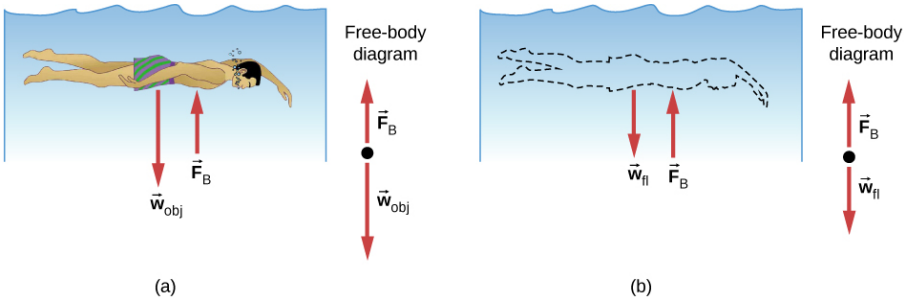
Ejercicios de Simulación Tema 2

La partícula flotante

- Volúmenes básicos.
- Estabilidad: Centro de gravedad/masas y de flotación.
- Fuerza de flotación: $F_B = \rho g V_s$, $\rho = densidad$, $V_s =$ Volumen sumergido.
- $V_{s-totalmente-sumergido} = 4\pi \cdot radius^3 \ / \ 3$.
- Parcialmente sumergido:
  - $h = location.y + radius - height \ / \ 2$.
  - $a = \sqrt{2h \cdot radius - h^2}$.
  - $V_s = \left(3 \cdot a^2 + h^2\right) \cdot \pi \ / \ 6$



Resolución

Se ha creado una clase partícula donde dentro de esta se aplican la fuerza a la partícula, donde tendremos en cuenta su radio ($\_r$) su masa ($\_m$) y su flotabilidad ($buoyancy$). También se ha creado una función para aplicar la gravedad $a = \dfrac{F_g}{m}$.

Y luego tenemos 3 casos concretos, cuando la partícula esta dentro del agua, cuando esta en la línea y cuando esta fuera:

- Cuando esté dentro del agua o en la línea se le aplicará la flotabilidad y una fuerza de rozamiento $F_{roz} = -vK$ siendo v la velocidad y K la fricción del agua. Estas se nos proporcionan en el enunciado.
- Cuando está fuera del agua no se le aplica ninguna fuerza salvo la de gravedad que se aplica en todo momento.

```
1    void calculateForces() {
2      float buoyancy;
3      PVector froz = new PVector(0, 0);
4      PVector vFb = new PVector(0, 0);
5      if (_position.y - _r >= water_height) { //dentro
6        buoyancy = 4.0 * PI * pow(_r, 3) / 3.0;
7        froz.y = - water_friction * _velocity.y;
8      } else if (_position.y + _r / 2 > water_height && _position.y - _r / 2 <
   water_height) { //en la línea
9        float h = _position.y + _r - height/1.5;
10       float a = sqrt(2 * h * _r - pow(h, 2));
11       buoyancy = (3 * pow(a,2) + pow(h, 2)) * PI * h / 6.0;
12       froz.y = -water_friction * _velocity.y;
13     } else { //fuera
14       buoyancy = 0;
15       froz.y = 0;
16     }
```

```
    vFb.y = - _d * G.y * buoyancy;
    accelerate(vFb);
    accelerate(froz);
  }
```

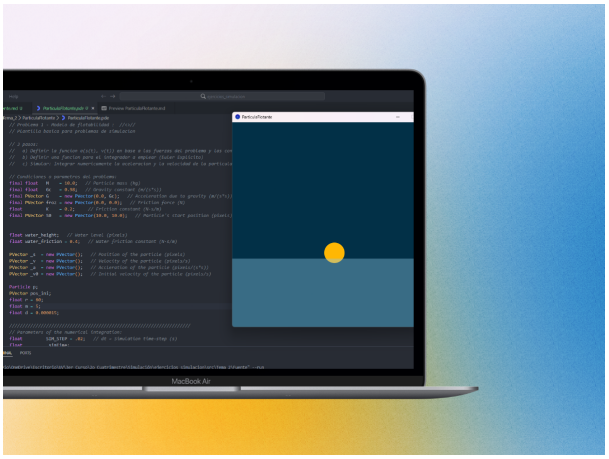


Figure 1: Video

La partícula splash

Resolución

A partir del ejercicio de la Partícula Flotante, he creado una clase Splash donde cuando por primera vez la partícula colisiona con el agua se generan 100 partículas de manera aleatoria sobre el eje y alejándose del centro de la partícula con un tiempo de vida.

```
// Generar gotas
for (int i = 0; i < 100; i++) {
      PVector drop = PVector.random2D();
      drop.y = abs(drop.y) * -2; // Asegura que la dirección es hacia arriba
      drop.mult(random(20, 50));
      drops.add(drop);
}
```

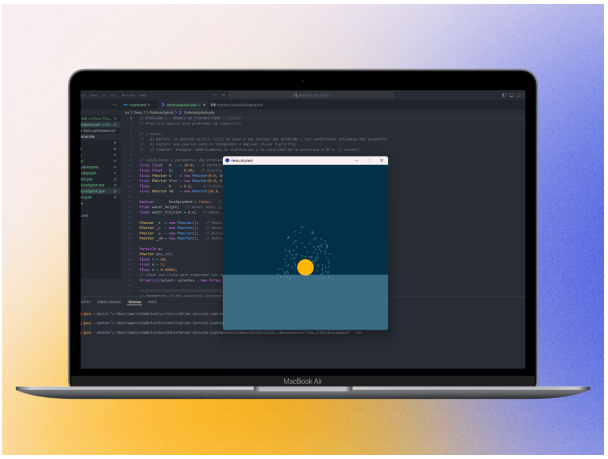
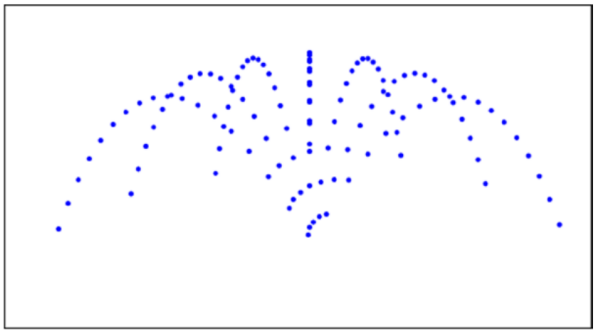


Figure 2: Video

Emisores de partículas. Fuente

Las partículas van apareciendo desde un punto: efectos de humo, explosiones, etc.

- Velocidad inicial: en función del escenario (ej. explosión, humo...).
- Fuerzas principales: Gravedad, viento ...
- Implementar el emisor (fuente) de la figura o similar:



Resolución

Se han creado unas partículas con un tiempo de vida y una masa m muy pequeña ya que se suponen que son gotas de agua emitidas por una fuente, se han generado con una aceleración y velocidad aleatorias. Además estas le he aplicado únicamente la fuerza de gravedad. Me he basado en las siguientes fórmulas:

$$F_g = m \cdot a \tag{1}$$

Se han puesto las velocidades y aceleración negativa por la referencia de ejes, por eso se ha puesto la gravedad positiva.

```
1  void update() {
2     PVector gravity_force = new PVector(0, gravity * mass); // Fuerza de gravedad
3     acceleration.add(gravity_force); // Agregar gravedad a la aceleración
4     velocity.add(acceleration);
5     position.add(velocity);
6     lifespan -= 2;
7  }
```

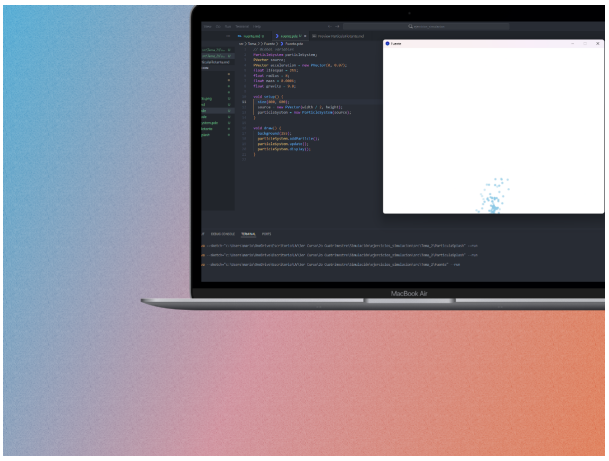


Figure 3: Video