

En la unidad 2 de nuestro curso hemos trabajado en el análisis de un problema construyendo un modelo con los elementos que intervienen en el problema y especificando los servicios que el programa debe ofrecer, bajo el paradigma de programación orientado a objetos.

Aprendimos a construir las clases que implementan el modelo de la solución del problema, identificando de manera informal los métodos de una clase y clasificarlos en métodos constructores, de consulta y de modificación. Utilizamos una arquitectura para un programa que permita repartir de manera adecuada las responsabilidades entre la interfaz de usuario y el modelo de la solución, y cómo relacionar dichos componentes. Finalmente, hemos aprendido a relacionar todos los conceptos vistos en las dos primeras unidades del curso.

Esta tarea integradora presenta una actividad en la cual se requiere aplicar todos los conocimientos adquiridos en la unidad 2. Por tanto, esta tarea es un instrumento para verificar el cumplimiento de los objetivos que han sido planteados para la unidad 2 descrita en el programa del curso.

Para llevar a cabo este ejercicio es necesario realizar las actividades listadas a continuación:

## **Actividades**

Lleve a cabo las siguientes actividades de cada una de las etapas de desarrollo de software:

1. Análisis del problema (Tabla de especificación y Lista de requerimientos funcionales en el formato visto en la clase de ingeniería de software 1, [descárguelo aquí](#)).
2. Diseño de la solución.
  - Elabore un diagrama de clases que modele la solución del problema de acuerdo con las buenas prácticas y los patrones de diseño revisados hasta el momento en el curso. Su diagrama debe incluir el paquete modelo y el de interfaz de usuario.
  - Tabla de trazabilidad entre el análisis y el diseño ([Link](#)).
3. Implementación en Java. Incluya en la implementación, los comentarios descriptivos sobre los atributos y métodos de cada clase. Recuerde que todos los artefactos generados en la fase de diseño e implementación deben ser en inglés.
4. Documentación en JavaDoc (Debe entregarse el JavaDoc generado y ubicarlo en la carpeta doc).
5. Usar GitHub como repositorio de código fuente y documentación utilizando la estructura de carpetas aprendida en clase. Recuerde que se debe evidenciar su avance a través de los días en el desarrollo de su tarea.

Recuerde que puede encontrar la Rúbrica de la tarea integradora en el siguiente [link](#).

**Nota:**

- Usted debe entregar la URL de su repositorio GitHub donde se deben encontrar los archivos de codificación en sus respectivos paquetes.
- Tenga en cuenta que su repositorio GitHub debe presentar una estructura base como por ejemplo:  
`videoGame/  
src/  
bin/  
doc/`
- Dentro de los directorios `src/` y `bin/` estarán presentes estos directorios (representando cada uno de sus paquetes):  
`ui/  
model/`
- El directorio `src` (source code) contiene sus clases `.java` dentro del directorio `ui/` y `model/`. Por otro lado, el directorio `bin` (binary files) contiene los archivos `.class` en el directorio `ui/` y `model/`. El directorio `doc` tendrá toda la documentación de análisis y diseño
- Su código debería compilar de acuerdo con lo explicado en la diapositiva 15 de esta presentación: <http://tinyurl.com/y3bd9bg2>

A continuación, encontrará un enunciado que narra de forma detallada la situación problemática que se espera usted solucione.

## Enunciado

Una empresa de videojuegos necesita que usted colabore con un videojuego, el juego consta de 10 niveles en los que el jugador recoge tesoros y pelea con enemigos. Han solicitado su ayuda para analizar los requerimientos y realizar el modelo de clases, responsabilidades y colaboraciones, por ahora SOLO para las funcionalidades del sistema relacionadas con el registro de jugadores, niveles, tesoros y enemigos.

**Registro de jugadores.** El sistema debe permitir registrar los jugadores, de ellos se maneja un nickname que lo identifica, un nombre, el puntaje inicial (el jugador inicia con 10), tiene un número de vidas (inicia con 5). El sistema debe permitir establecer el nivel en el que va el jugador teniendo en cuenta su puntaje y el puntaje requerido para pasar del nivel. Recuerde que el nickname es el identificador por lo que no podrá repetirse.

**Registro de nivel.** El sistema debe permitir registrar los diferentes niveles del juego. De un nivel se tiene un número que lo identifica, los puntos que se requiere para pasar al siguiente nivel. Para cada nivel, se deben registrar los tesoros y los enemigos que se debe vencer, y dependiendo de ellos se define el nivel de complejidad (alto, medio, bajo), si los puntos que otorgan los tesoros es mayor que los puntos de los enemigos es nivel bajo, si es igual es medio, y si son más los puntos de los enemigos que los puntos de los tesoros es alto.

**Registro de tesoros.** Los tesoros son el mecanismo que tienen los jugadores para alcanzar los puntos que se requieren para pasar un nivel. De los tesoros de un nivel, se tiene el nombre, una URL a la imagen que lo representa, el puntaje que otorga al jugador cuando es encontrado y la posición X y Y en pixeles en la que se encuentra (estas posiciones X y Y son generadas aleatoriamente, de acuerdo a la resolución de la pantalla utilizada (ver anexo)). En un nivel se puede encontrar el mismo tesoro en diferentes posiciones, es decir un diamante puede encontrarse en dos posiciones diferentes en un mismo nivel por lo que al momento de ingresar los tesoros al juego, se le preguntará al usuario, cuantos tesoros registrara para un mismo nivel.

**Registro de enemigos.** Los enemigos disminuyen el puntaje del jugador. De ellos se tiene un nombre (Identificador), un tipo, el puntaje que resta en caso de que le gane al jugador, el puntaje que se le suma si es derrotado y la posición X y Y en pixeles en la que se encuentra (estas posiciones X y Y son generadas aleatoriamente, de acuerdo a la resolución de la pantalla utilizada (ver anexo)). En un nivel no pueden repetirse los enemigos porque una vez derrotado, ya se sabría cómo derrotar a los demás del mismo nivel.

Los posibles tipos de enemigos en el juego son: ogros, abstractos, jefes y mágicos.

Para esta versión del juego, se tendrán un **máximo de 20 jugadores, 50 tesoros y 25 enemigos**. Es importante tener en cuenta que, al momento de iniciar el juego se crearán automáticamente los **10 niveles** y se debe preguntar al usuario cual es la resolución sobre la cual se ejecutará el juego. Ejemplo (El usuario escoge la resolución QHD) y esta información será utilizada al momento de generar randómicamente las posiciones en pantalla.

Para esta empresa es muy importante que el sistema tenga un muy buen desempeño, de manera que el despliegue de los tesoros y enemigos de un nivel, en la aplicación web no tarde más de 2 seg. Así mismo, la empresa quiere que el sistema funcione tanto como aplicación web como en una app para móviles.

El menú solicitado contendrá las siguientes opciones:

- Crear un Jugador

- Registrar enemigo a un nivel
- Registrar tesoro a un nivel
- Modificar el puntaje de un jugador.
- Incrementar nivel para un jugador, en caso que no pueda incrementar el nivel, debe informar al usuario que puntaje requiere para subir.
- Informar los tesoros y enemigos (separados por coma) de un nivel dado por el usuario
- Informar la cantidad encontrada de un tesoro en todos los niveles, es decir si el usuario quiere saber cuántos diamantes existen en todos los niveles.
- Informar la cantidad encontrada de un tipo de enemigo en todos los niveles, es decir si el usuario quiere saber cuántos ogros existen en todos los niveles.
- Informar el tesoro más repetido en todos los niveles.
- Informar el enemigo que otorga mayor puntaje y el nivel donde se ubica.
- Informar la cantidad de consonantes encontradas en los nombres de los enemigos del juego.
- Informar el top 5 de los jugadores de acuerdo al puntaje.

La entrega de la tarea integradora se encontrará dividida en dos partes:

Parte 1: Se entregará para 01 de octubre con los siguientes componentes:

- ❖ Análisis
- ❖ Diseño
- ❖ Implementación en java de:
  - Crear un Jugador
  - Registrar enemigo a un nivel
  - Registrar tesoro a un nivel: quiero 5 diamantes en el nivel 1
  - Modificar el puntaje de un jugador.
  - Incrementar nivel para un jugador, en caso que no pueda incrementar el nivel, debe informar al usuario que puntaje requiere para subir.
- ❖ Javadoc generado para las opciones anteriores

Parte 2: Se entregará completo para 15 de octubre

## Anexo: Resoluciones de pantalla

La resolución de una pantalla es el número total de píxeles que pueden ser mostrados en la pantalla de un monitor de ordenador, un televisor, un teléfono móvil o tableta y, en definitiva, cualquier dispositivo que tenga una pantalla. Todos los dispositivos tienen en sus pantallas una resolución concreta, y de ella va a depender la resolución de los vídeos que puedes llegar a alcanzar.

SIGLAS	NOMBRE	RESOLUCIÓN	DESCRIPCIÓN
<b>SD</b>	Standard Definition	640 x 480 píxeles	Uno de los primeros estándares de baja resolución. También se conoce como VGA, pero SD es un nombre más utilizado ahora.
<b>QHD</b>	Quarter of High Definition	960 x 540 píxeles	Significa literalmente un cuarto de alta definición, y ha sido utilizado en dispositivos de gamas bajas
<b>HD</b>	High Definition	1.280 x 720 píxeles	Es el primer estándar de alta resolución, y también se conoce como 720p. No llega a ser alta definición total, pero es como un paso previo muy utilizado. Es como la primera resolución considerada HD.
<b>FHD</b>	Full HD o Full High Definition	1.920 x 1.080 píxeles	También conocida como 1080p, es el estándar de alta definición total.
<b>QHD</b>	Quad High Definition	2.560 x 1.440 píxeles	También se conoce como 1440p o 2K, y es un estándar bastante utilizado en smartphones que quieren ir un poco más allá del Full HD.
<b>UHD</b>	Ultra High Definition	3.840 x 2.160 píxeles	Este es el famoso 4K, que también puedes encontrar como 2160p. Es un estándar de alta resolución muy superior al Full HD, y que poco a poco se empieza a convertir en lo normal en televisores de gamas medias y altas.
<b>UHD</b> <b>8K</b>	Ultra High Definition 8K	7.680 x 4.320 píxeles	El 8K o 4320p es el estándar de máxima resolución que se está empezando a ver cada vez más en televisores de gamas superiores. Aspira a ser el sucesor del 4K.