

Trabajo Práctico Grupal Obligatorio – Primer cuatrimestre 2021

Fecha límite de 1era. entrega: **01 / 05 / 2021**

Fecha límite de 2da. entrega: **05 / 06 / 2021**

Fecha límite de 3ra. entrega: **03 / 07 / 2021**

Fecha límite de entrega opcional: **12 / 07 / 2021**

Problema a resolver

La Secretaría de Igualdad de Géneros y Oportunidades de la CTA Avellaneda, considera posible implementar un sistema para obtener estadísticas de la registración anual de denuncias de femicidios ocurridos en el Área Metropolitana de Buenos Aires (AMBA). En este contexto, les ha requerido a estudiantes de Informática de UNDAV, el desarrollo de un prototipo de software que responda a las siguientes particularidades:

Debe solicitar el ingreso del año del informe (entre 2015 y 2021, exigiendo reingresos hasta asegurar su validez). Por cada municipio del AMBA del cual se hayan recibido denuncias, debe solicitar datos que ingresará el usuario responsable del sistema. Los datos son: el nombre del municipio, la cantidad de denuncias de femicidio (en el año del informe) y, de éstas, la cantidad de casos con denuncias previas de violencia de género, la fecha del primer caso denunciado en ese año y, cuando se trate de un municipio con más de un femicidio denunciado, la fecha del último caso. Como consecuencia, deberá mostrar en pantalla la información que más abajo se detalla.

El sistema debe ser programado en Python 3 y se requiere que ofrezca las siguientes prestaciones:

1. Solicite el ingreso de los datos mencionados y controle: la validez del año del informe y, para cada municipio, la validez de fechas (del primer caso y, cuando corresponda, del último caso denunciado, que no deberá ser anterior a la del primer caso), que la cantidad de denuncias de femicidio sea mayor que cero y menor que 200; a su vez, que la cantidad de casos con denuncia previa de violencia de género sea mayor o igual que cero, pero que no supere la cantidad de denuncias informada. Cuando un dato no sea válido, el programa debe solicitar el reingreso del mismo hasta asegurar su validez.
2. Calcule el porcentaje de casos con denuncias previas de violencia de género en cada municipio informado (cociente entre cantidad de casos con denuncias previas y cantidad de denuncias respectiva, multiplicado por cien), para mostrarlo en pantalla, simultáneamente, con los datos válidos del municipio.
3. Determine y muestre (con descripciones expresivas): la cantidad total de municipios informados, la cantidad de municipios que registren denuncias previas de violencia de género, la cantidad total de denuncias de femicidio, la cantidad total de casos con denuncia previa de violencia de género, el porcentaje de casos con denuncia previa respecto a la cantidad total de denuncias y, por último, la mayor cantidad de denuncias de femicidio registradas en un municipio.

Es muy probable que, posteriormente, se requiera modificar el sistema para que genere más prestaciones (nuevas estadísticas, restricciones, etc.).

Especificar, diseñar e implementar en Python, un programa (con todas sus funciones) que satisfaga el requerimiento citado anteriormente, respetando las condiciones de presentación que se describen a continuación.

Condiciones de presentación

*Antes de concluir el plazo preestablecido por cada fecha límite, se debe presentar el trabajo en 3 entregas (opcionalmente 4), con las características que se describen más abajo. Cada entrega constará de **un documento en procesador de textos** en el que se informen pasos de la metodología y, además, en la 3ra. entrega (y posible 4ta.), **cuatro (o más) archivos en formato .py** (código Python). El **documento en procesador de textos** debe contener:*

- a) Una **carátula** con los siguientes datos:
 - Carrera y Nombre de la materia
 - Año, Cuatrimestre, Nro. de Comisión y Docentes
 - Apellido y nombres de cada estudiante (hasta 6 integrantes por grupo y no menos de 5)
- b) **Informe** con el análisis del problema y los pasos de la solución requeridos en las sucesivas entregas. Dichos pasos están enunciados en la “metodología para la construcción de programas pequeños”, que puede leerse en el ítem 2.1 de la Unidad 2 del apunte “Informática. Introducción a la Programación usando Python como herramienta”.

✓ **1era. Entrega** Fecha límite de presentación: **sábado 1 de mayo**

A. **Analizar el problema.** Entender profundamente cuál es el problema que se trata de resolver, incluyendo el contexto en el cual se usará. Una vez analizado el problema, asentar el análisis por escrito.

B. **Especificar la solución.** Describir qué debe hacer el programa (incluyendo validaciones), sin importar el cómo. Determinar cuáles son los datos de entrada que se proveen, cuáles son las salidas que se deben producir. Documentar dicha especificación por escrito.

✓ **2da. Entrega** Fecha límite de presentación: **sábado 5 de junio**

A y B. **Análisis del problema y Especificación de la solución** (corregidos y aprobados)

C. **Diseñar la solución.** Describir cómo resolver el problema, cuáles son los algoritmos y las estructuras de datos que se usarán, cuáles son las funciones en que se divide el problema y cuál es la relación entre ellas y/o con el programa (datos de entrada y salidas respectivas). Analizar posibles variantes y tomar las decisiones usando como dato de la realidad el contexto en el que se aplicará la solución y los costos asociados a cada diseño. Asentar por escrito el diseño, asegurándose de que esté completo, incluyendo la Especificación de cada función.

✓ **3ra. Entrega** Fecha límite de presentación: **sábado 3 de julio**

A, B y C. **Análisis del problema, Especificación y Diseño de la solución** (corregidos y aprobados)

D. **Implementar el diseño.** Traducir a un lenguaje de programación (en nuestro caso Python) el diseño que fue elegido en el punto anterior. Enviar, a la tarea correspondiente, en formato .py el código Python (un módulo con el programa principal y, por cada integrante del grupo, el respectivo módulo con funciones).

E. **Probar el programa.** Diseñar un conjunto de pruebas que verifique el comportamiento de cada una de sus partes por separado, y también la correcta integración entre ellas. Utilizar la función print (“[DEBUG]”) como instrumento de diagnóstico para depuración. Asentar por escrito, casos que garanticen que todas las componentes estén probadas con datos “correctos”, “incorrectos” y “de borde”.

✓ **Entrega opcional** Fecha límite de presentación: **lunes 12 de julio**

F. **Mantener el programa.** Adaptar el programa para dar respuesta al nuevo requerimiento. Especificar, implementar y probar las nuevas funcionalidades del programa. Informe completo (A, B, C, D, E y F).

Todos los archivos (informe en procesador de textos y archivos de código Python) deberán ser **enviados a la tarea correspondiente en el Campus**, ya sea **por uno de los integrantes** del equipo, o **por todos y cada uno de los integrantes**, según corresponda a la respectiva entrega.