

# Stats Tracker

**Baruj Vladimir Ramirez Escalante, Camilo Andres Fierro Fierro, Fabian Andres Roza Rodriguez**

**No. de Equipo Trabajo: 3**

## I. INTRODUCCIÓN

Este documento contiene la información actual del estado del proyecto “Stats Tracker”, así como las ideas referentes al diseño final esperado por el equipo. También contiene datos del desarrollo y progreso que se ha tenido con la idea inicial, y como puede verse afectada por factores humanos y/o errores de implementación.

## II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Lo que se quiere lograr principalmente es responder a la pregunta: ¿es posible crear una aplicación que registre estadísticas varias, entre ellas, tiempo de uso, puntuaciones, comparación con otros usuarios, desempeño en general, de múltiples videojuegos, que estén dispuestas de forma fácil de entender para el usuario?

Existen algunas aplicaciones que agrupan determinados datos sobre el desempeño de un jugador, como es el caso de Tracker Network para algunos aspectos de la puntuación y las victorias/derrotas de algunos videojuegos, Discord, que registra las horas de la sesión en curso del jugador. Pero ninguna de estas agrupa ambos aspectos. Este proyecto buscará agrupar dichas funcionalidades y procurar que funcione de una forma amplia para múltiples videojuegos

## III. USUARIOS DEL PRODUCTO DE SOFTWARE

Este proyecto tiene como público objetivo a las personas que juegan videojuegos de manera casual pero les interesa saber cual es su espectro de mejora. De niños a adultos, se espera que su funcionamiento sea claro y fácil para todas las edades y que permita conocer con mayor detalle las estadísticas de juego del usuario y también que pueda interactuar con las mismas. Principalmente estaría enfocado hacia adultos y jóvenes pero sin excluir a niños o adultos mayores, ya que se espera que sea un software accesible para cualquiera.

## IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

Para que este proyecto funcione de la manera correcta y como se espera, es necesario cumplir con ciertos requisitos que afectan al usuario.

El principal requisito será que el usuario debe registrarse para poder utilizar el software, ya que este tendrá información variada de múltiples videojuegos y por lo tanto es necesario el registro de un usuario para enlazar esa información única.

Para ingresar datos, el usuario deberá escoger el videojuego que desea medir, y si existe la plantilla, rellenar datos conocidos para hacer el cálculo.

Se espera poder enlazar cuentas de otros juegos con el software para ahorrar tiempo al usuario cuando sea posible.

- *Creación de usuarios*

- *Descripción:*

- Permite a los usuarios crear una cuenta en la cual tener sus estadísticas guardadas para editarlas o crear nuevas

- *Acciones iniciadoras y comportamiento esperado:*

- El usuario deberá ingresar al software, en la pantalla de inicio aparecerán los cuadros de texto para ingresar su id y contraseña, o debajo una opción para registrarse en caso de no haberlo hecho aún. Una vez creada o autenticada pasará a su página de perfil.

*Requerimientos funcionales:*

Para que esta funcionalidad se lleve a cabo de manera adecuada, el usuario deberá ingresar su id y contraseña de manera correcta en la página inicial, y en caso de equivocarse deberá reintentarlo. Si no recuerda su contraseña, podrá pedir que sea cambiada y se contactará por correo electrónico para esto, luego el usuario creará una nueva contraseña que se asociará a su id. En caso de no existir la cuenta, se informará de esto y se preguntará si desea crear una nueva cuenta.

- *Obtención de datos del juego*

- *Descripción:*

- Obtener y calcular según sea el caso, los datos del jugador respecto a un solo juego, para mostrar o actualizar las estadísticas del mismo.

- *Acciones iniciadoras y comportamiento esperado:*

- El usuario deberá ingresar el nombre que tiene en el juego seleccionado o ingresar los datos requeridos cuando es un juego que no tiene información recopilada del usuario. Luego de esto se generarán las estadísticas actualizadas del jugador.

*Requerimientos funcionales:*

Cuando la cuenta del usuario esté autenticada, pasará a otra ventana con una lista de los juegos permitidos, si el jugador no encuentra el juego deseado, podrá escribir una petición para que se agregue. Si encuentra el juego que buscaba, podrá seleccionarlo e ir a otra pantalla, donde podrá ingresar o editar

información de su progreso o su nombre en el juego(según sea el caso) y con esta información se imprimirán las estadísticas disponibles actualizadas.

Si alguno de los datos no coincide o no es permitido para el formato, no permitirá el ingreso de los datos e informará de que se deben ingresar de manera correcta o corregir los campos que sean necesarios.

**IMPORTANTE:** En cada una de las entregas para reportar el avance en el desarrollo del proyecto, se especificarán las funcionalidades mínimas y las estructuras de datos mínimas requeridas que se deben implementar. También, se debe presentar un análisis (comparativo y asintótico) de la eficiencia de las estructuras de datos usadas.

## V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

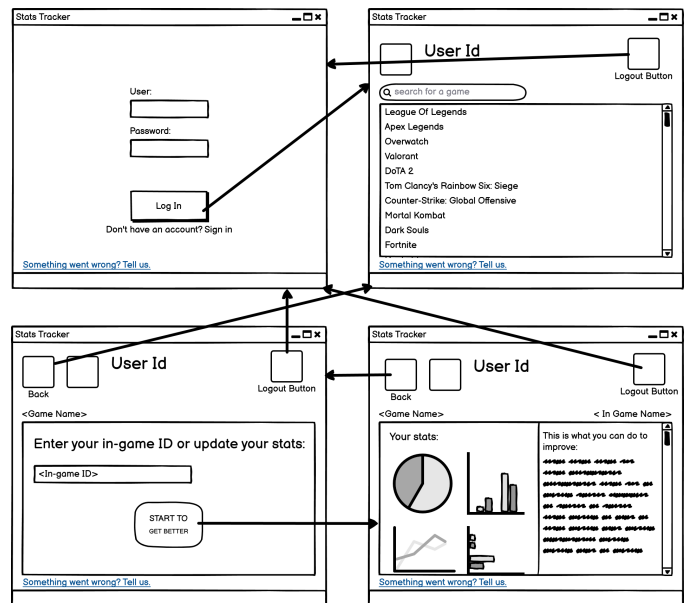
El concepto consiste en que, en primer lugar, se encuentra una pantalla, que le pide datos para identificarlo, este deberá suministrar un usuario y una contraseña, y en la parte inferior se encuentra un enlace para registrarse en caso de no tener una cuenta.

Si existe la cuenta, la siguiente ventana que mostrará, le mostrará una lista de juegos al usuario para elegir, de los cuales se tenga soporte, y a partir de ahí el usuario puede buscar un juego en específico.

Una vez elige el juego, dando doble click, y desde ahí existen dos opciones, la primera, es proporcionar el nombre que se tenga en el juego, o se actualizarán los datos requeridos por la aplicación.

Cuando el usuario da click el botón “Start to get better”, le mandará una pantalla al usuario, que muestre una comparación gráfica de sus estadísticas, y cómo han cambiado en el tiempo, y al lado, consejos de cómo se podría mejorar en el juego en cuestión.

Para la entrega se tiene que se debe ingresar la llave que se genera diariamente para la API, luego de esto se ingresa el nombre de usuario dentro del juego y con esto se obtiene la información de las últimas partidas jugadas.



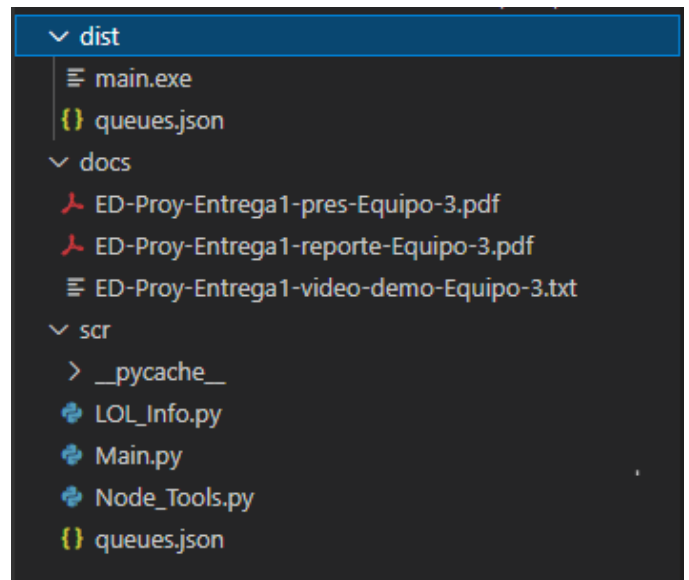
## VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El software se está desarrollando actualmente por medio de Visual Studio Code utilizando el lenguaje Python y las librerías que este ofrece.

Este proyecto está planeado para que se ejecute y funcione para Windows 10, se espera que el hardware del usuario no sea un impedimento para el funcionamiento de la herramienta.

## VII. PROTOTIPO DE SOFTWARE INICIAL

El prototipo presentado esta estructurado de la siguiente forma:



se puede encontrar el repositorio en github:  
<https://github.com/RzRz-dev/Stat-tracker>

### VIII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

Para esta entrega se usaron los arrays para la obtención de la información del jugador desde la API hasta nuestro software, donde se guardó y posteriormente usó.

Se hizo uso de una lista doblemente encadenada para poder realizar la impresión de los datos en pantalla y cambiar entre la información de las partidas anteriores o siguientes de manera fácil y rápida.

Para imprimir la lista de las últimas partidas jugadas, se usa una cola para sacar una a una las partidas e ir eliminándolas en base a su orden de llegada.

Con esto, se logró ingresar, borrar y recorrer las listas para obtener la información de una sola partida o de todas las partidas disponibles.

### IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

En la versión actual del desarrollo, la API que se está usando no nos permite hacer más de 200 solicitudes por cada 2 minutos, por lo que el flujo de datos que se puede manejar queda muy reducido. En un estado más avanzado de la app, se podrá registrar oficialmente para que pueda solicitar mas datos.

Número de datos	Tiempo
mil datos	1 minuto (60.9494 segundos)
10 mil datos	aprox 10 minutos (9.86)
100 mil datos	aprox hora y media (98.575 minutos)
1 millón de datos	aprox 16 horas (16.4291 horas)
10 millones de datos	aprox 7 días (6.8454 días)
100 millones de datos	aprox 69 días

Después de 10 mil datos el programa no se detiene pero tardaba demasiado debido al problema mencionado con la API, por lo que hicimos los aproximados en base al funcionamiento que sí fue posible medir y luego se usó el promedio de tiempo para cada caso de ejecución (11.82899903 s/caso)

Para nuestro software actual, determinamos que el análisis asintótico debe hacerse en base a  $O(n)$ , ya que nuestro proceso más “pesado”, recorre una lista de 20 elementos, para obtener los datos requeridos las “n” veces que sea necesario. Por lo tanto, “ $20n$  es  $O(n)$ ”.



### X. ROLES Y ACTIVIDADES

Se definieron los siguientes roles durante esta entrega:

INTEGRANTE	ROL(ES)	ACTIVIDADES REALIZADAS (Listado)
<b>Baruj Vladimir Ramirez Escalante</b>	Coordinador	Contactar a los integrantes
		Organizar las reuniones del grupo
	Líder	Llevar el plan del proyecto
		Aportar la estrategia de trabajo
<b>Camilo Andres Fierro Fierro</b>	Observador	Aportar ideas para el desarrollo de los objetivos
		Trabajar en conjunto para acelerar el proceso
	Animador	Mantener un ambiente bueno para el trabajo
<b>Fabian Andres Roza Rodriguez</b>	Investigador	Consultar la comunicación con las apis de riot
		Consultar la solución de algunos bugs y alternativas para el desarrollo del proyecto

	Experto	Manipulación de la api de riot games
		Diseño de la interfaz de usuario preliminar

## XI. DIFICULTADES Y LECCIONES APRENDIDAS

El principal problema fue la motivación de trabajo, ya que no se tiene claro que hacer sin tener bases en que sustentar el desarrollo. No tener claros los requerimientos del trabajo también afecta ya que se deben hacer cambios a la ruta planificada con anterioridad.

Es necesario trabajar primero en como debemos seguir con el desarrollo una vez se tienen todos los objetivos claros, en vez de trabajar en un punto para luego tener que continuar en base a lo que ya se tiene.

El API actualmente manejado no permite un gran recibimiento de datos.

## XII. REFERENCIAS BIBLIOGRÁFICAS

(2019, September 20) Análisis asintótico [Online]. Available: [https://es.wikipedia.org/wiki/Análisis\\_asintótico](https://es.wikipedia.org/wiki/Análisis_asintótico)

H. Barney (2020, March 20)How To Use Riot API With

Python[Online]. Available:

<https://towardsdatascience.com/how-to-use-riot-api-with-python-b93be82dbbd6>

LEAGUE OF LEGENDS [Online]. Available:

<https://developer.riotgames.com/docs/lol>

J.Quick (2019,December 12) How To use Live Share with Visual

Studio Code[Online]. Available:

<https://www.digitalocean.com/community/tutorials/how-to-use-live-share-with-visual-studio-code#:~:text=Live%20Share%20is%20an%20extension.a%20sever%20and%20debugging%20session.>