

[75.07 / 95.02]

## Algoritmos y programación III

Trabajo práctico 2 (enunciado)

Primer cuatrimestre del año 2020

# Índice

<b>1. Objetivo</b>	<b>3</b>
<b>2. Consigna general</b>	<b>3</b>
<b>3. Especificación de la aplicación a desarrollar</b>	<b>3</b>
3.1. Entidades	3
Jugador:	3
Panel:	3
Preguntas:	3
Multiplicadores:	5
Exclusividad de puntaje:	5
3.2. Flujo del programa	5
<b>4. Interfaz gráfica</b>	<b>5</b>
<b>5. Herramientas</b>	<b>6</b>
<b>6. Entregables</b>	<b>6</b>
<b>7. Formas de entrega</b>	<b>7</b>
<b>8. Evaluación</b>	<b>7</b>
<b>9. Entregables para cada fecha de entrega</b>	<b>7</b>
Entrega 0 (21/7 o 23/7 según ayudante asignado)	7
Entrega 1 (28/7 o 30/7 según ayudante asignado)	8
Entrega 2 (4/8 o 6/8 según ayudante asignado)	8
Entrega 3 (11/8 o 13/8 según ayudante asignado)	8
Entrega 4 - Final: (18/8 o 20/8 según ayudante asignado)	8
<b>10. Informe</b>	<b>9</b>
Supuestos	9
Diagramas de clases	9
Diagramas de secuencia	9
Diagrama de paquetes	9
Diagramas de estado	9
Detalles de implementación	9
Excepciones	9

# 1. Objetivo

Desarrollar una aplicación de manera grupal aplicando todos los conceptos vistos en el curso, utilizando un lenguaje de tipado estático (Java) con un diseño del modelo orientado a objetos y trabajando con las técnicas de TDD e Integración Continua.

# 2. Consigna general

Desarrollar la aplicación completa, incluyendo el **modelo** de clases e interfaz gráfica. La aplicación deberá ser acompañada por pruebas unitarias e integrales y documentación de diseño.

# 3. Especificación de la aplicación a desarrollar

La aplicación consiste en un juego por turnos, de dos jugadores conformado de un panel en el cual se mostraran preguntas con múltiples opciones de respuesta.

Cada pregunta será mostrada dos veces, una vez para cada jugador (al estar jugando los dos en la misma computadora, será responsabilidad de cada jugador no mirar la pantalla mientras el otro responde). El jugador dispone de un tiempo limitado para responder cada pregunta.

Existen varios tipos de preguntas, que asignan puntaje en forma diferenciada a cada jugador dependiendo de cómo responde cada uno.

También existen opciones como los multiplicadores y la exclusividad de puntaje que cada jugador puede utilizar para mejorar sus oportunidades de obtener puntos.

El objetivo del juego es lograr más puntos que el otro jugador respondiendo correctamente las preguntas.

## 3.1. Entidades

- **Jugador:**

Cada jugador tendrá una oportunidad de responder cada una de las preguntas.

- **Panel:**

Es el panel donde se mostrarán las preguntas junto con las opciones de respuesta, estará dividido en dos partes:

- *Sector pregunta:*

Es el sector en el cual se mostrará el texto de la pregunta.

- *Sector respuestas:*

Es el sector en el cual se presentan las opciones de respuesta, y en el que cada jugador seleccionará la/s opción/es que considere correctas.

- **Preguntas:**

- Existen distintos tipos de preguntas:

<b>Verdadero/Falso clásico</b>	
<b>Opciones</b>	2 (Verdadero o Falso)
<b>Comportamiento</b>	Asigna un punto a cada jugador que responda con la opción correcta.

<b>Verdadero/Falso con penalidad</b>	
<b>Opciones</b>	2 (Verdadero o Falso)
<b>Comportamiento</b>	Asigna un punto a cada jugador que responda con la opción correcta, y resta un punto a cada jugador que responda en forma incorrecta.

<b>Multiple choice clásico</b>	
<b>Opciones</b>	Desde dos y hasta 5
<b>Comportamiento</b>	Asigna un punto a cada jugador que acierte TODAS las opciones correctas.

<b>Multiple choice con puntaje parcial</b>	
<b>Opciones</b>	Desde dos y hasta 5
<b>Comportamiento</b>	Asigna un punto a cada jugador por cada opción correcta que seleccione (aunque no seleccione todas las correctas), siempre y cuando no haya seleccionado ninguna de las opciones incorrectas.

<b>Multiple choice con penalidad</b>	
<b>Opciones</b>	Desde dos y hasta 5
<b>Comportamiento</b>	Asigna un punto a cada jugador por cada opción correcta que seleccione (aunque no seleccione todas las correctas), y resta un punto a cada jugador por cada opción incorrecta que seleccione.

<b>Ordered choice</b>	
<b>Opciones</b>	Desde dos y hasta 5

<b>Comportamiento</b>	Se presentan opciones que deben ordenarse. Asigna un punto a cada jugador que coloque las opciones en el orden correcto.
-----------------------	--

<b>Group choice</b>	
<b>Opciones</b>	Desde dos y hasta 6, a colocar en dos grupos
<b>Comportamiento</b>	Se presentan dos grupos vacíos, y opciones que deben colocarse en alguno de esos dos grupos. Asigna un punto a cada jugador que coloque las opciones en el grupo correcto.

### ● Multiplicadores:

- Cada jugador tendrá dos multiplicadores (un multiplicador "x2" y un multiplicador "x3") que podrá elegir utilizar en cualquiera de las preguntas "con penalidad".
- El jugador podrá asignar el multiplicador a la pregunta cuando le llegue el turno de responder.
- Al calcular los puntos, se multiplicará x2 o x3 tanto los puntos por opción correcta como la penalidad, solo para el jugador que usó el multiplicador.

### ● Exclusividad de puntaje:

- Cada jugador tendrá dos opciones de "exclusividad de puntaje" que podrá elegir utilizar en cualquiera de las preguntas que no tienen "penalidad".
- El jugador podrá asignar la exclusividad de puntaje a la pregunta cuando le llegue el turno de responder.
- Al calcular los puntos, se asignará el doble del puntaje, pero solo en caso de que solo uno de los jugadores haya realizado la opción correcta. Es decir, si los dos jugadores eligen la opción correcta, no se asignará puntaje a ninguno de los dos. Si uno de los jugadores eligió la opción correcta, ese jugador conseguirá el doble del puntaje.
- Alcanza con que uno de los jugadores asigne la exclusividad de puntaje para que la regla afecte a ambos jugadores en esa pregunta.
- Si los dos jugadores asignan exclusividad de puntaje, entonces el efecto se duplica y se asignará el cuádruple de los puntos al jugador que elija la opción correcta (solo si el otro jugador no eligió la opción correcta a su vez).

## 3.2. Flujo del programa

- Fase inicial (una por cada jugador): Cada jugador selecciona su nombre.
- Fase de juego: Cuando ambos jugadores finalizaron su Fase inicial, se inicia esta fase, la cual consiste en mostrar a cada jugador las preguntas con sus opciones. Consta de un turno por vez por usuario..

## 4. Interfaz gráfica

La interacción entre el usuario y la aplicación deberá ser mediante una interfaz gráfica intuitiva. Consistirá en una aplicación de escritorio utilizando **JavaFX** y se pondrá mucho énfasis y se evaluará como parte de la consigna su **usabilidad**.

## 5. Herramientas

1. **JDK (Java Development Kit):** Versión 1.8 o superior.
2. **JavaFX**
3. **JUnit:** Framework de pruebas unitarias para Java.
4. **IDE (Entorno de desarrollo integrado):** Su uso es opcional y cada integrante del grupo puede utilizar uno distinto o incluso el editor de texto que más le guste. Lo importante es que el repositorio de las entregas no contenga ningún archivo de ningún IDE y que la construcción y ejecución de la aplicación sea totalmente independiente del entorno de desarrollo. Algunos de los IDEs más populares son:
  - a. [Eclipse](#)
  - b. [IntelliJ](#)
  - c. [Netbeans](#)
5. **Herramienta de construcción:** Se deberán incluir todos los archivos XML necesarios para la compilación y construcción automatizada de la aplicación. El informe deberá contener instrucciones acerca de los comandos necesarios (preferentemente también en el archivo README.md del repositorio). Puede utilizarse Maven o Apache Ant con Ivy.
6. **Repositorio remoto:** Todas las entregas deberán ser subidas a un repositorio único en GitHub para todo el grupo en donde quedarán registrados los aportes de cada miembro. El repositorio puede ser público o privado. En caso de ser privado debe agregarse al docente corrector como colaborador del repositorio.
7. **Git:** Herramienta de control de versiones
8. **Herramienta de integración continua:** Deberá estar configurada de manera tal que cada *commit* dispare la compilación, construcción y ejecución de las pruebas unitarias automáticamente. Algunas de las más populares son:
  - a. Travis-CI
  - b. Jenkins
  - c. Circle-CI

Se recomienda basarse en la estructura del [proyecto base](#) armado por la cátedra.

## 6. Entregables

Para cada entrega se deberá subir lo siguiente al repositorio:

1. Código fuente de la aplicación completa, incluyendo también: código de la prueba, archivos de recursos.
2. Script para compilación y ejecución (Ant o Maven).
3. Informe, acorde a lo especificado en este documento (en las primeras entregas se podrá incluir solamente un enlace a Overleaf o a Google Docs en donde confeccionen el informe e incluir el archivo PDF solamente en la entrega final).

No se deberá incluir ningún archivo compilado (formato .class) ni tampoco aquellos propios de algún IDE (por ejemplo .idea). Tampoco se deberá incluir archivos de diagramas UML propios de alguna herramienta. Todos los diagramas deben ser exportados como imágenes de manera tal que sea transparente la herramienta que hayan utilizado para crearlos.

## 7. Formas de entrega

Habrán **4 entregas formales** que tendrán una calificación de **APROBADO o NO APROBADO** en el momento de la entrega. Además se contará con una entrega 0 preliminar.

Aquel grupo que acumule 2 no aprobados, quedará automáticamente desaprobado con la consiguiente **pérdida de regularidad en la materia de todos los integrantes del grupo**. **En cada entrega se deberá incluir el informe actualizado.**

## 8. Evaluación

El día del vencimiento de cada entrega, cada ayudante convocará a los integrantes de su grupo, solicitará el informe correspondiente e iniciará la corrección mediante una entrevista grupal.

**Es imprescindible la presencia de todos los integrantes del grupo el día de cada corrección.**

Se evaluará el trabajo grupal y a cada integrante en forma individual. El objetivo de esto es comprender la dinámica de trabajo del equipo y los roles que ha desempeñado cada integrante del grupo. Para que el alumno apruebe el trabajo práctico debe estar aprobado en los dos aspectos: grupal e individual (se revisarán los *commits* de cada integrante en el repositorio).

Dentro de los ítems a chequear el ayudante evaluará aspectos formales (como ser la forma de presentación del informe), aspectos funcionales: que se resuelva el problema planteado y aspectos operativos: que el TP funcione integrado.

## 9. Entregables para cada fecha de entrega

Se sobreentiende que cada entrega consta de las **pruebas + el código** que hace pasar dichas pruebas).

### Entrega 0 (21/7 o 23/7 según ayudante asignado)

- Planteo de modelo tentativo, diagrama de clases general y diagrama de secuencia para el caso donde se pide a una pregunta de verdadero/falso clásico que evalúe una respuesta de un jugador y le asigne puntaje.
- Pruebas:
  - Una Pregunta de Verdadero/Falso clásico puede crearse indicándole cual es la respuesta correcta

- Una Pregunta de Verdadero/Falso clásico recibe una lista de respuestas y asigna correctamente puntos a los jugadores que respondieron correctamente

## Entrega 1 (28/7 o 30/7 según ayudante asignado)

- Pruebas:
  - Una Pregunta de Verdadero/Falso con penalidad puede crearse indicándole cual es la respuesta correcta
  - Una Pregunta de Múltiple Choice clásico puede crearse indicándole cuales son las opciones correctas
  - Una Pregunta de Múltiple Choice con puntaje parcial puede crearse indicándole cuales son las opciones correctas
  - Una Pregunta de Verdadero/Falso con penalidad recibe una lista de respuestas y asigna correctamente puntos a los jugadores que respondieron correctamente, y resta correctamente puntos a los jugadores que respondieron en forma incorrecta
  - Una Pregunta de Multiple Choice clasico recibe una lista de respuestas y asigna correctamente puntos a los jugadores que respondieron correctamente
  - Una Pregunta de Multiple Choice con puntaje parcial recibe una lista de respuestas y asigna correctamente puntos a los jugadores según las opciones correctas que hayan respondido

## Entrega 2 (4/8 o 6/8 según ayudante asignado)

- Pruebas similares a las de la entrega 1, con el resto de los tipos de preguntas
- Agregado de multiplicadores a cada tipo de pregunta, verificando que cumplan con el efecto indicado (x2 o x3)
- Planteo inicial de interfaz gráfica, pantalla donde se muestra una pregunta con sus opciones

## Entrega 3 (11/8 o 13/8 según ayudante asignado)

1. Modelo del juego terminado
2. Interfaz gráfica inicial básica: comienzo del juego y visualización del tablero e interfaz de usuario básica.
3. Modelo del manejo de turnos en el juego.



Entrega 4 - Final: (18/8 o 20/8 según ayudante asignado)

**Trabajo Práctico completo funcionando, con interfaz gráfica final e informe completo.**

**Tiempo total de desarrollo del trabajo práctico:**  
**5 semanas**

## 10. Informe

El informe deberá estar subdividido en las siguientes secciones:

### Supuestos

Documentar todos los supuestos hechos sobre el enunciado. Asegurarse de validar con los docentes.

### Diagramas de clases

Varios diagramas de clases, mostrando la relación estática entre las clases. Pueden agregar todo el texto necesario para aclarar y explicar su diseño de manera tal que logre el modelo logre comunicarse de manera efectiva.

### Diagramas de secuencia

Varios diagramas de secuencia, mostrando la relación dinámica entre distintos objetos planteando una gran cantidad de escenarios que contemplen las secuencias más interesantes del modelo.

### Diagrama de paquetes

Incluir un diagrama de paquetes UML para mostrar el acoplamiento de su trabajo.

### Diagramas de estado

Incluir diagramas de estados, mostrando tanto los estados como las distintas transiciones para varias entidades del modelo.

## Detalles de implementación

Deben detallar/explicar qué estrategias utilizaron para resolver todos los puntos más conflictivos del trabajo práctico. Mencionar qué patrones de diseño fueron utilizados y por qué motivos.

## Excepciones

Explicar las excepciones creadas, con qué fin fueron creadas y cómo y dónde se las atrapa explicando qué acciones se toman al respecto una vez capturadas.