

**PROYECTO DE ALGORITMIA Y PROGRAMACIÓN**

**ENTREGA 1**

**PRESENTADO A:**

### **CAMILO LLERENA RIASCOS**

**(Docente)**

**INTEGRANTES:**

**SARA MURIEL**

**UNIVERSIDAD DE ANTIOQUIA**

**DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**

**ALGORITMIA Y PROGRAMACIÓN**

**14/09/2024**

**TABLA DE CONTENIDO**

Descripción general del software............................................................................................. 3

Objetivo general......................................................................................................................3

Objetivos especificos............................................................................................................. 3

Beneficios.............................................................................................................................. 4

Requisitos del software.......................................................................................................... 5

Actividades y Cronograma tentativo...................................................................................... 6

Presupuesto ........................................................................................................ 6

Plan versionado.................................................................................................. 7

Flujograma...................................................... 8

Pseudocódigo........................................................................................ 9

**Descripción general del Programa**

Este proyecto se enfoca en desarrollar un sistema automatizado en Python llamado “EduGestor: Sistema de Gestión Académica” para la planificación académica semestral del programa de Ingeniería Industrial de la Universidad de Antioquia. El software gestiona la asignación de 1000 estudiantes a los cursos correspondientes, respetando la capacidad de las aulas y la secuencia de la malla curricular. Genera listados de estudiantes por curso en formatos Excel y CSV, organizados por semestre y asignatura, almacenados en una estructura de carpetas clara. Además, simula el rendimiento académico, asegurando que al menos el 70% de los estudiantes aprueben, optimizando así la gestión académica y administrativa. Adicionalmente genera un archivo en formato Excel donde cada registro representará los grupos de asignaturas y cada columna será información sobre las asignaturas (código de la asignatura, semestre, nombre, código del curso, cantidad de estudiantes, nota promedio)

**Objetivo General**

Desarrollar un sistema automatizado en Python que optimice la planificación académica semestral del programa de Ingeniería Industrial de la Universidad de Antioquia, gestionando de manera eficiente la asignación de estudiantes a las diferentes asignaturas, y generando listados organizados para su uso administrativo.

**Objetivos Específicos**

* Diseñar un algoritmo que asigne a los 1000 estudiantes de manera equitativa a las distintas asignaturas y grupos, maximizando la capacidad de las aulas y asegurando una distribución balanceada.
* Implementar un proceso que automatice la creación de listados de estudiantes por curso en formatos Excel y CSV, organizados por semestre y asignatura, el código del curso, nombre de la asignatura, cantidad de estudiantes
* Incorporar un módulo que permita simular y analizar el rendimiento académico de los estudiantes, generando notas de manera aleatoria para evaluar el progreso y asegurar la calidad educativa.
* Establecer una estructura clara y eficiente para el almacenamiento de los datos académicos, asegurando un fácil acceso y organización de la información para su posterior análisis.
* Asegurar que el software desarrollado sea compatible e integrable con otros sistemas administrativos y académicos de la universidad, facilitando una gestión unificada y coherente de los procesos educativos.

**Beneficios**

* Minimizar significativamente el tiempo dedicado a la planificación semestral, optimizando así el flujo de trabajo del personal académico.
* Promueve un ambiente educativo más innovador al liberar a los docentes y administradores de tareas repetitivas, permitiéndoles concentrarse en el desarrollo de nuevas metodologías y enfoques educativos que beneficien a los estudiantes.
* Distribución justa de estudiantes en los cursos, previniendo la saturación de aulas y asegurando una experiencia educativa equilibrada.
* Mejora el flujo de información entre diferentes departamentos de la universidad, lo que permite una gestión más coherente de los datos académicos y administrativos, optimizando la colaboración interdepartamental.
* Facilita la implementación de ajustes rápidos ante cambios en la cantidad de estudiantes o modificaciones en el currículo, asegurando que el sistema se mantenga relevante y efectivo en distintas circunstancias académicas.
* Permite el seguimiento continuo del desempeño de los estudiantes, ayudando a identificar aquellos que necesitan apoyo adicional y asegurando que se tomen medidas a tiempo para mejorar su éxito académico.

**Especificación de requisitos del software de planificación académica:**

**Requisitos funcionales**

• **Generación automática de listas de clases**: Cuando comienza el proceso de planificación de un nuevo semestre, el sistema debe generar automáticamente listas de clases para todas las materias de la carrera de Ingeniería Industrial, asignando a los estudiantes a cursos específicos según el porcentaje proporcional establecido.

**• Asignación equitativa de estudiantes a los cursos**: Si una materia tiene un límite de 30 estudiantes por aula y hay 90 estudiantes matriculados, el sistema debe distribuir a los estudiantes en tres grupos de manera equitativa, evitando sobrecargar un grupo sobre los demás, y si hay Si es un grupo con menos estudiantes de lo esperado, el sistema permitirá realizar ajustes manuales a la tarea.

**• Registro de eventos y auditoría de procesos:** El sistema debe registrar cada acción realizada por los usuarios, como crear listas de clases, que serán accesibles para los desarrolladores y administradores, modificar las tareas de los estudiantes y generar informes de auditoría para revisarlos más adelante.

* **Tipo de usuario**: El sistema debe de tener tipos de usuario para cada requerimiento, que los administradores sean quienes tienen acceso a los listados csv, los cuales contienen los datos de todos los estudiantes y capacidad de modificar los cursos y las notas de dichos estudiantes, y la interfaz de usuario donde el estudiante puede ver sus notas y a que grupo fue asignado.
* **Manejo de casos especiales:** El sistema debe tener como manejar casos especiales, como por ejemplo los estudiantes inscritos tarde o que el sistema los haya inscrito en un grupo que no los debía inscribir y los administradores y desarrolladores puedan tener acceso a ello.
* **Validación de datos:** El sistema deberá de reconocer cuando se equivoca al introducir información errada como tipo de dato o extensión de un número de registro.

**Requisitos no funcionales:**

* **Interfaz de usuario**: El sistema debe tener una interfaz gráfica amigable con los usuarios, una interfaz que sea intuitiva y fácil de utilizar.

**• Rendimiento**: El sistema debe ser capaz de generar listas de clases para un número variable de estudiantes y materias, adaptándose a las necesidades cambiantes de la carrera de Ingeniería Industrial sin comprometer el rendimiento.

• **Seguridad y privacidad de los datos**: El sistema debe implementar medidas de seguridad como autenticación de usuarios, cifrado de datos y control de acceso para proteger la información confidencial de los estudiantes.

**Plan de proyecto**

**Actividades Y Cronograma tentativo**

|  |  |  |  |
| --- | --- | --- | --- |
| Actividad | Semana | Hora | Notas |
| Análisis de los datos iniciales | 1 | 2 | Revisión y limpieza del archivo “datos.csv” para comprender la estructura de los datos y las variables necesarias. |
| Diseño de la estructura de carpetas y archivos | 1 | 2 | Crear un esquema de carpetas para organizar los listados de clase por semestre. |
| Modelado de la estructura del programa en Python | 2 | 4 | Definir la estructura de datos, como los estudiantes por semestre y asignatura. |
| Desarrollo de la función de asignación de grupos | 3 | 2 | Desarrollar un algoritmo para asignar a los 1000 estudiantes en grupos de acuerdo con los límites de aula. |
| Cálculo de los créditos por asignatura | 4 | 2 | Implementar la función que calcule los créditos por asignatura en función de la malla curricular. |
| Generación del listado de clases por semestre | 5 | 3 | Organizar los estudiantes en los grupos correctos según el semestre y la asignatura. |
| Validación de la asignación de grupos | 6 | 3 | Verificar que la distribución de estudiantes en grupos sea correcta y respeten los límites de aula. |
| Pruebas del programa | 7 | 3 | Realizar pruebas para verificar el correcto funcionamiento de la asignación de estudiantes a grupos. |
| Documentación del proyecto | 8 | 4 | Elaborar la documentación técnica del código y de los procesos. |

**PRESUPUESTO**

|  |  |  |
| --- | --- | --- |
| Concepto | Descripción | Costo estimado (cop) |
| Software y licencias | Software usado: visual studio code. | $0 (visual studio code) |
| Equipo de trabajo | 1 estudiante, se estipula el salario según el costo de la hora en el salario mínimo colombiano, en total se llevó 25 horas de trabajo. | $135.333 |
| Materiales de trabajo | Computadora (propia) | $2.500.000 |
| Total estimado | | $2.635.000 |

**PLAN VERSIONADO**

***Versión 1.0.0 (12 de septiembre de 2024)***

* Creación de la estructura base del programa en Python.
* Almacenamiento de los datos de los 1000 estudiantes.
* Desarrollo y ejecución del programa de asignación de estudiantes.
* Asignar a los estudiantes a sus grupos de clase.
* Generación automática de listado de estudiantes por curso y semestre.
* Validaciones iniciales para garantizar la correcta asignación de estudiantes y el manejo de errores comunes.

#### **Versión 1.1.0 (18 de septiembre de 2024)**

* Asignación de estudiantes a grupos de clase según el tamaño permitido por curso.
* Organización de los datos asegurando que los archivos se guarden correctamente en carpetas por semestre y curso.
* Optimización de la generación de archivos **CSV** y **Excel** para garantizar un formato estandarizado y mejorar la compatibilidad con otros sistemas administrativos.
* Implementación de validaciones más robustas de los datos de entrada, asegurando que los archivos **CSV** no contengan errores en el formato de las notas o en la distribución de estudiantes.
* Simulación del rendimiento académico con una tasa mínima de aprobación del 70%.
* Documentación descriptiva inicial, explicando el funcionamiento del sistema y brindando una explicación básica del código utilizado.

#### **Versión 1.2.0 (23 de septiembre de 2024)**

* Renovación de los datos conforme cambia el semestre

**Flujograma**

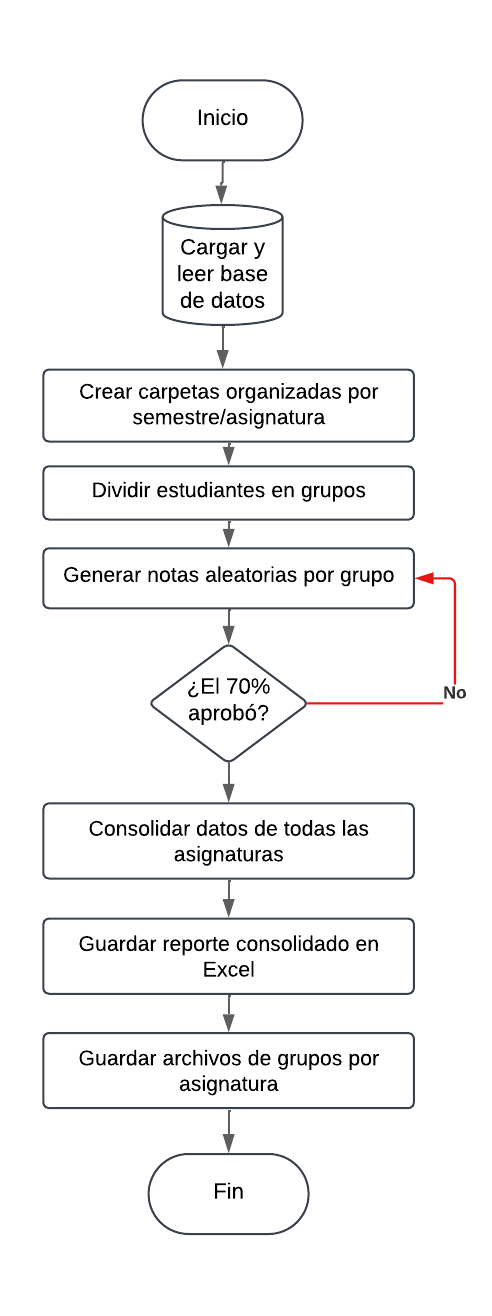


Ilustración flujodrama del programa.

**PSEUDOCÓDIGO**

**Inicio**

**Cargar y leer base de datos:**

- Leer un archivo CSV que contiene los nombres de los estudiantes.

- Generar IDs únicos para cada estudiante.

**Crear carpetas organizadas por semestre/asignatura:**

- Por cada asignatura en la lista de asignaturas:

- Crear una carpeta con el nombre del semestre.

- Dentro del semestre, crear una subcarpeta con el nombre de la asignatura.

**Dividir estudiantes en grupos**

- Para cada asignatura:

- Dividir a los estudiantes en grupos con un máximo permitido de estudiantes por grupo.

**Generar notas aleatorias por grupo**

- Por cada estudiante en cada grupo:

- Generar una nota aleatoria entre 2.0 y 5.0.

- Garantizar que el 70% de las notas sean ≥ 3.0 (aprobadas).

**¿El 70% aprobó?**

- Revisar si el porcentaje de aprobados cumple con el 70%.

- Si no: Volver a generar las notas (repetir este paso).

- Si sí: Continuar.

**Consolidar datos de todas las asignaturas**

- Combinar la información de estudiantes, grupos y notas por asignatura.

- Crear un reporte con columnas como:

- Código de asignatura, nombre, semestre, número total de estudiantes, nota promedio, etc.

**Guardar reporte consolidado en Excel**

- Crear un archivo Excel en la carpeta principal con todos los datos consolidados.

**Guardar archivos de grupos por asignatura**

- Por cada asignatura y grupo:

- Guardar un archivo Excel y CSV con los datos del grupo (ID, nombre, nota, grupo, etc.). **Fin**