



XXIII Salón de Informática

Metodologías, Herramientas y Procesos Modernos de Desarrollo de Software

Bogotá, Septiembre 9 a 12 de 2003

Patrones de Software

“La lingua-franca de los desarrolladores”

Jorge Humberto Arias Bedoya
Pontificia Universidad Javeriana
jo-arias@javeriana.edu.co



AGENDA

- Motivación
- Definición
- Categorías
- Lenguajes y catálogos
- Conclusiones



Motivación

**Aplicación
empresarial de
gran escala**



Motivación

**Aplicación
empresarial de
gran escala**

= Lógica Funcional



Motivación

- Requerimientos funcionales
- Casos de Uso
- Modelamiento del negocio

**Aplicación
empresarial de
gran escala**

= Lógica Funcional





Motivación

- Requerimientos funcionales
- Casos de Uso
- Modelamiento del negocio

**Aplicación
empresarial de
gran escala**

= Lógica Funcional + Lógica No funcional





Motivación

- Requerimientos funcionales
- Casos de Uso
- Modelamiento del negocio

**Aplicación
empresarial de
gran escala**

= Lógica Funcional + Lógica No funcional

- Integración (EAI, Connector)
- Acceso concurrente
- Transacciones distribuidas (ACID)
- Manejo de memoria
- Seguridad.
- Alta disponibilidad
- Manejo de threads.
- Pool de conexiones.
- Escalabilidad



Motivación

- Requerimientos funcionales
- Casos de Uso
- Modelamiento del negocio

¿Usted sólo costea esto?

**Aplicación
empresarial de
gran escala**

= **Lógica Funcional** + **Lógica No funcional**

- Integración (EAI, Connector)
- Acceso concurrente
- Transacciones distribuidas (ACID)
- Manejo de memoria
- Seguridad.
- Alta disponibilidad
- Manejo de threads.
- Pool de conexiones.
- Escalabilidad



Motivación

- Requerimientos funcionales
- Casos de Uso
- Modelamiento del negocio

**Aplicación
empresarial de
gran escala**

= Lógica Funcional + Lógica No funcional

**J2EE
.NET**

- Integración (EAI, Connector)
- Acceso concurrente
- Transacciones distribuidas (ACID)
- Manejo de memoria
- Seguridad.
- Alta disponibilidad
- Manejo de threads.
- Pool de conexiones.
- Escalabilidad



Motivación

**Aplicación
empresarial de
gran escala**

Requerimientos
Funcionales
= **Lógica Funcional** +

**Plataforma
Componentes
(J2EE, .NET)**

Requerimientos
No- Funcionales



Motivación

**Aplicación
empresarial de
gran escala**

Requerimientos
Funcionales
= **Lógica Funcional** +

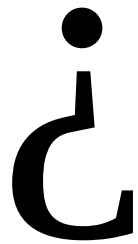
Libros
- Aprendiendo J2EE
en 21 días.
- J2EE al instante.

**Plataforma
Componentes
(J2EE, .NET)**

Requerimientos
No- Funcionales



Motivación



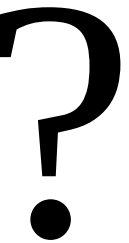
**Aplicación
empresarial
de gran
escala**

Requerimientos
Funcionales
= **Lógica Funcional** +

Libros
- Aprendiendo J2EE
en 21 días.
- J2EE al instante.

**Plataforma
Componentes
(J2EE, .NET)**

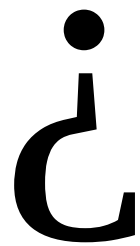
Requerimientos
No- Funcionales



¿Será suficiente?



Motivación



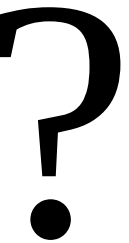
**Aplicación
empresarial
de gran
escala**

Requerimientos
Funcionales
= **Lógica Funcional** +

Libros
- Aprendiendo J2EE
en 21 días.
- J2EE al instante.

**Plataforma
Componentes
(J2EE, .NET)**

Requerimientos
No- Funcionales

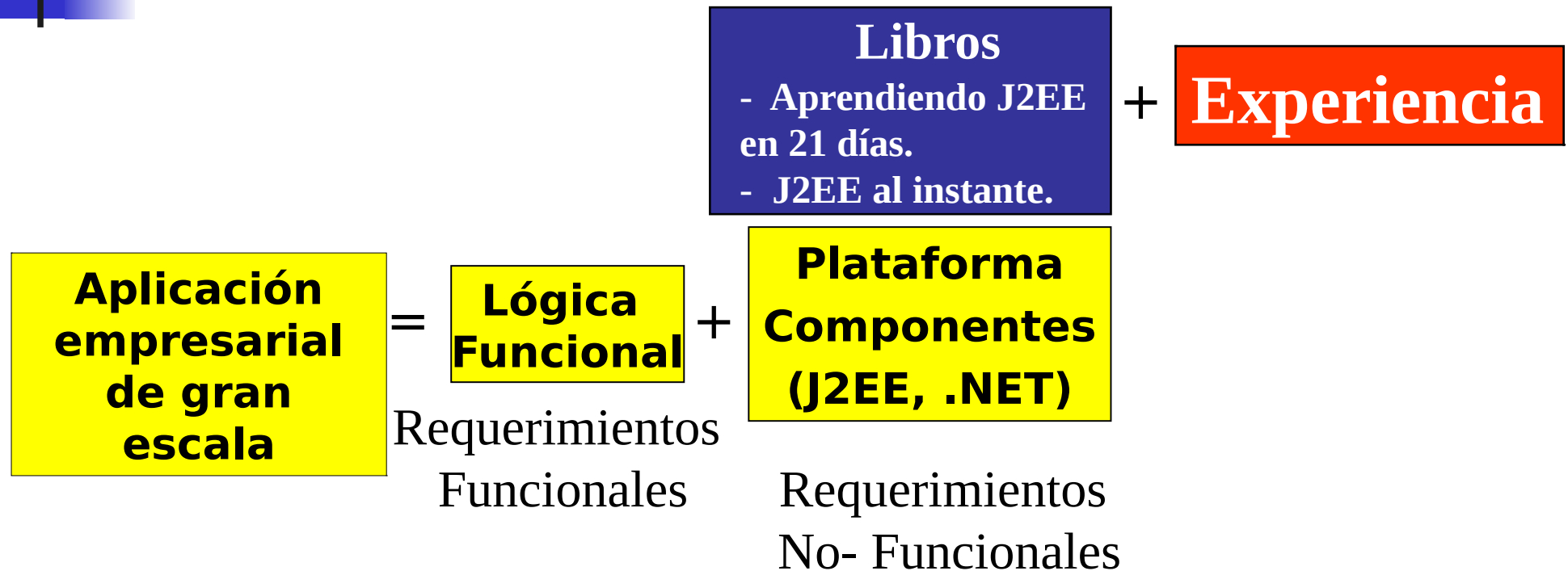


¿Será suficiente?

¿Qué falta?

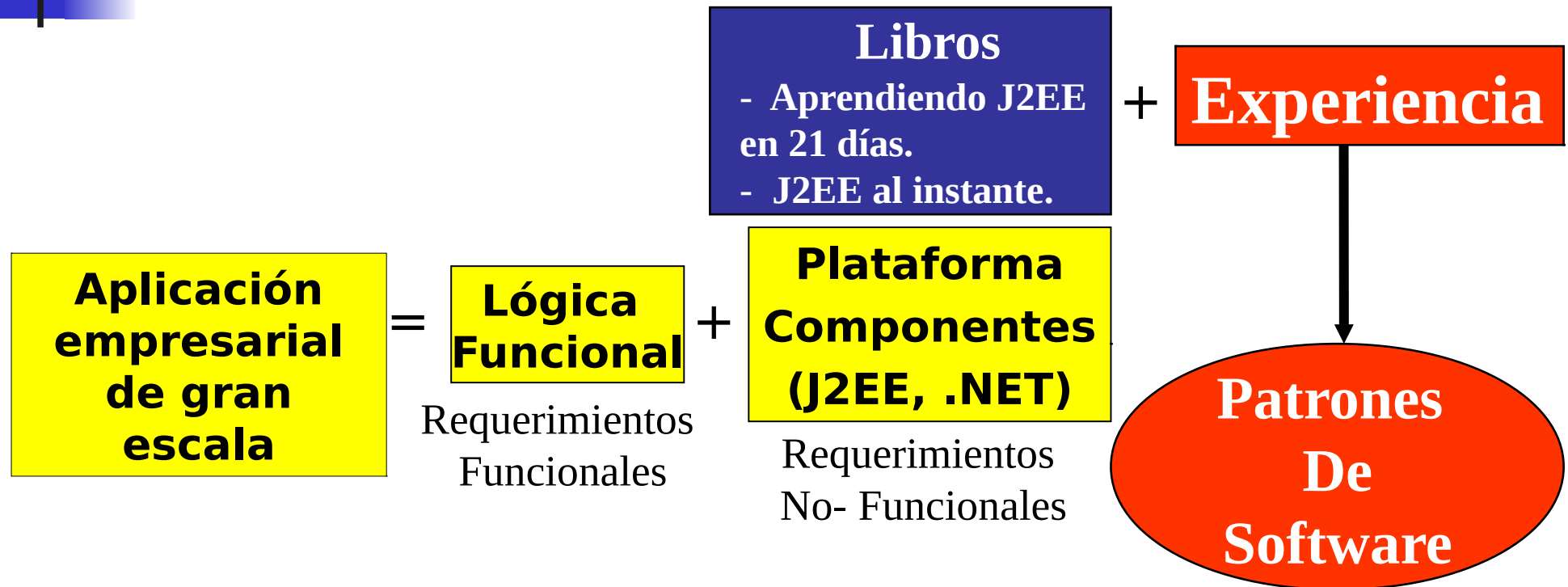


Motivación



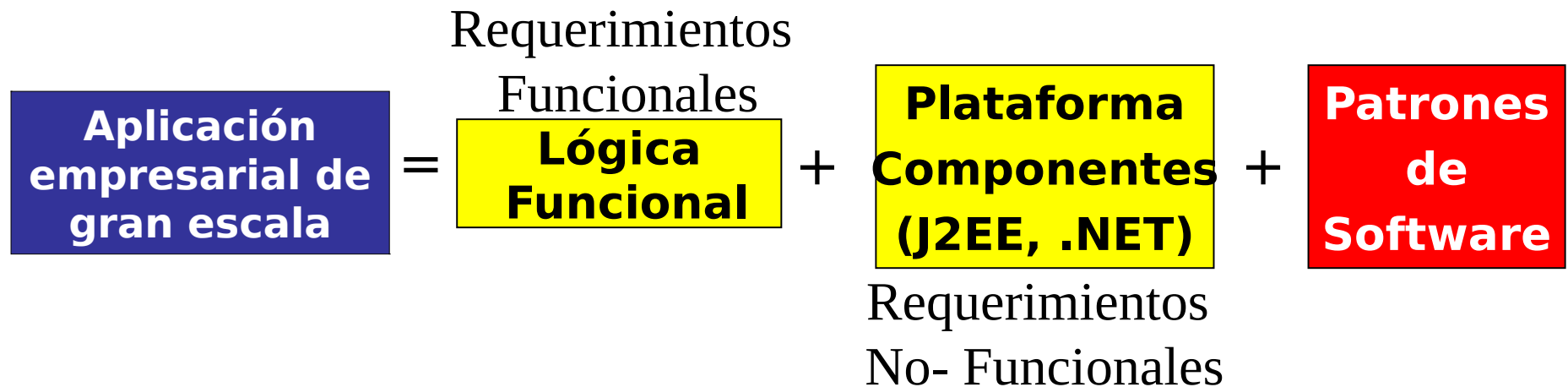


Motivación



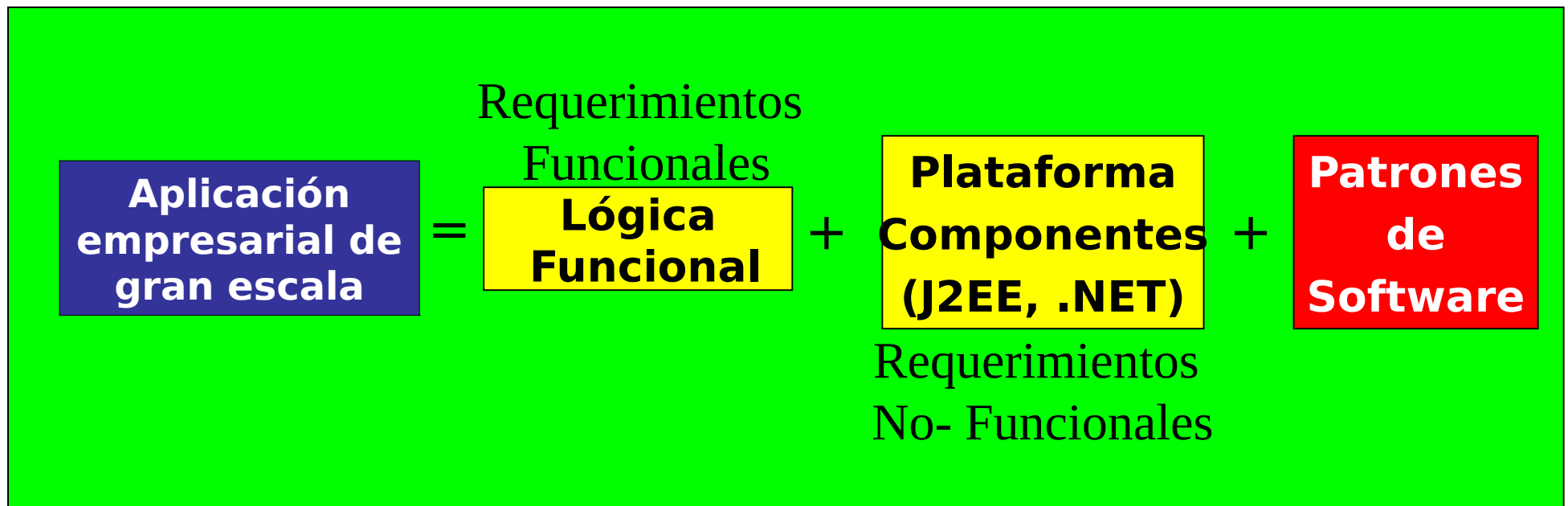


Motivación





Motivación



¿Será la formula del éxito?



Motivación

- Cómo comunicar experiencias?
- Cómo comunicar diseños?
 - Todos tenemos ideas diferentes de un mismo concepto.
- Emplear un lenguaje que sea comprensible por los desarrolladores, diseñadores, arquitectos.



Motivación

¿Cuál es éste lenguaje ?



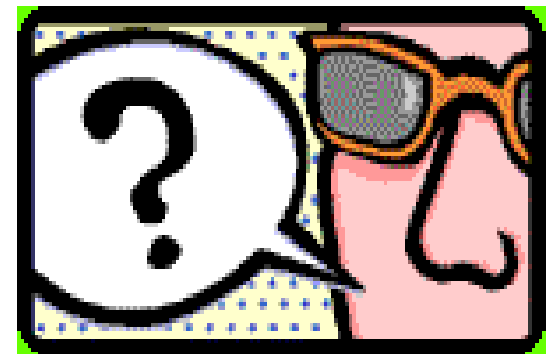
Motivación

Los patrones de software permiten establecer un lenguaje común para expresar y comunicar experiencias, diseños y buenas prácticas.

Motivación



Consultor en arquitectura
y plataformas



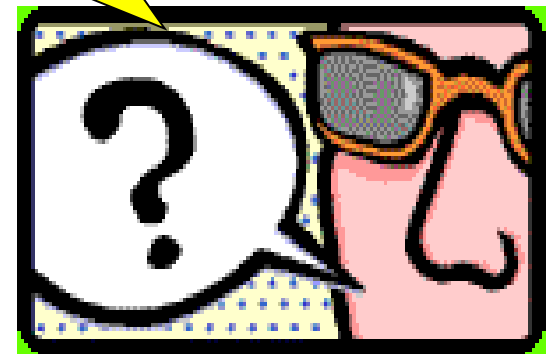
Desarrollador

Motivación

¿ Como hago para eliminar el tiempo de conexión a la base de datos en cada solicitud http, y tener así tiempos de respuesta más bajos?



Consultor en arquitectura
y plataformas



Desarrollador

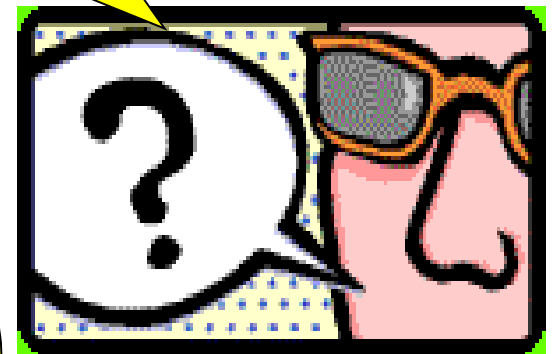
Motivación

¿ Como hago para eliminar el tiempo de conexión a la base de datos en cada invocación http, y tener así tiempos de respuesta más bajos?



Consultor en arquitectura
y plataformas

FACIL!!,



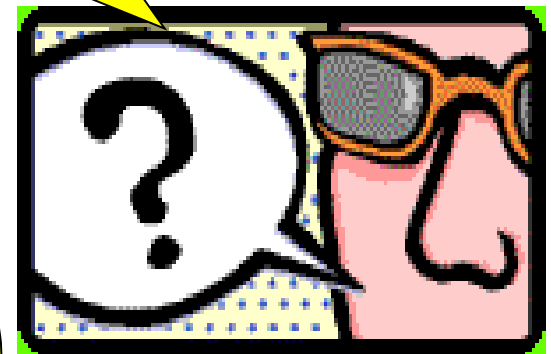
Desarrollador

Motivación



¿ Cómo hago para eliminar el tiempo de conexión a la base de datos en cada invocación http, y tener así tiempos de respuesta más bajos?

FACIL!!, Haga un singleton, que contenga un resource pooling que adapte a clases tipo abstract factory de conexiones, e implemente un observer para notificar el estado de la conexión



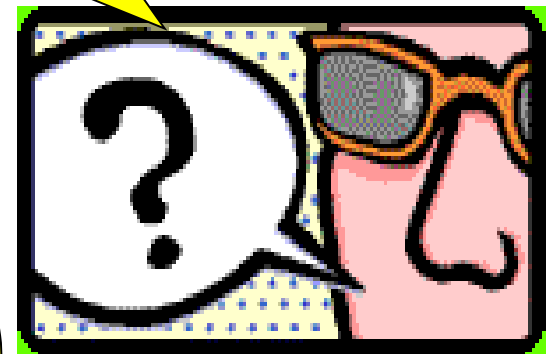
Desarrollador

Motivación



¿ Cómo hago para eliminar el tiempo de conexión a la base de datos en cada invocación http, y tener así tiempos de respuesta más bajos?

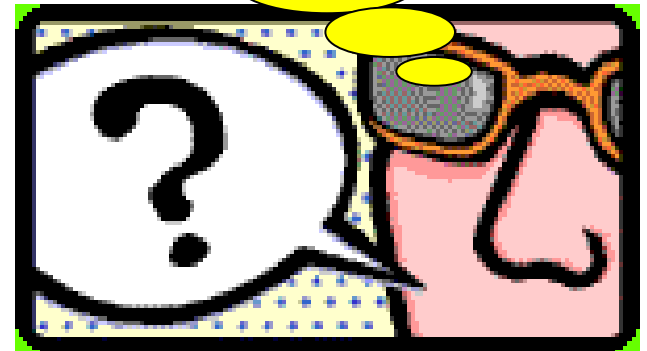
FACIL!!, Haga un singleton, que contenga un resource pooling que adapte a clases tipo abstract factory de conexiones, e implemente un observer para notificar el estado de la



Desarrollador

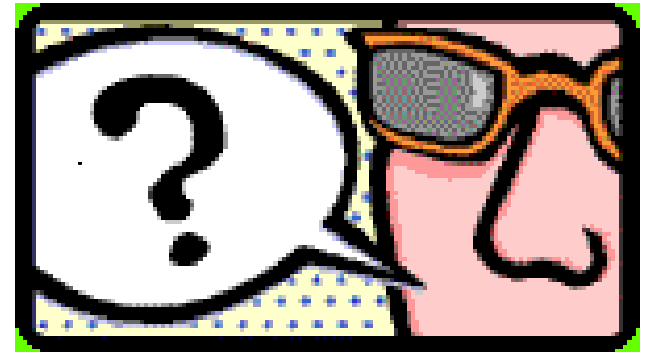
Motivación

Este tipo está loco!!!
No le he entendido nada.
¿De qué me hablaría?
¿En qué idioma me hablaría?



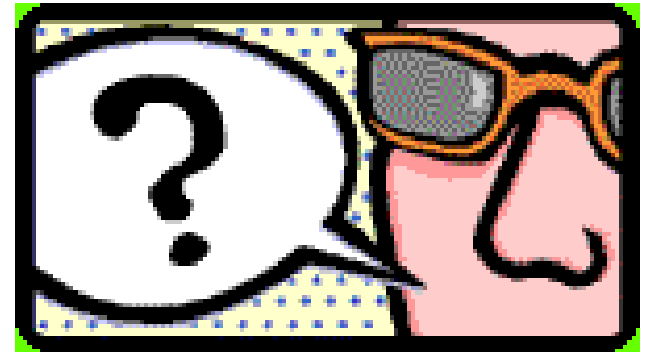
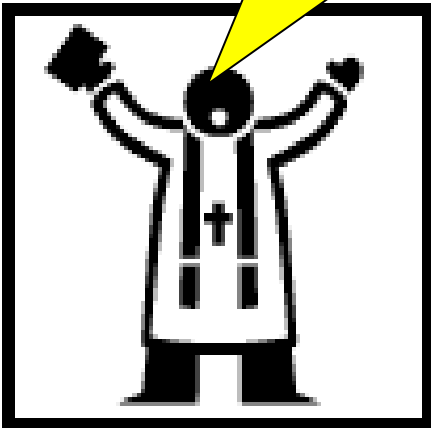
Motivación

¿Por qué estará tan callado?
¿Será que no está de acuerdo
conmigo en que sea un
abstract factory? Bueno, Le
voy a explicar la solución
usando un method factory?



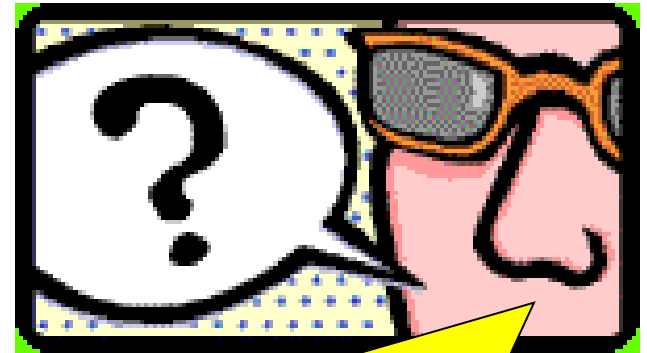
Motivación

Sí no estás de acuerdo con el abstract factory, no hay problema, con el method factory también funciona



Motivación

Sí no estás de acuerdo con el abstract factory, no hay problema, con el method factory también funciona



No, no, por el contrario creo que la mejor opción es usar un abstract de esos... Realmente, me ha quedado muy clara la solución... Tu siempre tan claro!!!



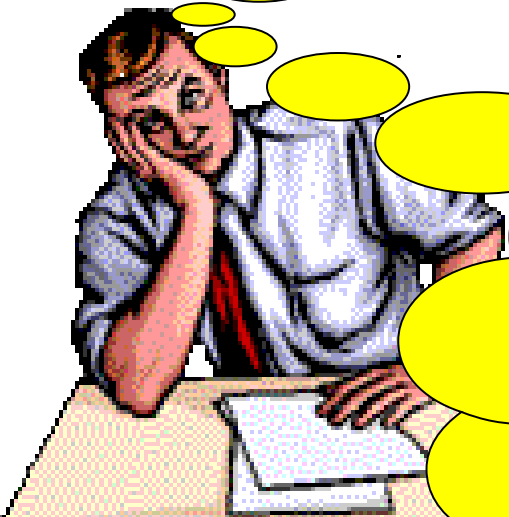
Motivación

Qué hago? Debo
encontrar una
solución pronto....





Motivación

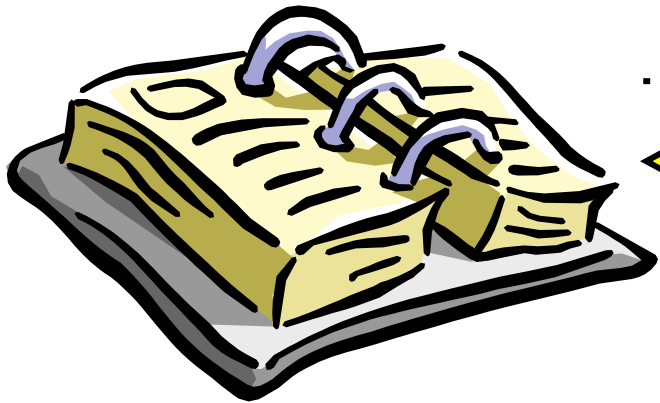


Qué hago? Debo encontrar una solución pronto....

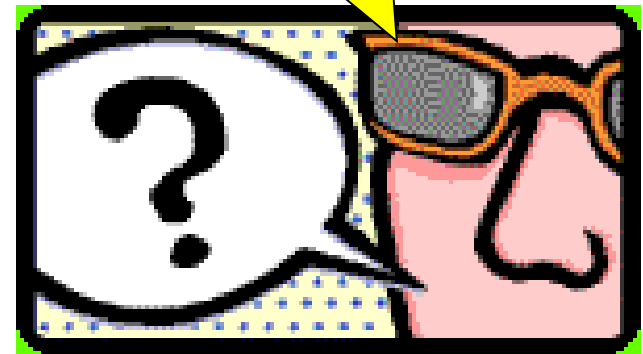
Uhy.... Ya sé!!! Iré al libro de "Best Practices for designing enterprise Systems", sí me costo U\$150 es por qué debe ser muy bueno

Motivación

(Buscando....)
**"Eliminar el tiempo de
conexión a la base de
datos en cada
invocación http"**

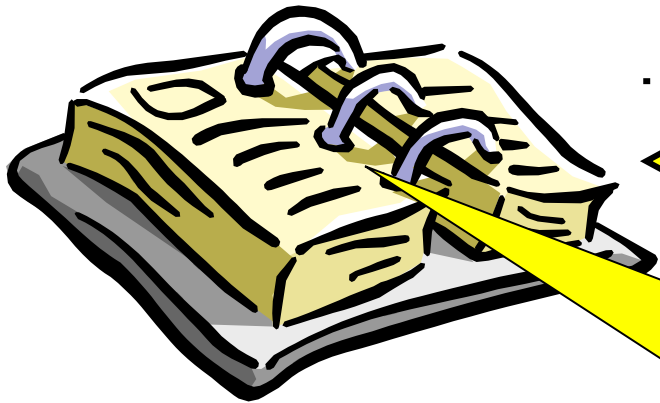


*Best practices for designing
enterprise systems*

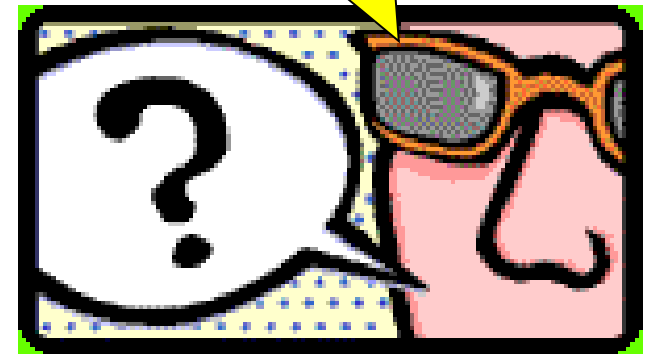


Motivación

(Buscando....)
"Eliminar el tiempo de
conexión a la base de
datos en cada
invocación http"



*Best practices for designing
enterprise systems*



Implemente un resource pooling singleton que adapte clases abstract factories, y se apoye en un modelo de notificación tipo call-back, tal como un observer remoto.



Motivación

Creo que ya no
entiendo el idioma
de la tecnología.





Motivación

Patrones de Software

“La lingua-franca de los desarrollares, arquitectos, diseñadores de software”



AGENDA

- ✓ Motivación
- Definición
- Categorías
- Lenguajes y Catálogos
- Conclusiones



Definición

- Qué es un patrón de software?

“Un patrón describe un problema de diseño recurrente, que surge en contextos específicos de diseño, y presenta un esquema genérico probado para la solución de este.

El esquema de la solución describe un conjunto de componentes, responsabilidades y relaciones entre de éstos, y formas en que dichos componentes colaboran entre sí.”

[Buschmann]



Definición

- Características de los patrones de software
 - No son inventados, provienen de la experiencia práctica.
 - Describen un grupo de componentes, sus relaciones, interacciones y responsabilidades.



AGENDA

- ✓ Motivación
- ✓ Definición
- Categorías
- Lenguajes y catálogos
- Conclusiones



Categorías de patrones

Patrones de Arquitectura
(MVC, Sistemas multicapas)

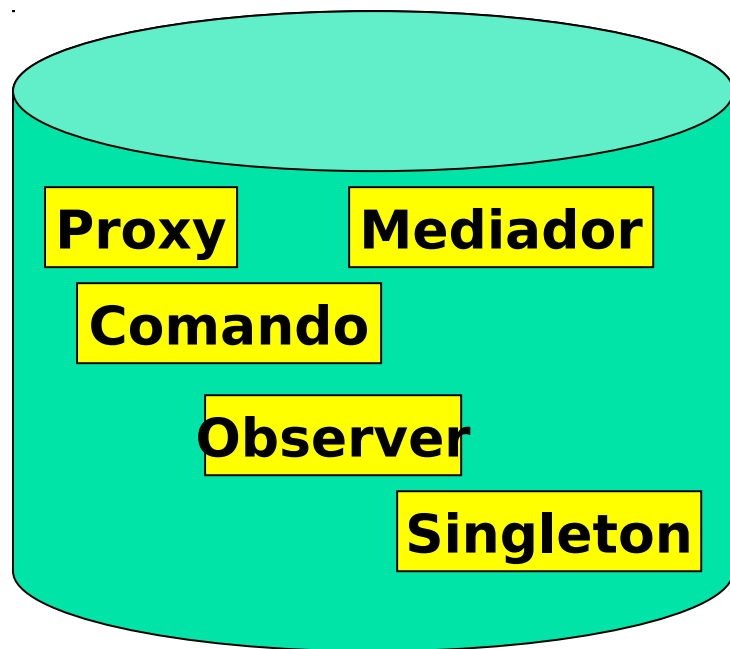


Patrones de diseño
(Independientes del lenguaje: Abstracto)

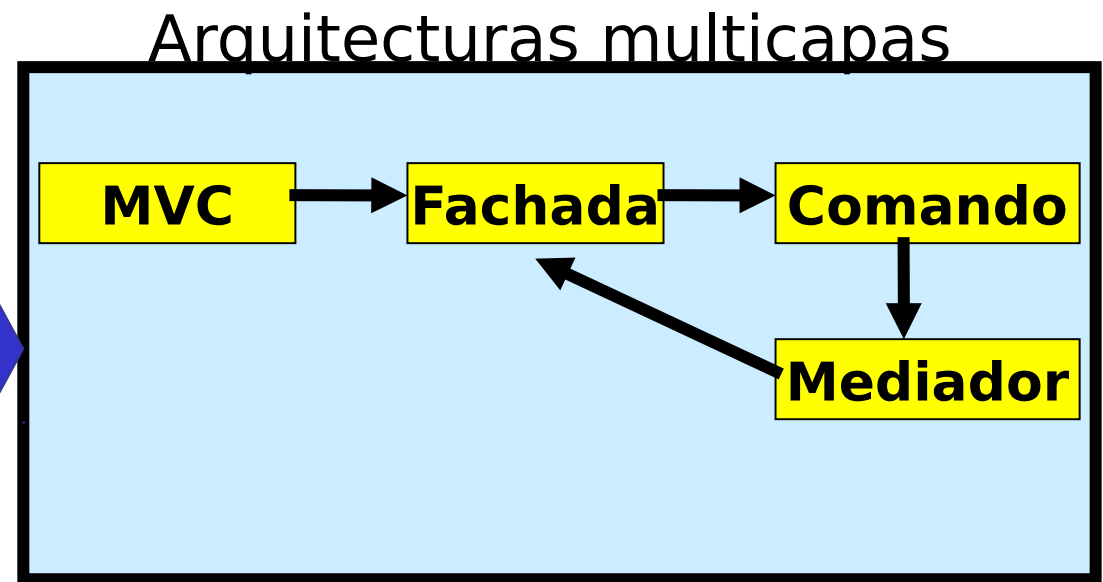
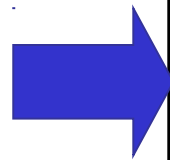


Patrones de programación
(Inherente al lenguaje : java, c++)

Lenguajes y Catálogos



Catálogo
(Domínio OO)



Lenguaje de Patrones



Lenguajes y Catálogos

- Catálogo
 - Se agrupan por paradigma, arquitectura, plataforma.
 - Catálogo GOF (Dominio OO)
 - Catálogo J2EE (Blueprints)
 - Catálogo .NET (.NET Patterns and Practices)
 - POSA (Patterns Oriented Software Architecture)
 - POSA 2 forma un lenguaje de patrones para abordar diseños y complejidades relacionadas con redes y concurrencia.



AGENDA

- ✓ Motivación
- ✓ Definición
- ✓ Categorías
- ✓ Lenguajes y catálogos
- Conclusiones



Conclusiones

- El diseño e implementación de aplicaciones empresariales, de gran escala, debe ser abordado bajo una continua incorporación de experiencias consignadas en patrones, guías de diseño y buenas prácticas.
- Una nueva “*lingua-franca*” ha llegado para quedarse en el mundo de los desarrolladores, arquitectos y diseñadores