Tarea semana 8: Manejo de archivosJorge Peñaloza

Programación Avanzada II

Instituto IACC

11 de octubre de 2021

**Desarrollo**

**Respuestas del problema**

***Contexto***

Lea atentamente el siguiente caso, analice la información de acuerdo a los contenidos revisados en la semana y conteste las preguntas que se presentan a continuación:

Recuerde que usted sigue trabajando para diseñar lógicamente un sistema para el registro de estudiantes de colegios privados de la Región Metropolitana de Santiago, recuerde que el sistema debe manejar las personas dentro del proceso como son: estudiantes con todos sus atributos (nombre, apellido, Rut, dirección) y apoderados también con sus atributos (nombre, apellido, Rut, dirección), que año y asignaturas cursan cada uno de los estudiantes, si realizan actividades extras, especificando cual actividad realizan.

Ahora bien, basándose en esa situación ponga en práctica sus conocimientos adquiridos y

explique lo siguiente:

1. Explique el uso de Directorios en el desarrollo del sistema (3 puntos).
2. Mencione los distintos tipos de archivos binarios, texto y CSV, utilizados en el desarrollo del sistema (3 puntos).
3. Utilice los archivos dados en cada caso planteado (3 puntos).

***Respuesta a pregunta 1***

Los directorios son elementos que emulan las carpetas físicas existentes permitiendo mantener la documentación ordenada dentro de los computadores.

Por ejemplo, en código de programación 1 se puede apreciar como listar los elementos del directorio desde donde se ejecuta el script, además en la figura 1 se puede ver el listado de elemento de la carpeta en curso.

Código de programación 1

*Archivo: directorios.py*

import os

direc = os.listdir(".")

for elemento in direc:

    print(elemento)

Fuente: Elaboración propia

Figura 1

*Vista de ejecución de código de programación 1.*

Texto

Descripción generada automáticamente

Fuente: elaboración propia

Por ejemplo, en código de programación 2 se puede apreciar como listar los elementos que del directorio además de identificar que archivos son con extensión PDF, además en la figura 2 se puede ver el listado de elementos de la carpeta.

Código de programación 2

*Archivo: archPDF.py*

import os

ejemplo\_dir = '/Users/ricar/Google Drive/IACC/2021/Programacion avanzada 2/Semana 8/'

contenido = os.listdir(ejemplo\_dir)

pdf = []

for fichero in contenido:

    fichero.endswith(".pdf")

    print(fichero, "es" if fichero.endswith(".pdf") else "no es", "PDF")

    if os.path.isfile(os.path.join(ejemplo\_dir, fichero)) and fichero.endswith('.pdf'):

        pdf.append(fichero)

print()

for fichero in pdf:

    print(fichero, "es un archivo con extension PDF")

Fuente: Elaboración propia

Figura 2

*Vista de ejecución de código de programación 2.*

Texto

Descripción generada automáticamente

Fuente: elaboración propia

***Respuesta a pregunta 2***

**Archivos binarios**

Siempre es necesario trabajar con los archivos según sus necesidades, por ejemplo, al leer un archivo binario es necesario utilizar el parámetro “b” para identificar que la lectura o escritura del archivo será byte a byte.

Por ejemplo, al leer el archivo miarchivo.txt se puede ver el contenido en la figura 3.

Figura 3

*Vista de archivo miarchivo.txt en Bloc de notas*

Interfaz de usuario gráfica, Aplicación, Tabla, Word

Descripción generada automáticamente

Fuente: elaboración propia

El con tenido de mi archivo se puede leer en formato binario utilizando código de programación 3 que lee en butes y para poder comprobar se muestran los datos con la función *chr(),* Se puede la ejecución de este ejemplo en la figura 4

Código de programación 3

*Archivo: leerbinario.py*

with open("miarchivo.txt","rb") as f:

    data = f.read()

for elemento in data:

    print(elemento, chr(elemento))

Fuente: Elaboración propia

Figura 4

*Vista de ejecución de código de programación 3.*

Texto

Descripción generada automáticamente

Fuente: elaboración propia

**Archivos de texto**

Los archivos de texto se guardan bytes, pero cuando se realiza la lectura de estos, se ejecuta de en forma interpretativa de forma que se puede ver su representación de texto en vez de su código ASCII.

Por ejemplo, al ejecutar código de programación 4 se crea un archivo que contiene el mensaje “hola mundo”, el cual se puede ver en la figura 5.

Código de programación 4

*Archivo: escribirtexto.py*

f = open ('holamundo.txt','w')

f.write('hola mundo')

f.close()

Fuente: Elaboración propia

Figura 5

*Vista de bloc de notas de archivo holamundo.txt*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Fuente: elaboración propia

**Archivos CSV**

Los archivos de texto que pueden venir con formatos específicos o estandarizados según sea la necesidad, los archivos CVS (Comma Separated Values) es un tipo de archivos que permite transportar información en forma de datos matriciales sonde las columnas van separadas por comas u otro carácter y las filas contienen la los datos completos de un registro con sus respectivos campos.

Por ejemplo, se conoce el archivo miarchivo.csv que contiene la información de la figura 6. Este archivo se lee con código de programación 5 que contempla una librería especializada en leer archivos CSV, cuando se ejecuta este código se pueden ver los datos en la figura 7.

Figura 6

*Vista de bloc de notas de archivo miarchivo.csv*

Texto

Descripción generada automáticamente

Fuente: elaboración propia

Código de programación 6

*Archivo:* *leercsv.py*

import csv

file = open('miarchivo.csv', "r")

reader = csv.reader(file, delimiter=';', quotechar=';', quoting=csv.QUOTE\_MINIMAL)

for row in reader:

    for columna in row:

        print(columna, end=" - ")

    print()

Fuente: Elaboración propia

Figura 7

*Vista de bloc de notas de archivo miarchivo.csv*

Texto

Descripción generada automáticamente

Fuente: elaboración propia

***Respuesta a pregunta 1 y 2 considerando datos de estudiantes, apoderados, asignaturas y actividades.***

Para ejemplificar un buen uso de archivos CSV, lo primero es definir los archivos y que datos contendrán, La estructura de archivos será la contenida en la figura 8, que contiene una relación similar a la de un diagrama UML para una base de datos.

Figura 8

*Estructura de archivos y datos*

Diagrama

Descripción generada automáticamente

Fuente: elaboración propia

En código de programación 7 se encuentra una clase que permite acceder a los datos de los archivos CSV de una forma más simple.

Código de programación 7

*Archivo:* *archivoCSV.py*

import csv

class archivoCSV:

    def \_\_init\_\_(self, archivo):

        self.archivo = archivo

    def setFile(self, archivo):

        self.archivo = archivo

    def getFile(self):

        return self.archivo

    def crearCabecera(self, \*arg):

        archivo = open(self.archivo,"w", newline="")

        escribir = csv.writer(archivo, delimiter=',', quoting=csv.QUOTE\_MINIMAL)

        escribir.writerow(arg)

    def insertar(self, \*arg):

        archivo = open(self.archivo,"a", newline="")

        escribir = csv.writer(archivo, delimiter=',',  quoting=csv.QUOTE\_MINIMAL)

        escribir.writerow(arg)

    def listar(self ):

        with open(self.archivo, newline='') as f:

            reader = csv.reader(f)

            for row in reader:

                print(row)

Fuente: Elaboración propia

En código de programación 8 se observa un script que crea los archivos con sus cabeceras, en código de programación 9 se muestra el script que permite insertar datos en los archivos, y en código de programación 10 se muestra un script que lee los datos de los archivos.

Código de programación 8

*Archivo:* *crearArchivos.py*

from archivoCSV import archivoCSV

estudiantes = archivoCSV("estudiantes.csv")

estudiantes.crearCabecera("nombre","apellido","rut","direccion","anio")

apoderados = archivoCSV("apoderados.csv")

apoderados.crearCabecera("nombre","apellido","rut","direccion","anio")

incripcionasignaturas = archivoCSV("incripcionasignaturas.csv")

incripcionasignaturas.crearCabecera("rut","codigo")

incripcionactividades = archivoCSV("incripcionactividades.csv")

incripcionactividades.crearCabecera("rut","codigo")

asignaturas = archivoCSV("asignaturas.csv")

asignaturas.crearCabecera("codigo","descripcion", "plan")

actividades = archivoCSV("actividades.csv")

actividades.crearCabecera("codigo","descripcion", "plan")

Fuente: Elaboración propia

Código de programación 9

*Archivo:* *insertardatos.py*

from archivoCSV import archivoCSV

estudiantes = archivoCSV("estudiantes.csv")

estudiantes.insertar("Jorge","Peñaloza","13834570-k","Las Camelias","2021")

estudiantes.insertar("Anais","Lopez","15321528-8","Las Magnolias","2022")

apoderados = archivoCSV("apoderados.csv")

apoderados.insertar("Segundo","Peñaloza","6525321-2","Las Camelias","2021")

asignaturas = archivoCSV("asignaturas.csv")

asignaturas.insertar("1","Ingles", "Verbos")

asignaturas.insertar("2","Matematica", "Integrales")

actividades = archivoCSV("actividades.csv")

actividades.insertar("1","Natacion", "5km por sesion")

incripcionasignaturas = archivoCSV("incripcionasignaturas.csv")

incripcionasignaturas.insertar("13834570-k","1")

incripcionasignaturas.insertar("13834570-k","2")

incripcionasignaturas.insertar("15321528-8","2")

incripcionactividades = archivoCSV("incripcionactividades.csv")

incripcionactividades.insertar("13834570-k","1")

Fuente: Elaboración propia

Código de programación 10

*Archivo:* *listar.py*

from archivoCSV import archivoCSV

estudiantes = archivoCSV("estudiantes.csv")

apoderados = archivoCSV("apoderados.csv")

asignaturas = archivoCSV("asignaturas.csv")

actividades = archivoCSV("actividades.csv")

incripcionasignaturas = archivoCSV("incripcionasignaturas.csv")

incripcionactividades = archivoCSV("incripcionactividades.csv")

estudiantes.listar()

apoderados.listar()

asignaturas.listar()

actividades.listar()

incripcionasignaturas.listar()

incripcionactividades.listar()

Fuente: Elaboración propia

En la figura 11 se puede ver cómo actúan cada uno de estos scripts cuando se ejecutan, una vez concluida la ejecución de todos los scripts se puede ver la figura 10 en donde se ven los archivos creados.

Figura 10

*Lista de archivos CSV creados*

Texto

Descripción generada automáticamente

Fuente: Elaboración propia

Figura 11

*Vista de script ejecutados por consola*

Texto

Descripción generada automáticamente

Fuente: elaboración propia

Ya confirmada la ejecución de cada script, se puede ver en la figura 12 como se ven todos los archivos CSV.

Figura 12

*Archivos CSV en bloc de notas*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Fuente: Elaboración propia

Bibliografía

IACC. (2020). Manejo de archivos. En *Programación avanzada 2. Semana 8.*

IACC. (2021). Tarea semana 8.