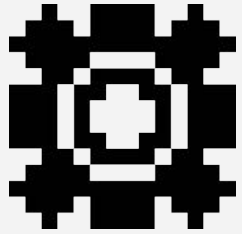


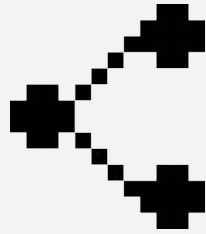
Desenvolvimento o de **jogos** com a Godot



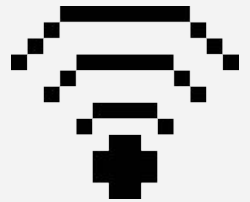
Resumo de conceitos



Nodos



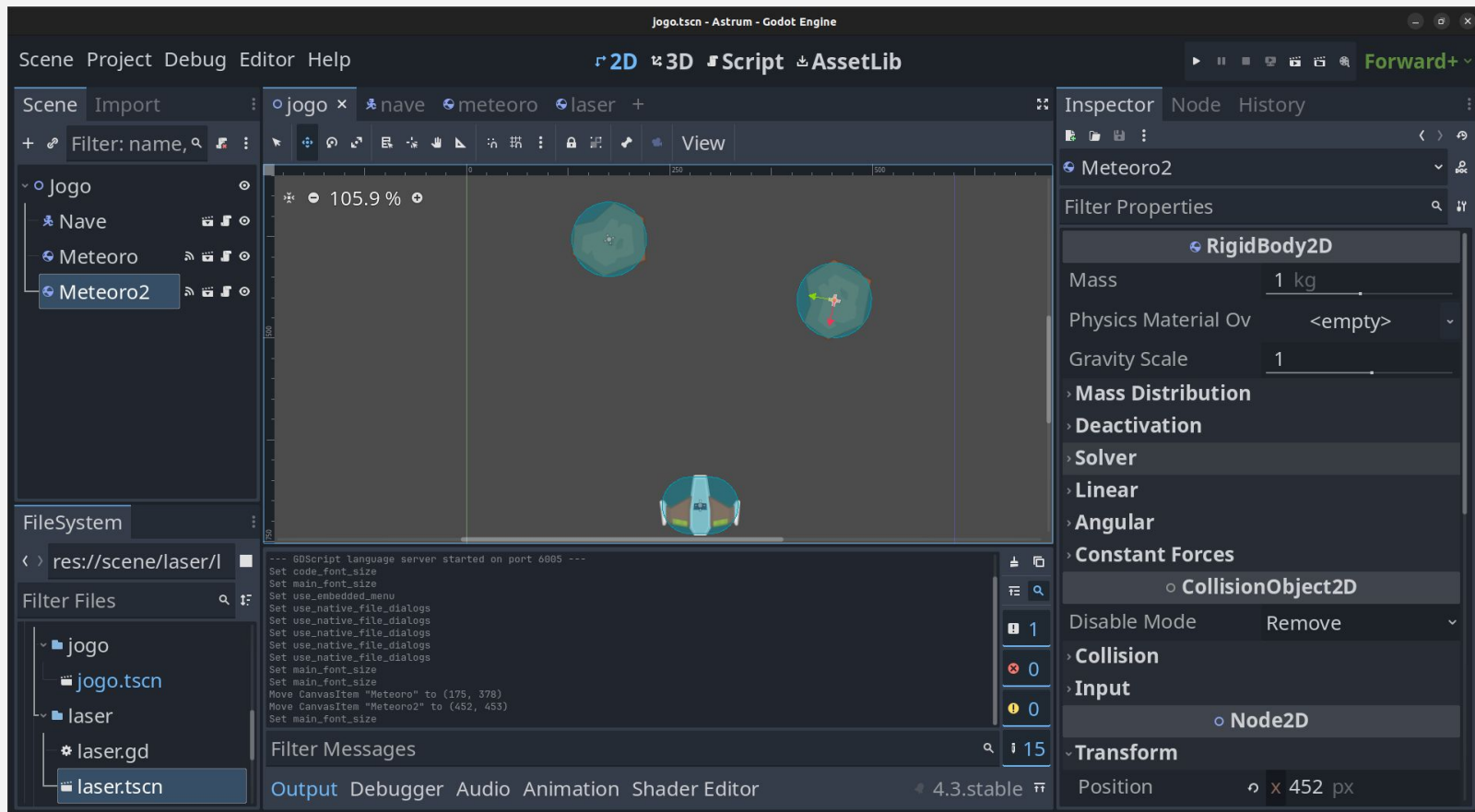
Cenas



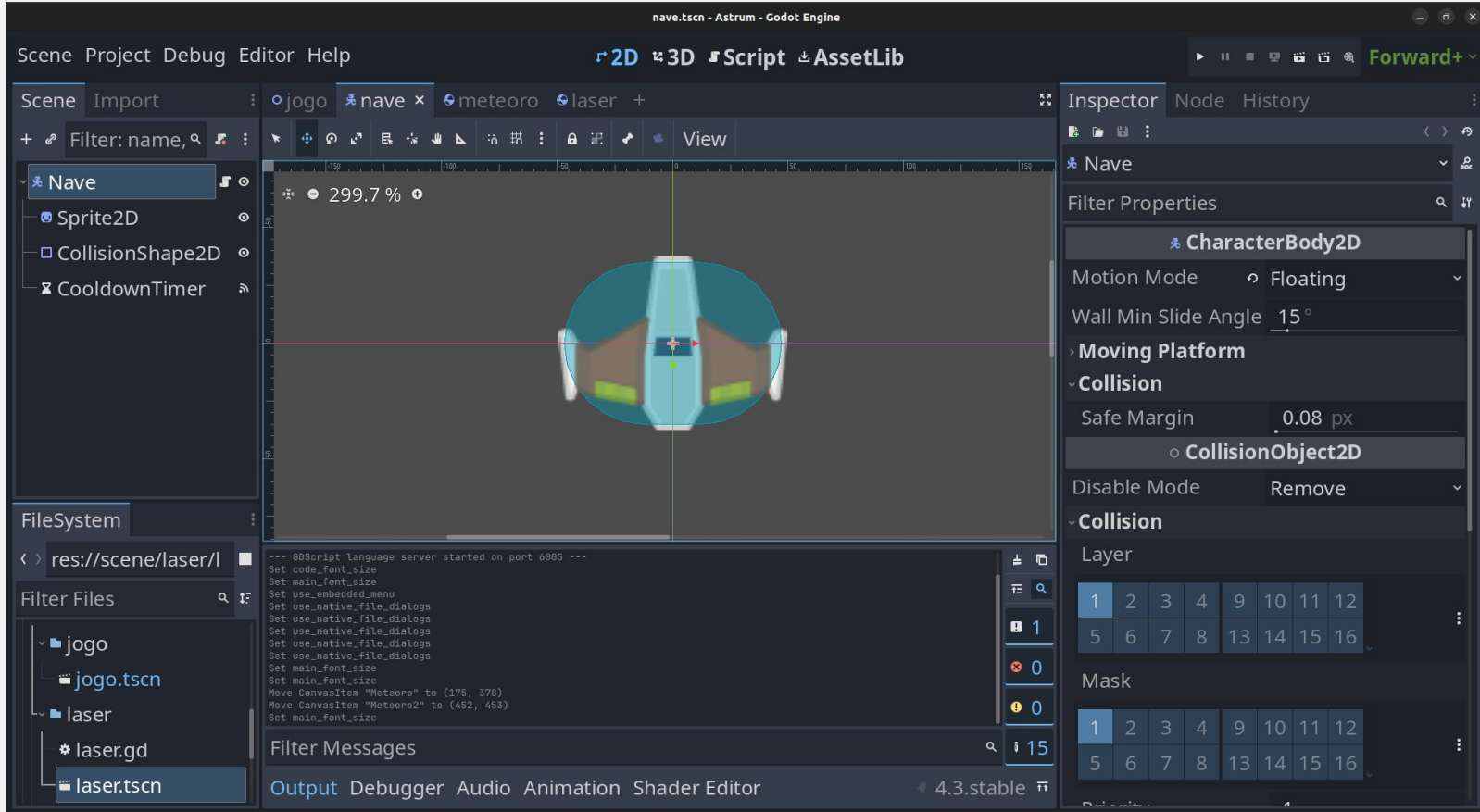
Sinais



jogo.tscn



nave.tscn



nave.gd



```
func _physics_process(delta: float) -> void:
    if Input.is_action_pressed("ui_accept"):
        atirar()

    var direction_x := Input.get_axis("ui_left", "ui_right")
    var direction_y := Input.get_axis("ui_up", "ui_down")
    var direction = Vector2(direction_x, direction_y)

    direction = direction.normalized()

    if direction:
        velocity = direction * SPEED
    else:
        velocity.x = move_toward(velocity.x, 0, SPEED)
        velocity.y = move_toward(velocity.y, 0, SPEED)

    move_and_slide()
```

nave.gd



```
var pode_atirar: bool = true
const LASER = preload("res://scene/laser/laser.tscn")

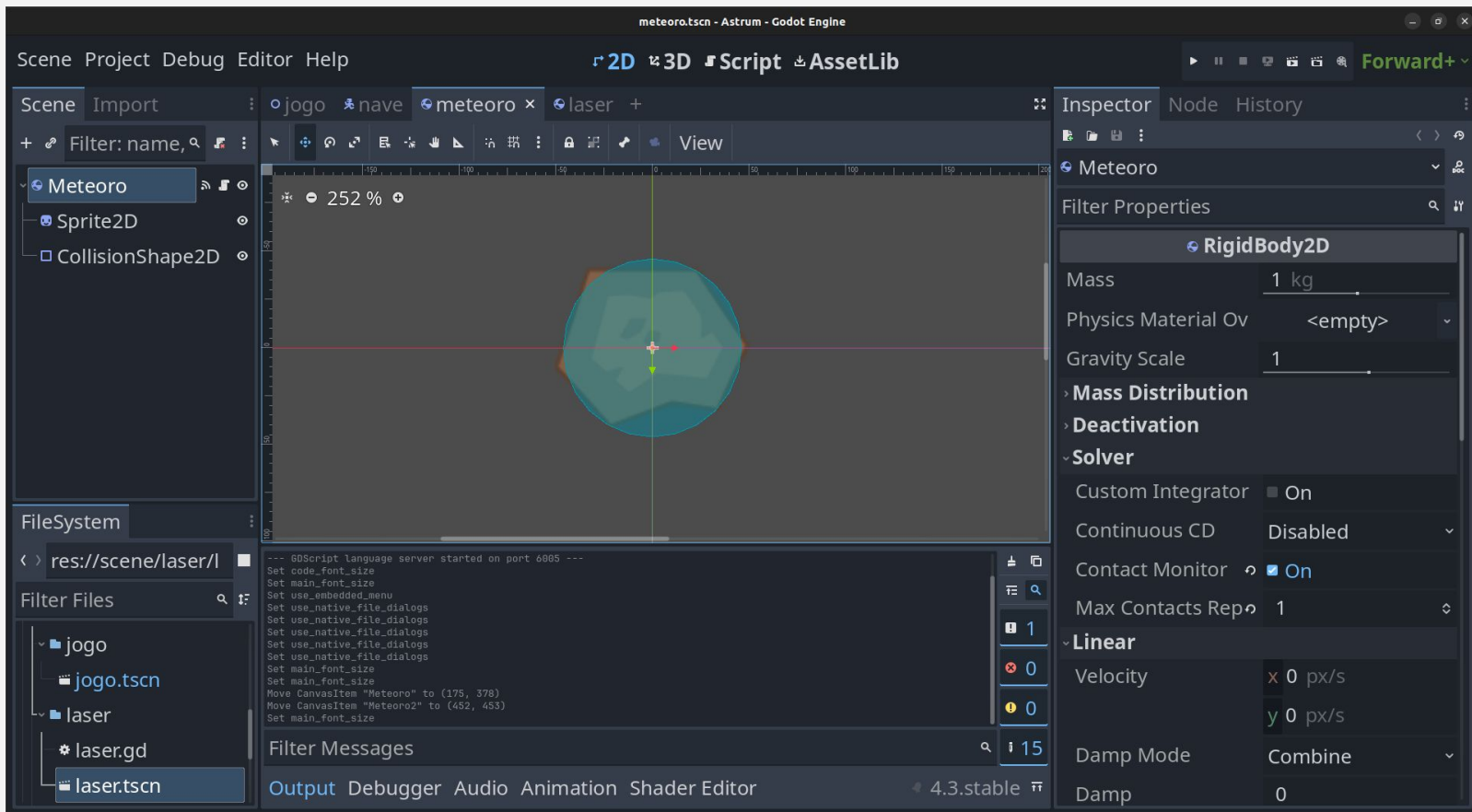
func atirar():
    if not pode_atirar:
        return

    pode_atirar = false
    print("ATIRANDO")
    var laser = LASER.instantiate()
    laser.global_position = self.global_position - Vector2(0, 40)
    add_sibling(laser)

    $CooldownTimer.start()

func _on_cooldown_timer_timeout() -> void:
    pode_atirar = true
```

meteoro.tscn



meteorito.gd

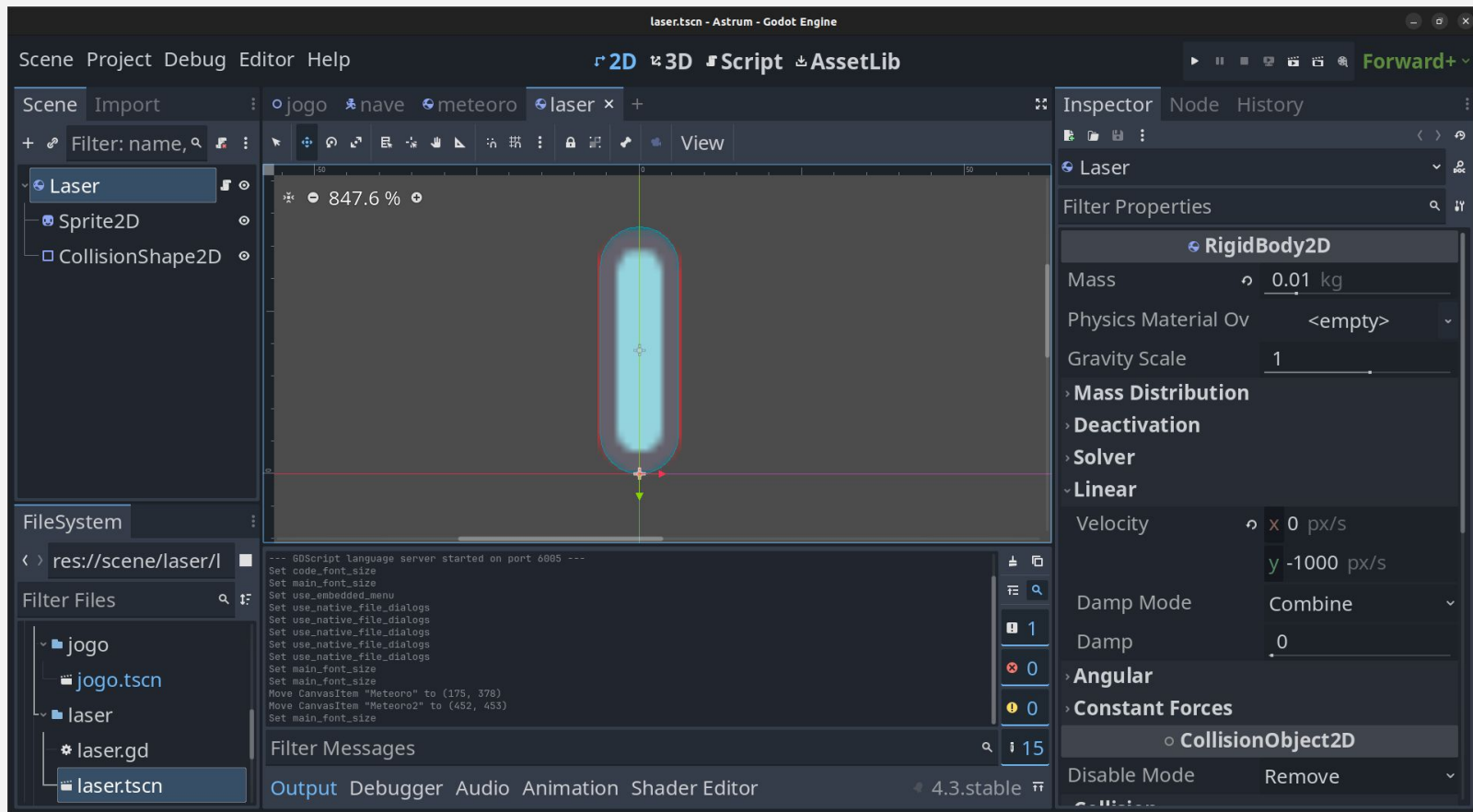


```
var vida: int = 3
const VELOCIDADE = 300

func _ready():
    linear_velocity = Vector2.RIGHT.rotated(global_rotation) * VELOCIDADE
    angular_velocity = randf_range(-15, 15)

func _on_body_entered(body: Node) -> void:
    if body is Laser:
        body.queue_free()
        vida -= 1
        if vida <= 0:
            self.queue_free()
    if body is Nave:
        body.queue_free()
```

laser.tscn



laser.gd



```
class_name Laser  
extends RigidBody2D
```



Dicas

- Participe de **Game Jams**
- Parcerias com artistas
 - Reduza o escopo
 - TERMINE

Recursos

- [Itch.io](https://itch.io) – Game jams e assets
- [Kenney.nl](https://kenney.nl) – Assets
- [Estúdio Recursivo](https://www.youtube.com/c/EstudioRecursivo) no YouTube

