



Proceso de evaluación para arquitecturas de software usadas en el sector empresarial (PEASSE)

Omar Salvador Gómez Gómez





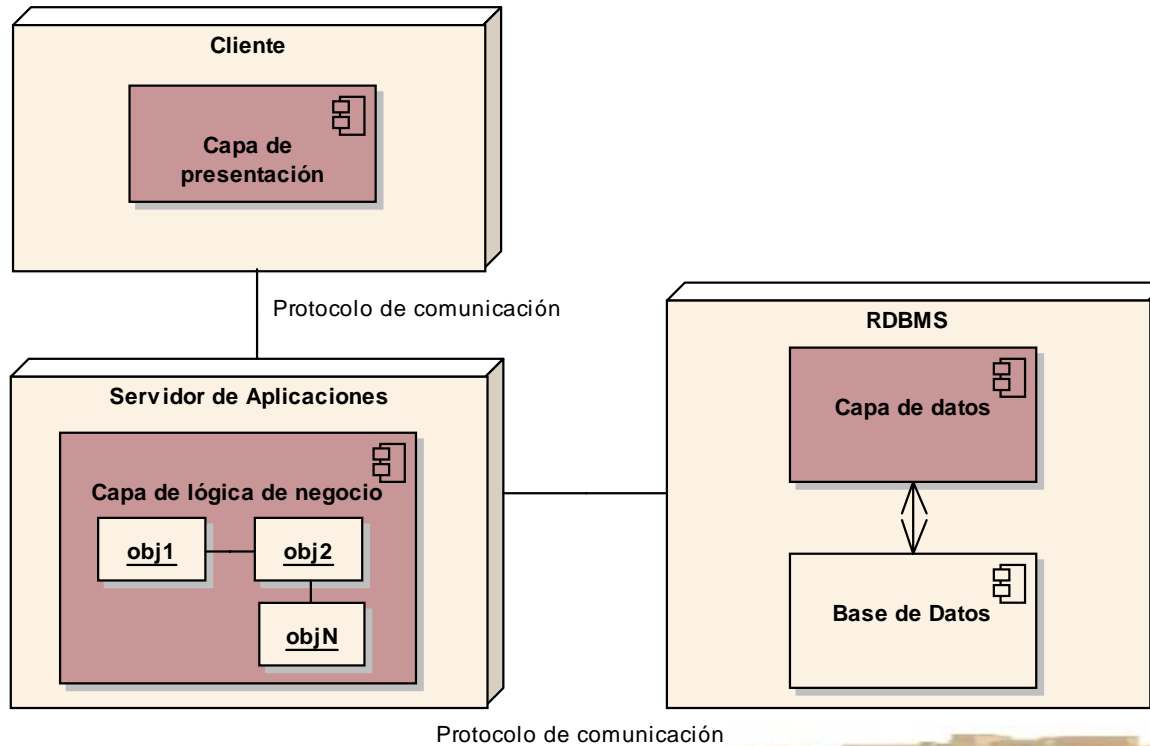
Agenda

- Introducción
- Problemática
- Antecedentes
- Solución propuesta
- Conclusiones



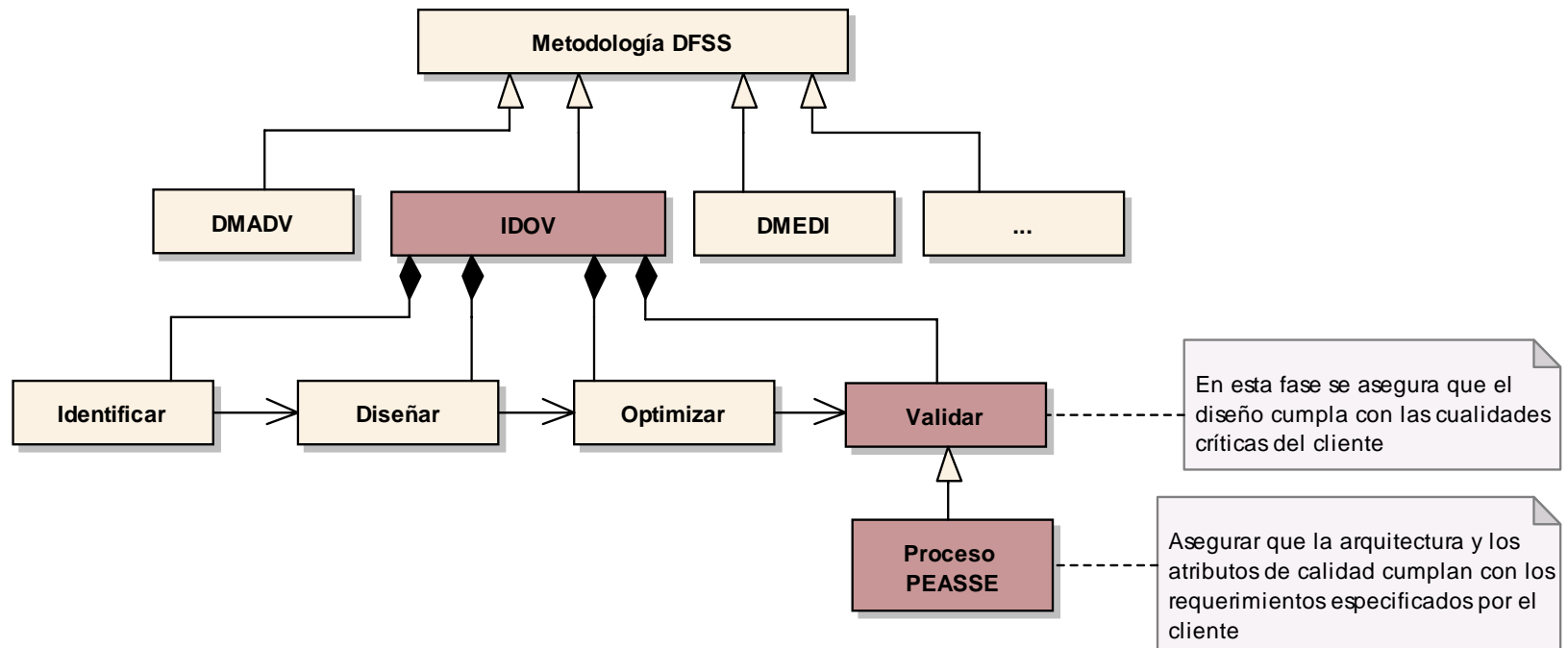
Introducción

- Arquitecturas de software usadas en el sector empresarial



Introducción

DFSS y PEASSE





Problemática

- El éxito o fracaso de un sistema de software esta determinado por su arquitectura
- ¿Cómo saben las empresas desarrolladoras de software si eligieron la arquitectura correcta para los sistemas de software que desarrollan?



Antecedentes

- Primeros esfuerzos realizados en evaluar arquitecturas de software:
 - Revisiones activas de diseño
- Algunos métodos de evaluación para AS usados en la actualidad y los atributos que predicen:
 - **SAAM**: Inicialmente creado para predecir facilidad de modificación
 - **ATAM**: Encuentra puntos sensitivos y Trade-offs entre atributos de calidad
 - **SNA**: Predice el grado de supervivencia
 - **ALMA**: Predice la facilidad de modificación





Antecedentes

- *Architecture-Level Modifiability Analysis (ALMA)*
 - ALMA predice el atributo de calidad facilidad de modificación
 - Los tres posibles objetivos de ALMA son:
 - Predecir el costo de mantenimiento
 - Evaluar riesgos
 - Comparar arquitecturas de software



Antecedentes

- Un escenario es una secuencia específica de pasos que involucra el uso o la modificación del sistema
- Pueden ser usados para evaluar atributos de calidad, ejemplos:
 - Escenarios que representen fallas para examinar la **disponibilidad** y **confiabilidad**
 - Escenarios que representen solicitudes de cambios para analizar la **facilidad de modificación**
 - Escenarios que representen amenazas al sistema para analizar la **seguridad**



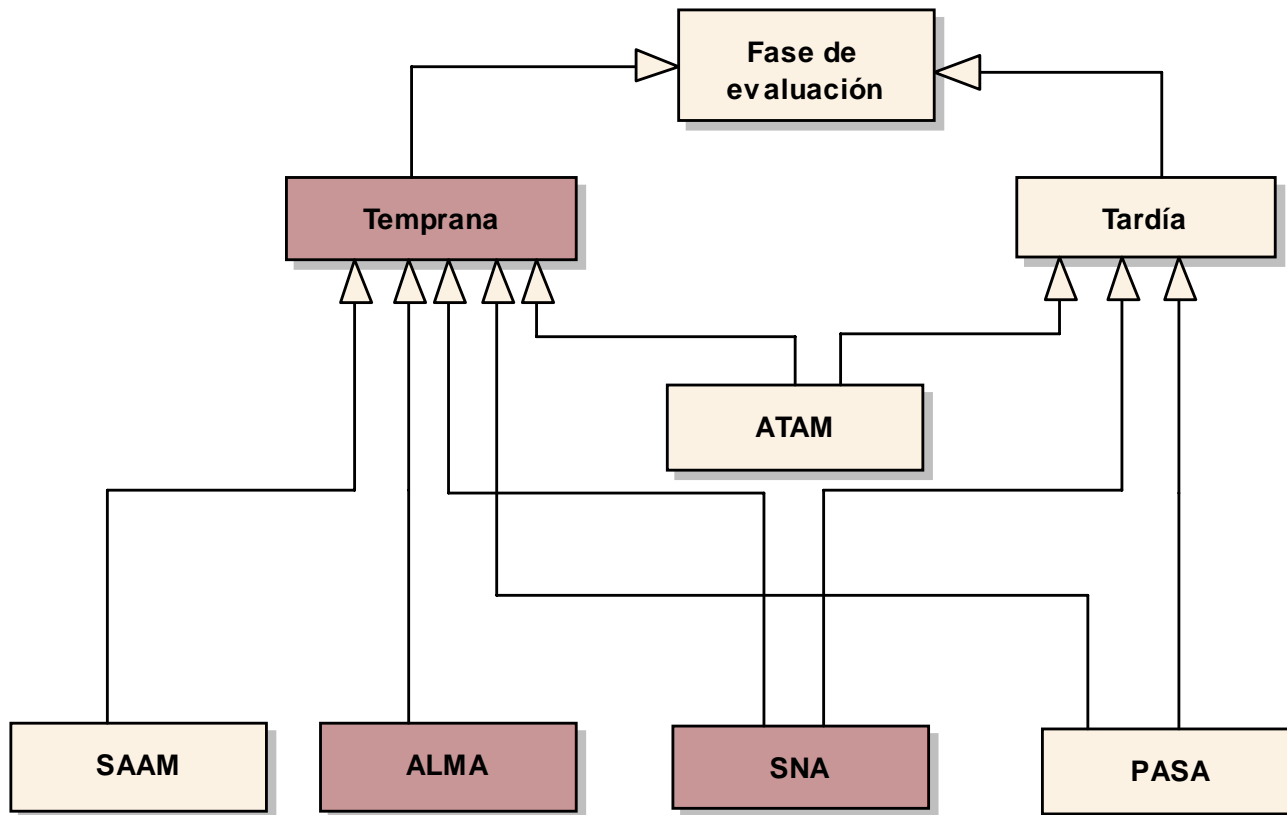
Antecedentes

- Tipos de escenarios:
 - **Directos.** No requieren modificaciones, la AS soporta directamente el escenario
 - **Indirectos.** Requieren modificaciones a la AS para satisfacer el escenario
 - **Exploratorios.** Cambios extremos al sistema
 - **Desarrollo.** Cambios anticipados al sistema



Antecedentes

Fases de evaluación



Antecedentes

- Patrones de diseño
 - Un patrón de diseño describe una estructura comúnmente recurrente de comunicación entre componentes que resuelve un problema de diseño general dentro de un contexto en particular
 - Patrones de diseño detectados en AS usadas en el sector empresarial:
 - MVC
 - *Singleton*
 - *Data Transfer Object*
 - *Facade*



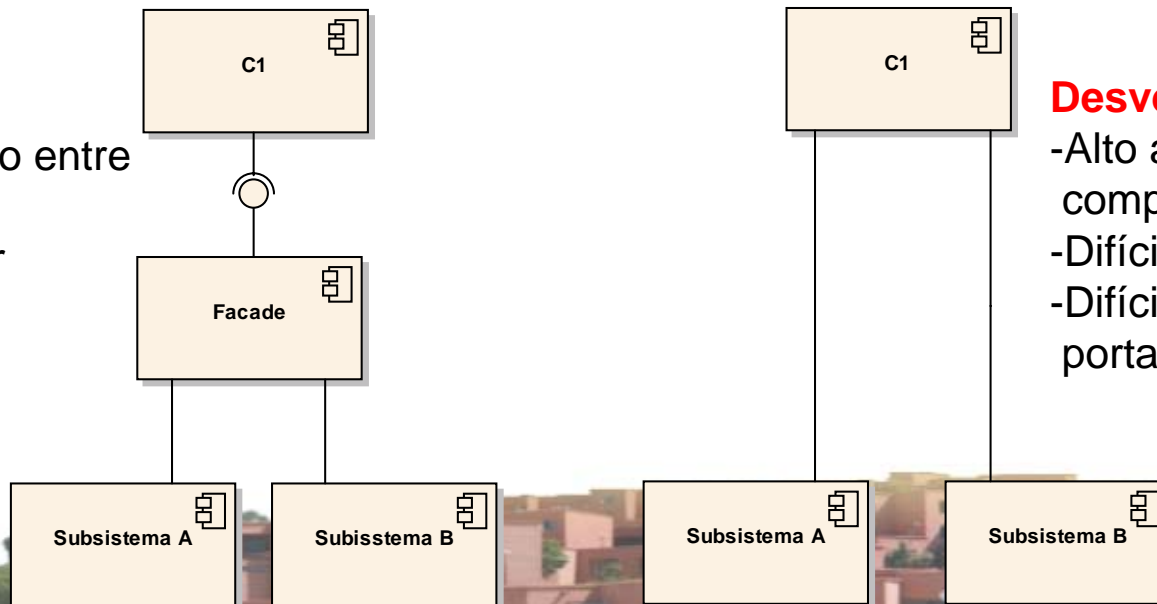
Antecedentes

- Patrón de diseño *Facade*

- **Intención:** Proporcionar una interfaz unificada para un conjunto de interfaces en un subsistema
- **Motivación:** Minimizar la comunicación y las dependencias entre subsistemas

Ventajas:

- Bajo acoplamiento entre componentes
- Fácil de modificar
- Fácil de lograr la portabilidad



Desventajas:

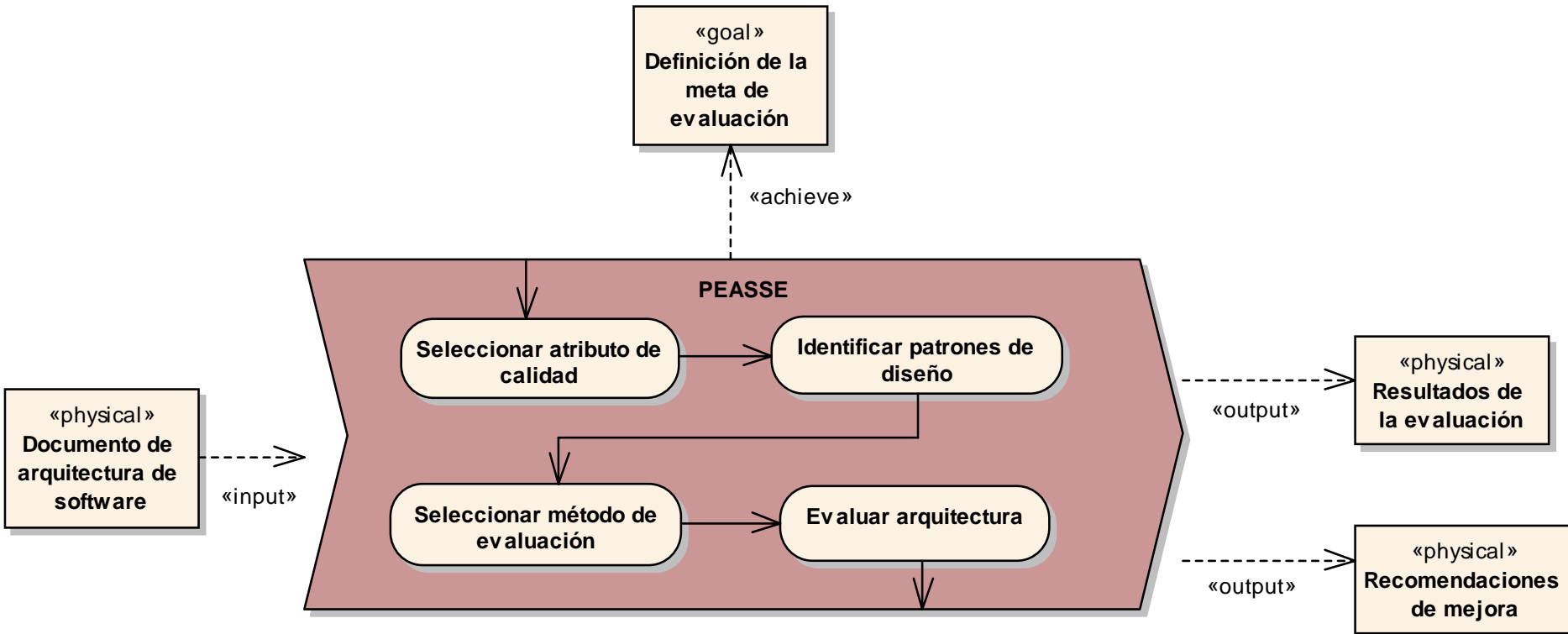
- Alto acoplamiento entre componentes
- Difícil de modificar
- Difícil de lograr la portabilidad



Solución propuesta



Proceso PEASSE

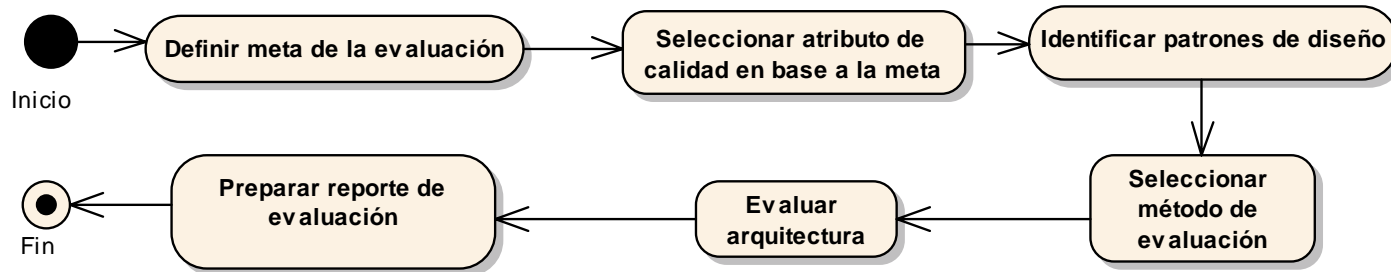




Solución propuesta



Pasos del proceso PEASSE



Conclusiones

- PEASSE puede ser utilizado en la fase de validación de la metodología DFSS conocida como IDOV
- Usado para evaluar AS en el sector empresarial
- Identifica patrones de diseño
- Utiliza ALMA y SNA como métodos de evaluación, sin embargo es flexible para utilizar otros métodos de evaluación de AS
- Concebido para ser usado en fases de evaluación tempranas
- Utiliza técnicas de evaluación cualitativas





Referencias

- Antony, Jiju and Ricardo Bañuelas Coronado. "Design for Six Sigma." *IEE Manufacturing Engineering*, vol. 81, no. 1, pp. 119-21, 2002.
- Len Bass, Paul Clements and Rick Kazman. *Software Architecture in Practice*. Edited by SEI Series In Software Engineering. second ed: Addison Wesley, 2003.
- Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. *Design Patterns Elements of Reusable Object-Oriented Software*. Edited by Addison-Wesley professional computing series: Addison Wesley, 1995.
- Bosch, Jan and Peter Molin. "Software Architecture Design: Evaluation and Transformation." Paper presented at the *Proceedings of IEEE Engineering of Computer Based Systems Symposium (ECBS '99)* 1999.
- Parnas, David L. and David M. Weiss. "Active Design Reviews: Principles and Practices." Paper presented at the *Proceedings of 18th International Conference on Software Engineering* 1985.
- Kazman, Rick, Mark Klein, Mario Barbacci, Thomas Longstaff, Lipson Howard F. and S. Jeromy Carriere. "The Architecture Tradeoff Analysis Method." Paper presented at the *Proceedings of ICECCS*, Monterey, CA 1998.
- Williams, Lloyd G. and Connie U. Smith. "Pasa SM: A Method for the Performance Assessment of Software Architecture." Paper presented at the *Proceedings of the 3rd Workshop on Software Performance*, Rome, Italy 2002.
- Bengtsson, PerOlof, Nico Lassing and Jan Bosch, Vliet, Hans van. "Architecture-Level Modifiability Analysis (ALMA)." *The Journal of Systems and Software* , vol. 69, pp. 129-147, 2004.
- Mead, Nancy R., Robert J. Ellison, Richard C. Linger, Thomas Longstaff and John McHugh. "Survivable Network Analysis Method." CMU/SEI, 2000.
- Paul Clements, Rick Kazman and Mark Klein. *Evaluating Software Architectures*. Edited by The SEI Series in Software Engineering: Addison Wesley, 2002.
- Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad and Stal Michael. *Pattern Oriented Software Architecture a System of Patterns*. Edited by Wiley series in software design patterns, vol. 1, John Wiley & Sons Ltd, 1996.



¿Preguntas?

