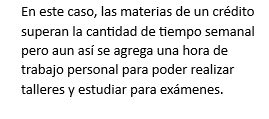
# Parcial 1

1. Para organizar la información de cada curso del pensum usaremos el siguiente formato: Código del curso, Cantidad de créditos, Horas de clase, horas de estudio. Esto para cada curso del pensum.

Text

Description automatically generated with medium confidence

1. Para implementar el algoritmo se deben realizar las siguientes preguntas al usuario.

* ¿A qué horas normalmente se acuesta?
  + Horarios de sueño predeterminado

ó

* + Hora en que se acuesta
* ¿Quiere reservar un horario semanal para actividades fuera de la universidad?
  + Si
  + ¿Qué hora del día NO tiene disponibles para estudiar?
    - Día
    - Hora
  + No
* ¿Cuántas materias tiene actualmente matriculadas?
* Código de materia
  + Cantidad de días que asiste a clase
    - Día
    - Hora inicio
    - Hora final

1. Para almacenar la información

* Int pensum[cantidad de materias][4]={{código del curso, cantidad de créditos, horas de clase, horas de estudio}}
  + Para este arreglo usaremos memoria dinámica
* int L[24]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

int M[[24]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

int m[24]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

int J[24]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

int V[24]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

int S[24]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

int D[24]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

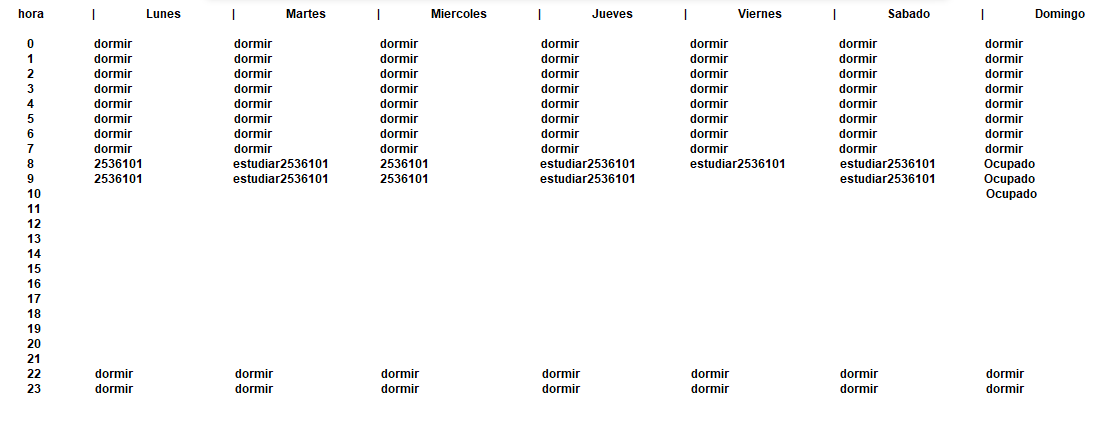
* + Cada uno representa un día de la semana y cada uno de sus valores una hora del día.

1. Funciones para usar:

* verificar\_horas\_clase (Código del curso, cantidad de horas de la clase)
  + Compara la cantidad de horas escritas por el usuario con la del archivo para asegurarse de que son iguales
* verificar\_hora\_valida(hora)
  + Se asegura que el usuario coloque valores desde 0 hasta 23
* verificar\_espacio\_libre (Dia, Hora)
  + Revisa que el horario digitado se encuentre en 0

1. implementación:

* Defino las variables y funciones.
* Doy a elegir al usuario entre un horario de sueño predeterminado o personalizado. En caso de ser personalizado pido la hora en que se acuesta (verifico que sea una hora valida), calculo 9 horas a partir de la hora dada, remplazo en las listas el rango de horas por unos (1).
* Pregunto al usuario si desea reservar tiempo para actividades fuera de la universidad. Si el casi es si en un ciclo le doy al usuario a elegir entre: rango de tiempo que no pueda estudiar o una hora especifica por día (verifico que sea una hora valida) (verifico que este libre) y remplazo en las listas esos horarios por unos (2), pregunto al usuario si desea reservar más horas o continuar con la ejecución del código.
* Pido al usuario ingresar el número de materias que ve.
* Inicio un ciclo que le pida: código de la materia, cantidad de días que ve clase
  + lo ingreso a otro ciclo que le pida el día (L M m J V S D) y el rango de horas (Hora de inicio y hora de finalización) (verifico que la hora sea válida) (verifico que las horas de clase coincidan con el archivo) (Verifico que la hora este libre).
    - Muestro la información administrada y pregunto al usuario si esta correcta.
* Reviso si es posible distribuir la cantidad de horas de trabajo personal en la cantidad de horas disponibles.
  + Si no es el caso se le dice al usuario que no tiene tiempo suficiente para la cantidad de materias que ve y el tiempo que tiene reservado finalizando con la ejecución del programa.
* Comienzo a distribuir las horas de estudio de forma aleatoria siguiendo los siguientes patrones:
  + Comenzar por los días con más horas libres
    - Tener un contador de horas libre por día
  + Tratar de no poner más de 4 horas continuas una misma materia
    - Verificar el código del curso que se encuentra en la hora anterior
  + Evitar ver todas las materias el mismo día
    - Verificar la cantidad de códigos diferentes cada día
  + Verificar si hay algún día con más horas disponibles en cada iteración
    - Repetir un ciclo con una condición que diga que día tiene mas horas disponibles
  + Comenzar siempre por las horas de la mañana
    - Comenzar por las posiciones mas bajas
  + Elegir una materia de manera aleatoria cuando tenga la misma cantidad de horas aun no asignadas
    - Usar random para que el software no realice siempre el mismo tipo de cronograma respetando los patrones
  + Elegir un día aleatorio de manera aleatoria cuando tenga la misma cantidad de horas disponibles
    - Usar random para que el software no realice siempre el mismo tipo de cronograma respetando los patrones

6)Ejemplo salido esperada: