

# SuperCollider

José Luis Caro Bozzino

Grado en Ingeniería Informática de Software

Sevilla, 2020

# Índice

- El lenguaje
- Formato de la memoria
- Experimentación
- Planificación temporal
- Conclusiones
- Bibliografía

# El lenguaje:

## Historia



James McCartney  
(1996)



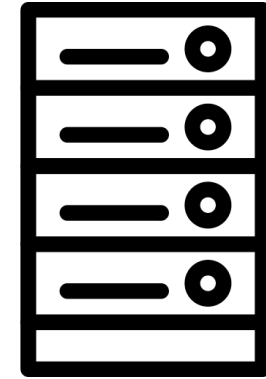
GNU  
(2002)

# El lenguaje:

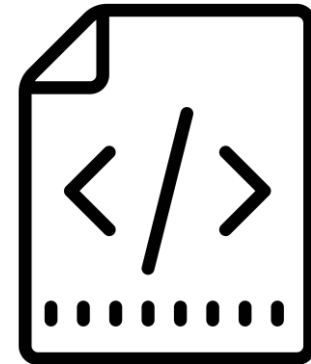
## Estructura



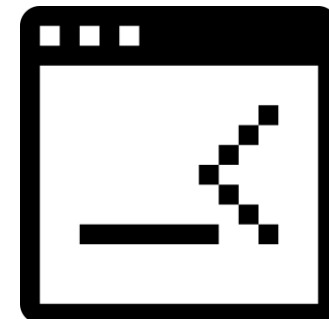
Scsynth



Sclang



Scide



# El lenguaje:

## Quarks



Creados por la comunidad y públicos



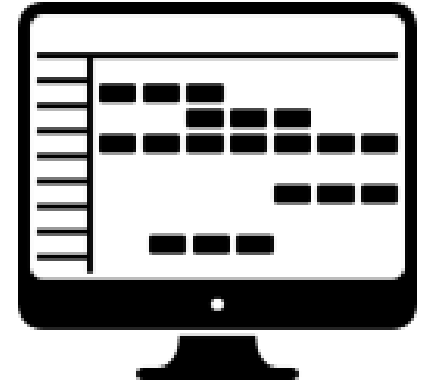
```
(  
  name: "DemonWidgets",  
  summary: "Widgets, gadgets and arcane devices for summoning demons.",  
  version: "1.0.0",  
  schelp: "DemonWidgets",  
  dependencies: ["Bjorkland", "Canvas3D", "cruciallib@tags/4.1.4"],  
  license: "GPL",  
  copyright: "Frank Furter, Dr.-Ing. 2015"  
)
```

# El lenguaje:

## Aplicaciones



Programación  
musical



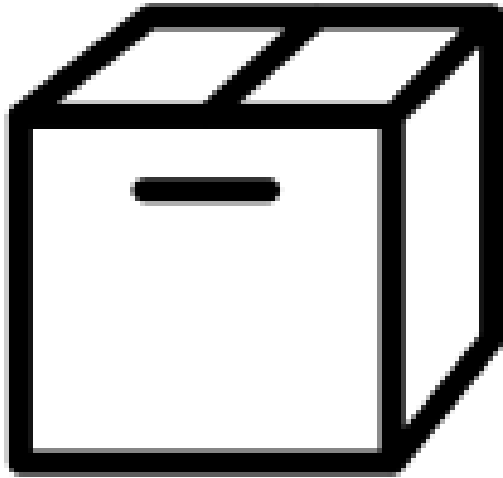
Experimentación  
acústica



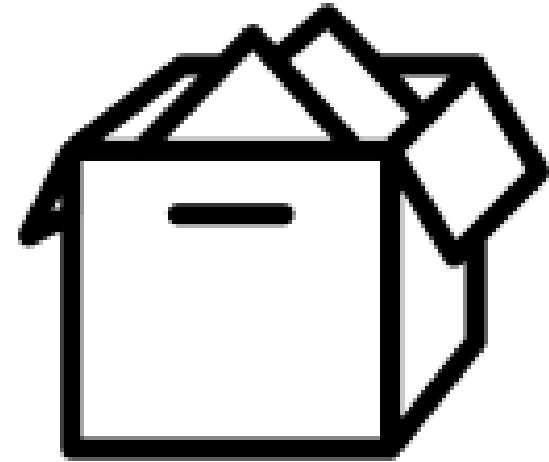
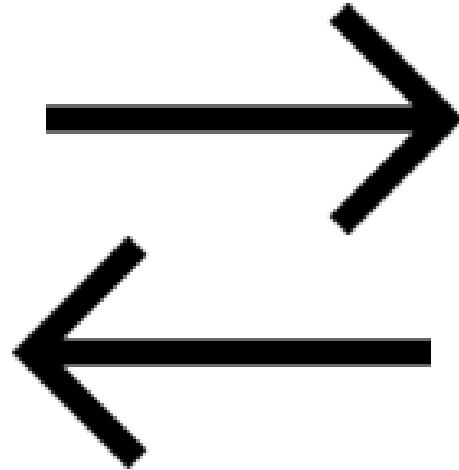
Live-coding



# Formato de la memoria: Modelo



Teoría



Ejemplos

# Formato de la memoria: Estructura

## Capítulos ↘



Fundamentos  
físicos



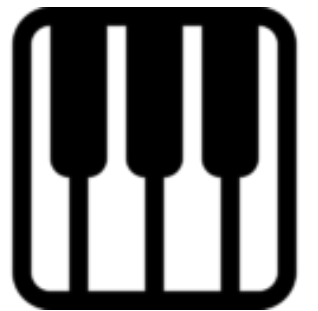
Objetos



Estructuras



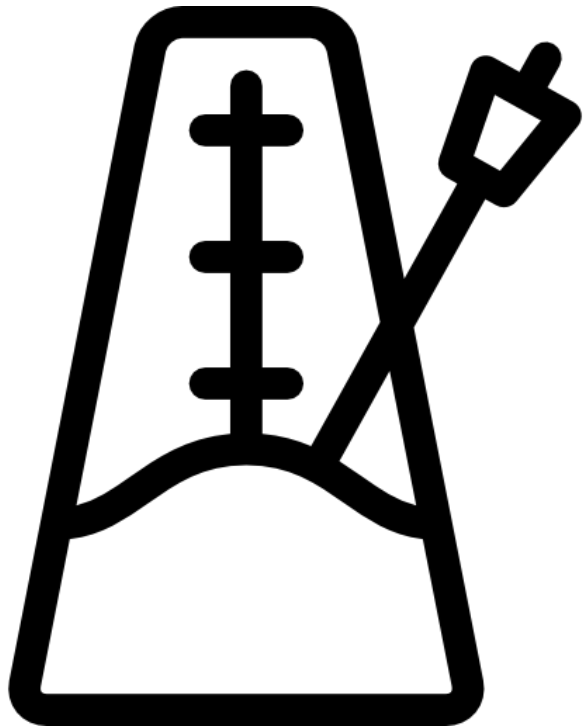
UGens



MIDI



# Experimentación: metrónomo.scd



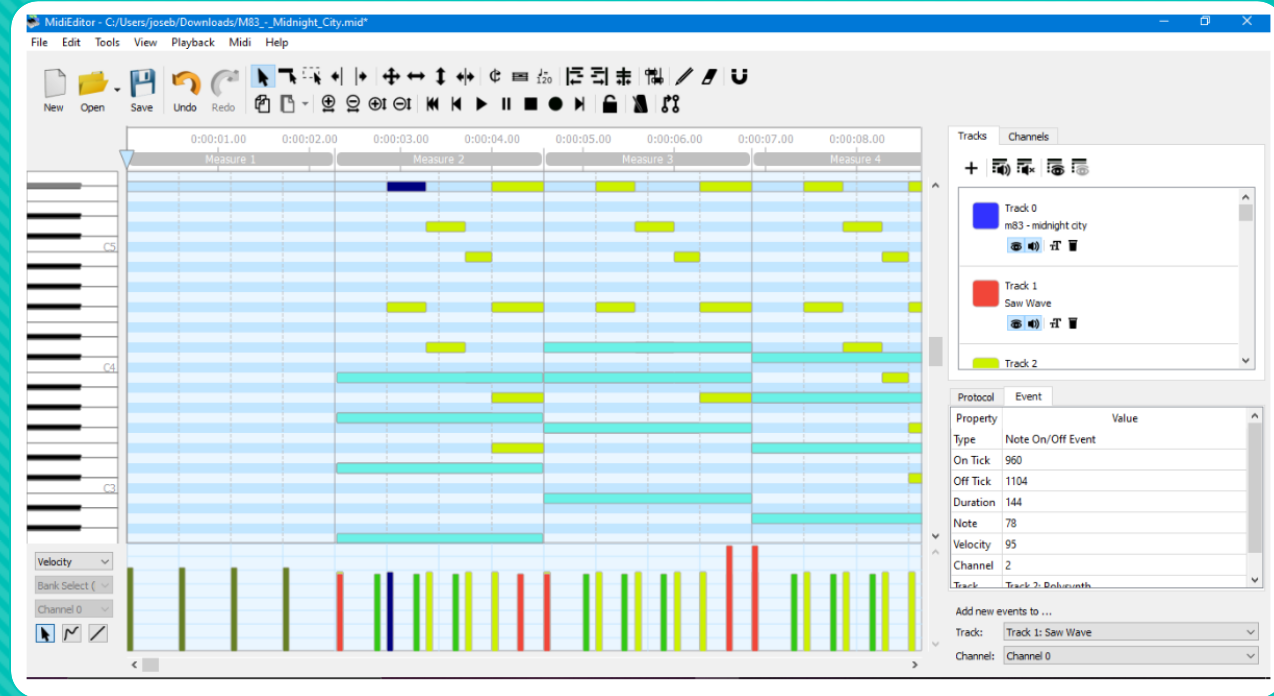
$$\text{beats por segundo} = \frac{60 \text{ segundos}}{\text{bpm}}$$

```
4 a = {arg bpm; Tdef(\Metronomo, {  
5   inf.do{  
6     {LFTri.ar(100)*Line.kr(1,0,0.1,doneAction:2)}.play;  
7       "tic".postln;  
8       (60/bpm).wait;  
9     }  
10  }  
11  
12  });  
13 }
```

```

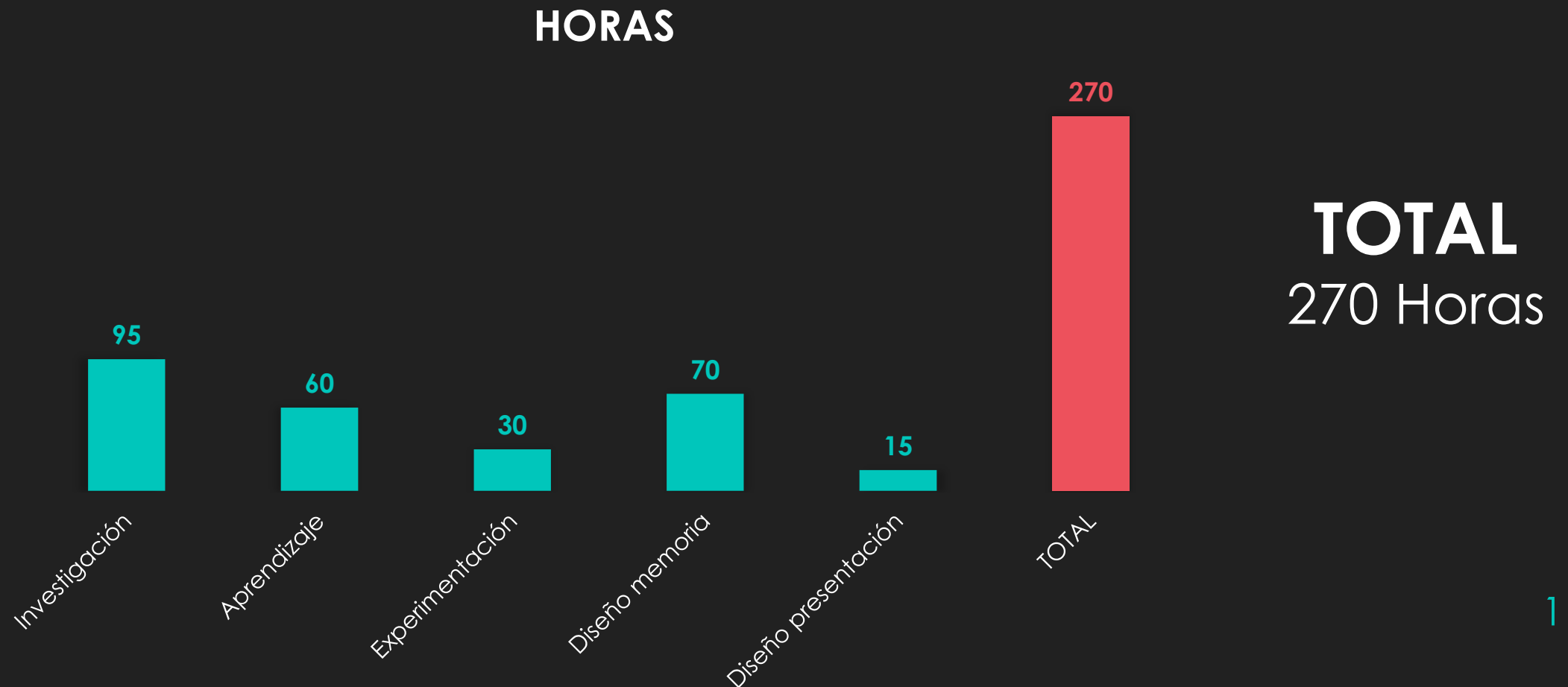
3 (
4 SynthDef.new(\midnightcitykeyboard, {
5   arg freq=440, atk=0.05, rel=1, amp=1.2, pan=0;
6   var sig, env;
7   env= EnvGen.kr(Env.new([0, 1, 0], [atk, rel], [-1, 1]), doneAction:2);
8   sig= Pan2.ar(LFTri.ar(freq), pan, amp)*env;
9   Out.ar(0, sig);
10 }).add;
11 )
12
13 (
14 p= Pbind(
15   \instrument, \midnightcitykeyboard,
16   \midinote, Pseq([
17     [78, 66], [74, 62], 71, [78, 66, 57, 52], 0,
18     [78, 66], 74, 71, [78, 66, 57], 0,
19     [78, 66], [74, 62], [71, 59], [78, 66, 54, 49], 0,
20     [78, 66], [74, 62], 71, [78, 66, 54, 49], [78, 66, 54, 49], [78, 66, 54, 49], [78, 66, 54, 49], [55, 50], 0
21   ], inf),
22   \dur, Pseq([
23     0.56, 0.56, 0.4, 0.82, 0.8,
24     0.56, 0.56, 0.4, 0.82, 0.8,
25     0.56, 0.56, 0.4, 0.82, 0.8,
26     0.56, 0.56, 0.4, 0.17, 0.17, 0.17, 0.17, 0.17, 0.5,
27   ], inf)
28 ), inf)
29 ).play;
30 )
31 )

```



# Experimentación: midnight-city-riff.scd

# Planificación temporal



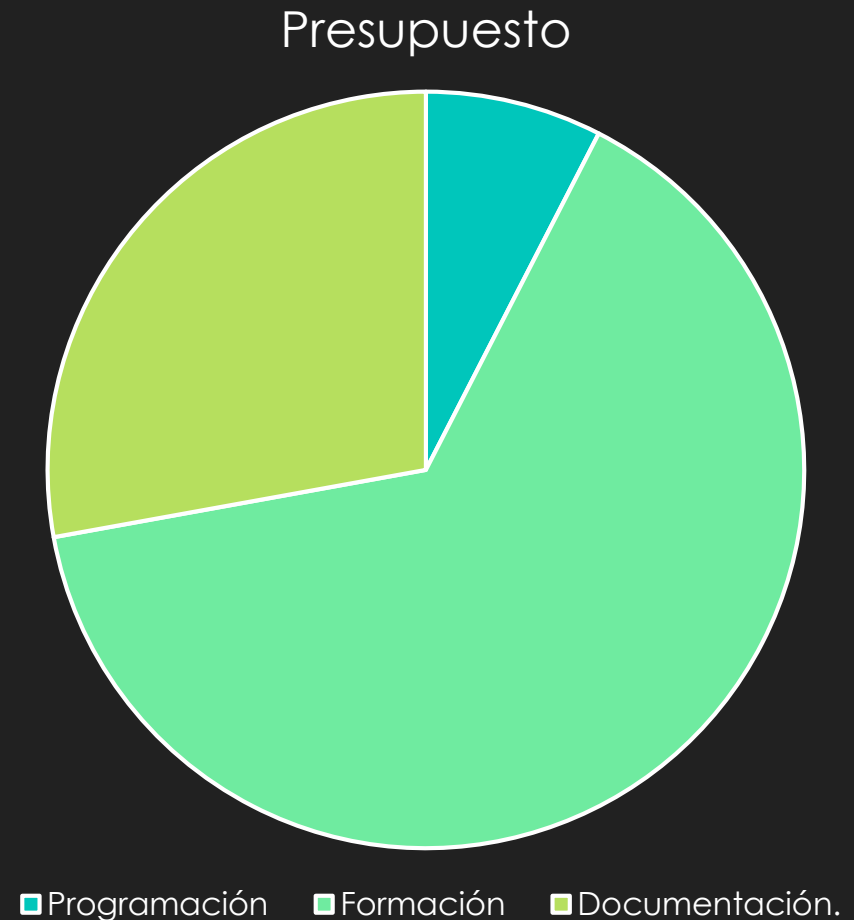
# Presupuesto estimado

**Programador:** 11'3€ /hora

**Formación:** 18'7€ /hora

**Técnico gestión de proyectos:** 14'7€ /hora  
(Documentación)

**TOTAL: 4487€**



# Conclusiones: SuperCollider

Lenguaje complejo con mucho potencial

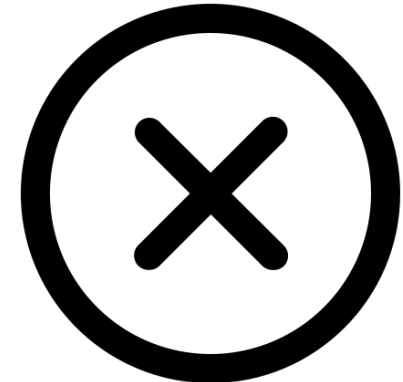
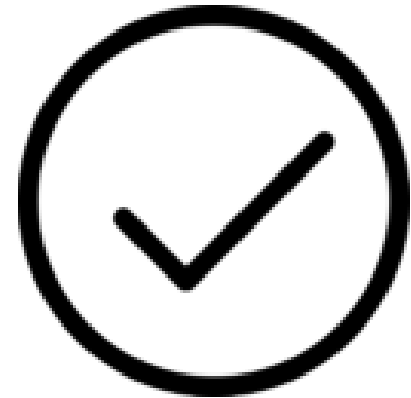
Ampliado y mantenido por la comunidad

Estructuras básicas sencillas y comunes en POO

Alta dificultad en tareas grandes

Inestable por su consumo computacional

Muy técnico



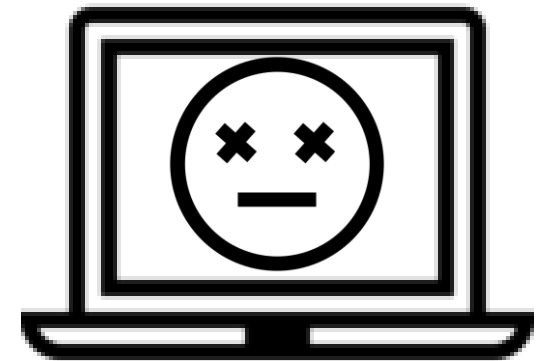
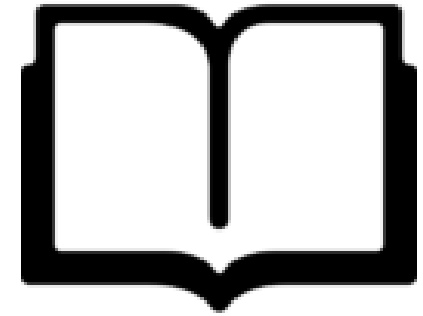
# Conclusiones finales

Poca documentación externa y mal elaborada

Documentación oficial muy compleja y escueta

Ejecuciones de código que cuelgan el equipo

Videotutoriales que abarcan demasiado en poco tiempo



# Bibliografía

13 sitios web, destacando:

- Web SuperCollider (<https://www.supercollider.github.io>)
- Curso de SuperCollider para principiantes (Ernesto Romero y Ezequiel Metri)
- Blog en español sobre SuperCollider  
(<http://supercolliderspanish.blogspot.com/>)
- Repositorio del proyecto (<https://github.com/joscarboz/SuperColliderTFG>)

# SuperCollider

José Luis Caro Bozzino

Grado en Ingeniería Informática de Software

Sevilla, 2020