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

CLIENTE	Game company
USUARIO	Administrator
REQUERIMIENTOS FUNCIONALES	R1: Create player R2:Register level R3:Register enemy R4:Register treasure R5: Modify score of player R6:Informing treasure and enemies from a level R7:Informing quantity of a treasure R8:Informing quantity of an enemy R9:Most repeated treasure R10:Enemy that give most quantity of points R11:Consonants in enemy's name R12:Top 5 of players by score R13:Register enemy in a level R14:Register treasure
CONTEXTO DEL PROBLEMA	Max of 20 players Max of 50 treasures Max of 25 enemies Automatically the game create 10 level at the beginning
REQUERIMIENTOS NO FUNCIONALES	The game has to operate in mobiles and web page The deployment of enemy and treasures has to be in less than 2 seconds

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

Name or identifier	R1: Register player
Resume	Take the required inputs and create an object of type player

Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	nickName	String	There isn’t any coincidence with the registered nicknames
	name	String	
	initialScore	double	
	lifes	int	
General activities necessary for obtain the result	1. Received the typed inputs 2. validate the information 3. create a object player		
Result or postcondition	Print message informing if the player was registered.Object of type player created		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R2:Register level		
Resume	Take the required inputs typed by the user and create an object of type level with respective attributes		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	Identifier	String	There isn’t any coincidence with the registered levels id
	levelUpPoints	double	
	Treasures	treasure	There is a empty space in treasures array
	Enemies	enemy	There is a empty space in enemies array and there isn’t other enemy of the same type
	Difficulty	String	

General activities necessary for obtain the result	1. Receive inputs for identifier and levelUpPoints 2. Register enemies 3. Register treasures 4. Define the difficulty 5. create an object of type level		
Result or postcondition	Print message informing if the level was registered.Object of type level created		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R3:Register enemy		
Resume	After user type the requested inputs,the system creates an object of class enemy		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	Id	int	There isn't any coincidence with the registered enemies id
	enemyType	String	Not be a empty space
	removedPoints	double	
	givenPoints	double	
	positionX	int	
	positionY	int	
General activities necessary for obtain the result	1. Receive inputs for enemy type,removedPoints and given points 2. Generate a random position 3. Create and object of type enemy		
Result or postcondition	Print message informing if the enemy was registered,object of type enemy created		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R4:Register treasure		
Resume	After user type the requested inputs,the system creates an object of class treasure		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	name	String	Not be a empty space
	url	String	Not be a empty space
	givenPoints	double	
	positionX	int	
	positionY	int	
General activities necessary for obtain the result	1. Receive inputs for name,url and given points 2. Generate a random position 3. Create and object of type treasure		
Result or postcondition	Print message informing if the enemy was registered,object of type treasure created		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	Strin	

Name or identifier	R5: Modify score of player		
Resume	According to the type of enemy that was defeated by the player,the system increases the score by the points that give that type of enemy.If the player was defeated will reduce the score of the player by the points that the type of enemy removes.		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	playerScore	double	
	pointsToOperate	double	
	playerlifes	int	
General activities necessary for obtain the result	1. Validate if the player was defeated or not 2. Get the value for operation 3. Modify the score of player 4. Print a message informing the modification		

Result or postcondition	A message on screen informing the point remove or increase in the player’s score,player score modificated		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R6:Informing treasure and enemies from a level		
Resume	the user chose a level from a list and the system print a message with the enemies and treasures contained in the chosen level		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	leveld	String	
General activities necessary for obtain the result	1. verify user’s selection 2. Get enemy names 3. Get treasure names 4. Print message with treasures and enemies names		
Result or postcondition	Message in screen with names of enemies and treasures present in a level		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R7:Informing quantity of a treasure		
Resume	The user type the name of the treasure and the system shows it’s quantity in a message		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**

	treasureName	String	the name type match with one of the name of treasures registered
General activities necessary for obtain the result	1. Verify that the name typed with one of the register treasure’s name 2. Search in each level the respective treasure and count the number of matches 3. Print a message informing the quantity of the same treasure in all the levels		
Result or postcondition	Message on screen informing the quantity of the same treasure in all the levels		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R8:Informing quantity of an enemy		
Resume	The user search a enemy by its id,the system search for match		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	EnemyId	String	The id typed match with one of the existent enemies id
General activities necessary for obtain the result	1. Verify that the id typed with one of the register t enemy’s id 2. Search in each level the chosen enemy and count the number of matches 3. Print a message informing the quantity of the same enemy in all the levels		
Result or postcondition	Message on screen informing the quantity of the same treasure in all the levels		

Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R9:Most repeated treasure		
Resume	The system count the quantity of treasure in all levels and return the one that is most repeated		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	treasureName	String	
General activities necessary for obtain the result	1. Get name of each treasure and count the times that is found in each level 2. Compare the count of all treasures 3. Print message with information of the treasure with the biggest count		
Result or postcondition	Message with information of the treasure with the biggest count		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R10:Enemy that give most quantity of points		
Resume	The system compare the quantity of points given by each type of enemy and return the one that gives the biggest value		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	EnemyId	String	

General activities necessary for obtain the result	1. The system get the type and the respective given points of each enemy in the game 2. It compare each one and chose the information of the enemy with the biggest value 3. Print message with the information		
Result or postcondition	Message on screen informing the level,type and point given by the enemy		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	msj	String	

Name or identifier	R11:Consonants in enemy's name		
Resume	Count the consonants in the name of each enemy in the game		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	EnemyName	String	
General activities necessary for obtain the result	1. Get the names of the enemies in the game 2. Count the consonants in each one 3. Print a message informing the result		
Result or postcondition	Message on screen with the total number of consonants in the game's enemies names		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	consonantsReport	String	

Name or identifier	R12:Top 5 of players by score
Resume	The system compare the score of each player then choose the one with the biggest score and print a message informing to the user

Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	playerScore	double	
General activities necessary for obtain the result	1. Get the score of each register player 2. Compare the scores and order all from the biggest to smallest 3. Return message with list of Nickname and score of the five players with the biggest score of all players		
Result or postcondition	Print on screen list of top 5 players		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	Top5players	String	

Name or identifier	R13:Register enemy in a level		
Resume	The system ask the user to choose a enemy and a level from the respective list of existent enemies and level then add the enemy to the level		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	enemySelection	int	Valid selection
	levelSelection	int	Valid selection
General activities necessary for obtain the result	1. Verify the selection of the enemy and level 2. Get the enemy 3. Get th level 4. Verify that there is a empty space in the level's array of enemies and there is not any enemy of the same type 5. Add the enemy to the level's array of enemies 6. Print message informing if the enemy was added		
Result or postcondition	Message that inform if the operation was possible,enemy added to the enemy's array of the respective level		

Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	enemyAdded	String	

Name or identifier	R14:Register treasure		
Resume	The system ask the user to choose a treasure and a level from the respective list of existent treasures and level then add the treasure to the level		
Inputs	**Input Name**	**Date type**	**Condition of selection or repetition**
	treasureSelection	int	Valid selection
	levelSelection	int	Valid selection
General activities necessary for obtain the result	1. Verify the selection of the treasure and level 2. Ask for the quantity of treasures of the same type the user want in the level 3. Get the treasure 4. Get th level 5. Verify that there are enough spaces in the level's array of treasures 6. Add the treasure to the level's array of treasures 7. Print message informing if the treasure was added		
Result or postcondition	Message that inform if the operation was possible,treasure added to the treasure's array of the respective level		
Outputs	**Output Name**	**Date type**	**Condition of selection or repetition**
	treasureAdded	String	

Requerimiento Funcional	Nombre de la Clase	Nombre del método
R1:Register Level	Class Level	Level(Identifier,levelUpPoints,treasures,enemies,difficulty)
	Class Game	addLevel(level)
R2:Register player	Class Player	Player(nickName,name,initialScore,lifes)
	Class Game	AddPlayer(player)
R3:Register Enemy	Class Enemy	Enemy(enemyType,removedPoints,givenPoints,positionX,positionY)
	Class Game	AddEnemy(Enemy)
R4:Register Treasure	Class Treasure	Treasure(name,url,givenPoints,positionX,positionY)
	Class Game	AddTreasure(Treasure)
R5:Modify Score of player	Class Game	getplayer
	Class player	getScore
	Class enemy	getGivenPoints,getRemovePoints
	Class Game	ModifyPlayerScore(playerScore,givenPoints,removePoints)
R6:Informing treasure and enemies from a level	Class Game	getLevelById
	Class Level	GetTreasures,GetEnemies
	Class Enemy	GetName
	Class Treasure	GetName
	Class Game	ShowMonsterAndTreasuresOfLevel(levelid)
R7:Informing quantity of a treasure	Class Game	getLevels
	Class Level	getTreasure
	Class Game	CountQuantityOfTypeTreasure(Treas

		ureName)
R8:Informing quantity of an enemy	Class Game	getLevels
	Class Level	getEnemies
	Class Game	CountQuantityOfTypeEnemy(levelId)
R9:Most repeated treasure	Class Game	SearchMostRepeatedTreasure(treasureName)
	Class Game	getLevels
	Class Game	CountQuantityOfTypeTreasureByLevel(TreasureName)
R10:Enemy that give most quantity of points	Class Level	getEnemies
	Class enemy	getGivenPoints
	Class Game	CompareGivenPoints(enemyGivenPoints)
R11:Consonants in enemy’s name	Class Game	getEnemies
	Class Game	countConsonants(enemyName)
R12:Top 5 of players by score	Class Game	getPlayers
	Class Player	getScore
	Class Game	ComparePlayersScore(playerScore)
R13:Register enemy in a level	Class Game	getEnemy
	Class Game	getLevelById
	Class Game	AddEnemyInLevel(Enemy)
R14:Register treasure in a level	Class Game	getTreasure
	Class Game	getLevelById
	Class Game	AddTreasureInLevel(treasure)