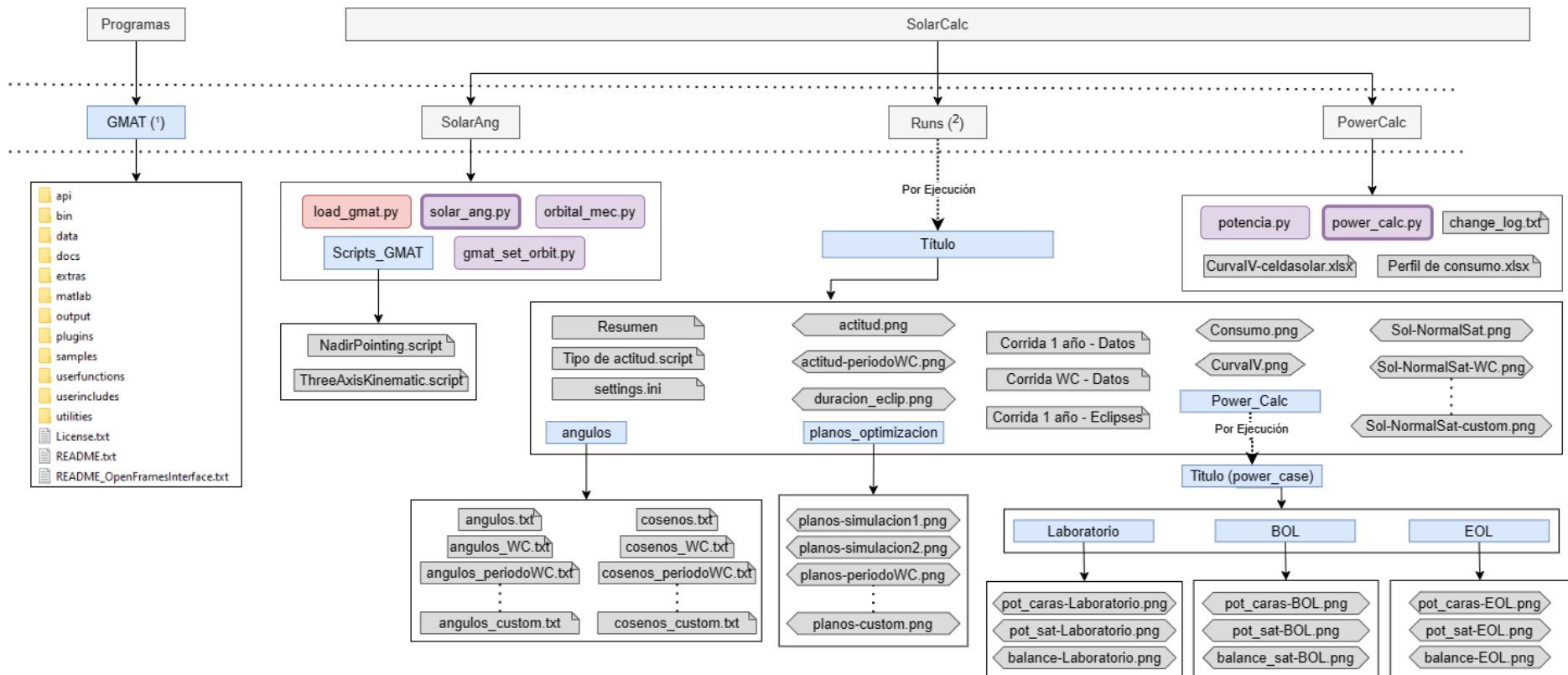


Árbol de contenido



Los ini siempre tienen que tener el formato `settings_v###.ini` donde `###` es el número de versión.

Input: setting.ini

PowerCalc:

- `calc_cons = True` - Calcular el consumo usando como input la distribución de celdas en los planos.
 - `distrib_type =`
 - N-max: distribuye la **cantidad de celdas** priorizando según el orden de las caras de mayor a menor **energía generada**
 - `cells [int]`: cantidad de celdas solares disponibles
 - `sup_disp1 [list[float]]`: lista de superficie disponible en cada plano
 - S-max: distribuye la **superficie de celdas** priorizando según el orden de las caras de mayor a menor **energía generada**
 - `cells [float]`: superficie total de celdas solares disponible
 - `sup_disp1 [list[float]]`: lista de superficie disponible en cada plan
 - N-fix: distribuye la **cantidad de celdas por cara** según se indica en `cells`
 - `cells [list[int]]`: cantidad de celdas solares disponibles
 - S-fix: distribuye la **superficie de celdas por cara** según se indica en `cells`
 - `cells [list[float]]`: superficie total de celdas solares disponible
- `calc_cons = False` - Calcular la distribución de celdas en cada cara según el consumo ingresado.
 - `distrib_type =`
 - `sup_disp-max`: distribuye las celdas según el espacio disponible priorizando según el orden de las caras de mayor a menor **energía generada**
 - `sup_disp1 [list[float]]`: lista de superficie disponible en cada plano
 - `sup_disp-fix`: distribuye las celdas según el espacio disponible priorizando según los **index de planos** en la variable `caras`
 - `caras [list[int]]`: lista de index de las caras en el orden que quiero priorizar el llenado
 - `sup_disp1 [list[float]]`: lista de superficie disponible en cada plano
 - %: distribuye determinados **porcentajes de la energía consumida** en las caras según los **index de los planos** en la variable `caras`.
 - `caras [list[int]]`: lista de index de las caras en el orden que quiero priorizar el llenado
 - `porcentaje_celdas [list[float]]`: lista de porcentajes (la suma de todos los elementos debe ser 1 (`sum(porcentaje_celdas)=1`))

¹ Como hay planos calculados que maximizan el coseno para el caso anual, WC y el periodoWC, este último siempre será priorizado. Si se desea que estos planos (o cualquier plano) sea descartado, ingresar 0 en su posición de la variable `sup_disp`.

Leer reporte de potencia y energía y graficar:

Para obtener los datos de potencia o de energía de algún caso en particular hay que:

```
In: import pandas as
In: Potencia = pd.read_csv("PATH_data_potencia_csv", sep="\t") # Si
se ejecutó PowerCalc, no es necesario correr esta linea.
In: Potencia.keys() # Lista de los nombres de las columnas
```

```
Index(['Unnamed: 0', 'Lab-fechas', 'Lab-tiempo', 'Lab-Cara_+X-W',
      'Lab-Cara_+Y-W', 'Lab-Cara_+Z-W', 'Lab-Cara_-X-W', 'Lab-Cara_-Y-W',
      'Lab-Cara_-Z-W', 'Lab-Panel_1-W', 'Lab-Panel_2-W', 'Lab-Panel_3-W',
      'Lab-Panel_4-W', 'Lab-Total-W', 'BOL-fechas', 'BOL-tiempo',
      'BOL-Cara_+X-W', 'BOL-Cara_+Y-W', 'BOL-Cara_+Z-W', 'BOL-Cara_-X-W',
      'BOL-Cara_-Y-W', 'BOL-Cara_-Z-W', 'BOL-Panel_1-W', 'BOL-Panel_2-W',
      'BOL-Panel_3-W', 'BOL-Panel_4-W', 'BOL-Total-W', 'EOL-fechas',
      'EOL-tiempo', 'EOL-Cara_+X-W', 'EOL-Cara_+Y-W', 'EOL-Cara_+Z-W',
      'EOL-Cara_-X-W', 'EOL-Cara_-Y-W', 'EOL-Cara_-Z-W', 'EOL-Panel_1-W',
      'EOL-Panel_2-W', 'EOL-Panel_3-W', 'EOL-Panel_4-W', 'EOL-Total-W'],
      dtype='object')
```

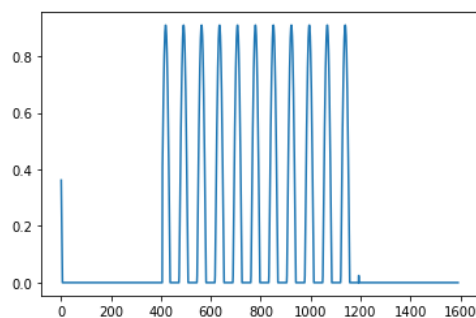
Out:

```
In: Potencia["Lab-Cara_+Z-W"]
```

```
0      0.362006
1      0.287860
2      0.211518
3      0.133565
4      0.054596
...
1587    0.000000
1588    0.000000
1589    0.000000
1590    0.000000
1591    0.000000
Name: Lab-Cara_+Z-W, Length: 1592, dtype: float64
```

Out:

```
In: plt.plot(Potencia["Lab-Cara_+Z-W"]) # Grafico: x: index 0,1,2,3... y: potencia
```



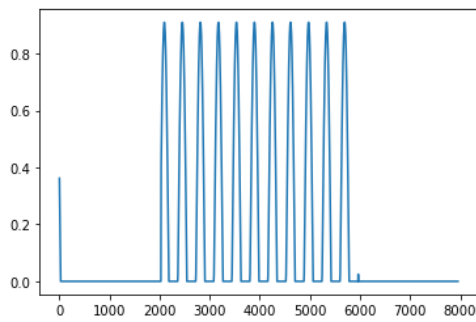
Out: [<matplotlib.lines.Line2D at 0x2490f715348>]

```
In: Potencia["Lab-tiempo"]
```

```
0      0.0
1      5.0
2     10.0
3     15.0
4     20.0
...
1587  7935.0
1588  7940.0
1589  7945.0
1590  7950.0
1591  7955.0
Name: Lab-tiempo, Length: 1592, dtype: float64
```

Out:

```
In: plt.plot(Potencia["Lab-fechas"], Potencia["Lab-Cara_+Z-W"])
```



Out: [matplotlib.lines.Line2D at 0x2490645f6c8>]

Generar un gráfico y reportes de cosenos/ángulos para un intervalo de tiempo deseado:

Se pueden generar gráficos y reportes de un intervalo de tiempo tomado de alguna de las simulaciones principales con el siguiente código, donde se definen los instantes de inicio y final del intervalo y luego se ejecuta la función `time_interval`.

```
# Modificar el inicio y el final del intervalo a gusto usando el módulo datetime
import datetime as dt
startWC -= dt.timedelta(minutes = 30)
endWC -= dt.timedelta(minutes = 30)

# Ejecutar la función time_interval
t_orb4, cos4, dist_sun4, axes4, paneles4, n_max4, Z_max4, plano_max_s4,
name_plano_max4 = gso.time_interval(file, name, values, num_orbWC, stepWC, startWC,
endWC, axes3, paneles3, period = period, name_plano_max = name_plano_max3, eclips_dur =
max(duration), case_in = "WC", case_out = "custom" )
```

- Case_in: simulación (reporte de cosenos) de la cual se quiere analizar un intervalo de tiempo
- Case_out: nombre del caso correspondiente al intervalo de tiempo analizado

Este código se puede ejecutar en las siguientes instancias:

- dentro de `solar-ang.py`, al final del script
- en la línea de comandos si todavía no se ejecutó `power_calc.py` ni otro caso de `solar_ang.py`
 - `file` debe ser el path absoluto de la carpeta de la corrida principal que está dentro de `Runs`
- Si se desea ejecutar después de `power_calc.py` hay que tomar las siguientes precauciones:
 - Comparar el path absoluto del archivo de cosenos del intervalo deseado con `file`

