**Entrega 1:**

Cristhian Alexis Mendoza Santander

Ana Sofía Murillo Hernández

Nataly Martínez González

**Algoritmia y Programación:**

Julián Andrés Castillo Grisales



**UNIVERSIDAD DE ANTIOQUIA**

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**

**MEDELLÍN -ANTIOQUIA**

**2024\_1**

**NAS PROGRAMMING**

* **DOCUMENTO DE VISIÓN**

El objetivo del proyecto NAS PROGRAMMING consistirá en realizar de forma automatizada la planeación de las matrículas de los estudiantes para el programa de Ingeniería Industrial del próximo semestre para una cantidad de mil (1000) estudiantes.

Este software asignará a cada estudiante el conjunto de materias que puede cursar según su semestre, teniendo en cuenta el límite del cupo del aula que se le asigna a cada materia por semestre y garantizando un cupo para cada estudiante.

Los objetivos principales del programa son:

* + Identificar correctamente la información completa de cada curso (número de créditos, código, HTD, HTI, número de estudiantes, total de cursos asignados y fecha de creación) así como la de los estudiantes (semestre, nombre, ID, etc.)
  + Leer y seleccionar correctamente la información de las listas brindadas, de manera que permita distribuir proporcionalmente a los estudiantes de diferentes semestres en el conjunto de asignaturas que se les permite cursar.

En un principio se busca que funcione de forma correcta pero limitada, esperando que con el tiempo y una modelación mas avanzada se logre una versión optimizada que permita el manejo de mayor información (No limitada).

Los beneficios del programa serían:

* La automatización del proceso de selección y distribución, lo que se traduce en un proceso más eficiente y que ahorra tiempo y esfuerzo manual.
* Mayor precisión en la selección, lo que disminuye la posibilidad de cometer errores en las matrículas.
* El manejo efectivo de la información, pues el programa se encarga de organizar la información para presentarla de forma estructurada.
* El programa puede ser modificado para adaptarse (ahora o en un futuro) según las necesidades que presente la institución.
* El programa es seguro y confiable, en pocas palabras es una solución que permite organizar y entregar resultados de forma optimizada.
* **ESPECIFICACIONES DE REQUISITOS**
* **Requisitos funcionales:**

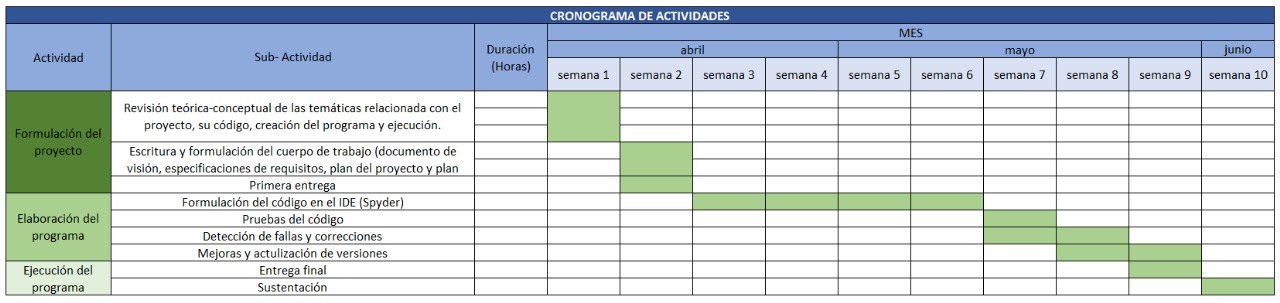
Para elaborar el programa se necesita un equipo con un procesador mínimo Core i3 o RYZEN 3, con una memoria mínimo de 8GB y un disco duro de 256GB; esto con el fin de tener la capacidad de soportar el entorno de ejecución de manera óptima; instalar el software de programación (Python) o en su defecto un entorno de desarrollo integrado (IDE) de mayor preferencia que permita la programación en el lenguaje de Python, internet y la persona responsable de realizar el programa. El programa debe cumplir con los siguientes requisitos:

* + El programa deberá ejecutar de manera correcta la selección de los archivos, debe reconocerlos y clasificarlos, deberá asignar carpetas por semestre y dentro de estas carpetas, las asignaturas que conforman el semestre. En la carpeta de cada asignatura deberá crear otras carpetas que serán los grupos de cada clase.
  + El programa debe asignar a los estudiantes que harán parte de estos grupos, teniendo en cuenta el límite en cada grupo.
  + Las carpetas de las asignaturas contendrán archivos de Excel en dos formatos (xlsx, CSV). El programa debe administrar los nombres de los estudiantes de tal manera que no se repitan por asignatura y grupo.
  + Una vez el programa realiza todos los procedimientos, estos se deberán guardar. Para ello se utilizará el módulo “logging” el cual, al momento de ejecutarse el programa, rastreará los procedimientos y luego los guardará. Para este guardado el “log” debe tener fecha, hora, minuto, segundo y milisegundo en un separador tabular, la instrucción a realizar y el tiempo que demora realizar la operación.
  + El programa debe tomar un dato de entrada y entregar una información coherente de salida.
* **Requisitos no funcionales:**
* El software debe ser capaz de manejar grandes niveles de información de manera que tanto la lectura como el resultado que se pide sea entregado de manera rápida y concisa.
* La información debe ser recolectada de la base de datos y asignada de manera precisa, pues se espera que cada estudiante sea asignado a sus cursos correspondientes, es decir, que se desea evitar los errores de clasificación lo más posible.
* El Software es altamente adaptable, es decir que a medida que se hagan actualizaciones, será capaz de manejar un nivel de información más grande que en su primera versión, así como entregar la información de manera más rápida.
* Es resistente a fallos y violaciones de seguridad. Es decir, que protege la información sensible tanto de los estudiantes como de la institución. Así mismo, es un programa de uso fácil y poco propenso a sufrir fallos (Como el colapso del software) en cualquier momento.
* Se espera que sea un software al que se pueda acceder en gran variedad de dispositivos sin importar su sistema operativo, sin que esto afecte la facilidad de uso.
* **PLAN DEL PROYECTO:**
* **Actividades:**
* 1) Revisión teórica-conceptual de las temáticas relacionada con el proyecto, su código, creación del programa y ejecución.
* 2) Elaboración del programa mediante código en el IDE escogido para cumplir con el objetivo del proyecto (Planeación del programa para el siguiente semestre).
* 3) Realizar pruebas para descartar o corregir errores o defectos en el código.
* 4) Describir la versión del software y planificar los avances del mismo y su cambio de versión.
* 5) Ejecución del programa.
* **Presupuesto del proyecto:**
  + Estudiantes en el proyecto: 3 estudiantes
  + Horas invertidas: 6 horas c/e por semana
  + Semanas investidas: 10 semanas c/e
  + Total de horas: (6\*3) \*10 = 180 horas

Teniendo en cuenta el salario mínimo para el 2024 ($1.300.000) y el valor hora ($6915), el costo del proyecto por semana es de $124.470 y su costo total es de $1.244.700.

c/e = Cada estudiante

* **Cronograma:**



* Semana 1 a 2: Se realizará la revisión bibliográfica para contextualizar a los estudiantes con el tema en relación a el proyecto a elaborar, para realizar la primera entrega del proyecto.
* Semana 2 a 5: Se iniciará con la programación del código hasta tener una versión inicial.
* Semana 5 a 6: En base a la primera versión, se realizarán las primeras pruebas, con el objetivo de detectar fallas.
* Semana 6 a 9: Se procederá a realizar las correcciones pertinentes, así como a realizar mejoras en el programa.
* Semana 10: Se hará la entrega final y la sustentación del programa.
* **PLAN VERSIONADO:**
* Versión 0.0.0 es la versión inicial del programa, esta versión se irá actualizando a medida que se realicen cambios, correcciones de errores y mejoras al programa. Los cambios en el versionado se realizarán de la siguiente forma:
  + *n.0.0*: El valor n se actualizará conforme se hagan grandes cambios en el diseño del programa y cuando este sea capaz de manejar una mayor cantidad de información. En este caso, la siguiente versión del programa sería la 2.0.0.
  + *1*.*n.0:* El valor n se actualizará conforme se vayan corrigiendo errores que presenta el programa en su fase de elaboración. Serán las que más se evidencien a la hora de entregar el primer programa.
  + *1.0.n:* El valor n se actualizará conforme se hagan cambios muy pequeños de corrección.
* **PLAN VERSIONADO (Actualizado para la entrega 2):**
* Versión 1.0.0 es la versión inicial del programa, se hace entrega de un algoritmo funcional que genera un archivo de mil estudiantes divididos por semestre y edad, los cuales serán clasificados en grupos por semestre, se repartirán en grupos de materias y de cursos de las mismas. Finalmente se generará una carpeta llamada ‘CarpetaArchivosTrabajoFinal’ que contendrá sub carpetas con todos los semestres, a su vez estos contendrán los cursos por semestre y dentro de estos los grupos de cada asignatura. Para llegar al plan de versionado 1.0.0 en la elaboración del logaritmo tuvimos que:
  + *0.1.0*: *Comenzar a trabajar sobre el algoritmo dado por el profesor. Inicialmente se ejecuta para generar el archivo de estudiantes y posteriormente se empieza a trabajar en el código propio de Nas Programming, empezando por definir las funciones para la asignación de cursos, el límite de estudiantes por curso según su semestre, y la asignación de horas de trabajo independiente y horas de trabajo del docente según el número de créditos.*
  + *0.2.0: Para el siguiente avance de la versión, se define una función que lea el archivo de estudiantes y luego otra que asigne a los estudiantes a los semestres según la proporción brindada.*
  + *0.3.0: Continuamos definiendo funciones para crear el código único de cada curso, calcular las horas de trabajo independiente y horas de trabajo del docente, y una función que permita crear los archivos con la información de cada curso. También se definen las funciones de logging para que tome la información del usuario y el log de eventos que registra el momento de ejecución de cada función.*
  + *0.4.0: Luego se define la función principal que llama a varias de las funciones ya definidas anteriormente para ejecutarlas y dentro de esta se definen 10 diccionarios (10 semestres) con los cursos por semestre (Se define la malla curricular).*
  + *0.4.1: Se termina de montar la malla curricular.*
  + *0.5.0: Se genera un ciclo for que genere los archivos de cada semestre, se registra la finalización de la ejecución del código y con un if se llama a la función principal para que esta sea ejecutada.*
  + *1.0.0: Versión final del algoritmo.*

*(Todas las actividades se van comentando a medida que se van generando para el entendimiento de lo que hace cada función tanto para nosotros como estudiantes como para el profesor)*