**ANÁLISIS**

**Descripción del contexto:**La empresa Epic Games está preocupada por la gran cantidad de recomendaciones que han dejado muchos de sus jugadores en el último par de meses, esta preocupación radica en algunos inconvenientes que los usuarios han sentido con el gameplay de sus partidas.

**Identificación y definición concreta del problema:**

* Los usuarios requieren encontrar partidas donde se enfrenten con jugadores que tengan un nivel de juego similar.
* La empresa debe diseñar un ranking en donde se categorice a los jugadores, según su destreza para el juego.
* Los usuarios requieren poder realizar partidas en modo “plataforma”, en la cual se limitan los jugadores que se unirán a la partida, de acuerdo a la plataforma que se esté utilizando.
* Los usuarios piden como una variante de juego, poder celebrar ocasiones especiales.
* La empresa ha aprobado la versión para el día de San Valentín, un modo de juego en donde sólo se puede utilizar la última arma que se levantó.

1. **Recopilación de Información**

**Fornite:**Es un videojuego desarrollado por las empresas Epic Games y People Can Fly que, a pesar de ser anunciado en 2011, no fue lanzado hasta el 25 de julio de 2017. Se trata de un juego de zombis en el que supervivientes controlados por humanos cooperarán online para mantener sus fortalezas a salvo de los ataques en oleadas de los muertos vivientes, que están intentando conquistar la tierra.

Fuente: <https://docs.google.com/document/d/1bqwF-1Vk-vd4-JlrMB7eY_AThAMdl0taRiEHogiHaAA/edit>

**San Valentín:**Es una festividad de origen cristiana que se celebra anualmente el 14 de febrero como conmemoración a las buenas obras realizada por san Valentín de Roma que están relacionadas con el concepto universal del amor y la afectividad.

Fuente: <https://es.wikipedia.org/wiki/D%C3%ADa_de_San_Valent%C3%ADn>

**Ranking:**Es una relación entre un conjunto de elementos tales que, para uno o varios criterios, el primero de ellos presenta un valor superior al segundo, este a su vez mayor que el tercero y así sucesivamente, permitiéndose que dos o más elementos diferentes puedan tener la misma posición. El orden se refleja asignando a cada elemento un ordinal, generalmente números enteros positivos.

Fuente: <https://es.wikipedia.org/wiki/Ranking>

**Desviación estándar:**Es una medida de dispersión para variables de razón (variables cuantitativas o cantidades racionales) y de intervalo. Se define como la raíz cuadrada de la varianza de la variable.

Su función es:

$ s =\sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i-\bar{x})^2$

N es el tamaño y x-bar la media aritmética de la muestra

Fuente del texto:

<https://es.wikipedia.org/wiki/Desviaci%C3%B3n_t%C3%ADpica>

Fuente de la imagen:

<http://www.alcula.com/es/calculadoras/estadistica/desviacion-estandar/>

**Ping:**Como programa, ping es una utilidad diagnóstica en redes de computadoras que comprueba el estado de la comunicación del host local con uno o varios equipos remotos de una red IP por medio del envío de paquetes ICMP de solicitud (ICMP Echo Request) y de respuesta (ICMP Echo Reply). Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada.

Fuente: <https://es.wikipedia.org/wiki/Ping>

**Plataforma:** Se refiere al sistema operativo o a sistemas complejos que a su vez sirven para crear programas, como las plataformas de desarrollo.

Fuente: <https://es.wikipedia.org/wiki/Plataforma>

1. **Búsqueda de soluciones creativas**

Alternativa 1: Listas enlazadas.

La lista enlazada es un TDA que nos permite almacenar datos de una forma organizada, al igual que los vectores pero, a diferencia de estos, esta estructura es dinámica, por lo que no tenemos que saber "a priori" los elementos que puede contener.

En una lista enlazada, cada elemento apunta al siguiente excepto el último que no tiene sucesor y el valor del enlace es null. Por ello los elementos son registros que contienen el dato a almacenar y un enlace al siguiente elemento. Los elementos de una lista, suelen recibir también el nombre de nodos de la lista.

Alternativa 2: ArrayList.

La clase ArrayList en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño. Además permite modificar, insertar y eliminar elementos en cualquier lugar del arreglo.

Alternativa 3: Pila.

Una pila es una lista ordenada o [estructura de datos](https://es.wikipedia.org/wiki/Estructura_de_datos" \o "Estructura de datos) que permite almacenar y recuperar datos, el modo de acceso a sus elementos es de modo que el primer elemento en entrar es el primer elemento en salir ([LIFO](https://es.wikipedia.org/wiki/LIFO" \o "LIFO)). Esta estructura se aplica en multitud de supuestos en el área de [informática](https://es.wikipedia.org/wiki/Inform%C3%A1tica" \o "Informática) debido a su simplicidad y capacidad de dar respuesta a numerosos procesos. En ésta estructura de datos se puede apilar o des apilar elementos, es decir modificar únicamente el extremo superior de la estructura.

Alternativa 4: Cola.

Una cola (también llamada fila) es una [estructura de datos](https://es.wikipedia.org/wiki/Estructura_de_datos" \o "Estructura de datos), caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pop por el otro. También se le llama estructura [FIFO](https://es.wikipedia.org/wiki/FIFO" \o "FIFO) (del inglés First In First Out), debido a que el primer elemento en entrar será también el primero en salir.

Las colas se utilizan en sistemas [informáticos](https://es.wikipedia.org/wiki/Inform%C3%A1tica" \o "Informática), [transportes](https://es.wikipedia.org/wiki/Transporte" \o "Transporte) y operaciones de [investigación](https://es.wikipedia.org/wiki/Investigaci%C3%B3n" \o "Investigación) (entre otros), donde los objetos, personas o eventos son tomados como datos que se almacenan y se guardan mediante colas para su posterior procesamiento. Este tipo de estructura de datos abstracta se implementa en [lenguajes orientados a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos" \o "Programación orientada a objetos) mediante clases, en forma de listas enlazadas.

Alternativa 5: Hash table.

es una [estructura de datos](https://es.wikipedia.org/wiki/Estructura_de_datos" \o "Estructura de datos) que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda: permite el acceso a los elementos almacenados a partir de una clave generada. Funciona transformando la clave con una [función hash](https://es.wikipedia.org/wiki/Funci%C3%B3n_hash" \o "Función hash) en un [hash](https://es.wikipedia.org/wiki/Hash), un número que identifica la posición donde la tabla hash localiza el valor deseado.

Las tablas hash se suelen implementar sobre [vectores](https://es.wikipedia.org/wiki/Vector_(programaci%C3%B3n)) de una dimensión, aunque se pueden hacer implementaciones multi-dimensionales basadas en varias claves. Como en el caso de los arrays, las tablas hash proveen tiempo constante de búsqueda promedio [O(1)](https://es.wikipedia.org/wiki/Cota_superior_asint%C3%B3tica" \o "Cota superior asintótica),[1](https://es.wikipedia.org/wiki/Tabla_hash" \l "cite_note-1)​ sin importar el número de elementos en la tabla. Sin embargo, en casos particularmente malos el tiempo de búsqueda puede llegar a O(n), es decir, en función del número de elementos.

Las anteriores estructuras de datos serán utilizadas en este caso para:

* Almacenar a los jugadores con sus respectivos puntajes para encontrar a los contrincantes con niveles de juego parecidos.
* Mostrar el ranking de los jugadores y asi, poder acceder a ellos de forma rápida y más efectiva.
* Manejar los jugadores con el tipo de plataforma de juego para que los usuarios conozcan a sus contrincantes o compañeros de juego.

1. **Transición de las ideas a los diseños preliminares**

Alternativa 1:

* Tamaño dinámico que implica una mejor optimización de la memoria.
* Los elementos se pueden insertar en una lista indefinidamente.
* las listas enlazadas sólo permiten acceso secuencial a los elementos.
* Las listas enlazadas simples, solo pueden ser recorridas en una dirección.
* Las listas simples requieren la dirección del nodo anterior para insertar o suprimir correctamente.

Alternativa 2:

* Un ArrayList tiene un tamaño dinámico.
* el ArrayList es unidimensional.
* un ArrayList puede cambiar de tamaño según se necesite en tiempo de ejecución.
* admite elementos duplicados y nulos, así como acceso por índice y otros métodos para trabajar con sus elementos.

Alternativa 3:

* En cada momento sólo se tiene acceso a la parte superior de la pila.
* Introduce un elemento en el extremo de la pila.
* Elimina un elemento de la pila (superior extremo).
* Indica si la pila está o no vacía.
* Permite realizar apilamiento, desapilamiento y verificación de vacío.

Alternativa 4:

* Los elementos salen en el mismo orden en que llegan.
* Almacena al final de la cola un elemento.
* Saca de la cola al elemento que se encuentra al frente.
* Se crea la cola vacía.

Alternativa 5:

* El acceso a los datos suele ser muy rápido.
* Debe tener una razón de ocupación no muy elevada.
* Función bien diseñada para evitar que hayan muchas colisiones.
* Necesidad de ampliar el espacio cuando los datos crecen mucho.
* Se suelen emplear listas para recorrer la totalidad de los elementos.

1. **Evaluación y selección de la mejor solución**

* Analizando cada uno de los casos, llegamos a la conclusión de que las alternativas 3,4 y 5.

1. **Preparación de informes y especificaciones**
2. **Implementación del diseño**

**ETAPA 2: PSP 0**

1. **Medición del tiempo**
2. **Registro de defectos**
3. **Registro de comentarios**

**ETAPA 3: DISEÑO**

1. **Definición de tipos abstractos de datos (TAD)**

|  |  |
| --- | --- |
| **TAD** | **Ranking** |
| ***Definición abstracta:***  El ranking es una relación entre un conjunto de elementos tales que, para uno o varios criterios, el primero de ellos presenta un valor superior al segundo, este a su vez mayor que el tercero y así sucesivamente, permitiéndose que dos o más elementos diferentes puedan tener la misma posición. | |
| ***Invariantes*** | |
| ***Operaciones:*** | |

|  |  |
| --- | --- |
| **TAD** | **Ping** |
| ***Definición abstracta:***  Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada. | |
| ***Invariantes*** | |
| ***Operaciones:*** | |

|  |  |
| --- | --- |
| **TAD** | **Plataforma** |
| ***Definición abstracta:***  Se refiere al sistema operativo o a sistemas complejos que a su vez sirven para crear programas, como las plataformas de desarrollo. | |
| ***Invariantes*** | |
| ***Operaciones:*** | |

|  |  |
| --- | --- |
| **TAD** | **Armas** |
| ***Definición abstracta:***  Objeto o instrumento que sirve para atacar a una persona o animal o para defenderse de ellos. | |
| ***Invariantes:*** | |
| ***Operaciones:*** | |

1. **Diseño de diagramas de clase**
2. **Diseño de casos de pruebas unitarias**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DISEÑO DE CASOS PARA PRUEBAS UNITARIAS | | | | | |
| Entrada | **Salida** | **Descripción** | **Nombre del método** | **Clase** | **Escenario** |
|  | True | El método retorna true si los jugadores han sido rankeados, retorna false si no lo son |  |  | SceneOne() |
|  | False |
|  | True | El método retorna true si la desviación estándar del ping de los jugadores de una partida es el mínimo posible, de lo contrario retorna false. |  |  | SceneTwo() |
|  | False |
|  | True |  |  |  | SceneThree() |
|  | False |
|  | True |  |  |  | SceneFour() |
|  | False |
|  | True |  |  |  | SceneFive() |
|  | False |
|  | True |  |  |  | SceneSix() |
|  | False |
|  | True |  |  |  | SceneSeven() |
|  | False |
|  | True |  |  |  | SceneEigh() |
|  | False |
|  | True |  |  |  | SceneNine() |
|  | False |

**ETAPA 4: IMPLEMENTACIÓN (***Eclipse****)***

1. **Estructuras de datos**
2. **Programa completo**
3. **Implementación de las pruebas unitarias**
4. **Inglés en codificación**
5. **Uso de un sistema de control de versiones**

La empresa Epic Games está preocupada por la gran cantidad de recomendaciones que han dejado muchos de sus jugadores en el último par de meses, esta preocupación radica en algunos inconvenientes que los usuarios han sentido con el gameplay de sus partidas.

La primera preocupación tiene que ver con la dificultad que tienen los usuarios de encontrar partidas donde se enfrenten con jugadores que tengan un nivel de juego parecido. Por eso le están pidiendo a la empresa que diseñe un ranking en donde pueda categorizar a los jugadores según su destreza para el juego (skill). La empresa había considerado esto alguna vez, pero al hacer el ranking temían que el tiempo promedio de espera para encontrar una partida aumentará mucho. Otra cosa que temían es que la diferencia de latencia entre los jugadores de la misma partida variara mucho (lo cual crearía desventaja para los jugadores, así tuvieran el mismo nivel de destreza para el juego), ya que actualmente uno de los principales factores para el emparejamiento de jugadores y partidas son el ping, y la geolocalización del jugador.

Otra de las constantes recomendaciones que le hacen a la empresa es que se puedan realizar partidas en modo “Plataforma”. Lo cual permite limitar los jugadores que se unirán a la partida de acuerdo a la plataforma que estén utilizando. Esto promoverá a las empresas como Microsoft, Nintendo y Playstation que promuevan las partidas exclusivas para su plataforma.

La última de las recomendaciones que más les llega, es la creación de variantes del juego para celebrar ocasiones especiales (como lo son navidad, halloween, pascua, etc). En la empresa han tenido muchas ideas acerca del tema que van desde convertirlo en un *mod* especial donde se batalle como en un MOBA o desarrollar un storyline especial como un MMORPG; como puedes ver estas ideas son muy descabelladas y no han llamado la atención de las personas encargadas. Actualmente la versión que aprobaron para el día de San Valentín es un modo de juego donde solamente puedes utilizar la última arma que levantaste. En este modo de juego no tienes límite de armas para llevar y solamente podrás cambiar de arma encontrando una nueva o agotando las municiones de tu arma actual, lo cual hará desaparecer el arma actual y te equipará con la última arma que levantaste antes que esa. Es importante que tengas en cuenta que todos los jugadores de Fornite por defecto salen con un Hacha, la cual es un arma de corto alcance con la cual pueden conseguir recursos, como las hachas no tienen municiones no se pueden desaparecer.

El equipo de Epic Games se ha enterado de tus grandes habilidades para resolver problemas y tu gran conocimiento de estructuras de datos por lo cual ha decidido contratarte para que presentes una propuesta de cómo implementar estas nuevas características en su producto, éstas son:

1. Proponer un sistema de clasificación que permita a los usuarios ser rankeados, que no afecte el tiempo promedio de búsqueda de una nueva partida entre n usuarios con ranking similar.
2. Se asegure que todas las personas en una partida tenga un latencia muy cercana. Es decir, que la desviación estándar del ping de los jugadores de una partida sea el mínimo posible (para asegurar la igualdad de condiciones en la partida).
3. Proponer una implementación para el modo plataforma.
4. Proponer una implementación del modo de juego que será lanzado para San Valentín.

Para cada una de los anteriores ítems estamos esperando una muy buena descripción del procedimiento , con una breve explicación de lo que está haciendo y el  por qué funciona.Usted debe utilizar el método de la ingeniería para resolver este problema y dejar evidencia en su informe de los resultados de cada fase. Recuerde revisar el [Resumen del Método de la Ingeniería](https://goo.gl/UdqecR) y el [ejemplo del Método de la Ingeniería aplicado a un problema](https://goo.gl/Lpv4Y3).

**Entregables:**

1. Informe PSP0. Cada estudiante debe entregar el informe de su desarrollo.
2. Entrega de informe del método de la ingeniería.
3. Diseño del TAD para cada estructura de datos requerida.
4. Diseño del diagrama de clases desacoplado y utilizando generics.
5. Diseño de los casos de prueba. Adicionalmente debe explicar cómo se resuelven dos casos, paso a paso (con dibujos, si es necesario), por cada estructura de datos diseñada e implementada.
6. Diseño del diagrama de clases de pruebas unitarias automáticas.
7. Todos los archivos deben estar almacenados en GitHub y debe evidenciarse su uso desde el inicio del proyecto.
8. Implementación de las pruebas unitarias automáticas.