Tarea Programada #1

1. Objetivos

- Desarrollar en el estudiante la capacidad de resolver problemas en contextos modernos de programación.
- Poner en práctica los conocimientos adquiridos hasta el momento, en temas como iteración, estructuras condicionales, funciones.
- Utilizar la estrategia divide y vencerás para resolver un problema general, solucionando los subproblemas que lo conforman.
- Integrar todos los conocimientos adquiridos para crear un producto de software con un propósito significativo.
- Desarrollar habilidades blandas para poder trabajar correctamente en equipo: ética, autodidactas, idiomas, empatía, trabajo en equipo, etc.
- Desarrollar estrategias de investigación y uso del idioma inglés según corresponda.
- Implementar las buenas prácticas de "código limpio" y eliminación de "olores de software".
- Uso deseable de Git Hub para el control de versiones.

2. Marco teórico

I. Archivos

Los archivos son conjuntos de datos residentes en almacenamiento secundario, como discos, que mantienen la información aun cuando se apague el computador. Los datos almacenados en archivos se conocen como datos persistentes.

Python ve cada archivo como un flujo secuencial de caracteres, donde una marca de EOF (*End of File*) determina el fin del archivo.

Con ello, usted puede crear archivos de texto de la extensión que guste, por ejemplo: ***.txt***, .csv, ***.html***, .xls, **.xml**, etc. Ya cada aplicación al abrirlo inteligentemente reacciona para abrirlo según corresponda.

Pero adicional a ello, se pueden guardar estructuras binarias, por ejemplo: listas, listas de listas, tuplas, diccionarios o listas de objetos.

Las posibles operaciones con archivos son: apertura del archivo, lectura, escritura y cerrado del archivo. Para mayor detalle referirse al capítulo 10 del libro *Introducción a la Programación en Python* del Profesor Jaime Solano. Adicionalmente puede consultar el siguiente vínculo.

Expresiones regulares

Es una secuencia de caracteres que forma un patrón de búsqueda, principalmente utilizada para la búsqueda de patrones de cadenas de caracteres u operaciones de sustituciones.

En el área de la programación las expresiones regulares son un método por medio del cual se pueden realizar búsquedas dentro de cadenas de caracteres. Sin importar la amplitud de la búsqueda requerida de un patrón definido de caracteres, las expresiones regulares proporcionan una solución práctica al problema. Adicionalmente, un uso derivado de la búsqueda de patrones es la validación de un formato específico en una cadena de caracteres dada, como por ejemplo fechas o identificadores.

Por lo tanto, en esta tarea, el uso de expresiones regulares es recomendado para determinar cuándo una palabra es un verbo en presente mediante sus clasificaciones: infinitivo (terminado en ar, er, ir), participio (terminado en ando, iendo) y gerundio (terminado ado, ido, to, so, cho.)

Para mayor información con respecto a la sintaxis y uso de expresiones regulares en Python 3.5.1, puede consultar el siguiente vínculo: https://docs.python.org/3/library/re.html

Ejemplo de expresión regular para validar una fecha en formato “dd/mm/aaaa”:

```
# Importar la librerIa para el manejo de expresiones regulares
import re

def esFechaValida(fecha):
    expresionRegularFecha = "[0-9]{2}\/[0-9]{1}[0-2]{1}\/[0-9]{4}"
    if re.match(expresionRegularFecha, fecha):
        return True
    else:
        return False
```

Referencias adicionales sobre expresiones regulares

https://platzi.com/blog/expresiones-regulares-python/

http://python-para-impacientes.blogspot.com/2014/02/expresiones-regulares.html

HTML 5

Antecedentes

Los orígenes de la Web

Internet no solo ha marcado uno de los más importantes avances tecnológicos del siglo XX, sino que también ha acompañado un cambio cultural de trascendencia que, en pleno siglo XXI, se mantiene en constante evolución. Pero toda historia tiene un comienzo, e Internet también lo tuvo, mucho antes de ser un fenómeno masivo.

La historia cuenta que el antecesor de Internet fue el proyecto conocido como ARPANET, una red descentralizada que algunos organismos estadounidenses utilizaron a partir de la década del sesenta. Sin embargo, el gran cambio se produciría entre fines de los ochenta y principios de los noventa, con la llegada de lo que se conoce como World Wide Web, es decir WWW, el sistema que se encarga de permitir la distribución de información mediante hipertexto.

De la mano de este cambio, comienza a popularizarse Internet en la población. Los usuarios ahora podían acceder a contenidos de la gran red, tan solo con disponer de una conexión mediante un módem y un navegador con la capacidad de interpretar contenidos de hipertexto. Esta etapa de Internet, que comprende aproximadamente desde principios de los noventa hasta el año 2003, es considerada como Web 1.0.

El concepto de este primer paradigma de la Web responde a la idea de una web “estática” o de una “sola vía”, donde el usuario es solo un “espectador” que recibe o lee contenidos, publicados por el Webmaster o dueño del sitio. Este paradigma se modificaría de manera sustancial con la llegada de la denominada Web 2.0.

Web 2.0

Los cambios en la Web no solo responden a temas tecnológicos, sino que estos van de la mano con la evolución de los hábitos de los usuarios, las tendencias en los modos de navegación, las necesidades del mercado y hasta con aspectos culturales que también influyen en este conjunto.

La Web 2.0 representa principalmente un cambio cultural en Internet. Los usuarios, cansados de un rol pasivo, comienzan a buscar alternativas de participación. Nace una web social, donde los blogs, las redes sociales y las aplicaciones online son las estrellas.Esto ocurre a partir del año 2004.

Web 3.0

El concepto de Web 3.0 es, quizás, más complejo de definir y discutido que el caso de sus predecesores: la Web 1.0 y 2.0. Existen diversas características que la definen, entre las cuales podemos mencionar: semántica, geolocalización, Web 3D, accesibilidad desde diversos dispositivos y también inteligencia artificial.

La Web semántica, como muchas veces se define a la Web 3.0, se refiere al uso de etiquetas o bien de metadatos para otorgar un significado semántico a los elementos de la Web. Esto posibilita cierta automatización y la posibilidad de utilizar, con un mayor nivel de eficiencia, los agentes inteligentes que pueden realizar detección de contenidos.

Las características de geolocalización, muy empleadas en los equipos móviles, también han llegado a nuestro escritorio. Aunque aún pueden no ser tan precisas, las técnicas cada vez son más depuradas, y las mejoras en este campo no detienen su avance. Poder identificar a una persona, un dispositivo o cualquier elemento de manera geoespacial abre todo un mundo de posibilidades en el campo de la informática y, en especial, para todo lo referente a Realidad Aumentada.

La posibilidad de acceder desde distintos dispositivos es una realidad para una gran cantidad de usuarios y un desafío muy importante para diseñadores y desarrolladores web. Los usuarios ya no están limitados a utilizar Internet desde una computadora de escritorio, ni siquiera dependen de una laptop. Teléfonos móviles, tablets, lectores de libros electrónicos y consolas de videojuegos son solo algunas de las posibilidades que se presentan para que el usuario pueda acceder a Internet en cualquier momento y desde cualquier lugar.

W3C

El World Wide Web Consortium (W3C) es el ente o consorcio, de alcance internacional, que se encarga de crear las reglas que se utilizan como recomendaciones fundamentales para la estandarización de los principales lenguajes y tecnologías utilizados en Internet, como el caso de HTML, CSS, XML, DOM y SVG

Lenguajes de etiquetas

Los lenguajes de etiquetas, también conocidos como lenguajes de marcado o de marcas, son los que nos permiten estructurar un documento mediante el uso de etiquetas. Un ejemplo muy popular de un lenguaje de etiquetas es HTML. Algunos otros son: XML, SGML, entre otros.

HTML

HTML (HyperText Markup Language o lenguaje de marcado de hipertexto) es el lenguaje de etiquetas que funciona como una de las piedras angulares de la World

Wide Web. Aunque la evolución de Internet nos ha traído muchos avances en lo que se refiere a tecnología (Web 2.0 y Web 3.0, mediantes), el lenguaje de etiquetas que se popularizó en la década del noventa sigue siendo fundamental para el desarrollo web, ya que es el que comprenden e interpretan los navegadores.

HTML5

HTML5 plantea una evolución necesaria para HTML, que luego de más de una década en la versión 4.01 necesitaba, de manera imperiosa, una renovación para estar al día con las necesidades del desarrollo web actual.

En HTML5, se destacan sus características semánticas, las posibilidades multimedia que incorpora, las nuevas funciones para formulario y las características que se definen para poder integrarse con tecnologías que permitirán abrir una nueva etapa en Internet, en lo que se refiere a la arquitectura de las aplicaciones. Por estos motivos, HTML5 es considerado como uno de los motores más importantes de la Web 3.0.

Ejemplo de estructura básica de un documento en formato HTML5

```
<!DOCTYPE html>

<html lang="es">

<head>
<title>Titulo de la web</title>
<meta charset="utf-8" />
<link rel="stylesheet" href="estilos.css" />
<link rel="shortcut icon" href="/favicon.ico" />
<link rel="alternate" title="Pozolería RSS" type="applicat
</head>

<body>
    <header>
       <h1>Mi sitio web</h1>
       <p>Mi sitio web creado en html5</p>
    </header>
    <section>
       <article>
           <h2>Titilo de contenido<h2>
           <p> Contenido (ademas de imagenes, citas, videc
       </article>
    </section>
    <aside>
       <h3>Titulo de contenido</h3>
           <p>contenido</p>
    </aside>
    <footer>
        Creado por mi el 2011
    </footer>
</body>
</html>
```

XML

Es un estándar ampliamente soportado para describir datos. XML es comúnmente usado para intercambiar datos entre aplicaciones sobre internet. crear marcas para virtualmente cualquier tipo de información, lo cual posibilita la creación de nuevos lenguajes de marcas para describir cualquier tipo de como fórmulas matemáticas, música, noticias, recetas, financieros, entre muchos otros.

Permite datos, reportes

Una de las características más importantes de XML es que describe los datos de forma tal que sean entendibles tanto para los humanos como para las computadoras.

A continuación se detalla un ejemplo de un documento XML:

```
<Books>
    <Book ISBN="0553212419">
        <title>Sherlock Holmes: Complete Novels...
        <author>Sir Arthur Conan Doyle</author>
    </Book>
    <Book ISBN="0743273567">
        <title>The Great Gatsby</title>
        <author>F. Scott Fitzgerald</author>
    </Book>
    <Book ISBN="0684826976">
        <title>Undaunted Courage</title>
        <author>Stephen E. Ambrose</author>
    </Book>
    <Book ISBN="0743203178">
        <title>Nothing Like It In the World</title>
        <author>Stephen E. Ambrose</author>
    </Book>
</Books>
```

Todo documento XML está compuesto de elementos que especifican la estructura del documento. Algunas de sus características son las siguientes:

- Los documentos XML delimitan los elementos con marcas o etiquetas de inicio y fin. Una marca de inicio consiste del nombre del elemento entre corchetes angulares. Ejemplo: **<author>**
- Una marca de cierre consiste del nombre del elemento precedido por un forward slash (/) entre los corchetes angulares. Ejemplo: **</author>**
- Las etiquetas inicial y final de un elemento encierran el texto que representa los datos Ejemplo: **<author>Stephen E. Ambrose</author>**
- Cada documento XML debe tener exactamente un elemento raíz que contiene todos los demás elementos Ejemplo: **<Books>**

Para mayor información con respecto a este tema puede acceder al siguiente recurso: http://www.w3schools.com/xml/

Por hacer:

Implementar una solución computacional ***sin*** interfaz gráfica, es decir, ejecutándose desde el Shell de Python con el único diseño que se especifica en cada rubro o NO se califica la tarea, por ende, nota 0. Además, la programación debe ser únicamente con las estructuras vistas en clase, no se permiten estructuras no enseñadas en la misma, por ende, el uso de IA´s no está autorizado, si se detecta en algún lugar, nuevamente nota 0 a ambos de la pareja.

Al abrir el aplicativo muestre el menú siguiente:

1. Crear BD dinámica
2. Registrar un estudiante
3. Generar reporte HTML y .csv
4. Respaldar en XML
5. Reporte por género (.docx)
6. Gestionar curva
7. Envió de correos para reposición.
8. Aplazados en al menos 2 exámenes (.pdf)
9. Estadística por generación
10. Reporte por sede con buen rendimiento.
11. Salir

A continuación, se detalla cada requerimiento:

1. Crear BD dinámica

Debe formar una lista de "X" estudiantes extrayendo aleatoriamente el Z% de estudiantes de cada fuente.
Fuente 1: Usando la librería "Names" permita generar "Y" cantidad de alumnos con: 1 nombre, 2 apellidos, 1 género. Si el usuario solicita la creación de 120 estudiantes e indica un Z% de 45, deben sacarse aleatoriamente 54 estudiantes -redonde a un número más grande-, pero sin repetir.
Fuente 2: Usando la lista de estudiantes suministrados en el momento de la revisión -va a variar según cada corrida en las revisiones de la tarea programada- si el archivo trajo 75 estudiantes debe sacar el 45%, si es par o la mitad redondeada al número mayor, es decir, si son 75 debería de sacar aleatoriamente 34 nombres aleatorios. Use el archivo que se le suministre en el momento de la revisión de la tarea, para sus practicas se le entregará uno semejante.
Observe Names genera géneros en inglés y los nombres proveídos están en español, genere su estrategia para procesar estandarizando.

Por ende, la BD dinámica, tendrá 54 + 34 = **88 estudiantes**.

Ello debe formar el formato para cada persona:
[(nombre,apellido1,apellido2),True]

Ejemplo:
[("Juan","Pérez","Mora"),True]

La lista de estudiantes se vería:

```
[
    [("Juan","Pérez","Mora"),True,] ,
    [("Ana","Araya","Fuentes"),False] ,
    ...,
    [("Rodolfo","Marín","Solís"),True]
]
```

Debería contener por ende **88 estudiantes**.

A ello debe generar automáticamente:

- Carné
- Correo electrónico
- Notas

Carné

Pregunte al usuario el rango de años, por ejemplo:

- ✓ Año inicial
- ✓ Año Final

Si el usuario especifica ***2021*** y ***2025***, entonces deben generarse carné con esos rangos de años. *Verifique* el segundo sea mayor o igual al primero, insista hasta ser correctos.

El carné de se forma de 10 dígitos numéricos: AñoSedeCualquiera

- 4 valores para el año
- 2 valores para la sede
- 4 valores para un valor cualquiera

La sede se obtiene según el archivo que nuevamente se entregará en el momento de la revisión. Se provee uno de ejemplo que variará. Eso sí, algorítmicamente verifiquen no sean repetidos, usted es libre de hacerlo de la manera que usted guste, con la única materia vista en el momento de asignar la tarea programada.
Si fueran 3 sedes entonces formaría los códigos, únicos códigos permitidos:

Código	Sede
01	Campus Tecnológico Central Cartago

02	Campus Tecnológico Local San Carlos
03	Campus Tecnológico Local San José

Al generarse aleatoriamente con carné válido sería: ***2021***021234

Ejemplos de carné inválidos serían: ***202***0021234, ***2022***041234, ***2022***03123

Al generar un carné, *debe verificar* sea único o debe insistir hasta generar uno diferente.

Correo electrónico

El correo electrónico debe formarse por:

- Inicial del nombre
- Apellido 1
- Valores Cualquiera del carné
- @estudiantec.cr

Si el estudiante se llama: Juan Pérez Mora y su carné generado es el ***2021***021234, por ende, el correo debe ser: jperez1234@estudiantec.cr, minúsculas y sin caracteres especiales. Recuerde *verificar* sea correo único en esa BD.

Notas

Deben generarse aleatoriamente 3 valores enteros diferentes de 0, y un cuarto valor y quinto obligatoriamente 0.0 (correspondiente a la nota final del curso y un posible redondeo o nota de acta), y colocarse en una tupla, por ejemplo**: (85,47,70,0.0, 0.0)**

Por cada corrida en la tarea programada, pregunte al usuario el porcentaje de las 3 evaluaciones fijas, valide siempre sumen 100.

Ejemplos

- 25,35,40 ello es correcto pues suma 100
- 20,30,50 ello es correcto pues suma 100
- 15,25,60 ello es correcto pues suma 100

Si el usuario indicara la sobresaliente, por ende, sería:

- Una nota 85 de 25%, entonces obtuvo un porcentaje de 21.25
- Una nota 47 de 35%, entonces obtuvo un porcentaje de 16.45
- Una nota 70 de 40%, entonces obtuvo un porcentaje de 28.0

Eso implica que: 21.25 + 16.45 + 28.0 suman 65.7

Posterior a todos los cálculos, actualice los últimos valores de la tupla: **(85,47,70,65.7, 65.7)**

Por ende, la lista de listas debe cambiar a:

```
[
[("Juan","Pérez","Mora"),True,2021021234,"jperez1234@estudiantec.cr", (85,47,70,65.7, 65.7)]
,
        ...
]
```

Contra esa estructura genere toda la tarea programada…

2. *Registrar un estudiante*

Recuerde solicitar la menos cantidad de datos a escribir por el usuario final con el fin de evitar cometa errores y le ensucie la BD.
Solicite el nombre completo, su género, su carne (no puede ser repetido y valídelo por una **expresión regular**), su correo electrónico (no puede ser repetido, valide por **expresión regular**) y las 3 notas (no pida los porcentajes, aplican los mismos al crear la BD Dinámica de esa corrida)
Los datos correspondiente a la nota final del curso y un posible redondeo o nota de acta, deben generarse dinámicamente.

3. *Generar reporte HTML y .css*

El formato de salida corresponde a un archivo .html.
Este archivo debe tener tres secciones a saber:

- El título de esta opción del menú.
- La tabla de análisis del documento.
- Reporte

Debe crearse la siguiente estructura visual para la tabla de análisis del documento al leer la lista de listas generada:

Detalle de notas						
Nombre	***Apellidos***	***Género***	***Carné***	***Correo***	***Notas***	***Estado***
Juan	Pérez Mora	Masculino	***2021***021234	jperez1234@estudiantec.cr	**85,47,70,65.7**	Reposición
.	.	.	.	.	.	.
.	.	.	.	.	.	.
					Valores extraídos de la tupla…	

Debe mostrarse una tabla de 2 colores horizontales, ello para mejorar la lectura.
Sea observador, muestre los títulos con los formatos indicados en la tabla.

El **reporte** debe indicar:
La base de datos posee **89 estudiantes (88 generados dinámicamente más el que acabo de ingresar)**, de los cuales hay: 50 aprobados para un 56.17%, 18 a reposición para un 20.22%, 21 reprobados para un 23.59%.
Considere que:

- <60, reprobado
- <70, reposición

- >70, aprobado

Adicional a ello, transfiera todos los datos de la tabla a un archivo .csv separador por coma para poder ser abiertos posteriormente en excell, ello facilitará la revisión de la tarea Programada.

4. Respaldar en XML

Cree la siguiente estructura con los 89 o más datos de la BD según corresponda:

```
<Estudiantes>
    <Generacion anno="2021">
        <Estudiante carne=2021021234>
            <nombre>Juan Pérez Mora</nombre>
            <genero>Masculino</genero>
            <correo>jperez1234@estudiantec.cr</correo>
            <notas>85,47,70,65.7, 65.7</notas>  #extraiga de tupla
            <estado>Reposición</estado>
        </Estudiante>
        …
    </Generacion>
    <Generacion anno="2022">
        …
    </Generacion>
    <Generacion anno="2023">
        …
    </Generacion>
    <Generacion anno="2024">
        …
    </Generacion>
    <Generacion anno="2025">
        …
    </Generacion>
</Estudiantes>
```

5. Reporte por género (.docx)

Genere 2 archivos de Word, uno con los datos de todas las mujeres (mujeres.docx) y otro con los datos completos de los hombres(hombres.docx), ordenados por posible redondeo o nota de acta, de mayor a menor:
Ejemplo de lo esperado:

Título del documento posible redondeo o nota de acta, 3 notas obtenidas, nombre completo, carné, correo Al final indique: porcentaje de cada evaluación con lo cual se generó la posible redondeo o nota de acta y la cantidad de personas según esa clasificación.

6. *Gestionar curva*

Solicite un porcentaje de curso a aplicar a la BD total, aplíquelo a todos y actualice el redondeo o nota de acta y genere 3 archivos .html según el estado (aprobado, reposición, reprobados)

Este archivo debe tener cuatro secciones a saber:

- El título de según el estado a mostrar.
- % de curva indicado (pregúntelo al usuario)
- La tabla de análisis del documento.
- Reporte

Debe crearse la siguiente estructura visual para la tabla de análisis del documento al leer la lista de listas generada:

Nombre	***Apellidos***	***Género***	***Carné***	***Correo***	***Notas***	***Curva***
Juan	Pérez Mora	Masculino	*2021*021234	jperez1234@estudiantec.cr	**85,47,70,65.7**	**67,67**
.	.	.	.	.	.	
.	.	.	.	.	.	
					Valores extraídos de la tupla…	Asumiendo en esta oportunidad se dijo *3*%

Debe mostrarse una tabla de 2 colores horizontales, ello para mejorar la lectura.
Sea observador, muestre los títulos con los formatos indicados en la tabla.

El **reporte** debe indicar:
Para el estado de: **Reposición**, hay X cantidad de estudiantes luego de la curva, del total de **89 estudiantes (88 generados dinámicamente más el que acabo de ingresar)**,
Si hubiera alumnos que cambiaron de estado dado el incremento, deben mostrarse en el archivo correspondiente.

7. *Envió de correos para reposición.*

Envíe un correo electrónico a todo estudiante que su nota de posible redondeo o nota de acta corresponda al estado de **Reposición**, indicando que debe hacer el mismo el día “X”, a la hora “Y”.

8. Aplazados en al menos 2 rubros (.pdf)

Genere un .pdf con toda la información de los alumnos que aplazaron (nota <70) las 2 o 3 evaluaciones originales. Muestre toda su información.

Título del documento 3 notas obtenidas, nombre completo, carné, correo Al final indique la **cantidad** de estudiantes que tuvieron este inconveniente para un **porcentaje de**, según el **total de la BD** (indíquelo) -89 según hemos mostrado ficticiamente en esta corrida-. Reporte de nota mínima -sin importar el número de evaluación- Reporte de nota máxima (inferior a 70) -sin importar el número de evaluación- Cantidad de reprobados en 2 exámenes Cantidad de reprobados en 3 exámenes

Por ejemplo, en el **TEC** hay planes remediales para quienes han bajado sus notas, por ejemplo: el Programa de Tutorías Estudiantil (PTE) o Programa de Rendimiento Académico en Matemática (RAMA) en la Escuela de Matemáticas.

9. Estadística por generación

Según las generaciones dinámicas que formaron los números de carné, cree el siguiente reporte:

	Aprobados	Reposición	Reprobados	totales
2021	9	5	5	19
2022	13	6	3	22
2023	11	3	5	19
2024	8	4	2	14
2025	14	1	0	15
Totales	55	19	15	

Si nota, los totales deben sumar la cantidad actual de la BD, en este caso para esta corrida, los 89 estudiantes que hemos ido indicando para este caso ficticio.

10. Reporte por sede con buen rendimiento.

Muestre al usuario final la lista de sedes enumeradas, el usuario debe mostrar el número de sede que desea, luego de ello muestre la lista enumerada de 1 a N de los únicos estudiantes de la sede que obtuvieron buena nota en las 3 evaluaciones -nota superior o igual a 70- de cada rubro.

Puntos a ser evaluados:

1. Correctitud de la solución computacional - 80%

Funcionalidad	**Valor**	**Estudiante**
Crear BD dinámica	10	E1
Registrar un estudiante	8	E2
Generar reporte HTML y css	10	E1
Respaldar en XML	10	E1
Reporte por género (.docx)	10	E2
Gestionar curva	10	E1
Envió de correos para reposición.	10	E2
Aplazados en al menos 2 exámenes (.pdf)	10	E1
Estadística por generación	12	E2
Reporte por sede con buen rendimiento.	10	E2

1. Análisis del código fuente - 10% -verificación no es creado con IA, si se detecta algo, no se califica toda la tarea-
    a. Validaciones
    b. Código documentado según corresponda
    c. Inspección de N bloques de código al azar para ver la programación
    d. Explicación de N secciones del código sea o creado por ese alumno, se asigna nota binaria.
2. Entregar un documento con los siguientes apartados: - 10%

REQUISITO PARA REVISAR EL PROYECTO

El requisito consiste en presentar la documentación del proyecto indicada en esta sección.

La nota de la documentación del proyecto sirve para aceptar o rechazar el proyecto: se revisan los proyectos que cumplan con este requisito en un 90% o más.

Enviar vía Tec Digital, sección EVALUACIONES, en la carpeta TP1, una carpeta comprimida (.rar, .zip, etc.) que contenga las siguientes partes:

- Parte 1: Una carpeta con la Documentación del proyecto (**nombre: documentaciónCódigos.PDF**).

- Portada. (2 p)
    i. Nombre del curso
    ii. Número de semestre y año lectivo
    iii. Nombres de los Estudiantes y números de carné
    v. Número de tarea programada
    vi. Fecha de entrega
    vii. Estatus de la entrega (definido por el responsable de la implementación de la tarea): [Deplorable|Regular|Buena|Muy Buena|Excelente|Superior]
- Índice. (2 p)
- Enunciado del proyecto. Hacer referencia a un archivo adjunto. (1 p)
- Justificación de eliminación de olores (párrafo descriptivo o ejemplos, indique 5) (10 p)
- Conclusiones del trabajo: (30 p)
    - Al menos 7 problemas encontrados y soluciones a los mismos (Plantilla_Bitacora_ResoluciónProb.xls). 15 p
    - Aprendizajes obtenidos.
        - Debe hacer un listado de todas las lecciones aprendidas producto del desarrollo de la tarea programada. Las lecciones aprendidas deben ser 8 de carácter personal y 8 de carácter técnico. Es decir 4 aportados por cada uno según corresponda. 15 p
- Lista de validaciones estratégicas. 10 p
- Reglamento de trabajo (5 p)
- 2 Agendas (5 p), 2 Minutas (5 p), y evidencias de asignación de responsabilidades: cronograma - (10 p).

- Estadística de tiempos (20 p): un cuadro que muestre el detalle de las actividades que realizó y las horas invertidas en cada una de ellas. La estadística permite medir el esfuerzo dedicado al trabajo en términos de actividades y tiempos, lo cual puede ser una base para calcular el esfuerzo requerido en futuros trabajos. No olvide investigar sobre el Personal Software Process (PSP) y sus implicaciones como programador.

  Ejemplos de actividades:

Actividad Realizada	Horas
Análisis de requerimientos	
Diseño de algoritmos	
Investigación de ...	
Programación	
Documentación interna	
Pruebas	
Elaboración de documentación del proyecto	
Etc.	
TOTAL	

- Parte 2: Una carpeta con el Programa fuente.

Condiciones generales:

Esta tarea programada se rige por las siguientes condiciones:
La tarea debe solucionarse usando listas de listas o no se revisa la tarea.

1. El desarrollo de la tarea es estrictamente en grupos de 2 estudiantes, si hay cambio de alguna pareja debe notificarse antes de mañana a las 12md al correo de la profesora lsarmiento@tec.ac.cr
2. La tarea NO DEBE implementarse con interfaz gráfica, diccionarios u IA´s.
3. Debe cumplir con todo lo indicado en la sección “Puntos a ser evaluados”
4. Deberá entregarse en tiempo y forma según el plazo establecido por el profesor al momento la lectura de este documento.
5. El lenguaje de programación a utilizar es Python v3.5.1 o superior.
6. Debe crear programación iterativa para dar solución a esta tarea.
7. Se cuenta con 3 semanas a partir del día de entrega de la tarea.
    1. Fecha de entrega al TEC Digital: viernes 2 de mayo del 2025, antes de las 11:45 pm. Sólo 1 de la pareja entrega y suban versiones previas de precaución.
8. Se coordinará día y hora para la revisión, debe presentarse el grupo completo a defender la tarea, en caso de no asistir, tendrá nota de 0 en el valor total de la tarea.
9. La tarea debe ser de su propia autoría o no se califica. **Se evaluará partes de la misma para ver si alguna IA influyó, si se comprobara ello, la tarea tendrá nota de 0 para ambos integrantes.**
10. Cada miembro debe realizar a conciencia la evaluación de Habilidades Blandas y entregarla en digital antes de la revisión de la tarea programada.
11. La nota final de cada estudiante se asignará por distribución de esfuerzo y cumplimiento en función a las responsabilidades y la nota obtenida.

Nota: El incumplimiento de alguna condición implicará una calificación de cero.