



TRABAJO FINAL INTEGRADOR PRIMER CUATRIMESTRE

En el trabajo integrador de este primer cuatrimestre, se debe desarrollar el análisis de una situación problemática a solucionar con programación en el TFI del segundo cuatrimestre.

La situación problemática a elegir corresponde al análisis para realizar una aplicación que pueda ser de utilidad para una organización comunitaria:

- Merendero
- Escuela
- Grupo misionero
- Otro a determinar por el grupo y tener la aprobación de la cátedra

CONFORMACIÓN DE GRUPOS

El trabajo es grupal y de hasta 4 personas (**de la misma comisión**). Una vez formados los grupos, se debe informar a la cátedra los integrantes por medio de un formulario que se encontrará en el aula virtual y Grupos de Telegram, se incluirán los nombres de los integrantes y quien es el integrante designado por el Grupo para subir el Trabajo.

CONSULTAS

Las consultas referentes a los enunciados de cada una de las etapas se atenderán por medio del foro y consultas por los medios habilitados (Zoom, Microsoft Teams, Telegram, etc...). Se dispondrán horarios en los cuales los profesores atenderán las inquietudes y/o problemas de cada uno de los grupos.

PAUTAS DE ENTREGA

Las entregas asociadas se realizarán **exclusivamente** por medio de la tarea *TP AED – Primer Trabajo Grupal*, creada para tal fin en el campus virtual de la materia. El archivo a subir debe tener el nombre:

Comisión_apellido1_apellido2_apellido3_apellido4.pdf

Comisión, seguida por los apellidos de los integrantes separados por guiones.

Cada entrega la realizará **sólo uno de los integrantes del grupo**, indicando claramente en el documento el nombre de cada uno de los integrantes y su email.

DOCUMENTACIÓN A ENTREGAR

Deberán **entregar a través de la tarea** el archivo pdf con el análisis completo de la situación problemática a resolver para la organización comunitaria.



Composición del Archivo PDF

Elección de la Organización Comunitaria:

Elegir la organización comunitaria de acuerdo con ello realizar una entrevista a fin de que puedan determinar las necesidades de la misma y traducir la mismas en requerimiento programables, indicando los factores que por los que hicieron esta elección.

Situación Problemática:

Este módulo se debe explicar la situación problemática a resolver y los métodos de recolección de la información (entrevista, documentos, etc.).

Estructura de la Información:

Determinar las entidades que se encuentran en la situación problemática elegida y realizar la estructura de la información de las mismas indicando solo:

- Entidad
- Atributos (formato de cada uno)

I. INTERFAZ DE USUARIO

El diseño visual de la interfaz es libre y se pueden utilizar elementos gráficos (logos, colores) para hacer las pantallas de la aplicación más atractivas.

Al ingresar a la aplicación, se presenta el Menú Principal, donde el usuario debe tener una cuenta creada para poder iniciar una sesión. A continuación, se muestra el esquema de opciones que debe respetar el menú principal.

Se debe diseñar la interfaz de usuario teniendo como obligación que entre las opciones debe tener:

- Mostrar la información xxxxx (a través de árboles o grafos)
- Listado ordenado de xxxx (e la programación se deberá elegir el método a usar)

Ejemplo:

```
Modulo Consultorio Veterinario
=====
1.- Iniciar Sesión
2.- Visualizar Lista de Espera de Turnos (informe)
3.- Registrar Evolución de la Mascota

4.- Cerrar la aplicación.

Ingrese una opción: _
```

```
Modulo del Asistente
=====
1.- Iniciar Sesión
2.- Registrar Mascota
3.- Registrar Turno
```



4.- Listado de Atenciones por Veterinario y Fecha

5.- Cerrar la aplicación.

Ingrese una opción: _

Módulo Administración

=====

1.- Registrar Veterinario

2.- Registrar Usuario Asistente

3.- Atenciones por Veterinarios

5.- Ranking de Veterinarios por Atenciones

6.- Cerrar la aplicación.

Ingrese una opción: _

II. GESTIÓN Y AUTENTICACIÓN DE USUARIOS

En este ejemplo, el usuario tiene la posibilidad de crear un nuevo usuario y la opción de crear un nuevo asistente. Para esto el programa le solicitará nombre de usuario y contraseña, condiciones que deberán tener en cuenta al diseñar su interfaz de usuario.

Nombre de usuario: Quedará definido por una cantidad mínima de 6 caracteres y máxima de 10, los cuales podrán ser letras, números y/o símbolos del conjunto {+,-,/,*,?,!,.}. Deberá cumplir con los siguientes requisitos

:

- Ser único para cada usuario registrado.
- Comenzar con una letra minúscula.
- Tener al menos 2 letras mayúsculas.
- Tener como máximo 3 dígitos.

Ejemplos de nombres de usuario incorrectos: AbC123 (no cumple con b), pTS!1234 (no cumple con d), g178Mci (no cumple con c), mARtin123gomez (tiene mas de 10 caracteres).

Ejemplos de nombres de usuario correctos: mARtin12, jo97!AR.

Contraseña: Su conformación no podrá darse al azar, sino que deberá respetar lo siguiente:

- Deberá contener al menos una letra mayúscula, una letra minúscula y un número.
- No podrá contener ningún carácter de puntuación, ni acentos, ni espacios. Sólo caracteres alfanuméricos.
- Deberá tener entre 6 y 32 caracteres.
- No debe tener más de 3 caracteres numéricos consecutivos.
- No debe tener 2 caracteres consecutivos que refieran a letras alfabéticamente consecutivas (ascendentemente). Este criterio es válido tanto para letras mayúsculas, minúsculas o combinación de ambas.



Ejemplos de contraseñas mal formadas: Ach32 (no cumple con c), dorit1234 (no cumple la regla a ni la d), sA;gotAP.10 (no cumple con b), aBuel123 (no cumple con e).

Ejemplo de contraseñas bien formadas: Achus32, 125Af89, aLejo123, DORIt045, 4aC2sa.

Nota: Tanto en el nombre de usuario como en la contraseña deben distinguirse mayúsculas y minúsculas.

III. PERSISTENCIA

Para no perder la información entre ejecuciones, es necesario mantener los datos de las cuentas de usuario registradas. Para cumplir con este objetivo deberá implementarse la persistencia de la información manejando archivos que se describirán más adelante en este apartado. La estrategia sugerida para cumplir con este objetivo es:

- Cuando se inicia la aplicación, se cargan los datos en estructuras de datos en memoria.
- Durante la ejecución de la aplicación se actualizan estas estructuras.
- Al terminar cada sesión se agrega el contenido de esta al archivo correspondiente.

Los datos de las cuentas de usuario se almacenarán en el archivo **Usuarios.dat**

Este archivo será de formato binario y guardarán variables diseñadas en una estructura de datos a fin de representar la información asociada a las cuentas de usuarios (es decir, nombre y contraseña). Cada vez que se inicie la aplicación, se utilizará la información almacenada en este archivo para validar el inicio de sesión y/o la creación de un nuevo usuario.

La estructura de la información para los Usuarios debe ser la siguiente:

Usuarios	
Usuario	Char [10]
Contraseña	Char [10]
Apellido y Nombre	Char [60]