Evaluación Final Transversal
Instrucciones y Pauta de Evaluación

Sigla	Nombre Asignatura	Tiempo Asignado	% Ponderación
FPY1101	Fundamentos de Programación	7 h	**40%**

1. Instrucciones generales para el/la estudiante

Esta es una evaluación que corresponde a una **ejecución práctica** y tiene un 40**%** de ponderación sobre la nota final de la asignatura.
El **tiempo** para desarrollar esta evaluación es de **7 horas** y se realiza de manera **individual** en **laboratorio**
La evaluación consiste en:
Desarrollar una aplicación en Python utilizando los conceptos de programación desarrollados durante la asignatura:

- Estructuras de entrada y salida
- Estructuras de decisión
- Estructuras de repetición
- Colecciones
- Funciones
- Manejo de archivos
- Uso de github como repositorio de código

Preste atención en la rúbrica, donde se señala explícitamente que de haber errores de sintaxis no obtendrá puntaje en ningún ítem.

2. Evaluación

- Desarrolle una aplicación en Python utilizando Visual Studio Code como entorno de desarrollo según el siguiente enunciado:

Una empresa necesita analizar los datos de sus trabajadores para generar algunos reportes, y le ha solicitado a usted que realice un prototipo en Python con los siguientes requerimientos:

La aplicación debe permitir analizar los sueldos de 10 empleados, los cuales para efectos de este prototipo se crearán de forma aleatoria entre $300.000 y $2.500.000. Utilice la siguiente lista para asignar los sueldos a cada empleado:

trabajadores = ["Juan Pérez”,”María García”,”Carlos López”,”Ana Martínez”,”Pedro Rodríguez”,”Laura Hernández”,”Miguel Sánchez”,”Isabel Gómez”,”Francisco Díaz”,”Elena Fernández"]

La aplicación deberá poseer un menú con las siguientes funcionalidades:

1. Asignar sueldos aleatorios
2. Clasificar sueldos
3. Ver estadísticas.
4. Reporte de sueldos
5. Salir del programa

Cada función se detalla a continuación:

1. **Asignar sueldos aleatorios**

Para la generación de estos sueldos debe crear una función capaz de generar los 10 sueldos de forma aleatoria los que serán usados posteriormente para la ejecución del programa.

2. Clasificar sueldos

Deberá desarrollar una función que permita mostrar la lista de empleados con su sueldo y su respectiva clasificación según el siguiente esquema:

```
Sueldos menores a $800.000 TOTAL: 2

Nombre empleado     Sueldo
Juan Pérez          $500000
María García        $700000

Sueldos entre $800.000 y $2.000.000
TOTAL: 2

Nombre empleado     Sueldo
Pedro Soto          $1100000
Isabel Gómez        $800000

Sueldos superiores a $2.000.000
TOTAL: 1

Nombre empleado     Sueldo
Miguel Sánchez      $2100000

TOTAL SUELDOS: $ 5200000
```

Ejemplo 1 - Tabla de Sueldos con solo 5 datos de sueldo

3. Ver estadísticas

Crear una función que permita mostrar por pantalla los siguientes datos con respecto a los sueldos:

- Sueldo más alto
- Sueldo más bajo
- Promedio de sueldos
- Media geométrica

4. Reporte de sueldos

La aplicación deberá poseer una función para mostrar el detalle de los sueldos de los trabajadores, según la siguiente regla de negocio:

- Descuento salud 7%
- Descuento AFP 12%
- Sueldo líquido calculado en base al sueldo base menos el descuento en salud y menos el descuento afp.

Y mostrarse como en la siguiente tabla de ejemplo:

```
Nombre empleado     Sueldo Base      Descuento Salud   Descuento AFP     Sueldo Líquido
Juan Pérez          $1000000         $70000            $120000           $810000
Pedro Soto          $800000          $56000            $96000            $648000
```

Ejemplo 2 – reporte de sueldos en pantalla

Estos datos se deberán exportar a un archivo de texto separado por comas (.csv) para su posterior lectura en otra aplicación.

5. **Salir del programa**

La aplicación deberá finalizar para salir el programa mostrando un mensaje con sus datos

```
Finalizando programa…
Desarrollado por Carlos Vergara
RUT 12.345.678-9
```

Ejemplo 3 - Mensaje de salida de la aplicación

3. Pauta de Evaluación

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					
	Muy buen desempeño 100%	Buen desempeño 80%	Desempeño aceptable 60%	Desempeño incipiente 30%	Desempeño no logrado 0%	Ponderación Indicador de Evaluación
IE 2.3.1 Programa estructuras de control y validación de acuerdo con las reglas de negocio planteadas.	Todas las expresiones lógicas utilizadas permiten el funcionamiento de la aplicación orientando el flujo del programa.	Las expresiones lógicas utilizadas permiten el funcionamiento de la aplicación, pero UNA de estas no funciona.	Las expresiones lógicas utilizadas permiten el funcionamiento de la aplicación, pero DOS de estas no funcionan.	Las expresiones lógicas utilizadas permiten el funcionamiento de la aplicación, pero TRES de estas no funcionan.	Las expresiones lógicas utilizadas permiten el funcionamiento de la aplicación, pero CUATRO O MÁS de estas no funcionan.	10%
IE 2.4.1 Utiliza estructuras de decisión permitiendo al programa seguir cambiar su flujo de ejecución según el caso planteado.	Utiliza estructuras de decisión de Python con la sintaxis, permitiendo al programa seguir el flujo necesario para su funcionamiento.	Utiliza estructuras de decisión de Python, pero UNA de las estructuras de decisión genera errores o no está presenta en el flujo del programa.	Utiliza estructuras de decisión de Python, pero DOS de las estructuras generan errores de flujo o no están presentes en el flujo de la aplicación.	Utiliza estructuras de decisión de Python, pero TRES de las estructuras generan errores de flujo o no están presentes en el flujo de la aplicación.	Utiliza estructuras de decisión de Python, pero CUATRO O MÁS de las estructuras generan errores de flujo o no están presentes en el flujo de la aplicación.	10%
IE 2.4.2 Utiliza estructuras de repetición permitiendo al programa iterar las veces necesarias según el caso planteado.	Utiliza estructuras de repetición de Python con la sintaxis, permitiendo al programa seguir el flujo necesario para su funcionamiento	Utiliza estructuras de repetición de Python, pero UNA de las estructuras genera errores o no está presente en el flujo del programa.	Utiliza estructuras de repetición de Python, pero DOS de las estructuras generan errores o no están presentes en el flujo del programa.	Utiliza estructuras de repetición de Python, pero TRES de las estructuras generan errores o no están presentes en el flujo del programa.	Utiliza estructuras de repetición de Python, pero CUATRO O MÁS de las estructuras generan errores o no están presentes en el flujo del programa.	10%
IL 3.1.1 Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado.	Declara todas las colecciones (listas y/o diccionarios) necesarias para el funcionamiento de la aplicación.	Declara la mayoría de las colecciones (listas o diccionarios) para el funcionamiento de la aplicación	Declara la mitad de las colecciones (listas o diccionarios) para el funcionamiento de la aplicación.	Declara menos de la mitad de las colecciones (listas y/o diccionarios) para el funcionamiento de la aplicación.	No declara las colecciones necesarias para el funcionamiento de la aplicación o hay un error de sintaxis.	10%
IL 3.2.1 Utiliza arreglos y matrices para la inserción, eliminación, modificación y	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	No aplica	No gestiona la información necesaria a través de las colecciones	10%

búsqueda de datos temporales para cumplir con los requerimientos del problema planteado.			necesarios en la aplicación.		declaradas o hay un error de sintaxis.	
IL 3.3.1 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	Genera un archivo csv, pero 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%
IL 4.1.1Utiliza 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.	10%
IL 4.2.1 Programa funciones que permitan la reutilización de código según el problema planteado.	Codifica las 4 funciones para solucionar el requerimiento.	Codifica 3 funciones solicitadas para solucionar el requerimiento.	Codifica 2 funciones solicitadas para solucionar el requerimiento.	Codifica 1 función solicitada para solucionar el requerimiento.	No codifica las funciones solicitadas o hay un error de sintaxis.	15%
IL 4.3 Integra funciones invocadas desde el programa principal para dar solución al problema planteado.	Usa las 4 funciones desde el programa principal para dar solución al requerimiento.	Usa 3 funciones desde el programa principal para dar solución al requerimiento.	Usa 2 funciones desde el programa principal para dar solución al requerimiento.	Usa 1 función desde el programa principal para dar solución al requerimiento.	No invoca las funciones solicitadas desde el programa principal o hay un error de sintaxis.	15%
					Total	**100%**