



# 10 Pines

Diseño a la Gorra - Episodio 03



Hernán Wilkinson



hernan.wilkinson@10pines.com



@HernanWilkinson

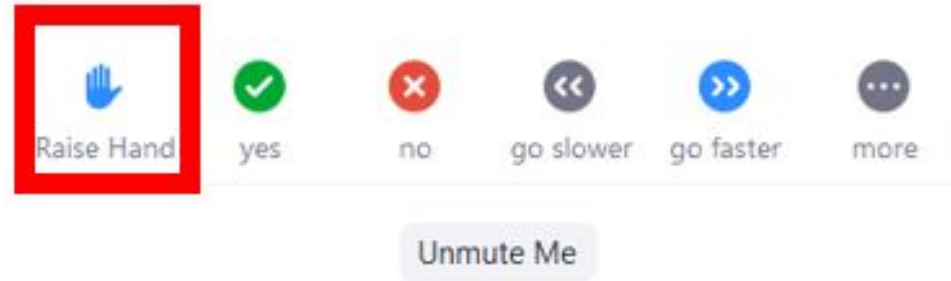


<https://alagorra.10pines.com>





Estar muteados a menos que sea necesario



No voy a poder leer el chat



La comunicación visual es importante. Usarla a discreción





# Diseño ¡a la gorra!

## ¡Bienvenidos!

Durante esta serie de Webinars exploraremos *qué* significa **Diseñar Software con Objetos** y *cómo* lo podemos hacer cada vez mejor.

Trataremos muchos temas que irán desde cuestiones filosóficas como qué significa Diseñar en nuestra profesión y dónde está expresado ese Diseño, pasando por consejos y heurísticas para diseñar "mejor" y terminado con ejemplos concretos de cómo aplicar esas heurísticas en la *vida real*.

Los webinars son "*language agnostic*", o sea que no dependen de un lenguaje de programación en particular, aunque los ejemplos que usaremos estarán hechos principalmente en **Java**, **JavaScript**, **Ruby**, **Python** y mi querido **Smalltalk** cuando amerite 😊.

Te esperamos todos los Martes a las 19 Hrs GMT-3 a partir del Martes 11 de Agosto de 2020. Para poder participar tenes que registrarte [acá](#).

Todo el código y presentaciones estarán disponibles para que lo puedan usar y consultar en cualquier momento [acá](#).

¡Trae ganas de aprender y pasarla bien!

## ¿Por qué a la Gorra?

Al igual que cuando Diseñamos Software está bueno usar una **Metáfora** para entender qué estamos modelando, en este caso usamos una metáfora para explicar cómo *financiaremos*

## Donaciones



\$100 - Casi una 🍷

Pagar

\$250 - Una buena 🍺

Pagar

\$500 - Menos que 🍔 + 🍷

Pagar



Donate



¿Querés donar un monto variado o en otra plataforma?



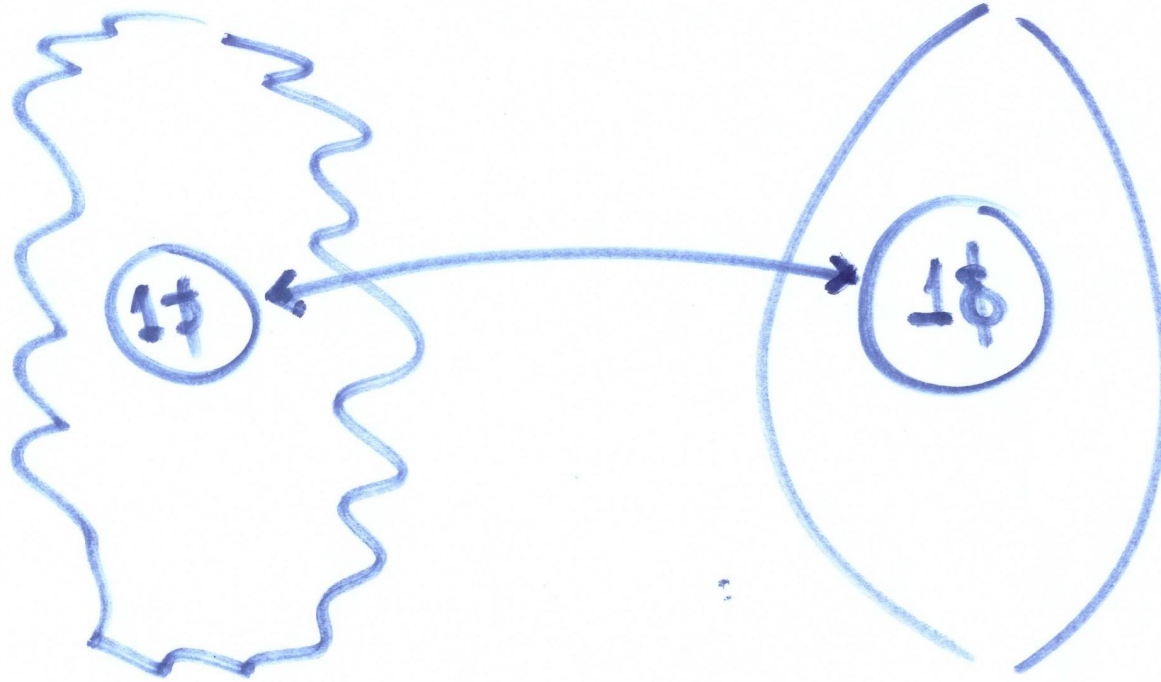
Dictado por:  
**Hernán Wilkinson**

<https://alagorra.10pines.com>

¿Qué vimos hasta ahora?



**Objeto = Representación** esencial de un **ente** del  
Dominio de Problema



Esa representación se realiza a través de los **mensajes**  
que sabe **responder**





# En los Diseños “pasa el tiempo” y deben Enseñarnos



$$D = \frac{1}{c} \frac{1}{l} \frac{dl}{dt} = \frac{1}{c} \frac{1}{P} \frac{dP}{dt}$$

$$D^2 = \frac{1}{P^2} \frac{P_0 - P}{P} \sim \frac{1}{P^2} \quad (1a)$$

$$D^2 = \frac{KQ}{3} \frac{P_0 - P}{P_0} \sim \frac{1}{2} KQ \quad (2a)$$

$$D^2 \sim 10^{-53}$$

$$Q \sim 10^{-26}$$

$$P \sim 10^8 \text{ g./cm}^3$$

$$\tau \sim 10^{10} (10^{11}) \text{ y}$$





Un Objeto debe **representar** el ente del dominio **de manera válida desde el momento en que existe**

Un Objeto debe “**enseñar**” de manera explícita **qué necesita para representar** ese ente y **fallar si se lo crea incorrectamente**  
(fail fast)





- **H1: Crear Objetos Completos**
- **H2: ¡No romper el encapsulamiento!**
- **H3: Crear Objetos Válidos**



¿Time o Hora o Tiempo o Momento?  
What time is it? → ¿Qué hora es?

¿Qué tiempo es? 🙅  
¿Qué momento es? 🙅



# Controversia vs. Confusión



*Number.isInteger(hour)*

¿Chequeo de tipo o  
chequeo de qué número representa?



1

①



↓

Ⓢ

↓





1

① I

①

①

1



1

0.1 I

1

I

1

1

1.0



1

0.1 I

1

I

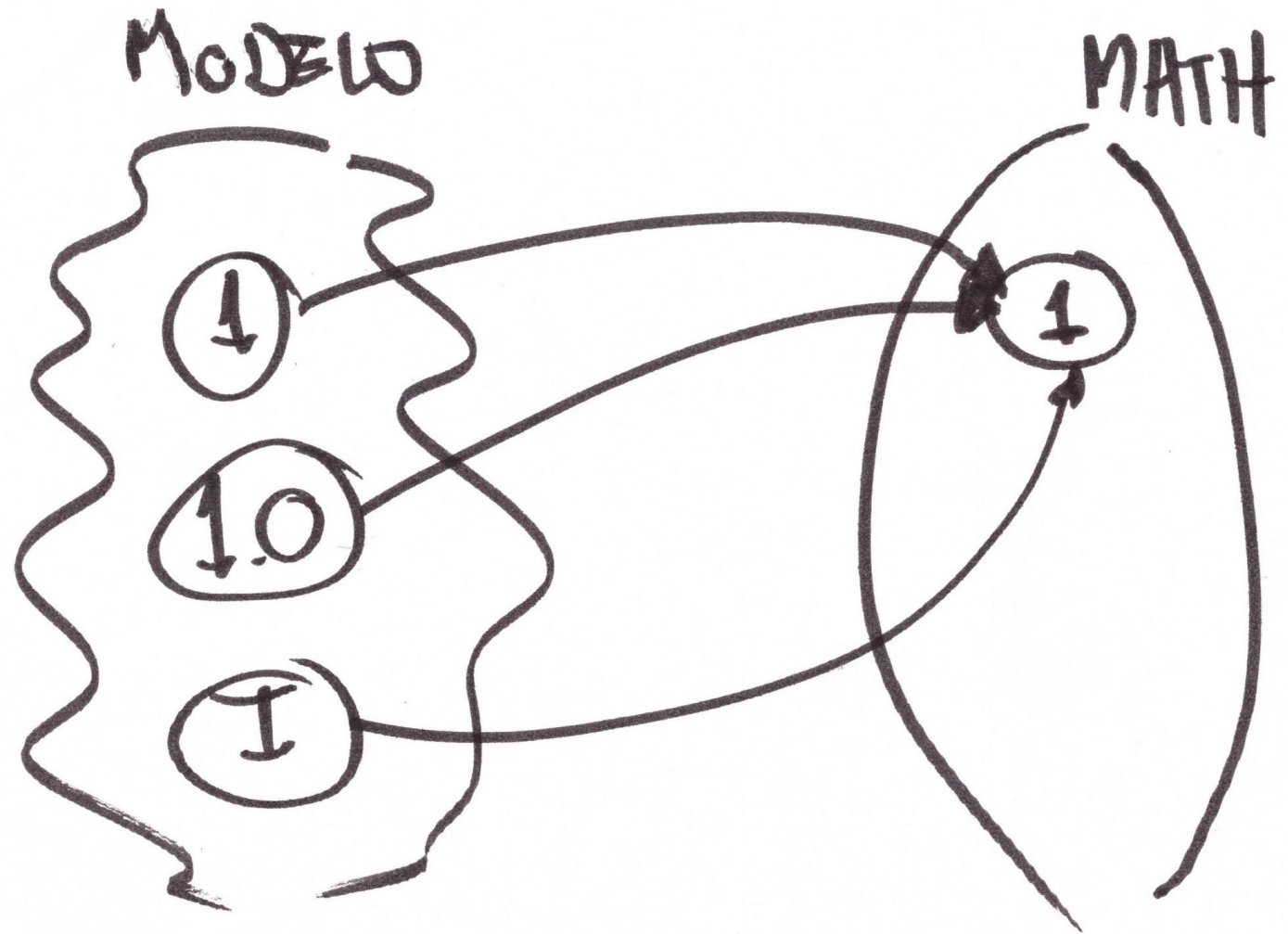
1

1

1.0

3/3





```
if (!(hour-Math.trunc(hour)===0 && hour >= 0 && hour <= 23))  
    throw new Error(Time.INVALID_HOUR);
```



*Number.isInteger(hour)* **no es chequeo de tipo**, es  
verificar si “hour” representa un “número entero”  
sin importar la implementación

*int* no son números

*float* no son números





Tip: Crea tus propios “modelos técnicos” o frameworks.

¡Hay que tener coraje! 💪💪💪

**En latinoamérica tenemos que dejar de  
“consumir” software y empezar a “construir”  
software**



# Metáfora



*ModelCreator, CompositeModelCreator*  
¿son buenos nombres?





*ModelCreator* →  
Asistente para Completar Formulario  
(**FormCompletionAssistant**)

*CompositeModelCreator* →  
Asistente para Completar Seccion  
(**FormSectionCompletionAssistant**)



H4: Usar **metáforas** para “*mapear*” el problema

Tip: No usar nombres de patrones en las clases  
como *CompositeModelCreator*





# Los Objetos no viven aislados



## H5: Modelar el “conjunto” de Objetos

Tip: Se pueden tener distintas implementaciones del “conjunto”: Transient, persistent, client, etc



H6: Usar updates atómicos



# Conclusiones



H4: Usar una metáfora para “mapear” el problema

H5: Modelar los “conjuntos” de objetos

H6: Usar updates atómicos





# Diseño ¡a la gorra!

## ¡Bienvenidos!

Durante esta serie de Webinars exploraremos *qué* significa **Diseñar Software con Objetos** y *cómo* lo podemos hacer cada vez mejor.

Trataremos muchos temas que irán desde cuestiones filosóficas como qué significa Diseñar en nuestra profesión y dónde está expresado ese Diseño, pasando por consejos y heurísticas para diseñar "mejor" y terminado con ejemplos concretos de cómo aplicar esas heurísticas en la *vida real*.

Los webinars son "*language agnostic*", o sea que no dependen de un lenguaje de programación en particular, aunque los ejemplos que usaremos estarán hechos principalmente en **Java**, **JavaScript**, **Ruby**, **Python** y mi querido **Smalltalk** cuando amerite 😊.

Te esperamos todos los Martes a las 19 Hrs GMT-3 a partir del Martes 11 de Agosto de 2020. Para poder participar tenes que registrarte [acá](#).

Todo el código y presentaciones estarán disponibles para que lo puedan usar y consultar en cualquier momento [acá](#).

¡Trae ganas de aprender y pasarla bien!

## ¿Por qué a la Gorra?

Al igual que cuando Diseñamos Software está bueno usar una **Metáfora** para entender qué estamos modelando, en este caso usamos una metáfora para explicar cómo *financiaremos*

## Donaciones



\$100 - Casi una 🍷

Pagar

\$250 - Una buena 🍺

Pagar

\$500 - Menos que 🍔 + 🍷

Pagar



Donate



¿Querés donar un monto variado o en otra plataforma?



Dictado por:  
**Hernán Wilkinson**

<https://alagorra.10pines.com>



Muchas gracias





# 10 Pines

Creative Software Development



[10pines.com](http://10pines.com)



[info@10pines.com](mailto:info@10pines.com)



+54 (011) 6091-3125 / 4893-2057



Av. Leandro N. Alem 896 6° - Bs. As. - Argentina



@10pines