

Universidad del Valle de Guatemala

Facultad de ingeniería

Computación Paralela y Distribuida

Catedrático: Luis García



Proyecto 1

Nelson Eduardo García Bravatti 22434

Gabriel Paz 221087

Joaquín Puente 22296

Guatemala, 5 de septiembre de 2025

Introducción:

Se presenta el desarrollo de un screensaver de ‘lluvia’ que simula las ondas radiales producidas por el impacto de gotas sobre una superficie de agua. El modelo implementa un campo de alturas $H(x,y,t)$ generado por la superposición lineal de contribuciones individuales de cada gota. La versión actual es secuencial y está estructurada para habilitar una paralelización posterior con OpenMP. El usuario especifica únicamente el tamaño de la ventana y el número de gotas N ; el sistema muestra los FPS para asegurar una experiencia fluida (≥ 30 FPS).

Antecedentes:

OpenMP es una API estándar para programación paralela en memoria compartida en C/C++ y Fortran, ampliamente soportada en sistemas Windows, macOS y Linux. En el ámbito físico, el principio de superposición establece que, cuando varias ondas coinciden, la perturbación total es la suma de sus perturbaciones individuales, produciendo interferencia constructiva o destructiva según su fase relativa. Estos conceptos fundamentan la versión paralela (por píxel o por ‘tiles’) del algoritmo.

Arquitectura General:

La implementación separa claramente la simulación (cálculo del campo H) del renderizado (SDL2 en el hilo principal):

- Captura y validación de argumentos (--width, --height, --N).
- Inicialización de SDL2: ventana, renderer y textura ARGB8888.
- Mundo (World): gestión de gotas activas, parámetros aleatorios y respawn por vida útil.
- Simulación: acumulación de $H(x,y)$ mediante la suma de contribuciones ‘ripple_contrib’ por gota.
- Sombreado: cálculo de normales por diferencias finitas y composición fotométrica (Lambert + Blinn, Fresnel, ambiente de cielo y absorción espectral) para un aspecto acuático realista.
- Presentación: copia del buffer a la textura SDL y presentación en pantalla, con FPS en el título.

Modelo Físico y Superposición:

Cada gota d con tiempo de impacto t_0 contribuye con un anillo principal centrado en $r \approx c \cdot (t - t_0)$, con envolvente gaussiana de ancho σ , amortiguamiento temporal e atenuación radial suave. Para mayor realismo se incluyen ripples capilares (gaussianas estrechas adyacentes a la cresta) y una salpicadura inicial breve con modulación angular. El height field $H(x,y,t)$ resulta de sumar las contribuciones de todas las gotas. La interferencia emerge de forma natural al superponer crestas y valles, reforzando o atenuando la amplitud en distintas regiones.

Paralelización Prevista con OpenMp:

La función de acumulación del campo H se presta a paralelización con OpenMP. Existen dos estrategias principales:

- 1) Paralelización por píxel: distribuir el doble bucle (y,x) y sumar las N gotas por píxel (sencillo, buen balance; costo computacional estable).
- 2) Paralelización por tiles/ROI: para cada gota, procesar únicamente la banda activa alrededor del anillo (culling), usando acumuladores por hilo y combinación final para evitar condiciones de carrera.

En ambos casos, SDL debe permanecer en el hilo principal; únicamente la simulación se paraleliza.

Entradas, Salidas y Parámetros:

Entradas: tamaño de ventana (`--width`, `--height`) y N gotas (`--N`). Opcionalmente se pueden exponer semilla, paletas y parámetros estéticos/numéricos en futuras versiones.

Salidas: ventana con render en tiempo real y título con FPS suavizados (~ 1 s).

Conclusiones y Recomendaciones:

La versión secuencial logra un aspecto de agua verosímil combinando un modelo de ondas simple pero efectivo y un sombreado basado en normales. Para cargas mayores (resoluciones altas o N grande), se recomienda paralelizar la acumulación del campo H con OpenMP, ya sea por píxel o por tiles con culling por anillo. Como trabajo futuro: exponer parámetros estéticos por CLI, añadir modo headless para perfilado, y evaluar técnicas SIMD y caches amistosas para maximizar el speedup y la eficiencia.

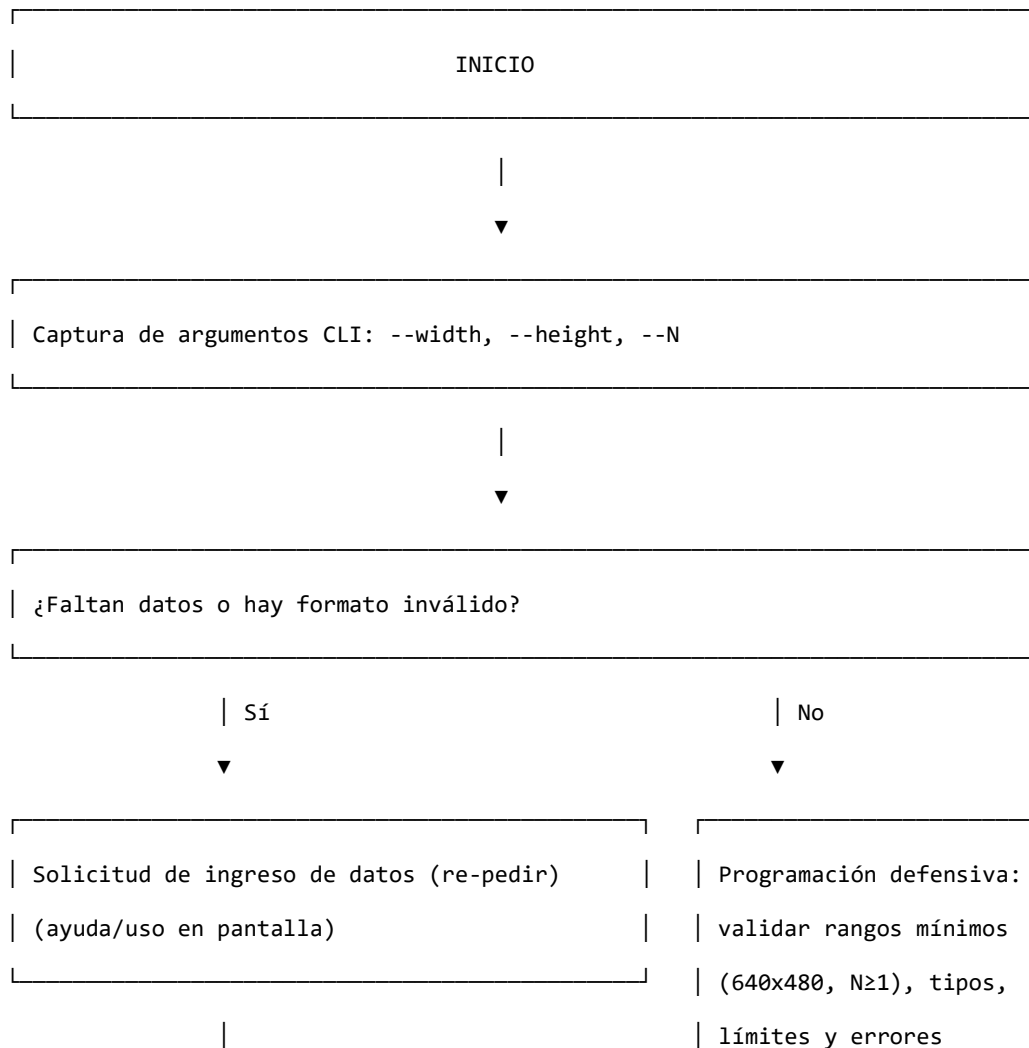
Bibliografía:

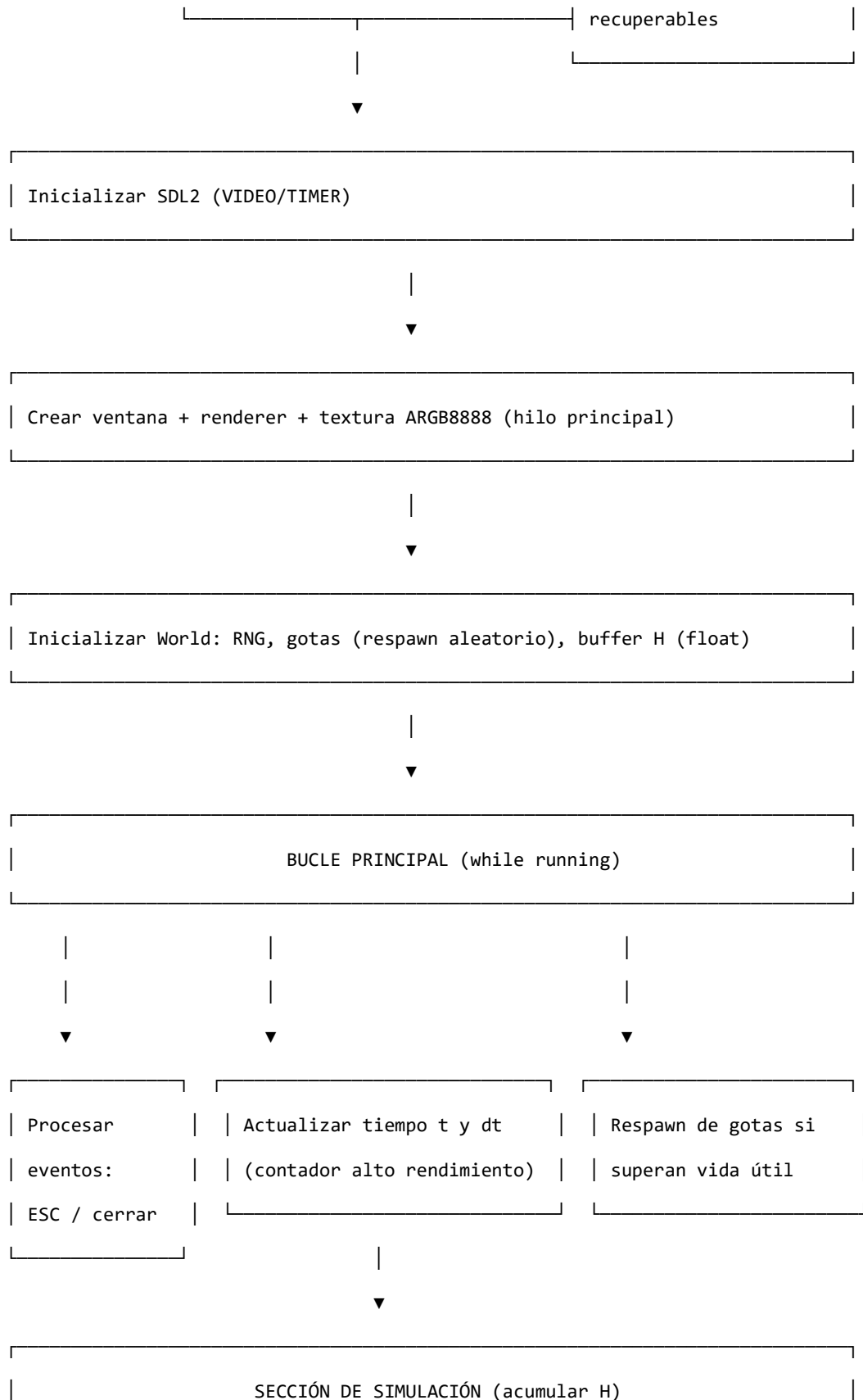
Urone, P. P., & Hinrichs, R. (2020, March 26). 13.3 Wave Interaction: Superposition and Interference - Physics | OpenStax. <https://openstax.org/books/physics/pages/13-3-wave-interaction-superposition-and-interference>

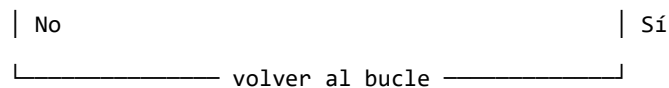
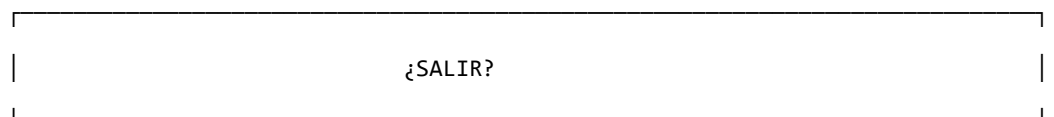
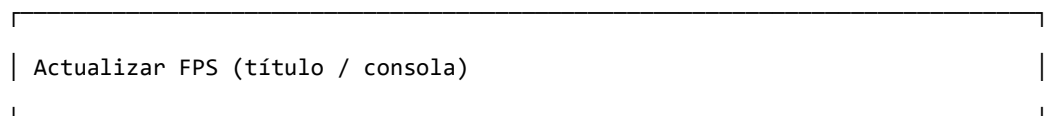
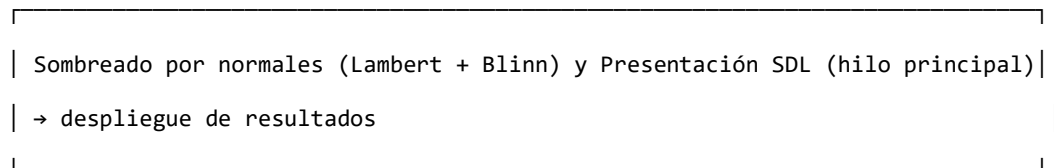
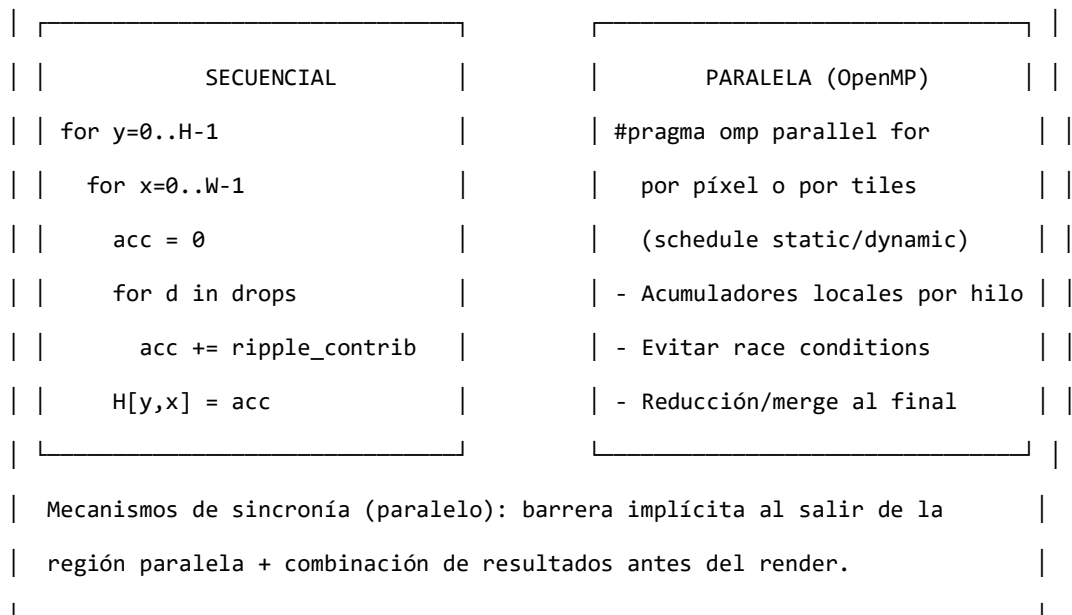
Tim.Lewis. (2025, June 27). Specifications - OpenMP. OpenMP. <https://www.openmp.org/specifications/>

SDL2/FrontPage. (n.d.). SDL2 Wiki. <https://wiki.libsdl.org/SDL2/FrontPage>

Anexo 1: Diagrama de Flujo del programa







Limpieza: destruir textura, renderer, ventana; SDL_Quit()

Anexo 2: Catálogo de Funciones

Función / Clase	Entradas (tipo)	Salidas (tipo)	Descripción
int main(int,char**)	--width:int, height:int, --N:int	-- int (código de salida)	Parseo y validación de argumentos, init SDL, loop principal, FPS y limpieza.
World (struct/clase)	AppConfig, interno	RNG Gestor de estado (drops, H)	Administra gotas activas, respawn aleatorio según vida útil y almacena el buffer H.
void respawn_drop(Drop& , float now)	Drop&, now:float	—	Reinicializa parámetros de una gota (posición, fuerza, vida, capilares, splash).
void maybe_respawn(float now)	now:float	—	Respawnea gotas cuya vida superó el umbral para mantener actividad visual continua.
float ripple_contrib(float x,float y,float t,const Drop& d)	x,y,t, d	float	Contribución de una gota en (x,y,t): cresta principal + capilares + splash con amortiguamiento.
void accumulate_heightfie ld_sequential(std::vec tor<float>& H, int W, int Hh, const std::vector<Drop>& drops, float t)	H, W, Hh, drops, t	H (modificado)	Acumula el campo de alturas sumando las contribuciones de todas las gotas (superposición lineal).

void shade_and_present(S DL_Renderer*, PixelBuffer&, const std::vector<float>& H, float slopeScale)	renderer*, textura, H, slopeScale	Frame en pantalla	Calcula normales por diferencias finitas y compone color (Lambert+Blinn, Fresnel, absorción).
--	--------------------------------------	-------------------	---

Anexo 3: Bitácora de pruebas

Figura 1: Pruebas Paralelas

suite	N	W	H	mode	p	run	mean_sim_ms	mean_shade_ms	mean_FPS
A	50	1024	768	omp	8	1	45.158631	24.131253	16.071429
A	50	1024	768	omp	8	2	42.297084	23.812169	17.000000
A	50	1024	768	omp	8	3	45.701387	25.056255	16.428571
A	50	1024	768	omp	8	4	50.586952	27.110185	13.642857
A	50	1024	768	omp	8	5	51.802915	28.405793	14.357143
A	50	1024	768	omp	8	6	49.500273	26.469670	14.857143
A	50	1024	768	omp	8	7	52.668549	28.802051	14.571429
A	50	1024	768	omp	8	8	49.765556	26.658614	14.785714
A	50	1024	768	omp	8	9	51.278932	28.081473	14.785714
A	50	1024	768	omp	8	10	49.953785	26.779126	14.714286
A	100	1024	768	omp	8	1	89.017226	28.896893	10.642857
A	100	1024	768	omp	8	2	87.295035	27.932099	10.714286
A	100	1024	768	omp	8	3	91.724728	29.977731	10.500000
A	100	1024	768	omp	8	4	89.828552	28.550630	9.928571
A	100	1024	768	omp	8	5	85.186089	27.093702	11.153846
A	100	1024	768	omp	8	6	88.011454	28.783327	10.571429
A	100	1024	768	omp	8	7	87.789958	27.770139	10.785714
A	100	1024	768	omp	8	8	86.528721	26.812097	10.857143
A	100	1024	768	omp	8	9	85.822714	27.783841	10.857143
A	100	1024	768	omp	8	10	86.030543	28.089503	10.923077
A	200	1024	768	omp	8	1	158.184893	29.098280	6.846154
A	200	1024	768	omp	8	2	154.643901	26.755972	7.230769
A	200	1024	768	omp	8	3	161.153147	27.832255	6.461538
A	200	1024	768	omp	8	4	156.349735	27.202332	7.076923
A	200	1024	768	omp	8	5	158.304559	27.981222	7.000000
A	200	1024	768	omp	8	6	155.159361	26.859643	7.307692
A	200	1024	768	omp	8	7	162.896223	30.168006	6.461538
A	200	1024	768	omp	8	8	161.144487	28.168326	6.846154
A	200	1024	768	omp	8	9	157.263752	27.872966	7.000000
A	200	1024	768	omp	8	10	154.463595	27.296912	7.076923

Figura 2: Pruebas Secuenciales

A	50	1024	768	seq	1	1	86.492224	66.498786	7.627740
A	50	1024	768	seq	1	2	86.541262	67.393759	7.524366
A	50	1024	768	seq	1	3	86.532643	66.603486	7.577659
A	50	1024	768	seq	1	4	86.931374	66.346395	7.531955
A	50	1024	768	seq	1	5	88.197593	67.532578	7.345106
A	50	1024	768	seq	1	6	91.277961	70.381849	7.159655
A	50	1024	768	seq	1	7	90.787601	69.315198	7.263841
A	50	1024	768	seq	1	8	90.920164	69.726529	7.219430
A	50	1024	768	seq	1	9	91.128103	69.656833	7.066441
A	50	1024	768	seq	1	10	92.014249	70.235648	7.171578
A	100	1024	768	seq	1	1	175.224390	72.531716	4.931446
A	100	1024	768	seq	1	2	176.121403	72.249444	4.970198
A	100	1024	768	seq	1	3	173.182811	70.755483	5.015833
A	100	1024	768	seq	1	4	183.554066	77.185955	4.805572
A	100	1024	768	seq	1	5	174.368394	71.189927	5.027013
A	100	1024	768	seq	1	6	174.120615	71.343808	4.994332
A	100	1024	768	seq	1	7	173.097490	70.763398	5.044579
A	100	1024	768	seq	1	8	174.666358	71.435581	5.012232
A	100	1024	768	seq	1	9	174.262492	71.311768	4.987492
A	100	1024	768	seq	1	10	176.274844	71.449571	4.931536
A	200	1024	768	seq	1	1	327.154915	72.461397	3.396915
A	200	1024	768	seq	1	2	331.390079	73.443747	3.205131
A	200	1024	768	seq	1	3	327.581349	71.976615	3.363463
A	200	1024	768	seq	1	4	325.813195	71.881226	3.397759
A	200	1024	768	seq	1	5	327.306872	71.700282	3.301906
A	200	1024	768	seq	1	6	329.552649	71.763882	3.369578
A	200	1024	768	seq	1	7	323.956949	71.471803	3.419730
A	200	1024	768	seq	1	8	325.186113	71.541421	3.395592
A	200	1024	768	seq	1	9	329.917615	71.984592	3.241343
A	200	1024	768	seq	1	10	328.582415	72.590018	3.192108

Cuadro 1: Resultados de pruebas

N	p	Tseq (ms)	Tpar (ms)	Speedup	Eficiencia	FPS_seq	FPS_par
50	8	89.082	48.871	1.823	0.228	7.349	15.121
100	8	175.487	87.724	2.000	0.250	4.972	10.693
200	8	327.644	157.956	2.074	0.259	3.328	6.931