

Creación de una BD de registros usando estrategia de Hash.

## 1. Contexto

Los sistemas de almacenamiento de archivos fueron en su principio, debido a las limitaciones tecnológicas, lineales: es decir, por algún “orden” los datos eran guardados en ellos.

La creación de tablas de Hash fue en su momento una forma de acelerar las búsquedas en aquellos sistemas que no permitían retroceso en el recorrido de datos (especialmente los primeros sistemas de cintas).

Usted va a crear una sistema físico con una tabla hash “vacía”, con varios registros, una función y una búsqueda de datos.

## 2. Implementación y Requerimientos

Usted debe crear un programa que cumpla los siguientes requerimientos:

- Al inicio, se verifica la creación de la tabla física (será un archivo binario LLAMADO “info.dat”, vacío inicialmente aunque lo que hará será tener 750 registros vacíos). Si NO existe, se crea vacío, con un tamaño de 750 items. Si existe, no se hace nada.
- Con la tabla física (info.dat), usted procederá a mostrar un menú, con las siguientes opciones:
  - Cargar datos
  - Buscar datos
  - Salir
- La opción Cargar Datos leerá las jugadas almacenadas en los archivos de salida de la temporada 1. Y calculará con una función hash LA POSICIÓN de dicha jugada en el archivo. Procederá a ver si el registro está vacío y lo está, escribirá ahí. Si el registro NO está vacío, creará otro archivo cuyo nombre usted podrá definir, y guardará ahí CONSECUTIVAMENTE todas las colisiones de llave. Dicho archivo debe quedar en el mismo directorio que “info.dat”. EVIDENTEMENTE puede haber varios archivos de estos. Los archivos conteniendo las colisiones deben tener por nombre “<posición>-col.dat” donde posición es el valor obtenido de la función de hash.
- La opción 2, “Buscar Datos” preguntará al usuario por la llave (un número entre 0 y 749), y procederá a buscar en el archivo info.dat el registro guardado en esa posición. Si no hay nada, pues responde que ese registro está vacío. Si hay algo, lo recupera y procede a ver si hubo colisiones, es decir, si existe el archivo de colisiones. En ese caso, lista tanto el registro almacenado en info.dat como los registros almacenados en el archivo de colisiones.
- Respecto a la función de Hash, debe usar los datos preguntados de la fecha, cuarto y el nombre del equipo local, para distribuir los juegos en los 750 registros posibles o los archivos de colisión.

- La función de Hash puede ser de cualquiera de los 2 algoritmos vistos en clase. Diseñela usted. Debe DOCUMENTAR en el archivo de python (que será un módulo) como la diseñó.

### 3. ENTREGA

#### DEBE ENTREGAR:

- Los archivos .py
- Implementar la función de hash en un módulo, el nombre del archivo debe ser hash\_function.py.
- Obviamente debe reutilizar la clase “punt\_play” de la tarea anterior.
- Haga módulos distintos para manejar el menú de opciones y la escritura/lectura de los archivos de registro.
- Coloque los archivos de ENTRADA y SALIDA en la carpeta “\data\segundaprogramada”, obviamente tendrá que copiar los archivos de salida de la tarea 1 a esta carpeta.
- Recuerde que **NO** puede usar bibliotecas o módulos que requieran **pip install**.
- Más específicamente, recuerde que solo puede usar las librerías y funciones definidas en la PYTHON STANDARD LIBRARY, tal como está publicada en: <https://docs.python.org/3/library/index.html>

FECHA DE ENTREGA: 1 DE MARZO DE 2025 ANTES DE LA MEDIA NOCHE.