**.



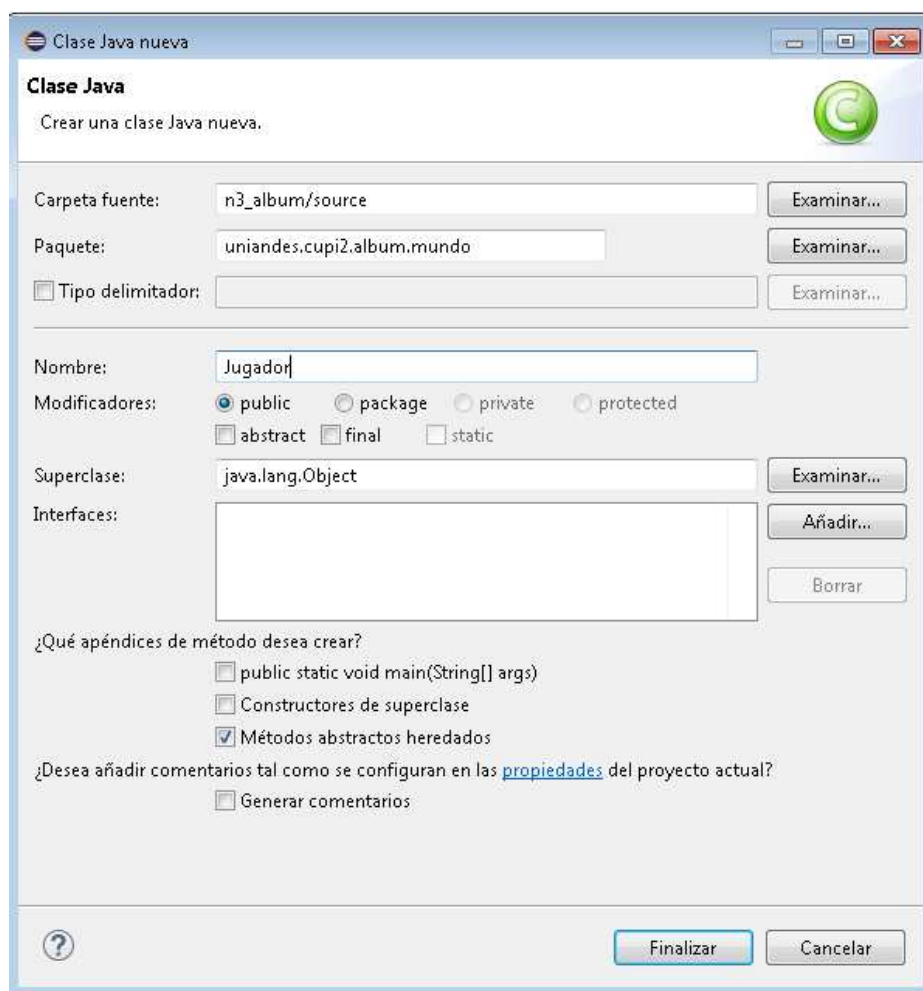
- b. Busque el paquete **uniandes.cupi2.album.mundo** dentro de la carpeta **source**.



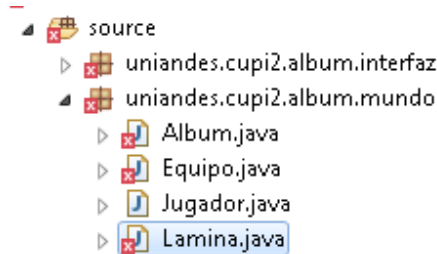
- c. Haga clic derecho sobre este paquete, y selecciona la opción **Nueva > Clase**



- d. En la ventana que aparece, digite el nombre de la nueva clase (Jugador) y presione el botón **Finalizar**.



- e. La nueva clase **Jugador** aparecerá en el paquete del mundo.



2. Adicione el enumerador de la clase. Para ello utilice el modelo del mundo (ModeloConceptual.png).
3. Adicione los atributos de la clase. En el modelo del mundo (ModeloConceptual.png) se encuentran los nombres y tipos de estos atributos.
4. Adicione el método constructor. En el modelo del mundo (ModeloConceptual.png) se encuentran los nombres y tipos de los parámetros que debe recibir el constructor. También se puede hacer uso de la siguiente documentación:

```
/**
 * Crea un nuevo jugador.<br>
 * <b>post:</b> Se inicializaron los atributos numeroCamiseta, nombre,
 * posicion, anioNacimiento, altura y peso con los valores recibidos por parámetro.
 * <br>
 * @param pNumeroCamiseta Número de la camiseta del jugador. pNumeroCamiseta
 * > 0.
 * @param pNombre Nombre del jugador. pNombre != null && pNombre != "".
 * @param pPosicion Posición del jugador. pPosicion pertenece a
 * {Posicion.ARQUERO, Posicion.VOLANTE, Posicion.DEFENSA, Posicion.DELANTERO}.
 * @param pAnioNacimiento Año de nacimiento del jugador. pAnioNacimiento >
 * 0.
 * @param pAltura Altura del jugador en metros. pAltura > 0.
 * @param pPeso Peso del jugador en kilogramos. pPeso > 0.
 */
```

5. Adicione los métodos según el modelo del mundo:
 - darNumeroCamiseta
 - darNombre
 - DarPosicion
 - darAnioNacimiento
 - darAltura
 - darPeso
 - darLamina
 - modificar. Para este método puede hacer uso de la siguiente documentación:

```
/**
 * Cambia el jugador
 * @param pNumeroCamiseta Nuevo número de camiseta del jugador.
 * @param pNombre Nuevo nombre del jugador. pNombre != null && pNombre != "".
```

```
* @param pPosicion Nueva posición del jugador. pPosicion != null &&
pPosicion pertenece a {Posicion.ARQUERO, Posicion.DEFENSA, Posicion.VOLANTE,
Posicion.DELANTERO,
*     Posicion.DESCONOCIDA}.
* @param pAnioNacimiento Nuevo año de nacimiento del jugador.
pAnioNacimiento > 0.
* @param pAltura Nueva altura del jugador. pAltura > 0.
* @param pPeso Nuevo peso del jugador. pPeso > 0.
*/
```

- pegarLamina. Para este método puede hacer uso de la siguiente documentación:

```
/**
 * Pega la lámina del jugador al álbum.<br>
 * <b>post: </b>Si no existía la lámina, se creó la lámina de tipo JUGADOR,
 * con el nombre de imagen correspondiente.
 * @return True si pegó la lámina, false si la lámina ya estaba pegada.
 */
```

Parte 3: Completar la clase Equipo

Completar los TODOs que comienzan por: **//TODO Parte3 PuntoY**

Para completar el constructor de la clase, utilice la información que se encuentra en documento de consideraciones adicionales de diseño (ConsideracionesAdicionalesDeDiseno.pdf).

Parte 4: Completar la clase Album

Completar los TODOs que comienzan por: **//TODO Parte4 PuntoY**

Validación

En esta sección se explica cómo validar el desarrollo del ejercicio. Para comprobar el completo y correcto funcionamiento de su ejercicio usted debe:

1. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz. Los resultados obtenidos al ejecutar el programa deben ser iguales a los del video demo, disponible en el [sitio web del curso](#).
2. Ejecutar las pruebas automáticas disponibles en el ejercicio. En el siguiente video: <https://youtu.be/rVd4AD8XMJk> se explica cómo efectuar esas pruebas. Estas pruebas deben presentar resultados en verde (0 errores y 0 fallas) cuando el ejercicio ha sido completado correctamente.

Tenga en cuenta que esas pruebas **no son exhaustivas** y que su correcto funcionamiento no garantiza que no haya ningún error en su programa.

Entrega

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

1. Indente el código fuente de todas las clases del mundo. En el siguiente enlace <https://youtu.be/cwQ9QiauaSc> encuentra un video que explica cómo indentar el código fuente de su ejercicio.
2. Limpie el proyecto para que la entrega no contenga archivos ejecutables ni temporales (<https://youtu.be/mbcpY46wXS0>).
3. Construya el archivo entregable con su ejercicio desarrollado y validado completamente. En el siguiente video <https://youtu.be/xuSDFfEZW78> se explica detalladamente el proceso para producir el comprimido del ejercicio y enviarlo a SicuaPlus. Renombre el archivo a entregar con su login de la siguiente forma:

n<nivel del ejercicio>_<login estudiante>.zip

Por ejemplo: ***n3_tsuarez.zip***

La no indentación del código fuente o el nombramiento incorrecto del ejercicio en su entrega es una acción penalizada en la plantilla de calificación del mismo.

4. Entregue el archivo del ejercicio vía SicuaPlus, de acuerdo con las normas, fecha y hora de entrega.