

TABLA DE ESPECIFICACIÓN DEL PROBLEMA DE INGENIERÍA DE SOFTWARE, identificando los siguientes elementos

CLIENTE	Empresa de videojuegos
USUARIO	Jugadores
REQUERIMIENTOS FUNCIONALES	<p>R1- Registro de jugadores</p> <p>R2- Registro de nivel</p> <p>R3- Registro de tesoros.</p> <p>R4- Registro de enemigos.</p> <p>R5- Modificar el puntaje de un jugador</p> <p>R6- Incrementar el nivel para un jugador</p> <p>R7- Informar los tesoros y enemigos de un nivel</p> <p>R8- Informar la cantidad encontrada de un tesoro en todos los niveles</p> <p>R9- Informar la cantidad encontrada de un tipo de enemigo en todos los niveles</p> <p>R10- Tesoro más repetido</p> <p>R11-Mayor enemigo</p> <p>R12- Consonantes encontradas</p> <p>R13- Top 5</p>
CONTEXTO DEL PROBLEMA	Una empresa de videojuegos quiere desarrollar su videojuego de 10 niveles en los que el jugador recoge tesoros y pelea con enemigos
REQUERIMIENTOS NO FUNCIONALES	<p>-El despliegue de los tesoros y enemigos de un nivel, en la aplicación web no tarde más de 2 seg</p> <p>-Que el sistema funcione tanto como aplicación web como en una app para móviles.</p>

Tabla de análisis de requerimientos funcionales (Nota: Una tabla por cada requerimiento funcional)

Nombre o identificador	R1- Registro de jugadores		
Resumen	Se debe registrar los jugadores, almacenando su nickname, nombre, el puntaje inicial (inicia con 10), el número de vidas (inicia con 5), el nivel en el que va el jugador teniendo en cuenta su puntaje y el puntaje requerido para pasar del nivel.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	nickname	String	debe ser unico
	name	String	
Actividades generales necesarias para obtener los resultados	Se deben pedir los datos de nombre y nickname para registrar el jugador y verificar que no esté repetido el nickname. También asignarle un puntaje y número de vidas inicializados con en 10 y 5, y un nivel que comienza en 1.		
Resultado o postcondición	imprimir un mensaje de éxito o error y guardar los datos del jugador		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	Mensaje de error Si el nickname ya está registrado

Nombre o identificador	R2- Registro de nivel		
Resumen	El sistema debe permitir registrar los diferentes niveles del juego: El número que lo identifica, los puntos que se requiere para pasar al siguiente nivel, los tesoros y los enemigos que se debe vencer, y el nivel de complejidad (alto, medio, bajo)		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	id	int	
	scoreToNextLevel	int	
	treasures	Treasure	
	enemies	Enemy	
Actividades generales necesarias para obtener los resultados	Se deben pedir los datos necesarios para registrar el nivel, crear los enemigos y tesoros y calcular el nivel de dificultad del nivel dependiendo de los puntos que dan estos.		
Resultado o postcondición	Almacenar los datos del nivel e imprimir un mensaje de éxito o error		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R3- Registro de tesoros.		
Resumen	Se le preguntará al usuario, cuantos tesoros registrara para el mismo nivel, y para cada uno se debe registrar 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 (generadas aleatoriamente, de acuerdo a la resolución de la pantalla utilizada) en píxeles en la que se encuentra.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	numberOfTreasures	int	
	name	String	
	imageURL	String	
	points	int	
Actividades generales necesarias para obtener los resultados	Se deben pedir los datos del registro de tesoros		
Resultado o postcondición	Almacenar los datos y mandar un mensaje de éxito o error		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R4- Registro de enemigos.		
Resumen	Registrar el nombre (Identificador), el tipo (ogros, abstractos, jefes y mágicos), 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 (generadas aleatoriamente, de acuerdo a la resolución de la pantalla utilizada en píxeles en la que se encuentra).		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	id	int	
	enemyType	String	
	pointsGiven	int	
	pointsSubtracted	int	
Actividades generales necesarias para obtener los resultados	Se deben pedir los datos del registro del enemigo		
Resultado o postcondición	Almacenar los datos y mandar un mensaje de éxito o error		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición

	msg	String	
--	-----	--------	--

Nombre o identificador	R5- Modificar el puntaje de un jugador		
Resumen	Se debe modificar el puntaje de un jugador		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	nickname	String	
	points	int	
Actividades generales necesarias para obtener los resultados	buscar el jugador por su nickname y Cambiar el puntaje del jugador ingresado, por los puntos ingresados		
Resultado o postcondición	imprimir mensaje de confirmación		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R6- Incrementar el nivel para un jugador		
Resumen	Se debe incrementar el nivel de un jugador ingresadp		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	nickname		
Actividades generales necesarias para obtener los resultados	Buscar el jugador ingresado y sumarle 1 a la variable nivel		
Resultado o postcondición	imprimir mensaje de confirmacion		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R7- Informar los tesoros y enemigos		
Resumen	Se deben informar los tesoros y enemigos (separados por coma) de un nivel dado por el usuario		

Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	lvl	int	debe ser un número entre el 1 y el 10
Actividades generales necesarias para obtener los resultados	Buscar todos los tesoros y enemigos registrados en ese nivel y concatenarlo en un string		
Resultado o postcondición	Todos los nombres de los tesoros y enemigos del nivel		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R8- Informar la cantidad de un tesoro en todos los niveles		
Resumen	Se debe 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.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	treasurename	String	
Actividades generales necesarias para obtener los resultados	Buscar si ese tesoro ya está registrado y contar cuantos hay en todos los niveles		
Resultado o postcondición	Un mensaje informando cuantos tesoros de ese tipo hay en el juego		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R9- Informar la cantidad de un enemigo en todos los niveles		
Resumen	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		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	enemyName	String	
Actividades generales necesarias para obtener los resultados	Buscar si ese enemigo ya esta registrado en el juego y contar cuantos hay en todo el juego		
Resultado o postcondición	un mensaje informando cuantos enemigos de este tipo hay		

Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R10- Tesoro más repetido		
Resumen	Informar el tesoro más repetido en todos los niveles.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
Actividades generales necesarias para obtener los resultados	Buscar cual es el tesoro que más veces esta en todo el juego		
Resultado o postcondición	un mensaje informando cual es el tesoro que más se repite en el juego		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R11- Mayor enemigo		
Resumen	Informar el enemigo que otorga mayor puntaje y el nivel donde se ubica.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
Actividades generales necesarias para obtener los resultados	Buscar el enemigo que mayor puntaje otorga al ser derrotado		
Resultado o postcondición	Un mensaje informando cuál es el mayor enemigo		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R12- Consonantes encontradas		
Resumen	Informar la cantidad de consonantes encontradas en los nombres de los enemigos del juego.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición

Actividades generales necesarias para obtener los resultados	Buscar todos los nombres de los enemigos del juego y contar cuántas consonantes tienen en total		
Resultado o postcondición	Un mensaje informando cuantas consonantes tienen		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Nombre o identificador	R13- Top 5		
Resumen	Informar el top 5 de los jugadores de acuerdo al puntaje.		
Entradas	Nombre entrada	Tipo de dato	Condición de selección o repetición
Actividades generales necesarias para obtener los resultados	Buscar todos los jugadores y encontrar los 5 con mayor puntaje		
Resultado o postcondición	Un mensaje informando los mejores 5 jugadores		
Salidas	Nombre entrada	Tipo de dato	Condición de selección o repetición
	msg	String	

Tabla de trazabilidad

Requerimiento Funcional	Nombre de la Clase	Nombre del método
ReqFunc001 Registro de jugadores	Clase Player	Player()
	Clase VideoGame	addPlayer()
	Clase VideoGame	searchPlayer()
	Clase VideoGameApp (paquete ui)	registerPlayer()
ReqFunc002 Registro de nivel	Clase VideoGameApp (paquete ui)	registerLevel()
	Clase VideoGame	addLevel()
	Clase VideoGame	searchLevel()

	Clase Level	Level()
ReqFunc003 Registro de tesoros.	Clase VideoGameApp (paquete ui)	registerTreasure()
	Clase VideoGame	addTreasure()
	Clase Level	addTreasure()
	Clase Treasure	Treasure()
ReqFunc004 Registro de enemigos.	Clase VideoGameApp (paquete ui)	registerEnemy()
	Clase VideoGame	addEnemy()
	Clase Level	searchEnemy()
	Clase Level	addEnemy()
	Clase Enemy	Enemy()
ReqFunc005 Modificar el puntaje de un jugador	Clase VideoGameApp (paquete ui)	modifyPoints()
	Clase VideoGame	changePoints
ReqFunc006 Incrementar el nivel para un jugador	Clase VideoGameApp (paquete ui)	levelUp()
	Clase VideoGame	levelUp()
ReqFunc007 Informar los tesoros y enemigos de un nivel	Clase VideoGameApp (paquete ui)	informTreasuresAndEnemies()
	Clase VideoGame	informTreasuresAndEnemies()
	Clase Level	getTreasureNames() getEnemyNames()
	Clase Enemy	getId()
	Clase Treasure	getName
ReqFunc008 Informar la cantidad de un tesoro en todos los niveles	Clase VideoGameApp (paquete ui)	numberOfTreasures()
	Clase VideoGame	numberOfTreasures()
	Clase Level	countNumberOfTreasures()
ReqFunc009	Clase VideoGameApp (paquete ui)	numberOfEnemies()
	Clase VideoGame	numberOfEnemies()

Informar la cantidad de un enemigo en todos los niveles	Clase Level	countNumberOfEnemies()
ReqFunc010 Tesoro más repetido	Clase VideoGameApp (paquete ui)	mostRepeatedTreasure()
	Clase VideoGame	mostRepeatedTreasure()
	Clase Level	getTreasuresLength() isRegistered() getTreasureName()
	Clase Treasure	getName()
ReqFunc011 Mayor enemigo	Clase VideoGameApp (paquete ui)	biggestEnemy()
	Clase VideoGame	biggestEnemy()
	Clase Level	getEnemiesLength() isEnemyRegistered() getEnemyName() getEnemyPointsGiven()
	Clase Enemy	getLength() getPointsGiven() getId()
ReqFunc012 Consonantes encontradas	Clase VideoGameApp (paquete ui)	numberOfConsonants()
	Clase VideoGame	consonants() isConsonant()
	Clase Level	getEnemiesLength() isEnemyRegistered()
	Clase Enemy	getId()
ReqFunc013 Top 5	Clase VideoGameApp (paquete ui)	top5()
	Clase VideoGame	top5()
	Clase Player	getNickname() getPoints()

Diagrama de Clase:

