

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL ROSARIO

TRABAJO PRÁCTICO Nº 2: CONSIGNAS

ALGORITMOS Y ESTRUCTURAS DE DATOS

INTRODUCCIÓN	3
DESARROLLO	3
Módulo 0: inicialización	5
Módulo 1: Estudiantes	5
Módulo 2: Moderadores	7
BONUS TRACK 1: EDADES	7
BONUS TRACK 2: MATCHEOS COMBINADOS	8

Introducción

En este TP2 vamos a introducir el concepto de variables **del tipo array** (unidimensionales y bidimensionales), junto con Procedimientos y Funciones con pasaje de parámetros.

Deben realizar la codificación en lenguaje Python de lo requerido para este TP2.

Junto al código de Python, indicar claramente la declarativa de variables utilizadas, en especial, las referidas a las variables del tipo array (dimensiones, tamaño de cada arreglo y su tipo de datos).

Tener en cuenta al implementar los arrays en Python, respetar el marco teórico de la cátedra sobre arrays, es decir, deben tener **dimensión y tamaño fijo** (no modificable en tiempo de ejecución del programa) y mismo tipo de dato en cada elemento.

Antes de comenzar con este TP, por favor leer el documento **Enunciado General** de manera completa, para saber de qué va el negocio, y entender el modelo de datos requerido. Es obligatoria también la lectura de las **preguntas frecuentes**, para evitar prácticas que resten puntos al momento de la entrega.

Desarrollo

En esta segunda etapa deberán realizar lo siguiente:

- Al comenzar con la ejecución del programa, hacer que el usuario elija la opción “logueo” o “registrarse”.
 - Los estudiantes estarán guardados en un arreglo, y deberán **darse de alta** al comienzo de la ejecución del programa. Deberán ser mínimo 4 estudiantes y máximo 8.
 - Los moderadores estarán guardados en un arreglo, y deberán **darse de alta** al comienzo de la ejecución del programa. Deberán ser mínimo 1 moderador y máximo 4.
 - **Para acceder a la sección de logueo, deberá haber aunque sea 1 moderador y 4 estudiantes cargados.**
- Hacer el menú para cada tipo de usuario (Estudiante y Moderador).
 - Pedir un **nombre de usuario (email) y contraseña** al operador. **Verificar** que existan el usuario y su clave guardados en el sistema, y de acuerdo al tipo de usuario que está ingresando, presentarle el menú correspondiente.
 - Reusar el control de claves desarrollado en el TP1, permitiendo hasta 3 intentos como máximo.

- Tener en cuenta que, además de ingresar un par usuario / contraseña correctos, también se debe chequear que el estado del usuario sea “ACTIVO” (string). Caso contrario, el login NO será correcto.
- Tanto el ID de cada estudiante como de cada moderador va a ser siempre un número entero auto-incremental, que comienza en 0.

Si el usuario que ingresa (por medio de su nombre de usuario y su clave) coincide con un usuario guardado, y el mismo es del tipo **Estudiante**, deberán mostrar el menú completo correspondiente a un usuario de ese tipo:

1. Gestionar mi perfil
 - a. Editar mis datos personales
 - b. Eliminar mi perfil
 - c. Volver
2. Gestionar candidatos
 - a. Ver candidatos
 - b. Reportar un candidato
 - c. Volver
3. Matcheos
 - a. Ver matcheos
 - b. Eliminar un matcheo
 - c. Volver
4. Reportes estadísticos
0. Salir

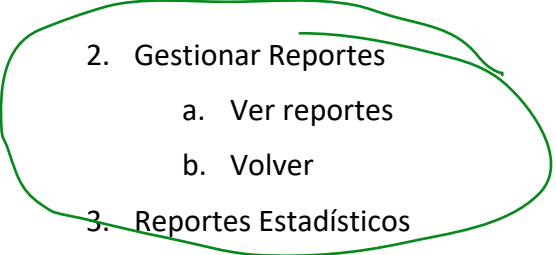
Tener en cuenta que apenas se ejecuta el programa, solamente se verán las opciones:

1. Gestionar mi perfil
2. Gestionar candidatos
3. Matcheos
4. Reportes estadísticos
0. Salir

Y luego las opciones con letras o sub-opciones se mostrarán luego de elegir alguna opción principal. Al elegir la opción “Volver” en cualquiera de los sub-menús, se volverá al menú anterior.

Si el usuario que ingresa (por medio de su nombre de usuario y su clave) coincide con un usuario guardado, y el mismo es del tipo **Moderador**, deberán mostrar el menú completo correspondiente a un usuario de ese tipo:

1. Gestionar usuarios
 - a. Desactivar usuario
 - b. Volver

- 
- 2. Gestionar Reportes
 - a. Ver reportes
 - b. Volver
 - 3. Reportes Estadísticos

Módulo 0: inicialización

Desarrollar un procedimiento que se llame al comienzo de la ejecución de nuestro programa. El mismo deberá recorrer la matriz de likes, y llenarla de 0s y 1s de manera aleatoria. De esta manera, cada vez que el programa se ejecute, vamos a tener cargada cierta interacción entre los estudiantes.

Módulo 1: Estudiantes

Para esta segunda etapa, solamente se pedirán realizar los **módulos:**

- 1. **Gestionar mi perfil**
 - a. Editar mis datos personales
 - b. Eliminar mi perfil
- 2. **Gestionar candidatos**
 - a. Ver candidatos
 - b. Reportar un candidato
- 4. **Reportes Estadísticos**

El resto de las opciones deberán mostrar un **cartel aclaratorio** que diga “En Construcción”.

Descripción de los módulos:

1.a Editar mis datos personales: en esta opción, el estudiante logueado deberá poder modificar la información ingresada para su fecha de nacimiento, sexo, biografía, hobbies y cualquier otro campo que crean relevante.

1.b Eliminar mi perfil: mostrar un cartel aclaratorio, preguntando confirmación si realmente se desea eliminar el perfil. En caso de elegir la opción “sí”, cambiar el **estado del usuario por inactivo**, y volver a la pantalla de login del trabajo.

2.a Ver candidatos: al ingresar a esta opción, se deben mostrar todos los estudiantes. Mostrar nombre, fecha de nacimiento, edad, biografía y hobbies. Una vez mostrada toda la información, permitir ingresar en una variable me-gusta el nombre del estudiante con el cual le gustaría en un futuro hacer un Matcheo. Verificar que el nombre sea correcto.

Para mostrar la edad, sabiendo que la fecha de nacimiento se guarda como string en el formato YYYY-MM-DD siendo YYYY el año, MM el mes y DD el día (ej: el 1 de diciembre de 2002 sería 2002-12-01), calcular y mostrar la edad del candidato.

Se recomienda utilizar una matriz del tamaño de la cantidad de usuarios, lleno con 0s y 1s o de Trues y Falses, donde guardemos los likes. Por ejemplo, consideremos la matriz L:

	0	1	2	3	4	5	6	7
0	0	1	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0
2	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

En este caso, $L[0, 1] = 1$, pero $L[1, 0] = 0$. Esto quiere decir que el usuario 0 le dio like al usuario 1, pero el usuario 1 NO le dio like al usuario 0, por ende, no hay match.

Por otro lado, $L[2, 1] = 1$ y $L[1, 2] = 1$, por ende, hubo un match entre ambos usuarios. Recordemos que sólo los estudiantes pueden ser candidatos y darse like entre ellos

2.b Reportar un Candidato: al ingresar a esta sección, se deberá pedir el ID o nombre de usuario (cualquiera de los dos) de un candidato, y el motivo por el cual se está reportando al mismo. Validar que el candidato exista, y dar de alta un reporte con el ID del usuario actual como reportante, el ID del usuario reportado, el motivo dado y el estado inicial 0 (ver sección 2.a de moderadores para obtener más información acerca de los distintos estados de un reporte).

4. Reportes Estadísticos mostrar:

- con qué porcentaje de todos los candidatos posibles (distintos a nosotros) se matcheo ambas veces
- a cuántos le pusimos "me gusta" pero no nos han devuelto el match
- cuántos estudiantes nos dieron "me gusta", a los cuáles nosotros no le hemos dado "me gusta todavía". Mostrarlo de la siguiente forma:

Matcheados sobre el % posible: 50%

Likes dados y no recibidos: 2

Likes recibidos y no respondidos: 3

Módulo 2: Moderadores

Para esta segunda etapa, solamente se pedirán realizar los módulos:

1. Gestionar usuarios
 - a. Desactivar usuario
 - b. Volver
2. Gestionar Reportes
 - a. Ver reportes
 - b. Volver

1.a Desactivar un Usuario: se ingresa el ID o el nombre de usuario, se busca si el usuario existe, y luego se muestra una pantalla idéntica a la de “Eliminar mi Perfil”.

2.a Ver reportes: los reportes pueden tener como estado 0, 1 y 2. 0 quiere decir que el reporte todavía no ha sido visto por un moderador o un administrador, 1 quiere decir que el reporte ha sido tomado y el reportado ha sido desactivado, y 2 quiere decir que el reporte ha sido ignorado.

Se mostrarán los reportes los cuáles tanto el usuario reportante como el reportado están activos, y que tengan un estado = 0. Por cada reporte, se preguntará cómo se quiere proceder: ignorar reporte o bloquear al reportante. En caso de seleccionar la primer opción, el estado del reporte será actualizado a 2. En caso de seleccionar la segunda opción, el estado del reporte será actualizado a 1, y el usuario reportado será desactivado (su estado será inactivo).

Bonus Track 1: edades

Un desarrollador anterior a nosotros creó un reporte con las edades de los estudiantes que utilizan la aplicación. El mismo solamente devuelve un arreglo desordenado con las posibles edades de los mismos (sin repetir). El resultado para nuestros 6 estudiantes fue un arreglo con los siguientes valores estáticos:

`edades = [21, 18, 20, 19, 23, 24]`

Teniendo en cuenta el resultado de dicho reporte, se pide un algoritmo que primero ordene dicho arreglo, y luego realice una búsqueda secuencial para detectar si hay algún “hueco” en las edades detectadas. Llámese hueco a un número faltante para poder tener una secuencia

autoincremental. Por ejemplo, en la secuencia 1,2,3,5 tendríamos 1 hueco, y el elemento faltante sería 4.

Bonus Track 2: matcheos combinados

Sacando desde nuestro arreglo la cantidad de usuarios estudiantes cargados, calcular la cantidad de matcheos posibles, suponiendo que cualquier estudiante podría matchear con cualquier estudiante que no sea sí mismo, independientemente del sexo.