Universidad de San Carlos de Guatemala
Escuela de Ingeniería en Ciencias y Sistemas, Facultad de Ingeniería
Introducción a la programación y computación 2, 2do. Semestre 2022.

PROYECTO #1

202111563– Jose Pablo Menard Pimentel

Resumen

En este proyecto se presenta un problema en el cuál se necesita un programa que revise las rejillas con una muestra de las células de un paciente en específico, las cuales siguen un algoritmo para cambiar su estado, ya sea a “contagiado” o “sano”, luego de verificar el estado de las rejillas después de ciertos periodos, el programa debe deducir la gravedad de la enfermedad, mientras muestra gráficamente cómo estos las células van cambiando de estado mientras transcurren los periodos indicados, posteriormente se debe enviar un reporte del estado general del paciente. Todo esto se debe lograr utilizando y aplicando de manera práctica todos los conceptos básicos de estructuras de datos aprendidas en la clase, específicamente se plantea el concepto de “listas enlazadas”, lo cual sirve para la lógica principal del problema que es presentado en el proyecto, también serán implementados conceptos gráficos para generar la parte visual del proyecto, utilizando específicamente la herramienta “graphviz”, que sirve, entre otras cosas, para mostrar nodos gráficamente, esto será necesario para mostrar el proceso de cambio de células.

Palabras clave

- Estructuras
- Listas
- Células
- Métodos
- Clases

Abstract

In this project a problem is presented in which a program is needed to check the grids with a sample of cells from a specific patient, which follow an algorithm to change their status, either to "infected" or "healthy" , after verifying the state of the grids after certain periods, the program must deduce the severity of the disease, while graphically showing how these cells change state while the indicated periods elapse, then a report of the general state must be sent of the patient. All this must be achieved using and applying in a practical way all the basic concepts of data structures learned in the class, specifically the concept of "linked lists" is proposed, which serves for the main logic of the problem that is presented in the project. , graphic concepts will also be implemented to generate the visual part of the project, specifically using the “graphviz” tool, which serves, among other things, to show nodes graphically, this will be necessary to show the process of changing cells.

Key words clave

- Structures
- Lists
- Cells
- Methods
- Classes

Introducción

La base primordial de la realización de este proyecto radica en la Programación Orientada a Objetos, debido a que la creación de clases, instancias, atributos y métodos vienen de este paradigma de programación, así que es estrictamente necesario tener claras las bases de dicho campo. También es necesario tener una idea clara de los conceptos referentes a estructura de datos, debido a que estas aplicaciones son la columna vertebral de este proyecto. Es importante recalcar que, fuera del ámbito práctico que exige el proyecto, una de las partes más importantes del mismo es la lógica que tiene que implementar el desarrollador para resolver el problema, ya que, sin un análisis preciso y estructurado del planteamiento del problema, no es posible llegar a una solución clara, por lo que es necesario entender por completo cada inciso que se pide en el programa, para que sea más fácil realizar la parte práctica.

Desarrollo del tema

a) Python:

En términos técnicos, Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Es muy atractivo en el campo del Desarrollo Rápido de Aplicaciones (RAD) porque ofrece tipificación dinámica y opciones de encuadernación dinámicas.

Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

Por tanto, esto reduce el costo de mantenimiento y de desarrollo del programa porque permite que los equipos trabajen en colaboración sin barreras significativas de lenguaje y experimentación.

Además, soporta el uso de módulos y paquetes, lo que significa que los programas pueden ser diseñados en un estilo modular y el código puede ser reutilizado en varios proyectos. Una vez se ha desarrollado un módulo o paquete, se puede escalar para su uso en otros proyectos, y es fácil de importar o exportar.

Por otro lado, uno de los beneficios más importantes de Python es que tanto la librería estándar como el intérprete están disponibles gratuitamente, tanto en forma binaria como en forma de fuente.

Tampoco hay exclusividad, ya que Python y todas las herramientas necesarias están disponibles en todas las plataformas principales. Por lo tanto, es una opción multiplataforma, bastante tentadora para los desarrolladores que no quieren preocuparse por pagar altos costos de desarrollo.

Python es un lenguaje de programación de propósito general, que es otra forma de decir que puede ser usado para casi todo. Lo más importante es que se trata de un lenguaje interpretado, lo que significa que el código escrito no se traduce realmente a un

formato legible por el ordenador en tiempo de ejecución.
Este tipo de lenguaje también se conoce como «lenguaje de scripting» porque inicialmente fue pensado para ser usado en proyectos sencillos.
El concepto de «lenguaje de scripting» ha cambiado considerablemente desde su creación, porque ahora se utiliza Python para programar grandes aplicaciones de estilo comercial, en lugar de sólo las simples aplicaciones comunes.

Una encuesta realizada en 2019 entre los usuarios de Python indicó que los usos más populares eran para el desarrollo web y el análisis de datos. Sólo alrededor del 6% de los encuestados lo utilizaron para el desarrollo de juegos o el desarrollo de aplicaciones. (Centro de Formación Técnica Para la Industria, 2020) Las aplicaciones para las cuales Python es útil son muchas, unas de las más famosas son:

Data science: Tras la creación de los motores numéricos como “Pandas” o “NumPy”, Python está desbancando MATLAB, un lenguaje utilizado por científicos a la hora de trabajar con un gran número de datos. La razón es la misma que en los anteriores apartados; la sencillez y la potencia para trabajar con un gran número de datos, unidos al gran número de bibliotecas existentes, hacen que Python sea ideal para este tipo de tareas.

Inteligencia artificial: Seguro que durante los últimos años has oído hablar muchísimo de la inteligencia artificial (IA). Gran parte de su avance se debe a Python. Su facilidad de escritura y su robustez han convertido a Python en el aliado perfecto de la IA. Su capacidad de plasmar ideas complejas en pocas líneas, unidas al gran número de frameworks existentes, han hecho que Python sea uno de los lenguajes de programación que están impulsando a la IA.

Blockchain: La base de datos distribuida Blockchain, conocida mundialmente por ser la base sobre la que se sustentan las criptomonedas, también funciona muy bien junto a Python. Como lenguaje versátil, seguro y rápido, es muy útil para formar cadenas de bloques, e incluso permite a los desarrolladores crear una cadena de bloques sencilla en menos de 50 líneas de código, haciendo sencillo algo muy complejo.

Machine learning: El machine learning o aprendizaje automático es otra de las tecnologías que está cambiando el mundo tal y como lo conocemos. La robótica y la IA son ahora capaces de aprender por sí mismas a medida que van procesando más y más datos. De esta forma, obtienen información cada vez más relevante que les permite tomar las decisiones adecuadas. Por supuesto, Python es también muy eficaz en este campo, en el tratamiento de datos eficaz es esencial.

Desarrollo web: Python también permite desarrollar webs complejas en menos líneas de código, lo que permite que estas sean más ligeras y optimizadas. Django es uno de los frameworks de Python más populares de la actualidad, que puede ser utilizado para crear webs dinámicas y muy seguras. Python es también muy utilizado para hacer scraping, es decir, para obtener información de todo tipo de webs, tal y como lo hacen Netflix, Instagram o Pinterest. (Santander | Becas, 2021)

Su filosofía fue la misma desde el primer momento: crear un lenguaje de programación que fuera muy fácil de aprender, escribir y entender, sin que esto frenara su potencial para crear cualquier tipo de aplicación. En aquellos años, el hardware que había no permitía tal cosa, y es por eso por lo que Python ha resurgido durante los últimos años, porque el avance de la tecnología ha permitido alcanzar el objetivo inicial de este lenguaje de programación adelantado a su tiempo.

b) POO

La programación orientada a objetos se basa en el concepto de crear un modelo del problema de destino en sus programas. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. Python es un lenguaje orientado a objetos. Los objetos definidos en Python tienen las características siguientes:

-Identidad. Cada objeto debe ser distinguido y ello debe poder demostrarse mediante pruebas. Las pruebas is e is not existen para este fin.

-Estado Cada objeto debe ser capaz de almacenar el estado. Para este fin, existen atributos, tales como variables de instancias y campos.

-Comportamiento. Cada objeto debe ser capaz de manipular su estado. Para este fin existen métodos.

Python incluye las características siguientes para dar soporte a la programación orientada a objetos:

Creación de objetos basada en clases. Las clases son plantillas para la creación de objetos. Los objetos son estructuras de datos con el comportamiento asociado.

Herencia con polimorfismo. Python da soporte a la herencia individual y múltiple. Todos los métodos de instancias de Python son polimórficos y se pueden alterar temporalmente mediante subclases.

Encapsulación con ocultación de datos. Python permite ocultar los atributos. Cuando se ocultan los atributos, se puede acceder a los mismos desde fuera de la clase únicamente mediante los métodos de la clase. Las clases implementan métodos para modificar los datos. (IBM, 2021)

Durante años, los programadores se han dedicado a construir aplicaciones muy parecidas que resolvían una y otra vez los mismos problemas. Para conseguir que los esfuerzos de los programadores puedan ser reutilizados se creó la posibilidad de utilizar módulos. El primer módulo existente fue la función, que somos capaces de escribir una vez e invocar cualquier número de veces.

Sin embargo, la función se centra mucho en aportar una funcionalidad dada, pero no tiene tanto interés

con los datos. Es cierto que la función puede recibir datos como parámetros y puede devolverlos, pero los trata de una estructura muy volátil, centrada en las operaciones. Simplemente hace su trabajo, procesando los parámetros recibidos y devuelve una respuesta.

En las aplicaciones en realidad los datos están muy ligados a la funcionalidad. Por ejemplo, podemos imaginar un punto que se mueve por la pantalla. El punto tiene unas coordenadas y podemos trasladarlo de una posición a otra, sumando o restando valores a sus coordenadas. Antes de la programación orientada a objetos ocurría que cada coordenada del punto tenía que guardarse en una variable diferente (dos variables para ser exacto: x, y) y las funciones de traslación estaban almacenadas por otra parte. Esta situación no facilitaba la organización del código ni tampoco su reutilización.

Con la Programación Orientada a Objetos se buscaba resolver estas situaciones, creando unas mejores condiciones para poder desarrollar aplicaciones cada vez más complejas, sin que el código se volviera un caos. Además, se pretendía dar una de pautas para realizar las cosas de manera que otras personas puedan utilizarlas y adelantar su trabajo, lo que deriva en mayores facilidades para la reutilización del código.

La POO no es difícil, pero es una manera especial de pensar, a veces subjetiva de quien la programa, de manera que la forma de hacer las cosas puede ser diferente según el programador.

Aunque podamos hacer los programas de formas distintas, no todas ellas son correctas, lo difícil no es programar orientado a objetos sino programar bien.

Pensar en términos de objetos es muy parecido a cómo lo haríamos en la vida real. Por ejemplo, vamos a pensar en un coche para tratar de modelizarlo en un esquema de POO.

Diríamos que el coche es el elemento principal que tiene una serie de características, como podrían ser el color, el modelo o la marca. Además, tiene una serie de funcionalidades asociadas, como pueden ser ponerse en marcha, parar o aparcar.

Por tanto, pensar en objetos requiere analizar qué elementos vas a manejar en tus programas, tratando de identificar sus características y funcionalidades. Una vez tengamos un ecosistema de objetos, éstos colaborarán entre sí para resolver los objetivos de las aplicaciones. (desarrolloweb, 2021)

En los lenguajes puramente orientados a objetos, tendremos únicamente clases y objetos. Las clases permitirán definir un número indeterminado de objetos, que colaboran entre ellos para resolver los problemas.

Con muchos objetos de diferentes clases conseguiremos realizar las acciones que se desean implementar en la funcionalidad de la aplicación.

Además, las propias aplicaciones como un todo, también serán definidas por medio de clases. Es decir, el taller de coches será una clase, de la que podremos crear el objeto taller de coches, que utilizará objetos coche, objetos de clase herramienta, objetos de clase mecánico, objetos de clase recambio, etc.

Las características de orientación a objetos fueron agregadas a muchos lenguajes existentes durante ese tiempo, incluyendo Ada, BASIC, Lisp más Pascal, entre otros. La adición de estas características a los lenguajes que no fueron diseñados inicialmente para ellas condujo a menudo a problemas de compatibilidad y en la capacidad de mantenimiento del código.

Los lenguajes orientados a objetos "puros", por su parte, carecían de las características de las cuales muchos programadores habían venido a depender. Para saltar este obstáculo, se hicieron muchas tentativas para crear nuevos lenguajes basados en métodos orientados a objetos, pero permitiendo algunas características imperativas de maneras "seguras".

c) Listas enlazadas

Una lista enlazada o estructura ligada, es una estructura lineal que almacena una colección de elementos generalmente llamados nodos, en donde cada nodo puede almacenar datos y ligas a otros nodos. De esta manera los nodos pueden localizarse en cualquier parte de la memoria, utilizando la referencia que lo relaciona con otro nodo dentro de la estructura.

Las listas enlazadas son estructuras dinámicas que se utilizan para almacenar datos que están cambiando constante mente. A diferencia de los vectores, las estructuras dinámicas se expanden y se contraen haciéndolas más flexibles a la hora de añadir o eliminar información.

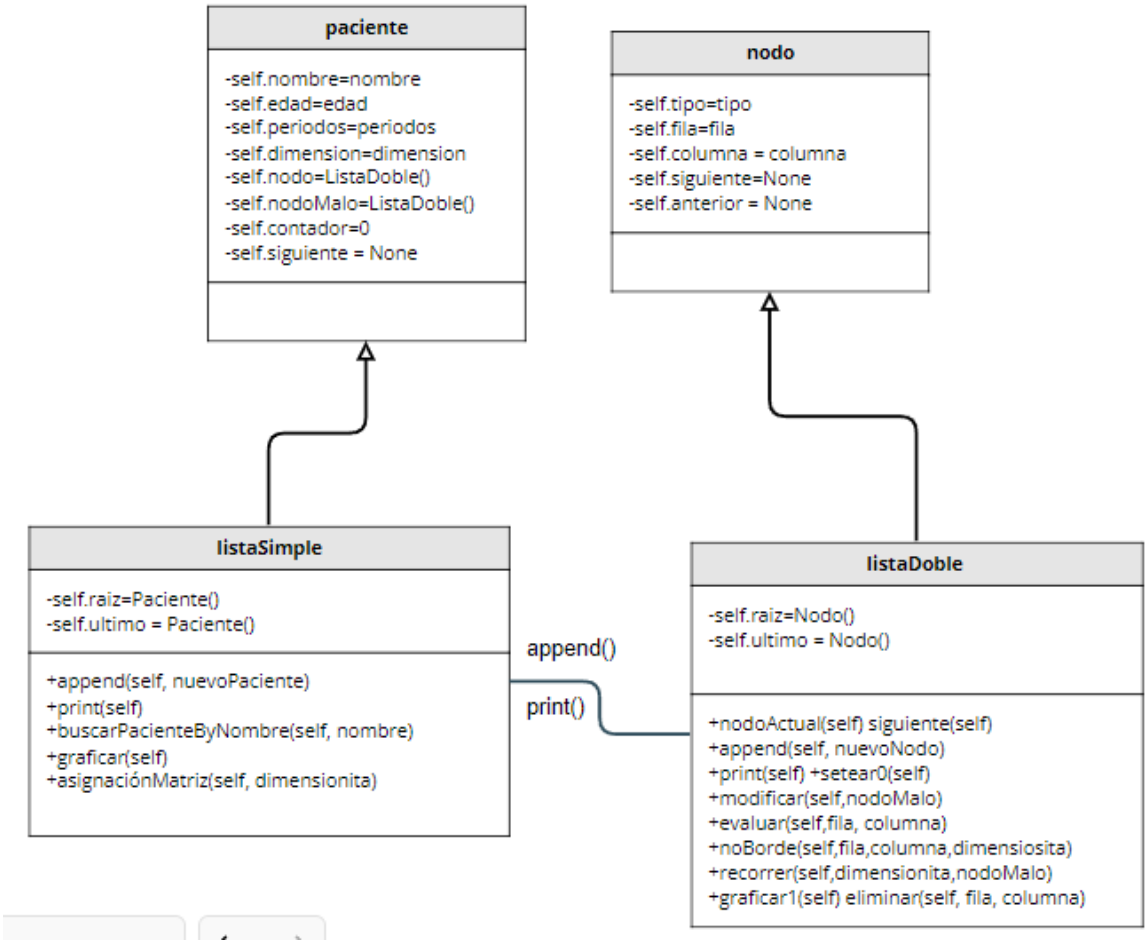
Las listas enlazadas permiten almacenar información en posiciones de memoria que no sean contiguas; y se almacena en los elementos nodos. Estos nodos poseen dos campos uno para almacenar la información o valor del elemento y otro para el enlace que determina la posición del siguiente elemento o nodo de la lista. (monografías, 2014)

Una lista doblemente enlazada es una lista lineal en la que cada nodo tiene dos enlaces, uno al nodo siguiente, y otro al anterior.

Las listas doblemente enlazadas no necesitan un nodo especial para acceder a ellas, pueden recorrerse en ambos sentidos a partir de cualquier nodo, esto es porque a partir de cualquier nodo, siempre es posible alcanzar cualquier nodo de la lista, hasta que se llega a uno de los extremos.

Los requisitos previos para programar una lista doblemente enlazada son el conocimiento de los tipos de datos, estructuras, el uso de typedef, los punteros, las funciones usuario y las listas enlazadas simples.

Diagrama de clases:



Conclusiones

Luego de realizar una abundante investigación referente a toda la estructura de nuestro proyecto, desde las bases iniciales (como la extensa explicación sobre el lenguaje en el que se trabajó, en este caso Python) pasando por la estructura inicial del proyecto (como la Programación Orientada a Objetos) hasta las aplicaciones específicas del mismo (como el uso de las listas enlazadas) se puede concluir abiertamente que la realización de este proyecto no puede ser posible sin tener un amplio manejo de la estructura anteriormente mencionada, ya que el conocimiento es utilizado como una torre, en la que sí los bloques de la base están tambaleantes, toda la torre se caerá, por lo que es de suma importancia conocer realmente bien todos los conceptos principales, por muy básicos que parezcan, para poder realizar de manera correcta este proyecto.

También se logra concluir a través del análisis de la realización del proyecto que un factor determinante, no sólo en el proyecto, sino que también en la programación en general es, no solo la paciencia, sino también la capacidad de analizar de manera clara y concisa las instrucciones que se están recibiendo, ya que desde el momento de la lectura del proyecto es importante poner en practica tus dotes de ingeniero para poder realizar una comprensión completa de los objetivos que requiere el proyecto que se está presentando, una vez estos requerimientos estén claros, el resto de la parte práctica fluirá con mucha más facilidad.

Referencias bibliográficas

Centro de Formación Técnica Para la Industria. (22 de 06 de 2020). *AULA21*. Obtenido de https://www.cursosaula21.com/que-es-python/

desarrolloweb. (18 de 09 de 2021). *Qué es la programacion orientada a objetos*. Obtenido de https://desarrolloweb.com/articulos/499.php

IBM. (17 de 08 de 2021). *Programación Orientada a Objetos*. Obtenido de https://www.ibm.com/docs/es/spss-modeler/saas?topic=language-object-oriented-programming

monografías. (02 de 08 de 2014). *Listas Enlazadas*. Obtenido de https://www.monografias.com/trabajos101/las-istas-enlazadas/las-istas-enlazadas2

Santander | Becas. (09 de 04 de 2021). *Python: qué es y por qué deberías aprender a utilizarlo*. Obtenido de https://www.becas-santander.com/es/blog/python-que-es.html