

# Evaluación Formativa N°3

Nombre: Diseño de aplicaciones con Python I

Sigla	Nombre Asignatura	Tiempo Asignado	% Ponderación
FPY1101	Fundamentos de Programación	120 minutos	0%

## 1. Situación evaluativa

<input checked="" type="checkbox"/>	Ejecución práctica
-------------------------------------	--------------------

## 2. Agente evaluativo

<input checked="" type="checkbox"/>	Heteroevaluación	<input type="checkbox"/>	Coevaluación	<input type="checkbox"/>	Autoevaluación
-------------------------------------	------------------	--------------------------	--------------	--------------------------	----------------

### 3. Tabla de Especificaciones

Resultado de Aprendizaje	Indicador de Logro (IL)	Indicador de Evaluación (IE)	Ponderación Indicador Logro	Ponderación Indicador de Evaluación <sup>1</sup>
RA3 Construye estructuras de datos temporales, utilizando arreglos para almacenar datos, con el fin de darle persistencia a éstos y trabajar con problemas que necesiten guardar información temporalmente.	IL 3.1 Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado.	Identifica colecciones y arreglos que permitan el almacenamiento de datos según los requerimientos del problema planteado.	20%	20%
	IL 3.2 Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado.	Utiliza arreglos y matrices para la inserción, eliminación, modificación y búsqueda de datos temporales para cumplir con los requerimientos del problema planteado.	20%	20%
	IL3.3 Utiliza archivos para lograr la persistencia de los datos de la aplicación según el caso planteado.	Utiliza archivos para lograr la persistencia de los datos de la aplicación según el caso planteado.	10%	10%
RA4 Modulariza programas, mediante funciones con y sin parámetros, para simplificar y darle eficiencia a las soluciones construidas.	IL 4.1 Utiliza librerías de sistema para la optimización del código según el caso planteado.	Utiliza librerías de sistema para la optimización del código según el caso planteado.	20%	20%
	IL 4.2 Programa funciones que permitan la reutilización de código según el problema planteado.	Programa funciones que permitan la reutilización de código según el problema planteado.	20%	20%

<sup>1</sup> La ponderación del Indicador de Logro, debe ser distribuida como puntaje entre las preguntas que lo conformen.

	IL 4.3 Integra funciones invocadas desde el programa principal para dar solución al problema planteado.	Integra funciones invocadas desde el programa principal para dar solución al problema planteado.	10%	10%
Total			100%	100%

## 4. Instrucciones generales para el/la estudiante

Esta es una evaluación que corresponde a una prueba de ejecución práctica y tiene un **0%** de ponderación sobre la nota final de la asignatura.

El **tiempo** para desarrollar esta evaluación es de **120 minutos** y se realiza en **parejas** en **laboratorio PC**.

**La evaluación consiste en:**

- Desarrollo de una aplicación en Python y en entorno de desarrollo Visual Studio Code
- Uso de colecciones
- Uso de archivos de texto
- Uso de librerías estándar de python

## 5. Evaluación

Desarrolle una aplicación en Python utilizando Visual Studio que permita resolver el siguiente caso:

Una empresa necesita desarrollar una aplicación que permita registrar los sueldos brutos de los trabajadores y calcular el líquido a pagar. Para ello necesita que la aplicación cumpla con las siguientes funcionalidades

1. Registrar trabajador
2. Listar los todos los trabajadores
3. Imprimir planilla de sueldos
4. Salir del Programa

### Registrar trabajador

Para registrar un trabajador se requiere los siguiente: Nombre y Apellido, Cargo, Sueldo bruto. Una vez ingresado los datos, se deben calcular, los valores de acuerdo con la siguiente tabla

Trabajador	Cargo	Sueldo Bruto	Desc. Salud	Desc. AFP	Líquido a pagar
Homero Simpson	CEO	1000000	70000	120000	810000

Debe validar que todos los datos sean ingresados.

### Listar los todos los trabajadores

Debe mostrar en la pantalla la lista de todos los trabajadores similar al ejemplo anterior de registrar un solo trabajador

### Imprimir planilla de sueldos

Para imprimir la planilla, el usuario puede seleccionar imprimir todos o por algún cargo en específico. Estos cargos deben estar previamente definidos en algún tipo de colección de Python en el código y por lo menos deben ser tres, por ejemplo: CEO, Desarrollador, Analista de datos.

Al seleccionar uno de los cargos, se generará un archivo de texto (.txt) con el detalle de los sueldos. Este debe tener la misma forma del registro completo de las opciones anteriores, pero en archivo de texto.

**Cada una de estas opciones** de la aplicación debe estar desarrollada en **una función** que debe ser llamada desde el programa principal

El programa debe funcionar hasta que el usuario decida finalizar el programa

Al finalizar, debe subir el archivo de comprimido como evidencia del trabajo realizado. Luego, debe aplicar la pauta de evaluación formativa (Coevaluación).

## 6. 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					Ponderación Indicador de Evaluación
	Muy buen desempeño 100%	Buen desempeño 80%	Desempeño o aceptable 60%	Desempeño incipiente 30%	Desempeño no logrado 0%	
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	-----	Falta alguna de las colecciones (listas y/o diccionarios) necesarias para el funcionamiento correcto de la aplicación	-----	No declara las colecciones necesarias para el funcionamiento 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		Usa colecciones con algunos errores en su uso para		No gestiona la información necesaria a través de las colecciones	20%

	gestión adecuada de los datos a tratar.	-----	tratar los datos necesarios en la aplicación.	-----	declaradas o hay un error de sintaxis.	
Utiliza archivos para lograr 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.	-----	El archivo csv no contiene todos los datos solicitados en el caso planteado o no coincide con los datos mostrados en pantalla.	-----	No genera el archivo csv o hay un error de sintaxis.	<b>10%</b>
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.	-----	Usa las librerías necesarias para optimizar el funcionamiento del código, pero no genera datos aleatorios.	-----	No utiliza librerías útiles para el desarrollo de la aplicación o hay un error de sintaxis.	<b>20%</b>
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	<b>20%</b>
Integra funciones invocadas desde el programa principal para dar solución al problema planteado.	Hace uso de las 4 funciones solicitadas en el	Hace uso de 3 de las funciones solicitadas en el enunciado	Hace uso de 2 de las funciones solicitadas en el	Hace uso de 1 de las funciones solicitadas en el	No invoca las funciones solicitadas desde el programa	<b>10%</b>

	enunciado desde el programa principal	desde el programa principal	enunciado desde el programa principal	enunciado desde el programa principal	principal o hay un error de sintaxis	
Total						100%