

## Tarea Programada #2

### 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 y secuencias.
- 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.
- Desarrollar estrategias de investigación y uso del idioma inglés según corresponda.
- Implementar las buenas prácticas de “código limpio” y eliminación de “olores de software”

### 1. 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.

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](#).

## II. Interfaz gráfica (tkinter)

Las aplicaciones para los usuarios finales, son más atractivas e intuitivas si se cuenta con una interfaz gráfica (GUI), es por ello que los lenguajes de programación proveen herramientas



para agilizar el proceso.

Python en nuestro caso, cuenta con el módulo [Tkinter](#), de tal manera que nos dota de un conjunto de librerías para el desarrollo de Interfaces de usuario, por ejemplo: ventanas, botones, etiquetas y cajas de texto, entre otros.

Algunos vínculos adicionales en los cuales puede encontrar información de algunas otras librerías para manejo del GUI en Python son:

- <http://insights.dice.com/2014/11/26/5-top-python-guis-for-2015/>
- <http://python-guide-pt-br.readthedocs.io/en/latest/scenarios/gui/>

## III. Algunos controles de la Interfaz Gráfica (GUI) .

Por lo general los formularios cuentan con elementos que permiten solicitar datos y mostrar la información. Un buen diseño de los mismos luego de comprender lo requerimientos facilitan el funcionamiento del software y permiten la satisfacción del cliente.

Los controles generales son:

Nombre del control	Funcionalidad	Ejemplo
Caja de Texto	Permite el ingreso de un texto corto.	Nombre: <input type="text"/>
Área de Texto	Permite el ingreso de un texto de más extensión. Máximo 255 caracteres.	Descripción del producto <input type="text"/>
Botones de radio	Corresponde a la selección de un criterio único, por ende es excluyente. Alguno obligatoriamente debe estar seleccionado.	Tipo de cliente: <input type="radio"/> Empresa <input checked="" type="radio"/> Particular

Caja de Chequeo	Corresponde a la selección de ningún criterio o toda la cantidad de opciones que el usuario desee.	<input checked="" type="checkbox"/> Usar formato de campo <input type="checkbox"/> Coincidencia exacta <input type="checkbox"/> Buscar hacia atrás <input type="checkbox"/> Desde el principio
Caja de Selección	El usuario debe seleccionar un valor, de los valores desplegados en la lista.	País de Origen <input type="text"/>
Botones	Corresponde a las acciones que deben realizarse.	Ingresar Limpiar

#### IV. SCRUM (Metodología de desarrollo ágil)

Para realizar todas las etapas del ciclo de vida del desarrollo de Software, una forma de administrar proyectos. Hoy día, las empresas siguen procesos más simples y más fluidos, realizando menos documentación y haciendo cada integrante el trabajo por interés y afinidad a las tareas.



Te recomendamos:

1. Ver el video: [¿Qué es la metodología SCRUM?, hasta el minuto 2:40](#)
2. **Certificarte gratis** (Scrum Study) en Scrum algún día para presentación de tu currículum: <https://www.scrumstudy.com>. Esto será una carta de presentación adicional para buscar empleo.
3. Escuchar el audio: [SCRUM - Marco de trabajo para equipos ágiles, ten paciencia, empieza en el minuto 0:28 segundos.](#)

#### V. SMTP (Envío de correos)

El correo electrónico es considerado el servicio más utilizado de Internet. Por lo tanto, la serie de protocolos TCP/IP ofrece una gama de protocolos que permiten una fácil administración del enrutamiento del correo electrónico a través de la red.



El **protocolo SMTP** (*Protocolo simple de transferencia de correo*) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto. Éste es un protocolo que funciona en línea, encapsulado en una trama TCP/IP. El correo se envía directamente al servidor de correo del destinatario.

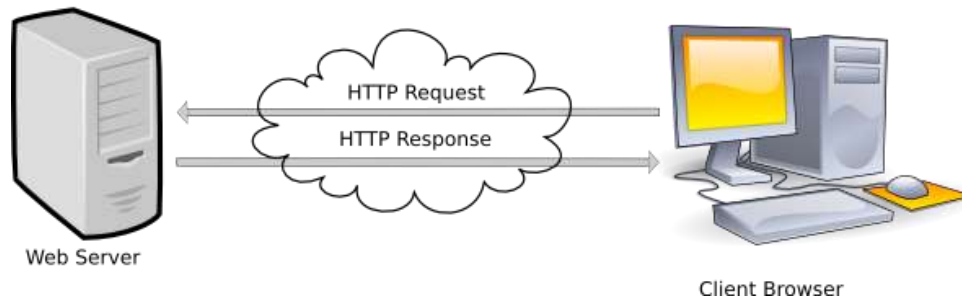
Para mayor información con respecto a este tema puede acceder a los siguientes recursos:

- <https://www.chilkatsoft.com/python-email.asp>
- <https://support.google.com/a/answer/176600?hl=es>

**PONTE TUS PROPIOS LÍMITES, NO DEJES QUE NADIE LO HAGA POR TI...**

## VI . HTTP Request/Response

HTTP (en español *protocolo de transferencia de hipertexto*) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force. Está basado en el modelo solicitud / respuesta, de modo que hay dos tipos de mensajes HTTP: la solicitud y la respuesta. Es uno de los métodos más básicos que las computadoras usan para comunicarse entre sí, donde una máquina envía una solicitud (request) de datos y otra máquina responde (response) la solicitud. Navegar por una página web es un ejemplo de comunicación request/response.



Para mayor información con respecto a este tema puede acceder a los siguientes recursos:

- <https://docs.python.org/3/library/http.client.html>
- [http://www.tutorialspoint.com/http/http\\_tutorial.pdf](http://www.tutorialspoint.com/http/http_tutorial.pdf)

## VII . 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, mediante), 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

```
1  <!DOCTYPE html>
2
3  <html lang="es">
4
5  <head>
6  <title>Titulo de la web</title>
7  <meta charset="utf-8" />
8  <link rel="stylesheet" href="estilos.css" />
9  <link rel="shortcut icon" href="/favicon.ico" />
10 <link rel="alternate" title="Pozolería RSS" type="applicat
11 </head>
12
13 <body>
14   <header>
15     <h1>Mi sitio web</h1>
16     <p>Mi sitio web creado en html5</p>
17   </header>
18   <section>
19     <article>
20       <h2>Titilo de contenido</h2>
21       <p>Contenido (ademas de imagenes, citas, vider
22     </article>
23   </section>
24   <aside>
25     <h3>Titulo de contenido</h3>
26     <p>contenido</p>
27   </aside>
28   <footer>
29     Creado por mi el 2011
30   </footer>
31 </body>
32 </html>
```



## 2. Integratec: Padrinos ayudando y dando ánimo a los primer ingreso.

*¿Fuiste tú apadrinado?  
¿Te gustaría ser un mentor?*

***Te contamos al respecto...***

**Integratec** es el ejemplo de un equipo de voluntariado que ya se desenvuelve en todas las universidades y que han tomado al TEC como ejemplo en liderazgo para ajustar sus modelos. Claro está, cada universidad tiene su equipo con diversos nombres.

¿Sabías que en la VIESA se han dado cuenta que ha habido situaciones en la que los chicos han obtenido su empleo gracias a tener un certificado de participación en **Integratec**?, ¡Como lo has oído!... ese título por voluntariado, ha hecho la diferencia para que una transnacional prefiera a un chico(a) de la colección de currículos que tenían para escoger.

Los voluntarios de Integratec son formados por un equipo interdisciplinario de funcionarios, ellos, además los sensibilizan y posteriormente, se les asigna responsabilidades. Es un proceso de voluntariado de un año. ¿Y eso por qué?, porque se forman a partir de agosto, ya en noviembre-diciembre cada mentor contacta a sus ahijados y empieza a recomendarlo, según el nuevo estudiante le consulte y según patrones generales de información que deben estratégicamente compartir.

### ***¿Quién es el mentor?***

Un estudiante TEC capacitado y formado para guiar a un grupo de estudiantes de primer ingreso de una carrera. Les da seguimiento personal y grupal.

### ***¿Quién puede tener mentor?***

El estudiante de primer ingreso que solicite ser apadrinado cuando llena sus datos de primer ingreso. Si no se solicita, no se va a tener.



Síguenos en:

<https://www.tec.ac.cr/programa-integratec>

[https://www.instagram.com/integratec\\_itcr/?hl=es-la](https://www.instagram.com/integratec_itcr/?hl=es-la)

<https://www.facebook.com/integratecCR/>

**PONTE TUS PROPIOS LÍMITES, NO DEJES QUE NADIE LO HAGA POR TI...**



### 3. Por hacer

Implementar una solución computacional hipotética de atención a la generación 2021. Y para ello, cree todo lo solicitado.

Esta tarea incluye interfaz gráfica, para ello se tiene los siguientes botones:

1. Estudiantes por sede.
2. Estudiantes de carrera por sede. (inactivo hasta no tener Estudiantes por sede)
3. Crear mentores. (inactivo hasta no tener Estudiantes por sede)
4. Asignar mentores. (inactivo hasta no tener Creados los mentores)
5. Actualizar estudiante. (inactivo hasta no tener Estudiantes de carrera por sede)
6. Generar reportes.
7. Crear base de datos en Excel.
8. Enviar correo.
9. Eliminar mentor. (Trío de programadores)
10. Salir

#### 1. **Estudiantes por sede.**

Define la cantidad de estudiantes de primer ingreso que entran por sede.

Predeterminadamente ya se conocen las 5 sedes TEC:

- 1- CAMPUS TECNOLÓGICO CENTRAL CARTAGO
- 2- CAMPUS TECNOLÓGICO LOCAL SAN CARLOS
- 3- CAMPUS TECNOLÓGICO LOCAL SAN JOSÉ
- 4- CENTRO ACADÉMICO DE ALAJUELA
- 5- CENTRO ACADÉMICO DE LIMÓN

Según la página: <https://www.tec.ac.cr/sedes-recintos>

Obtenga las carreras por sede de: <https://www.tec.ac.cr/carreras> . Cuide no repetir nombres de carreras por sede. Obtenga eso técnicamente con un request.

Usted debe obtener el total de primeros ingresos por sede y determinar una cantidad aleatoria de estudiantes por cada carrera en cada sede. Almacene esa información en **una matriz** con la siguiente forma:

CTCC	...	{carrera1:cantidad1, carrera2:cantidad2}
CTLSC	...	...
CTLSJ	...	...
CAA	400	{"Ingeniería en Computación":225, "Ingeniería en Electrónica":175}
CAL	...	...

**Defina usted la interfaz gráfica de esta ventana.**

Por ejemplo, si el usuario dice al CAA, le ingresan 400, entonces aleatoriamente divida 400 en la cantidad de carreras que tenga la sede, por ejemplo, dado es Alajuela sería:

{"Ingeniería en Computación":225, "Ingeniería en Electrónica":175}

Eso sí, verifique que la suma de estudiantes por carrera, sume el total indicado por el usuario en la sede.

**PONTE TUS PROPIOS LÍMITES, NO DEJES QUE NADIE LO HAGA POR TI...**

## 2. **Estudiantes de carrera por sede.**

Cree dinámicamente la base de datos de todos los admitidos en cada carrera por cada sede, para ello siga las siguientes instrucciones:

- Genere un diccionario de estudiantes:

{**Carné:** [**nombre completo**, **teléfono**, **correo**, **sede**, **carrera**, **carnéDeMentor**]}

Según las situaciones indicadas:

### **Carné:**

Generado por 202100####, significa: año, sede y aleatorio (4 números).

Sede:

01- **CAMPUS TECNOLÓGICO CENTRAL** CARTAGO

02- **CAMPUS TECNOLÓGICO LOCAL** SAN CARLOS

03- **CAMPUS TECNOLÓGICO LOCAL** SAN JOSÉ

04- **CENTRO ACADÉMICO DE** ALAJUELA

05- **CENTRO ACADÉMICO DE** LIMÓN

Verifique sea único a la hora de crearlo.

### **Nombre completo:**

Genere los nombres formados por nombre y 2 apellidos usando una librería de Python.

### **Teléfono:**

Validar que comience con 8, 6, 7 o 9, seguido de 7 dígitos aleatorios.

No puede haber duplicados.

### **Correo Electrónico:**

Fórmelo por: primera letra del nombre + primer apellido + @estudiantec.cr

Ejemplo: Juan Pérez: [jperez@estudiantec.cr](mailto:jperez@estudiantec.cr)

Validar que no esté repetido, en caso de estar repetidor usar la siguiente letra. Ej:

[juperez@estudiantec.cr](mailto:juperez@estudiantec.cr)

### **Sede:**

Determine aleatoriamente a qué sede, según las 5 sedes del Tec.

### **Carrera:**

Dado la sede aleatoria determinada para este estudiante, determina a qué carrera pertenece.

### **CarnéDeMentor:**

Posteriormente se asignará, por ahora asigne el valor inicial de 0.

Analice en equipo de ingenieros: ¿Cuál será la mejor estrategia para ir creando los admitidos por carrera de cada sede?

### 3. Crear mentores.

Dada la cantidad de estudiantes de primer ingreso generados por sede en cada carrera, crear un 5% de mentores en esa carrera es esa sede. Sólo créelos, posteriormente habrá otra funcionalidad para la asignación respectiva.

Utilice el mismo algoritmo que permitió crear estudiantes de primer ingreso, con sólo: carné, nombre completo, carrera y correo.

Verifique el carnet y el correo no se repite en estudiantes ni en mentores, a la hora de crearlos.

Para no perder los datos, guárdelos en un archivo binario de mentores usando **una matriz** con un diccionario separado por sedes.

Ejemplo:

CTCC	{ <b>Carné:</b> [nombre completo, carrera, correo], <b>Carné:</b> [nombre completo, carrera, correo],...}
CTLSC	...
CTLSJ	...
CAA	...
CAL	...

### 4. Asignar mentores.

Según la cantidad de mentores, asignar a los estudiantes con misma carrera y sede de forma distribuida a los mentores. Agregar el número de carné del mentor en el estudiante.

### 5. Actualizar estudiante

Pida el carné (si no lo encuentra mostrar mensaje de error) y muestra los datos en una ventana de formulario, donde sea posible modificar: **nombre completo**, **teléfono**, **correo**. Guardar los cambios luego de verificar su formato y la restricción de no duplicidad. Usted es libre de crear la ventana con la apariencia que guste. Debería poder actualizar un estudiante de primer ingreso o mentor, eso sí, al aparecer sus datos, indique el tipo de estudiante que es.

### 6. Generar reportes.

Este botón abre una nueva ventana con 3 opciones de reportes (6 en caso del trío de programación), cada reporte genera un archivo HTML. Los botones de esta ventana:

- Reporte por sede.
- Reporte por carrera.
- Reporte por Mentor.
- Reporte de estudiantes por sede no atendidos.
- Reporte de estudiantes por sede atendidos.
- Lista de mentores de todas las sedes.

Para ello:

- **Reporte por sede:** para cada sede muestra el nombre carrera y datos completos de los estudiantes. Haga una tabla independiente por sede.
- **Reporte por carrera:** pedir la carrera y generar un html, debe mostrarse tipo lista numerada, no definida; todos los datos del estudiante y la sede a la que pertenece. Muestre los datos ordenados por sede.
- **Reporte por Mentor:** en orden de sede muestre: la sede, el nombre del mentor y la información de los estudiantes asignados como lista numerada.
- **Reporte de estudiantes por sede no atendidos:** Solicita la sede y devuelve un html que indica carrera y datos completos de los estudiantes que no poseen mentor. Agrúpelos por carrera. Al finalizar indique el total de estudiantes que quedaron sin atención de un mentor.
- **Reporte de estudiantes por sede atendidos:** Solicita la sede y devuelve un html que indica carrera y datos completos de los estudiantes que poseen mentor. Agrúpelos por carrera. Indique además el nombre del mentor para cada estudiante. Al finalizar indique el total de estudiantes que son atendidos por un mentor.
- **Lista de mentores de todas las sedes:** Muestre por sede y por carrera, todos los datos de los mentores. Al final indique por cada carrera sin importar la sede el nombre de la carrera y el total general de mentores para esa carrera. El diseño de esta página HTML ustedes lo definen.

## 7. Crear base de datos en Excel

Cree un archivo string a guardar en disco duro con extensión “.csv”

Nombre del archivo estará formado por: BDIntegraTEC + fecha y hora del sistema + “.csv”

Ejemplo: bdIntegraTEC20-10-2020\_19-40.csv

Allí se integra la información de la siguiente manera:

Sede, Carrera, Carne, Nombre, Correo, teléfono, Booleano (True: estudiante, False: mentor).

## 8. Enviar correo

Solicita el correo de un encargado y adjunta el .csv

Debe validar el correo contra una expresión regular, el formato: texto@texto.texto

## 9. Eliminar mentor.

Cree una ventana con un selector en donde pueda escoger el mentor (carné-nombre completo) a eliminar y un botón de eliminar.

Valide:

- Si el mentor tiene estudiantes asignados, debe ir a eliminar primero el mentor a cada estudiante asignado, posteriormente eliminar el mentor.
- Si no tiene estudiantes asignados, sólo se borra él.

La próxima vez, al volver a llamar esta funcionalidad, ya ese mentor no debe aparecer en la lista de mentores.

**PONTE TUS PROPIOS LÍMITES, NO DEJES QUE NADIE LO HAGA POR TI...**

### Puntos para evaluar:

1. Correctitud de la solución computacional - 80%

Funcionalidad	Procesos	Valor(2)	Valor(3)				
Estudiantes por sede.	Según solicitud.	9	7				
Estudiantes de carrera por sede.	Inactivo hasta no tener Estudiantes por sede. Según solicitud.	9	7				
Crear mentores.	Inactivo hasta no tener Estudiantes por sede. Según solicitud.	9	7				
Asignar mentores.	Inactivo hasta no tener Creados los mentores. Según solicitud.	9	7				
Actualizar estudiante	Inactivo hasta no tener Estudiantes de carrera por sede. Según solicitud.	9	6				
Generar reportes.	<table border="1"> <tr> <td>Duos (10 cada reporte)</td> <td>Tríos (7 cada uno)</td> </tr> <tr> <td> <ul style="list-style-type: none"> <li>- Reporte por sede.</li> <li>- Reporte por carrera.</li> <li>- Reporte por Mentor.</li> </ul> </td> <td> <ul style="list-style-type: none"> <li>- Reporte por sede.</li> <li>- Reporte por carrera.</li> <li>- Reporte por Mentor.</li> <li>- Reporte de estudiantes por sede no atendidos.</li> <li>- Reporte de estudiantes por sede atendidos.</li> <li>- Lista de mentores de todas las sedes.</li> </ul> </td> </tr> </table> <p>Según solicitud.</p>	Duos (10 cada reporte)	Tríos (7 cada uno)	<ul style="list-style-type: none"> <li>- Reporte por sede.</li> <li>- Reporte por carrera.</li> <li>- Reporte por Mentor.</li> </ul>	<ul style="list-style-type: none"> <li>- Reporte por sede.</li> <li>- Reporte por carrera.</li> <li>- Reporte por Mentor.</li> <li>- Reporte de estudiantes por sede no atendidos.</li> <li>- Reporte de estudiantes por sede atendidos.</li> <li>- Lista de mentores de todas las sedes.</li> </ul>	30	42
Duos (10 cada reporte)	Tríos (7 cada uno)						
<ul style="list-style-type: none"> <li>- Reporte por sede.</li> <li>- Reporte por carrera.</li> <li>- Reporte por Mentor.</li> </ul>	<ul style="list-style-type: none"> <li>- Reporte por sede.</li> <li>- Reporte por carrera.</li> <li>- Reporte por Mentor.</li> <li>- Reporte de estudiantes por sede no atendidos.</li> <li>- Reporte de estudiantes por sede atendidos.</li> <li>- Lista de mentores de todas las sedes.</li> </ul>						
Crear base de datos en Excel.	Según solicitud.	9	6				
Enviar correo	Según solicitud.	9	6				
Eliminar mentor(3)	Inactivo hasta que se creen los mentores. Según solicitud.	0	6				
Interfaz Gráfica	Ventanas comunes	7	6				

2. Olores de software y buenas prácticas en programación - 5%
3. Robustez de la solución computacional (validaciones) - 5%
4. Entregar un documento con los siguientes apartados: - 10%

## Documentación

Como motivo de promover las metodologías ágiles, se implementara Scrum para la documentación del proyecto. Pero no olviden entregar la evaluación de Habilidades Blandas.

### Trabajo en grupo:

- Deben crear un archivo tipo documento de Word en el siguiente enlace: <https://drive.google.com/drive/folders/1ScUO1K0ZEVAQbhSsmKi8j4WWpmEOZ2Xn?usp=sharing> con el nombre "NombrePareja1-NombrePareja2Scrum" **USAR LOS USUARIOS PROPIOS DE DRIVE, NO ANONIMOS**
- Al inicio del archivo debe poseer un encabezado como el siguiente:

Fecha Inicial: ---\*

Grupo de Trabajo: Nombre persona1 y Nombre persona2

Scrum Máster: Nombre del Scrum Máster

- Después, según la metodología Scrum se requiere de reuniones diarias, lo cual se simulará mediante entradas en el archivo según el siguiente formato:

Fecha:	
Nombre de la persona 1:	
¿Qué he hecho?	
¿Qué haré a continuación?	
¿En qué ocupo ayuda?	
Fecha:	
Nombre de la persona 2:	
¿Qué he hecho?	
¿Qué haré a continuación?	
¿En qué ocupo ayuda?	

- Se espera que sean 3 entradas semanales, por lo que a la entrega del proyecto se esperan un **MINIMO** de 18 entradas en total (9 por miembro de la pareja).
- Se espera que al menos 4 de las 9 entradas individuales se ingrese algún ejemplo o evidencia del avance (parte de código, información investigada con links o referencias, entre otros)

RECUERDE: Los archivos en el drive mantienen un historial de lo hecho, por lo que deben de llevar las entradas con tiempo y no realizar todos unos días antes de la entrega.

**EL INCUMPLIMIENTO DE LO ANTERIOR IMPIDE LA REVISION DE LA TAREA PROGRAMADA Y OCACIONA UN 0 AUTOMATICAMENTE EN LA NOTA.**

**PONTE TUS PROPIOS LÍMITES, NO DEJES QUE NADIE LO HAGA POR TI...**

## Condiciones generales:

Esta tarea programada se rige por las siguientes condiciones:

1. La tarea debe solucionarse usando listas o diccionarios o no se revisa la tarea.
2. El desarrollo de la tarea es estrictamente en grupos de 2 estudiantes, si hay cambio de alguna pareja debe notificarse antes de lunes a las 12md al correo de la profesora lsarmiento@tec.ac.cr
3. La tarea DEBE implementarse con interfaz gráfica.
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: **Sábado 5 de diciembre del 2020, antes de las 11:45 pm.**
1. 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.
2. Cada miembro debe realizar a conciencia la evaluación de Habilidades Blandas.

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