

(**Conocimiento** expresado en un)

***Modelo Computable***

de un ***Dominio de Problema***

de la Realidad



# Conclusiones

- El modelo está expresado en el código fuente
- No es fácil estimar el “diseño”
- La “construcción” no cuesta nada, lo costoso es el diseño
- Un “modelo computable” implica “ejecución” que implica “paso del tiempo”



“Buen Software” == “Buen Modelo”



# Buen Modelo

- **Eje Implementativo:** Cómo “ejecuta” en el ambiente técnico
- **Eje Descriptivo:** Qué tan bien está descrito el modelo, qué tan “entendible” es
- **Eje Funcional:** Qué tan buena es la representación del dominio



## Eje Implementativo

Un modelo es bueno cuando ejecuta en el tiempo esperado usando los recursos definidos como necesarios

- Performance
- Espacio
- Escalabilidad
- Todo lo relacionado con “Requerimientos No Funcionales” (Atributos de Calidad)
- Es la parte de “detalle” del desarrollo



# Ejemplo



# Eje Descriptivo

Un modelo es bueno cuando “nos enseña”, se lo puede “entender” y por lo tanto “cambiar”

- Importantísimo usar buenos nombres
- Importantísimo usar mismo lenguaje que el del dominio de negocio
- El código debe ser “agradable” (habitable)
- Es la parte “artística” del desarrollo
  - <https://blog.10pines.com/2012/02/02/the-art-of-naming/>
  - <https://blog.10pines.com/2012/01/12/about-names-when-designing-with-objects/>



# Ejemplo



# Eje Funcional

Un modelo es bueno cuando puede representar correctamente toda observación de aquello que modela

- Si aparece algo nuevo en el dominio, debe aparecer algo nuevo en el modelo (no modificarlo)
- Si se modifica algo del dominio, solo se debe modificar su representación en el modelo
- Relación 1:1 dominio-modelo (isomorfismo)
- Es la parte “observacional” del desarrollo



# Ejemplo



# Interludio



Los calendarios son un excelente ejemplo de modelado y uso de metáforas





Available online at [www.sciencedirect.com](http://www.sciencedirect.com)



Computer Languages, Systems & Structures ■■■ (■■■) ■■■-■■■

COMPUTER  
LANGUAGES,  
SYSTEMS &  
STRUCTURES

[www.elsevier.com/locate/cl](http://www.elsevier.com/locate/cl)

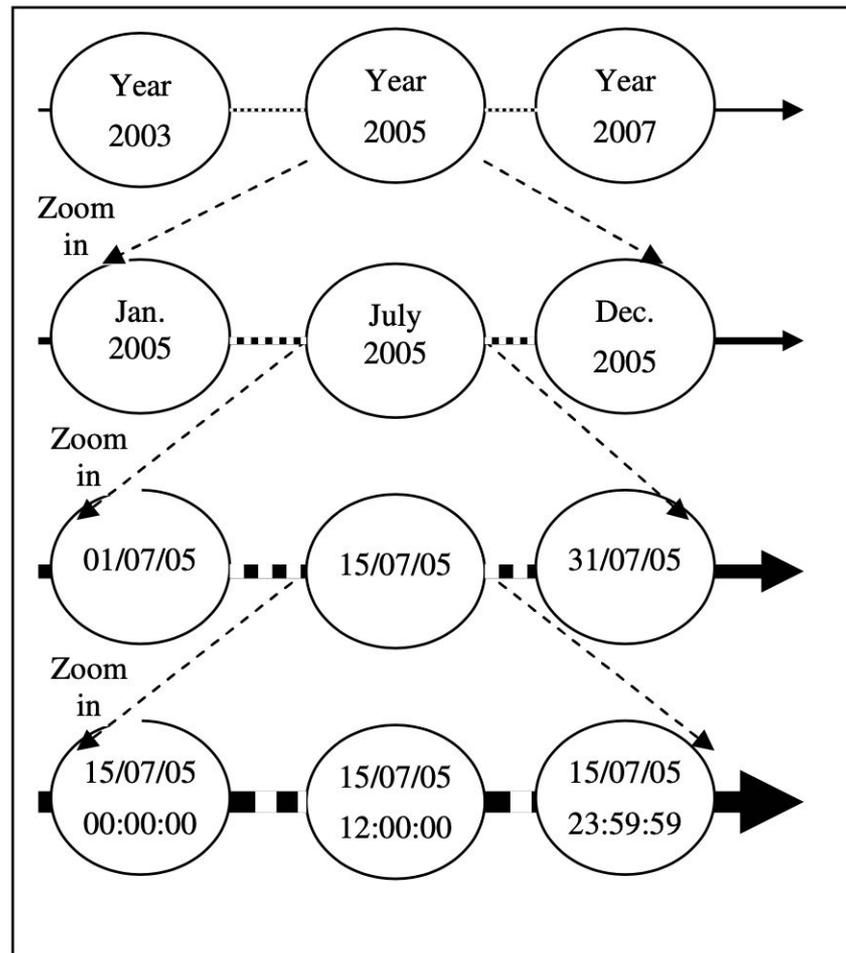
## A point based model of the Gregorian Calendar

Hernán Wilkinson<sup>a,\*</sup>,<sup>1</sup>, Máximo Prieto<sup>b</sup>, Luciano Romeo<sup>a,2</sup>

<sup>a</sup>Tacuarí 202, 7mo Piso, C1071AAF, Buenos Aires, Argentina

<sup>b</sup>Lifia - Facultad de Informática, Universidad Nacional de La Plata, cc11, 1900, La Plata, Argentina

Received 3 October 2005



Un **muy buen ejemplo** de un  
**muy mal modelo** es  
*java.util.Calendar*



# Fin de Interludio



# Conclusiones - Buen Modelo

- **Eje Implementativo:** Cómo “ejecuta” en el ambiente técnico
- **Eje Descriptivo:** Qué tan bien está descrito el modelo, qué tan “entendible” es
- **Eje Funcional:** Qué tan buena es la representación del dominio

