

MANUAL DEL ALGORITMO PARA: **GENERADOR DE HORARIOS**

BY. PAULINA AMEZCUA

Comprender este código es bastante sencillo ya que se realizó con métodos fáciles para la realización más veloz de este, es realmente un programa sencillo para aquellos que tienen una mayor experiencia programando.

No se trabajó con archivos, todo está impuesto en la misma consola de Python, y ahí mismo arroja los resultados, los cuales deberían ensamblarse con el frontend.

//Esta librería se necesita importar ya que nos ayudará más tarde con la selección de horario

```
import random
```

//Abrimos primero una lista vacía llamada profesores

```
profesores = []
```

//En esta sección vamos a imprimir en la pantalla todos los horarios que se pueden elegir por si es que el profesor no puede ir a dar clase, el usuario seleccionará uno

```
def mostrar_menu_horas_no_trabajo():
```

```
    print("\nSeleccione el horario en el cual el profesor no puede trabajar:")
```

```
    print("\n1- 09:00 am - 10:55 am LUNES")
```

```
    print("2- 11:00 am - 12:55 pm LUNES")
```

```
    print("3- 01:00 pm - 02:55 pm LUNES")
```

```
    print("4- 03:00 pm - 04:55 pm LUNES\n")
```

```
    print("\n5- 09:00 am - 10:55 am MARTES")
```

```
    print("6- 11:00 am - 12:55 pm MARTES")
```

```
    print("7- 01:00 pm - 02:55 pm MARTES")
```

```
    print("8- 03:00 pm - 04:55 pm MARTES\n")
```

```
    print("\n9- 09:00 am - 10:55 am MIERCOLES")
```

```
    print("10- 11:00 am - 12:55 pm MIERCOLES")
```

```
    print("11- 01:00 pm - 02:55 pm MIERCOLES")
```

```
    print("12- 03:00 pm - 04:55 pm MIERCOLES\n")
```

```

print("\n13 09:00 am - 10:55 am JUEVES")
print("14- 11:00 am - 12:55 pm JUEVES")
print("15- 01:00 pm - 02:55 pm JUEVES")
print("16- 03:00 pm - 04:55 pm JUEVES\n")
print("\n17 09:00 am - 10:55 am VIERNES")
print("18- 11:00 am - 12:55 pm VIERNES")
print("19- 01:00 pm - 02:55 pm VIERNES")
print("20- 03:00 pm - 04:55 pm VIERNES\n")

```

//Aquí, sabiendo que carrera imparte el profesor las imprimimos para que el usuario pueda escoger una de las dos, esta dentro de una función como el caso anterior ya que son parte de una selección del menú

```

def mostrar_menu_carreras():
    print("\nSeleccione la carrera que imparte el profesor:")
    print("\n1- INGENIERIA EN INFORMATICA")
    print("2- INGENIERIA EN COMPUTACION\n")

```

//Aquí presentamos otra función, la cual muestra las materias dependiendo la opción anteriormente seleccionada

```

def mostrar_menu_materias(carrera):
    print("\nElija la materia que el profesor imparte:")

```

//Aquí se muestran las materias en caso de haber seleccionado la opción de informática, en total son 20 materias de esta carrera

//Aclaremos que los "SI" de lado de cada materia quieren decir que es una materia que requiere implementación de computadoras, por los cuales se deberían conjuntar a aulas que efectivamente cuenten con computadoras pero no se realizó esa sección

```

if carrera == "1": //Si la carrera es 1

    # Grupo de 20 materias para INGENIERIA EN INFORMATICA

    materias = [ //Abrimos el listado de las materias

        "LOGICA Y CONJUNTOS", "INTRODUCCION A LA FISICA", "TALLER DE INTRODUCCION A LA COMPUTACION",

        "TALLER DE COMUNICACION ORAL Y ESCRITA", "INTRODUCCION A LA COMPUTACION (SI)", "TALLER DE REDACCION",

        "INTRODUCCION A LA PROGRAMACION (SI)", "ELEMENTOS DE PROBABILIDAD Y ESTADISTICA",

```

"TALLER DE PROGRAMACION ORIENTADA A OBJETOS (SI)", "AUDITORIA DE SISTEMAS (SI)",
"SISTEMAS DE INFORMACION FINANCIEROS", "SISTEMAS DE INFORMACION PARA LA
MANUFACTURA",
"TALLER DE ESTRUCTURA DE DATOS", "ADMINISTRACION DE RECURSOS HUMANOS",
"PROGRAMACION Y LOGICA FUNCIONAL (SI)",
"LEGISLACION EN INFORMATICA", "TALLER DE PROGRAMACION DE SISTEMAS (SI)",
"SISTEMAS DE INFORMACION ADMINISTRATIVOS", "INVESTIGACION DE OPERACIONES", "GRAFICAS
POR COMPUTADORA (SI)"

]

elif carrera == "2": //Si la carrera es 2

Grupo de 43 materias para INGENIERIA EN COMPUTACION

materias = [
"FUNDAMENTOS DE LA PROGRAMACION (SI)", "ETICA Y LEGISLACION", "LOGICA MATEMATICA",
"PRECALCULO", "FUNDAMENTOS DE FISICA", "INTRODUCCION A LA INGENIERIA", "PROGRAMACION
ESTRUCTURADA (SI)",
"MATEMATICA DISCRETA", "CALCULO DIFERENCIAL E INTEGRAL", "MECANICA",
"ADMINISTRACION DE PROYECTOS TECNOLOGICOS",
"EXPRESION ORAL Y ESCRITA", "PROGRAMACION ORIENTADA A OBJETOS (SI)", "ALGEBRA LINEAL",
"ECUACIONES DIFERENCIALES", "CIRCUITOS ELECTRICOS Y ELECTROMAGNETISMO (SI)", "SISTEMAS
DIGITALES (SI)",
"ADMINISTRACION", "ESTRUCTURA DE DATOS (SI)", "PROBABILIDAD Y ESTADISTICA", "METODOS
NUMERICOS",
"ARQUITECTURA DE COMPUTADORAS (SI)", "PROGRAMACION PARA INTERNET (SI)", "LIDERAZGO Y
EMPRENDIMIENTO",
"ANALISIS DE ALGORITMOS (SI)", "BASES DE DATOS (SI)", "SISTEMA OPERATIVOS (SI)",
"FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL (SI)", "REDES DE COMPUTADORAS (SI)",
"SEMINARIO DE INTEGRACION DE PROTOCOLO",
"INNOVACION TECNOLOGICA", "INTERACCION HUMANO COMPUTADORA (SI)", "INGENIERIA DE
SOFTWARE (SI)",
"PROGRAMACION DE BAJO NIVEL (SI)", "TEORIA DE LA COMPUTACION", "SEMINARIO DE
INTEGRACION DE DESARROLLO",
"LABORATORIO ABIERTO: DISEÑO (SI)", "COMPILADORES (SI)", "SEGURIDAD EN LA INFORMACION
(SI)",
"LABORATORIO ABIERTO: CONSTRUCCION (SI)", "PROGRAMACION PARALELA Y CONCURRENTE (SI)",

"SEMINARIO DE INTEGRACION COMUNICACION", "LABORATORIO ABIERTO: PRUEBA (SI)"

]

// Después del bloque if-elif, donde se determina la carrera seleccionada y se asigna la lista de materias correspondiente a esa carrera, hay un bloque else.

Este bloque se ejecutará si la variable carrera no es ni "1" ni "2", es decir, si no coincide con ninguna de las condiciones anteriores.

else:

materias = []

// Se utiliza un bucle for para iterar sobre las materias en la lista materias e imprimir cada materia junto con su índice.

El uso de enumerate con start=1 asegura que los índices comiencen desde 1 en lugar de 0, lo cual es más amigable para el usuario.

for index, materia in enumerate(materias, start=1):

print(f"{index}. {materia}")

//Aquí tenemos otra función pero ahora para agregar al profesor, este es completamente manual, se agrega tal cual el nombre y no un numero como en los casos anteriores.

def agregar_profesor():

nombre = input("\nIngrese el nombre del profesor: ")

//Todo este bloque siguiente de código es para que se logre seleccionar un horario diferente al que el profesor eligió (Ya que el que se eligió por el usuario es cuando el profesor no puede impartir la clase, por lo tanto para generar un horario de trabajo se debe seleccionar uno de los horarios restantes

mostrar_menu_horas_no_trabajo() //Aquí llamamos a la función llamada
mostrar_menu_horas_no_trabajo() donde se mostrará la lista de las horas que hay en total

user_input = int(input("Seleccione una opción: ")) //Aquí se solicita ingresar una opción seleccionando un número. La entrada del usuario se captura como una cadena (string), y luego se convierte a un entero usando int() para asegurarse de que la entrada sea un número entero. El resultado se almacena en la variable user_input.

Ya que es necesario tener valores enteros para lo siguiente que se realizará, lo cual es el uso de la librería random

while user_input < 1 or user_input > 20: //Aquí especificamos que sea un numero entre 1 y 20, el bucle while se ejecuta si la entrada del usuario está fuera del rango

```
print("Entrada inválida. Por favor introduzca un número entre 1 y 20.") //Entonces arrojaría una nota sobre que la opción es invalida
```

```
user_input = int(input("Seleccione una opción: ")) //Y nos regresa al inicio del bucle
```

```
random_number = random.randint(1, 20) //Genera un número aleatorio entre 1 y 20 y lo almacena en la variable random_number
```

```
//Este bucle while asegura que el número aleatorio no sea igual a la opción seleccionada por el usuario. Si por casualidad son iguales, se genera un nuevo número aleatorio hasta que sean diferentes.
```

```
while random_number == user_input:
```

```
    random_number = random.randint(1, 20)
```

```
//Aquí están todas las posibles opciones a elegir de manera random, los numero principales en enteros para poder trabajar de manera más libre con la librería
```

```
//Aquí puede tomarse como ser de fuerza bruta ya que fluctúa las combinaciones seleccionables hasta que encuentre la que no sea la seleccionada por el usuario
```

```
horas_no_trabajo = ""
```

```
if random_number == 1:
```

```
    horas_no_trabajo = "09:00 am - 10:55 am LUNES"
```

```
elif random_number == 2:
```

```
    horas_no_trabajo = "11:00 am - 12:55 pm LUNES"
```

```
elif random_number == 3:
```

```
    horas_no_trabajo = "01:00 pm - 02:55 pm LUNES"
```

```
elif random_number == 4:
```

```
    horas_no_trabajo = "03:00 pm - 04:55 pm LUNES"
```

```
elif random_number == 5:
```

```
    horas_no_trabajo = "09:00 am - 10:55 am MARTES"
```

```
elif random_number == 6:
```

```
    horas_no_trabajo = "11:00 am - 12:55 pm MARTES"
```

```
elif random_number == 7:
```

```
    horas_no_trabajo = "01:00 pm - 02:55 pm MARTES"
```

```
elif random_number == 8:
```

```
    horas_no_trabajo = "03:00 pm - 04:55 pm MARTES"
```

```
elif random_number == 9:
```

```
    horas_no_trabajo = "09:00 am - 10:55 am MIERCOLES"
```

```

elif random_number == 10:
    horas_no_trabajo = "11:00 am - 12:55 pm MIERCOLES"
elif random_number == 11:
    horas_no_trabajo = "01:00 pm - 02:55 pm MIERCOLES"
elif random_number == 12:
    horas_no_trabajo = "03:00 pm - 04:55 pm MIERCOLES"
elif random_number == 13:
    horas_no_trabajo = "09:00 am - 10:55 am JUEVES"
elif random_number == 14:
    horas_no_trabajo = "11:00 am - 12:55 pm JUEVES"
elif random_number == 15:
    horas_no_trabajo = "01:00 pm - 02:55 pm JUEVES"
elif random_number == 16:
    horas_no_trabajo = "03:00 pm - 04:55 pm JUEVES"
elif random_number == 17:
    horas_no_trabajo = "09:00 am - 10:55 am VIERNES"
elif random_number == 18:
    horas_no_trabajo = "11:00 am - 12:55 pm VIERNES"
elif random_number == 19:
    horas_no_trabajo = "01:00 pm - 02:55 pm VIERNES"
elif random_number == 20:
    horas_no_trabajo = "03:00 pm - 04:55 pm VIERNES"
else:
    print("Opción seleccionada")

```

//El usuario selecciona una carrera y luego, si la opción es válida (1 o 2), se pide que seleccione una materia.

```

mostrar_menu_carreras() //Se llama a la función

opcion_carrera = input("Seleccione una opción: ") //Se le solicita al usuario que ingrese una opción y
almacena la entrada en la variable opcion_carrera. Esta entrada se mantiene como una cadena (string).

//Evalúa la opción seleccionada por el usuario.

```

```
carrera = ""
```

//Si la opción es "1", se asigna la cadena "INGENIERIA EN INFORMATICA" a la variable carrera.

```
if opcion_carrera == "1":
```

```
    carrera = "INGENIERIA EN INFORMATICA"
```

Si la opción es "2", se asigna la cadena "INGENIERIA EN COMPUTACION" a la variable carrera.

```
elif opcion_carrera == "2":
```

```
    carrera = "INGENIERIA EN COMPUTACION"
```

Si la opción no es ni "1" ni "2", imprime un mensaje indicando que la opción no es válida y deja la variable carrera como una cadena vacía.

```
else:
```

```
    print("Opción no válida. Se dejará la carrera vacía.")
```

//Aquí ponemos las carreras a seleccionar llevando un proceso parecido al anterior de si escoge la opción 1 o la 2, es ligeramente igual a el paso anterior en esta sección, pero se usa en este diferente contexto. La opción anterior era para mostrar materias según la carrera seleccionada por el usuario.

Esta parte se usa después de que el usuario selecciona la carrera y antes de que elija una materia específica.

```
if opcion_carrera in ["1", "2"]:
```

```
    mostrar_menu_materias(opcion_carrera)
```

```
    opcion_materia = input("Seleccione una opción: ")
```

```
    materia = ""
```

```
    if 1 <= int(opcion_materia) <= (20 if opcion_carrera == "1" else 43):
```

```
        materias = [
```

```
            "LOGICA Y CONJUNTOS", "INTRODUCCION A LA FISICA", "TALLER DE INTRODUCCION A LA COMPUTACION",
```

```
            "TALLER DE COMUNICACION ORAL Y ESCRITA", "INTRODUCCION A LA COMPUTACION (SI)", "TALLER DE REDACCION",
```

```
            "INTRODUCCION A LA PROGRAMACION (SI)", "ELEMENTOS DE PROBABILIDAD Y ESTADISTICA",
```

```
            "TALLER DE PROGRAMACION ORIENTADA A OBJETOS (SI)", "AUDITORIA DE SISTEMAS (SI)",
```

```
            "SISTEMAS DE INFORMACION FINANCIEROS", "SISTEMAS DE INFORMACION PARA LA MANUFACTURA",
```

```

        "TALLER DE ESTRUCTURA DE DATOS", "ADMINISTRACION DE RECURSOS HUMANOS",
        "PROGRAMACION Y LOGICA FUNCIONAL (SI)",

        "LEGISLACION EN INFORMATICA", "TALLER DE PROGRAMACION DE SISTEMAS (SI)",

        "SISTEMAS DE INFORMACION ADMINISTRATIVOS", "INVESTIGACION DE OPERACIONES", "GRAFICAS
        POR COMPUTADORA (SI)"

    ] if opcion_carrera == "2" else [

        "FUNDAMENTOS DE LA PROGRAMACION (SI)", "ETICA Y LEGISLACION", "LOGICA MATEMATICA",

        "PRECALCULO", "FUNDAMENTOS DE FISICA", "INTRODUCCION A LA INGENIERIA",
        "PROGRAMACION ESTRUCTURADA (SI)",

        "MATEMATICA DISCRETA", "CALCULO DIFERENCIAL E INTEGRAL", "MECANICA",

        "ADMINISTRACION DE PROYECTOS TECNOLOGICOS",

        "EXPRESION ORAL Y ESCRITA", "PROGRAMACION ORIENTADA A OBJETOS (SI)", "ALGEBRA LINEAL",

        "ECUACIONES DIFERENCIALES", "CIRCUITOS ELECTRICOS Y ELECTROMAGNETISMO (SI)", "SISTEMAS
        DIGITALES (SI)",

        "ADMINISTRACION", "ESTRUCTURA DE DATOS (SI)", "PROBABILIDAD Y ESTADISTICA", "METODOS
        NUMERICOS",

        "ARQUITECTURA DE COMPUTADORAS (SI)", "PROGRAMACION PARA INTERNET (SI)", "LIDERAZGO Y
        EMPRENDIMIENTO",

        "ANALISIS DE ALGORITMOS (SI)", "BASES DE DATOS (SI)", "SISTEMA OPERATIVOS (SI)",

        "FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL (SI)", "REDES DE COMPUTADORAS (SI)",

        "SEMINARIO DE INTEGRACION DE PROTOCOLO",

        "INNOVACION TECNOLOGICA", "INTERACCION HUMANO COMPUTADORA (SI)", "INGENIERIA DE
        SOFTWARE (SI)",

        "PROGRAMACION DE BAJO NIVEL (SI)", "TEORIA DE LA COMPUTACION", "SEMINARIO DE
        INTEGRACION DE DESARROLLO",

        "LABORATORIO ABIERTO: DISEÑO (SI)", "COMPILADORES (SI)", "SEGURIDAD EN LA INFORMACION
        (SI)",

        "LABORATORIO ABIERTO: CONSTRUCCION (SI)", "PROGRAMACION PARALELA Y CONCURRENTES (SI)",

        "SEMINARIO DE INTEGRACION COMUNICACION", "LABORATORIO ABIERTO: PRUEBA (SI)"

    ]

    materia = materias[int(opcion_materia) - 1]

else:

    print("Opción no válida. Se dejará la materia vacía.")

```


//Aquí se almacenan los datos en las diferentes variables, es un diccionario de datos que se han ingresado del profesor

```
profesor = {  
    "Nombre": nombre,  
    "Horas_no_trabajo": horas_no_trabajo,  
    "Carrera": carrera,  
    "Materia": materia  
}  
profesores.append(profesor)
```

//Indica que se guardaron los datos bien

```
print("Profesor agregado exitosamente.")
```

Se define una función llamada generar_tabla_horarios que imprime una tabla de horarios para profesores basada en la información almacenada en la lista profesores.

```
def generar_tabla_horarios():
```

```
    print("\nTabla de Horarios para Profesores:")
```

//Imprime el encabezado de la tabla con cuatro columnas: "Nombre", "Horas no trabajo", "Carrera", y "Materia". La cadena de formato "{:<20}" asegura que cada columna tenga un ancho de 20 caracteres y esté justificada a la izquierda.

```
    print("{:<20} {:<20} {:<20} {:<20}".format("Nombre", "Horas no trabajo",  
                                             "Carrera", "Materia"))
```

//Itera sobre la lista de profesores utilizando enumerate para obtener tanto el índice (i) como el diccionario del profesor (profesor).

```
    for i, profesor in enumerate(profesores):
```

//Imprime una fila de la tabla para cada profesor.

Los valores del diccionario (profesor["Nombre"], profesor["Horas_no_trabajo"], etc.) acomodan en columnas con tamaños similares al encabezado.

```
    print("{:<20} {:<20} {:<20} {:<20}".format(profesor["Nombre"],  
                                             profesor["Horas_no_trabajo"], profesor["Carrera"],  
                                             profesor["Materia"]))
```

//Verifica si la lista de profesores está vacía

```

if not profesores:

    print("No hay profesores en la lista.")

    return

//Nos imprime la indicación de que pulsar si se desea o no eliminar a un profesor, en este caso cualquier
numero sería cualquier profesor y 0 sería no eliminar

opcion_eliminar = input("\nIngrese el número del profesor que desea eliminar (o '0' para volver): ")

//Si la opción es "0", la función se detiene y retorna, lo que significa que el usuario eligió no realizar ninguna
eliminación.

if opcion_eliminar == "0":

    return


try: //Se encarga de manejar posibles errores

    indice_profesor = int(opcion_eliminar) - 1

    if 0 <= indice_profesor < len(profesores): //Convierte la opción proporcionada por el usuario
(opcion_eliminar) a un entero y resta 1 para obtener el índice de la lista profesores.

        profesor_eliminado = profesores.pop(indice_profesor) // Verifica si el índice que se calculó está
dentro de los límites de la lista.

        //Se presume que se eliminó

        print(f"Profesor '{profesor_eliminado['Nombre']}' eliminado exitosamente.")

    else:

        //Informa si no existe el profesor en el numero ingresado

        print("Número de profesor no válido.")

        //Aparece así si el usuario ingresa algo que no sea un número

except ValueError:

    print("Por favor, ingrese un número válido.")

//Otra función, esta muestra el menú principal del programa

def main():

    while True:

        print("\nMenú:")

        print("\n1. Agregar profesor")

        print("\n2. Generar tabla de horarios")

        print("\n3. Salir\n")

```

//El submenú para cada caso

```
opcion_menu_principal = input("\nSeleccione una opción: ")
```

//Si la opción es 1 salen todas esas opciones

```
if opcion_menu_principal == "1":
```

```
    while True:
```

```
        print("\nSubmenú - Agregar Profesor:")
```

```
        print("\n1. Nombre del profesor")
```

```
        print("2. Horas que no puede trabajar")
```

```
        print("3. Carrera")
```

```
        print("4. Volver al menú principal\n")
```

```
    opcion_submenu_agregar_profesor = input("\nSeleccione una opción: ")
```

```
    if opcion_submenu_agregar_profesor == "1":
```

```
        agregar_profesor()
```

//Aquí se van generando conforme se vayan llenando las opciones

```
    elif opcion_submenu_agregar_profesor == "2":
```

```
        mostrar_menu_horas_no_trabajo()
```

```
        mostrar_menu_carreras()
```

```
        opcion_carrera = input("\nSeleccione una opción: ")
```

```
        if opcion_carrera in ["1", "2"]:
```

```
            mostrar_menu_materias(opcion_carrera)
```

```
        else:
```

```
            print("Opción no válida.")
```

```
    elif opcion_submenu_agregar_profesor == "3":
```

```
        mostrar_menu_carreras()
```

```
        opcion_carrera = input("\nSeleccione una opción: ")
```

```
        if opcion_carrera in ["1", "2"]:
```

```
            mostrar_menu_materias(opcion_carrera)
```

```
        else:
```

```
            print("Opción no válida.")
```

```
elif opcion_submenu_agregar_profesor == "4":
    mostrar_menu_carreras()
    opcion_carrera = input("\nSeleccione una opción: ")
    if opcion_carrera in ["1", "2"]:
        mostrar_menu_materias(opcion_carrera)
    else:
        print("Opción no válida.")
elif opcion_submenu_agregar_profesor == "5":
    break
else:
    print("Opción no válida. Inténtelo de nuevo.")
```

//Generador de la tabla

```
elif opcion_menu_principal == "2":
    generar_tabla_horarios()
```

//Opción de finalización

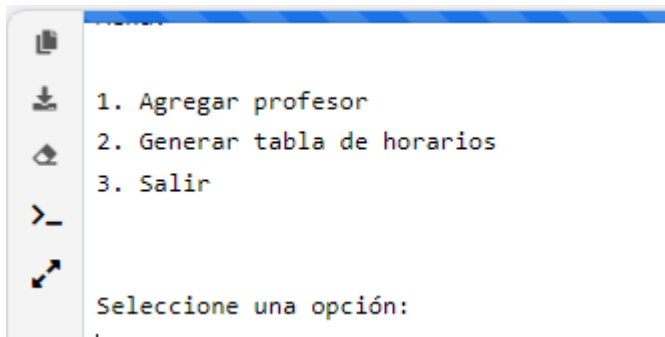
```
elif opcion_menu_principal == "3":
    print("Saliendo del programa. FIN")
    break
```

```
else:
    print("Opción no válida. Inténtelo de nuevo.")
```

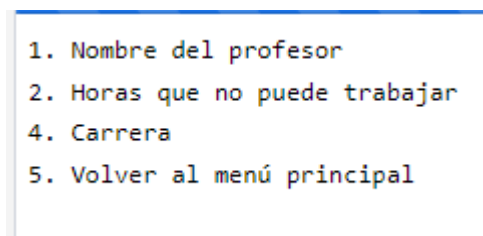
//Esto es útil porque organiza el de inicialización (como la definición de funciones y variables).

```
if __name__ == "__main__":
    main()
```

Los resultados que se arrojan de este código son los siguientes:



Aquí vemos el menú principal con las opciones principales que justo eran las esperadas, agregar profesor y generar una tabla, vamos paso por paso eligiendo lo que se debe hacer.



Después se nos despliega un submenú, con estas opciones, si presionamos 1 nos genera la opción de agregar el nombre del profesor de manera manual, para una mayor agilidad yo solo coloqué Juan

```
Seleccione una opción:
```

```
1
```

```
Ingrese el nombre del profesor:
```

```
Ingrese el nombre del profesor:
```

```
Juan
```

```
Seleccione el horario en el cual el profesor no puede trabajar:
```

```
1- 09:00 am - 10:55 am LUNES
```

```
2- 11:00 am - 12:55 pm LUNES
```

Después se despliega una lista de horarios, de las horas con los días a un lado, el usuario debe elegir alguno de todos si es que tiene algún horario con poca posibilidad de asistencia.

17 09:00 am - 10:55 am VIERNES
18- 11:00 am - 12:55 pm VIERNES
19- 01:00 pm - 02:55 pm VIERNES
20- 03:00 pm - 04:55 pm VIERNES

Seleccione una opción:

1

Aquí vemos más claramente el final del listado, elegimos la opción 17, o sea un viernes de 9:00 am a 10:55am

Seleccione una opción:

17

Seleccione la carrera que imparte el profesor:

- 1- INGENIERIA EN INFORMATICA
- 2- INGENIERIA EN COMPUTACION

En seguida para seguir con el llenado de formulario nos arroja esta opción, sobre cuál es la carrera que el profesor imparte, entonces seleccionamos 1 “ingeniería en informática” por ejemplo, lo cual hace que nos arroje un listado completo de las materias (El sí de al lado era para implementarlo a las que si necesitan computadora, para después agregarlo a las aulas que si tenían)

Elija la materia que el profesor imparte:

- 1. LOGICA Y CONJUNTOS
- 2. INTRODUCCION A LA FISICA
- 3. TALLER DE INTRODUCCION A LA COMPUTACION
- 4. TALLER DE COMUNICACION ORAL Y ESCRITA
- 5. INTRODUCCION A LA COMPUTACION (SI)
- 6. TALLER DE REDACCION

- 16. LEGISLACION EN INFORMATICA
- 17. TALLER DE PROGRAMACION DE SISTEMAS (SI)
- 18. SISTEMAS DE INFORMACION ADMINISTRATIVOS
- 19. INVESTIGACION DE OPERACIONES
- 20. GRAFICAS POR COMPUTADORA (SI)

Seleccione una opción:

12|

Seleccionamos la 12

- 1. Nombre del profesor
- 2. Horas que no puede trabajar
- 4. Carrera
- 5. Volver al menú principal

Seleccione una opción:

|

Después como el formulario está completo, colocamos el numero 5 para simplemente regresar a poder generar la tabla con los datos ya ingresados.

Tabla de Horarios para Profesores:

Nombre	Horas no trabajo	Carrera	Materia
Juan	11:00 am - 12:55 pm	LUNES	INGENIERIA EN INFORMATICA SISTEMAS DE INFORMACION PARA LA MANUFACTURA

Seleccionamos la opción dos de generar tabla, y aquí tenemos efectivamente la tabla realizada, el formato aquí no era muy importante hasta que llegase el frontend.