



PROYECTO FINAL

Grupos de Trabajo

Deberán formar grupos de 2 (dos) personas, los grupos deberán ser formados por los estudiantes e informarlos a la Cátedra vía mail.

Fecha de Entrega del Temario: 2 de diciembre de 2022 a las 17:00 hrs.

Fecha límite de Entrega: 11 de diciembre de 2022 a las 23:59 hrs.

Fecha de Presentación: Para la presentación del proyecto debe realizar un ppt en el cual explican cómo encararon la solución del problema. Las fechas de sustentación se pueden programar con la cátedra de lo contrario deben ser en las fechas de las finales programadas por la Universidad.

Sistema a Implementar

El Gobierno Argentino a través del proyecto Un País Digital busca agilizar e innovar la atención ciudadana, para lograr estos objetivos se han propuesto realizar las siguientes tareas:

- Implementar proyectos, productos y servicios digitales que promuevan la modernización de las gestiones provinciales, municipales y de la Ciudad Autónoma de Buenos Aires.
- Integrar los sistemas de gestión entre las distintas jurisdicciones, favoreciendo el intercambio y la transparencia de la información.
- Articular iniciativas en materia de gestión y servicios digitales con el sector privado, el campo académico, la sociedad civil y la comunidad

Como producto de este proyecto, en la actualidad se cuenta con puntos de acceso a internet de la red WIFI de País Digital la cual brinda acceso gratuito a internet en espacios y dependencias públicas de todo el país.

El sistema a implementar consistirá en un programa que analice la información de los puntos que brindan acceso gratuito a internet en espacios y dependencias públicas de todo el país y administre las conexiones y desconexiones de los usuarios a los mismos.

Los datos son reales y obtenidos de la página correspondiente del Gobierno Nacional.

Al enunciado se adjuntan dos archivos de ejemplo: el primero, “municipios.csv”, contiene la definición de un conjunto de municipios, en sus respectivos departamentos, a su vez dentro de sus respectivas provincias; el segundo, contiene información de los puntos de acceso disponibles en cada municipio.

Es necesario poder cargar archivos de municipios distintos, así como puntos de acceso nuevos en los mismos, a medida que vayan adhiriendo al programa. La información debe poder provenir de archivos csv en el formato de los provistos, o del teclado en cualquier momento. Puedo cargar varios archivos de municipios o puntos de acceso en cualquier momento de la ejecución.

A su vez, la información provista no es necesariamente confiable, debido a que es cargada por humanos, por lo que debe tenerse en cuenta esto a la hora de manejar la carga de datos. Frente a un nuevo ingreso inválido, debe avisarse al usuario (e incluir en el mensaje información que permita determinar qué dato fue el inválido) e ignorar esa información, el mismo será responsable de corregirla y reintentar la carga más adelante.

El programa deberá realizarse en Python, mostrando una clara separación de la interfaz del usuario respecto del procesamiento de los datos (back-end). Se espera que los alumnos desarrollen su sistema alrededor de una interfaz gráfica, con la que el usuario interactúe, en detrimento de la terminal, que no debe usarse ni para ingresar información ni para visualizarla.

Usted deberá usar para la implementación los conocimientos adquiridos durante el cuatrimestre y se valorará nuevos conocimientos a usar como valor agregado como parte de su proyecto. Se espera la correcta elección de estructuras de datos apropiadas para cada caso, no bastando solamente con que el programa funcione.

Recuerde que todos los integrantes del grupo deberán poder explicar lo implementado. Tener en cuenta especialmente a la hora de utilizar herramientas por fuera de lo incluido en el temario.

Requerimientos del proyecto:

Para implementar el programa a usted y su equipo deben:

1. Identificar los objetos que forman parte del sistema a implementar, crearlos y usarlos en la solución propuesta por su grupo.
2. De las estructuras vistas: Listas secuenciales, listas encadenadas simples, árboles binarios de búsqueda, pilas, colas, conjuntos, diccionarios y tuplas, Identificar las diferentes estructuras a implementar acorde a las funcionalidades que se les pedirán.
3. Manejo de la dataset del archivo descargado.
4. Hacer las validaciones correspondientes para evitar que el programa salga abruptamente.
5. Manejo de archivos

El sistema debe poder administrar distintos puntos de acceso (access point, o, coloquialmente, antenas WiFi) distribuidos en todo el territorio nacional. Los puntos de acceso deben agregarse o quitarse del sistema cuando comiencen o dejen de funcionar en el mundo real. Un punto de acceso puede instalarse o romperse en cualquier momento.

El sistema debe a su vez permitir a las computadoras de los usuarios conectarse o desconectarse de un punto de acceso. La computadora sólo debe poder estar conectada a un único punto de acceso a la vez.

Las computadoras se identifican con una dirección MAC única del equipo en el mundo, y al momento de conectarse, el punto de acceso debe asignarle una dirección IP única en ese router (es decir, puede haber direcciones repetidas en distintos routers). Para simplificar, puede pensar tanto las IPs como las MACs como números enteros, aunque este no sea el caso en el mundo real. Un punto de acceso puede tener un máximo de 20 conexiones simultáneas.

Cabe destacar que los ids de las locaciones secundarias (municipios y departamentos) solo son únicas dentro de su supraadministración. Es decir, pueden existir dos municipios con el mismo id, mientras estén en diferentes departamentos, de la misma forma que pueden existir dos departamentos distintos siempre y cuando estén en provincias diferentes.

El sistema debe poder imprimir el número de conexiones activas (no cerradas) en una provincia, municipio o departamento dado. Es decir, puedo pedir el número de conexiones activas (no cerradas) actualmente de la Provincia de Buenos Aires entera, o del municipio de 9 de Julio únicamente.

Finalmente, el sistema debe permitir visualizar todas las conexiones en la red nacional en un determinado intervalo de tiempo (por ejemplo, todas las conexiones que se iniciaron entre ayer a las 2 de la tarde y hoy a las 4:05 de la mañana). Esto vale para todas las conexiones, ya sea que sigan abiertas o que se hayan cerrado.

El sistema debe poder persistir su información entre cierres del mismo. Es decir, toda la información en el sistema, luego de cerrarse, debe seguir allí cuando vuelva a abrir el programa, sin necesidad de realizar ningún procedimiento manual como usuario. Asuma que la única forma de cerrar el sistema será mediante la opción correspondiente del menú.

Presentación y Entrega

En la fecha de entrega del trabajo práctico final, según cronograma (11 de Diciembre de 2022 a las 23:59 hrs), cada grupo deberá entregar a través del campus en un archivo zip con todos los archivos fuente (.py) funcionando.

En el archivo zip entregado debe estar el programa con las funcionalidades que fueron asignadas, las cuales deberán cumplir con las características de diseño e implementación descritas en este documento.

En la fecha acordada para la sustentación del trabajo práctico final, cada grupo tendrá que explicar la forma en que se realizó la solución del problema usando la presentación realizada. El ppt debe incluir la lógica y funcionalidades utilizadas para el desarrollo y perfecto funcionamiento del programa. Posteriormente debe poner funcionar el programa, y responder las preguntas de los docentes acerca de la implementación y del por qué de la elección de ciertas herramientas sobre otras. Todos los integrantes deben estar familiarizados con la totalidad del código, no pudiendo excusarse en la autoría

del mismo.

Criterios de evaluación y aprobación.

Debe hacer las validaciones correspondientes para evitar que el programa salga en forma abrupta

Si al seleccionar alguna opción del Menú, no se tienen resultados, la aplicación lo deberá informar con un mensaje al usuario y seguir funcionando correctamente.

Además, se valorará que cumpla con las características de eficiencia y estilo. Un programa es eficiente cuando usa los recursos en forma adecuada.

Un programa tiene estilo cuando está escrito de manera clara y simple. Esto incluye respetar las partes del programa, usar variables con nombres significativos, no repetir código innecesariamente, etc.

Legibilidad del código (separación en funciones / archivos, nomenclatura de variables y funciones, etc.).

Reutilización de código.

Armado de clases.

Tiempo de ejecución de las distintas funcionalidades.

Observaciones

- No utilizar tildes en el código ni en los datos de entrada en absoluto, para prevenir conflictos de formatos en el código fuente y archivos de datos al pasar la información de un sistema operativo al otro.
- Debe hacer las validaciones correspondientes para evitar que el programa salga en forma abrupta
- Respetar los formatos de entrada y salida especificados.
- Recomendamos testear su programa a conciencia: intentar romperlo con muchos inputs diferentes, tratando de lograr que funcione mal y arreglando los errores que vayan surgiendo, hasta convencerse de que ya no quedan más errores.