

## **Proyecto Final**

El proyecto final es un Trabajo Práctico Integrador que propone utilizar integralmente la mayor cantidad de estructuras de datos y conceptos desarrollados a lo largo del año. Con ese objetivo les planteamos este caso:

*Una estación meteorológica registra a diario una serie de datos como la temperatura máxima, la mínima, la humedad media, la presión atmosférica, la dirección predominante del viento y su velocidad, también la precipitación acumulada. Estos valores se obtienen a diario. La estación analiza la información para obtener series de datos que le permitan alimentar modelos de predicción del tiempo. Nos solicitan que desarrollemos un programa para poder almacenar los registros diarios y procesar los valores almacenados para la elaboración de los pronósticos del tiempo.*

*El registro de datos diario tendrá la siguiente estructura:*

*regdiario=<ddmmyyyy E Z, tmax E Z, tmin E Z, HUM E (0..100), PNM E Z, DV E (0..360), FF E Z, PP E (0..1000), borrado E Lógico >*

*El significado de éstos campos es el siguiente:*

*ddmmyyyy ésta es la fecha del registro, dd es el día, mm el mes, yyyy el año, por ejemplo el día 25 de octubre tendría el valor: 25102022, el primero de noviembre tendría el valor 01112022, etc. (Aclaración: es una forma muy común en el ámbito meteorológico usar este formato para las bases de datos de los registros climáticos, en lugar de un tipo compuesto).*

*tmax y tmin, son las temperatura máximas y mínimas ocurridas ese día, solo se registran valores enteros (sin decimales).*

*HUM es la humedad promedio del día registrado, este valor se expresa en números enteros y puede variar desde 0 a 100 (por que la humedad se expresa en por ciento).*

*PNM este valor es la presión atmosférica, se registra el promedio del día son valores en hectopascal, los valores varían entre 900 y 3500.*

*DV este valor representa la dirección del viento que se toma en grados y en valores enteros de 0 a 360, se toma la dirección que tenía el viento cuando se produjo el máximo registro del día.*

*FF es la velocidad del viento, se registra la velocidad máxima ocurrida en el día.*

*PP precipitación pluvial acumulado, se registra el total acumulado en 24 hs*

*borrado para la baja lógica del registro.*

*Para el tratamiento de los datos, el programa a desarrollar debe poder crear y mantener un archivo con los registros diarios. Los datos de un mismo año están en un único archivo.*

*A partir de los datos registrados el mismo programa debe permitir obtener determinados resultados que vamos a explicar a continuación:*

*Para ordenar el trabajo a desarrollar presentamos el menú de opciones que el programa debería presentarle al usuario:*

- Alta de un registro diario
- Suprimir un registro diario
- Modificar un registro, se busca por la fecha
- Mostrar todos los registros activos
- Buscar registro de un día dado y mostrar todos los parámetros
- Listar el día o días de máxima temperatura en lo que va del año
- Listar el día o días de máxima precipitación en lo que va del año
- Listar las fechas de mayor a menor velocidad de viento
- Realizar una copia de seguridad del archivo del año en curso
- Salir

*Las opciones de alta, suprimir, modificar y mostrar se ejecutarán directamente sobre el archivo. La opción suprimir consiste en aplicar borrado lógico.*

*Para la **búsqueda** de datos de un día dado, se pasarán los registros diarios del archivo a un arreglo y la búsqueda se realizará sobre el arreglo. Para buscar se debe implementar una función que devuelva el índice del arreglo dónde se encuentra el registro buscado y sino está que devuelva -1. Esta función debe ser recursiva.*

*Para listar el día o días con **máxima temperatura**, se creará una lista (Lista Simplemente Encadenada) de registros con la misma máxima temperatura (en caso que haya más de un día con la misma máxima temperatura) y se mostrará las fechas de los días en que ocurrieron esas máximas temperaturas.*

*Para listar las fechas de mayor a menor **velocidad de viento**, se pasarán los registros diarios del archivo a un arreglo y se ordenará por velocidades del viento mayor a menor (usar un método estable) y se informarán las fechas y el viento correspondiente de los diez primeros días de mayores velocidades ordenados de mayor a menor.*

*Para listar el día o días con **máxima precipitación**, se ordenará el arreglo por **precipitaciones** de menor a mayor (usar un método estable distinto al anterior) y se informarán las fechas y las precipitaciones correspondientes de los diez primeros días de mayores precipitaciones ordenados de mayor a menor.*

*Para el manejo de **arreglos** y de **listas** deberá implementar las acciones y funciones desarrolladas a lo largo del año (función Vacía(), función Llena(), acción Insertar(), acción Ordenar(), acción Mostrar(), etc.*

*La opción de **copia de seguridad** consiste en crear un archivo nuevo (con otro nombre) con sólo los registros no borrados del archivo original.*

## Algunas consideraciones para ayudar al diseño:

- Para cumplir con el formato del diseño:

*regdiario=<ddmmyyyy E Z, tmax E Z, tmin E Z, HUM E (0..100), PNM E Z, DV E (0..360), FF E Z, PP E (0..1000), borrado E Lógico >*

El registro en C se definirá de la siguiente forma:

```
typedef struct {  
    long int ddmmyyyy;  
    int tmax;  
    int tmin;  
    int HUM;  
    int PNM;  
    int DV;  
    int FF;  
    int PP;  
    int borrado;  
} Tregdiario;
```

En la carga de datos deberá controlarse que la información se encuadre en el dominio previsto. Por ejemplo DV que es la dirección del viento, debe ser un número entero entre 0 y 360. Si está fuera de ese rango, el valor se debe pedir de nuevo. Y así con los demás campos.

- La cátedra proveerá archivos con registros cargados para efectuar pruebas. No obstante recomendamos cargar registros con valores aleatorios (de un año dado) para que hagan sus propias pruebas.
- Para cada opción del menú se debe diseñar un módulo adecuado.
- Todas las opciones del menú deben efectuarse sobre registros activos, los borrados no deben ser utilizados.
- El programa comenzará solicitando el nombre del archivo con el cuál va a trabajar, si el archivo no existe lo creará.

## ¿Qué, cómo y cuándo?

- El proyecto final debe resolverse en grupos de 2 alumnos. Se aceptarán grupos de 3 alumnos en caso que sea debidamente justificado. La fecha límite de conformación de grupos es el 07 de Noviembre.
- Los días 07, 09 y 14 de Noviembre se dispondrán para consultas generales sobre el proyecto final, en los horarios habituales de clases prácticas. En las clases del 07 y el 09 los grupos deberán mostrar los Diseños (algoritmos) que dan solución al problema planteado.
- **Consultas en Laboratorio**: 09 Noviembre de 11 a 13 y 16 a 18hs en el Laboratorio 101 del Pab 2 (UNRC). Por EVELiA se informarán otros días de consulta posibles.
- **IMPORTANTE**: Quienes no formen parte de algún grupo a la fecha límite serán considerados en condición de libre.
- El tiempo estimado de realización del Trabajo Práctico Integrador es de aproximadamente 20 horas en total.
- El proyecto final debe incluir:
  - 1) un documento en formato PDF con el Diseño (algoritmo).
  - 2) un archivo con la Implementación (programa fuente en C).
- El proyecto final se entregará mediante la Sección de Actividades del aula virtual.
  - Se fija el lunes 14 de Noviembre como fecha límite para la entrega del algoritmo (Diseño) con la resolución del proyecto. El algoritmo debe respetar las convenciones y estar debidamente comentado.
  - Se fija el miércoles 16 de Noviembre como fecha límite para la entrega del código fuente (Implementación) con la resolución del proyecto. El código fuente debe respetar las convenciones y estar debidamente comentado.
  - El responsable del grupo será el encargado de adjuntar el algoritmo (Diseño) y el código fuente (implementación) en el aula virtual.
- El proyecto es obligatorio y debe aprobarse (igual que un parcial). No tiene recuperatorio.
- El grupo realizará defensa del proyecto en caso que la cátedra lo considere necesario (en fecha a definir en cada caso).