

<u>UTN – FRMDP Mar del Plata TUP - Laboratorio 2 - Trabajo Práctico Final</u> <i>Propuesta Tema: Laboratorio</i>	<i>Integrantes del grupo</i>	<i>Nota</i>
--	-------------------------------------	--------------------

Introducción

Con el propósito principal de integrar todo lo aprendido en la materia laboratorio 2, a los estudiantes que así lo deseen, se les permite proponer una temática particular elegida por los integrantes de su grupo.

El sistema deberá contemplar la administración de **como mínimo, tres tipos de datos (una de ellas debe ser una estructura compuesta)**, realizando la persistencia de estas en archivos binarios.

Fundamentación

El valor pedagógico de la propuesta se apoya en el aprendizaje colaborativo **(se formarán grupos de 2 alumnos mínimo y 4 máximo)**. Es importante aclarar que la cantidad mínima de tipos de datos a usar por grupo es 3 tipos de datos. Todo debe implementarse mediante librerías, de manera tal que el main del proyecto solo llame al menú Ppal.

Para que este tipo de proyectos sea más exitoso, deben llevarse a cabo desde un enfoque que facilite alcanzar los Objetivos de Aprendizaje propuestos.

Una de las ideas centrales es desarrollar competencias profesionales y preparar al futuro programador para el mundo laboral y el trabajo en equipo.

En un ambiente de aprendizaje colaborativo, los estudiantes:

- Construyen conocimiento y en lugar de recibirlos en forma pasiva;
- Se involucran y se comprometen directamente con el descubrimiento de nuevo conocimiento;
- Se exponen a puntos de vista alternativos e ideas contrapuestas, de forma tal que pueden sacar sus propias conclusiones y así transformar conocimientos y experiencias previas y de esta manera comprender con mayor profundidad;
- Transfieren conocimientos y habilidades a nuevas situaciones o circunstancias;
- Se responsabilizan y apropian tanto de su aprendizaje continuo de contenidos curriculares, como del desarrollo propio de competencias;
- Los estudiantes colaboran para el aprendizaje del grupo y el grupo colabora en el aprendizaje individual de estos.

Objetivos

De aprendizaje:

- Gestión de archivos binarios.
- Recursividad.
- Listas enlazadas, simples o dobles
- Árboles binarios.
- Pilas y Filas implementadas con Listas (simples o dobles).
- Estructura de datos compuestas. (arreglo de listas, listas de listas, árboles de listas, etc.)
- Trabajar en forma colaborativa.

Metodológicos:

- Ser capaces de trabajar en un proyecto complejo, aplicando técnicas de desarrollo de software.
- Lograr integrar contenidos de otras asignaturas.
- El grupo deberá ir mostrando el avance sobre el trabajo en clase.

Modo de Evaluación del Trabajo Práctico

- Se establece el desarrollo de un trabajo práctico final, brindando una fecha límite de entrega de este: **Según planificación de cada comisión**
- Si el sistema está codificado en su totalidad y funciona correctamente, se considerará aprobado con una nota mínima de 6.
- Si el sistema no está codificado en su totalidad se considerará desaprobado y el grupo presentará la versión final en la fecha de recuperatorio.
- Para aprobar de forma directa (promocionar) es necesario entregar y presentar en la primera fecha.
- En la fecha de recuperatorio deberá cumplir las pautas mínimas establecidas precedentemente para la aprobación de la instancia recuperatoria. Vale aclarar que no puede aprobar de manera directa.
- **Es obligatorio la presentación de este trabajo para aprobar la materia.**

Pautas Generales

Como metodología de trabajo, se requiere crear un documento de texto (o carpeta) en Google Drive que será compartido a todos los miembros del grupo (y también al equipo docente, publicando el link vía campus virtual en el foro correspondiente), con el fin de plasmar los avances del proyecto de forma de construir la siguiente documentación a entregar:

Informe final - Documentación a entregar: [10 puntos]

- Diario de Trabajo/bitácora del proyecto.
- Matriz de soluciones: Que problema tuvieron y cómo lo resolvieron.
- Diagrama de estructuras: Esquema de las estructuras utilizadas y sus relaciones.
- Un archivo ejecutable para poder correr desde cualquier pc.
- Manual del Usuario.
- Documentación en formato pdf e impresa en papel tamaño A4, fuente Verdana de tamaño 10 para el texto en general y 12 para títulos y subtítulos (a criterio de cada equipo). El interlineado deberá ser el normal, con espaciado anterior y posterior de cada párrafo en 0 y sangrías solamente de 1era línea en caso de necesitarla. La alineación de los párrafos será justificada excepto en aquellos casos como títulos y subtítulos en los cuales debe ser centrada y/o Izquierda. El encabezado deberá incluir el nombre de la materia con su año de cursada y tecnicatura, el nombre del proyecto y el logo de la UTN. En el pie de página, incluir los integrantes y la numeración de la página. Consultar por posibles modificaciones ANTES de imprimir.

Asimismo, deberán crear [un Repositorio en GitHub o un enlace en Google Drive](#) para distribuir las tareas entre los integrantes del grupo y trabajar de forma organizada. **A medida que avancen con el desarrollo del trabajo, realizarán capturas de pantalla y las adjuntarán al Diario de trabajo.**

Al momento de efectuar la entrega del proyecto, deberán agregar todo el material digital solicitado precedentemente.

La función principal - Main() y menús: [10 puntos]

IMPORTANTE: LA NAVEGABILIDAD DEL PROGRAMA.

El sistema deberá contar con una presentación amigable con el usuario, construir menús de acceso a las diferentes estructuras y funcionalidades del sistema, **y de manera directa o indirecta, permitir probar todas las funciones desarrolladas.**

El desarrollo del sistema deberá ser ordenado, identificando con comentarios cada una de las funciones realizadas, explicando brevemente lo que realizan. Se tendrá en cuenta, al momento de evaluar, la prolijidad del código y la organización de los módulos. Se recomienda agrupar los mismos por funcionalidad. Usar nombres significativos en todo momento y en todo el desarrollo.

El sistema deberá brindar tres tipos de acceso (perfiles):

- Para administradores
- Para profesionales del Laboratorio (bioquímicos o técnicos)
- Para administrativos del Laboratorio

Detalle de estructuras y funcionalidad básicas: [50 puntos]

Importante, el sistema debe trabajar en memoria principal consumiendo las estructuras dinámicas generadas para tal fin. Al inicio del sistema se consultarán los archivos binarios y se actualizará esa información al finalizar la ejecución del programa. La catedra se reserva la posibilidad de introducir algún cambio/modificación en alguno de los aspectos del proyecto según lo considere necesario y/o conveniente. En tal caso, se informara a cada grupo oportunamente.

A continuación, se brinda una breve narrativa acerca del "circuito administrativo" del sistema en cuestión de manera de entender con mayor precisión, los alcances del proyecto a desarrollar.

*"En un hospital de niños se tiene entre otros servicios, el de Laboratorio Central. Su función es la de ingresar y posteriormente procesar todas las muestras de los pacientes que son requeridas por el personal médico de la institución. Es decir que el profesional médico, una vez que atiende a un paciente, solicita que al mencionado se le realice un conjunto de diferentes prácticas de laboratorio (**Ingreso de Laboratorio**) para luego evaluar resultados.*

*Para cada uno de los **Ingresos de Laboratorio** que se realicen por cada paciente, se deben almacenar un Nro de Ingreso, fecha de ingreso, fecha de retiro de resultados, el paciente en cuestión (dni), la matricula del profesional que solicita ese ingreso de laboratorio, un indicador que permita saber si fue dado de baja o no y finalmente cada una de las practicas asociadas a ese ingreso. A su vez, de cada **Practica de Laboratorio** se debe almacenar un numero de práctica, un indicador que permita saber si fue dado de baja o no y un nombre de práctica.*

Con respecto a las Practicas de Laboratorio asociadas a cada ingreso de cada paciente; estas NO deben formar parte de los datos del Ingreso, sino que deben ser relacionadas con uno de ellos. Por lo tanto, se debe poder almacenar en una entidad diferente, cada una de las practicas de un ingreso de manera que esta entidad sea el nexo entre los ingresos y sus prácticas (y permita el almacenamiento de los resultados de estas); así como los Ingresos son el nexo entre las practicas que incluyen y el paciente para el cual fueron solicitadas.

*Por lo tanto, la entidad que soporte las **Practicas por Ingreso**, deberá almacenar: un Identificador de practica por ingreso, el Ingreso en cuestión, cada una de las practicas asociadas al mismo y el resultado obtenido para cada una de ellas.*

*En cuanto a los datos personales que se deben poder gestionar de los **Pacientes**; se requiere: su apellido y nombre, edad, dni, dirección, teléfono y un indicador que permita saber si fue dado de baja.*

*Por último, hay que recordar que los **Empleados del Laboratorio** se clasifican en 3 diferentes categorías: administrativos, técnicos y bioquímicos y cada uno de ellos según su categoría tiene asociado un perfil, el cual le habilita o no acceso a las diferentes funcionalidades del sistema. Para cada empleado se deben almacenar: DNI, apellido y nombre, teléfono, usuario y contraseña y finalmente el tipo de acceso o perfil. Es decir: administrador, profesional de laboratorio (técnico o bioquímico) o administrativo".*

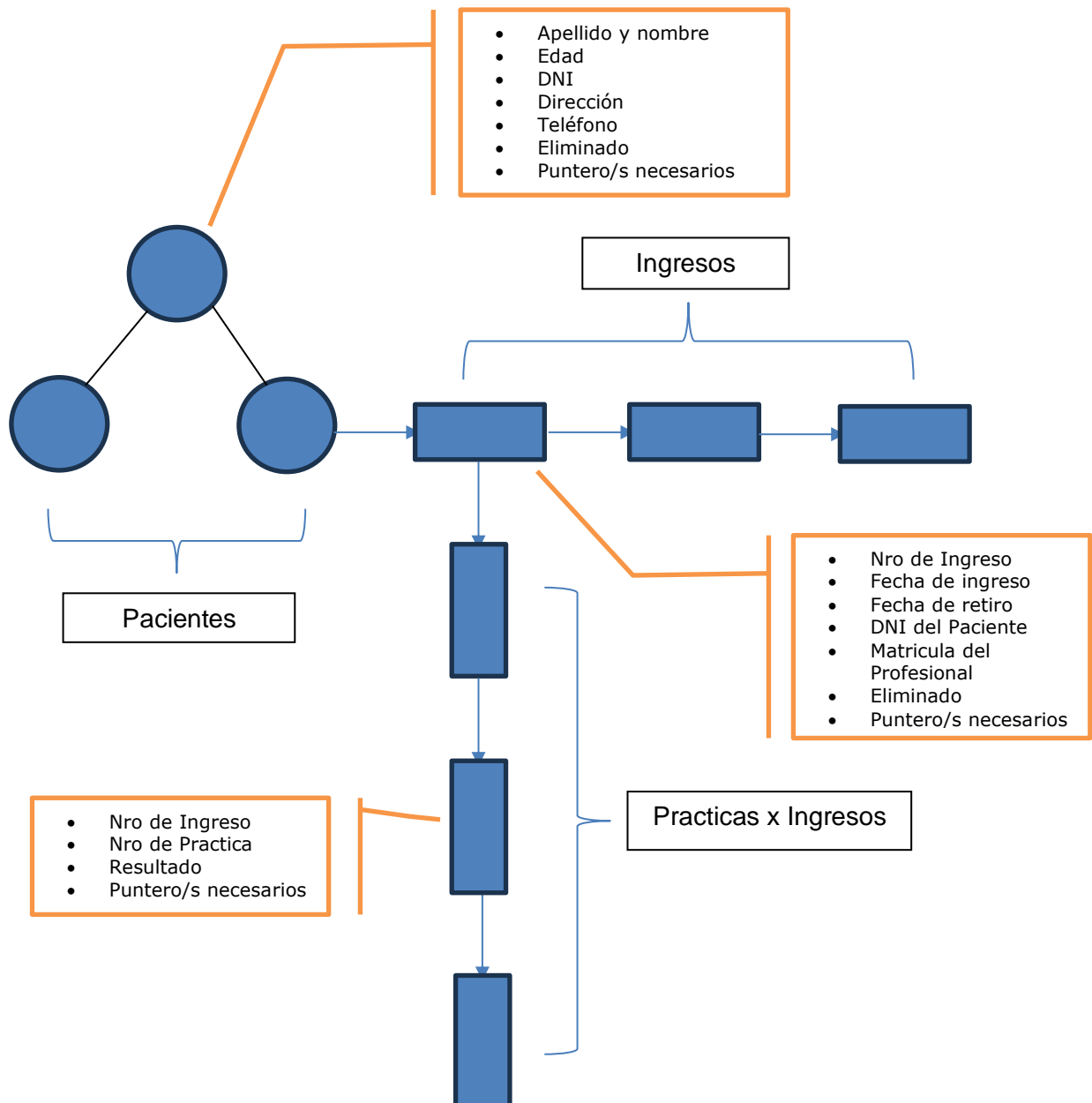
Teniendo en cuenta la narrativa anterior junto a lo explicado y detallado en clases, se **propone/sugiere** la información que debería ser reflejada en estructuras junto a la funcionalidad asociada a las mismas y su correspondiente soporte en disco (archivos) para lograr la persistencia de la información. En caso de creerlo conveniente, es posible la implementación de estructuras/archivos adicionales, pero en todo momento la cátedra debe ser informada de tal situación para contar con la correspondiente aprobación. No es posible reducir la cantidad de datos que almacena una estructura, aunque si pueden adicionarse y/o modificarse (siempre con aprobación de la cátedra).

<p>Estructura: Ingreso de Laboratorio (struct ingresos)</p> <ul style="list-style-type: none"> • Nro de Ingreso (int) • Fecha de ingreso (string[10]) – por defecto, la actual • Fecha de retiro (string[10]) • DNI del Paciente (int) • Matricula del Profesional solicitante (int) • Eliminado (int) – por defecto: 0 • Puntero/s necesarios 	<p>Funciones Asociadas</p> <ul style="list-style-type: none"> • Alta_de_ingreso: debe verificarse la existencia del paciente antes de dar el alta y debe incluir al menos 1 practica de laboratorio • Modificación_de_ingreso: solo pueden modificarse las fechas y matricula del solicitante • Baja_de_ingreso: deben actualizarse en cascada (dar de baja), las practicas asociadas al ingreso
<p>Estructura: Práctica de Laboratorio (struct practicas)</p> <ul style="list-style-type: none"> • Nro de Practica (int) • Nombre de Practica (string[30]) • Eliminado (int) – por defecto:0 	<p>Funciones Asociadas</p> <ul style="list-style-type: none"> • Alta_de_practica • Modificacion_de_practica: solo su nombre • Baja_de_practica: solo si no fue incluida en ningún ingreso
<p>Estructura: Paciente (struct pacientes)</p> <ul style="list-style-type: none"> • Apellido y nombre (string[40]) • Edad (int) • DNI (int) • Dirección (string[30]) • Teléfono (string[15]) • Eliminado (int) – por defecto:0 • Puntero/s necesarios 	<p>Funciones Asociadas</p> <ul style="list-style-type: none"> • Alta_de_paciente: debe verificarse que el mismo no exista antes del alta • Modificacion_de_paciente • Baja_de_paciente: debe verificarse que no tenga ingresos asociados
<p>Estructura: Empleado de Laboratorio (struct empleados_laboratorio)</p> <ul style="list-style-type: none"> • DNI (int) • Apellido y nombre (string[40]) • Usuario (string[20]) • Contraseña (string[20]) • Perfil (string[20]) 	<p>Funciones Asociadas</p> <ul style="list-style-type: none"> • Alta_de_empleado: debe verificarse que el mismo no exista antes del alta • Modificacion_de_empleado • Baja_de_empleado
<p>Estructura: Practicas X Ingreso (struct pracXingreso)</p> <ul style="list-style-type: none"> • Nro de Ingreso (int) • Nro de Practica (int) • Resultado (string[40]) • Puntero/s necesarios 	<p>Funciones Asociadas</p> <ul style="list-style-type: none"> • Alta_de_pxi: verificar que existan el nro de ingreso y de practica • Modificacio_de_pxi: puede afectar a un nro de practica y al resultado solamente • Baja_de_pxi

Con respecto a los TDA intervinientes:

- Los pacientes deben ser almacenados en un árbol binario de búsqueda.
- Los ingresos deben almacenarse en una lista (simple o doble). Los cuales se desprenderán de cada uno de los nodos del árbol.
- Las practicas por ingreso deben ser almacenadas en una lista (simple o doble), las cuales se desprenderán de cada uno de los nodos de la lista de ingresos.

Gráficamente:



Detalle de funcionalidad adicional: [30 puntos]

A continuación, se detalla la funcionalidad más allá de la básica, que debe incluir el proyecto:

- Ingresos:
 - Listado general de ingresos (sin detalle de prácticas), con detalles del paciente y posibilidad de filtrar por "fecha desde" y "fecha hasta". Es decir que la fecha del ingreso, este incluido en ese rango. Ordenado de manera descendente por fecha de ingreso.
 - Consulta de 1 ingreso en particular incluyendo las practicas del mismo con sus respectivos nombres y los resultados obtenidos. El ingreso debe poder ser filtrado por nro de ingreso, fecha de ingreso o por dni del paciente.
- Pacientes:
 - Listado general de pacientes ordenado por apellido y nombre.
 - Consulta de 1 paciente en particular, filtrado por dni.
 - Listado de ingresos por paciente (sin detalle de prácticas).
 - Consulta de 1 ingreso en particular, filtrado por nro o fecha de ingreso, con detalle de prácticas y resultados obtenidos.
- Prácticas de Laboratorio:
 - Listado general de practicas de laboratorio ordenadas por nombre.
 - Listado de practicas que "comiencen con". En este caso debe poder filtrar todas las practicas cuyo nombre comiencen con lo seleccionado por el usuario. Ejemplo: si se elige la combinación "**he**", deberían mostrarse practicas como: "**he**mograma o **he**patograma" etc.
- Empleados de Laboratorio:
 - Listado general de empleados ordenado por apellido y nombre.
 - Consulta de 1 empleado en particular, filtrado por dni.
 - Nota: solo en caso de estar logueado un usuario "administrador", debe poder verse la contraseña que usa el empleado para ingresar al sistema. Caso contrario mostrar "*****" (asteriscos) como contraseña.

Consideraciones generales para tener en cuenta:

1. En aquellos casos donde pudo haberse eliminado por ejemplo un ingreso o un paciente etc, cuando un usuario con perfil de administrador solicite listados, los eliminados deben poder verse de todos modos; no siendo así en caso contrario.
2. Mantener la consistencia y/o integridad de los datos. Es decir que, por ejemplo, no debería poder eliminarse una practica que haya sido incluida en al menos 1 ingreso.
3. Realizar las validaciones que considere necesarias al momento de dar de alta, modificar o eliminar información de cualquier tipo.

Para la evaluación de este apartado, se tendrán en cuenta los algoritmos que administran las estructuras de datos, la responsabilidad que se les asigna a cada una de las estructuras, la persistencia de estas en archivos binarios y la lógica de control aplicada en la interacción de dichas estructuras en el sistema.

Tabla de puntuación:

Obtenido	10	20	30	40	50	60	70	80	90	100
Nota	1	2	2	4	5	6	7	8	9	10
Observación	Desaprobado					Aprobado				