

Introducción a la Programación

I Semestre – 2022

Integrantes:

Samuel Garcés Castillo – 2022129139

Carlos Daniel Guzmán Ramírez – 2022437782

Tarea Programada #1

Profesora:

Laura Guiselle Coto Sarmiento

Tutores:

Daniel Araya Sambucci

Fecha de entrega:

Jueves 28 de abril

Estatus de la entrega:

Samuel Garcés – 2022129139
Carlos Guzmán – 2022437782

Índice

Enunciado del proyecto	3
Justificación de eliminación de olores	3
Conclusiones del trabajo	5
Reglamento de trabajo	6
Agendas y Minutas	7
Cronograma	9
Estadísticas de tiempos	10

Samuel Garcés – 2022129139

Carlos Guzmán – 2022437782

Enunciado del proyecto:

Este trabajo fue realizado bajo las especificaciones indicadas en el documento “TP1_Criptografía_2022.pdf”, el cual fue entregado y consultado a través de la plataforma del TEC Digital, en el apartado de “Documentos públicos de Taller de Programación GR 2”. Puede consultarse dicha información a través del siguiente enlace:

https://tecdigital.tec.ac.cr/dotlrn/classes/CA/IC1803/S-1-2022.CA.IC1803.2/file-storage/view/public%2Ftareas-programadas%2FTP1_Criptograf%C3%ADa_2022.pdf

Eliminación de olores:

- Rigidez:** El código está hecho de manera simple, siguiendo una serie de patrones que han sido cuidadosamente documentados, permitiendo así una fácil manipulación y actualización del código.

```

103 | ### Definición de Funciones ###
104 |
105 | # 1 - Cifrado César
106 | > def procesarCodCesar(pfrase): # Proceso de Codificación...
123 | > def procesarDecodCesar(pfrase): # Proceso de Decodificación...
140 | > def obtenerCodCesar(accion): ...
156 |
157 | # 2- Cifrado por llave
158 | > def procesarCodLlave(pfrase, pclave): # Proceso de Codificación...
185 | > def procesarDecodLlave(pfrase, pclave): # Proceso de Decodificación...
209 | > def obtenerCodLlave(accion): ...
228 |
229 | # 3 - Sustitución Vigenere
230 | > def procesarCodVigenere(pfrase, pcifra): ...
254 | > def procesarDecodVigenere(pfrase, pcifra): ...
278 | > def obtenerCodVigenere(accion): ...
298 |

```

- Fragilidad:** Fueron creadas una serie de validaciones recursivas para cada tipo de entradas de datos que fueran necesarias a lo largo del código. Además, se implementó un try-catch fuera de las llamadas respectivas a cada función, permitiendo así el amortiguamiento de posibles errores.

```

11 | # Validación general
12 | > def validarString(string): # Valida que solo se ingrese una palabra, con formato de valores alfabéticos ...
26 | > def validarFrase(pValidar): # A diferencia de la anterior, esta validación admite espacios en blanco, permitiendo formar frases largas...
40 | > def validarOpcion(opcion, tope): # Esta función permite validar las opciones en un menú de selección numérica
41 | """
42 | Funcionamiento: Validar las entradas del menú
43 | Entradas: opcion (str) dato a trabajar, tope (int) valor mayor de las opciones seleccionables
44 | Salidas: (Booleano) realimentar al usuario con la corrección de posibles errores o permitir el avance del proceso.
45 | """
46 | if re.match("^[a-zA-Z]+$", opcion) and int(opcion)>= 0 and int(opcion)<= tope: # Valida que el dato sea de tipo entero, dentro del rango establecido
47 |     return True
48 | elif re.match("^[0-9]+$", opcion):
49 |     print("Debe ingresar un valor numérico")
50 |     return False
51 | else:
52 |     print("Debe ingresar una opción válida")
53 |     return False
54 | > def validarNumero(pnum, numeroMinimo): # Permite validar unicamente valores numéricos mayores a un número definido, y con máximo infinito...
78 | > def validarNumero2Digitos(pnum, numeroMinimo, numeroMaximo): # Permite validar valores numéricos mayores a un número definido, y con máximo definido...
102 |
636 | try:
637 |     opcion = input("Seleccione una opción: ")
638 |     if validarOpcion(opcion, 8)==False: # Validación de las opciones
639 |         return menu()
640 |     elif opcion == "0": # Finaliza el proceso
641 |         return "\n---Ejecución finalizada---"
642 | > else: # Selecciona opciones del menú...
709 | except:
710 |     print("\n>>> Ocurrió un error, vuelva a intentarlo") # Atrapa y corrige en caso de recibir un error desconocido
711 |     return menu()
712 |

```

Samuel Garcés – 2022129139

Carlos Guzmán – 2022437782

- c. **Inmovilidad:** Las funciones fueron hechas para usarse varias veces en todo el código de manera que sea fácil de reutilizar a lo largo de este.

```
def elegirAccion(ejercicio):
    """
    ...
    """
    accion = input(f"\n¿Que acción desea realizar? - {ejercicio} \n1- Codificar    2- Decodificar?    0- Regresar al menú\n>>> ")
    if validarOpcion(accion, 2):
        if accion == "0":
            return 0
        elif accion == "1":
            return 1
        else:
            return 2
    else:
        return elegirAccion(ejercicio)
```

- d. **Viscosidad:** El programa permite realizar todo tipo de cambio sin afectar el funcionamiento del código en general.
- e. **Complejidad innecesaria:** Se buscaron formas de hacer las funciones de la manera más simple posible dentro de los temas vistos en clase.

```
# 6 - Mensaje inverso
def procesarCodMInverso(pfrase, accion): # Proceso de Codificación
    """
    ...
    """
    return f"Mensaje {accion}: {pfrase[::-1]}"
```

- f. **Repetición innecesaria:** Ninguna función fue ‘copiada’ y ‘pegada’, no se repiten pedazos de códigos innecesarios. En caso de necesitarse una función en repetidas ocasiones, se define la función de manera que pueda ser utilizada en cualquier momento.
- g. **Opacidad:** El código está escrito de manera que sea muy fácil de leer e interpretar, ya sea en el nombre de las variables o en la forma en que se realizan las funciones.

```
def procesarCodCesar(pfrase): # Proceso de Codificación
    """
    Funcionamiento: Codificar una frase con el método de Cifrado César
    Entradas: pfrase (string) frase a codificar
    Salidas: Resultado del proceso
    """
    alfabeto = "abcdefghijklmnopqrstuvwxyz" # String con el Alfabeto a trabajar
    fraseCodificada, letraFrase = "", 0
    while letraFrase < len(pfrase):
        if pfrase[letraFrase] == " ":
            fraseCodificada += " "
            letraFrase += 1
        if (alfabeto.find(pfrase[letraFrase]) != -1): # Si se encuentra el valor a trabajar dentro del string,
            fraseCodificada += alfabeto[alfabeto.find(pfrase[letraFrase])+3] # se procede a concatenar su valor respectivo
            letraFrase += 1
    return "Mensaje codificado: "+fraseCodificada.upper()
```

Conclusiones del trabajo

Problemas encontrados y soluciones a los mismos

A lo largo de la realización del trabajo, diversos desafíos fueron enfrentados para alcanzar el éxito. Entre ellos se pueden destacar los siguientes:

n°	Día que descubre el problema	Descripción del problema	Día de solución	Descripción de la solución.	Tiempo transcurrido en horas
1	Lunes 18 de abril	En la función del dífrado por llave, es necesario el avance y comparación entre los datos de la frase a trabajar, junto con su respectiva llave paralelamente, y el ciclo for presenta problemas para trabajar dicha lógica	Lunes 18 de abril	Se trabajó alternativamente con un ciclo while, aplicando la lógica de "ciclos dentro de ciclos", aumentando y disminuyendo los valores respectivos según se requiera	1,5 horas
2	jueves 21 de abril	La función del XOR presenta errores graves durante el ingreso de la frase a decodificar	Domingo 24 de abril	Se discutió el problema con el asistente de la profesora (Daniel), y se concluyó en que se ajustaría el programa para permitir la decodificación hasta una vez finalizada la codificación	3 horas (a lo largo de la semana)
3	Lunes 25 de abril	La codificación y decodificación vigenera tiene una vulnerabilidad, al permitir realizar el proceso independientemente de la cifra numérica indicada	lunes 25 de abril	Se corrigió la labor del parametro pcifra dentro del procesamiento de ambas funciones	1 hora

Aprendizajes obtenidos

N°	Carácter personal	Carácter técnico
1	Trabajo en equipo	Uso de la lógica de módulos para la optimización del código
2	Distribución de las labores	Codificación de if-else en una sola línea
3	Búsqueda de soluciones diversas - autodidacta	Codificación de ciclos "for" en una sola línea
4	Solución de problemas	Validación recursiva de los datos de entrada
5	Optimización del tiempo	Utilización del formato f"..." para concatenar strings

Samuel Garcés – 2022129139

Carlos Guzmán – 2022437782

6	Autocritica	Creación de menús complejos
7	Reconocimiento de errores	Utilización de funciones estratégicas de listas
8	Análisis crítico de código ajeno	Utilización estratégica de strings combinados con listas
9	Simulación de un ambiente laboral real	Aplicación práctica de expresiones regulares

Reglamento de trabajo

1. La plataforma GitHub será el medio oficial para la manipulación de los archivos y el código de software.
2. Cada integrante debe trabajar un mínimo de 4 horas por semana en sus labores correspondientes.
3. Debe cumplirse con una técnica de trabajo limpia y ordenada en todo momento.
4. Debe seguirse una lógica de programación libre de olores de software.
5. Debe trabajarse única y exclusivamente en la rama personal de cada integrante, y se debe informar al coordinador general cuando se solicite el agregado a la rama principal.
6. En caso de ser necesario, debe informarse a los demás integrantes del equipo acerca de cualquier cambio que se realice en la rama principal.
7. Cualquier comentario, sugerencia, crítica constructiva o demás aporte que se brinde a cualquiera de los integrantes deberá realizarse de manera respetuosa, y de la misma manera se debe interpretar de manear proactiva para el beneficio de ambas partes.
8. No se tolerarán discusiones acaloradas o faltas de respeto ante ninguno de los integrantes de este equipo. Cualquier falta a esta regla podrá ser motivo de amonestación.

Tarea Programada #1

7

Samuel Garcés – 2022129139

Carlos Guzmán – 2022437782

Agendas y minutas

Agenda 2022-S1-01

Fecha: 4/4/2022		Hora de Inicio: 7:30 pm		Hora Final: 8:30 pm	
Lugar de reunión: Discord					
Invitados a la reunión: Carlos Guzmán y Samuel Garcés					
Tiempo	Tema	Responsable	Requiere llevar	Logro Esperado	
7:30 – 7:50	Definición de roles y creación del repositorio en GitHub	Carlos Guzmán	Nada	Determinar rol	
7:50 – 8:10	Delegación de responsabilidades y estandarización de código	Samuel Garcés	Nada	Aligeramiento del trabajo	
8:10 – 8:30	Asuntos varios	Todos	Nada	Abarcar cualquier asunto no contemplado en la agenda prevista	

Los tiempos no deberían sumar máximo 1 hora.

Minuta 2022-S1-01

Fecha: 4/4/2022		Hora de Inicio: 7:30 pm		Hora Final: 8:30 pm
Lugar de reunión: Discord				
Asistentes a la reunión: <ul style="list-style-type: none">- Carlos Guzmán- Samuel Garcés		Hora de llegada: 7:30 pm		
Ausentes en la reunión:		Justificación:		
Tema	Fecha límite de cumplimiento	Acuerdos	Responsable	
Definición de roles	4/4/2022	Se acuerda que Carlos Guzmán será el Coordinador general y encargado de desarrollo. Samuel Garcés será el Encargado de calidad	Todos	
Creación del repositorio en GitHub	4/4/2022	Carlos Guzmán se encargó de crear el repositorio de GitHub, y lo compartió con Samuel Garcés	Carlos Guzmán	
Delegación de responsabilidades	4/4/2022	Se acuerda que Samuel Garcés se encargará de realizar el código referente a los ejercicios impares (1,3,5,7) y Carlos Guzmán se encargará de los pares (2,4,6,8)	Todos	
Estandarización de código	8/4/2022	En los posteriores días, Carlos Guzmán se encargará de crear los estándares de código, y compartirlos con los compañeros del grupo	Carlos Guzmán	
Asuntos varios	4/4/2022	Se acordó seguir una serie de reglas para mantener la calidad del proyecto a realizar. Además, se coordinó que al terminar un ejercicio, se deberá compartir con el compañero de grupo para proceder con su revisión	Todos	

Tarea Programada #1

8

Samuel Garcés – 2022129139

Carlos Guzmán – 2022437782

Agenda 2022-S1-02

Fecha: 26/4/2022		Hora de Inicio: 7:30 pm		Hora Final: 8:30 pm
Lugar de reunión: Discord				
Invitados a la reunión: Carlos Guzmán y Samuel Garcés				
Tiempo	Tema	Responsable	Requiere llevar	Logro Esperado
7:30 – 7:50	Revisión del código y análisis de posibles errores	Carlos Guzmán	Archivo .py descargado	Asegurar la calidad del código elaborado
7:50 – 8:10	Revisión prevista de la documentación	Samuel Garcés	Documentación respectiva al proyecto	Confirmar el avance adecuado con respecto a la documentación, y coordinar una fecha estimada para la finalización
8:10 – 8:30	Retroalimentación positiva con respecto al trabajo a lo largo de la presente Tarea programada	Todos	Nada	Compartir comentarios y sugerencias acerca del trabajo realizado, y contribuir en la madurez grupal del equipo de trabajo

Los tiempos no deberían sumar máximo 1 hora.

Minuta 2022-S1-02

Fecha: 26/4/2022		Hora de Inicio: 7:30 pm		Hora Final: 8:30 pm
Lugar de reunión: Discord				
Asistentes a la reunión: <ul style="list-style-type: none">- Carlos Guzmán- Samuel Garcés		Hora de llegada: 7:30 pm		
Ausentes en la reunión:		Justificación:		
Tema	Fecha límite de cumplimiento	Acuerdos	Responsable	
Revisión del código y análisis de posibles errores	26/4/2022	Se realizan todas las pruebas correspondientes a los escenarios previstos, se hacen ajustes menores en el formato del código, y se confirma la calidad del trabajo	Todos	
Revisión prevista de la documentación	28/4/2022	Se comprueba el avance correcto en la documentación. Se discuten detalles técnicos respecto al manual de usuario y se establece una fecha tentativa de entrega del proyecto para el jueves 28 de abril	Todos	
Retroalimentación positiva del trabajo	26/4/2022	Se comparten consejos entre ambas partes, se discuten errores sucedidos a lo largo del trabajo, y se brindan sugerencias para la mejora de futuros trabajos.	Todos	

9

Cronograma

Escuela de Computación		Escuela de Computación																											
Nombre de la asignación		Tarea Programada #1																											
Fecha de inicio:		4.04.2022																											
Fecha Final:		27.04.2022																											
Recursos Técnicos:		Carlos Guzmán (I1), Samuel Garcés (I2)																											
		Semana 1							Semana 2							Semana 3													
Tarea a Realizar	Lunes 4	Martes 5	Miércoles 6	Jueves 7	Viernes 8	Sábado 9	Domingo 10	Lunes 11	Martes 12	Miércoles 13	Jueves 14	Viernes 15	Sábado 16	Domingo 17	Lunes 18	Martes 19	Miércoles 20	Jueves 21	Viernes 22	Sábado 23	Domingo 24	Lunes 25	Martes 26	Miércoles 27	Jueves 28				
Planificación inicial	Todos	Todos																											
Creación del repositorio en GitHub		I1	I1																										
Creación del menú			I1	I1																									
Documentación inicial			I2		I2																								
Revisión y corrección de errores				Todos	Todos																								
Semana Santa						Semana Santa																							
Creación del ejercicio 1								I2	I2																				
Creación del ejercicio 2								I1	I1																				
Revisión y corrección de errores									Todos	Todos																			
Creación del ejercicio 3										I2	I2																		
Creación del ejercicio 4										I1	I1																		
Revisión y corrección de errores												Todos	Todos																
Creación del ejercicio 5																						I2	I2						
Creación del ejercicio 6																						I1	I1						
Revisión y corrección de errores																						Todos	Todos						
Creación del ejercicio 7																							I2	I2					
Creación del ejercicio 8																							I1	I1					
Revisión y corrección de errores finales																								Todos	Todos				
Documentación Final																						Todos	Todos	Todos	Todos				

Samuel Garcés – 2022129139

Carlos Guzmán – 2022437782

Estadísticas de tiempos

Actividad realizada	Horas
Análisis de problemas	8
Diseño de algoritmos	5
Investigación de métodos diversos	6
Programación	14
Documentación del programa	2
Pruebas del programa	4
Elaboración del manual de usuario	2
Elaboración de documentación del proyecto	6
TOTAL	47 horas