**Trabajo Final – Algoritmos y Programación**

Miguel Monsalve, Mariana Pineda, Laura Rincón Aldana

Facultad de Ingeniería, Universidad de Antioquia

2553935: Algoritmia y Programación

Prof. Victor Mercado

4 de junio de 2024

**Documento de Visión**

**Nombre del Proyecto**

Planificador Académico Ingeniería Industrial

**Descripción General**

El objetivo del proyecto es desarrollar una planeación académica integral para el programa de Ingeniería Industrial de la Universidad de Antioquia. Este proyecto se centrará en la distribución porcentual de los estudiantes en diversos semestres y asignaturas. Para lograrlo, se creará un software utilizando el lenguaje Python, empleando datos proporcionados por el docente.

**Beneficios del Software**

**Optimización del proceso de planificación académica:**

Mejora la eficiencia y precisión en la organización del programa de estudios.

**Distribución equitativa de estudiantes:**

Asegura una asignación balanceada de los estudiantes en los diferentes semestres y materias, evitando sobrecargas y subutilización de recursos.

**Automatización del proceso:**

Reduce la carga administrativa y el riesgo de errores humanos, permitiendo un manejo más ágil y confiable de la planificación académica.

**Requisitos Funcionales:**

***Carga de archivos:***

En la primera entrega se había descrito el importe de las librerías “pandas” y “random”. Sin embargo, para la entrega final se necesitaron otras librerías tales como: “datetime”, “os”, “logging” y “time” con el fin de manipular los datos, generar secuencias aleatorias, interactuar con el sistema operativo, generar mensajes de log y usar funciones relacionadas con el tiempo. En esta entrega se cargaron los nombres y apellidos desde los archivos CSV en GitHub, también las funciones mencionadas en la primera entrega como read\_csv para leer los datos y posteriormente convertirlos a listas.

***Generar nombres y apellidos aleatorios:***

En la primera entrega se había descrito el uso de un ciclo “while” y de la función random.choice para la generación de los elementos aleatorios hasta la cantidad de 1000 elementos. De esta manera, la función selecciona dos nombres y dos apellidos de manera aleatoria mientras la contadora se incrementa en 1. Sin embargo, no habíamos mencionado el hecho de utilizar un conjunto, que nos permitiría tener elementos únicos. La cual sería la corrección para esta entrega. En la anterior habíamos mencionado un condicional para esto, pero es mejor la funcionalidad del conjunto.

***Asignación de semestres proporcionalmente:***

Desde la primera entrega se había descrito la necesidad de una lista vacía y un ciclo “for” para iterar sobre cada semestre y con la tabla porcentajes determinar cuántos estudiantes se asignan por semestre. Para esta entrega se define que la lista se llama “semestres” y que el ciclo “for” extiende la lista con el número de estudiantes correspondientes según las proporciones. Para esto, cada proporción se multiplica por 10 para obtener el número entero de estudiantes por semestre.

***Generación de datos ficticios:***

Para esta entrega se anexa una lista vacía “data” que almacena los datos de los estudiantes. Con un ciclo “for” se recorre el conjunto de estudiantes y la lista de semestres, para cada estudiante se crea un diccionario con información personal: nombre, semestre, cédula y correo electrónico. Para la creación del último se hizo una combinación de los dos primeros nombres y el dominio de la Universidad de Antioquia. Se agrega también un diccionario “datos\_semestre” en el que la clase es el número del semestre y el valor con el porcentaje y el cupo máximo de estudiantes por aula.

***Generación de Códigos Asignaturas:***

Para la creación del código de la asignatura se crea una función “generar\_codigo\_asignatura” que tiene los parámetros nombre, semestre, créditos y consecutivo. Que genera el nombre con las primeras iniciales de cada palabra de la asignatura, el semestre, los créditos y un consecutivo. Tal como se había planeado desde la primera entrega.

***Creación de la Estructura de las Carpetas:***

Para esta entrega se crea una lista vacía “planificación”. Luego se define la ruta base\_dir donde se guardarán los archivos, se crearán utilizando “os.makedirs”. Con un ciclo “for” se itera sobre cada asignatura y para cada una se obtiene el nombre, el semestre y los créditos.

***Filtrado de Estudiantes por Semestre:***

Para esta entrega se crea una lista de “estudiantes\_semestre” que contiene los estudiantes cuyo semestre coincide con el semestre actual. El límite de estudiantes se obtiene del diccionario anteriormente creado “datos\_semestres”. Se calcula el número de grupos necesarios dividiendo el número total de estudiantes entre el límite por aula. Se itera con el ciclo “for” sobre el número de grupos necesarios y para cada asignatura se genera un código de asignatura utilizando la función anteriormente descrita “generar\_codigo\_asignatura”. Se definen las horas de trabajo docente y las horas de trabajo independiente con condicionales anidados tal como se había postulado en la primera entrega. Posteriormente, se crea un diccionario con la información del grupo y se añade a la lista “planificacion”.

La información contiene: código de la materia, las horas de trabajo docente, las horas de trabajo independiente, el numero total de estudiantes por grupo y en el semestre, el código del curso, el numero total de cursos asignados y la fecha de creación. Esta ultima se había mencionado en la primera entrega de utilizar la función datetime.now(), de la cual prescindimos en esta entrega.

***Asignación de Estudiantes:***

En la anterior entrega se había descrito el uso de un ciclo y de la función random.randint() para seleccionar un número aleatorio entre 0 y la cantidad de estudiantes disponibles. Y por último Asignar el nombre completo del estudiante seleccionando su semestre actual. Sin embargo, para esta entrega se seleccionan los estudiantes correspondientes al grupo actual utilizando el índice del bucle.

***Crear carpetas:***

En la anterior entrega no se habían descrito a detalle las carpetas a utilizar para ordenar la información de forma adecuada. Se agrega por tanto una ruta “semestre\_dir” para el semestre actual. La cual se crea con “os.makedirs”. De forma análoga se crea la carpeta “asignatura\_dir”.

***Ajustes Finales y Guardado de los Archivos:***

En la anterior entrega no se habían concretado los detalles del guardado y del formato. Para esta se crea un DataFrame de la lista “estudiantes\_grupo”. Se genera el nombre del archivo de acuerdo a las especificaciones del trabajo: la cadena “nombre\_archivo” con el código de la asignatura, el nombre de la misma sin espacios, el número de estudiantes en el grupo y el numero del grupo. Se guardan los archivos con to\_csv y con to\_excel. En la entrega pasada se había mencionado el guardado con la función open(), csv.writer() y close(). Para esta se prescinde de las mismas. Finalmente se crea un DataFrame a partir de la lista “planificación” y se guarda en csv, con la misma función mencionada.

**Requisitos No Funcionales:**

***Eficiencia y rapidez***

El software debe ser capaz de generar y gestionar datos de manera rápida y eficiente, optimizando el tiempo de procesamiento.

***Usabilidad***

La interfaz debe ser intuitiva y fácil de entender, permitiendo a los usuarios operar el sistema sin dificultades.

***Extensibilidad***

El sistema debe ser fácilmente adaptable y extensible a otras áreas o programas académicos.

***Seguridad***

Debe garantizar la seguridad y protección de la información personal, implementando medidas robustas para evitar accesos no autorizados y asegurar la confidencialidad de los datos.

**Plan de Proyecto**

**Actividades:**

**Recolección de requisitos:** 3 días

**Diseño del software:** 5 días

**Implementación de funcionalidades:** 10 días

**Pruebas y depuración:** 7 días

**Documentación del software:** 3 días

**Entrega del proyecto:** 1 día

**Cronograma**

|  |  |  |
| --- | --- | --- |
| **Actividades** | **Fecha Inicio** | **Finalización** |
| Recolección de requisitos | 28/03/2024 | 01/04/2024 |
| Diseño del software | 20/04/2024 | 25/04/2024 |
| Implementación de funcionalidades | 01/05/2024 | 10/05/2024 |
| Pruebas y depuración | 23/05/2024 | 30/05/2024 |
| Documentación del software | 01/06/2024 | 03/06/2024 |
| Entrega del proyecto | 03/06/2024 | 04/06/2024 |

**Presupuesto**

**Total de horas de práctica de formación:** 50 horas

**Presupuesto total:** 5 SMLV

**Plan de Versionado**

**Versión 1.0:**

Desarrollo inicial del software con carga de archivos y generación de nombres aleatorios.

**Fecha de inicio:** 20/04/2024

**Fecha de finalización:** 28/04/2024

**Versión 2.0:**

Implementación de asignación de semestres proporcionalmente y generación de datos ficticios.

**Fecha de inicio:** 29/04/2024

**Fecha de finalización:** 04/05/2024

**Versión 3.0:**

Creación de la estructura de carpetas y asignación de estudiantes.

**Fecha de inicio:** 05/05/2024

**Fecha de finalización:** 10/05/2024

**Versión 4.0:**

Pruebas, depuración y documentación del software.

**Fecha de inicio:** 23/05/2024

**Fecha de finalización:** 03/06/2024

**Entrega Final:**

Entrega del proyecto completo.

**Fecha de entrega:** 04/06/2024