

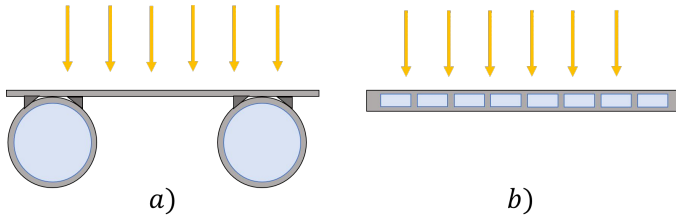
# Minichannel LFC

---

Implementación en Python de Minichannel Linear Fresnel Collector

[pcastilloquezada@gmail.com](mailto:pcastilloquezada@gmail.com)

# Colector de Minicanales

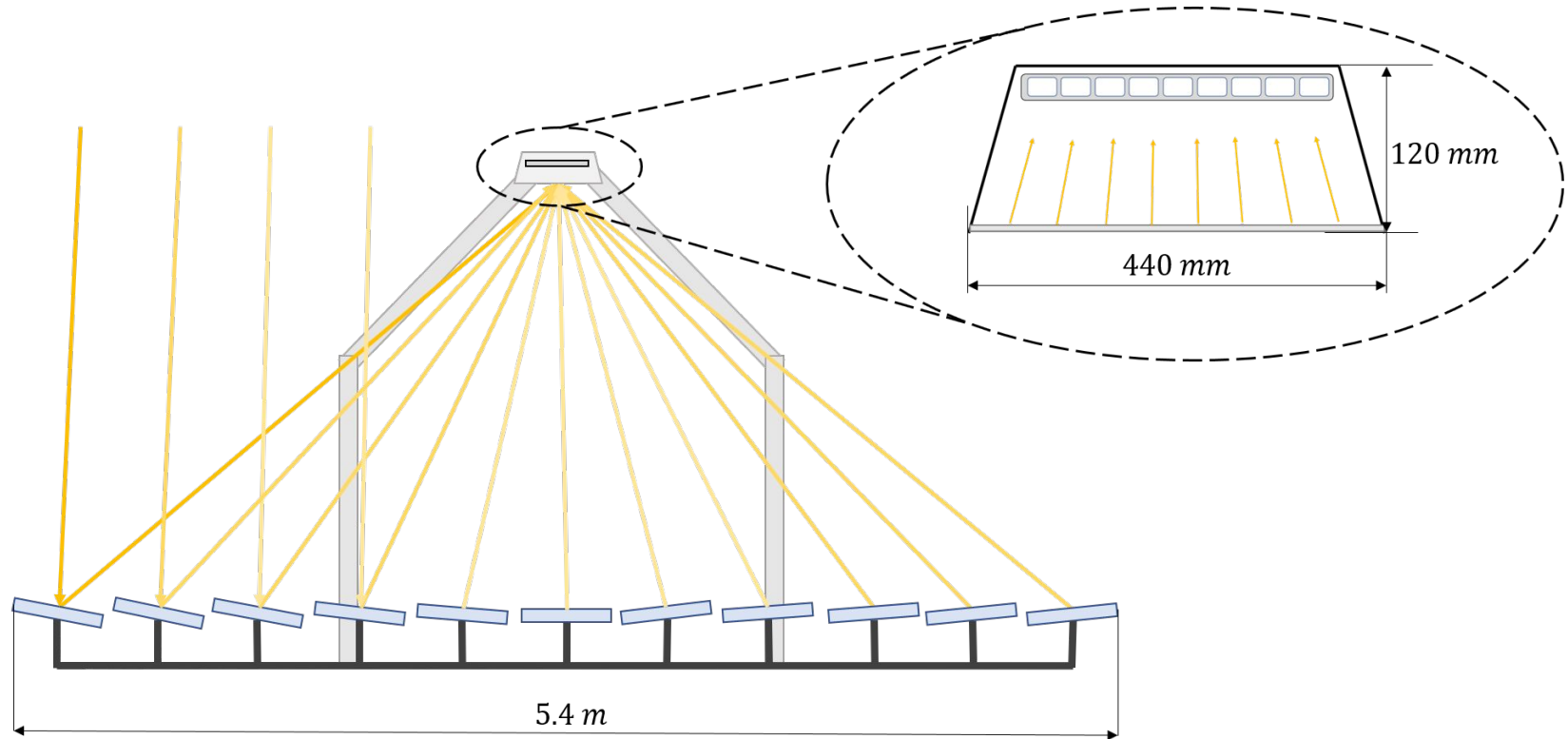


## Trabajos previos

1. **Diaz.** Performance model of a novel evacuated-tube solar collector based on minichannels. 2011.
2. **Robles, et al.** Aluminum minichannel solar water heater performance under year-round weather conditions. 2014.
3. **Hota, et al.** Two-phase flow performance prediction for minichannel solar collectors. 2019.

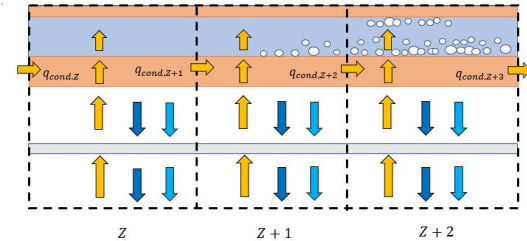
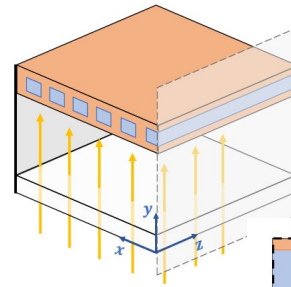
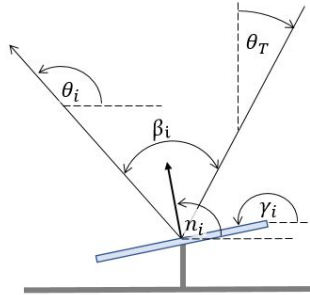
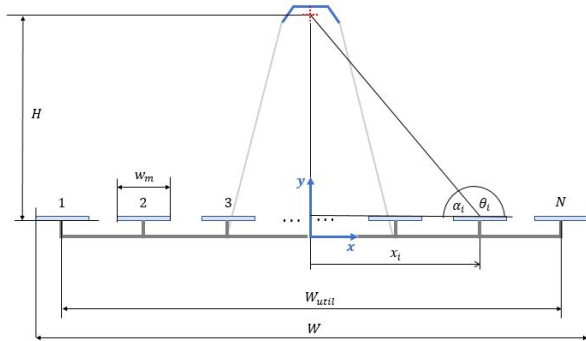
¿Qué pasaría con  
concentración?

# El diseño: Colector Fresnel + Minicanales



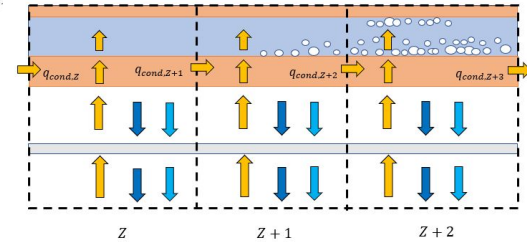
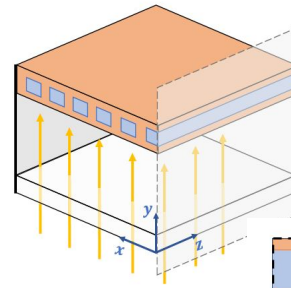
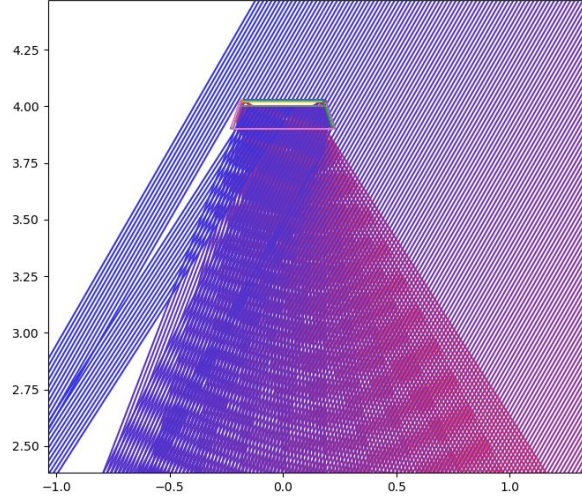
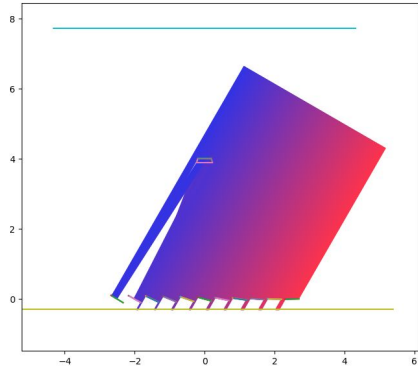
# Investigación actual

**Tesis:** Caracterización óptica y termodinámica de un colector solar Fresnel con absorbedor de minicanales para generación directa de vapor.



# Investigación actual

**Poster SWC:** Optical and thermal assessment of a linear Fresnel collector using a minichannel absorber tube for medium temperatures applications.



# Implementación en Python

- La idea: imitar los Types de TRNSYS
  - Se utiliza Python y su lenguaje orientado a objetos
  - *Class and Methods*
-

# Class & Methods

Una **clase** es más que una variable.

Una **clase** puede almacenar:

**variables:** `self.anchos = W`

**métodos:** `self.posicionEspejos()`

Puede invocar métodos dentro de métodos:

`self.construccion()`

Es bastante útil para trabajar con “objetos”  
físicos con muchas variables y funciones.

```
class MiniLFCollector:

    def __init__(self):
        pass

    def posicionEspejos(self, W, w_m,
                        N_m, a, L):

        self.anchos = W
        self.anchos_espejo = w_m
        ...

    def construccion(self, ...
                    self.posicionEspejos(W, w_m...
                    self.GeometriaReceptor(...
                    ....
```

---



# Cómo funciona?

Hay 3 comandos fundamentales en la clase

**MiniLFCollector:**

**colector\_1 = MiniLFCollector()**

*#Esto crea un instancia de la clase  
MiniLFCollector llamada **colector\_1**.*

**colector\_1.construccion(variables)**

*#Ejecuta los métodos (derecha),  
creando y almacenando los datos de  
diseño del colector.*

**colector\_1.simulacion(cond\_iniciales)**

*#Ejecuta la simulación de  
raytraicing y luego la térmica.*

```
class MiniLFCollector:
```

```
def construccion(self,...):  
    self.posicionEspejos(...)  
    self.GeometriaReceptor(...)  
    self.anguloRefleccion(...)  
    self.GeometriaAbsorbedor(...)  
    self.GeometriaMinicanal(...)  
    self.dict_LFC_bodies()  
  
def simulacion(self,...):  
    self.rotacionEspejos(...)  
    self.simulacionRaytraicing(...)  
    self.CondInicial(...)  
    . = self.simulacion_thermal(...)
```

Demo

**Cosas por hacer:**

La documentación

Dónde descargarlo?



*[github.com/pcastilloq/miniLFC](https://github.com/pcastilloq/miniLFC)*