



Instituto Tecnológico De Costa Rica

Estructuras de Datos

Proyecto Estructuras Dinámicas Lineales

Johan Rodriguez y Gabriel Alfaro y Aaron Gonzalez

San Carlos

Octubre

2022

Proyecto Estructuras Dinámicas Lineales

Índice

Descripción del problema	2
Solución del problema	4
Resultados	5
Conclusión	7
Recomendaciones	7

Descripción del problema

El problema planteado para el primer proyecto programado del curso de Estructuras de Datos del segundo semestre de 2022 consiste en la realización de un programa que registre datos del clima y tiempo de manera diaria, dentro del cual se pueda consultar información para conocer los reportes diarios y promedios mensuales y anuales del clima. Los objetivos de este proyecto consisten en el desarrollo de un programa en el lenguaje C++ (utilizando las ventajas de la orientación a objetos), la implementación de estructuras de datos dinámicas que permitan el acceso y ordenamiento de la información, y la implementación de un nuevo tipo de datos estudiado en clase: los punteros.

El programa del proyecto debe de presentar la capacidad de ingresar información sobre personas, lluvias, regiones, lugares y efimeridades. La inserción de personas y efimeridades debe de darse mediante una lista doble ordenada por nombre y por fecha, respectivamente. El grupo de lluvias y regiones debe de insertarse por medio de una lista simple, con inserción al inicio de esta. Finalmente, la inserción de lugares está preestablecida para darse mediante una lista circular, con inserción al inicio, y los registros de tiempo deben de hacerse en una sublista simple ordenada por fecha.

Dentro del programa se deben crear punteros programados para relacionar diversas listas entre sí, de forma que cada región se pueda relacionar con distintos lugares, y que cada lugar pueda relacionarse con los registros de tiempo. Además, se debe de

relacionar cada persona con los registros del clima, mediante una sublista simple cuya inserción sea al inicio de esta. Cada uno de los objetos, que poseen características propias, deben de poder crear una relación con aquellos objetos correspondientes.

Existen reglas para la inserción y actualización de la información de las listas. El programa debe de permitir la inserción de los datos de todas las listas y sublistas, así como la modificación de estos datos. Debe permitir la eliminación de nodos en todas las listas y sublistas, y debe de validar que no se ingresen datos repetidos o erróneos (un ejemplo de esto sería el tiempo; las 26 horas, o 30 de febrero). Además, en el programa debe de haber por defecto un mínimo de 10 datos dentro de cada lista y sublista.

Las consultas del proyecto deben presentar información procesada a base de los datos obtenidos en las listas y las sublistas. El usuario debe poder consultar por la fecha completa con la salida de “sol” más temprana de un determinado año, así como el ocultamiento más tardío en el mismo año. El programa ha de presentar también las fechas que generan la mayor diferencia en minutos de la salida del sol en un determinado año, además de imprimir el mes que más extremos de lluvia y sequía ha tenido de un lugar y año específico. En el caso de un empate entre meses se deben imprimir todos los meses que tienen el empate máximo. Por último, el programa debe de poder determinar e imprimir la persona que más registros de tiempo posee relacionados.

Con respecto a los reportes, el usuario debe de poder consultar por la impresión de la información de todas las listas y su información relacionada, los horarios de la salida y la puesta del sol separado por meses de un año, la precipitación mensual promedio de cada lugar y región en un año determinado (mediante reportes separados), y las variables climatológicas de todos los lugares de una región específica durante un periodo de tiempo establecido. Además, debe determinar e imprimir si hay periodos en un año de cambios en tipos de días de lluvia, en los cuales se debe indicar la fecha de inicio y final del cada periodo encontrado (los periodos deben ser iguales o superiores a siete

días seguidos). Finalmente, se debe de imprimir los porcentajes de la clasificación de la lluvia, los días de lluvia y los extremos de temperatura de cada mes para un año y lugar específico.

Solucion del problema

Para la solución del problema planteado en el primer proyecto programado del curso de Estructuras de Datos del segundo semestre de 2022, se utilizaron las estructuras de listas, sublistas, clases y plantillas para el manejo y acceso de la información. Se decidió realizar un diagrama de clases UML que caracterice a cada lista y sublista con sus atributos respectivos, así como la relación entre las listas.

Como se puede observar en la Figura 1, se tienen las diversas listas y sublistas establecidas en la descripción del problema, y, añadido a éstas, se encuentra una clase denominada “Proxy”. Este objeto consiste en un intermediario entre las distintas listas que ayuda a conectar mediante punteros los datos de los nodos que requieren crear relaciones entre sí. El objeto Proxy posee una estructura de enlace a los nodos que conecta, además de que es una sublista, por lo que tiene un nodo siguiente. Esta integración al programa permite relacionar en ambas direcciones dos listas, de forma que cada clase posee un nuevo Proxy que acepta varios tipos de datos, permitiendo adquirir todos los atributos requeridos para las consultas y los reportes.

En cuanto a la lógica para realizar el programa, se estableció una división de tareas para cada integrante del grupo, con el objetivo de cumplir las tareas establecidas antes de la fecha de entrega del proyecto. Se trabajó el sistema mediante la aplicación de programación modular, con diversos archivos interconectados que componen el programa.

Para la resolución de consultas y reportes, se construyó el programa de manera que itere por las listas para encontrar y extraer los datos necesarios, de acuerdo con lo

solicitado, y se conformaron en conjuntos de datos que serían posteriormente enviados a imprimir. Sin embargo, algunas de las consultas o reportes requieren de algunas fechas especificadas o información determinada, por lo que se verificó para estos casos que se acepte la entrada de datos por parte del usuario dependiendo del interés de este, validando la entrada de datos para que concuerde con los parámetros e imprimiendo el conjunto de datos requerido.

En el caso de las consultas en específico, se accedió a los datos guardados en las listas y se buscó, mediante la validación de parámetros de la información, los extremos consultados por el usuario. En la consulta de los extremos de la salida y entrada del sol, se pide al usuario por el año del cual se desea consultar el dato, y, dentro de los datos que posean en el atributo de fecha el valor del año ingresado, se extrae la hora y fecha del momento más temprano y más tarde de la salida y ocultamiento del sol, respectivamente. Luego se procede a imprimir los datos en la interfaz del usuario, concluyendo con la consulta.

Los procesos realizados para imprimir los reportes conllevan una mayor complejidad que aquellos requeridos para las consultas. Por esta parte, se buscaron en las listas del programa los datos pertinentes al reporte consultado por el usuario. Estos datos se ordenan en conjuntos de datos que son procesados para ser validados y confirmar su relevancia en el reporte, mediante cálculos y parámetros. Finalmente, se muestran al usuario en el formato establecido por los lineamientos del proyecto.

Resultados

Los resultados presentados por el programa demuestran una interfaz de usuario agradable para el cliente que utilice la plataforma, con textos grandes y claros, y todas las opciones de lo que puede realizar el programa se presentan al usuario en la página principal. La creación, modificación y muestra de datos a partir de las estructuras implementadas en el sistema fueron probadas y completadas. A partir de esto se generaron

los reportes y las consultas, que recolectan los datos pertinentes a clima y lo previamente registrado en el sistema y los muestran de manera correcta en la interfaz, y se presentan las tablas con los datos ordenados en el formato establecido.

En cuanto a los aspectos que dificultaron la finalización del programa, hubo confusión en diversos cálculos de reportes, debido a que la forma de guardado de datos complicó el acceso a los mismos para expresar la información en el formato debido. Principalmente el segundo y el sexto reporte se postraron complejos.

Con respecto al segundo reporte, relacionado con la impresión de los horarios de la salida y puesta del sol de diversos días en un mes separados por año, existió una dificultad al recolectar todos los datos necesarios para imprimir el reporte. Según lo establecido en los lineamientos del proyecto, la información recopilada es presentada en listas de gran magnitud, por lo que los conjuntos de datos creados para su solución terminaron siendo de gran tamaño y difíciles de accesarlos y mantenerlos ordenados para ser mostrados.

Para el reporte número seis, que consistía en la determinación e impresión de periodos en un año específico de cambios en tipos de días de lluvia, así como la indicación de la fecha de inicio y final de cada periodo encontrado, en donde los periodos deben ser mayores a 7 días, se requería una gran cantidad de efimeridades distintos de un año. Esto complicó el ordenamiento de los parámetros y la organización de los mismos en rubros predeterminados por los lineamientos.

A pesar de las complicaciones, el programa planteado para el proyecto programado dio resultados positivos y se logró cumplir con todas las consultas y los reportes. Además, se implementó y probó la interfaz del usuario y todas su opciones, para finalmente determinar el correcto funcionamiento del programa.

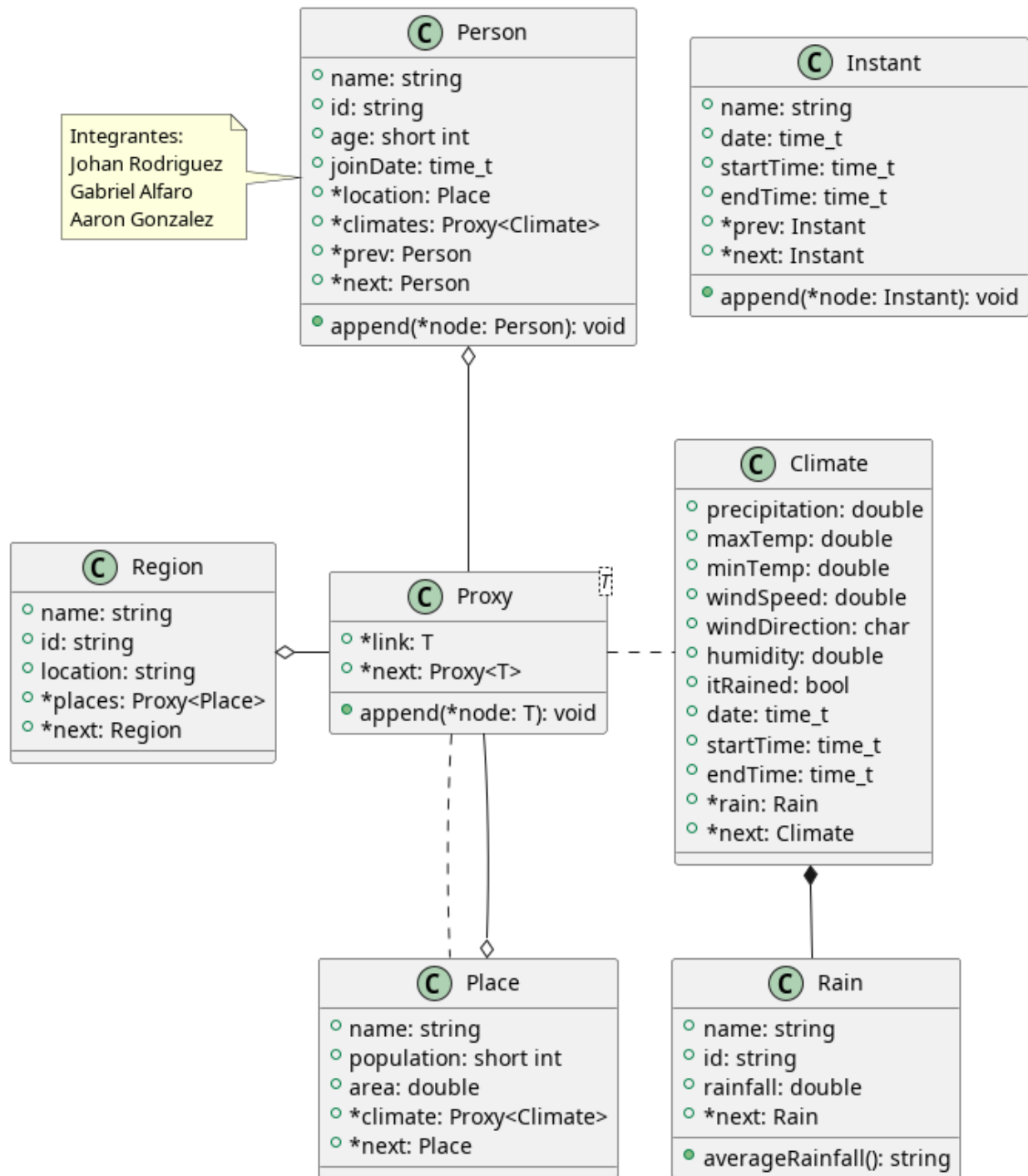
Conclusion

Se puede concluir que el enfoque orientado a objetos ha probado ser una herramienta de suma utilidad en cuanto a la programación de código grande que requiera de interconexiones entre cada objeto de su programa, en este caso, las conexiones mediante punteros de las listas y sublistas. Además, se puede indicar con seguridad que la creación de un método de punteros, en el caso de interés la creación de la clase Proxy, resultó ser óptimo para la resolución del problema planteado, interconectando las listas de una forma eficaz y que permitió el acceso y búsqueda de información por este medio.

Recomendaciones

Existen pocas recomendaciones para este proyecto, sin embargo, son de suma importancia. Para el proyecto se utiliza un archivo Makefile y el linker Mold para ensamblar el programa. Por lo tanto, si no se cuenta con esas dependencias, no será posible compilar el código fuente para generar el archivo binario. Es por ende necesario obtener las dependencias para poder correr el código de manera óptima.

Por otro lado, el desarrollo y diseño total del programa fue hecho en el sistema operativo Linux, por lo tanto, si se tienen errores o fallos en cuanto a dependencias, bibliotecas y demás funcionalidades, en otro sistema operativo, se recomienda corroborarlo, ya que es muy probable que se deba a un error de compatibilidad entre sistemas operativos y no un error propio de las estructuras y clases definidas en este programa.

**Figura 1***Diagrama de clases del sistema*