



Instituto Tecnológico de Costa Rica.

Campus Tecnológico Central de Cartago.

Escuela de Ingeniería en Computación.

Taller de Programación.

Prof. Ing. William Mata Rodríguez.

Programa 1

Eventos de Atletismo.

Ledvin Manuel Leiva Mata – 2023071280.

Fecha de entrega: 30 de abril del 2023, semana 11.

1 semestre, 2023.

Contenido

Enunciado del proyecto	3
Temas investigados	4
• Modulo Os	4
• Modulo Sys.....	4
• Modulo PyCountry	5
• Modulo datetime	5
• Modulo dns Resolver.....	6
• Modulo smtpplib	6
• Modulo ReportLab	7
• Modulo Email.....	7
• Os.system('CLS').....	8
• validate_email()	8
• Función sorted y lambda	9
• Herramientas de depuración (debugger) en programación.	9
• Explicación de las funciones del “debugger” del IDE que usted está usando. 10	
• La importancia de la documentación de software.....	11
• Librerías en Python.....	11
Conclusiones del trabajo	12
Errores durante el proyecto	12
Aprendizajes obtenidos	12
LISTA DE REVISIÓN DEL PROYECTO	14
Referencias	15

Enunciado del proyecto

Desarrollar una aplicación para el análisis de marcas que hacen los atletas en determinadas disciplinas de atletismo

Algunas de estas disciplinas son:

- Las carreras de velocidad, donde hay pruebas como las de 100 y 200 metros planos

- Los lanzamientos donde hay pruebas como el disco, el martillo y la jabalina.

A su vez estas pruebas se pueden dividir en categorías por edades, por ejemplo, la

categoría U18 se refiere a atletas que tienen 18 años o menos.

Por último, las pruebas pueden clasificarse por el sexo: femenino o masculino.

Ejemplos de estas pruebas son:

- Carreras de velocidad - 200 m - Categoría U20 – Femenino

- Carreras de saltos – 100 m vallas – Categoría Mayor – Femenino

- Carreras de velocidad - 100 m - Categoría Mayor – Masculino

Los atletas pueden participar en diferentes pruebas en los eventos deportivos que se

organicen. Cada evento tiene las pruebas que va a realizar y los atletas deciden las

pruebas en que van a participar. Las marcas que van haciendo los atletas se van registrando para luego obtener información general y detallada por atleta

Temas investigados

- **Modulo Os**

El módulo Os de Python es una biblioteca integrada que proporciona un conjunto de funciones para interactuar con el sistema operativo subyacente. Este módulo permite a los programadores interactuar con los recursos del sistema operativo como archivos, directorios, procesos, variables de entorno, etc. Además, el módulo Os proporciona una capa de abstracción para la portabilidad de código, lo que significa que los programas escritos en Python que utilizan el módulo Os se pueden ejecutar en diferentes sistemas operativos sin necesidad de modificar el código. En resumen, el módulo Os es una herramienta esencial para la manipulación y gestión de recursos del sistema operativo en Python.

Para usar el módulo Os en Python, primero debemos importarlo en nuestro programa. Después de la importación, podemos utilizar sus funciones para realizar operaciones en el sistema operativo. Algunas de las funciones más comunes en el módulo Os son: `os.getcwd()` para obtener el directorio de trabajo actual, `os.listdir()` para obtener una lista de archivos y directorios en un directorio, `os.mkdir()` para crear un nuevo directorio, `os.remove()` para eliminar un archivo, `os.rename()` para renombrar un archivo y `os.system()` para ejecutar un comando del sistema operativo. Es importante tener en cuenta que las funciones del módulo Os pueden variar según el sistema operativo, por lo que es recomendable consultar la documentación oficial para obtener más información sobre las funciones específicas disponibles en cada plataforma.

- **Modulo Sys**

El módulo sys es una biblioteca de Python que proporciona variables y funciones que interactúan con el intérprete de Python. Este módulo proporciona acceso a algunas variables de sistema utilizadas o mantenidas por el intérprete de Python, como la ruta de búsqueda del módulo. También proporciona una forma de interactuar con el intérprete de Python durante el tiempo de ejecución, a través de los argumentos de línea de comandos y las variables de entorno.

Entre las funciones más útiles del módulo sys se encuentra la función `sys.argv`, que proporciona una lista de argumentos de línea de comandos pasados al script de Python. Esta función se utiliza ampliamente en scripts de Python para hacer que el script sea más dinámico y flexible. Otra función

útil es `sys.exit()`, que detiene la ejecución del programa y devuelve un código de salida al sistema operativo.

- **Modulo PyCountry**

El módulo `pycountry` es una biblioteca de Python que proporciona una interfaz para acceder a los datos ISO de países, territorios y monedas. Con este módulo, se puede obtener información sobre los códigos ISO de países, incluyendo nombres, códigos de dos letras y tres letras, códigos numéricos y más.

Para utilizar el módulo `pycountry`, primero se debe instalar a través del administrador de paquetes de Python, como `pip`. Luego, se puede importar el módulo en el archivo de Python para acceder a sus funciones y clases. Con `pycountry`, se pueden realizar operaciones como buscar un país a partir de su código ISO, obtener la lista completa de países y territorios, y acceder a información adicional, como la moneda utilizada en un país determinado.

- **Modulo datetime**

El módulo `datetime` es una biblioteca de Python que permite trabajar con fechas y horas de forma sencilla y eficiente. Este módulo proporciona varias clases y funciones para trabajar con fechas y horas, como la clase `datetime`, que permite representar una fecha y hora específicas, y la clase `timedelta`, que permite realizar cálculos con fechas y horas.

El módulo `datetime` se utiliza comúnmente en aplicaciones que manejan fechas y horas, como en el desarrollo de sistemas de reservas, en el procesamiento de datos financieros y en el análisis de datos de series de tiempo. También es útil en aplicaciones que requieren el registro de eventos y la gestión del tiempo, como en el registro de actividad en servidores o en el control de acceso de usuarios.

Entre las funciones más comunes del módulo `datetime` se encuentran la obtención de la fecha y hora actual del sistema, la conversión de una cadena de texto en una fecha y hora, la comparación de fechas y horas, y el cálculo de la diferencia entre dos fechas o horas. En resumen, el módulo `datetime` es una herramienta fundamental para el trabajo con fechas y horas en Python y se utiliza en una amplia variedad de aplicaciones.

- **Modulo dns Resolver**

El módulo `dns.resolver` es una herramienta de Python que permite realizar consultas DNS (Domain Name System) utilizando diferentes tipos de registros. Este módulo proporciona una interfaz fácil de usar para acceder a los servidores DNS y obtener información sobre los nombres de dominio y sus registros.

Para utilizar el módulo `dns.resolver`, es necesario importarlo en nuestro código. A continuación, se pueden crear objetos `Resolver` para establecer la configuración de resolución de DNS, y realizar consultas a través de la función `query()`. También se pueden especificar diferentes tipos de registros DNS, como A, MX, CNAME, NS, entre otros.

En el código se utiliza para comprobar si un correo es válido o no.

- **Modulo smtplib**

El módulo `smtplib` es una biblioteca en Python que proporciona una interfaz para enviar correos electrónicos utilizando el protocolo Simple Mail Transfer Protocol (SMTP). Permite establecer una conexión a un servidor SMTP, autenticarse con credenciales (si es necesario), y enviar correos electrónicos a través de esa conexión.

Para utilizar el módulo `smtplib`, es necesario importarlo en el script de Python, y luego crear una instancia de la clase `SMTP`. A continuación, se establece la conexión con el servidor SMTP utilizando el método `connect()`, y se autentica (si es necesario) utilizando el método `login()`. Una vez conectado y autenticado, se puede enviar un correo electrónico utilizando el método `sendmail()`, que toma como argumentos la dirección de correo electrónico del remitente, la dirección de correo electrónico del destinatario, y el mensaje a enviar. Después de enviar el correo electrónico, se cierra la conexión utilizando el método `quit()`.

- **Modulo ReportLab**

El módulo ReportLab es una biblioteca de Python que permite generar documentos PDF de manera dinámica. ReportLab proporciona un conjunto de herramientas para la creación de documentos PDF personalizados, lo que permite al usuario crear documentos de alta calidad de forma programática. Los documentos generados pueden contener texto, gráficos, tablas, imágenes y otros elementos.

El módulo se utiliza en aplicaciones empresariales para generar informes, facturas, certificados, etiquetas y otros tipos de documentos. El uso del módulo ReportLab permite a los desarrolladores crear informes personalizados y de alta calidad en menos tiempo y con menos esfuerzo que si tuvieran que hacerlo manualmente. La biblioteca también tiene características avanzadas para el control de la paginación y el diseño del documento, lo que permite una mayor flexibilidad en la creación de documentos.

- **Modulo Email**

El módulo email es una biblioteca de Python para la creación, manipulación y envío de mensajes de correo electrónico. Proporciona una interfaz para construir mensajes de correo electrónico con encabezados, destinatarios, cuerpo y adjuntos. Además, permite el análisis y la manipulación de mensajes de correo electrónico recibidos. Esta biblioteca es útil para automatizar el envío de correos electrónicos desde una aplicación, ya sea para notificar a los usuarios o para informar sobre eventos importantes. También puede ser utilizado para analizar y extraer información de los mensajes recibidos, como, por ejemplo, para realizar análisis de texto o minería de datos.

- **Os.system('CLS')**

El método `os.system('cls')` es una función que pertenece al módulo `os` de Python y se utiliza para limpiar la pantalla de la consola en Windows. El argumento `'cls'` es un comando de Windows que se utiliza para borrar la pantalla de la consola.

Cuando se ejecuta el comando `os.system('cls')`, la pantalla de la consola se borra, lo que hace que la consola quede en blanco y se elimine todo lo que se había escrito anteriormente. Este método es útil para mejorar la presentación del texto en la consola, ya que permite que el texto se muestre de manera más clara y legible.

Es importante destacar que este método solo funciona en sistemas operativos Windows, en sistemas operativos basados en Unix, como Linux o macOS, se utiliza el comando `'clear'` en lugar de `'cls'`. Por lo tanto, es importante tener en cuenta la plataforma en la que se está ejecutando el código para asegurarse de utilizar el comando correcto.

- **validate_email()**

El método `validate_email()` es una función de la biblioteca `validate_email` en Python, que se utiliza para validar direcciones de correo electrónico. Este método toma una cadena de texto como entrada que representa una dirección de correo electrónico y devuelve un valor booleano que indica si la dirección es válida o no.

La función `validate_email()` utiliza una variedad de técnicas para validar la dirección de correo electrónico, como verificar la sintaxis correcta de la dirección y comprobar si el dominio asociado a la dirección existe y tiene registros MX (Mail Exchange) configurados correctamente. También puede comprobar si la dirección de correo electrónico está en una lista de correos no deseados o si tiene una mala reputación.

- **Función sorted y lambda**

La función sorted de Python para ordenar una lista de elementos. En particular, el argumento key que se le pasa a la función sorted se utiliza para especificar la clave de ordenamiento, es decir, la función que se va a utilizar para determinar el orden en el que se deben ordenar los elementos.

En este caso, la expresión lambda x: x[0] se utiliza como función de clave de ordenamiento. Esta expresión es una función anónima que recibe como argumento x y devuelve el primer elemento de x (es decir, x[0]). Por lo tanto, cuando se llama a sorted con key=lambda x: x[0], la función de ordenamiento utilizará el primer elemento de cada elemento de la lista para determinar el orden de los elementos.

En resumen, el argumento key de la función sorted se utiliza para especificar la función que se va a utilizar para determinar el orden de los elementos, y la expresión lambda x: x[0] que se utiliza en este caso se encarga de ordenar los elementos de la lista por el valor del primer elemento de cada uno.

- **Herramientas de depuración (debugger) en programación.**

¿Qué es?

Las herramientas de depuración (debugger) son programas informáticos que permiten a los programadores encontrar y corregir errores en el código fuente de un programa. Estas herramientas funcionan como un asistente para el programador, permitiéndoles examinar el estado interno del programa en diferentes puntos de su ejecución, y así identificar errores lógicos o de sintaxis que puedan estar causando problemas.

¿Cómo se usa?

Las herramientas de depuración se utilizan en el proceso de desarrollo de software para asegurar que el programa funciona correctamente. El programador puede establecer puntos de interrupción en el código, lo que permite que el programa se detenga en un momento determinado y le permita examinar el estado interno del programa en ese momento. También puede examinar variables y expresiones en tiempo real, y ejecutar el código paso a paso para ver cómo se está ejecutando el programa en general.

- **Explicación de las funciones del “debugger” del IDE que usted está usando.**

Debugger de Visual Studio para Python

El "debugger de Visual Code para Python" es una herramienta que permite depurar código de Python en el entorno de desarrollo integrado (IDE) Visual Studio Code. Esta herramienta cuenta con varias funciones que ayudan al usuario a detectar y corregir errores en su código, las cuales se describen a continuación:

- **Breakpoints:** Permite al usuario detener la ejecución del programa en un punto específico para analizar el estado de las variables y el flujo del programa en ese momento.
- **Step over/into/out:** Estas funciones permiten al usuario avanzar la ejecución del programa línea por línea. "Step over" ejecuta la línea actual y pasa a la siguiente, "step into" entra en la función que se llama en la línea actual y "step out" sale de la función actual y regresa a la línea anterior.
- **Watch:** Esta función permite al usuario monitorear el valor de una variable en tiempo real mientras el programa se está ejecutando.
- **Call Stack:** Muestra una lista de las funciones que se han llamado y que se están ejecutando actualmente, lo que puede ayudar a entender cómo el programa llegó a su estado actual.
- **Debug Console:** Permite al usuario ingresar comandos de Python en tiempo de ejecución para evaluar expresiones y modificar el valor de las variables.

Para usar el debugger de Visual Code para Python, el usuario debe primero agregar breakpoints en el código en los puntos que desea inspeccionar. Luego, debe ejecutar el programa en modo de depuración, lo que le permitirá detener la ejecución en los breakpoints agregados. A partir de ahí, puede utilizar las funciones de depuración descritas anteriormente para analizar el estado del programa y corregir cualquier error.

- **La importancia de la documentación de software**

La documentación de software es esencial para cualquier proyecto de desarrollo de software. Proporciona información detallada sobre el software, desde su diseño hasta su implementación y mantenimiento, lo que ayuda a los desarrolladores a entender mejor el código, su funcionalidad y cómo utilizarlo de manera efectiva.

La documentación del software tiene varios beneficios. En primer lugar, ayuda a los desarrolladores a entender el diseño del software, sus características y funcionalidades, lo que les permite trabajar más eficientemente y reducir los errores. Además, permite a los desarrolladores compartir su conocimiento y experiencia con otros miembros del equipo, lo que puede ayudar a reducir la dependencia de una sola persona o equipo.

La documentación del software también es importante para los usuarios finales. Proporciona una guía detallada sobre cómo utilizar el software, lo que reduce la curva de aprendizaje y ayuda a los usuarios a utilizar el software de manera efectiva. Además, la documentación del software también puede incluir información sobre la solución de problemas, lo que ayuda a los usuarios a resolver cualquier problema que puedan encontrar.

- **Librerías en Python**

En Python, una biblioteca o librería se refiere a un conjunto de módulos que contienen una serie de funciones y métodos predefinidos que pueden ser importados y utilizados en un programa. Las bibliotecas de Python pueden ser incluidas en un programa para ampliar su funcionalidad y permitir la reutilización de código. Algunas bibliotecas populares de Python incluyen NumPy para computación científica, Pandas para manipulación de datos, Matplotlib para visualización de datos y Django para desarrollo web. Para utilizar una biblioteca en Python, se debe importar en el código mediante la declaración "import" seguida del nombre de la biblioteca.

Conclusiones del trabajo

Errores durante el proyecto

El único problema presentado fue a la hora de trabajar con la validación de correo electrónico, ya que este se puede validar su estructura para todos los dominios, pero no es posible validar para todos los dominios si este existe en realidad.

Por ejemplo, Outlook desde hace un tiempo implemento una restricción a su EULA para evaluar dicho estado, por lo tanto no se puede verificar si el correo existe, por lo tanto se tuvo que implementar en el código una excepción con los dominios pertenecientes a **Outlook, es decir Outlook.com, Hotmail.com, yahoo.com, yahoo.es, Hotmail.es y Outlook.es**, como se puede ver en la siguiente captura de pantalla:

```
def validar_correo_electronico(correo_electronico):  
    # validar que el correo sea unico  
  
    for correo in atletas:  
        if correo_electronico == correo[7]:  
            return False  
  
    domain = correo_electronico.split('@')[1] # Obtener el dominio del correo electronico  
  
    if domain == 'hotmail.com' or domain == 'yahoo.com' or domain == 'outlook.com' or domain == 'outlook.es' or domain == 'yahoo.es' or domain == 'hotmail.es':  
        return True # si el dominio pertenece a outlook devolver true automaticamente, ya que el EULA no permite comprobar su existencia.
```

Mas allá de esto no se presentó ningún problema al cual no se le pudo dar una solución sencilla ya sea de sintaxis o lógica.

Aprendizajes obtenidos

En lo personal no obtuve muchos aprendizajes obtenidos más allá del desconocimiento del EULA de Outlook y el cómo trabajar un archivo PDF sin pasarlo a HTML primero y trabajarlo con CSS, es decir trabajarlo directamente desde Python o cualquier lenguaje.

Actividad realizada	Horas
Análisis del problema	1
Diseño de algoritmos	1
Investigación de...	2
Programacion	8
Documentacion interna	1
Pruebas	1
Elaboración del manual de usuario	1
Elaboración de documentación del proyecto	1
Etc.	
TOTAL	16

LISTA DE REVISIÓN DEL PROYECTO

Concepto	Puntos originales	Avance 100/%/0	Puntos obtenidos	Análisis de resultados
Registrar disciplinas <ul style="list-style-type: none"> - Agregar - Consultar - Modificar - Eliminar 	0 0 2 2	100%		Funcionalidad dada por el profesor Funcionalidad dada por el profesor
Registrar pruebas por disciplina <ul style="list-style-type: none"> - Agregar - Consultar - Modificar - Eliminar 	3 1 3 3	100%		
Registrar atletas <ul style="list-style-type: none"> - Agregar - Consultar - Modificar - Eliminar 	3 1 3 4	100%		
Registrar eventos <ul style="list-style-type: none"> - Agregar - Consultar - Modificar - Eliminar 	3 1 3 3	100%		
Registrar marcas <ul style="list-style-type: none"> - Agregar - Consultar - Modificar - Eliminar 	4 1 4 5	100%		
Marcas por evento <ul style="list-style-type: none"> - Datos del reporte - Orden ascendente/descendente 	4 2	100%		
Marcas por atleta <ul style="list-style-type: none"> - Datos del reporte - Orden ascendente/descendente - Correos electrónicos 	4 2 10	100%		
Mejores marcas por pruebas <ul style="list-style-type: none"> - Datos del reporte - Orden ascendente/descendente 	4 2	100%		
Uso de archivos PDF para reportes	8	100%		
Validaciones de datos	10	100%		
Ayuda (manual de usuario desplegado en el programa)	5	100%		
TOTAL	100	100%		
Partes desarrolladas adicionalmente		100%		

Referencias

La documentación oficial de Python: <https://docs.python.org/>

Python Software Foundation. (2021). time — Time access and conversions.

Python 3.11.3 documentation. <https://docs.python.org/3/library/time.html>

Python Software Foundation. (2021). subprocess — Subprocess management.
Python

Python Software Foundation. (2021). os — Miscellaneous operating system
interfaces. 3.11.3 documentation. <https://docs.python.org/3/library/subprocess.html>

Python 3.11.3 documentation. <https://docs.python.org/3/library/os.html>

Python Software Foundation. (2021). str.title(). Python 3.11.3 documentation.
<https://docs.python.org/3/library/stdtypes.html#str.title>

El tutorial de Python de W3Schools: <https://www.w3schools.com/python/>

La documentación de las librerías y módulos de Python, por ejemplo, la
documentación de la librería datetime:
<https://docs.python.org/3/library/datetime.html>

Artículos y tutoriales en línea de programación en Python, por ejemplo, los
tutoriales de Real Python: <https://realpython.com/>

Páginas de preguntas y respuestas como Stack Overflow:
<https://stackoverflow.com/>

Blogs y sitios web de programación en Python, como el blog de PyPI (Python
Package Index): <https://blog.pypi.org/>

Microsoft. (2021). Debugging in Visual Studio Code.
<https://code.visualstudio.com/docs/editor/debugging>