

Taller de programación

Semestre I

Año 2025

Matias Benavides Sandoval

2025102376

Luis Carlos Tinoco Vargas

2025073699

TP#1

9/05/2025

Estatus de la Tarea:

índice

1. Enunciado del proyecto

pag 3.

2. Justificación de olores de software	pag 3.
3. Conclusiones del trabajo	pag 4.
4. Bitácora de programador	pag 5.
5. Aprendizajes obtenidos	pag 6.
○ Carácter personal	
○ Carácter técnico	
6. Validaciones estratégicas	pag 7.
7. Reglamento de trabajo	pag 7.
8. Agendas y minutas	pag 8-9.
○ Agenda 2025-S1-##	
○ Minuta 2025-S1-##	
○ Agenda 2025-S1-##	
○ Minuta 2025-S1-##	
9. Estadística de tiempos	pag 10.
○ Matias	
○ Luis	

Enunciado del proyecto:

Para conocer más sobre las instrucciones de esta Tarea programada, vea al archivo TP#1_NotasEstudiantes_S12025.pdf, en el que se encuentra todo lo que se evaluara en este trabajo

Eliminación de olores:

1. Usar la misma función para llamar a la Base de datos en vez de hacerlo para cada función por separado (a veces fueron necesarios ciertos cambios dependiendo de la función)

2. Generar los html con funciones separadas, para así no sobrecargar funciones y dejarlas solo para una función, en este caso solo generar los html

3. Comentar el código para así después ser capaz de entenderlo y que los demás también lo hagan, ser claros y hacer buenos comentarios.

```

'''
'''
Funcionamiento:
- Agrega estudiantes a la base de datos en un archivo CSV.
- Genera datos aleatorios o utiliza datos de un archivo de texto para los estudiantes.
Entradas:
- writer: objeto csv.writer para escribir en el archivo CSV.
- líneas: lista de líneas del archivo de texto con datos de estudiantes.
- cantidad: int, cantidad de estudiantes a agregar.
- anno1: int, año inicial para generar el carné.
- anno2: int, año final para generar el carné.
- n1, n2, n3: int, pesos de las evaluaciones.
Salidas:
- Escribe los datos de los estudiantes en el archivo CSV.
'''
'''

```

```
def leerCsv(archivoCsv):
    """
    Función:
    - Lee y carga los datos de un archivo CSV.
    Entradas:
    - archivoCsv: str, ruta del archivo CSV a leer.
    Salidas:
    - Retorna una lista con todas las filas del archivo CSV.
    - Muestra mensaje de error si el archivo no existe.
    """
    try:
        with open(archivoCsv, "r", newline="", encoding="utf-8") as archivo:
            lector = csv.reader(archivo)
            return list(lector)
    except FileNotFoundError:
        print(f"\nError: El archivo '{archivoCsv}' no fue encontrado.")
        return False
```

```

Función:
- Genera reporte HTML
Responsabilidades:
- Solo generación de HTML
Entradas:
- estudiantes: lista de tuplas procesadas
Salidas:
- estadísticas: tupla con estadísticas
Escribe archivo "Reporte.html"
'''
total, aprobados, reprobacion, reprobados = estadísticas
with open("Reporte.html", "w", encoding="utf-8") as f:
    # Encabezado
    f.write("""<DOCXVPE.html>

<html>
<head>

<title>Reporte de Notas</title>
<style>
table { border-collapse: collapse; width: 100%;
tr, td { padding: 5px; text-align: left; border: 1px solid #ddd; }
th { background-color: #666666; color: white; }
tr>th-child(even) { background-color: #f2f2f2; }
</style>
</head>
<body>
<h1>Reporte de Estudiantes</h1>
<table>
<tr>
<th>Nombre</th>
<th>Genero</th>
<th>Curso</th>
<th>Curso</th>
<th>Notas</th>
<th>Estado</th>
</tr>
'''
    # Datos
    # For: for i in estudiantes:
    f.write("""
<tr>
<td>{est[0]}</td>
<td>{est[1]}</td>
<td>{est[2]}</td>
<td>{est[3]}</td>
<td>{est[4]}</td>
<td>{est[6]}</td>
</tr>
'''
    # Resumen
    f.write("""
<h3>Resumen
<table>
<tr>
<td>Total estudiantes: {total}</td>

```

4. Eliminar código que no se use, o que haya sido usado solo para pruebas específicas pero que ya no sirven, borrarlo para que así, no dificulte la comprensión del código.

5. Eliminar el uso de cosas muy complejas pero innecesarias que solo hacen mas dificil la lectura y comprension del codigo.

Conclusiones del trabajo:

Al finalizar este trabajo (TP), hemos podido, evaluar nuestro nivel a la hora de hacer un programa, hemos aprendido sobre el uso del HTML, el css, el XML, la creación de una base de datos, y el uso de diversas librerías y herramientas para poder lograr el trabajo.

Hemos aprendido sobre buenas practicas del software, la eliminación de olores de software, el uso de funciones con responsabilidad única, el uso, creación y modificación de archivos, html, csv, y xml.

También sobre como se trabaja en proyectos, tanto para los siguientes proyectos (independientemente del curso), como para un futuro laboral, también sobre las habilidades blandas y su utilidad, y el como podemos mejorar cada uno de nosotros en diversos aspectos, como la organización, el buen uso del tiempo, la responsabilidad, y eficiencia y eficacia a la hora de trabajar en esta clase de proyectos.

Además, pudimos detectar nuestras debilidades, y nuestras fortalezas, como también comprender el no acaparar mucho y también dejar al compañero trabajar, y pedir ayuda, como también el buscar información, soluciones y métodos, de diversas fuentes para lograr la culminación del trabajo.

Y con todo esto, ser un poco mejor programadores, para los siguientes desafíos.

Tecnológico de Costa Rica
IC-1803 Taller de Programación
Bitácora de programador

	Día que descubre el problema	Descripción del problema	Día de solución	Descripción de la solución.	Tiempo transcurrido en horas
1	3/5/2025	El programa no encontraba los archivos	3/5/2025	Abrir la tarea programada desde el archivo unico para ella y asi el programa solo vera lo de esa carpeta	2 horas
2	3/5/2025	En la base de datos, los datos no aparecian por mas de no dar error	3/5/2025	Habia un error de logica que no pegaba un nombre tras otro si no que los remplazaba, habia que cambiar eso y ya	1 hora
3	4/5/2025	No creaba los tres archivos de los que se aplazaban, solo uno	4/5/2025	Habia un problema en el cual al crear un archivo no le cambie el nombre por loo cual creaba los 3 igual y los sobreescribia, solo era crear tres pro aparete y cambiar el nombre por aparte	1 hora
4	4/5/2025	Cuando aparecia un error, las funciones no se dejaban de llamar a si mismas	4/5/2025	Habia un problema con el nombre de algunas funciones y se estaban llamando entre ellas sin parar, con cambiar el nombre de algunas y mejorar los returns se arreglo	
5	4/5/2025	El XML, no mostraba lso datos correctos	4/5/2025	Un error con el como identificaba posiciones de las notas en la base de datos hacia que no le pusiera el estado a ningun estudiantes, puse un formato fijo y agarraba el ultimo se arreglo	1 hora
6					
7					

Aprendizajes obtenidos:

Carácter personal:

1. Buen uso del tiempo, organización, un calendario y notas.
2. Investigación y todo lo que refiera con la búsqueda de información tanto en internet como en otras personas.
3. El como ser mas productivo y poder hacer cosas en menos tiempo, por la falta de distractores externos.
4. Como sacar provecho (a veces funcional otras no tanto) a una parte mas creativa en la solución de problemas.
- 5.
- 6.
- 7.
- 8.

Carácter técnico

1. Uso de HTML y CSS del cual desconocía casi por completo, pero del cual tuve que aprender.
2. Uso de librerías, para diferentes funciones como el crear el PDF, o las propias ER.
3. El cómo se crean bases de datos en .csv, sus funcionalidades, y posibles utilidades de estas en un proyecto mas profesional.
4. Usar y hacer cosas en XML, algo que del todo desconocía y tuve que aprender como usar.
- 5.
- 6.
- 7.
- 8.

Validaciones estratégicas:

```
def crearBaseDeDatos(prioridad, porcentaje, cantidad, porcentaje2, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72, n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86, n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100):
    """
    Funcionamiento:
    - Valida las variables ingresadas y llama a la función principal para crear la base de datos.
    Entradas:
    - prioridad: int, cantidad de estudiantes del primer recurso.
    - porcentaje: int, porcentaje de estudiantes del primer recurso.
    - cantidad: int, cantidad de estudiantes del segundo recurso.
    - porcentaje2: int, porcentaje de estudiantes del segundo recurso.
    - n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72, n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86, n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100: int, número de los estudiantes.
    Salidas:
    - None: no se pudo crear la base de datos.
    - True: se pudo crear la base de datos.
    """
    try:
        # Validar los datos
        if prioridad < 0 or porcentaje < 0 or cantidad < 0 or porcentaje2 < 0 or n1 < 0 or n2 < 0 or n3 < 0 or n4 < 0 or n5 < 0 or n6 < 0 or n7 < 0 or n8 < 0 or n9 < 0 or n10 < 0 or n11 < 0 or n12 < 0 or n13 < 0 or n14 < 0 or n15 < 0 or n16 < 0 or n17 < 0 or n18 < 0 or n19 < 0 or n20 < 0 or n21 < 0 or n22 < 0 or n23 < 0 or n24 < 0 or n25 < 0 or n26 < 0 or n27 < 0 or n28 < 0 or n29 < 0 or n30 < 0 or n31 < 0 or n32 < 0 or n33 < 0 or n34 < 0 or n35 < 0 or n36 < 0 or n37 < 0 or n38 < 0 or n39 < 0 or n40 < 0 or n41 < 0 or n42 < 0 or n43 < 0 or n44 < 0 or n45 < 0 or n46 < 0 or n47 < 0 or n48 < 0 or n49 < 0 or n50 < 0 or n51 < 0 or n52 < 0 or n53 < 0 or n54 < 0 or n55 < 0 or n56 < 0 or n57 < 0 or n58 < 0 or n59 < 0 or n60 < 0 or n61 < 0 or n62 < 0 or n63 < 0 or n64 < 0 or n65 < 0 or n66 < 0 or n67 < 0 or n68 < 0 or n69 < 0 or n70 < 0 or n71 < 0 or n72 < 0 or n73 < 0 or n74 < 0 or n75 < 0 or n76 < 0 or n77 < 0 or n78 < 0 or n79 < 0 or n80 < 0 or n81 < 0 or n82 < 0 or n83 < 0 or n84 < 0 or n85 < 0 or n86 < 0 or n87 < 0 or n88 < 0 or n89 < 0 or n90 < 0 or n91 < 0 or n92 < 0 or n93 < 0 or n94 < 0 or n95 < 0 or n96 < 0 or n97 < 0 or n98 < 0 or n99 < 0 or n100 < 0:
            print("Error: Los datos no son válidos.")
            return False
        # Crear la base de datos
        with open('base_datos.csv', 'w', encoding='utf-8') as archivo:
            writer = csv.writer(archivo)
            writer.writerow([prioridad, porcentaje, cantidad, porcentaje2, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72, n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86, n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100])
        return True
    except Exception as e:
        print(f"Error: {e}")
        return False
```

Ayuda al código que crea la base de datos, a no poder recibir valores incorrectos y que luego podrían generar problemas.

```
def obtenerCurvaAux(porcentaje):
    """
    Funcionamiento:
    - Valida que el porcentaje esté en el rango correcto
    Entradas:
    - porcentaje: float, valor a validar
    Salidas:
    - float: porcentaje si es válido
    - None: si el porcentaje está fuera de rango
    """
    if 0 <= porcentaje <= 100:
        return porcentaje
    print("El porcentaje debe estar entre 0 y 100")
    return None
```

Evitar que se puedan meter datos erróneos en el HTML de la curva para que así pueda hacerlo correctamente

```
try:
    with open(nombreArchivo, 'r', encoding='utf-8') as archivo:
        return list(csv.reader(archivo))
except FileNotFoundError:
    print(f"Error: El archivo '{nombreArchivo}' no fue encontrado.")
    return False

if Csv is False:
    print("No se pudo leer el archivo CSV.")
```

En los casos en los cuales por alguna razón no pueda encontrar el archivo

O en los casos que no lo pueda leer por algún error

Reglamento de trabajo:

-Hacer las cosas, bajo los marcos del tiempo establecidos (Todo lo que sea posible y si no tener una razón).

-Tener al día al compañero sobre como va el avance del trabajo y sobre cuanto le falta y correcciones que ha hecho.

-Asumir las responsabilidades repartidas de manera equitativa y respetar el trabajo del compañero (Si lo hace uno que el otro no se ponga a hacerlo por que sí).

-Mantener el orden el código y a la hora de unir los códigos, no hacer grandes cambios sin estar de acuerdo y preguntar.

-Respetarnos entre nosotros en todo momento.

Agendas y minutas:

Agenda 2025-S1-##

Fecha: 28-4-2025			Hora de Inicio: 7:00pm	Hora Final: 8:00pm
Lugar de reunión: Virtual				
Invitados a la reunión: Matias Benavides y Luis Tinoco				
Tiempo	Tema	Responsable	Requiere llevar	Logro Esperado
15min	Lectura de la TP	Luis	La tarea programada	Lograr comprender completamente que es lo que se debe de hacer para la realización de la tarea
20min	División de tareas	Matias	Un esquema para dividir los ejercicios de la Tarea	Lograr estar de acuerdo sobre el cómo se dividirá las tareas
10min	Fuentes de información	Luis	Información (métodos), para lograr hacer la tarea	Tener suficiente información para realizar la tarea
15min	Planes para la documentación	Matias	Un plan para rellenar los documentos	Terminar que hacer la idea de la documentación de la Tarea Programada

Minuta 2025-S1-##

Fecha:1-5-2025		Hora de Inicio: 7:00pm		Hora Final: 8:00pm	
Lugar de reunión: Virtual					
Asistentes a la reunión: Matias Benavides Luis Tinoco			Hora de Llegada: 7:00pm		
Ausentes en la reunión:			Justificación:		
Tema		Fecha límite de cumplimiento	Acuerdos		Responsable
Planes para la documentación		3-5-2025	Terminar el planeamiento de la documentación		Matias
División de tareas		3-5-2025	Haber dividido justamente las tareas		Luis
Fuentes de información		5-5-2045	Haber recopilado, la suficiente información, para realizar la tarea		Matias
Realizar lógica de la TP		6-5-2025	Tener un plan para la lógica de la TP		Luis

Agenda 2025-S1-##

Fecha: 4-5-2025		Hora de Inicio: 6:30pm	Hora Final: 7:15pm	
Lugar de reunión: Virtual				
Invitados a la reunión: Matias Benavides y Luis Tinoco				
Tiempo	Tema	Responsable	Requiere llevar	Logro Esperado
20min	Avances de la TP	Luis y Matias	Sus respectivos avances de la tarea programada.	Tener ya un avance considerable en las diferentes partes del proyecto.
15min	División de las próximas tareas	Matias y Luis	Las partes que siguen que quiera hacer cada uno.	Llegar a un acuerdo sobre que va a ser lo siguiente que va a hacer cada uno.
15min	revisión y ayuda de errores	Luis y Matias	Los errores que se ha encontrado y no ah podido solucionar	Ayudarnos entre nosotros para poder solventar estos errores

Minuta 2025-S1-##

Fecha:6-5-2025		Hora de Inicio: 6:30pm		Hora Final: 7:15pm	
Lugar de reunión: Virtual					
Asistentes a la reunión: Matias Benavides Luis Tinoco			Hora de llegada: 6:30pm		
Ausentes en la reunión:			Justificación:		
Tema		Fecha límite de cumplimiento	Acuerdos		Responsable
Terminar por completo los primeros avances		5-5-2025	Terminar todo el código por separado cada uno.		Matias y Luis
Terminar las ultimas tareas que fueron asignadas		7-5-2025	Terminar lo que queda del proyecto y unir todo.		Luis y Matias
Corregir todos los errores que se encontraron		6-5-2045	Haber corregido todos los errores y tener el proyecto prácticamente listo		Matias y Luis

Cronograma:

Escuela de Computación
Nombre de la asignación: Tarea programada
Fecha de Inicio:
22/4/2025
Fecha Final:
9/5/2025
Recursos Técnicos: Matias (M),
Luis (L)

Tarea a Realizar	Semana 1							Semana 2							Semana 3						
	LunesX	MartesX	MiércolesX	JuevesX	ViernesX	SábadoX	DomingoX	LunesX	MartesX	MiércolesX	JuevesX	ViernesX	SábadoX	DomingoX	LunesX	MartesX	MiércolesX	JuevesX	ViernesX	SábadoX	DomingoX
Retos 1,3, 4, 6, 8										M	M	M		M	M						
Retos 2,5, 7, 9, 10																					
Documentacion																Ambos	Ambos				
Unificaico del codigo																Ambos	Ambos				
Correccion de Errores														Ambos	Ambos	Ambos	Ambos	Ambos			
Macro																					

Estadística de tiempos:

Matias:

Luis:

Actividad Realizada	Horas	Actividad Realizada	Horas
Análisis de requerimientos	1 hora	Análisis de requerimientos	1 hora
Reto1	4 horas	Registrar estudiante	2 horas
Reto3	3 horas	Lógica de la función	1.30 horas
Reto4	2 horas	Reto 7	3 horas
Reto6	2 horas	Reto 9	2 horas
Reto8	2 horas	Reto 10 y 11	3 horas
Documentación del Código	2 horas	Arreglar errores de los retos	2.30 horas
Elaboración de documentación del proyecto	1 hora	Unificación de códigos	1 hora
Investigación de métodos para hacer los proyectos	2 horas	Errores de código unificado	2 horas
Total	19 horas	Total	17 horas