

## 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 y orientación a objetos (objetivo de esta tarea), estructuras condicionales, funciones y estructuras de datos (listas, matrices y diccionarios).
- 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](#).

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

## 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.
  - a. [¿Por qué usar Scrum?](#)
  - b. [Desarrollo iterativo y roles](#)
  - c. [Tiempo limitado](#)
 Extraído de: [Lista de videos](#).
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/Exploratory	Predictable
Documentation	Minimal	Comprehensive
Process flow	Iterative	Linear
Adaptation to change	Low	High
Response to change	Adaptability	Sustainability
Management style	Decentralized	Autocratic
Leadership	Collaborative	Command and control
Performance measurement	Business value	Plan conformity
Return on investment	Early/throughout project life	End of project life

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

## V. Base de datos

Medio de almacenamiento persistente...de una forma más formal sería:

### ¿QUÉ SON LAS BASES DE DATOS?

1. CONJUNTO ORGANIZADO DE INFORMACIÓN SOBRE UN MISMO TEMA.  
Ej. LA LISTA DE ALUMNOS DE UN CURSO Y SUS CALIFICACIONES.
2. QUE SE ALMACENAN COMO UN SISTEMA.  
Ej. COMO TABLAS PARA GUARDAR LA INFORMACIÓN DE LOS ALUMNOS Y SUS CALIFICACIONES.
3. QUE PUENTE CONSULTARLOS CUANDO SE NECESITA.  
Ej. APLICANDO LA LISTA DE ALUMNOS O PARA VERIFICAR SUS CALIFICACIONES.

**MOTOR DE BASES DE DATOS ES EL SOFTWARE PARA ORGANIZAR Y CONSULTAR LA INFORMACIÓN. LA BASE DE DATOS ES SOLO LA INFORMACIÓN.**

HAY DOS TIPOS PRINCIPALES DE MOTORES:

- RELACIONALES (SQL)**  
ORGANIZAN LA INFORMACIÓN EN TABLAS Y ENTRAN QUE LOS DATOS SE GUARDEN.  
*MySQL, PostgreSQL, Oracle, SQL Server*
- NO RELACIONALES (NoSQL)**  
NO USAN TABLAS Y LOS DATOS PUEDEN REPETIRSE EN DIFERENTES LUGARES PARA ACCEDER MÁS RÁPIDO.  
*Cassandra, MongoDB, Redis, Firebase*

EDteam | CONVIÉRTETE EN UN PROFESIONAL DE LAS BASES DE DATOS EN: [EDTEAM / CURSOS](https://ed.team/cursos/sql)

---

### SINTAXIS BÁSICA DE SQL

SQL es un lenguaje de consulta estructurada que permite administrar y recuperar información en la base de datos.

Indica las columnas que se quieren consultar.

Selecciona las tablas.

Define las condiciones para que se cumpla la consulta.

Muestra cómo se agruparán los registros. Se usa Group By cuando existen funciones como COUNT().

Indica qué condición debe cumplirse como resultado. COUNT() cuenta los libros y la regla determina que sean mayores a 3.

Muestra el orden de la consulta.

```

SELECT nombre, autor, genero
FROM libros
WHERE genero = 'Novela negra';

SELECT autor, COUNT(libroID)
FROM libros
WHERE genero = 'Novela negra'
GROUP BY autor
HAVING COUNT(libroID) > 3
ORDER BY nombre ASC;

```

EDteam | Diseña y administra bases de datos SQL en: [ed.team/cursos/sql](https://ed.team/cursos/sql)

### COMANDOS BÁSICOS DE SQL QUE DEBES CONOCER

`SELECT * FROM courses WHERE name = 'SQL'`

- SELECT**: Consulta sobre los datos almacenados en una base de datos.
- INSERT**: Añade nuevos datos a la tabla cuando lo necesites.
- CREATE**: Según lo que requieras, puedes crear una nueva tabla o base de datos.
- DROP**: Elimina definitivamente una base de datos o tabla.
- ALTER**: Haz cambios en tu base de datos, tabla, vista u otros objetos.
- DELETE**: Elimina uno o múltiples registros de una tabla.
- UPDATE**: Modifica los registros de toda una tabla o de campos específicos.

Guarda estos comandos y verás cómo facilitan tu trabajo.

Domina las bases de datos con EDteam en: [ed.team/cursos/sql](https://ed.team/cursos/sql)

---

### Motores de BASE DE DATOS relacionales

- ORACLE**: El motor relacional comercial más antiguo. Su creador, Larry Ellison, estuvo en el comité que definió SQL.
- MICROSOFT SQL SERVER**: Multiplataforma desde 2017. Son líderes en Business Intelligence (integrando más apps en el mismo paquete).
- MYSQL**: Es el motor más usados en la web y preferido por los CMS clásicos que usan PHP como WordPress, Magento, Drupal, etc).
- SQLITE**: Es una base de datos embebida en el programa. Al estar integrada en todos los teléfonos se usa para almacenamiento interno de Apps.
- MARIADB**: Fork derivado de MySQL a partir de su compra por Oracle. Compatible con MySQL para poder cambiar un motor por otro.
- POSTGRESQL**: Inició como un proyecto universitario llamado INGRES, inspirado en Oracle. Usan funciones y triggers que MySQL no tuvo por años.

## VI. 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

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



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, vide
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>
```

## VII . 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/>

#### VIII . PDF

El formato de documento portátil ( PDF ), estandarizado como ISO 32000 , es un formato de archivo desarrollado por Adobe en 1993 para presentar documentos , incluido el formato de texto y las imágenes, de una manera independiente del software de aplicación , el hardware y los sistemas operativos . <sup>[21][3]</sup> Basado en el lenguaje PostScript , cada archivo PDF contiene una descripción completa de un documento plano de diseño fijo, que incluye texto, fuentes , gráficos vectoriales , imágenes rasterizadas y otra información necesaria para mostrarlo. Según: <https://en.wikipedia.org/wiki/PDF>





## 2. Licencias de Conducir de Costa Rica

*¿Ya eres mayor de edad?*

*¿Ya pensaste que para tu currículum es atractivo tener licencia de conducir?*

*¿Ya pensaste la importancia de tener una licencia de conducir?*

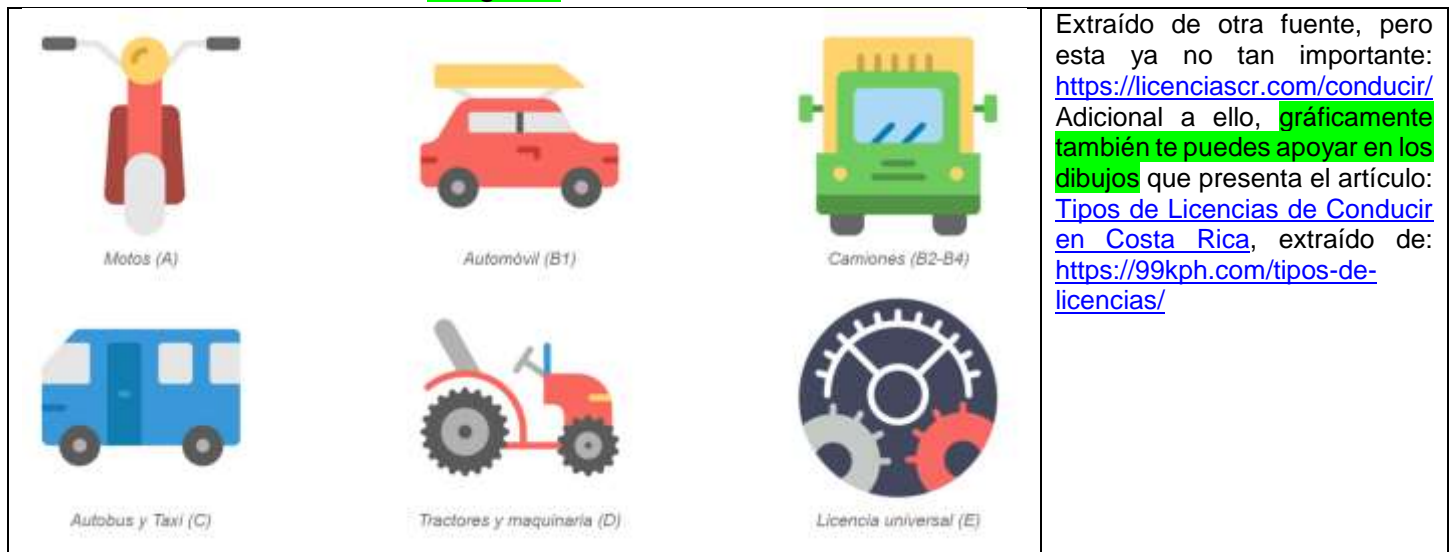
*No importa no tengas carro, tener licencia no implica tener carro, pero si implica más puertas laborales abriéndose.*

**Aprendamos al respecto...**

**¿Sabías los tipos de licencias que hay en tu país Costa Rica?**

Sino, más vale vayas a hacer un **mapa mental** de: [¿Cuáles son los tipos de licencia de conducir en Costa Rica?](https://practicatest.cr/blog/licencias/tipos-licencia-conducir-costa-rica) detallado en: <https://practicatest.cr/blog/licencias/tipos-licencia-conducir-costa-rica>

Pero visto como una **infografía** en:



En el proceso de obtener licencia requiere que conozcas de la: [Dirección General de Educación Vial](https://licenciascr.com/direccion-general-de-educacion-vial/) y que realices un **mapa mental** de: **¿Dónde queda la Dirección General de Educación Vial?**

Esto lo puede aprender en: <https://licenciascr.com/direccion-general-de-educacion-vial-dgev/>

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

### 3. Por hacer

El Consejo de Seguridad Vial (COSEVI) lo ha contratado a usted y a su equipo para *crear un sistema de información* que le permita crear reportes como los citados (tome tiempo para ver el video y los gráficos anunciados allí) en: [Conductores y licencias](https://datosabiertos.csv.go.cr/dashboards/19785/conductores-y-licencias/) a ser observado en: <https://datosabiertos.csv.go.cr/dashboards/19785/conductores-y-licencias/>, pues parte de sus objetivos generales es:

*"Desarrollar una estructura física, tecnológica y humana que promueva la excelencia en el servicio y la satisfacción total del usuario. Promover el uso de tecnología de punta en el desarrollo de la seguridad vial. Diseñar, establecer y mantener programas permanentes de capacitación y actualización, (técnica, administrativa y de cultura organizacional) dirigidos al desarrollo personal del grupo humano que conforma la organización."*

Al abrir la aplicación, **cargue en la RAM de memoria secundaria la lista de objetos si existe.**

Esta tarea incluye interfaz gráfica, para ello se tiene los siguientes botones como **menú principal**:

1. Crear XML
2. Crear licencias
3. Renovar licencias
4. Generar PDF
5. Reportes de Excel
6. Acerca de
7. Salir

#### 1. Crear XML

Extrae de Internet la información de [¿Cuáles son los tipos de licencia de conducir en Costa Rica?](#) y crea un XML que contenga:

- Tipo de licencia, subtipo de licencia, comentario y requisitos.

Pues luego de leer este archivo generado, debe crearse una lista simple con todos los subtipos de licencias extraídas para poder crear posteriormente las licencias.

(Haga impresiones para que en el Shell de Python se pueda ver la evolución del proceso)

#### 2. Crear licencias

Únicamente, indique al usuario en una nueva ventana:

- este título,
- la instrucción: ¿A cuántos personas desea crear licencias?, acompañadas por una caja de texto que reciba un valor entero entre 1 y 250.
- y los botones: "Generar", "Limpiar" y "Regresar"

Si se generan, debe crearse cada objeto con la siguiente estructura:

Cédula	9 dígitos enteros *	Generados aleatoriamente
Nombre Completo	Nombre y 2 apellidos	Generados por el sistema

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

	separados por espacio																	
Fecha de Nacimiento	dd-mm-aaaa*	Generados por el sistema, pero recuerde que sólo los mayores de edad pueden tener licencia.																
Fecha de Expedición	dd-mm-aaaa*	Tomada del sistema con el día actual																
Fecha de Vencimiento	dd-mm-aaaa*	Generada según la fecha del sistema y mediante los criterios: Si su edad está entre 18 y 25, su licencia vence en 3 años, si es mayor a ello, vence en 5 años. (Se generalizará sin importar el tipo de licencia)																
Tipo de licencia		Según los datos leídos en los subtipos de licencias provenientes del XML, crear una lista simple y escoger aleatoriamente un subtipo para registrarlo en este rubro.																
Tipo de Sangre		Determinado aleatoriamente de las opciones de la lista simple un valor: O+, O-, A+, A-, B+, B-, AB+, AB-																
Donador		Registrar un valor aleatorio booleano, indica si es donador de órganos o no.																
Sede		Según el primer carácter de la cédula generada, determinar dónde debe ir a gestionar su licencia. Analice y comprenda que: <table><tr><td>Cédula</td><td>Cantidad de opciones en dónde gestionar la licencia</td></tr><tr><td>1,8,9</td><td>2</td></tr><tr><td>2</td><td>3</td></tr><tr><td>3</td><td>1</td></tr><tr><td>4</td><td>1</td></tr><tr><td>5</td><td>2</td></tr><tr><td>6</td><td>2</td></tr><tr><td>7</td><td>2</td></tr></table> <p>Pues si es cédula de Alajuela, pues ir a:</p> <ul style="list-style-type: none"><li>- Gran área metropolitana</li><li>- Montecillos Alajuela</li><li>- Tránsito San Ramón</li><li>- Zona Norte</li><li>- San Carlos</li></ul> <p>Según: ¿Dónde queda la Dirección General de Educación Vial?</p> <p>Ustedes como ingenier@s definen la estrategia.</p>	Cédula	Cantidad de opciones en dónde gestionar la licencia	1,8,9	2	2	3	3	1	4	1	5	2	6	2	7	2
Cédula	Cantidad de opciones en dónde gestionar la licencia																	
1,8,9	2																	
2	3																	
3	1																	
4	1																	
5	2																	
6	2																	
7	2																	
Puntaje	Un valor entero* entre 0 y 12 únicamente.	12= licencia sin infracciones Menor o igual a 6= debe volver a hacer el examen teórico. 0= se retira su licencia permanentemente.																
Correo electrónico	Contra Expresión Regular	Se formará contra el primer apellido, letra del segundo apellido y letra del nombre. Ejemplo: Ana Mora Rojas. morara@gmail.com																

Considere que ningún valor puede quedar en blanco.

\* Valide contra expresión regular

Los objetos deben ser guardados en una lista general y respaldados a memoria secundaria.

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

### 3. Renovar licencia

Debe crearse una ventana con un título, una caja de texto y los botones: “Renovar”, “Limpiar” y “Regresar”.

Si la cédula es correcta y existe, y se confirma positivamente, se renueva su licencia según la edad de la persona. No olvide analizar el puntaje.

Sino, se emite el texto según corresponda.

### 4. Generar PDF

Debe crearse una ventana con un título, una caja de texto y los botones: “Generar”, “Limpiar” y “Regresar”.

Si la cédula es correcta y existe, y se confirma positivamente, se crea un pdf con los datos respectivos, muy fielmente a:

REPUBLICA DE COSTA RICA
Licencia de Conducir
Nº: <b>CI-103500620</b> (Número de cédula)
Expedición: 27-05-2021 (según fecha del sistema)
Nacimiento: 10-12-2003
Vencimiento: <b>27-05-2024</b> (según la edad)
Tipo: <b>B1</b>
Donador
T.S. <b>O+</b>
<b>JUAN PÉREZ DÍAS</b>
27-05-2021 9:30 San Sebastián

Sino, se emite el texto según corresponda.

### 5. Reportes de Excel.

Sin importar el tiempo de reporte, debe generarse en cada salida un archivo de texto para ser abierto en Excel. Genere un submenú para mostrar:

- a. Totalidad de licencias

<b>Solicita</b>	Nada									
<b>Devuelve</b>	Título del reporte Fecha y hora del sistema									
	<b>Cédula</b>	Nombre	FechaNac	FechaExp	FechaVenc	TipoLicen	TipoSangre	Donador	Sede	Puntaje

- b. Por tipo de licencia

Cree una ventana gráfica con las opciones en una lista de selección con todos los tipos según corresponda

<b>Solicita</b>	Determinar de la caja de selección una opción de: A,B,C,D, E			
<b>Devuelve</b>	Título del reporte Fecha y hora del sistema <table><tr><td>Cédula</td><td>Nombre</td><td>TipoLicen</td></tr></table>	Cédula	Nombre	TipoLicen
Cédula	Nombre	TipoLicen		

- c. Examen por sanción

Considere son los menores e iguales a 6 pero no cero.

<b>Solicita</b>	Nada			
<b>Devuelve</b>	Título del reporte			
	Fecha y hora del sistema			
	<b>Cédula</b>	Nombre	TipoLicen	Puntaie

d. Los donantes de órganos

Se muestran únicamente los donantes donares de órganos.

<b>Solicita</b>	Nada									
<b>Devuelve</b>	Título del reporte Fecha y hora del sistema									
	Cédula	Nombre	TipoLicen							

e. Licencia anulada

Recuerde son los de puntaje 0.

<b>Solicita</b>	Nada									
<b>Devuelve</b>	Título del reporte Fecha y hora del sistema									
	Cédula	Nombre	FechaNac	FechaExp	FechaVenc	TipoLicen	TipoSangre	Donador	Sede	

f. Licencias por sede

Usted es libre de crear la venta gráfica que requiera para que el usuario pueda escoger la sede deseada.

<b>Solicita</b>	Se determina de cuál sede de desea obtener las licencias.									
<b>Devuelve</b>	Título del reporte Fecha y hora del sistema									
	Cédula	Nombre	FechaNac	FechaExp	FechaVenc	TipoLicen	TipoSangre	Donador	Sede	Puntaje

**6. Acerca de**

Colocar los datos de los programadores.

**7. Salir**

Indica el siguiente mensaje:

***“No olvides gestionar pronto tu licencia”***

Y cierra la aplicación.



### Puntos para evaluar:

1. Correctitud de la solución computacional - 80%

Funcionalidad	Procesos	Valor	Responsable																		
Base de Datos	Lista de Objetos. Calificación binaria y obligatoria o no se califica TP#3, se carga de memoria secundaria.	5	A1 y A2																		
Definición de la clase	Set y get para los 11 atributos, más los métodos generales.	10	A1 y A2																		
Menú	Ventana principal del software. Con todos los botones necesarios. Carga de Memoria secundaria.	3	A1																		
Crear XML	Lee de WEB. Forma el XML.	10	A1																		
Crear licencias	Ventana solicitando cantidad y su validación Creación de N objetos insertados en lista Almacenamiento en memoria secundaria de la lista binaria de objetos.	12	A2																		
Renovar licencia	Ventana y validación respectiva. Renovación según corresponda y la realimentación del caso.	5	A2																		
Generar PDF	Ventana y validación respectiva. Si la cédula es correcta y existe, y se confirma positivamente, se crea un pdf con los datos respectivos, muy fielmente a lo solicitado.	7	A1																		
Reportes en Excel	<table><tr><td>a. Totalidad de licencias</td><td>7</td><td>A1</td></tr><tr><td>b. Por tipo de licencia</td><td>9</td><td>A1</td></tr><tr><td>c. Examen por sanción</td><td>7</td><td>A1</td></tr><tr><td>d. Los donantes de órganos</td><td>7</td><td>A2</td></tr><tr><td>e. Licencia anulada</td><td>7</td><td>A2</td></tr><tr><td>f. Licencias por sede</td><td>9</td><td>A2</td></tr></table> Responsabilidades del Alumno 1 y Alumno2	a. Totalidad de licencias	7	A1	b. Por tipo de licencia	9	A1	c. Examen por sanción	7	A1	d. Los donantes de órganos	7	A2	e. Licencia anulada	7	A2	f. Licencias por sede	9	A2	48	
a. Totalidad de licencias	7	A1																			
b. Por tipo de licencia	9	A1																			
c. Examen por sanción	7	A1																			
d. Los donantes de órganos	7	A2																			
e. Licencia anulada	7	A2																			
f. Licencias por sede	9	A2																			
Acerca de	Ventana con los datos de los programadores.	1	A1																		
Salir		1	A2																		

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, ***semana a semana estrictamente o no se califica el 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/1-5JrhgrTQD625-V8BHVPGVpoMVIDSKmZ> 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.

**No pierda la buena costumbre de medir sus tiempos**, se recomienda leer el libro: Introducción al Proceso de Software Personal (PSP)

**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, matrices o diccionarios según se especificó 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 hoy a las 12md al correo de la profesora [lsarmiento@tec.ac.cr](mailto:lsarmiento@tec.ac.cr) con copia al asistente
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. 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.**