**Documento del modelo (versión simple 0-D)**

**0) Alcance y supuestos**

* **Sistema bien agitado (0-D)**: una sola concentración promedio c(t) en toda la cabina.
* **Isotermo**; **ρ** (aire) **constante**.
* Flujo 2D (por unidad de profundidad).
* El **ventilador acelera la velocidad** desde U0 hasta Utarget con una dinámica de primer orden.
* La **pintura** se inyecta como un **término fuente** que puede encenderse/apagarse en una ventana temporal.

**1) Variables y parámetros**

* Lx [m], Ly [m]
* V=LxLy [m³] (por unidad de profundidad)
* U(t) [m/s], Q(t) [m²/s] (caudal 2D)
* ρ [kg/m³]
* c(t) [–] (fracción másica de pintura en el gas)
* cin [–] (aire de entrada, típico 0)
* m˙ [kg/s] (2D) tasa de inyección de pintura
* ton,toff [s] (ventana del spray)
* U0,Utarget [m/s], τs [s] (parámetros del ventilador)

**2) Ecuaciones del modelo (numeradas)**

**(1) Volumen efectivo 2D**

 V=Lx Ly

**Por qué:** define el “tamaño” del mezclador 0-D.  
**Dónde en el código:** sim\_logic.py → simulate() al inicio.

**(2) Relación caudal–velocidad**

 Q(t)=U(t) Ly

**Por qué:** en 2D, el caudal por unidad de profundidad es velocidad por altura.  
**Dónde:** simulate() en cada paso (antes de integrar c).

**(3) Dinámica del ventilador (1er orden)**

 dUdt=Utarget−Uτs

**Por qué:** el ventilador no salta instantáneamente; modelamos su **respuesta** con un lag simple.  
**Dónde:** simulate() dentro del bucle temporal (bloque “fan”).

**(4) Spray ON/OFF (compuerta temporal)**

 g(t)={1,ton≤t<toff0,en otro caso (si no se define toff, g=1 para t≥ton) m˙(t)=m˙  g(t)

**Por qué:** permitir que la concentración **suba** mientras rociamos y luego **baje** cuando apagamos.  
**Dónde:** gate\_spray() y uso en simulate() al calcular m\_src.

**(5) Balance de masa de especie (forma conservativa 0-D)**

 ddt(ρVc)=ρ Q(t)(cin−c)+m˙(t)

**Por qué:** (Entrada limpia) ρQcin, (Salida) ρQc, (Fuente) m˙.  
**Dónde:** base para la ODE que integramos en simulate().

**(6) Ecuación diferencial para c(t)**

 dcdt=Q(t)V (cin−c)+m˙(t)ρ V

**Por qué:** despeje de (5) con ρ,V constantes.  
**Dónde:** simulate() calcula dc\_dt con Q(t) del (2) y m˙(t) del (4).

**(7) Condiciones iniciales**

 c(0)=0,U(0)=U0

**Por qué:** aire limpio al inicio y velocidad inicial deseada.  
**Dónde:** simulate() antes del bucle.

**(8) Integración temporal explícita (Euler hacia adelante)**

 Un+1=Un+Δt  Utarget−Unτs  cn+1=cn+Δt[QnV(cin−cn)+m˙nρ V]

con Qn=UnLy.  
**Por qué:** método simple y estable para este sistema 0-D con Δt pequeño.  
**Dónde:** simulate() dentro del bucle temporal.

**Proyección física:** tras actualizar c, se limita a [0,1].  
**Dónde:** simulate() (“límites físicos”).

**3) Qué se grafica y cómo sale del modelo**

* **Gráfica 1 — Concentración en el tiempo**: t vs c(t) usando (8).  
  *Código:* run\_sim.py → G1\_concentracion\_vs\_t.png.  
  *Interpretación:*  
  css=  cin+m˙ρQ si Q y m˙ son constantes;  
  si el spray se apaga (m˙=0), c(t) **decay** ∼e−(Q/V)t.
* **Gráfica 2 — Velocidad del fluido en el tiempo**: t vs U(t) integrando (3).  
  *Código:* run\_sim.py → G2\_velocidad\_vs\_t.png.  
  *Interpretación:* rampa suave desde U0 hacia Utarget con constante de tiempo τs.

**4) Dónde está cada pieza en el código (resumen)**

* **Parámetros**: params.json (incluye U0, Utarget, τs, ton, toff).
* **Cálculos base**: simulate()
  + (1) V=LxLy
  + (7) inicialización c(0),U(0)
  + (3) actualización U y (2) Q=ULy en cada paso
  + (4) gate\_spray() para m˙(t)
  + (6)/(8) integración explícita de c(t)
  + almacenamiento de series t,c,U
* **Gráficas**: run\_sim.py (solo dos salidas: c(t) y U(t)).

**5) Notas prácticas (rápidas)**

* **Velocidad que “empieza en 0.2 y aumenta”**: fija fan.U0 = 0.2, fan.U\_target al plateau deseado, fan.tau\_ssegún la rapidez de estabilización.
* **“Sube y luego baja” la concentración**: usa spray.t\_off para cortar la fuente; la caída es más rápida con Q mayor (constante de tiempo V/Q).
* **Unidades**: L en m, V en m³ (2D), U en m/s, Q en m²/s, ρ en kg/m³, m˙ en kg/s, t en s.