

UniGenius: Planificador Estudiantil 2024

Jimena Idarraga Rincón

Carlos Eduardo Trujillo Posada

Maria Fernanda Colorado Muñoz

Documento de visión:

Se quiere generar una base de datos solicitando un listado de 1000 estudiantes, donde se incluyen nombres y apellidos. Esto es para distribuir porcentualmente a los estudiantes en los 10 semestres de la carrera.

Para esto se requiere crear una planeación del programa para el próximo semestre académico, donde se considera una tabla de asignaturas; la cual incluye el nombre, el nivel, núcleo formativo y los créditos para cada asignatura.

A continuación, se presentan los pasos a cumplir para la realización de este trabajo:

Este software esta desarrollado en el lenguaje de programación Python, en el que primero creamos la lista para los mil estudiantes, con la librería panda, la cual brinda herramientas para leer y editar archivos CSV, posteriormente utilizamos las funciones de ciclos para distribuir los estudiantes por semestre según las indicaciones de proporcionalidad brindadas en el documento inicial, de este modo:

- Primer semestre 14% que serían 140 estudiantes.
- Segundo semestre 12% que serían 120 estudiantes.
- Tercer semestre 12% que serían 120 estudiantes.
- Cuarto semestre 11% que serían 110 estudiantes.
- Quinto semestre 10% que serían 100 estudiantes.
- Sexto semestre 10% que serían 100 estudiantes.
- Séptimo semestre 9% que serían 90 estudiantes.
- Octavo semestre 8% que serían 80 estudiantes.
- Noveno semestre 7% que serían 70 estudiantes.
- Décimo semestre 6% que serían 60 estudiantes.

A partir de tener esta distribución de estudiantes por semestres se calculará la cantidad de grupos que tiene cada curso, esto se obtuvo a partir de el cociente entre la cantidad de estudiantes por semestre y el cupo máximo de estudiantes por grupo en cada semestre. Dejando para el primer semestre 5 grupos por curso igual que para el segundo semestre, para el tercer semestre habría 4 grupos por curso, en el cuarto semestre habrá 5 grupos por curso, en el quinto y sexto semestre habrá 4 grupos, en el séptimo semestre habrá 5 grupos por curso, en el octavo habrá 4 grupos por curso, igual que en el noveno semestre y en el décimo semestre habrá 6 grupos por curso.

Para el almacenar los créditos de la materia se hará un diccionario donde la clave sea el nombre de la materia y el valor será la cantidad de créditos que esta tiene.

La información de los cursos se almacenará en un formato de lista la cual tendrá el siguiente orden:

1. Código Asignatura (C.A.): Este código se representa con las tres primeras letras del nombre de la asignatura, luego dos números que representan el semestre al que pertenecen y el último dígito representa la cantidad de créditos de la asignatura.
2. Horas de trabajo docente (HTD): se calcularán con un condicional según el crédito en la asignatura, si la asignatura tiene cuatro, las horas de trabajo docente serán 96, si son tres créditos, el HTD será de 64 y si la cantidad de créditos es menor o igual a dos se multiplica la cantidad de créditos por 16 horas de trabajo docente.
3. Las horas de trabajo independiente (HTI): se calcularán igual que las horas docentes, quedando así: si la asignatura tiene cuatro créditos, las horas de trabajo independiente serán 120, si son tres créditos, el HTI será de 80 y si la cantidad de créditos es menor o igual a dos se multiplica la cantidad de créditos que tiene la asignatura por 32 horas de trabajo independiente.
4. El número total de estudiantes: este lo podemos hallar sumando la cantidad de estudiantes que hay en cada grupo de la asignatura, que se podría utilizar la cantidad de estudiantes en el semestre puesto que el resultado sería igual en este ejercicio práctico, pero cuando vamos a la realidad, donde los estudiantes cancelan y pierden materias, es preferible que este dato se mantenga actualizando constantemente para permitir un mejor seguimiento a los estudiantes y sus necesidades.
5. El código del curso (CC): será el puesto respecto al total de cursos que se realizarían por semestre, irá del 1 al número de cursos dado; se podría calcular por una función cíclica que reparte el total de estudiantes hasta completar el total por semestre para la asignatura.
6. El total de cursos asignados (TCA): será el código de curso máximo, que se puede hallar por el cociente del total de estudiantes en el semestre y el campo máximo por aula, o utilizar el código del curso, tomando el valor máximo que este tome.

7. Fecha de creación del documento: utilizaremos la librería datetime con su función date.today () para que guarde la fecha de la edición de el documento si se requiere alguna modificación la cual se presentara en un formato de año, mes, día.

Con los datos anteriores podemos crear los planos CSV mediante la librería CSV con su función csv.writer la cual permite escribir documentos CSV, la información de cada curso se utilizarán las listas anidadas para almacenar la información antes de exportar los archivos.

Especificación de requisitos:

Requisitos funcionales:

Para el buen funcionamiento del software, se ha visto que los elementos del siguiente listado son necesarios.

- Creación de un listado de 1000 estudiantes con datos personales únicos, distribuidos por semestre según la tabla proporcionada.
- Utilización de los archivos "nombres.csv" y "apellidos.csv" para generar nombres de estudiantes aleatorios.
- Garantizar que no existan estudiantes con el mismo nombre y apellido en el listado generado.
- Creación de una tabla de asignaturas para cada semestre, incluyendo el código de asignatura, horas de trabajo docente, horas de trabajo independiente, número total de estudiantes, etc.
- Distribución equitativa de los estudiantes en las asignaturas, respetando el cupo máximo por aula especificado en la tabla.
- Generación de listas de estudiantes para cada curso, asegurando que no haya repeticiones dentro de un mismo grupo.

Requisitos no funcionales:

Un requisito no funcional, que puede mejorar la adaptación del software es utilizar los documentos de identidad de los estudiantes como elementos guía puesto que en la práctica los nombres y apellidos se pueden repetir; y los documentos de identidad son únicos para cada estudiante.

- Rendimiento: El programa debe funcionar rápido y sin hacer esperar mucho tiempo mientras se genera la lista de estudiantes y asignaturas.
- Seguridad: Los nombres y detalles que el programa usa para crear los estudiantes deben ser falsos y nadie más que el autor debería poder verlos o acceder a ellos.
- Usabilidad: El programa debe ser fácil de entender y usar. No se puede estar luchando con él, solo se necesita poder hacer lo pedido sin complicaciones.

- Fiabilidad: el programa no puede presentar errores o que posiblemente se caiga mientras está creando las listas de estudiantes y asignaturas. Confiando en él para que funcione correctamente.
- Compatibilidad: Usar el programa sin importar qué tipo de computadora tenga. Debería funcionar en cualquier sistema operativo y con las últimas versiones de Python sin problemas.

Plan de Proyecto:

Este proyecto al ser académico no tiene un presupuesto fuerte y la inversión que se le da es en tiempo de trabajo, tiempo el cual se repartió equitativamente.

1. Metas:

- Crear una herramienta que facilite la organización de asignaturas y estudiantes para el próximo semestre universitario.
- Hay que asegurar que los estudiantes se distribuyan equitativamente en las asignaturas, siguiendo las pautas proporcionadas por el profesor.
- Cumpliendo con todos los requisitos establecidos para el proyecto, tanto en funcionalidad como en calidad.

2. Tareas:

Fase de Preparación:

- Revisión detallada de los requisitos del proyecto.
- Definición de objetivos específicos y medibles.
- Asignación de roles y responsabilidades entre los miembros del equipo.
- Elaboración del plan de proyecto, incluyendo actividades, cronograma y presupuesto.

Fase de Desarrollo:

- Implementación del código para crear listado de mil estudiantes.
- Desarrollar el código para distribuir los estudiantes por semestre y grupo de asignatura.
- Aplicación de código para calcular las horas de trabajo docente y las horas de trabajo independiente de cada curso.
- Implementación del código para generar el código de cada curso con la cantidad de estudiantes y el total de cursos asignados y la fecha de creación del documento

Fase de Pruebas y Ajustes:

- Ejecución de pruebas integrales para identificar posibles errores o fallos.
- Realización de ajustes y correcciones basados en los resultados de las pruebas.
- Optimización del rendimiento y la usabilidad del software para una mejor experiencia del usuario.

Fase de Entrega:

- Preparación de la documentación final del proyecto, incluyendo informes de progreso y resultados.
- Entrega del proyecto completo al profesor dentro del plazo establecido.

3. Cronograma:

- Preparación del proyecto.
- Desarrollo del software.
- Pruebas y ajustes.
- Entrega del proyecto.

4. Presupuesto:

- Cada estudiante participante dedicará 4 horas de tiempo de práctica de formación al proyecto, considerando un equipo de tres estudiantes, el presupuesto total será de 12 horas de práctica de formación.

5. Notas Adicionales:

- Se utilizará una herramienta de gestión de proyectos como Microsoft 365 para mantener un seguimiento constante del avance del proyecto.
- Se realizaron reuniones regulares para evaluar el progreso y realizar ajustes según sea necesario.
- Se mantendrá una comunicación abierta y constante entre los miembros del equipo para abordar cualquier problema o duda que pueda surgir durante el desarrollo del proyecto.

Plan de versionado:

El plan versionado hasta la fecha límite de esta primera entrega solo tiene el análisis del problema y un resumen del desarrollo de posibles soluciones, lo que cumple lo indicado en la fase de preparación del proyecto. No obstante, cabe aclarar que no se ha iniciado el desarrollo de software, esta parte del proyecto se irá actualizando con el cumplimiento de las etapas del proyecto.