

## Tarea Programada 1

---

### Contexto

Las bases de datos contienen múltiples registros organizados en archivos de acceso fácil y rápido con estructuras de datos complejas.

En esta tarea haremos un sistema sencillo que emule los archivos más antiguos, archivos secuenciales de registros. Para simplificar, usaremos registros de tamaño FIJO.

Se le entregarán 2 archivos:

- Un archivo generador de registros que pide al usuario la cantidad de registros a crear y crea los mismos aleatoriamente.
- Un archivo localizador de registros, que recibe la posición del registro en el archivo e imprime el registro en esa posición.

Debe usarlos como referencia para construir los programas que se especifican en los requerimientos.

### Requerimientos

1. Solo puede usar la biblioteca standard de Python.
2. Usando como referencia/base el archivo generador, debe crear un programa que:
  - Tenga a su disposición más datos para generar (más nombres, más provincias y más distritos. Incluso invente algunas 2-3 provincias más)
  - **Agregue un campo de “numero de empleado” que sea un entero POSITIVO**
  - Le pregunte al usuario cuantos registros quiere generar.
  - **Genere el archivo, CON LOS REGISTROS INSERTADOS EN ORDEN ASCENDENTE DE NUMERO EMPLEADO. Dicho numero debe ser aleatorio y NO puede hacerlo consecutivo.**
  - **El archivo DEBE TENER EXTENSIÓN .bin**
3. Usando como referencia el archivo de recuperación de registros:
  - Haga un programa que le pida al usuario:
    - O bien la posición de registro a recuperar
    - O el numero de empleado a recuperar
4. Si el usuario ingresa la posición: va directamente y lo recupera. Observe que el programa de referencia NO LEE UNO POR UNO SINO QUE CALCULA Y MUEVE LA LECTURA A LA POSICIÓN NECESITADA.

5. Si el usuario ingresa el numero de empleado, el programa HACE BUSQUEDA BINARIA calculando la posición y localizando (o no) el registro de la persona con el numero de empleado. Si no lo encuentra debe indicar que no existe.
6. Use módulo para evitar la repetición de código.
7. **Es deseable que use la estructura vista en programación de separar la interacción y la lógica con un “controler” y un proveedor de servicio.**
8. PARA ENTREGAR ESTE PROYECTO, escoja entre:
  - a. Suba los archivos como hace habitualmente al aula virtual (no suba archivos .bin o será penalizado)
  - b. Suba un archivo de texto con el URL del repositorio de github. **DICHO REPOSITORIO DEBE SER PUBLICO. SI ES PRIVADO SE LE PONDRÁ UN CERO.**

FECHA DE ENTREGA: antes de la media noche del 24 de Febrero de 2026.