

Trabajo Práctico Grupal Obligatorio – Segundo cuatrimestre 2021

Fecha límite de 1era. entrega: **09 / 10 / 2021**

Fecha límite de 2da. entrega: **08 / 11 / 2021**

Fecha límite de 3ra. entrega: **04 / 12 / 2021**

Fecha límite de entrega opcional: **13 / 12 / 2021**

Problema a resolver

El Programa Transversal de Políticas de Género y Diversidad de la UNDAV, evalúa la posibilidad de **implementar un sistema para producir un informe estadístico semestral de las denuncias de Situaciones de Violencia Machista, Acoso Sexual y Discriminación por Identidad de Género y/o Vivencia de la Sexualidad, ocurridos en el ámbito de la Universidad**. 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 (no anterior a 2021)**, exigiendo reingresos hasta asegurar su validez y el **semestre (1 o 2)**. Los datos serían ingresados por un usuario responsable del sistema que, por cada denuncia recibida, deberá ingresar un **número de expediente (identificador del caso)**, la **fecha de la denuncia**, el **género** auto percibido de la persona **denunciante**, el **claustró** al que pertenece, el **tipo (o los tipos)** de situación/es vivenciada/s, el **género de la persona denunciada y el claustró al que pertenece**. 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** y el semestre del informe y, para cada denuncia, la **validez de su fecha** (que sea posible y esté comprendida en el período informado), la **validez de género (m: mujer, v: varón, x: otro)** y de **claustró (e: estudiante, n: no-docente, d: docente, g: graduado)**, la **clasificación en, al menos uno, o más, de cuatro tipos** de situación vivenciada (hechos de violencia sexual, hechos de acoso sexual, hechos con connotación sexista, comportamientos y acciones de violencia). Muestre en pantalla, simultáneamente, los datos válidos de la denuncia. Cuando un dato no sea válido, el programa debe solicitar su reingreso hasta asegurar la validez.
2. Contabilice los casos de denuncia en que la situación vivenciada clasifique, simultáneamente, en más de un tipo (por ej., “hechos de connotación sexista” y “comportamientos y acciones de violencia”, etc.), como así también, los casos de denuncias entre pares del mismo claustró (estudiante/estudiante, docente/docente, no-docente/no-docente, graduado/graduado).
3. Determine y muestre (con descripciones expresivas): la **cantidad total de denuncias informadas**, el **porcentaje de casos de denuncia que clasifiquen en más de un tipo de situación vivenciada** (cociente entre la cantidad total de denuncias que clasifica en más de un tipo de situación vivenciada y la cantidad total de denuncias informadas, multiplicado por cien), **la cantidad de denunciantes por género**, la **cantidad de denunciantes por claustró**, la **cantidad total de denuncias entre pares del mismo claustró y, por último, el mayor número de expediente ingresado**.

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, 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 9 de octubre**

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: **lunes 8 de noviembre**

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 4 de diciembre**

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 13 de diciembre**

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.