

Simulación de Horario de Clases

Manual de Usuario

Cifuentes Argueta, Sergio Daniel

Centro Universitario de Occidente

Quetzaltenango, Septiembre 2023

INDICE

INDICE.....	2
INSTALACION.....	3
Instale Docker.....	3
Clonar el repositorio de simulación.....	3
Cree la imagen de Docker.....	3
Ejecute el contenedor Docker.....	3
REQUISITOS DE DATOS DE ENTRADA.....	4
profesores.csv.....	4
materias.csv.....	5
salons.csv.....	5
dispProfesor.csv.....	5
dispSalon.csv.....	6
profMateria.csv.....	7
asignaciones.csv.....	7
CONFIGURACIÓN DEL MODELO.....	7
EJECUCIÓN DE SIMULACIONES.....	8
Cargar Datos.....	8

INSTALACION

Para ejecutar la simulación de programación del aula utilizando Docker y una imagen de Conda, siga estas instrucciones de instalación paso a paso:

Instale Docker

Si aún no tiene Docker instalado en su sistema, puede descargarlo e instalarlo siguiendo la guía de instalación oficial para su sistema operativo:

[Guía de instalación de Docker para Windows](#)

[Guía de instalación de Docker para macOS](#)

[Guía de instalación de Docker para Linux](#)

Asegúrese de que Docker esté en funcionamiento antes de continuar con los siguientes pasos.

Clonar el repositorio de simulación

Clona el repositorio de simulación desde el repositorio de código fuente (por ejemplo, GitHub) a tu máquina local:

```
git clone https://github.com/SergioCifuentes/schedule-sim
```

Cree la imagen de Docker

Navegue hasta el directorio raíz del repositorio clonado y cree la imagen de Docker utilizando el Dockerfile proporcionado. Reemplace el nombre de su imagen con el nombre de la imagen que desee:

```
cd your-simulation docker build -t your-image-name .
```

Ejecute el contenedor Docker

Ahora que ha creado la imagen de Docker, puede ejecutar un contenedor de Docker desde ella. Reemplace `your-image-name` con un nombre significativo para su contenedor:

```
docker run -it --name your-container-name your-image-name
```

Y finalmente podrás acceder a la aplicación mediante localhost:5000.

REQUISITOS DE DATOS DE ENTRADA

La carga de archivos CSV es fundamental para la funcionalidad de la simulación. Aquí verás todos los archivos necesarios para lograr una simulación correcta.

Para comenzar se necesita que todos los archivos se encuentren en un solo directorio, esto para que al momento de cargar los archivos solo se tenga que pasar el PATH del directorio y estos son los archivos que debe contener.

profesores.csv

Contiene información de los docentes y sus respectivas carreras.

```
id,nombre,carrera
1,"Francisco Rojas","Sistemas"
2,"Moises Granados","Sistemas"
3,"Pedro Domingo","Sistemas"
4,"Mauricio Lopez","Sistemas"
5,"Jose Maldonado","Civil"
6,"Humberto","Civil"
7,"Carlos Chavarria","Civil"
```

materias.csv

Contiene información de los diferentes cursos como lo es el semestre y carrera.

```
id,nombre,carrera,semestre,obli,no_periodos
169,"Matematica Basica 1","Comun",1,true,2
17,"Social Humanista 1","Comun",1,false,1
216,"Quimica 1","Comun",1,true,1
170,"Matematica Basica 2","Comun",2,true,2
29,"Social Humanista 2","Comun",2,true,1
5,"Tecnicas De Estudio","Comun",2,true,1
147,"Fisica Basica","Comun",2,true,1
107,"Matematica Intermedia1","Comun",3,true,2
150,"Fisica 1","Comun",3,true,1
112,"Matematica Intermedia 2","Comun",4,true,1
```

salons.csv

Contiene información de los salones que dispone el módulo, incluyendo capacidades de estudiantes.

```
id,nombre,capacidad_comoda,capacidad_maxima
1,"Salon 1",40,50
2,"Salon 2",40,50
3,"Salon 3",40,50
4,"Salon 4",60,70
5,"Salon 5",60,70
```

dispProfesor.csv

Contiene horarios de los profesores, los cuales serán de suma importancia al momento de asignación.

```
id_profesor, inicio, fin
1, "12:00", "22:00"
2, "10:00", "12:00"
2, "14:00", "20:00"
3, "14:00", "22:00"
4, "14:00", "22:00"
5, "14:00", "22:00"
6, "14:00", "22:00"
```

dispSalon.csv

Contiene horarios de la disponibilidad de los salones, los cuales serán de suma importancia al momento de asignación

```
id_salon, inicio, fin
1, "12:00", "14:00"
1, "14:00", "22:00"
2, "14:00", "22:00"
3, "14:00", "22:00"
4, "14:00", "22:00"
5, "14:00", "22:00"
```

profMateria.csv

- Contiene información sobre la capacidad de cada profesor.

```
id_profesor,id_curso,fijo
1,2820,true
1,2796,false
1,2835,false
1,2819,true
1,2823,false
2,2796,true
2,2798,false
2,2796,false
2,2800,true
```

asignaciones.csv

Contiene información de las secciones de los cursos y el número de estudiantes asignados.

```
id_materia,sec,numero_estudiantes
169,"A",60
169,"B",60
169,"C",55
169,"D",45
17,"A",50
17,"B",50
17,"C",40
17,"D",40
216,"A",50
216,"B",50
```

CONFIGURACIÓN DEL MODELO

Antes de empezar con las simulaciones, se pueden hacer algunas configuraciones al modelo. Principalmente esto se logra modificando el archivo *constant.py* en donde podremos modificar, el nombre de las entradas, el PATH por defecto, el tiempo en minutos de cada periodo y el método para resolver el modelo.

```
#nombres de archivos
ASIGNACIONES = "asignaciones.csv"
DISP_PROF = "dispProfesor.csv"
DISP_SALON = "dispSalon.csv"
MATERIAS = "materias.csv"
PROFESORES = "profesores.csv"
PROF_MATERIA = "profMateria.csv"
SALONES = "salones.csv"

#Path
PATH="schedule-sim/resources/scripts/"

#System Variables
TIEMPO_PERIODO=60 #minutes

CBC="cbc"
CBC_PATH="C:\\Users\\sergi\\Documents\\Coding\\Python\\solvers\\Cbc-2.7.5-win64-intel11.1\\bin\\cbc.exe"
```

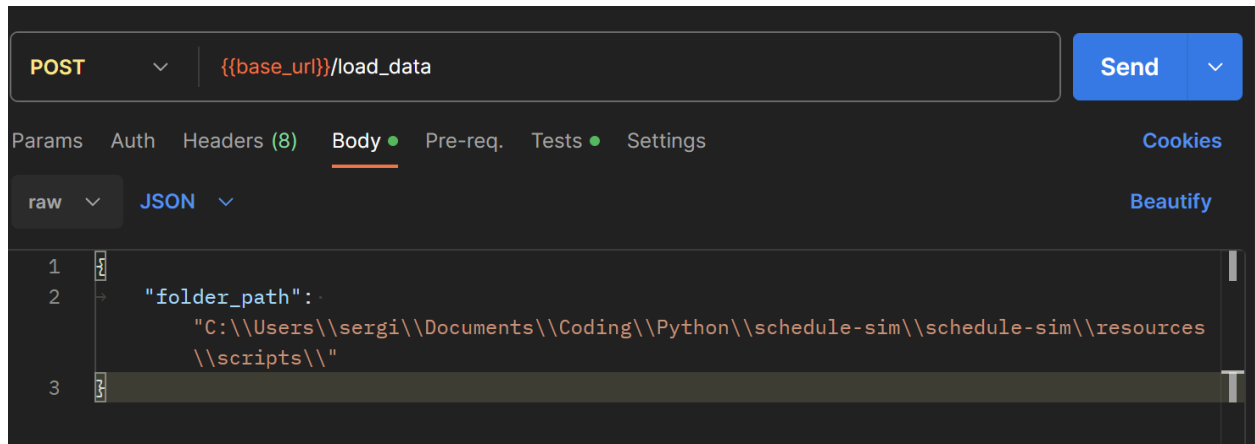
Finalmente podemos configurar el tiempo de ejecución de la simulación

EJECUCIÓN DE SIMULACIONES

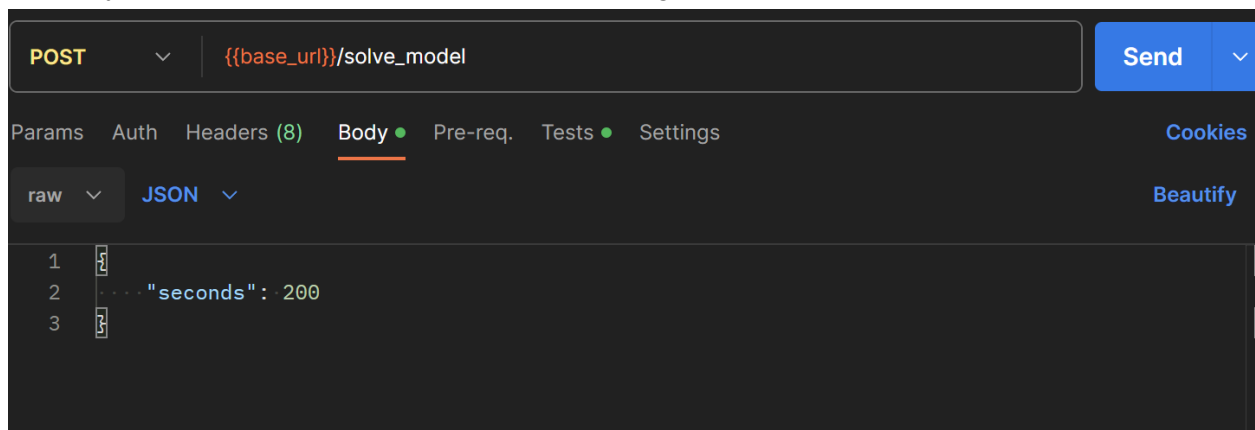
Puede interactuar con la simulación mediante solicitudes HTTP a los puntos finales de la API. Puede utilizar una herramienta como curl, un navegador web o herramientas de prueba de API especializadas como Postman para realizar estas solicitudes.

Cargar Datos

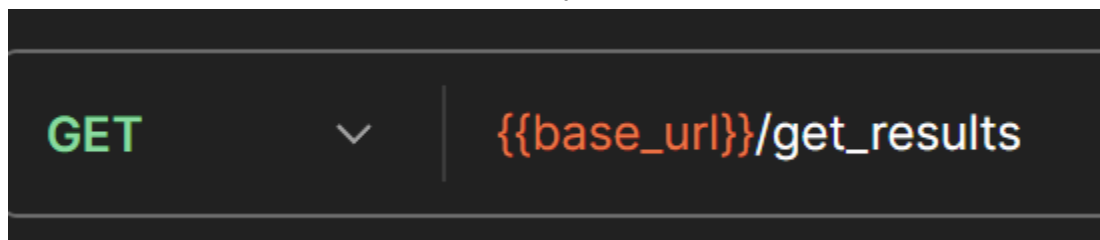
Para esto debemos de usar el POST /load_data.



Cuando los datos han sido cargados podemos empezar la simulación con el POST `/solve_model`. En donde especificamos el tiempo límite para la simulación. Este POST abrirá un Thread ya que el tiempo de solución puede ser largo.



Para verificar la finalización de la simulación usamos el GET `/get_results`. Si esta se encuentra finalizada devolverá el id de los resultados y si no mostrará que la solución aún no existe.



Finalmente tenemos dos opciones, podemos ver los resultados en forma de JSON o ver la imagen de los resultados. Para esto se usa el GET `/get_results/<ID>` y `/get_image/<ID>` respectivamente.

GET



{{base_url}}/get_results/23

GET



{{base_url}}/get_image/33