

8__DESIGN__OF__A__SOLAR__COLLECTOR

ChatGPT para Justo Fuentes

2025-09-29

Contents

Ejercicio: Diseño de un colector solar plano	1
Preguntas a resolver	1
Checklist para desarrollar la solución en GitHub Codespaces	2
Explicación técnica (Markdown)	2
requirements.txt sugerido	5
Checklist solicitado	5

A continuación se detalla el **ejercicio propuesto en español** basado en el ejemplo resuelto del PDF *DESIGN OF A SOLAR COLLECTOR*, con énfasis en el uso de los números de **Reynolds (Re)** y **Nusselt (Nu)**. Incluye también un **checklist** breve para guiar el flujo de trabajo en **GitHub Codespaces**.

Ejercicio: Diseño de un colector solar plano

Se requiere diseñar un colector solar plano capaz de proporcionar un caudal de aire de **1 m³/s** a una temperatura de **40 °C**, en condiciones climáticas similares a las descritas:

- Temperatura ambiente promedio: **30 °C**.
- Latitud: **10° N**.
- Radiación media diaria en la estación de secado: **20 MJ/m² · día**.
- Densidad del aire: **1.28 kg/m³**.
- Conductividad térmica del aire: **k = 0.025 W/m · K**.
- Viscosidad dinámica del aire: **= 1.8 × 10⁻⁴ kg/m · s**.

Se selecciona un colector con dimensiones iniciales de **8.5 m × 4 m × 0.05 m** (longitud × ancho × profundidad).

Preguntas a resolver

1. Calcule el **diámetro hidráulico** del ducto del colector.
 2. Determine la **velocidad media del aire** en el ducto.
 3. Calcule el **número de Reynolds (Re)**. ¿El flujo es laminar o turbulento?
 4. Estime el **número de Nusselt (Nu)** usando la correlación: $[Nu = 0.02, Re^{0.8}]$
 5. Calcule el **coeficiente de transferencia de calor convectivo (h)**.
 6. Analice si el valor obtenido de (h) es consistente con el valor de referencia dado (**22 W/m² · K**).
-

Checklist para desarrollar la solución en GitHub Codespaces

1. Creación del repositorio

- Nombre sugerido: `solar_collector_design`
- Incluir `.gitignore` para Python.

2. Iniciar GitHub Codespaces

- Abrir el repositorio en un nuevo Codespace.

3. Instalar extensiones

- Python, Pylance, Streamlit.

4. Creación del entorno virtual

```
python -m venv .venv
source .venv/bin/activate    # Linux/Mac
.venv\Scripts\activate      # Windows
```

5. Archivos con ayuda del chatbot en Codespaces

- `README.md`: descripción del ejercicio y pasos de resolución.
- `requirements.txt`: incluir `numpy`, `pandas`, `streamlit`, `matplotlib`.

6. Instalación de módulos

```
pip install -r requirements.txt
```

7. Desplegar en Streamlit

- Crear `app.py` con entradas para dimensiones, propiedades del aire y correlaciones.
- Mostrar resultados de `Re`, `Nu` y `h`.
- Ejecutar:

```
streamlit run app.py
```

8. Alojamiento en Streamlit Community Cloud

- Conectar el repositorio de GitHub.
- Seleccionar rama `main` y archivo de inicio `app.py`.
- Publicar y compartir el enlace.

¿Quieres que prepare también un **modelo de `app.py` en Python (Streamlit)** para que los estudiantes introduzcan parámetros, calculen automáticamente `Re`, `Nu` y `h`, y comparen con los valores de referencia?

Aquí tienes un **modelo de `app.py` con Streamlit** listo para usar en **GitHub Codespaces**. Incluye explicación en Markdown, entradas editables, cálculos paso a paso de **`Re`**, **`Nu`** y **`h`**, y verificación con el valor de referencia **$22 \text{ W/m}^2 \cdot \text{K}$** del ejemplo.

Explicación técnica (Markdown)

La app calcula parámetros de diseño internos de un colector solar plano con ducto rectangular:

- **Geometría y caudal** Área de flujo ($A = b \cdot a$) con ($b =$) anchura ($a =$) profundidad. Velocidad media ($V = Q/A$).

- **Diámetro hidráulico** ($D_h = \frac{2ab}{a+b}$).
- **Régimen de flujo** ($Re = \frac{\rho, V, D_h}{\mu}$). Criterios: laminar ($Re < 2300$), transición ($2300 \leq Re \leq 4000$), turbulento (> 4000).
- **Transferencia de calor** Correlación del ejemplo (turbulento interno, diseño preliminar): ($Nu = 0.02, Re^{0.8}$). Luego ($h = \frac{Nu, k}{D_h}$).

Valores por defecto replican el caso del apéndice: ($Q = 1 \text{ m}^3/\text{s}$), ($L = 8.5 \text{ m}$), ($b = 4.0 \text{ m}$), ($a = 0.05 \text{ m}$), ($\rho = 1.28 \text{ kg}/\text{m}^3$), ($\mu = 1.8 \times 10^{-5} \text{ kg}/(\text{m} \cdot \text{s})$), ($k = 0.025 \text{ W}/(\text{m} \cdot \text{K})$). Se espera ($Re \approx 3.52 \times 10^4$), ($Nu \approx 86.7$), ($h \approx 21.9 \text{ W}/\text{m}^2\text{K}$).

```
# app.py
# Diseño interno de colector solar: Re, Nu y h para ducto rectangular
# Listo para GitHub Codespaces + Streamlit
# Autor: plantilla docente
# Requisitos: streamlit, numpy

import math
import numpy as np
import streamlit as st

st.set_page_config(page_title="Colector solar: Re, Nu, h", layout="wide")
st.title("Diseño de colector solar plano: Reynolds, Nusselt y coeficiente h")

with st.expander("Instrucciones y modelo físico", expanded=True):
    st.markdown(
        """
        **Qué hace esta app**

        1) Calcula el **diámetro hidráulico**:  $(D_h = \frac{2ab}{a+b})$ 
        2) Calcula la **velocidad media**:  $(V = Q/A)$ , con  $(A = b \cdot a)$ 
        3) Calcula **Reynolds**:  $(Re = \frac{\rho V D_h}{\mu})$ 
        4) Usa la correlación del ejemplo:  $(Nu = 0.02 Re^{0.8})$ 
        5) Obtiene **h**:  $(h = \frac{Nu k}{D_h})$ 

        **Criterios de régimen**
        - Laminar:  $Re < 2300$ 
        - Transición:  $2300 \leq Re \leq 4000$ 
        - Turbulento:  $Re > 4000$ 

        > Esta correlación es una **aproximación didáctica** apropiada para el ejemplo del apéndice. Para diseño
        """
    )

# -----
# Entradas
# -----
st.subheader("Parámetros de entrada")

col1, col2, col3 = st.columns(3)
```

```

with col1:
    Q = st.number_input("Caudal volumétrico Q [m³/s]", 0.001, 10.0, 1.0, step=0.01, format="%.3f")
    L = st.number_input("Longitud del ducto L [m]", 0.1, 100.0, 8.5, step=0.1)
    b = st.number_input("Ancho interno b [m]", 0.01, 20.0, 4.0, step=0.1)

with col2:
    a = st.number_input("Profundidad interna a [m]", 0.005, 1.0, 0.05, step=0.005, format="%.3f")
    rho = st.number_input("Densidad del aire [kg/m³]", 0.5, 2.0, 1.28, step=0.01, format="%.2f")
    mu = st.number_input("Viscosidad dinámica [kg/(m·s)]", 1e-6, 1e-3, 1.8e-5, step=1e-6, format="%.6f")

with col3:
    k = st.number_input("Conductividad térmica k [W/(m·K)]", 0.010, 0.100, 0.025, step=0.001, format="%.3f")
    T_in = st.number_input("Temperatura de entrada aire [°C] (opcional)", -10.0, 80.0, 30.0, step=0.5)
    T_out = st.number_input("Temperatura objetivo aire [°C] (opcional)", -10.0, 100.0, 40.0, step=0.5)

# -----
# Cálculos
# -----
A = b * a # área de flujo
V = Q / A # velocidad media
Dh = 2.0 * a * b / (a + b) # diámetro hidráulico
Re = rho * V * Dh / mu # Reynolds

# Validación de dominio para Nu
Nu = 0.02 * (Re ** 0.8) if Re > 0 else np.nan
h = (Nu * k) / Dh if Dh > 0 else np.nan

# Clasificación del régimen
if Re < 2300:
    regimen = "Laminar"
elif Re <= 4000:
    regimen = "Transición"
else:
    regimen = "Turbulento"

# Referencia del apéndice (~22 W/m2K)
h_ref = 22.0
dif_h = h - h_ref if np.isfinite(h) else np.nan

# -----
# Resultados
# -----
st.subheader("Resultados")

c1, c2, c3, c4 = st.columns(4)
c1.metric("Área de flujo A [m²]", f"{A:.4f}")
c2.metric("Velocidad V [m/s]", f"{V:.3f}")
c3.metric("D_h [m]", f"{Dh:.4f}")
c4.metric("Reynolds Re [-]", f"{Re:,.0f}")

c5, c6, c7 = st.columns(3)
c5.metric("Nusselt Nu [-]", f"{Nu:.1f}")
c6.metric("h [W/m²·K]", f"{h:.1f}")

```

```

c7.metric("Régimen", regimen)

st.info(
    f"Comparación con referencia del ejemplo: h_ref {h_ref:.1f} W/m²·K; "
    f"Δh = {dif_h:+.1f} W/m²·K." if np.isfinite(dif_h) else
    "Sin comparación disponible."
)

# -----
# Sensibilidad rápida
# -----
with st.expander("Sensibilidad rápida (opcional)"):
    st.markdown(
        """
Ajuste a (profundidad) y b (ancho) para estudiar efectos geométricos:
- A mayor a o b crece el área A, baja V y tiende a bajar Re y h.
- Reducir a incrementa V y Re, pero puede aumentar pérdidas de carga.
        """
    )

st.caption("Plantilla educativa. Verifique correlaciones específicas si el diseño final exige mayor pre")

```

requirements.txt sugerido

```

streamlit>=1.36
numpy>=1.26

```

Checklist solicitado

1. **Repositorio:** cree solar_collector_design y añada .gitignore para Python.
2. **Codespaces:** abra un Codespace desde el repo.
3. **Extensiones:** instale Python, Pylance y Streamlit.
4. **Entorno:**

```

python -m venv .venv
source .venv/bin/activate    # Linux/Mac
.venv\Scripts\activate      # Windows

```

5. **Archivos con el chatbot:** redacte README.md con objetivos, ecuaciones y pasos; genere requirements.txt.

6. **Instalación:**

```

pip install -r requirements.txt

```

7. **Despliegue local:**

```

streamlit run app.py

```

8. **Streamlit Community Cloud:** conecte el repo, seleccione rama main, archivo app.py, publique y comparta el enlace.