

TECH CONFERENCE

# DotNet 2020

#DotNet2020

## Luces, cámara, ¡acción!



## ORGANIZATION

plain concepts 

## PLATINUM SPONSORS



## COLLABORATORS



# Thank you!



# Marco de la Cruz Sánchez

Programador de cosas en Plain Concepts

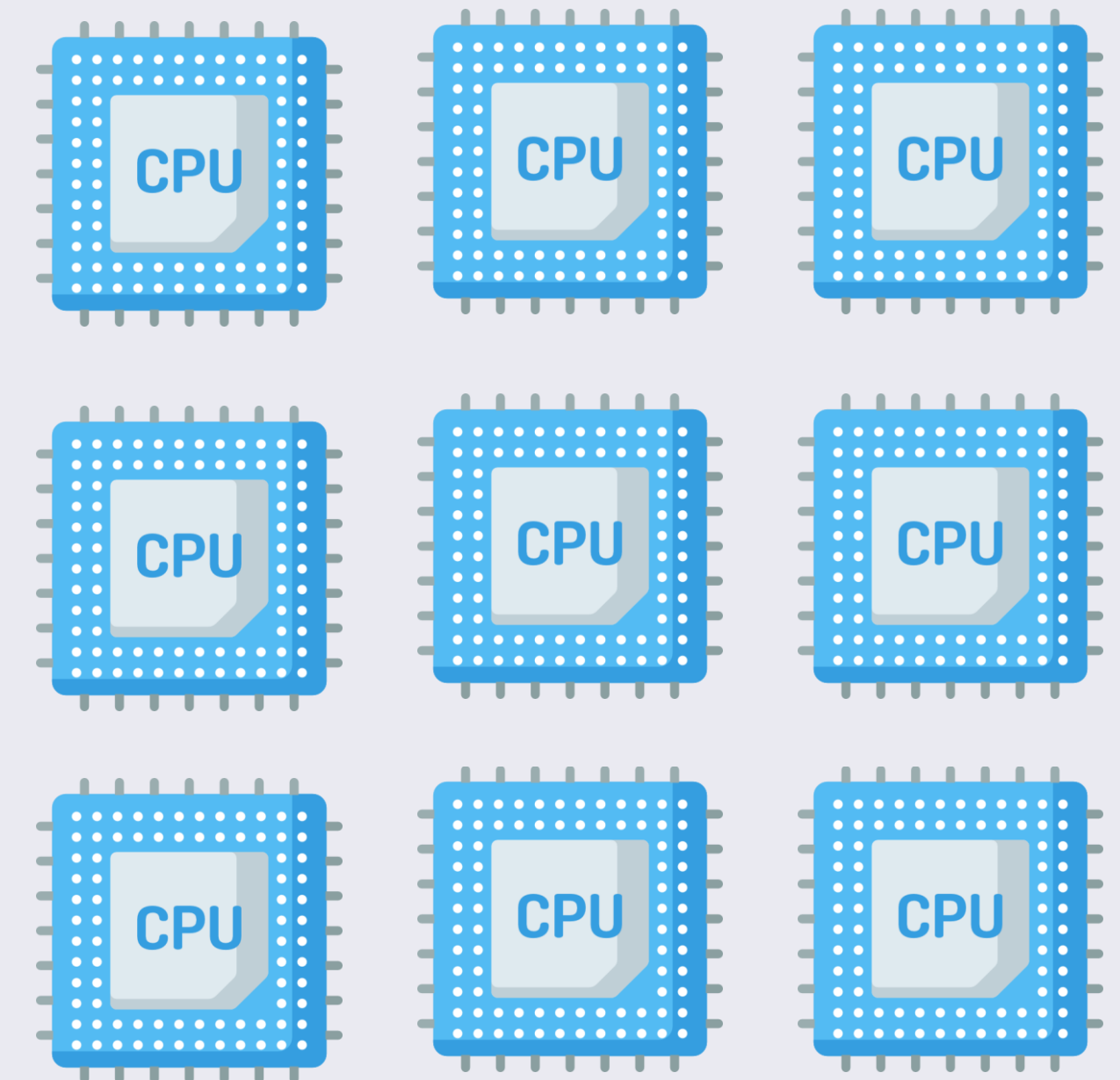
@beasync

mdcruz@plainconcepts.com

# La carrera de los nanómetros



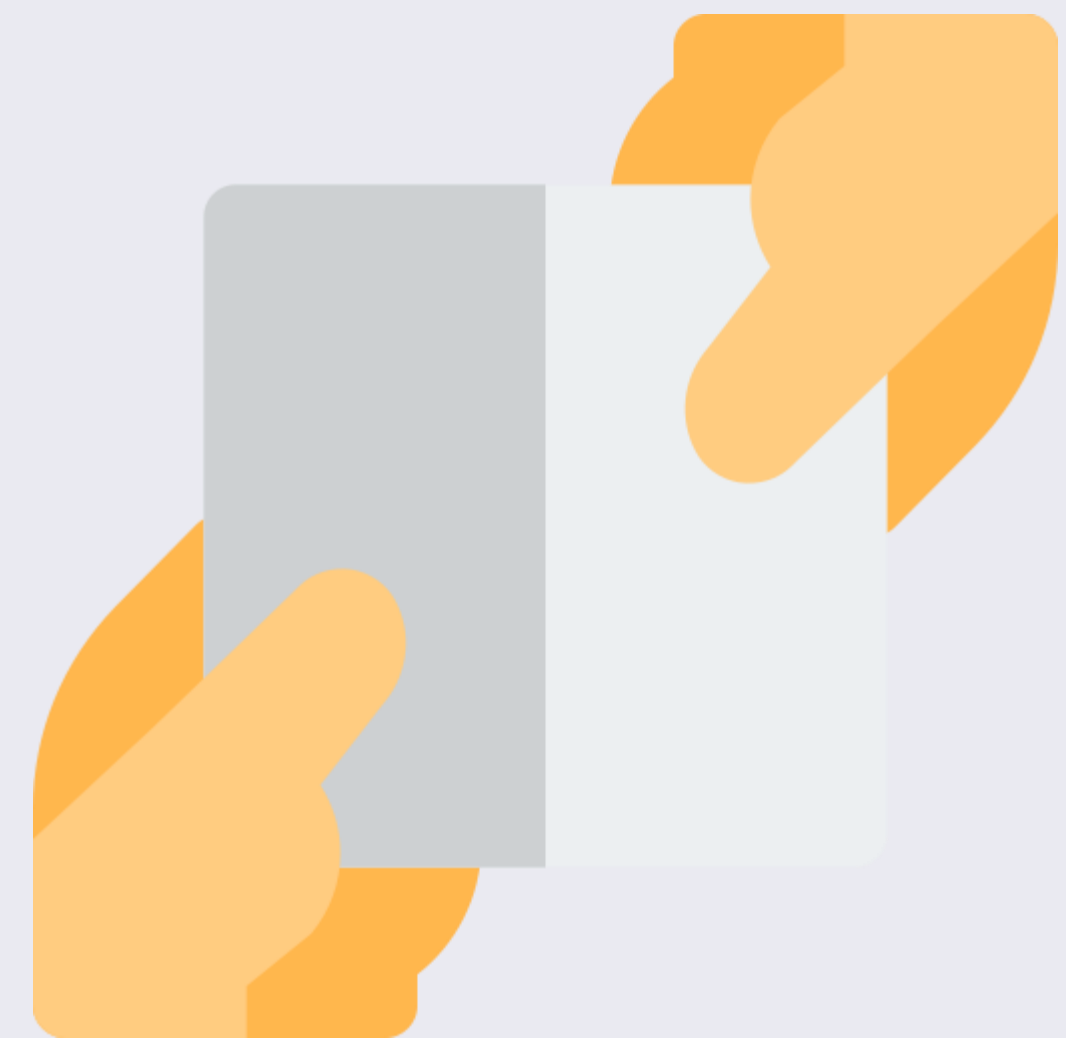
# Programación concurrente y distribuida



***"Los sistemas concurrentes  
no son fáciles de entender"***



# Threads y locks





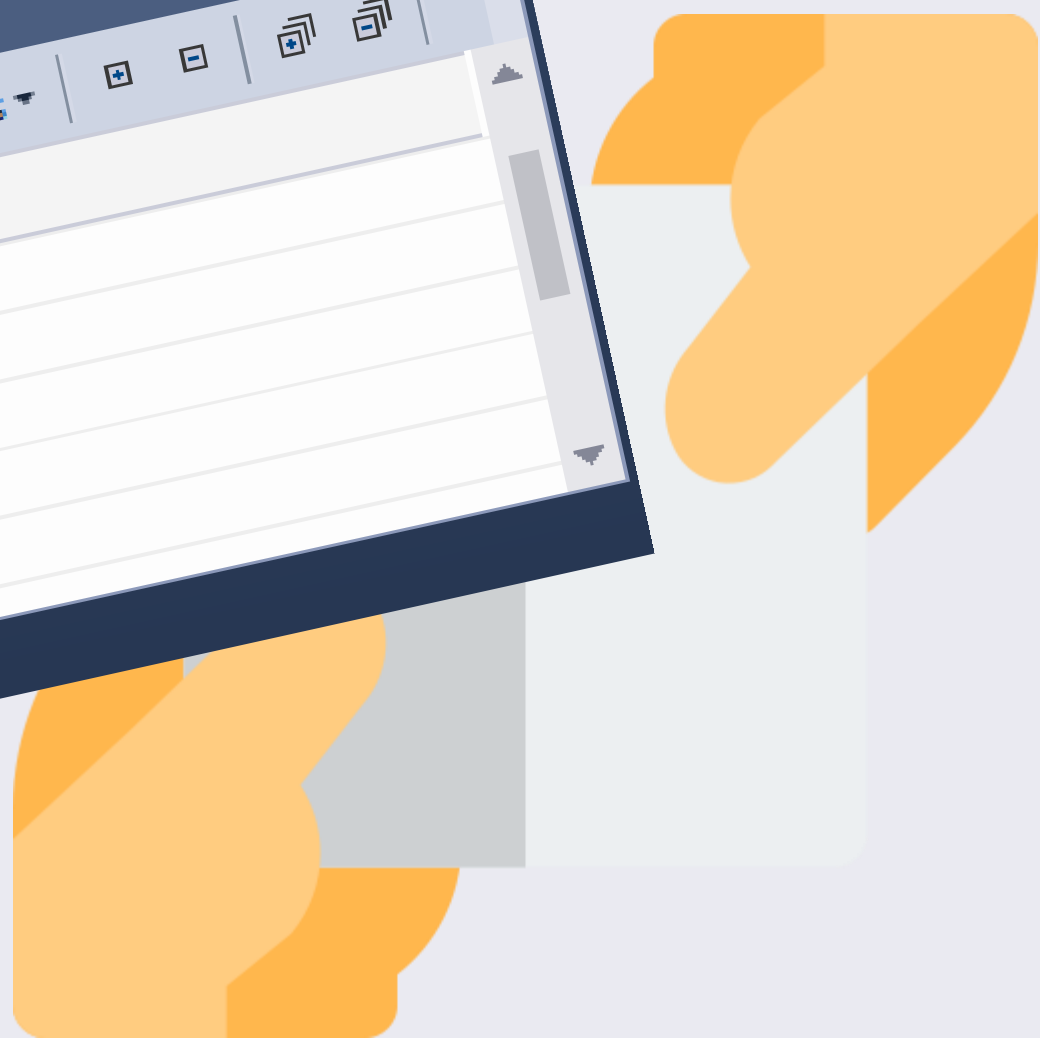
Threads

Search:

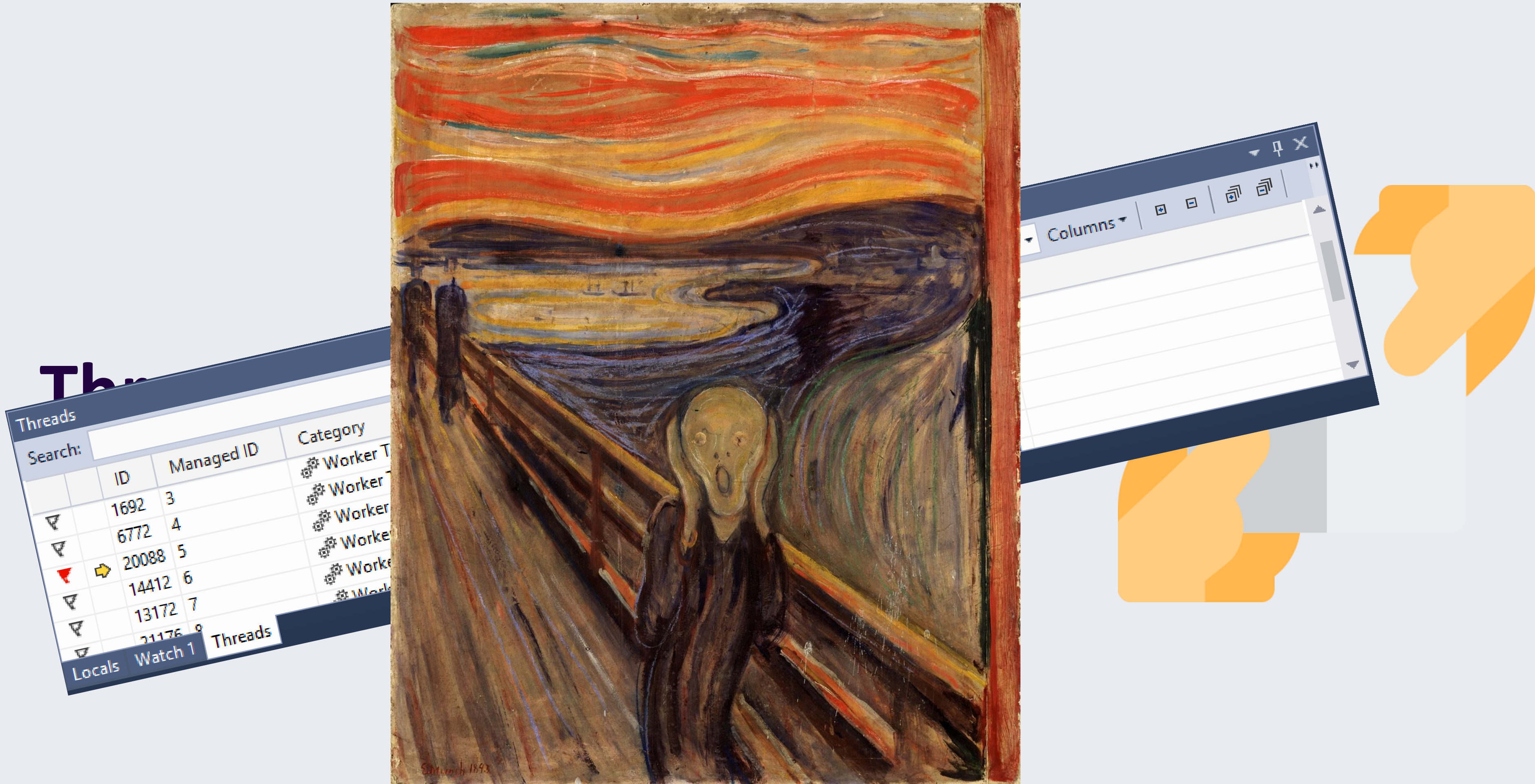
Group by: Process ID

|  | ID    | Managed ID | Category      | Name      | Location                                    |
|--|-------|------------|---------------|-----------|---|
|  | 1692  | 3          | Worker Thread | <No Name> | ConsoleApp11.exe!ServerClass.InstanceMethod |
|  | 6772  | 4          | Worker Thread | <No Name> | ConsoleApp11.exe!ServerClass.InstanceMethod |
|  | 20088 | 5          | Worker Thread | <No Name> | ConsoleApp11.exe!ServerClass.InstanceMethod |
|  | 14412 | 6          | Worker Thread | <No Name> | ConsoleApp11.exe!ServerClass.InstanceMethod |
|  | 13172 | 7          | Worker Thread | <No Name> | ConsoleApp11.exe!ServerClass.InstanceMethod |
|  | 21176 | 8          | Worker Thread | <No Name> | ConsoleApp11.exe!ServerClass.InstanceMethod |

Locals Watch 1 Threads









# El modelo de actores



**En ese caso,  
¿qué es el modelo de actores  
del que usted me habla?**



***"Un actor es la unidad  
fundamental de computación"***

Carl Hewitt



- 0. Procesamiento:** queremos que realice un trabajo
- 1. Persistencia:** queremos que recuerde cosas
- 2. Comunicación:** queremos que sea social



***“The isolated ants lived just six days, whereas group-living ants lived up to 66 days.”***

 Behavioural Ecology and Sociobiology



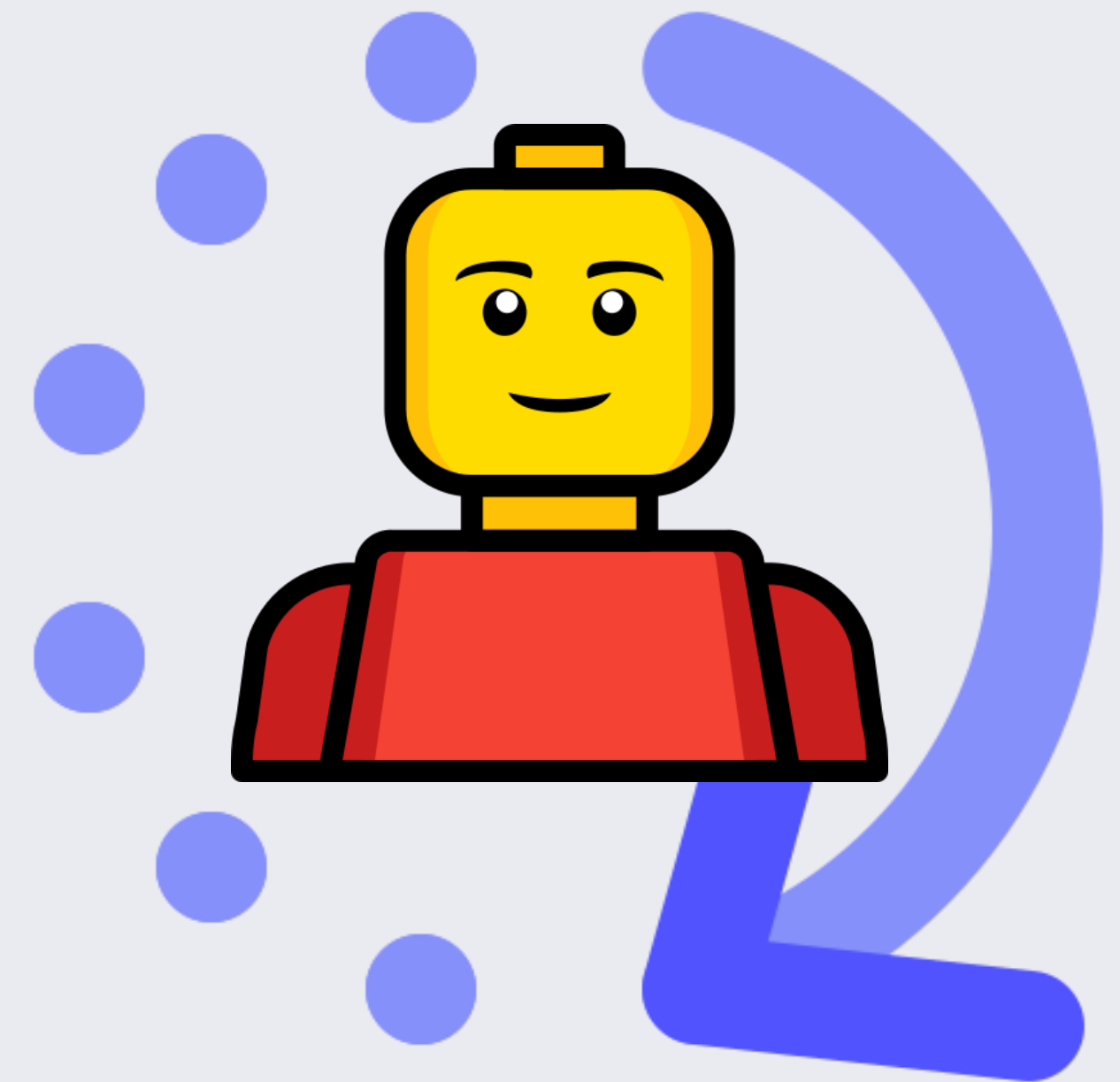
Illustration by Ellen Surrey



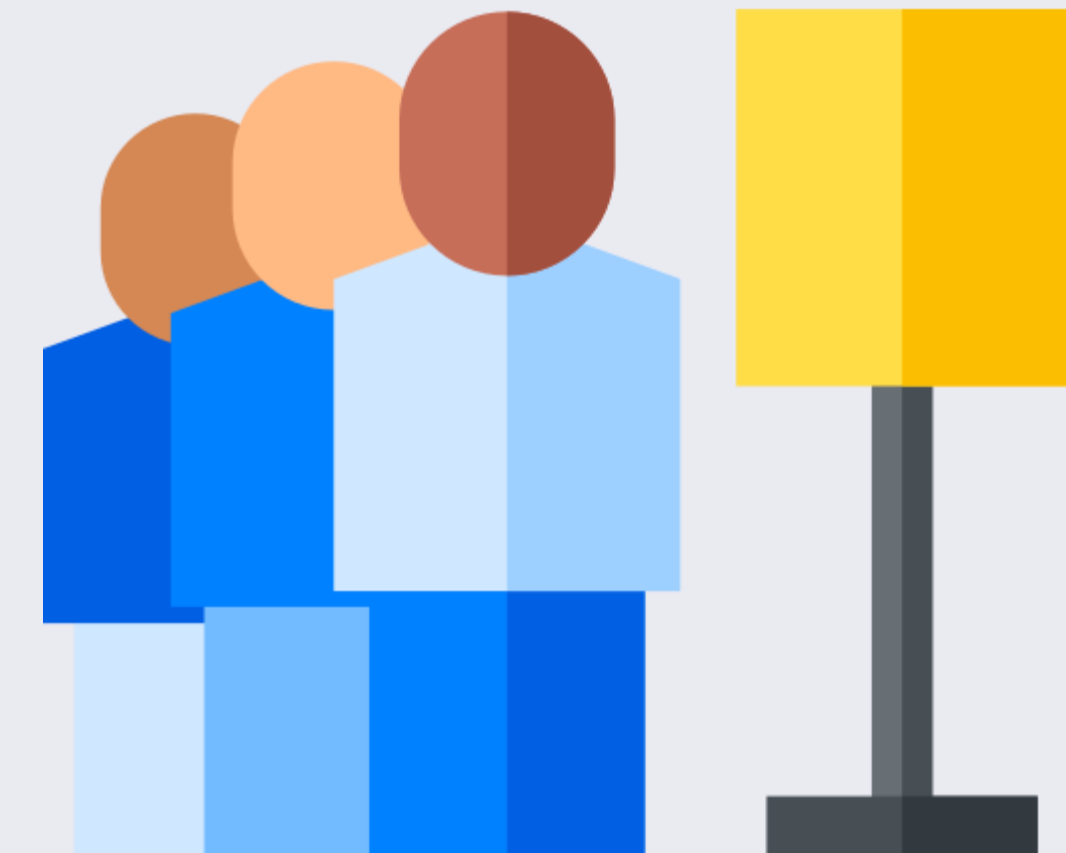
## *Actores y direcciones*



## *Mensajes y recursividad*



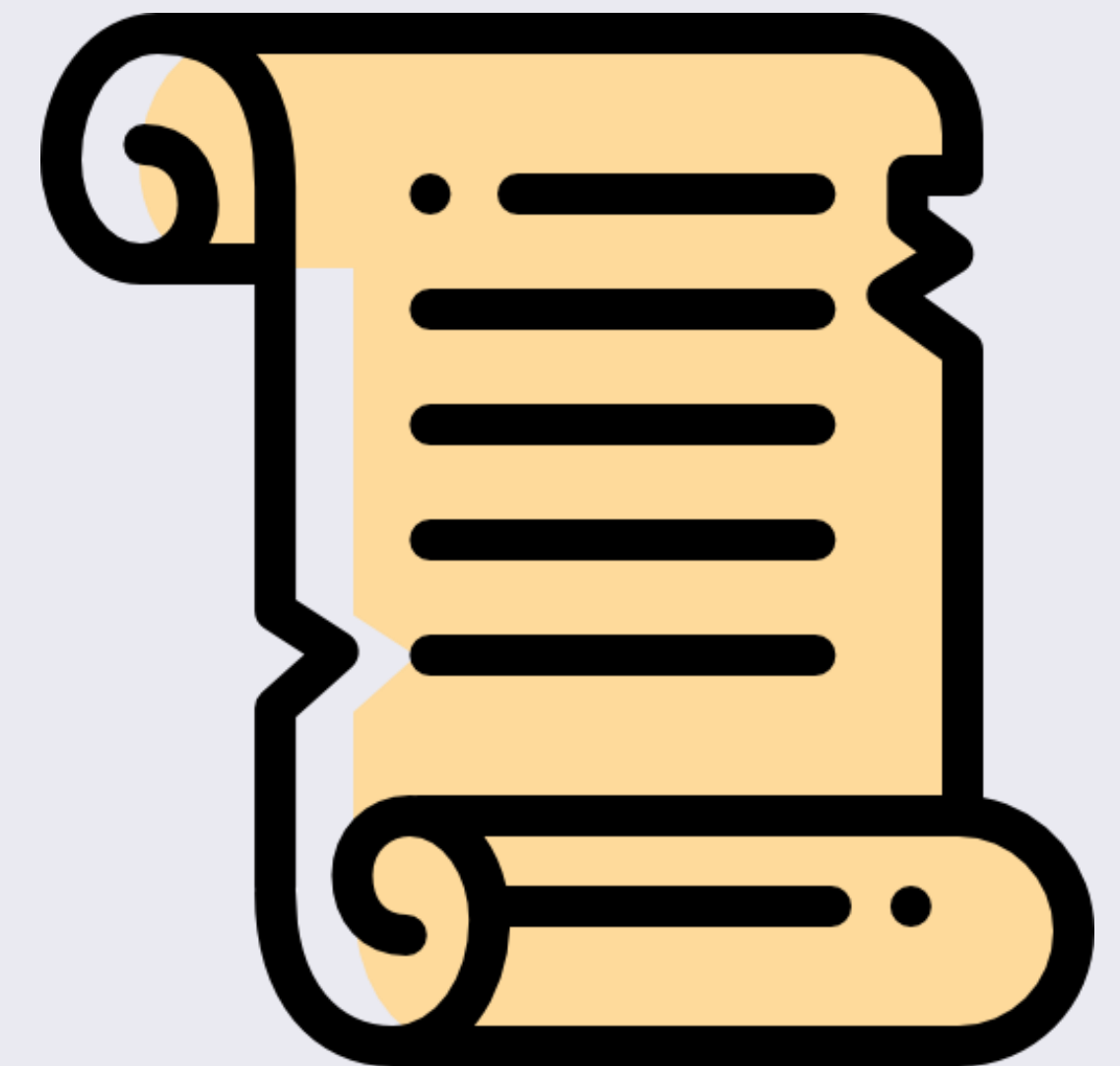
*De 1 en 1, señores...*



# ***Axiomas del modelo de actores***



0. **Crear más actores**
1. **Enviar mensajes a actores de los que disponga sus direcciones previamente**
2. **Designar que ocurrirá con el siguiente mensaje que reciba**



*En perspectiva,  
¿en que nos ayuda el modelo  
de actores?*





0. Eludimos la complejidad del multi-threading
1. Simplificación mediante actores single-threaded
2. Modelo de programación lock-free



**¿Y el performance?**



**Me interesa,  
¿siguiente paso?**



## Libros y Tesis

[An actor model implementation for the OCaml programming language](#)

[Reactive Messaging Patterns with the Actor Model](#)

## Vídeos

[The Actor Model \(everything you wanted to know...\)](#)

[Reliable Async Systems with Coyote \(Pt. 1 & 2\)](#)

[Distributed Application Runtime \(Dapr\)](#)

## Herramientas

<https://dapr.io/>

<https://microsoft.github.io/coyote/>

[Entity functions](#)



Thanks and ...  
See you soon!

Thanks also to the sponsors.  
Without whom this would not have been possible.

plain concepts 





<https://github.com/beasync/dotnet2020>

