

## Tarea Programada #3

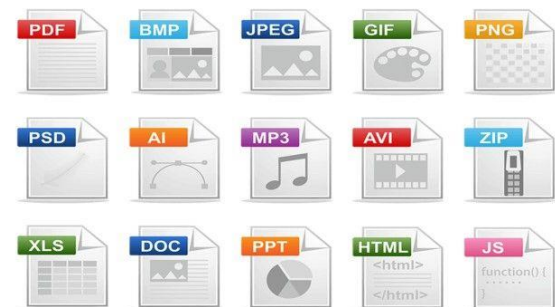
### 1. Objetivos

- 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, secuencias y en especial **Orientación a Objetos**.
- 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.

### 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.

Con ello, usted puede crear archivos de texto de la extensión que guste, por ejemplo: **.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](#).

## 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:

- <https://acortar.link/uyPCh2>
- <https://www.pysimplegui.org/en/latest/>

## 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 los 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.	<input type="button" value="Ingresar"/> <input type="button" value="Limpiar"/>

#### IV. 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, video
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>
```

## V. XML

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



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/>

Universidad Nacional Mayor de San Marcos  
IEEE Student Branch  
A Student Chapter of the IEEE Communications Society

## ¿QUÉ ES XML?

**XML** Extensible Markup Language  
o  
Lenguaje de Marcado Extensible

Es un subconjunto de SGML (Estándar Generalised Mark-up Language), simplificado y adaptado a Internet

Es un tipo de lenguaje de marcado o conjunto de códigos (denominados etiquetas) que definen la estructura y el significado de los datos.

HTML XML WML SGML

ieeee.comsoc.unmsm

Universidad Nacional Mayor de San Marcos  
IEEE Student Branch  
A Student Chapter of the IEEE Communications Society

## ¿PARA QUÉ SIRVE EL XML?

Puede utilizarse varias veces de distintas maneras en diferentes sistemas, independientemente de la plataforma o sistema operativo hardware que se esté usando.

**Facilita el intercambio de datos**

**Separa los datos de HTML**

**Simplifica el cambio a una nueva aplicación**

ieeee.comsoc.unmsm

Universidad Nacional Mayor de San Marcos  
IEEE Student Branch  
A Student Chapter of the IEEE Communications Society

## ESTRUCTURA XML

**Nombre del atributo** **Contenido del elemento**

`<autor país = "España"> Jose Ramon </autor>`

**Nombre del elemento** **Etiqueta de fin**

**Valor del atributo**

ieeee.comsoc.unmsm

Universidad Nacional Mayor de San Marcos  
IEEE Student Branch  
A Student Chapter of the IEEE Communications Society

## COMPONENTES XML

**Elementos**  
PIEZA LÓGICA DEL MARCADO. SE REPRESENTA CON UNA CADENA DE TEXTOS (DATOS) ENCERRADA ENTRE ETIQUETAS. PUEDEN EXISTIR ELEMENTOS VACÍOS (<BR/>). LOS ELEMENTOS PUEDEN CONTER ATRIBUTOS.

**Instrucciones**  
ORDENES ESPECIALES PARA SER UTILIZADAS POR LA APLICACIÓN QUE PROCESA  
<?XML-stylesheet TYPE="text/css" href="estilo.css">

**Comentarios**  
INFORMACIÓN QUE NO FORMA PARTE DEL DOCUMENTO. COMIENZAN POR <!-- Y TERMINAN POR -->.

**Declaraciones de tipo**  
ESPECIFICAN INFORMACIÓN ACERCA DEL DOCUMENTO  
<?DOCTYPE PERSONA SYSTEM "PERSONA.DTD">

**Secciones CDATA**  
SE TRATA DE UN CONJUNTO DE CARACTERES QUE NO DEBEN SER INTERPRETADOS POR EL PROCESADOR.  
<![CDATA[ AQUÍ SE PUEDE METER CUALQUIER CARÁCTER, COMO <, &, ", ... SIN QUE SEAN INTERPRETADOS COMO MARCACIÓN ]>

ieeee.comsoc.unmsm



## V. SCRUM (Metodología de desarrollo ágil)

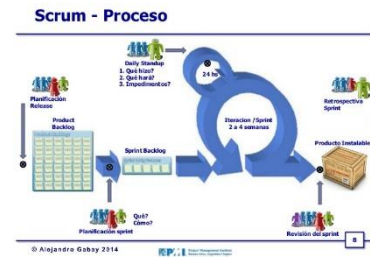
Para realizar todas las etapas del ciclo de vida del desarrollo de Software, una forma de administrar proyectos. Hoy día, las empresas siguen procesos más simples y más fluidos, realizando menos documentación y haciendo cada integrante el trabajo por interés y afinidad a las tareas.

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 currículo: <https://www.scrumstudy.com>. Esto será una carta de presentación adicional para buscar empleo.
  - a. [¿Por qué usar Scrum?](#)
  - b. [Desarrollo iterativo y roles](#)
  - c. [Tiempo limitado](#)
3. Escuchar el audio: [SCRUM - Marco de trabajo para equipos ágiles, ten paciencia, empieza en el minuto 0:28 segundos.](#)
4. Analice el siguiente cuadro resumen:

How the Project is Managed

|                         | Scrum                         | Traditional         |
|-------------------------|-------------------------------|---------------------|
| Emphasis                | People                        | Process             |
| Size                    | Small to large                | Large               |
| Domain                  | Unpredictable/<br>Exploratory | Predictable         |
| Documentation           | Minimal                       | Comprehensive       |
| Process style           | Iterative                     | Linear              |
| Upfront Planning        | Low                           | High                |
| Perspective to Change   | Adaptability                  | Sustainability      |
| Management Style        | Decentralized                 | Autocratic          |
| Leadership              | Collaborative                 | Command and control |
| Performance Measurement | Business value                | Plan conformity     |
| Returns on Investment   | Early/throughout project life | End of project life |



## Sistema del Tribunal Supremos de Elecciones:

Ayude al Tribunal Supremo de Elecciones a registrar y reportar los nacimientos y parentescos de una persona, para ello implemente los siguientes requisitos:

1. El software debe poseer una interfaz gráfica idéntica a:
  - a. Ventana de registro:



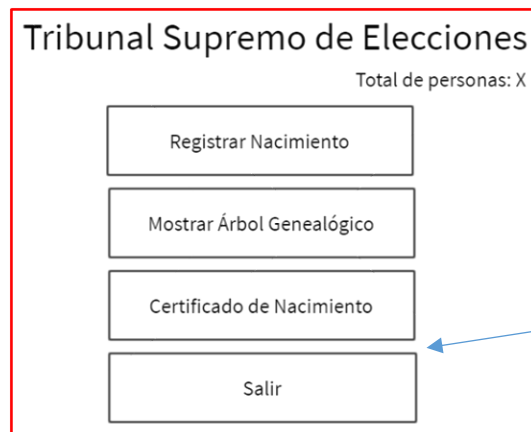
Diagrama de la ventana de registro:

- Usuario:
- Contraseña:
- 

Es una ventana en dónde se ingresa en una caja de texto: Usuario y Contraseña, se tienen los botones de Limpiar e Ingresar. ***Se verifica en un archivo de texto*** si el usuario está autorizado.

| Item       | Validaciones  |
|------------|---|
| Usuario    | Debe poseer el formato de un correo electrónico   |
| Contraseña | <ul style="list-style-type: none"><li>- Debe tener un al menos un número</li><li>- Debe tener al menos una letra</li><li>- Mayor o igual a 5 caracteres</li><li>- Menor o igual a 15 caracteres</li></ul> |

Usted es libre de diseñar el archivo del texto con el formato que usted desee para verificar la información de acceso.

**b. Ventana principal:**


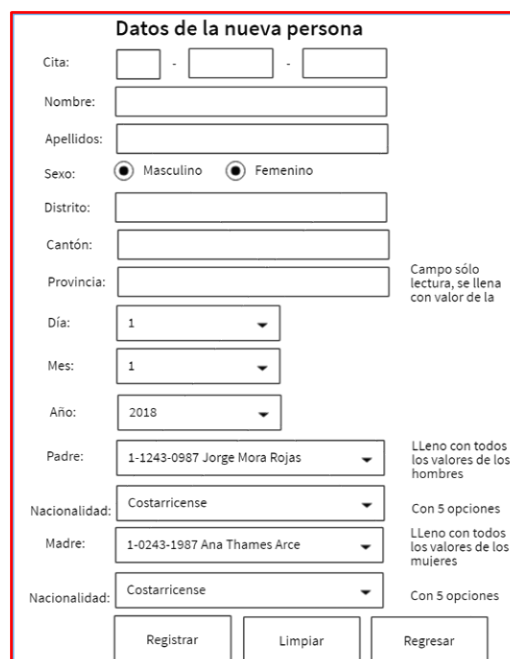
Adicionar:  
**Reportes**

Posee el título: “Tribunal Supremo de Elecciones”

Posee un subtítulo que muestra el contador de las personas registradas: “Total de personas: X” (x representa la cantidad de objetos “persona” de la lista binaria)

Posee los siguientes botones:

- Registrar Nacimiento
- Mostrar árbol genealógico
- Certificado de Nacimiento
- Reportes
- Salir

**c. Registro de Nacimiento:**


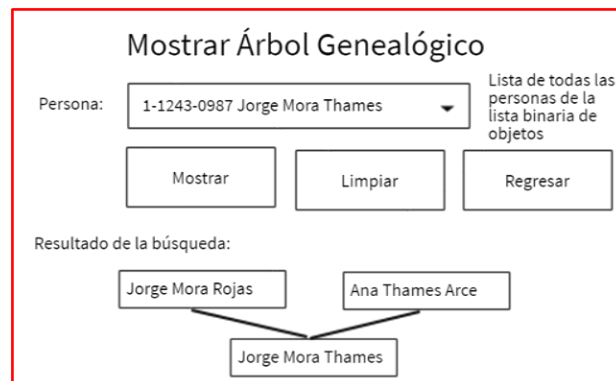
Regresar: cierra la ventana y aparece el menú principal.

- Note la calidad de la columna visual y tamaños.
- El sexo debe ser una respuesta excluyente.

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

- Debe actualizar a memoria secundaria el archivo binario, correspondiente a una lista de objetos.
- Defina su propia estrategia para tener 5 hombres y 5 mujeres previas que pueden ser los padres originales.

*d. Mostrar Árbol Genealógico:*



Siempre debe mostrar sus padres, según la información que tenga registrada.

## 2. Certificado de Nacimiento:

Cree una ventana en dónde puede digitar un número de cédula, valide el ingreso del dato contra una *expresión regular*. Si la cédula no existe, notifique al usuario. En caso de existir, realice lo siguiente:

- Crea 2 archivos:
  - un archivo con el formato de salida corresponde a un archivo .html
  - un archivo con el formato de salida correspondiente a un archivo XML
- Formato del archivo HTML debe poseer la siguiente estructura visual idéntica para el Certificado:

| Certificado de Nacimiento |                          |
|---------------------------|--------------------------|
| <b>Al tomo:</b>           | 0234                     |
| <b>Asiento:</b>           | 0987                     |
| <b>Cita:</b>              | 3-0234-0987              |
| <b>Dice que:</b>          | Juan Pérez Díaz          |
| <b>Sexo:</b>              | Masculino                |
| <b>Nació en:</b>          | Oriental Central Cartago |
| <b>El día:</b>            | 10/12/1990               |
| <b>Padre:</b>             | Jorge Pérez Fuentes      |
| <b>Nacionalidad:</b>      | Costarricense            |
| <b>Madre:</b>             | Ana Díaz Mora            |
| <b>Nacionalidad:</b>      | Costarricense            |

- Adicional a ello, cree el archivo XML según corresponda.

Los archivos se deben guardar en el directorio actual.



Esta opción permite comprar certificaciones de nacimiento en formato digital

Comprar

₡ 1,575

Puede observar la referencia generada de:

<https://www.consulta.tse.go.cr/appcdi#/productos>, seleccione

“Comprar” en:

- Determine el criterio de búsqueda.
- Coloque el dato.
- De clic en “Buscar persona”
- Observe sus datos en la parte inferior al dar click en la

opción: “Ver Detalle”

## Reportes

Cree una ventana con las siguientes opciones, todos deben crear archivos.html:

1. Lista de personas por provincia.
2. Lista de personas nacidos a partir de...
3. Lista de personas del mismo hospital.

### - Lista de personas por provincia:

Debe mostrar una ventana con una caja de selección, allí despliega el nombre de las 7 provincias de Costa Rica. El usuario determina la provincia que desea, se activa el botón de **Crear**. **Regresar** debe estar siempre activo (cierra la ventana actual y regresa a la ventana principal).

**Crear**: Notifica de archivo creado satisfactoriamente y genera un HTML, usted es libre de mostrar la información dentro de una tabla con el formato que usted desee. No olvide el título. Debe mostrar la siguiente información: cédula, nombre completo, fecha de nacimiento, sexo. Muestre **primero las mujeres** y luego los hombres.

### - Lista de personas nacidos a partir de:

Debe mostrar una ventana con una caja de texto, debe indicarse un valor entero de 4 dígitos (valide contra expresión regular), al llenarse correctamente, activa el botón de **Crear**. **Regresar** debe estar siempre activo.

**Crear**: Notifica de archivo creado satisfactoriamente y genera un HTML, usted es libre de mostrar la información dentro de una tabla con el formato que usted desee. Debe mostrar la siguiente información: cédula, nombre completo, fecha de nacimiento, sexo, nombre de la madre y del padre. Muestre la información **agrupados por año de nacimiento**. No importa el orden.

### - Lista de personas del mismo hospital:

Recorra toda la lista de objetos y extraiga el tomo de forma única, sin repetir de las personas registradas en la base de datos. Pues ello representa el libro de actas de cada uno de los hospitales del país.

Cree una ventana gráfica que muestre el título de la funcionalidad y una lista de selección que muestra todos los "tomos" únicos, junto al título: "Hospitales".

Adicional activa el botón de **Crear**. **Regresar** debe estar siempre activo.

**Crear**: Notifica de archivo creado satisfactoriamente y genera un HTML, usted es libre de mostrar la información dentro de una tabla con el formato que usted desee. Debe mostrar la siguiente información: cédula, nombre completo, fecha de nacimiento, sexo, nombre de la madre y del padre. Muestre **primero los hombres** y luego las mujeres.



Los **requerimientos no funcionales** son:

- Una GUI puede crearse mediante alguna librería complementaria de Python. Ej.: tkinter (Si desea puede investigar alguna otra librería o sistema para GUI's)
- La lógica del programa debe ser en el lenguaje PYTHON
- El problema debe resolverse con una lista binaria de Objetos.
- La clase debe respetar la siguiente estructura:

| <i>Atributos</i>   | <i>Tipo</i>                    |
|--------------------|--------------------------------|
| Cédula             | Strings porque incluye guiones |
| Nombre completo    | strings                        |
| Sexo               | Letra: M o F, string           |
| Ubicación          | Strings                        |
| Fecha nacimiento   | Strings                        |
| Padre              | Strings                        |
| Nacionalidad Padre | Strings                        |
| Madre              | Strings                        |
| Nacionalidad Madre | Strings                        |

Para instanciar se deben llamar todos los datos, observe el sistema "imaginario" únicamente crea la funcionalidad de registrar, no tiene, modificar ni eliminar.

- Se espera que el grupo de trabajo deduzca todas las validaciones necesarias para el correcto funcionamiento del programa, así como un uso fácil e intuitivo.

### 3. Por hacer

Implementar una solución computacional que permita al usuario hacer uso de las funcionalidades solicitadas mediante el concepto de orientación a objetos.

#### Puntos a ser evaluados:

1. Correctitud de la solución computacional - 80%

| Requerimiento   | Valor | Responsable |
|---|-------|-------------|
| Carga estructura binaria en aplicación al abrirla     |       |             |
| Ventana de registro y lee de archivo de texto.        |       |             |
| Ventana principal                                     |       |             |
| Ventana de reportes                                   |       |             |
| Registro de nacimiento                                |       |             |
| Mostrar árbol genealógico                             |       |             |
| Certificado de nacimiento                             |       |             |
| Archivo XML   |       |             |
| Lista de personas por provincia.                      |       |             |
| Lista de personas nacidos a partir de...              |       |             |
| Lista de personas del mismo hospital.                 |       |             |
| Definición de Clase (requerido para revisar la tarea) |       |             |
| Lista Binaria de Objetos                              |       |             |

2. Robustez de la solución computacional (validaciones y manejo de excepciones) - 10%

3. Entregar un documento con al menos los siguientes apartados: - 10%

#### **Apartado #1**

- Siga las instrucciones según Esteban Leiva le indique.

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

- 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.

## **Apartado #2**

Día de entrega debe entregar junto al código .py un documento word con la siguiente información:

Estadística de tiempos: 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 |
|---------------------|-------|
|---------------------|-------|

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

|   |  |
|---|--|
| Análisis de requerimientos                |  |
| Diseño de algoritmos                      |  |
| Investigación de ...                      |  |
| Programación                              |  |
| Documentación interna                     |  |
| Pruebas                                   |  |
| Elaboración de documentación del proyecto |  |
| Etc.                                      |  |
| <b>TOTAL</b>                              |  |

### Condiciones generales:

Esta tarea programada se rige por las siguientes condiciones:

1. El desarrollo de la tarea es estrictamente en grupos de 2 estudiantes.
2. 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. **Lunes 20 de noviembre a las 11:45 pm**
5. El lenguaje de programación a utilizar es Python v3
6. Debe crear programación Orientada a Objetos para dar solución a esta tarea.
7. Se cuenta con 3 semanas a partir del día de entrega de la tarea.
8. Debe presentarse el grupo completo a defender la tarea virtualmente, en caso de no asistir algún miembro, tendrá nota de 0 en el valor total de la tarea. Las fechas y horas se coordinan en función a los horarios acordados.
9. El código debe ser código limpio y no debe poseer olores de software.
10. Cada miembro debe realizar a conciencia la evaluación de Habilidades Blandas.

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