

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS DEPARTAMENTO DE CIENCIAS COMPUTACIONALES



Seguridad

SECCIÓN D04

INTEGRANTES DE EQUIPO:

- Gildo López Miguel Ángel
- González Ramírez Alan Leonardo
- Montejano Paredes Diego Fernando

PROFESOR: Vázquez Curiel Armida Griselda

26/02/2024

Actividad 2.1 Respaldo

Sistema de respaldo

Se refiere a un conjunto de procedimientos, hardware y software diseñados para proteger y respaldar la información y los datos almacenados en un sistema informático. El propósito principal de un sistema de respaldo es asegurar que los datos críticos estén a salvo.



sean recuperables en caso de pérdida, corrupción, fallos de hardware, ataques de malware u otros desastres.

Los sistemas de respaldo pueden ser programados para realizarse de forma periódica, por ejemplo, diariamente o semanalmente, y pueden ser almacenados en medios externos como memorias USB o en la nube. Esto garantiza que los datos estén disponibles y se puedan restaurar en caso de que el sistema principal falle. Además, los sistemas de respaldo también permiten recuperar los datos de forma rápida y eficiente en caso de un apagón o una falla en el hardware.

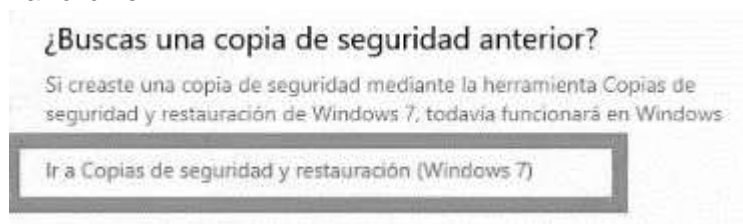
Esto puede ser especialmente importante en entornos comerciales, donde una interrupción prolongada del servicio puede tener un impacto significativo en la productividad de la empresa o local donde se esté ejecutando algún sistema. En resumen, un sistema de respaldo es un componente fundamental de cualquier sistema operativo, ya que proporciona protección y seguridad a los datos y configuraciones importantes, asegurando la continuidad del negocio en caso de una interrupción o fallo del sistema.

Sistema de respaldo de Windows

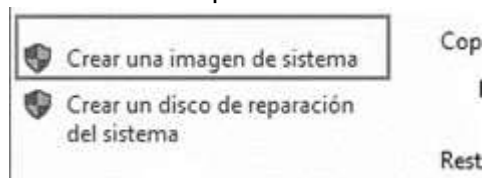
Windows cuenta con una herramienta de respaldo llamada "Copia de seguridad y restauración" que permite realizar copias de seguridad de los archivos y configuraciones importantes en el sistema. Esta herramienta puede ser accedida desde el Panel de control y se puede programar para realizar copias de seguridad periódicas.

Pasos para usar dicha herramienta:

1. Es necesario contar con una unidad con suficiente espacio de almacenamiento para hacer la copia de seguridad, para esto podemos hacer una partición en el mismo disco duro o usar un medio externo.
2. Dirigirse a "Configuración" -> "Actualización y seguridad" -> "Copia de seguridad"
3. Dar click en:



4. Seleccionar la opción:



5. Elegir la opción deseada:

☒ En un disco duro

Backup (E:) 126,89 GB disponibles

☐ En uno o más DVDs

☐ En una ubicación de red

Seleccionar...

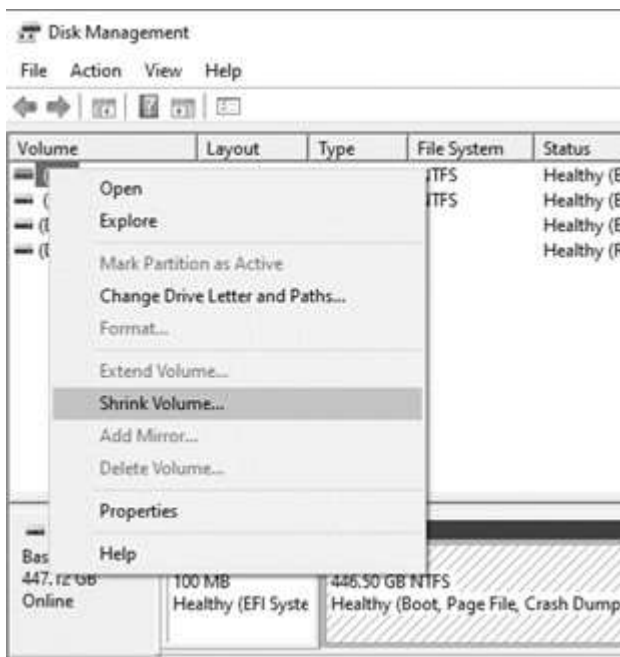
6. Dar click en:

Iniciar la copia de seguridad

7. Una vez terminado el proceso, el sistema estaría respaldado.

Pasos para hacer una partición dentro del mismo disco duro donde se encuentra Windows:

1. Presione la tecla Windows + X y seleccione "Administrador de disco" en el menú desplegable.
2. Seleccione el disco duro que desea particionar y haga clic en "Redimensionar volumen".



3. Arrastre la barra del espacio sin asignar hacia la partición que desea reducir.

4. Enter the amount of space to shrink in MB: 407701

5. Haga clic en "Nuevo volumen simple" y siga las instrucciones en pantalla para crear una nueva partición.

6. Asigne una letra de unidad y un nombre a la partición.

7. Haga clic en "Finalizar" y espere a que se complete el proceso.

Sistema de respaldo en python (Con encriptación simple)

```
def backup_folder(src: str, dst: str, password: str):

    # Obtener nombres de carpetas/archivos
    files = get_filenames(src)

    # Archivo de salida
    output = io.BytesIO()

    # Escribir la propia contraseña en el archivo
    output.write(password.encode('utf-8'))

    # Escribir en 4 bytes la cantidad total de archivos
    output.write(struct.pack('I', len(files)))

    for f in files:

        # Nombre del archivo
        # Sin el nombre de la carpeta raíz
        subpath = os.path.relpath(f, src)

        # Escribir en 4 bytes la longitud del nombre del archivo
        output.write(struct.pack('I', len(subpath)))

        # Escribir el nombre del archivo como tal
        output.write(subpath.encode('utf-8'))

        # Escribir la longitud de los datos
        output.write(struct.pack('I', os.path.getsize(f)))

        # Escribir los datos como tal
        input_file = open(f, 'rb')
        output.write(input_file.read())
        input_file.close()

    ### ENCIPTACION ###

    # Crear lista a partir de los bytes escritos
    byte_array = list(output.getvalue())

    # Generar contraseña y encriptar solo si la proporciona el usuario
    if len(password) != 0:
        # Generar un entero a partir de la contraseña dada
        key = 0
        for c in password:
            key += ord(c)

        # Modificar cada byte, sumandole la llave
        for i in range(len(byte_array)):
            byte_array[i] = (byte_array[i] + key) % 255

    # Escribir el archivo ya encriptado en el disco
    output_file = open(dst, 'wb')
    output_file.write(bytearray(byte_array))
    output_file.close()

    # Mostrar mensaje de éxito
    showinfo('Operación exitosa', 'Los archivos han sido respaldados correctamente')
```

Como se menciona en los comentarios del código, el algoritmo toma el nombre de los archivos de manera recursiva y los guarda en una lista. Posteriormente se crea un objeto de tipo ByteIO, el cual nos permite usarlo como si fuera un archivo. Escribimos en el archivo al principio la contraseña ingresada por el usuario si es que la proporcionó, posteriormente escribimos en 4 bytes la cantidad total de archivos.

Posteriormente el código itera entre cada archivo abriéndolo y repitiendo los siguientes pasos para cada uno de ellos:

- A cada directorio se le resta la parte del directorio raíz, por ejemplo:
c:/Users/Documents/Prueba/Archivo.png, solo dejamos la parte de: Archivo.png
- Escribimos en 4 bytes, la longitud de la string del nombre del archivo
- Escribimos el nombre del archivo (variable)
- Escribimos en 4 bytes, la longitud de los datos del archivo
- Escribimos los datos del archivo (variable)

En este punto tenemos el archivo de la siguiente manera: (En este ejemplo se está usando como contraseña: ABCD1234)

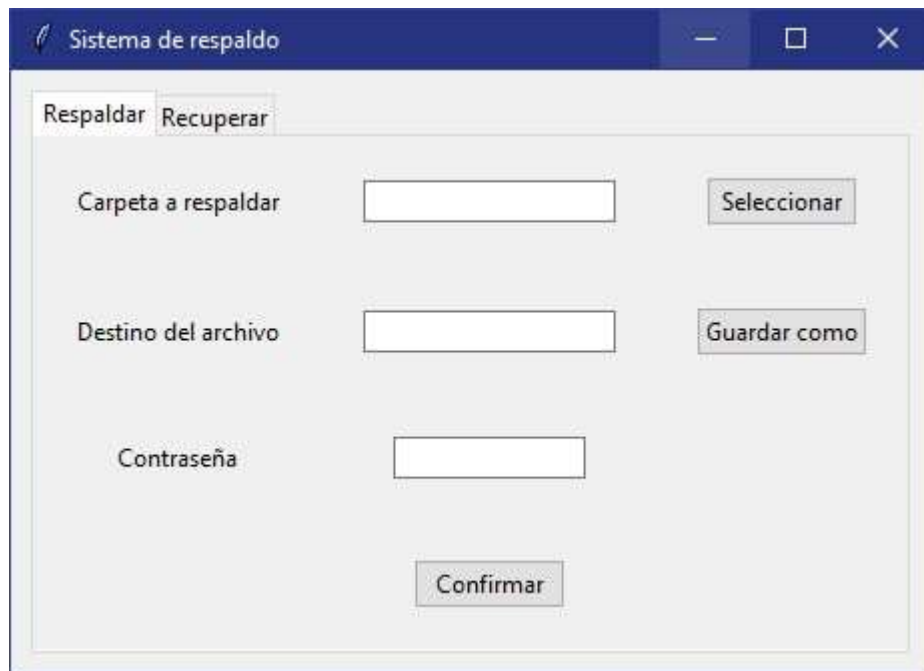
A	B	C	D	1	2	3	4							
				V	A	R						V	A	R
				V	A	R						V	A	R
				V	A	R						V	A	R
				V	A	R						V	A	R
				V	A	R						V	A	R
				V	A	R						V	A	R
				V	A	R						V	A	R

Donde var es tamaño variable y depende de lo que indiquen los enteros.

Encriptación

Para encriptar los archivos básicamente se suma cada valor ascii de la contraseña proporcionada y se guarda en un número, digamos que se genera el número 540, este número se le sumará a cada byte de archivo, modificandolo completamente, sin embargo, esto es reversible restando el mismo valor en el algoritmo de recuperación. Finalmente se escribe el archivo en el disco.

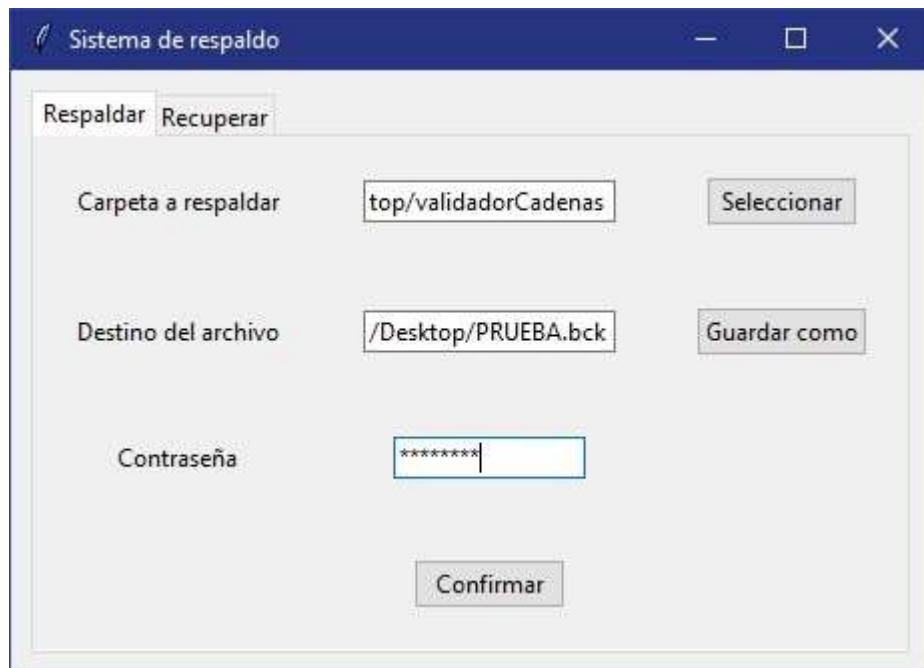
Demostración de uso



The screenshot shows a window titled "Sistema de respaldo" with a dark blue header bar containing a minimize, maximize, and close button. Below the header, there are two tabs: "Respaldo" (selected) and "Recuperar". The main area contains three input fields with corresponding buttons to their right:

- "Carpeta a respaldar" with an empty text box and a "Seleccionar" button.
- "Destino del archivo" with an empty text box and a "Guardar como" button.
- "Contraseña" with an empty text box.

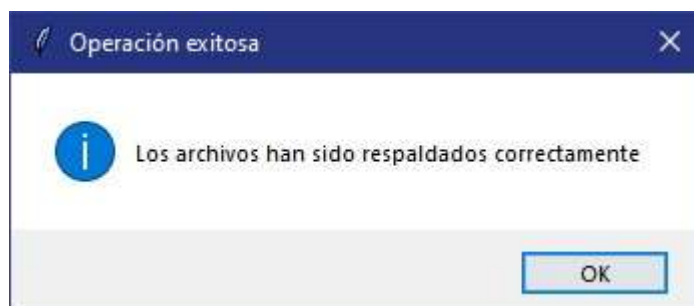
At the bottom center, there is a "Confirmar" button.



This screenshot shows the same "Sistema de respaldo" window, but with the input fields populated:

- "Carpeta a respaldar" contains the text "top/validadorCadenas".
- "Destino del archivo" contains the text "/Desktop/PRUEBA.bck".
- "Contraseña" contains seven asterisks "*****".

The "Confirmar" button remains at the bottom center.



The screenshot shows a small dialog box titled "Operación exitosa" with a dark blue header bar and a close button. The main area features a blue information icon (i) on the left and the text "Los archivos han sido respaldados correctamente" on the right. At the bottom right, there is an "OK" button.

Archivo antes de ser encriptado:


```
ACD1234/.....CMakeLists.txt#...cmake_minimum_required(VERSION 3.27)project(test LANGUAGES CXX)set(CMAKE_CXX_STANDARD 20).set(CMAKE_CXX_STANDARD_REQUIRED ON).set(CMAKE_CXX_EXTENSIONS OFF).add_compile_options(-Wall -pedantic -Wextra).link_libraries(ftwui-component ftwui-screen ftwui-dom).file(GLOB_RECURSE SOURCES CONFIGURE_DEPENDS "${CMAKE_CURRENT_SOURCE_DIR}/src/*.cpp").add_executable(test ${SOURCES}).target_include_directories(test PRIVATE "${CMAKE_CURRENT_SOURCE_DIR}/include").....vscode/settings.json#....["files.associations": [{"charconv": "cpp","csdtdf": "cpp","vector": "cpp","memory": "cpp","bitset": "cpp"}].....build/CMakeCache.txt#9...# This is the CMakeCache file..# For build in directory: c:/Users/angel/Desktop/validadorCadenas/build..# It was generated by CMake: C:/Program Files/CMake/bin/cmake.exe..# You can edit this file to change values found and used by cmake..# If you do not want to change any of the values, simply exit the editor..# If you do want to change a value, simply edit, save, and exit the editor..# The syntax for the file is as follows:..# KEY:TYPE=VALUE..# KEY is the name of a variable in the cache..# TYPE is a hint to GUIs for the type of VALUE, DO NOT EDIT TYPE!..# VALUE is the current value for the KEY.....#####.### EXTERNAL cache entries.#####.....//Path to a program..CMAKE_ADDR2LINE:FILEPATH=C:/mingw64/bin/addr2line.exe.....//Path to a program..CMAKE_AR:FILEPATH=C:/mingw64/bin/ar.exe.....//No help, variable specified on the command line..CMAKE_BUILD_TYPE:STRING=Debug.....//Enable/Disable color output during build..CMAKE_COLOR_MAKEFILE:BOOL=ON.....//No help, variable specified on the command line..CMAKE_CXX_COMPILER:FILEPATH=C:/mingw64/bin/g++.exe.....//A wrapper around 'ar' adding the appropriate '-plugin' option..// for the GCC compiler..CMAKE_CXX_COMPILER_AR:FILEPATH=C:/mingw64/bin/gcc-ar.exe.....//A wrapper around 'ranlib' adding the appropriate '-plugin' option..// for the GCC compiler..CMAKE_CXX_COMPILER_RANLIB:FILEPATH=C:/mingw64/bin/gcc-ranlib.exe.....//Flags used by the CXX compiler during all build types..CMAKE_CXX_FLAGS:STRING=.....//Flags used by the CXX compiler during DEBUG builds..CMAKE_CXX_FLAGS_DEBUG:STRING=g-.....//Flags used by the CXX compiler during MINSIZEREL builds..CMAKE_CXX_FLAGS_MINSIZEREL:STRING=-Os -DNDEBUG.....//Flags used by the CXX compiler during RELEASE builds..CMAKE_CXX_FLAGS_RELEASE:STRING=-O3 -DNDEBUG.....//Flags used by the CXX compiler during RELWITHDEBINFO builds..CMAKE_CXX_FLAGS_RELWITHDEBINFO:STRING=-O2 -g -DNDEBUG.....//Libraries linked by default with all C++ applications..CMAKE_CXX_STANDARD_LIBRARIES:STRING=-kernel32 -user32 -lgdi32 -winspool -lsHELL32 -ole32 -oleaut32 -luuid -lcomctl32 -ladvapi32.....//No help, variable specified on the command line..CMAKE_CXX_COMPILER:FILEPATH=C:/mingw64/bin/g++.exe.....//Path to a program..CMAKE_DLLTOOL:FILEPATH=C:/mingw64/bin/dlltool.exe.....//Flags used by the linker during all build types..CMAKE_EXE_LINKER_FLAGS:STRING=.....//Flags used by the linker during DEBUG builds..CMAKE_EXE_LINKER_FLAGS_DEBUG:STRING=.....//Flags used by the linker during MINSIZEREL builds..CMAKE_EXE_LINKER_FLAGS_MINSIZEREL:STRING=.....//Flags used by the linker during RELEASE builds..CMAKE_EXE_LINKER_FLAGS_RELEASE:STRING=.....//Flags used by the linker during RELWITHDEBINFO builds..CMAKE_EXE_LINKER_FLAGS_RELWITHDEBINFO:STRING=.....//No help, variable specified on the command line..CMAKE_EXPORT_COMPILE_COMMANDS:BOOL=TRUE.....//Value Computed by CMake..CMAKE_FIND_PACKAGE_REDIRECTS_DIR:STATIC=C:/Users/angel/Desktop/validadorCadenas/build/CMakeFiles/pkgRedirects.....//Convert GNU import libraries to MS format (Requires Visual Studio)..CMAKE_GNUtoMS:BOOL=OFF.....//Install path prefix, prepended onto install directories..CMAKE_INSTALL_PREFIX:PATH=C:/Program Files (x86)/test.....//Path to a program..CMAKE_LINKER:FILEPATH=C:/mingw64/bin/ld.exe.....//Path to a program..CMAKE_MAKE_PROGRAM:FILEPATH=C:/mingw64/bin/make.exe.....//Object files created by the linker during the compilation
```

Decoded text

[illegible]

A simple vista se puede notar que con este simple algoritmo, ocultamos la información de una manera eficiente.

Conclusiones

Gildo López Miguel Ángel: Respaldo la información es crucial en la era digital en la que vivimos. Almacenar copias de seguridad de nuestros archivos y datos es una medida preventiva que puede ahorrarnos muchos problemas y dolores de cabeza en caso de pérdida de información debido a fallas técnicas, hackeos o errores humanos. Es importante tomar en serio la importancia de respaldar la información y hacerlo de manera regular para garantizar la protección y disponibilidad de nuestros datos.

González Ramírez Alan Leonardo: El respaldo y la recuperación son pilares fundamentales de la gestión de datos, garantizando la seguridad y la continuidad del sistema frente a posibles pérdidas o fallos. Una estrategia integral, respaldada por prácticas regulares, es esencial para mitigar riesgos y garantizar la integridad de la información crítica.

Montejano Paredes Diego Fernando: El sistema de recuperación de un sistema operativo no es una característica secundaria, sino fundamental que sustenta la fiabilidad, seguridad y resistencia del sistema. Su importancia va más allá de los usuarios individuales y se extiende a empresas, organizaciones, lo que lo pone de rol crítico que desempeña en el ecosistema digital actual.

Referencias

Whitman, M. E., & Mattord, H. J. (2022). *Principles of Information Security*. Cengage.

Erickson, J. (2019). *Hacking: The Art of Exploitation*. W. Ross MacDonald School Resource Services Library.

Villanueva, A. (2021, November 5). *Las 7 capas de seguridad digital - OSTEC: Segurança Digital De resultados*. OSTEC. <https://ostec.blog/es/seguridad/las-7-capas-de-seguridad-digital/>