

Práctica “sistema distribuido para las elecciones presidenciales”



GRUPO: 4CM3

EQUIPO 3

INTEGRANTES:

• MAYA ROCHA LUIS EMMANUEL

• OSUNA BANDA ITZEL ARELY

• MIRANDA MOJICA ERICK

• BARBOSA PEÑA XAVIER MARISTIN

**Ejercicio 1**

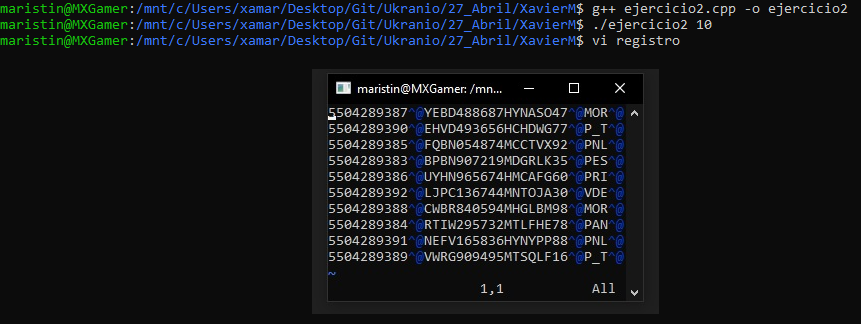
Con este tipo de registro y para 70 millones de votantes

1. ¿cuánto espacio de almacenamiento será necesario?
   1. **2.2 GB**
2. ¿Es posible almacenarlo en su disco duro?
   1. **Sí, es posible.**
3. ¿En una memoria USB?
   1. **Si, también**
4. ¿Y en un archivo virtual dentro de la RAM?
   1. **También, pero depende mucho de las especificaciones del equipo.**

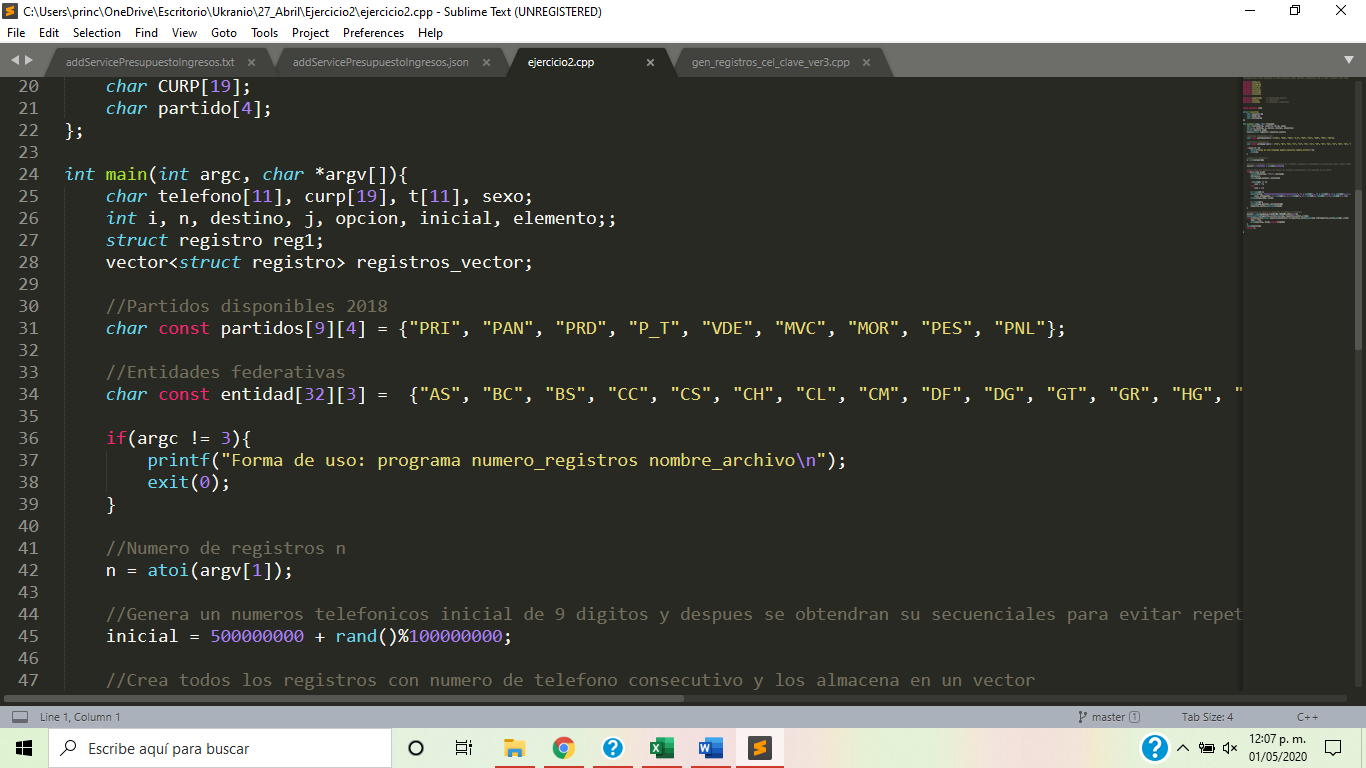
**Ejercicio 2**

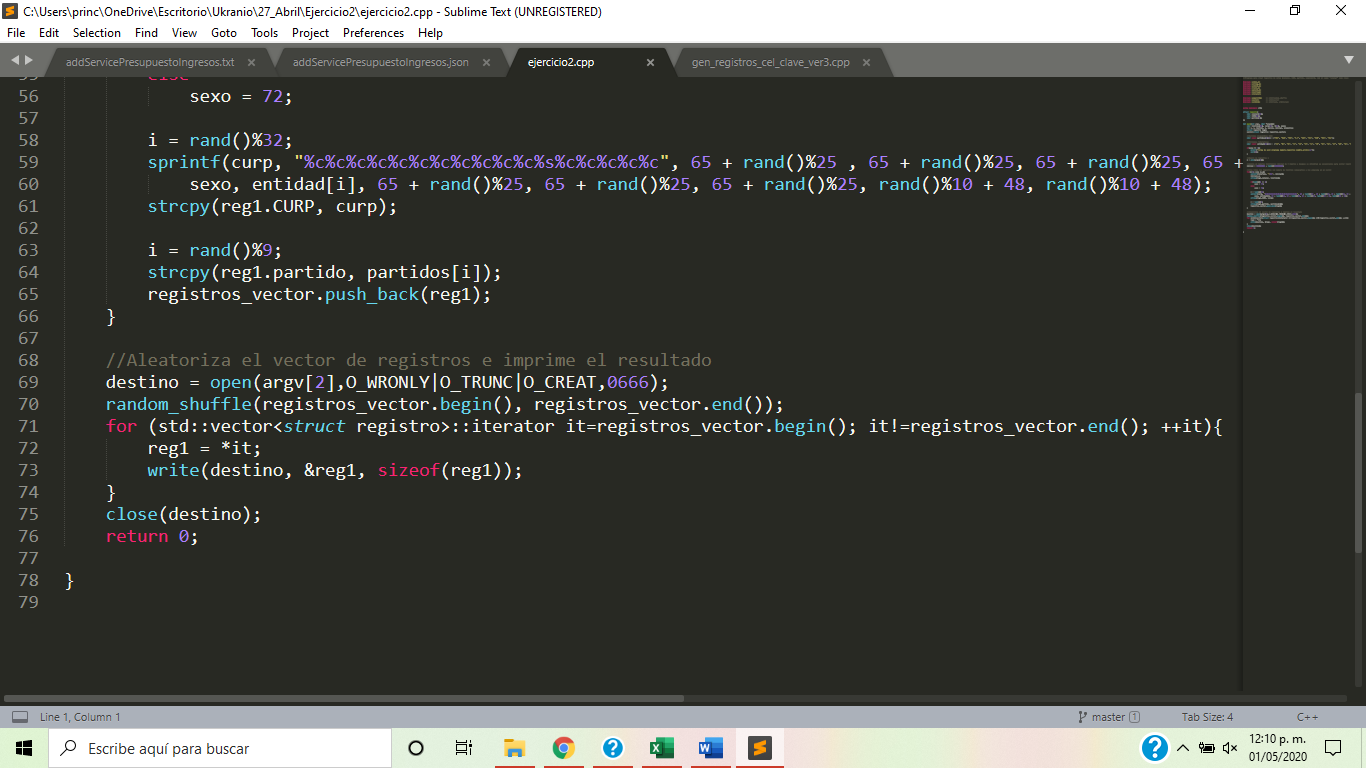
**En mi servidor web se encuentra el programa gen\_registros\_cel\_clave\_ver3.cpp , el cual se encarga de generar n registros de votaciones con información aleatoria, pero evitando que exista un número de teléfono celular repetido. Modifique este programa para que almacene los n registros en un archivo (véase el capítulo 8 del manual de programación de sistemas LINUX), cuyo nombre se reciba como parámetro en la línea de comandos. Usando el comando time tabule los tiempos que demora en almacenar 7000, 70000,700000 y 7000000 registros en un archivo, con la computadora más rápida del equipo. Usaremos estos archivos para pruebas posteriores.**

**Importante: En lugar de enviar a la pantalla la salida con: write(1, &reg1, sizeof(reg1)); envíe el registro reg1 a un archivo, pasándole a write como primer parámetro el descriptor de archivo devuelto por la función open(). Para visualizar el contenido del archivo debe utilizar un editor en la línea de comandos como vi, pico o nano, debido a que el sistema operativo incluye separadores binarios entre cada campo de la estructura.**

****

Modificaciones al programa gen\_registros\_cel\_clave\_ver3.cpp:





Tiempos en los que demora la computadora más rápida del equipo.



|  |  |
| --- | --- |
| #Número de registros | Tiempo de demora |
| 7,000 | 0.040s |
| 70,000 | 0.326s |
| 700,000 | 3.212s |
| 7,000,000 | 32.438s |
| 70,000,000 | 4m51.583s |

**Ejercicio 3**

**Reutilice las clases PaqueteDatagrama y SocketDatagrama para programar un cliente que lee un archivo que contiene un único registro, y lo manda en un mensaje UDP hacia el servidor de base de datos. El servidor de base de datos, después de recibir el registro lo almacena en un archivo cuyo nombre recibe como parámetro en la línea de comandos.**

Cliente envía los datos en mensaje UDP hacia servidor.

**Captura de pantalla de un celular

Descripción generada automáticamente**

Servidor que recibe los datos y lo almacena en un archivo llamado registro pasado por consola

**Captura de pantalla de un celular

Descripción generada automáticamente**

**Mostramos el contenido del archivo**

**Captura de pantalla de un celular con texto

Descripción generada automáticamente**

**Captura de pantalla de un celular con texto

Descripción generada automáticamente**