

## Algorítmica y Programación por Objetos 1

### Ejercicio Nivel 5

### Sudoku

#### Descripción global

El Sudoku es un juego muy popular que ayuda a desarrollar habilidades lógicas. Fue planteado por primera vez a finales de la década de los 70 y popularizado en el mundo entero a finales del año 2004. Este juego se ha difundido rápidamente debido a sus simples reglas y a que puede ser jugado por cualquiera proporcionando un reto interesante.

#### Objetivos del ejercicio

En el presente ejercicio usted trabajará en:

- Diseño e implementación de una interfaz gráfica de usuario.

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

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

#### Metodología

El juego Sudoku se desarrollará en dos fases:

- En la primera fase (nivel 5), se desarrollará una primera versión de la interfaz gráfica del juego. El esqueleto entregado al estudiante no tiene el paquete de la interfaz, y en el paquete del mundo tiene algunas clases que permitirán que la aplicación tenga una funcionalidad limitada del juego.
- En la segunda fase (nivel 6), el estudiante deberá completar el ejercicio que empezó a desarrollar en la primera fase, cambiando el mundo y la interfaz para que se garantice que el juego funcione de la manera como se describe en el documento de descripción y el documento de requerimientos funcionales.

En el ejercicio desarrollado en el nivel 5, se encontrarán muchos métodos en el mundo que retornarán un valor por defecto o aleatorio, sin que se hagan las validaciones o cálculos necesarios para que se cumpla la funcionalidad del juego, pero que permitirán que haya una interacción entre la interfaz y el mundo.

De igual manera, aunque al finalizar el nivel se tendrá una interfaz 100% funcional, algunos elementos de la interfaz solo podrán ser desarrollados en la segunda fase. Podrá ver qué debe hacer la interfaz en este nivel en el demo del ejercicio.

## 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.

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-n5>.
2. Descargue el esqueleto del ejercicio (<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo1/ejercicio-n5>).
3. Descomprima este archivo e importe el proyecto llamado **n5\_sudoku** 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>).

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>.

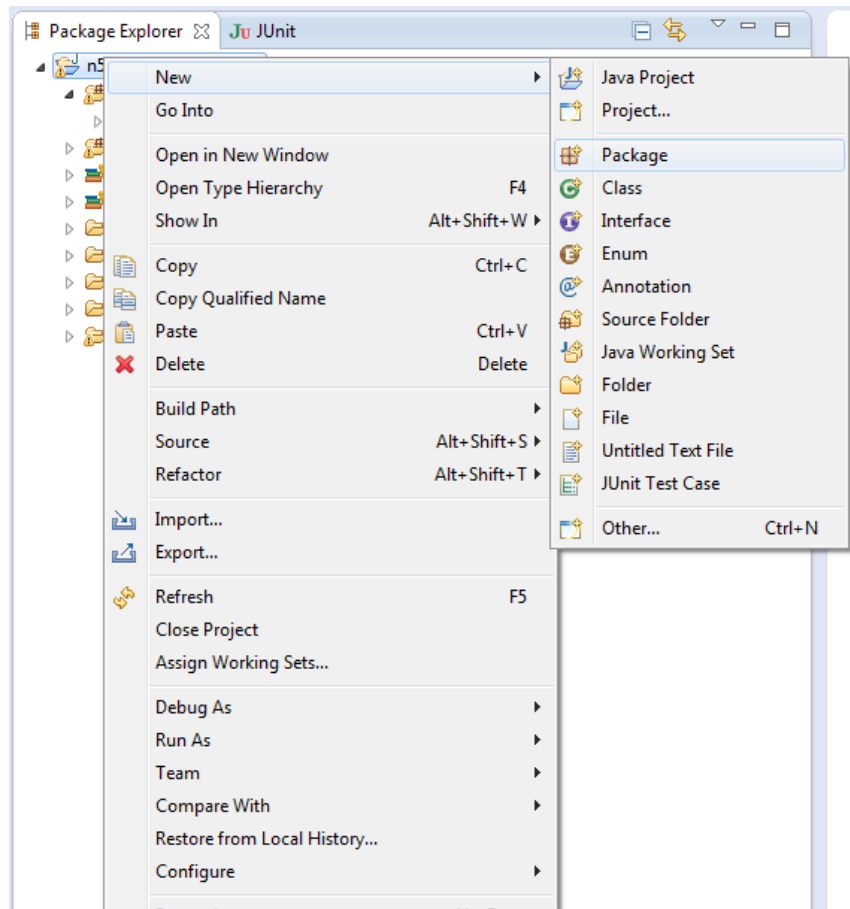
4. Lea el enunciado del problema disponible en:  
**n5\_sudoku/docs/specs/Descripcion.pdf.**
5. Lea el documento de requerimientos funcionales disponible en:  
**n5\_sudoku/docs/specs/RequerimientosFuncionales.pdf.**  
Identifique los requerimientos funcionales del ejercicio.
6. Explore el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en:  
**n5\_sudoku/docs/specs/ModeloConceptual.png.**  
Identifique las clases, relaciones entre clases, atributos y métodos.
7. Revise el proyecto Java desde eclipse. Identifique los elementos del mundo (clases, atributos y métodos).

Se recomienda que consulte los siguientes videos:

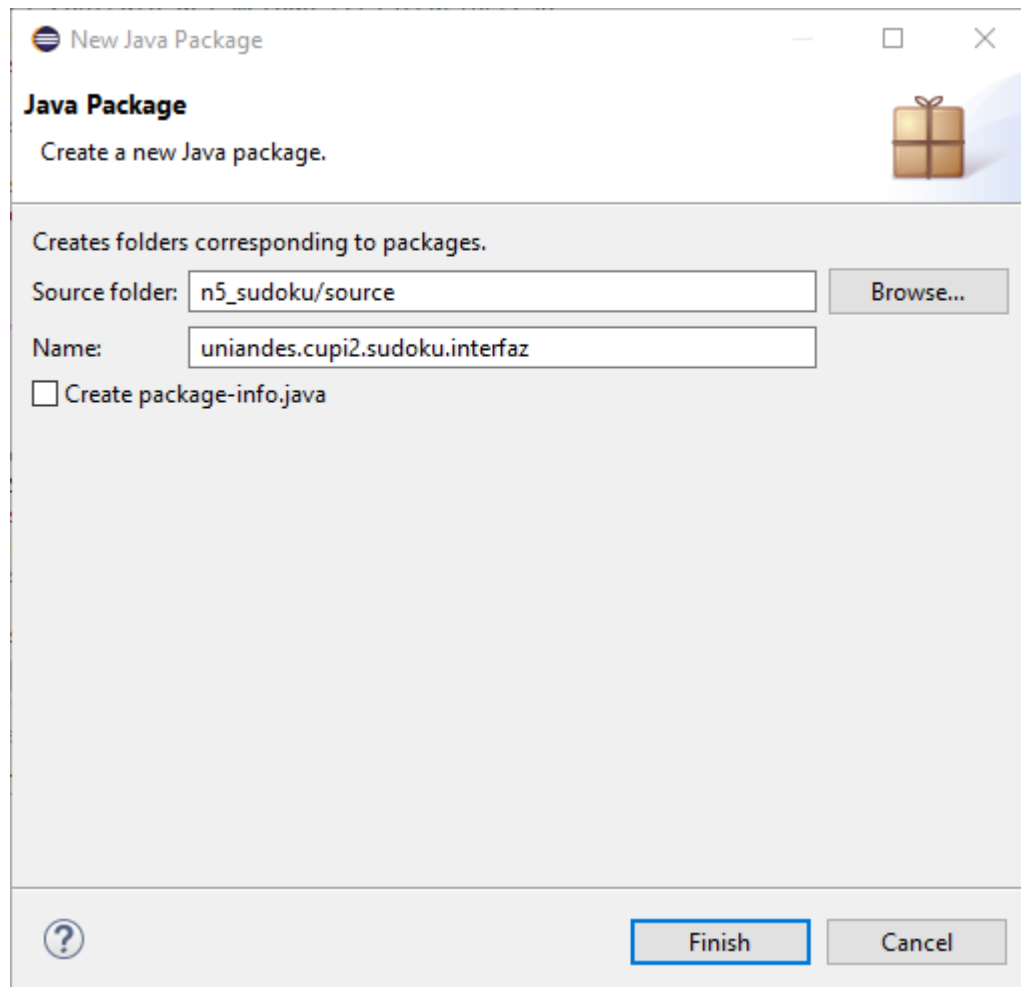
- <https://youtu.be/KmhDGxPIGwl>. Este video le explicará cómo acceder al explorador de Eclipse.
  - <https://youtu.be/tYxAl0F6s-c>. Este video le explicará cómo listar y acceder directamente a los métodos y atributos de una clase en forma rápida y eficiente.
  - <https://youtu.be/Lnu9udEXTew>. Este 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.
8. Desde Eclipse revise la documentación de las clases del mundo. Esto le permitirá entender para qué sirve cada método y cada uno de los atributos.
9. Revise el archivo que está disponible en:  
**n5\_sudoku/docs/specs/ConsideracionesAdicionalesDeDisenho.pdf**, que le dará sugerencias para la implementación gráfica.

## Parte 1: Creación de la interfaz gráfica

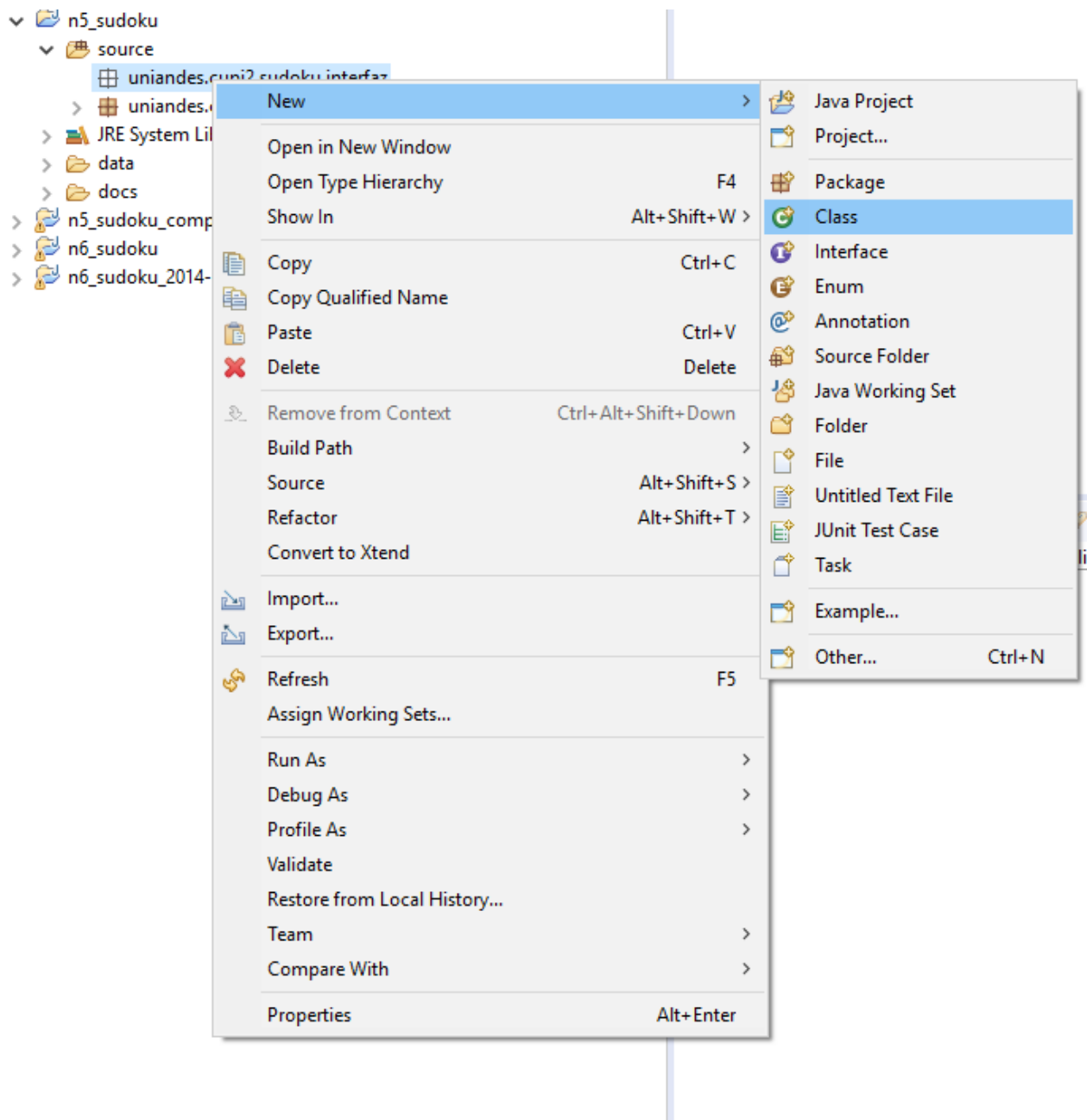
- El objetivo de este ejercicio es que usted cree la interfaz gráfica de la aplicación **n5\_sudoku**. El modo como debe lucir dicha Interfaz puede encontrarlo en el documento **Descripcion.pdf**.
- No puede utilizar herramientas de generación de interfaces para construir la interfaz gráfica. Las clases que componen la interfaz deben construirse según la arquitectura vista en el curso.
- Las clases de la interfaz, junto con sus atributos y métodos, **deben estar documentados**.
- Las clases de la interfaz deben quedar ubicadas en el paquete `uniandes.cupi2.sudoku.interfaz`. Para crear el paquete siga los siguientes pasos:
  1. Ubíquese en la carpeta source, haga clic derecho y seleccione la opción New del menú contextual. Luego seleccione Package.



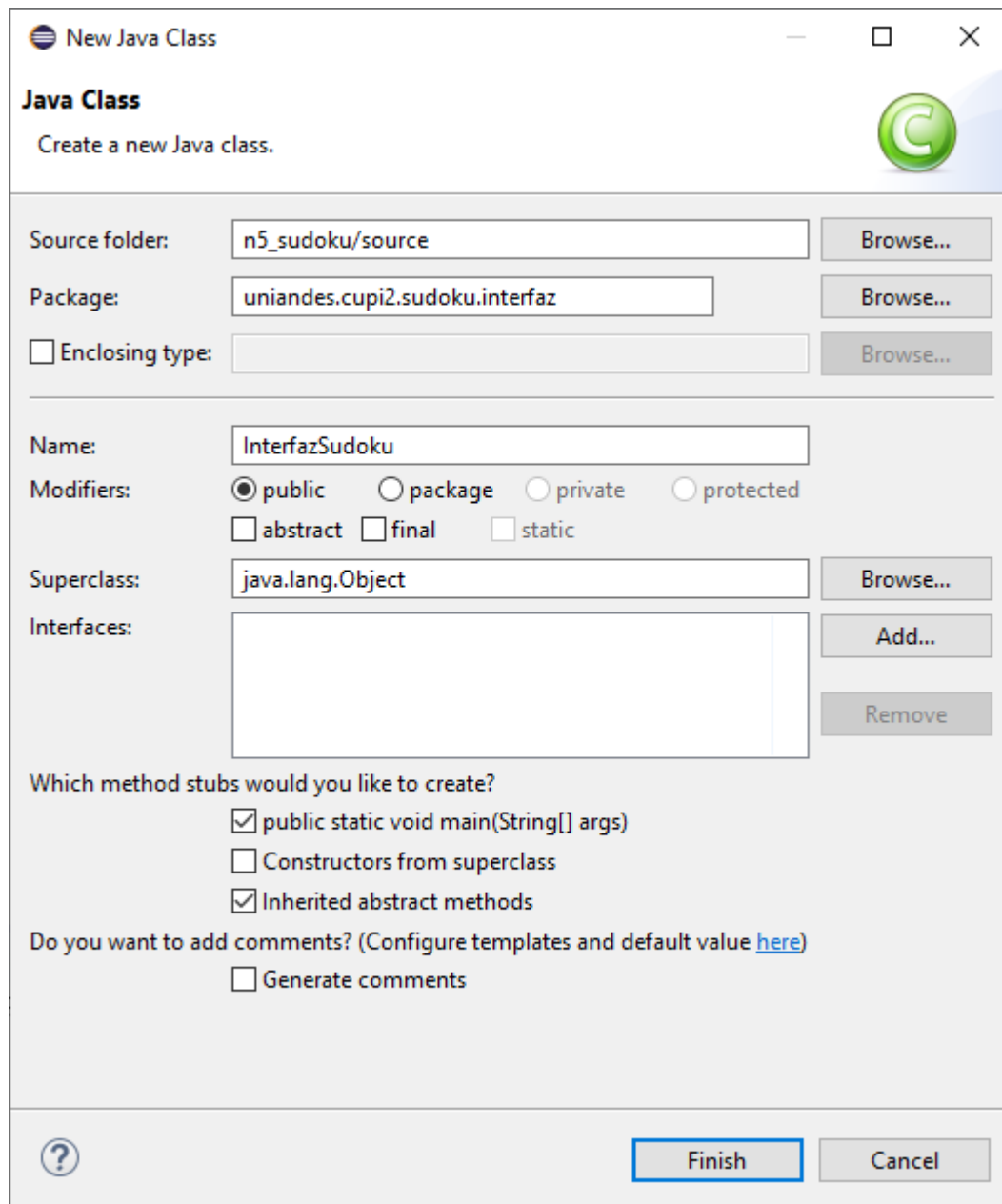
2. En el cuadro de diálogo ingrese el nombre del paquete:  
uniandes.cupi2.sudoku.interfaz.



- Para crear la clase principal de la interfaz (la que corresponde a la ventana principal de la aplicación) siga las siguientes instrucciones:
  1. Ubíquese en el directorio uniandes.cupi2.sudoku.interfaz, haga clic derecho y seleccione la opción New del menú contextual. Luego seleccione Clase.



2. En el cuadro de diálogo ingrese el nombre de la clase: InterfazSudoku (nombre obligatorio) y seleccione la opción para crear de forma automática el método `main`.



**New Java Class**

**Java Class**  
Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?  
☒ public static void main(String[] args)  
☐ Constructors from superclass  
☒ Inherited abstract methods

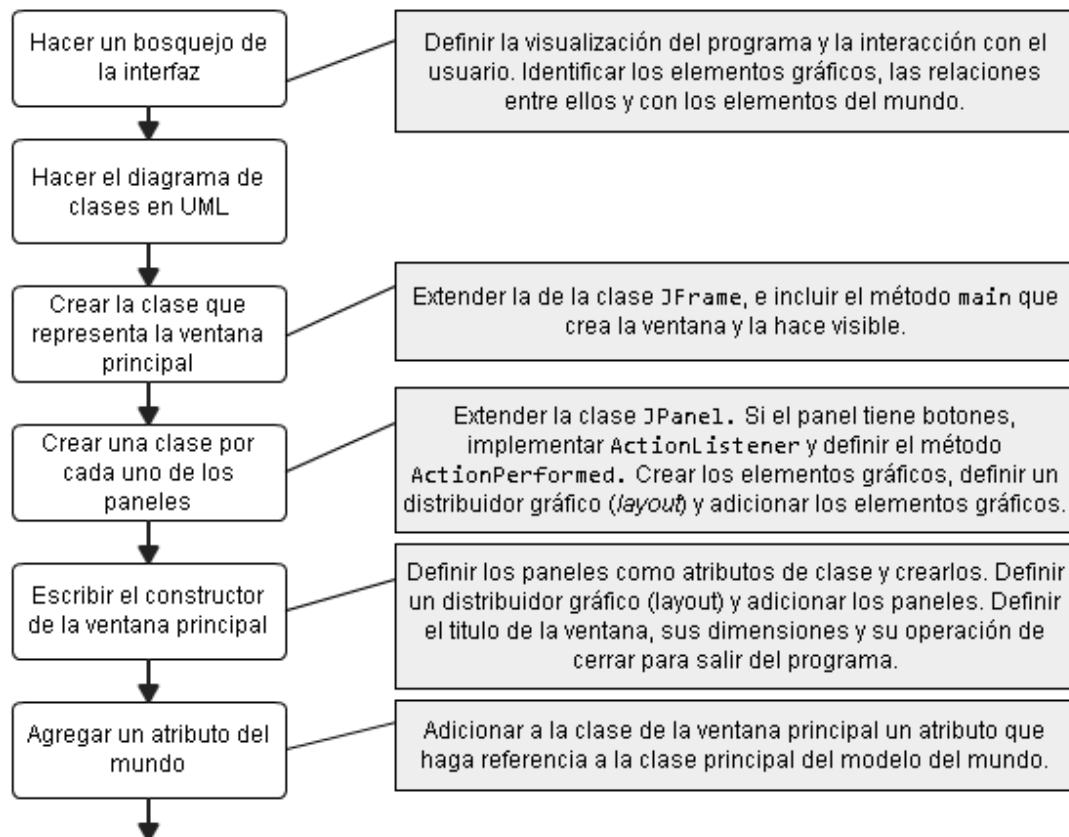
Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

- Juegue con los elementos gráficos vistos en el nivel para obtener la vista esperada. Busque el layout conveniente (BorderLayout o GridLayout) y combínelo con bordes vacíos para acomodar correctamente los elementos.
- Debe incluir en su desarrollo el manejo de extensiones del ejercicio (la parte correspondiente a la interfaz), es decir, los métodos reqFuncOpcion1( ) y reqFuncOpcion2( ). Estos métodos llaman al metodo1() y metodo2() del mundo, respectivamente, y muestran una ventana de diálogo con la respuesta (String) generada por estos métodos.
- La interacción con las clases del mundo, debe establecerla estudiando las clases y los métodos correspondientes.
- Debe usar la arquitectura vista en el curso, definiendo las responsabilidades que debe tener cada clase.

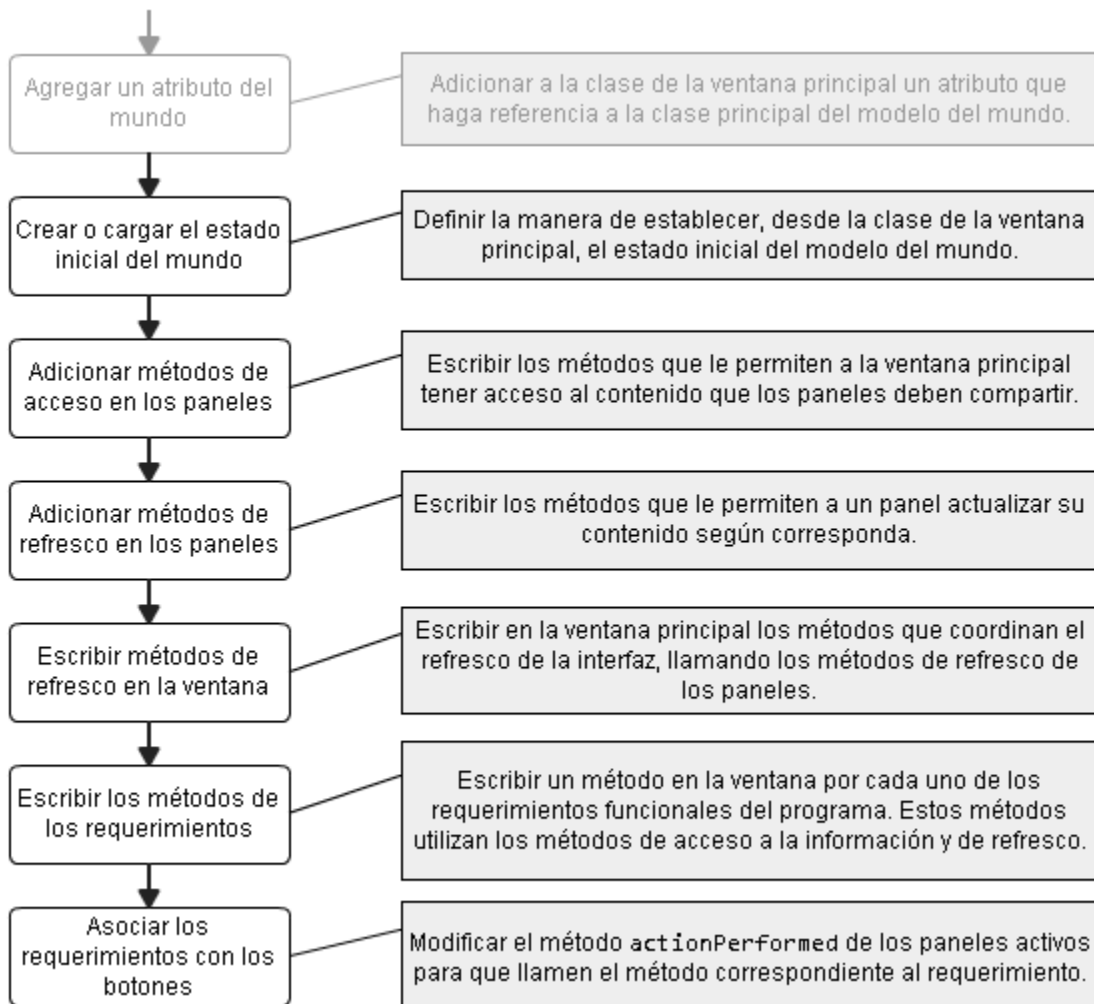


## Parte 2: Creación de la interfaz gráfica

Proceso sugerido para la creación de la interfaz<sup>1</sup>:



<sup>1</sup> Tomado de Jorge A. Villalobos. (2006). Fundamentos de Programación. Colombia: Pearson Educación de México. Páginas 329-330.



Creando la solución de un problema (interfaz).

Debe documentar todas las clases, constantes, atributos y métodos que cree.

### Parte 3: Javadoc

Genere la documentación del proyecto en formato HTML (Javadoc) siguiendo los pasos que se encuentran en los siguientes videos:

Windows: [https://www.youtube.com/watch?v=vnF\\_NZdR1t4](https://www.youtube.com/watch?v=vnF_NZdR1t4)

Mac: <https://www.youtube.com/watch?v=4ixjDpnEqQc>

## Validación

Para comprobar el funcionamiento de su ejercicio usted puede:

1. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz.
2. Los resultados obtenidos al ejecutar el programa deben ser iguales a los del video demo, disponible en el [sitio web del curso](#).
3. La ventana principal y de los mensajes que debe mostrar están disponibles también en el documento **Descripcion.pdf** (tenga en cuenta que no todos los deberá generar en esta primera fase).

## Entrega

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: **n5\_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.