

RETO 3: Speedrunning de Videojuegos

Objetivo

Poner en práctica los conceptos aprendidos en clase acerca de las estructuras de datos módulo No. 3 sobre ADTs de mapas ordenados como los árboles binarios de búsqueda (BST) y árboles balanceados rojo-negro (RBT) y realizar búsquedas eficientes de información.

Específicamente se pretende:

1. **Practicar** los conceptos sobre estructuras de tablas ordenadas (árboles binario de búsqueda y balanceados).
2. **Integrar** el uso de las estructuras lineales (tablas de hash, Lista, pilas y colas) a los árboles binarios.
3. **Practicar** los algoritmos de ordenamientos y búsquedas lineales.
4. **Utilizar** adecuadamente el patrón MVC (Modelo-Vista-Controlador).
5. **Aprender** a cargar y procesar en memoria datos en formato CSV.
6. **Utilizar** adecuadamente el administrador de versiones GIT y GitHub.
7. **Aprender** a trabajar en equipo

Fecha Límite de Entrega

Máxima fecha de entrega para el 8 de noviembre antes de la media noche (11:59 p.m. hora BrightSpace).

Actualizaciones

A continuación, encontrará un listado de todas las actualizaciones realizadas sobre el enunciado para claridad de los requerimientos y objetivos del Reto.

1. Se **editaron** los requerimientos del 1 al 7 para facilitar su entendimiento.
2. Se **modificaron** los ejemplos funcionales de los requerimientos del 1 al 7 para que coincidieran con la información disponible en el subconjunto de datos -small.
3. Se **agregaron** los ejemplos funcionales de los requerimientos del 1 al 7 utilizando el subconjunto de datos -small.
4. Se **modificaron** las recomendaciones para implementar adecuadamente los requerimientos del 1 al 7.
5. Se **modificaron** las variables de entrada y la respuesta esperada de los requerimientos 6 y 7 para ajustar su implementación.
6. Se **modifico** el algoritmo para calcular la rentabilidad del requerimiento 7 para facilitar su entendimiento e implementación.

Contexto

El *speedrunning* existe como una subcultura¹ dentro del mundo de los videojuegos. Esto sucede cuando un individuo intenta vencer parte o la totalidad de un videojuego lo más rápido posible. Esto puede incluir completar niveles individuales, objetivos específicos o completar el juego con limitaciones únicas según lo decida la comunidad o el jugador². Algunos jugadores y creadores de contenido de videojuegos dedican su tiempo en mejorar estos registros tan solo por milésimas de segundo, con el objetivo de ser acreedores del record mundial en diferentes videojuegos y categorías. Los jugadores que habitualmente realizan *speedruns* suelen tener tiempos de juego completamente inalcanzables tanto para jugadores casuales como para otros jugadores competitivos de videojuegos.

Carga de Datos

Los datos para este reto fueron tomados del repositorio “**Game Speedrun Records**”³, que contiene los mejores registros en diferentes videojuegos actualizados hasta el 5 de abril de 2022.

El archivo `game_data.csv` contiene la información necesaria relacionada con los videojuegos en general. En la Tabla 1 puede verse un resumen de las columnas incluidas en este archivo.

Nombre de la Columna	Descripción
Game_Id	Identificador del videojuego en el conjunto de datos.
Abbreviation	Abreviación del nombre del videojuego.
Name	Nombre completo del videojuego.
Genres	Género del videojuego.
Platforms	Plataforma o plataformas en las que se encuentra disponible el videojuego.
Total_Runs	Cantidad total de intentos de romper algún récord en el juego.
Release_Date	Fecha de publicación original del videojuego en formato AAAA-MM-DD.

Tabla 1. Descripción de los campos para los videojuegos en el archivo CSV `game_data`.

Por su parte, el archivo `category_data.csv` contiene la información de los registros históricos de los registros de speedrun, separados en varias categorías para un cada juego. Los detalles de las propiedades de cada registro se pueden ver en Tabla 2. Donde se puede notar que ambos archivos comparten la columna `Game_Id`.

¹ Speedrunning's starting line: An intro guide to gaming's seemingly intimidating subculture, by Jhaan Elker julio 13, 2021, URL: <https://www.washingtonpost.com/video-games/2021/07/13/how-get-into-speedrunning/>

² What Is Speedrunning, URL: <https://www.speedrun.com/knowledgebase/what-is-speedrunning>

³ Game Speedrun Records, URL: <https://www.kaggle.com/datasets/matheusturatti/game-speedrun-records>

Nombre de la Columna	Descripción
Game_Id	Identificador unico del videojuego en el archivo.
Category_Id	Identificador de la categoría del récord
Subcategory_Id	identificador de la subcategoría del record
Misc	Indica si la categoría es considerada como <i>miscelanea</i> o no
Category	Nombre de la categoría
Num_Runs	Cantidad total de intentos que intentan romper el récord.
Subcategory	Nombre de la subcategoría
Time_0	Tiempo en segundos para el mejor intento registrado.
Country_0	País o países de origen del jugador o jugadores con el mejor tiempo registrado.
Players_0	Nombre del jugador o jugadores con el mejor tiempo registrado.
Record_Date_0	Fecha y hora de realización del mejor tiempo en formato YYYY-MM-DDThh:mm:ssZ.
Time_1	Tiempo en segundos para el segundo mejor tiempo registrado.
Country_1	País o países de origen del jugador o jugadores con el segundo mejor tiempo registrado.
Players_1	Nombre del jugador o jugadores con el segundo mejor tiempo registrado.
Record_Date_1	Fecha y hora de realización del segundo mejor tiempo en formato YYYY-MM-DDThh:mm:ssZ.
Time_2	Tiempo en segundos para el tercer mejor tiempo registrado.
Country_2	País o países de origen del jugador o jugadores con el tercer mejor tiempo registrado.
Players_2	Nombre del jugador o jugadores con el tercer mejor tiempo registrado.
Record_Date_2	Fecha y hora de realización del tercer mejor tiempo en formato YYYY-MM-DDThh:mm:ssZ.

Tabla 2. Descripción de los campos para la información de los registros por categorías en el archivo CSV `category_data`.

Para evitar problemas de buffer en la lectura de los archivos se recomienda aumentar el tamaño de los campos de lectura de la librería **Python CSV** al máximo posible para el sistema con el siguiente comando en la librería CSV en el `controler.py` del Reto.

```
import csv

...

csv.field_size_limit(2147483647)
```

En algunos casos experimentales puede que Python y el IDE declaren que se alcanzó el límite de recursión con un mensaje `"RecursionError: maximum recursion depth exceeded in comparison"`, en este caso se recomienda actualizar en el `view.py` este límite con las siguientes líneas de código:

```
import sys

...

default_limit = 1000

sys.setrecursionlimit(default_limit*10)
```

Trabajo Propuesto

Parte 1: Configuración Repositorio

Complete los siguientes pasos para configurar su repositorio de trabajo:

1. Cree en GitHub un repositorio basado en la plantilla propuesta para el reto, el cual se encuentra en el URL: <https://github.com/ISIS1225DEVs/Reto3-Template>
2. Renombre el repositorio de su reto con el formato **Reto2-G<<Número del grupo>>** ej.: **Reto2-G01** para el grupo 1 de la sección 2.
3. Edite el **README** del repositorio e incluya los nombres completos, correo Uniandes y códigos de los miembros del equipo de trabajo.
4. Realice el procedimiento según lo aprendido en clase para clonar el repositorio en su máquina local y sincronizarlo con su repositorio en GitHub.
5. Descargue los datos desde la sección unificada del curso y cópielos en la carpeta **data** del repositorio local.

Parte 2: Carga de Datos

En la sección unificada en la sección de reto 1 encontrarán los datos oficiales del proyecto. el ZIP contiene varios archivos con los sufijos -small, -large, -5pct, -10pct, -20pct, -30pct, -50pct y -80pct. Son archivos con diferente número de registros. (ej.: el archivo `category_data_utf-20pct.csv` contiene un quinto de los datos con 5067 registros y el archivo `category_data_utf-8-large.csv` contiene la totalidad de los datos con 25334 registros). Esto facilita la implementación y pruebas en computadores con memoria RAM y procesadores reducidos.

Para responder a los requerimientos presentados deberán cargar la información de los archivos entregados; recuerde que solo se permite leer una vez la información de cada archivo y que las pruebas finales sobre sus algoritmos serán sobre los archivos **-large.csv**.

Al final de la carga de datos debe reportar los siguientes datos:

- El total de videojuegos cargados desde el archivo de videojuegos.
- Mostrar los primeros 3 y últimos 3 registros de videojuegos cargados con las siguientes características:
 - ID del videojuego.
 - Nombre del videojuego.
 - Género del videojuego.
 - Plataformas en la que está disponible el videojuego.
 - Cantidad total intentos para romper el récord en el videojuego.
 - Fecha de publicación original del videojuego.
- El total de los registros cargados desde el archivo registro de speedrun.
- Mostrar los primeros 3 y últimos 3 registros de categorías de speedruns cargados con las siguientes características:
 - ID del videojuego.
 - Nombre del videojuego.
 - Nombre de la categoría.
 - Nombre de la subcategoría.
 - Nombre del jugador con mejor tiempo registrado.
 - nacionalidad del jugador con mejor tiempo.
 - Tiempo obtenido y fecha cuando se obtiene el mejor tiempo en el registro.

Recomendaciones:

- El contenido de la fecha de lanzamiento del videojuego (*Release_Date*) y los tiempos obtenidos en una categoría para ese videojuego (*Record_Date_0*, *Record_Date_1*, *Record_Date_2*). Tiene en dos formatos de Date time diferentes. Recuerde configurarlos de acuerdo con lo especificado en la Tabla 1 y la Tabla 2.
- Se recomienda utilizar librerías por extensión de Python como `tabulate`⁴ para imprimir adecuadamente los resultados como se muestran en los ejemplos.

Nota: Los ejemplos del documento están hechos con el subconjunto de datos más pequeño ("`_small.csv`").

EJEMPLO: Carga de archivos `-small`.

```
Loaded Speedrunning data properties...
Total loaded videogames: 103
Total loaded category records: 126
```

The first 3 and last 3 videogames loaded in ADTs are...

Data from games displayed as read from CSV file

Game_Id	Release_Date	Name	Abbreviation	Platforms	Total_Runs	Genres
597	2021-07-21	Super Monkey Ball: Banana Mania	smbbm	PlayStation 4, Xbox One, PC, Switch, PlayStation 5, Xbox Series X, Xbox Series S	236	3D Platformer
113	1999-09-30	Crash Team Racing	ctr	PlayStation 2, PlayStation, PlayStation 3, PlayStation Portable, PlayStation Vita, Playstation TV, PSN Download	611	Racing
474	2018-05-03	Animal Crossing	ac	GameCube, Wii	192	Misc
948	2005-10-24	Shadow the Hedgehog	shadowthehedgehog	GameCube, PlayStation 2, Xbox	108	Action
855	2021-04-04	Valheim	valheim	Unknown	148	Sandbox
646	2018-01-01	Mega Man X6	mmx6	GameCube, PlayStation 2, PlayStation, PlayStation 4, Xbox One, PC, PSN Download, Switch	155	2D Platformer

The first 3 and last 3 category records loaded in ADTs are...

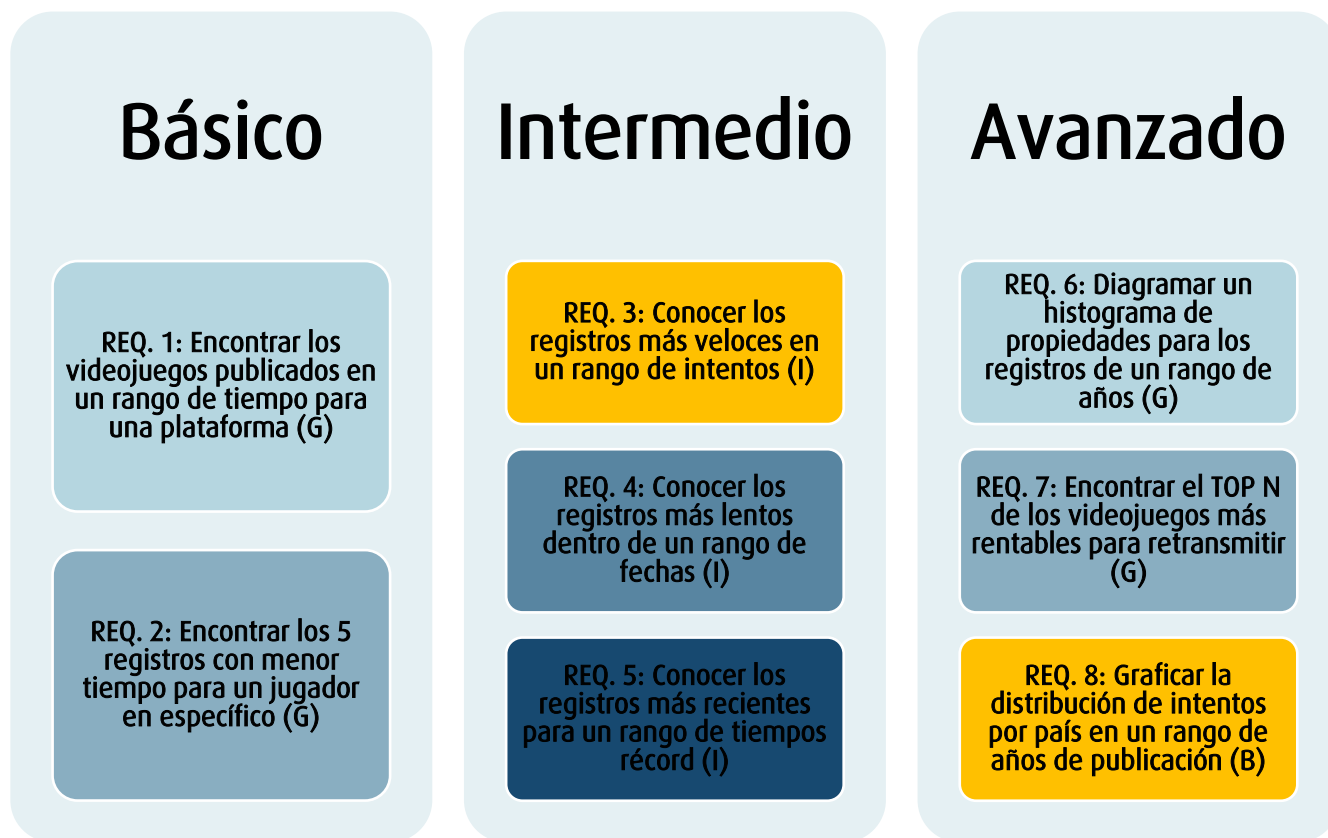
Data from games displayed as read from CSV file

Game_Id	Record_Date_0	Num_Runs	Name	Category	Subcategory	Country_0	Players_0	Time_0
476	2022-03-23 02:09:18	14	Hitman 3	Season 1	SA/SO, Pro, Standard	United Kingdom	moodehh	252.00
5	2021-08-13 21:15:53	4	Seterra	193 UN Member States	Multiple Choice, Website	Poland	Imperator_Maximus	263.00
916	2021-12-31 04:43:58	1	Final Fantasy Record Keeper	DragonKing	FFXIII	Brazil	FelipeNascimento83	28.70
19	2022-01-28 21:56:18	25	ROBLOX: Piggy	Chapter 9: City	2-Player	Portugal, Australia	3hhh, barhdye	87.00
971	2019-10-30 04:12:25	3	Tony Hawk's Pro Skater	Any% New Game+	Unknown	United States	Nami	89.00
477	2022-03-29 09:34:20	82	Octodad: Dadliest Catch	Any% PC	Unknown	United States	theTrevdog	788.89

⁴ Python's `tabulate`, PyPi URL: <https://pypi.org/project/tabulate/>

Parte 3: Desarrollo de los Requerimientos

El resumen de los requerimientos se muestra en la siguiente tabla y se explican detalladamente en la siguiente sección.



Para este reto se han identificado ocho (8) requerimientos, siete (7) obligatorios y un (1) bono. Divididos de acuerdo con su dificultad en nivel básico, intermedio y avanzado. Adicionalmente, es importante resaltar que tres (3) de estos requerimientos se deben desarrollar de forma individual y los cinco (5) restantes son grupales.

NOTA: En caso de que el equipo este conformado solamente por dos integrantes deberán resolver solamente seis (6) requerimientos obligatorios; el equipo podrá escoger cuál de los requerimientos básicos va a realizar.

Requerimiento No. 1 (Grupal): Encontrar los videojuegos publicados en un rango de tiempo para una plataforma

Como jugador de **Deseo** encontrar los videojuegos de una plataforma lanzados en un rango de fechas.

Los **parámetros de entrada** de este requerimiento son:

- Plataforma (Platforms)
- Límite inferior de fecha de lanzamiento (Release_Date).
- Límite superior de fecha de lanzamiento (Release_Date).

La **respuesta esperada** debe contener:

- El número total de videojuegos disponibles en la plataforma.
- El número de videojuegos disponibles en el rango de fechas de publicación para la plataforma.
- Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los videojuegos deberá mostrar la siguiente información:
 - Nombre del videojuego.
 - Abreviación.
 - Género del videojuego
 - Plataformas disponibles.
 - Cantidad total intentos para romper un récord en el videojuego.
 - Fecha de publicación.

Recomendación:

- En caso de que un videojuego esté disponible en más de una plataforma debe contarlo para cada plataforma diferente
- Cuando existan varios videojuegos con la misma fecha de publicación (Release_Date), utilice la abreviación de este en orden alfabético (Abbreviation) y el nombre completo (Name) como criterio completo de ordenamiento.

EJEMPLO: Deseo conocer los videojuegos publicados entre 2000-01-01 y el 2005-12-31 disponibles en la plataforma "PC".

===== Req No. 1 Inputs =====						
Games released between '2000-01-01' and '2005-12-31'						
In platform: 'PC'						
===== Req No. 1 Answer =====						
Available games in 'PC': 74						
Date range between '2000-01-01' and '2005-12-31'						
Released games: 10						
----- Videogames release details -----						
There are '9' elements in range.						
The first 3 and last 3 in range are:						
+-----+-----+-----+-----+-----+-----+-----+						
Release_Date Count Details						
+-----+-----+-----+-----+-----+-----+-----+						
2005-11-15	1	+-----+-----+-----+-----+-----+-----+-----+				
		Total_Runs	Name	Abbreviation	Platforms	Genres
		+-----+-----+-----+-----+-----+-----+-----+				
		55	CrossCode	crosscode	PlayStation 4,	RPG
					Xbox One, PC,	
2004-12-20	1	+-----+-----+-----+-----+-----+-----+-----+				
		Total_Runs	Name	Abbreviation	Platforms	Genres
		+-----+-----+-----+-----+-----+-----+-----+				
		86	Iconoclasts	iconoclasts	PlayStation	Puzzle
					Vita,	
2003-12-30	1	+-----+-----+-----+-----+-----+-----+-----+				
		Total_Runs	Name	Abbreviation	Platforms	Genres
		+-----+-----+-----+-----+-----+-----+-----+				
		408	Diablo III: Reaper of Souls	d3_ros	PlayStation 3,	Action
					Xbox 360,	
2001-06-27	1	+-----+-----+-----+-----+-----+-----+-----+				
		Total_Runs	Name	Abbreviation	Platforms	Genres
		+-----+-----+-----+-----+-----+-----+-----+				
		914	Diablo II: Lord of Destruction	d2lod	PC	Action
2000-11-06	1	+-----+-----+-----+-----+-----+-----+-----+				
		Total_Runs	Name	Abbreviation	Platforms	Genres
		+-----+-----+-----+-----+-----+-----+-----+				
		257	Bloodstained: Curse of the Moon	bscotm	Nintendo 3DS,	Action
					PlayStation	
2000-09-13	1	+-----+-----+-----+-----+-----+-----+-----+				
		Total_Runs	Name	Abbreviation	Platforms	Genres
		+-----+-----+-----+-----+-----+-----+-----+				
		334	Slendytubbies	slendytubbies	PC	Horror

Requerimiento No. 2 (Grupal): Encontrar los 5 registros con menor tiempo para un jugador en específico

Como jugador **Deseo** encontrar los 5 registros con lapsos más cortos para un jugador específico que obtuvo el mejor tiempo.

ordenados por tiempo obtenido del **mejor tiempo** de menor a mayor.

Los **parámetros de entrada** de este requerimiento son:

- Nombre del jugador (Player_0).

La **respuesta esperada** debe contener:

- El número de registros del jugador en donde obtuvo el **mejor tiempo** (Time_0).
- El número de intentos (Num_Runs) que ha realizado el jugador para obtener el mejor tiempo.
- Los 5 registros con el menor tiempo (Time_0) registrado por el jugador. Cada uno de los registros deberá mostrar la siguiente información:
 - Nombre del videojuego.
 - Nombre de la categoría.
 - Nombre de la subcategoría.
 - Número de intentos de romper el récord.
 - Nombre del jugador
 - Nacionalidad del jugador
 - Tiempo obtenido y fecha en que obtuvo del récord.

Recomendación:

- Ordene los registros del jugador por el criterio compuesto de mejor tiempo (Time_0), fecha en que obtuvo el récord (Record_Date_0) y nombre del videojuego (Name).

EJEMPLO: Deseo conocer los registros de speedrun más veloces para el jugador *Flamming*.

```
===== Req No. 2 Inputs =====
Speedrun records for player: 'Flamming'

===== Req No. 2 Answer =====
Player 'Flamming' has '5' Speedrun record attempts

----- Player 'Flamming' details -----
There are only '2' elements in range.
```

Time_0	Record_Date_0	Name	Players_0	Country_0	Num_Runs	Platforms	Genres	Category	Subcategory
1762.00	2021-03-09 15:57:00	Castle Crashers	Flamming	United States	2	PlayStation 3, Xbox 360, PlayStation 4, Xbox One, PC, Switch	Brawler	Any% Bear	Unknown
3346.00	2019-05-17 19:58:15	Castle Crashers	Flamming	United States	3	PlayStation 3, Xbox 360, PlayStation 4, Xbox One, PC, Switch	Brawler	All Arenas	Unknown

Requerimiento No. 3 (Individual): Conocer los registros más veloces en un rango de intentos

Como jugador **Deseo** conocer los registros con menor duración dentro de un rango de los intentos para romper el récord de tiempo.

Los **parámetros de entrada** de este requerimiento son:

- Límite inferior del número de intentos para romper el récord (Num_Runs).
- Límite superior del número de intentos para romper el récord (Num_Runs).

La **respuesta esperada** debe contener:

- El número de registros que cumplen con los criterios del rango de búsqueda.
- Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los registros deberá mostrar la siguiente información:
 - Nombre del videojuego.
 - Nombre de la categoría.
 - Nombre de la subcategoría.
 - Número de intentos de romper el récord.
 - Nombre del jugador
 - Nacionalidad del jugador
 - Valor del mejor tiempo obtenido y fecha en que obtuvo del récord.

Recomendación:

- Ordene los registros de cada elemento dentro del rango por el criterio compuesto del mejor tiempo (Time_0), la fecha en que se obtuvo el récord (Record_Date_0) y el nombre del videojuego (Name).

EJEMPLO: Deseo conocer los registros más lentos que estén en un rango entre **21** y **75** intentos de romper el récord

```

===== Req No. 3 Inputs =====
Category records between '21' and '75' attempts.

===== Req No. 3 Answer =====
Attempts between '21' and '75'
Total records: 10

----- Videogames release details -----
There are only '5' elements in range.
  
```

Num_Runs	Count	Details									
		Time_0	Record_Date_0	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date
22	2	4263.00	2019-08-21 14:21:36	Metroid Fusion	CScottyM	United States	Game Boy Advance, Nintendo 3DS, Wii U Virtual Console	Action-Adventure	Any% Easy	Unknown	21-11-2004
		1258.58	2021-02-01 08:29:14	Half-Life: Opposing Force	Tecna	Russia	PC, Macintosh, Linux	First-person shooter	WON (RTA)	Scripted	23-08-2010
25	3	1503.00	2021-10-16 23:04:25	Luigi's Mansion: Dark Moon	blarz	United States	Nintendo 3DS	Action-Adventure	Gloomy Manor	Unknown	05-03-2001
		657.98	2020-11-08 00:36:13	Shadow the Hedgehog	BlazinZzetti	United States	GameCube, PlayStation 2, Xbox	Action	Glitchless	Neutral	24-10-2005
		87.00	2022-01-28 21:56:18	ROBLOX: Piggy	3hhh, barhdye	Portugal, Australia	Xbox, PC, Android, iOS, Macintosh, HTC Vive, Oculus VR	Puzzle	Chapter 9: City	2-Player	23-01-2020
36	1	224.40	2021-11-13 04:45:37	Spelunky 2	d_tea	Canada	PlayStation 4, Xbox One, PC, Switch, PlayStation 5, Xbox Series X, Xbox Series S	Roguelike	No Gold Sunken City%	Unknown	15-09-2020
49	2	1641.00	2020-11-18 03:28:56	Ib	plank	South Korea	PC	RPG	Promise of Reunion	Unknown	01-01-1987
		1136.10	2022-02-04 03:02:32	The Forest	AwezumMan, Unknown	United States, Unknown	PlayStation 4, PC	Horror	Coop Glitchless any%	Normal	01-01-1999
56	2	3931.00	2016-11-02 05:02:28	Kingdom Hearts: Birth by Sleep Final Mix HD	TheNannerpuss12	United States	PlayStation 3, PlayStation 4, Xbox One, PC, PlayStation 5, Xbox Series X	RPG	Any% PS3,Aqua,Critica	1/Level 1	28-06-2019
		470.45	2020-03-28 02:42:16	Mario Kart Wii	batcake	United States	Wii U, Wii, Wii Mini	Racing	Banana Cup	No Skips	10-04-2008

Requerimiento No. 4 (Individual): Conocer los registros más lentos dentro de un rango de fechas

Como jugador **Deseo** conocer los registros de los mejores tiempos con mayor duración para un rango de fechas: horas en donde se obtuvo.

Los **parámetros de entrada** de este requerimiento son:

- Límite inferior de la fecha: hora en que se obtuvo el récord (Record_Date_0).
- Límite superior de la fecha: hora de que se obtuvo el récord (Record_Date_0).

La **respuesta esperada** debe contener:

- El número de registros que cumplen con los criterios del rango de búsqueda.
- Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los registros deberá mostrar la siguiente información:
 - Nombre del videojuego.
 - Nombre de la categoría.
 - Nombre de la subcategoría.
 - Número de intentos de romper el récord.
 - Nombre del jugador
 - Nacionalidad del jugador
 - Valor del mejor tiempo obtenido y fecha en que obtuvo el récord.

Recomendación:

- Ordene los registros de cada elemento dentro del rango por el criterio compuesto por el mejor tiempo obtenido (Time_0), el número de intentos para romper el récord (Num_Runs), y el nombre del videojuego (Name) .

EJEMPLO: Deseo conocer los registros más lentos obtenidos entre las 04:03 del 2021-10-17 y las 15:48 del 2020-12-31.

===== Req No. 4 Inputs =====

Category records between '2019-03-06T04:03:53Z' and '2021-10-17T15:48:00Z' datetime.

===== Req No. 4 Answer =====

Attempts between '2019-03-06T04:03:53Z' and '2021-10-17T15:48:00Z'

Total records: 80

----- Videogames release details -----

There are '80' elements in range.

The first 3 and last 3 in range are:

Record_Date_0	Count	Details									
2021-10-17 15:48:00	1	Num_Runs	Time_0	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date
		15	90.00	Plague Inc.	Sam_Mercedez	Australia	Android, iOS	Strategy	Main Game	Casual,Virus,iOS	14-10-2003

2021-10-16 23:04:25	1	Num_Runs	Time_0	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date
		25	1503.00	Luigi's Mansion: Dark Moon	blarz	United States	Nintendo 3DS	Action-Adventure	Gloomy Manor	Unknown	05-03-2001

2021-10-02 17:22:43	1	Num_Runs	Time_0	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date
		3	480.00	Call of Duty: Black Ops Cold War Zombies	valley172, Mhyke	Canada, Canada	PlayStation 4, Xbox One, PC, Xbox One X, PlayStation 4 Pro, Xbox One S, PlayStation 5, Xbox Series X, Xbox Series S	First-person shooter	Firebase Z	2P Co-Op,Round 15,Rampage	12-11-2020

2019-03-19 11:10:19	1	Num_Runs	Time_0	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date
		1	3498.67	Assault Android Cactus	MrBond	United States	PlayStation 4, Xbox One, PC, Switch	Action	All Androids Boss Rush	Unknown	01-01-2014

2019-03-11 11:25:10	1	Num_Runs	Time_0	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date
		2	5578.00	Iconoclasts	DLKurosh	Unknown	PlayStation Vita, PlayStation 4, Xbox One, PC, Switch	Puzzle	100%	Relaxed	20-12-2004

2019-03-06 04:03:53	1	Num_Runs	Time_0	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date
		2	397.00	Pokémon Puzzle Challenge	FFRPro21	Canada	Game Boy Color, Nintendo 3DS, Virtual Console, Game Boy Player	Puzzle	Garbage!	Normal,99,999 points,Scroll Off	02-08-1991

Requerimiento No. 5 (Individual): Conocer los registros más recientes para un rango de tiempos récord

Como jugador **Deseo** conocer los registros del mejor tiempo más recientes que estén entre un rango de tiempos.

Los **parámetros de entrada** de este requerimiento son:

- Límite inferior de la duración para el **mejor tiempo** registrado (Time_0).
- Límite superior de la duración para el **mejor tiempo** registrado (Time_0).

La **respuesta esperada** debe contener:

- El número de registros que cumplen con los criterios del rango de búsqueda.
- Los 3 primeros y últimos registros disponibles en dicho rango. Cada uno de los registros deberá mostrar la siguiente información:
 - Nombre del videojuego.
 - Nombre de la categoría.
 - Nombre de la subcategoría.
 - Número de intentos de romper el récord.
 - Nombre del jugador
 - Nacionalidad del jugador
 - Valor del mejor tiempo obtenido y fecha en que obtuvo del récord.

Recomendación:

- Ordene los registros de cada elemento dentro del rango por el criterio compuesto de la fecha en que obtuvo el récord (Record_Date_0), del número de intentos para romper el récord (Num_Runs) y el nombre del videojuego (Name).

EJEMPLO: Deseo conocer los mejores registros de speedrun más recientes que estén en un rango entre 542.10 y 1887.50 segundos.

===== Req No. 5 Inputs =====
Category records between '542.1' and '1887.5' runtime.

===== Req No. 5 Answer =====
Attempts between '542.1' and '1887.5'
Total records: 32

----- Videogames release details -----
There are '32' elements in range.
The first 3 and last 3 in range are:

Time_0	Count	Details																				
593.00	1	<table><tr><th>Record_Date_0</th><th>Num_Runs</th><th>Name</th><th>Players_0</th><th>Country_0</th><th>Platforms</th><th>Genres</th><th>Category</th><th>Subcategory</th><th>Release_Date</th></tr><tr><td>2019-04-22 09:34:28</td><td>3</td><td>League of Legends</td><td>PEEPPEEP00P0026</td><td>Eswatini</td><td>PC, Macintosh</td><td>Strategy</td><td>50 Kills</td><td>Season 9</td><td>01-01-2009</td></tr></table>	Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date	2019-04-22 09:34:28	3	League of Legends	PEEPPEEP00P0026	Eswatini	PC, Macintosh	Strategy	50 Kills	Season 9	01-01-2009
Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date													
2019-04-22 09:34:28	3	League of Legends	PEEPPEEP00P0026	Eswatini	PC, Macintosh	Strategy	50 Kills	Season 9	01-01-2009													
618.10	1	<table><tr><th>Record_Date_0</th><th>Num_Runs</th><th>Name</th><th>Players_0</th><th>Country_0</th><th>Platforms</th><th>Genres</th><th>Category</th><th>Subcategory</th><th>Release_Date</th></tr><tr><td>2020-05-28 08:26:38</td><td>2</td><td>Overwatch</td><td>Koi, Adamlash, NasalNozzle, goadiroth</td><td>United States, Canada, United Kingdom, United States</td><td>PlayStation 4, Xbox One, PC, Switch, PlayStation 5, Xbox Series X, Xbox Series S</td><td>First-person shooter</td><td>Uprising (All Heroes)</td><td>PC, Glass Cannon</td><td>24-05-2016</td></tr></table>	Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date	2020-05-28 08:26:38	2	Overwatch	Koi, Adamlash, NasalNozzle, goadiroth	United States, Canada, United Kingdom, United States	PlayStation 4, Xbox One, PC, Switch, PlayStation 5, Xbox Series X, Xbox Series S	First-person shooter	Uprising (All Heroes)	PC, Glass Cannon	24-05-2016
Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date													
2020-05-28 08:26:38	2	Overwatch	Koi, Adamlash, NasalNozzle, goadiroth	United States, Canada, United Kingdom, United States	PlayStation 4, Xbox One, PC, Switch, PlayStation 5, Xbox Series X, Xbox Series S	First-person shooter	Uprising (All Heroes)	PC, Glass Cannon	24-05-2016													
657.98	1	<table><tr><th>Record_Date_0</th><th>Num_Runs</th><th>Name</th><th>Players_0</th><th>Country_0</th><th>Platforms</th><th>Genres</th><th>Category</th><th>Subcategory</th><th>Release_Date</th></tr><tr><td>2020-11-08 00:36:13</td><td>25</td><td>Shadow the Hedgehog</td><td>BlazinZretti</td><td>United States</td><td>GameCube, PlayStation 2, Xbox</td><td>Action</td><td>Glitchless</td><td>Neutral</td><td>24-10-2005</td></tr></table>	Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date	2020-11-08 00:36:13	25	Shadow the Hedgehog	BlazinZretti	United States	GameCube, PlayStation 2, Xbox	Action	Glitchless	Neutral	24-10-2005
Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date													
2020-11-08 00:36:13	25	Shadow the Hedgehog	BlazinZretti	United States	GameCube, PlayStation 2, Xbox	Action	Glitchless	Neutral	24-10-2005													
1823.00	1	<table><tr><th>Record_Date_0</th><th>Num_Runs</th><th>Name</th><th>Players_0</th><th>Country_0</th><th>Platforms</th><th>Genres</th><th>Category</th><th>Subcategory</th><th>Release_Date</th></tr><tr><td>2021-02-28 11:18:51</td><td>11</td><td>Call of Duty: Black Ops III Zombies</td><td>Unknown</td><td>Unknown</td><td>PlayStation 3, Xbox 360, PlayStation 4, Xbox One, PC</td><td>First-person shooter</td><td>Shadows of Evil</td><td>No Gums, 1 Player, Round 30</td><td>06-11-2015</td></tr></table>	Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date	2021-02-28 11:18:51	11	Call of Duty: Black Ops III Zombies	Unknown	Unknown	PlayStation 3, Xbox 360, PlayStation 4, Xbox One, PC	First-person shooter	Shadows of Evil	No Gums, 1 Player, Round 30	06-11-2015
Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date													
2021-02-28 11:18:51	11	Call of Duty: Black Ops III Zombies	Unknown	Unknown	PlayStation 3, Xbox 360, PlayStation 4, Xbox One, PC	First-person shooter	Shadows of Evil	No Gums, 1 Player, Round 30	06-11-2015													
1864.00	1	<table><tr><th>Record_Date_0</th><th>Num_Runs</th><th>Name</th><th>Players_0</th><th>Country_0</th><th>Platforms</th><th>Genres</th><th>Category</th><th>Subcategory</th><th>Release_Date</th></tr><tr><td>2020-03-28 13:09:40</td><td>3</td><td>OneShot</td><td>Tbantzt</td><td>Indonesia</td><td>PC</td><td>RPG</td><td>2016 Release</td><td>100%, Glitchless</td><td>01-01-2002</td></tr></table>	Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date	2020-03-28 13:09:40	3	OneShot	Tbantzt	Indonesia	PC	RPG	2016 Release	100%, Glitchless	01-01-2002
Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date													
2020-03-28 13:09:40	3	OneShot	Tbantzt	Indonesia	PC	RPG	2016 Release	100%, Glitchless	01-01-2002													
1879.00	1	<table><tr><th>Record_Date_0</th><th>Num_Runs</th><th>Name</th><th>Players_0</th><th>Country_0</th><th>Platforms</th><th>Genres</th><th>Category</th><th>Subcategory</th><th>Release_Date</th></tr><tr><td>2021-07-04 22:25:43</td><td>8</td><td>Silent Hill 3</td><td>MrMcsqueazy</td><td>United Kingdom</td><td>PlayStation 2, PlayStation 3, Xbox 360, PC</td><td>Horror</td><td>Extra New Game</td><td>Console</td><td>01-01-2013</td></tr></table>	Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date	2021-07-04 22:25:43	8	Silent Hill 3	MrMcsqueazy	United Kingdom	PlayStation 2, PlayStation 3, Xbox 360, PC	Horror	Extra New Game	Console	01-01-2013
Record_Date_0	Num_Runs	Name	Players_0	Country_0	Platforms	Genres	Category	Subcategory	Release_Date													
2021-07-04 22:25:43	8	Silent Hill 3	MrMcsqueazy	United Kingdom	PlayStation 2, PlayStation 3, Xbox 360, PC	Horror	Extra New Game	Console	01-01-2013													

Requerimiento No. 6 (Grupal): Diagramar un histograma de propiedades para los registros de un rango de años

Como analista de videojuegos **Deseo** graficar un histograma mostrando la distribución de ciertas propiedades registradas para videojuegos publicados entre años de las fechas de publicación. **IMPORTANTE**, para este requerimiento no se permite el uso de librerías externas.

Los **parámetros de entrada** de este requerimiento son:

- Límite inferior del año de lanzamiento (Tomado del `Release_Date`).
- Límite superior del año de lanzamiento (Tomado del `Release_Date`).
- Numero de segmentos en que se divide el rango de propiedad en el histograma (`N`).
- Numero de niveles en que se dividen las marcas de jugadores en el histograma (`x`).
- La opción de consultar las siguientes propiedades de los registros:
 - El mejor tiempo registrado (`Time_0`), el segundo mejor tiempo (`Time_1`), el tercer mejor tiempo (`Time_2`).
 - El tiempo promedio registrado (el promedio de `Time_0`, `Time_1`, y `Time_2`).
 - El número de intentos registrados (`Num_Runs`).

La **respuesta esperada** debe contener:

- El número total de registros consultados.
- El número total de registros incluidos en el conteo para graficar el histograma.
- Valor mínimo y valor máximo dentro del rango del histograma.
- El histograma con la distribución de la propiedad dividido por rango y niveles.

IMPORTANTE, el conteo⁵ de las propiedades se debe hacer utilizando las funciones de los ADT en la librería `DISCLib`, y puede comprobar el cálculo utilizando las funciones en `Pandas`⁶ o `plotille`⁷.

Recomendación:

- Algunos registros de `speedrun` **NO** poseen los valores para el segundo y tercer mejor tiempo, valide su existencia antes de incluirlo en el conteo.
- Aproxime el conteo de las marcas de nivel (`x`) por su número entero más cercano.
- Tenga cuidado con el residuo al dividir los segmentos del histograma porque puede generar errores de conteo, ajústelo sumando todo el residuo al principio o final del rango.

⁵ Python Histogram Plotting: NumPy, Matplotlib, Pandas & Seaborn, URL: <https://realpython.com/python-histograms/#histograms-in-pure-python>

⁶ `pandas.Series.value_counts`, URL: https://pandas.pydata.org/docs/reference/api/pandas.Series.value_counts.html

⁷ `plotille` 4.0.2, URL: <https://pypi.org/project/plotille/>

EJEMPLO: Graficar el histograma de los **tiempos promedios** para los videojuegos publicados entre los años 2017 y 2021, dividiendo el rango de tiempos promedios en 5 segmentos y cada nivel representa 7 registros.

```
===== Req No. 6 Inputs =====
Count map (histogram) of the feature: 'Time_Avg'
Data between release years of '2017' and '2021'
Number of bins: 5
Registered attemps per scale: 7

===== Req No. 6 Answer =====
There are '40' attempts on record.
Lowest value: '0.14'
Highest value: '162696.50'
The histogram counts '40' attempts.
'Time_Avg' Histogram with '5' bins and '7' attempts per mark lvl.
+-----+-----+-----+-----+
| bin                | count | lvl | mark |
+-----+-----+-----+-----+
| (0.14, 32539.42]   | 38    | 5   | ***** |
+-----+-----+-----+-----+
| (32539.42, 65078.69] | 1     | 0   |      |
+-----+-----+-----+-----+
| (65078.69, 97617.96] | 0     | 0   |      |
+-----+-----+-----+-----+
| (97617.96, 130157.23] | 0     | 0   |      |
+-----+-----+-----+-----+
| (130157.23, 162696.50] | 1     | 0   |      |
+-----+-----+-----+-----+
NOTE: Each '*' represents 7 attempts.
```

Requerimiento No. 7 (Grupal): Encontrar el TOP N de los videojuegos más rentables para retransmitir

Como dueño de un servicio de streaming **Deseo** encontrar los videojuegos más rentables para retransmitir en mi plataforma. Dicho cálculo de rentabilidad tiene sus pasos específicos descritos a continuación.

Los **parámetros de entrada** de este requerimiento son:

- Plataforma de interés (Platforms).
- TOP N de los videojuegos para retransmitir (N)

La **respuesta esperada** debe contener:

- El número total de videojuegos disponibles en dicha plataforma.
- Los N videojuegos más rentables para transmitir. Cada uno de los videojuegos debe tener la siguiente información:
 - Nombre del videojuego.
 - Abreviación.
 - Género del videojuego
 - Cantidad total intentos para romper un récord en el videojuego.
 - Fecha de publicación.

- Nombres de los jugadores con mejor tiempo.
- Dinero esperado por la transmisión del videojuego⁸

INSTRUCCIONES ADICIONALES (Calculo de rentabilidad)

Para estimar el **dinero esperado** por transmitir un videojuego (**Stream_Revenue**) se deben tener en cuenta la cantidad total de intentos de videojuego, el año de publicación, el tiempo promedio para la transmisión y su representación en el mercado.

Primero, para calcular el **Factor de antigüedad (Antiquity)**, se tiene en cuenta que los juegos clásicos y más modernos tienen mayor interés. Para esto se utiliza la fecha de publicación (Release_Date) en la siguiente función con el **año de publicación (rlsy)**:

$$Antiquity(rlsy) = \begin{cases} rlsy - 2017, & rlsy \geq 2018 \\ -\frac{1}{5}rlsy + 404.6, & 1998 < rlsy < 2018 \\ 5, & rlsy < 1998 \end{cases}$$

Segundo, se calcula el **Factor de Popularidad (Popularity)** utilizando el logaritmo natural del **número de total de intentos del videojuego (Total_Runs)**,

$$Popularity = \ln(Total_Runs)$$

Tercero, se ignoran por completo cualquier intento con una categoría **miscelánea (Misc)** igual a verdadero.

Cuarto, calcular el tiempo **promedio de transmisión (Time_Avg)** como el promedio de los **tiempos récords (Time_0, Time_1, Time_2)** registrados como se muestra en la siguiente función.

$$Time_{Avg} = \text{mean}(Time_0, Time_1, Time_2)$$

Quinto, como se muestra en la formula, calcule el **ingreso esperado por cada transmisión (Revenue)** multiplicando la **popularidad** del juego por el **tiempo promedio en minutos** y dividirlo por su **antigüedad**.

$$Revenue = \frac{Popularity * Time_{Avg}}{Antiquity}$$

Sexto, encuentre la **fracción representativa del mercado (Market_Share)** para cada videojuego utilizando el nombre de la **plataforma (Platforms)** como indicativo. Esto es equivalente a dividir el número de **transmisiones (gt)** registradas de ese juego dentro de esa plataforma sobre el **total de transmisiones (pt)** de juegos de esa plataforma. Ej.: si para la **plataforma "PC"** se encuentran 100 registros y para un juego como **DOTA** hay 20 registros dentro de esos 100, la fracción de mercado para ese juego debe ser 20/100, es decir 0.20.

$$Marketshare = \frac{gt}{pt} = \frac{gt}{\sum platform\ category_i}$$

⁸ Este dinero esperado debe calcularse siguiendo las indicaciones del requerimiento.

Séptimo, multiplique el ingreso esperado y la fracción del mercado para calcular la ganancia esperada por transmitir el videojuego (Stream_Revenue).

$$\text{Streamrevenue} = \text{Revenue} * \text{Marketshare}$$

NOTA 1: Recuerde que los porcentajes de participación y el dinero esperado pueden variar dependiendo el tamaño de los datos utilizado.

NOTA 2: Si un videojuego está en varias plataformas debe contar su participación en todas las plataformas en las que sea relevante.

Recomendaciones:

- Algunos registros de speedrun **NO** tienen los valores del segundo y tercer mejor tiempo, verifique esto antes de calcular el promedio.
- Aproxime todos los cálculos a la segunda cifra decimal. Ej.: 1209.349068 se aproxima a 1209.40.

EJEMPLO: Identificar los videojuegos más rentables de transmitir para la plataforma "PC"

```
===== Req No. 7 Inputs =====
Find the TOP '5' games for 'PC' platform.

Filtering records by platform...
Removing miscellaneous streaming revenue...

===== Req No. 7 Answer =====
There are '75' records for 'PC'...
There are '61' unique games for 'PC'...

----- TOP 5 GAMES FOR PC -----
There are only '5' elements in range.
```

Name	Release_Date	Platforms	Genres	Stream_Revenue	Market_Share	Time_Avg	Total_Runs
Guild Wars 2	2018-11-21	PC	RPG	476.62	0.03	162696.50	727
Red Dead Redemption 2	2018-04-20	PlayStation 4, Xbox One, PC, Google Stadia, PlayStation 5, Xbox Series X, Xbox Series S	Action-Adventure	107.67	0.03	46449.33	130
Final Fantasy VIII	2016-02-25	PlayStation 2, PlayStation, PlayStation 3, PlayStation 4, Xbox One, PC, Playstation TV, Switch, PlayStation 5, Xbox Series X	RPG	48.96	0.01	55586.50	257
Stardew Valley	2018-11-13	PlayStation 4, Xbox One, PC, Android, iOS, Switch	Sandbox	19.95	0.01	13443.51	796
Call of Duty: Black Ops IIII Zombies	2019-02-04	PlayStation 4, Xbox One, PC	First-person shooter	10.22	0.03	4262.00	651

Requerimiento No. 8 (Bono Grupal): Graficar la distribución de intentos por país en un rango de años de publicación

Como jugador de videojuegos frecuente **Deseo** visualizar gráficamente en un mapa de clústeres la distribución de las nacionalidades de los poseedores de **mejores tiempos** de los videojuegos

Los **parámetros de entrada** de este requerimiento son:

- Año de publicación sobre el cual se quiere obtener los histogramas (Tomado del `Release_Date`)
- Límite inferior de la duración del **mejor tiempo** del récord (`Time_0`).
- Límite superior de la duración del **mejor tiempo** del récord (`Time_0`).

La **respuesta esperada** debe contener:

- El número total de registros de speedrun en dicho año y rango.
- Un mapa interactivo de clústeres que muestre todos los registros de speedrun en dicho año y rango.

Para completar este requerimiento recomendamos utilizar la librería por extensión de Python llamada “`folium`” que se puede instalar en su ambiente por medio del comando “`pip install folium`”.

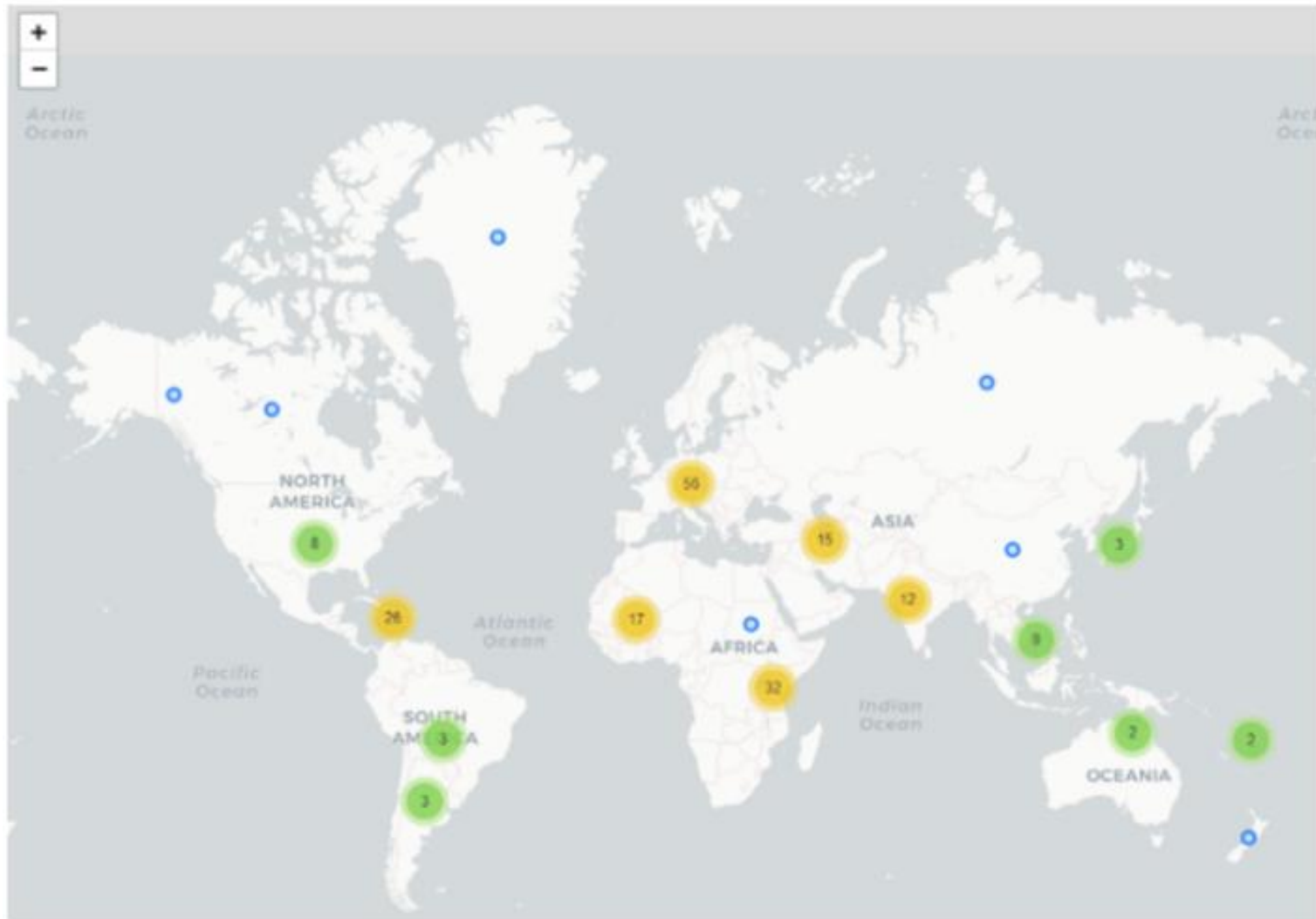
Para más información sobre esta librería dirigirse a los siguientes enlaces:

- Enlace oficial de PYPI, URL: <https://pypi.org/project/folium/>
- Enlace oficial de la librería, URL: <https://github.com/python-visualization/folium>
- Enlace al tutorial de la Librería. URL: <https://python-visualization.github.io/folium/quickstart.html>

Recomendación:

- Utilice los árboles generados para el requerimiento 6 para obtener los rangos.
- **Solamente** puede utilizar librerías para la representación gráfica en el mapa, todas las consultas de datos deben realizarse utilizando DISCLib y las ADTs disponibles para la clase.

EJEMPLO: Deseo el mapa de distribución para los videojuegos publicados en el año **2017** con un rango de tiempos entre **1000** y **5000** segundos.



Parte 4: Análisis de resultados

Dentro del proyecto debe incluir un documento en la carpeta **Docs** en formato **PDF** donde se evidencie el análisis de complejidad, las pruebas de tiempos de ejecución y de memoria utilizada para cada requerimiento. Se sugiere que el documento tenga la siguiente distribución del contenido:

- Nombres, código y correo Uniandes de los integrantes del grupo.
- Para los requerimientos individuales se debe indicar que estudiante del equipo lo realizó.
- Análisis de complejidad de cada uno de los requerimientos en **Notación O**.
- Breve justificación de la complejidad temporal dada.
- Pruebas de tiempos de ejecución y memoria utilizada para cada uno de los requerimientos. En estas pruebas se deben incluir:
 - las tablas de tiempos de ejecución registrados para los requerimientos del 1 al 7 (incluye el bono si aplica).
 - Graficas de los tiempos de ejecución registrados para los requerimientos del 1 al 7 (incluye el bono si aplica).
 - Un análisis de resultados comparándolo con la estimación en **Notación O**.

Recomendaciones:

- Tomen como guía las herramientas, metodología y análisis realizados en los laboratorios; en especial los laboratorios 4 y 5.
- Ejecute las pruebas de los requerimientos siempre con los mismos parámetros de entrada y con los archivos más grande que pueda procesar su computador.
- Ejecute las pruebas de los requerimientos con su configuración optima seleccionada para el catálogo (RBT, BST, LINKED_LIST, ARRAY_LIST, PROBING, CHAINING) y los algoritmos de ordenamiento (Shell, Insertion, Merge, etc.).

Entrega

Para realizar la entrega del reto deben:

1. Agregar los usuarios de los monitores y profesores del curso a su organización de GitHub para hacer la entrega adecuada de la actividad,
2. Dar permisos adecuados repositorio a los monitores y al profesor, de lo contrario el taller NO podrá ser calificado,
3. Asegurarse que la visibilidad del repositorio entregado sea privada y que solo pueda accederse con los permisos configurados para los integrantes del grupo,
4. Enviar el enlace de GitHub en la actividad correspondiente dentro de Bloque Neón Uniandes (BrightSpace).
5. Incluir en el repositorio GIT todo el material, código y documentos solicitados durante la actividad.

IMPORTANTE: Recuerde que solo se calificará el material hasta el último **COMMIT** realizado previo a la fecha límite (**11:59 PM**) del **8 de noviembre de 2022**.