



EVALUACION N°3 UNIDAD 3

FPY11	01	FUNDAMENTOS	DE PROGRAM	ACIÓN	
NOMBRE			SECCIÓN		
RUT			FECHA		

PUNTAJES Y NOTA / ESCALA DE EXIGENCIA (60%)					
100%	NOTA: 7.0	PUNTAJE OBTENIDO			
60%	NOTA: 4.0	NOTA			

INSTRUCCIONES GENERALES:

El desarrollo de la evaluación es individual y PRESENCIAL.

La nota 4,0 se obtiene logrando un 60% del puntaje total.

Cuando termine el encargo, comprima la carpeta donde ha dejado los trabajos de la prueba y siga las instrucciones del docente.

DEBE UTILIZAR LAS INSTRUCCIONES Y CONVENCIONES VISTAS EN CLASES.

Usar lenguaje de programación PYTHON e IDE VSCODE

LOS TRABAJOS ENTREGADOS Y CONSIDERADOS PLAGIOS TENDRÁN
UNA EVALUACIÓN DE 1.0.

ENUNCIADO

Una automotora, requiere almacenar la información de los vehículos que vende, por lo que se pide crear una aplicación con las siguientes opciones:

Opción 1: Guardar.

Debe guardar y validar los siguientes datos:

Patente : Solo 6 caracteres.

Marca : 3 caracteres como mínimo. Modelo : 3 caracteres como mínimo.

Año del vehículo : Sobre 1980.
Valor : Mínimo 500.000.



Opción 2: Buscar.

Permite buscar a un vehículo por su Patente y mostrar toda su información que ha sido almacenada en la opción anterior. Además, debe mostrar la cantidad de años que han pasado desde su fabricación.

Opción 3: Listar.

Debe mostrar a todos los vehículos registrados en la aplicación y su estado (vendido o no).

Opción 4: Imprimir

Debe imprimir el contrato de compraventa, donde el número se obtiene del año, mes, día, hora, minutos y segundos, además, deberá mostrar los datos del vehículo a vender, el cual, se buscará por medio de la patente. <u>Tener presente que el vehículo debe estar disponible para la venta</u>, confirme si se venderá para cambiar el estado a vendido.

Opción 5: **Salir**. Cierra la aplicación.

OTROS REQUISITOS:

Debe usar **funciones**, **listas**, **usar menú**, los **mensajes** deben ser claros para los usuarios tanto si hay error como cuando fue exitosa la solicitud, validar las opciones como el ingreso de datos mencionados anteriormente, limpiar pantalla, entre otros que permitan el buen funcionamiento de la aplicación. Los datos deben ser almacenados en un archivo con formato **CVS**. Utilizar solo **librerías** vistas en clases. Utilizar **GitHub** para ir respaldando los distintos cambios realizados en el código.

DEBE UTILIZAR LAS INSTRUCCIONES Y CONVENCIONES VISTAS EN CLASES.

ENTREGA

Para la entrega, deberá comprimir los archivos .py con el siguiente nombre: nombreAlumno_PGY1121_SECCION, Ejemplo: JuanPerez_PGY1121_001D



Rúbrica: Evaluación 3

FPY1101

FUNDAMENTOS DE PROGRAMACIÓN

Categoría	% logro	Descripción niveles de logro
Muy buen desempeño	100%	Demuestra un desempeño destacado, evidenciando el logro de todos los aspectos evaluados en el indicador.
Buen desempeño	80%	Demuestra un alto desempeño del indicador, presentando pequeñas omisiones, dificultades y/o errores.
Desempeño aceptable	60%	Demuestra un desempeño competente, evidenciando el logro de los elementos básicos del indicador, pero con omisiones, dificultades o errores.
Desempeño incipiente	30%	Presenta importantes omisiones, dificultades o errores en el desempeño, que no permiten evidenciar los elementos básicos del logro del indicador, por lo que no puede ser considerado competente.
Desempeño no logrado	0%	Presenta ausencia o incorrecto desempeño.

Indicador de Evaluación	Categorías de Respuesta				Pond eració	
	Muy buen desempeño 100%	Buen desempe ño 80%	Desempeño aceptable 60%	Desempeñ o incipiente 30%	Desempeño no logrado 0%	n Indica dor de Evalu ación
Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado.	Declara las colecciones (listas y/o diccionarios) necesarias para el funcionamiento correcto de la aplicación.	NO APLICA	Falta alguna de las colecciones (listas y/o diccionarios) necesarias para el funcionamiento correcto de la aplicación.	NO APLICA	No declara las colecciones necesarias para el funcionamient o correcto de la aplicación o hay un error de sintaxis.	20%
Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado.	Las colecciones utilizadas en la aplicación permiten la gestión adecuada de los datos a tratar.	NO APLICA	Usa colecciones con algunos errores en su uso para tratar los datos necesarios en la aplicación.	NO APLICA	No gestiona la información necesaria a través de las colecciones declaradas o hay un error de sintaxis.	20%
Utiliza archivos para logar la persistencia de los datos de la aplicación según el caso planteado.	Genera un archivo en formato csv con los resultados obtenidos en la aplicación.	NO APLICA	El archivo csv no contiene todos los datos solicitados en el caso planteado o no coincide con los datos mostrados en pantalla.	NO APLICA	No genera el archivo csv o hay un error de sintaxis.	10%



Utiliza librerías de sistema para la optimización del código según el caso planteado.	Usa las librerías necesarias para optimizar el funcionamiento del código y permite la generación de datos aleatorios.	NO APLICA	Usa las librerías necesarias para optimizar el funcionamiento del código, pero no genera datos aleatorios.	NO APLICA	No utiliza librerías útiles para el desarrollo de la aplicación o hay un error de sintaxis.	<mark>15%</mark>
Programa funciones que permitan la reutilización de código según el problema planteado	Codifica las 4 funciones solicitadas en el enunciado	Codifica solo 3 funciones solicitadas en el enunciado	Codifica solo 2 funciones solicitadas en el enunciado	Codifica solo 1 función solicitada en el enunciado	No codifica las funciones solicitadas en el enunciado o hay un error de sintaxis	15%
Integra funciones invocadas desde el programa principal para dar solución al problema planteado.	Hace uso de las 4 funciones solicitadas en el enunciado desde el programa principal	Hace uso de 3 de las funciones solicitadas en el enunciado desde el programa principal	Hace uso de 2 de las funciones solicitadas en el enunciado desde el programa principal	Hace uso de 1 de las funciones solicitadas en el enunciado desde el programa principal	No invoca las funciones solicitadas desde el programa principal o hay un error de sintaxis	10%
Utiliza un sistema de control para administrar las versiones código (GitHub) en el desarrollo de una aplicación.	Crea un repositorio con el código de la aplicación	NO APLICA	NO APLICA	NO APLICA	NO APLICA	10%
Total					100%	