

1. **Entrada de datos (GUI)**

Se leen: **Re**, **D**, **k**, **error** (tolerancia). En el diagrama también aparecen **L** y **Q** y **g**, pero en este flujo sólo se usan para contexto; el cálculo mostrado es del **factor de fricción f**.

2. **Chequeo de régimen**

Si **Re** < **2300** \Rightarrow flujo laminar.

3. **Caso laminar**

Se calcula **f** = **64/Re** y se marca el método “**Laminar flow**”. Fin.

4. **Si no es laminar**

Se calcula la **rugosidad relativa**: **rel_roughness** = **k/D**.

5. **Umbral para “fully rough”**

Si **rel_roughness** > **0**, se fija un límite **Re_limit** = **3500 / (k/D)** (si **k=0**, el límite es ∞). Este umbral decide si se está en transición o ya en “completamente rugoso”.

6. **¿Tubería lisa? (k = 0)**

Si **rel_roughness** == **0** se aplica el modelo de **turbulento liso (Prandtl–Kármán / Von Karman–Prandtl, VK)** mediante **iteración**.

7. **Inicialización para iteración (liso)**

Se arranca con un **f** inicial y se itera con

$$f_{\text{new}} = [2 \log_{10}(Re \sqrt{f}) - 0.8]^{-2}$$

hasta que el **error relativo** $er = |(f_{\text{new}} - f)/f_{\text{new}}|$ sea \leq **error**.

8. **¿Transición rugosa?**

Si **k/D** > **0** y **Re** < **Re_limit**, se usa **Colebrook–White** (transición). También es **iterativo**.

9. **Iteración (Colebrook–White)**

Con un **f** inicial, se actualiza:

$$f_{\text{new}} = [-2 \log_{10}((k/3.71D) + 2.51/(Re \sqrt{f}))]^{-2}$$

Se repite hasta cumplir la tolerancia **error**.

10. **¿Completamente rugoso?**

Si **Re** \geq **Re_limit**, se asume régimen **fully rough** y se usa la forma explícita de **Prandtl–Kármán (VK2)**:

$$f = [2 \log_{10}(D/k) + 1.14]^{-2} \text{ (sin iteración).}$$

11. **Salida**

Se devuelve **f** y una etiqueta **method** indicando qué correlación se usó:

- “Laminar flow”

- “Turbulent, smooth pipe (VK)”
- “Turbulent, transition (Colebrook–White)”
- “Turbulent, fully rough (VK2)”

12. Cierre del flujo

El diagrama muestra los rombos de decisión/loops para las iteraciones y la condición “**er** > **error?**” que controla cuándo detenerse.