



INTRODUCCIÓN A LA PROGRAMACIÓN 2025/26

Práctica final (grupal)

Clínica estética antiaging tUReJuveneCes



Este documento es el guion de la práctica final de la asignatura Introducción a la Programación que se realizará en grupo, con un total de **3 o 4 miembros por grupo**.

Los grupos se crearán en Aula Virtual obligatoriamente, estudiantes que no estén en un grupo no tendrán derecho a evaluación de la práctica grupal.

La práctica se realizará con un compilador, se recomienda “Lazarus”, accesible desde MyApps.

Objetivos

- Tipos de datos compuestos: registros y arrays de registros.
- Gestión de datos almacenados en ficheros de texto.
- Modularización mediante el uso de unidades y subprogramas.
- Uso de estructuras de control, validación de entradas y ordenación de datos.

Entrega

La práctica se entregará por completo en Aula Virtual. Un **responsable del grupo** será el encargado de subir el código y memoria, será la única persona que se comunicará con el profesor para resolver dudas. Se dispondrá de plazo para la entrega hasta el **viernes 9 de enero** a las 23:55 horas.

Se entregará:

- Un único archivo con el código del programa principal del ejercicio con la extensión **.PAS**

- Otro fichero .PAS que desarrolle la unidad donde se encuentra toda la funcionalidad
- Un fichero con un ejemplo de la clínica exportado y con el historial utilizado como pruebas en la elaboración de la práctica
- Una memoria explicativa que explique todo lo desarrollado durante la práctica en formato .PDF. La memoria debe ser autoexplicativa, con imágenes de ejecución en la consola y debe incluir: portada con el nombre del equipo, miembros del equipo, hojas numeradas y un índice.

Normas

Es obligatoria la entrega de prácticas y respetar los plazos.

IMPORTANTE:

- **Una práctica plagiada, hecha con Chatgpt o similar implicará un suspenso y una falta grave que puede implicar un suspenso.**
- La lectura de variables por teclado NUNCA podrá hacerse en los subprogramas de la Unidad.
- El profesorado puede requerir la asistencia a su despacho para probar la práctica si fuera necesario, siendo obligatoria la existencia de todo el equipo. Dicho profesor puede preguntar a cualquier persona del equipo y solo esa persona puede contestar.
- Los estudiantes que, de manera individual, no cumplimenten el cuestionario de valoración de sus compañeros no tendrán derecho a evaluación de la práctica grupal.
- **En esta práctica, al ser subprogramas muy definidos, salvo para la opción 2 que se deberán leer los datos en el programa principal. El resto de subprogramas podrán leer datos por teclado en su interior (read o readln).**

Ejercicio:

Se desea construir una aplicación que permita gestionar la información de los pacientes que acuden a una clínica. Para ello, se definirá un tipo de dato compuesto (registro) que contenga la información personal, médica y administrativa de cada paciente.

La aplicación permitirá realizar operaciones básicas como añadir nuevos pacientes, listar los existentes, buscar por historia clínica, filtrar por seguro médico y calcular la facturación total.

Por cada paciente se almacena la siguiente información:

- Nombre y apellidos del
- Edad (entre 16 y 120) del paciente
- sexo
- historiaclínica: Código identificador formado por 9 letras
- Ingreso : Fecha de ingreso (día, mes, año) creada de manera automática.
- Seguro: si tiene o no seguro médico.
- Facturación: total facturado al paciente

El programa mostrará al usuario un **menú con las siguientes opciones:**

1. **Cargar datos desde un fichero de texto** (tpacientes.txt)

2. **Añadir un nuevo paciente**
3. **Listar todos los pacientes registrados**
 - Al listar los pacientes, se deberán mostrar **ordenados alfabéticamente por nombre** usando el algoritmo de **intercambio directo** (bubble sort).
4. **Mostrar pacientes con seguro médico**
5. **Buscar paciente por número de historia clínica**
6. **Calcular y mostrar el total facturado por la clínica**
7. **Guardar los datos actuales en el fichero de texto y salir del programa**

El programa mostrará el menú y realizará la opción escogida hasta que el usuario escriba el valor 7.

A continuación, se explica cada una de las opciones:

1 Cargar datos desde un fichero de texto (pacientes.txt)

- Esta opción permite **leer desde un fichero de texto** los datos previamente almacenados.
- El programa deberá:
 - Abrir el fichero pacientes.txt en modo lectura.
 - Cargar los datos en el programa.
- Validar el número de pacientes (máximo 2000).

2 Añadir un nuevo paciente

- Sigue al usuario todos los campos del paciente (nombre, edad, sexo, ingreso, seguro, facturación). El código de historial clínico se genera de manera automática. El código se genera con letras de A a Z aleatorias hasta completar un código de nueve letras.
- Almacena el nuevo paciente.
- Crea automáticamente su fichero individual de historial, vacío, con el nombre: <codigoHistoriaClinica>.txt → Ejemplo: ABCDADAZE.txt.
- Validar los datos introducidos y comprobar que no se supera el límite de 2000 pacientes.

3 Listar todos los pacientes registrados

- Muestra todos los pacientes por pantalla con formato legible ordenados usando el **algoritmo de intercambio directo (bubble sort)**.
- **Después de mostrar el listado**, debe ofrecer un **submenú adicional con estas opciones**:
 - ¿Qué desea hacer ahora?
 - Ver. Ver historial de un paciente
 - Agregar. Agregar entrada al historial de un paciente

- Salir. Volver al menú principal

Subopciones de la opción 3:

- **Ver historial:**
 - Sigue el código de historia clínica.
 - Abre el fichero correspondiente (XXXXXXX.txt) y muestra su contenido línea a línea.
- **Añadir al historial:**
 - Sigue el código de historia.
 - Pide al usuario una línea de texto (descripción de diagnóstico, seguimiento, etc.).
 - Añade esa línea al final del fichero XXXXXX.txt.

El contenido del historial puede incluir entradas de tipo:

10/12/2025 - Diagnóstico: hipertensión leve.
12/01/2026 - Control: presión arterial estable.

4 Mostrar pacientes con seguro médico

- Filtra y muestra solo los pacientes con seguro = true.
- También deben mostrarse **ordenados alfabéticamente** por nombre (intercambio directo).
- Tras mostrar el listado, se mostrará el **mismo submenú de la opción 3** y permitirá las mismas opciones: ver, agregar o salir de la opción para volver al menú principal.

5 Buscar paciente por número de historia clínica

- Sigue un código de historia clínica.
- Si existe, muestra todos los datos del paciente y de su historia clínica.
- Si no, informa de que no se ha encontrado.

6 Calcular y mostrar el total facturado por la clínica

- Recorre todos los pacientes y suma el valor del campo facturación.
- Muestra el total acumulado por pantalla.

7 Guardar los datos actuales en el fichero de texto y salir

- Sobrescribe el fichero pacientes.txt con los datos actuales.
- Cada línea debe contener los campos en el formato original, separados por |. Ejemplo:

Maria Gomez|22|F|HCRRTQAET|15|5|2024|TRUE|243.50
Juan Perez|32|M|MAAEWAJZW|15|5|2025|TRUE|2434.50

- Finaliza la ejecución del programa tras mostrar un mensaje de confirmación.
- **Nota:** Los ficheros individuales de historial no se tocan aquí (se modifican en opciones 3 y 4).

Consejos:

- Revisad las funciones que tenéis para strings
- Recordad que el tipo boolean es un TRUE o FALSE, pero no se puede leer por teclado, pues sería un string 'TRUE' o 'FALSE'. O, lo que es lo mismo, si lo leéis de un fichero de texto será también un string, tendréis que transformarlo a Boolean.
- En este caso el assign para los ficheros de historial de cada paciente, sí se deberían de hacer en los subprogramas, pero por que son específicos para cada caso, no para todo el programa completo, como pudiera pasar con el fichero general de pacientes.txt.