

Trabajo Práctico de fin de curso – Python II

Modalidad del trabajo

El presente trabajo práctico tiene como fin evaluar los conocimientos adquiridos por el estudiante a lo largo del cursado de Python 2. El estudiante contará con una determinada cantidad de tiempo para llevar adelante el trabajo desde el momento en que recibe el enunciado, hasta una hora antes de hacer la defensa del mismo frente al instructor. Durante el transcurso del tiempo comentado anteriormente, el estudiante es libre de consultar otros medios (como paginas web) o al instructor para poder sacarse las dudas que tenga, ya sea por parte del enunciado o para dar solución a algún punto del trabajo. Una vez cerrado el tiempo para entregar los trabajos, los cuales deberán subirse a la plataforma del Classroom (revisar **presentación del examen** para saber más sobre el tema), el estudiante deberá coordinar una llamada con el instructor del curso y, durante un tiempo no mayor a quince minutos, deberá mostrar el funcionamiento de su programa, como también su código, con el fin de defender el mismo frente a las consultas que plantee el instructor. Una vez terminado la defensa del trabajo, si el instructor aprueba el trabajo práctico, se le hará llegar al estudiante un cuestionario que contará con una n cantidad de preguntas sobre los temas vistos a lo largo del curso, el mismo corresponde al examen teórico. Si el estudiante pasa con éxito las dos instancias, habrá terminado con éxito el curso de Python 2.

Presentación del examen

La presentación del examen debe tener el siguiente nombre: *nombre_apellido_trabajo_practico_final_python_2.py*, donde el estudiante deberá remplazar *nombre_apellido* por su nombre y apellido, respectivamente. A continuación, se puede ver un ejemplo de cómo debe estar nombrado el archivo de la solución del examen:



Gaston_Gonzalez_trabajo_practico_final_python_2.py

En caso de que el estudiante decida hacer uso de más de un archivo para el desarrollo de su programa, debe subir todos los archivos al Classroom, y el archivo *nombre_apellido_trabajo_practico_final_python_2.py* debe ser el archivo principal del programa que interactúe con el resto de los archivos, sin excepción. Es importante aclarar que, en caso de que el estudiante trabaje con varios archivos, la ausencia de uno al momento de subir la resolución a la plataforma significará que no fue presentado y el instructor dará dado por incompleto el trabajo, ¡revise que haya subido todos los archivos correspondientes!

La entrega de los mismos se hará en el Classroom, en la plataforma tendrán habilitado la sección para subir los archivos en la tarea.

Corrección del examen

El instructor corregirá el trabajo y se subirán los comentarios correspondientes en la plataforma para que el estudiante para hacer las revisiones correspondientes. A continuación, dejamos una tabla con los valores correspondiente de las calificaciones:

Porcentaje obtenido	Condición del examen
Entre 0 y 69 %	Reprobado
Entre 70 y 100 %	Aprobado

Es importante destacar que la corrección de los puntos no es lineal, esto quiere decir que si el estudiante no logro terminar algunos de los puntos o el funcionamiento final no es el esperado, no se dará el punto por descartado, si no que el instructor revisará el código hecho y determinará el puntaje que le corresponde.

Extra

Al terminar todo curso, la academia y el instructor alcanzan a los estudiantes un pequeño cuestionario donde se les consulta sus impresiones sobre distintos aspectos del curso, con el fin de hacer un feedback que nos permita hacer correcciones para futuros cursos que se dicten. La encuesta es anónima y es obligatoria para rendir el examen de fin de curso.

Sin nada más que comentar, solos nos queda desearles: ¡éxitos!

Trabajo Práctico Final

Caso de estudio: biblioteca

Una biblioteca ha contratado su servicio para llevar adelante el desarrollo de un software que le permita administrar el préstamo de los libros. Los libros cuentan con una serie de datos que el programa debe manejar: título, autor, edición, lugar de impresión, editorial, si es una traducción, cantidad de páginas y la condición. La condición de un libro es la situación actual en la que el mismo se encuentra, la cual puede ser algunas de las siguientes:

- **Préstamo en proceso:** esta condición se desarrolla cuando el libro está en medio de un préstamo a alguno de los afiliados a la biblioteca.
- **Disponible:** esta condición indica que el libro está en condición de ser prestado a algún afiliado.
- **Retraso:** esta condición se aplica cuando un libro no ha sido devuelto una vez llegado la fecha de devolución.
- **En restauración:** hay libros que, llegado cierto momento, pueden necesitar una restauración, en dichas situaciones no se puede prestar el libro.

Es importante manejar la situación en la que únicamente se puede prestar un libro cuando éste esté en condición de disponible, toda otra condición debe hacer que el programa impida registrar el préstamo. La condición de un libro debe poder ser cambiada por el usuario en cualquier momento.

Al momento de realizar un préstamo, se deben registrar el nombre completo de la persona a la que se va a prestar, teléfono y mail para poder contactarse de ser necesario, fecha de inicio de préstamo, fecha de devolución y libro a prestar.

Dentro de las funcionalidades que le programa debe proporcionar, están las siguientes:

- Libros:
 - Dar de alta un libro
 - Modificar los datos de un libro
 - Dar de baja un libro
 - Consultar un libro en particular por su nombre
 - Mostrar todos los libros cargados

- **Préstamos:**
 - Registrar un préstamo
 - Dar por terminado un préstamo
 - Reclamar un préstamo: esta situación se plantea cuando el usuario controla los prestamos vigentes y el programa detecta que hay prestamos que no se devolvieron, inmediatamente el programa muestra los datos de la persona a la que se le presto el libro para que el usuario para comunicarse con él/ella y así poder acordar la devolución.

El desarrollo del software es libre y queda a disposición de los programadores, pero hay unas ciertas condiciones que se deben cumplir al momento de llevar adelante el desarrollo del programa:

1. **El programa debe aplicar el Paradigma Orientado a Objetos:** la cantidad de objetos y su interacción es libre de manejar por el programador, pero debe haber como mínimo una clase que permita levantar el programa.
2. **Se debe desarrollar una interfaz gráfica:** el programa debe contar con una interfaz gráfica construida usando el módulo tkinter. Queda a disposición del programador la distribución gráfica del programa.
3. **Perduración de datos:** se debe desarrollar una solución que permita que los datos cargados por los usuarios del sistema perduren en el tiempo y no se pierdan una vez terminada la ejecución del programa. Queda a disposición del programador si desea utilizar archivos, una base de datos SQLite o ambas, así como también queda a su elección la forma en la que decide manejar la información en dichas herramientas.