



Facultad de Ingeniería
Universidad de Cuenca
Grado en Ingeniería de Sistemas
Curso 2020-2021

Calidad de Software

Tema: Sistemas Híbridos

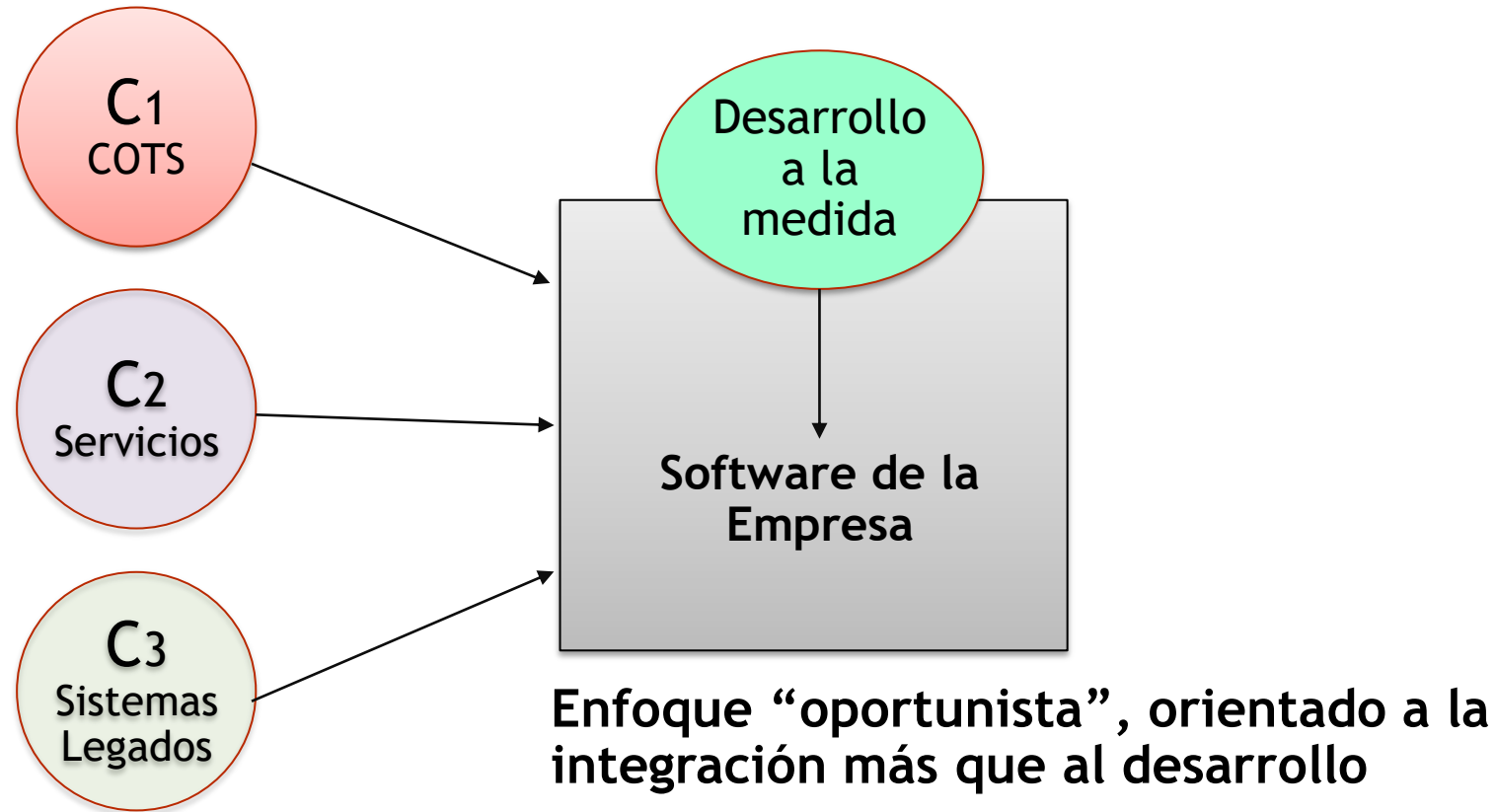
Ing. Priscila Cedillo O. PhD.
Departamento de Ciencias de la Computación
Universidad de Cuenca, Ecuador
email: priscila.cedillo@ucuenca.edu.ec

Sistemas Híbridos - Generalidades

- n La mayoría de sistemas de software hoy en día se pueden construir mediante la integración de componentes de diversa naturaleza.
 - Comerciales
 - Código Libre
 - Componentes
 - Legados
 - Etc.
- n Así se forman arquitecturas híbridas



Sistemas Híbridos



Diversos componentes de proveedores externos a la organización que se integran con algún software hecho a medida.

Sistemas Híbridos

- n Los componentes de software utilizados en este tipo de enfoques arquitectónicos incluyen componentes desarrollados por terceros, generalmente conocidos como componentes “*Off-The-Shelf*” (OTS)
 - Componentes comerciales (COTS)
 - Componentes gratuitos y de código abierto (FOSS)
 - Servicios Web
- n Software desarrollado a la medida
- n Sistemas legados



Componentes COTS (Commercial Off-The-Shelf)

- n “Un componente es una **unidad** de composición de aplicaciones software, que posee un **conjunto de interfaces** y un conjunto de **requisitos**, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio” Clemens Szyperski
- n El adjetivo COTS se refiere a **un tipo particular de componente**, caracterizado por:
 - ser de índole comercial,
 - generalmente de grano grueso y
 - de bajo coste,
 - que es adquirido, seleccionado, probado, validado e integrado por desarrolladores de un sistema software basado en componentes para satisfacer ciertas necesidades del sistema, a partir de unos requisitos específicos



K. C. Wallnau, D. Carney, and B. Pollack.
How COTS software affects the design of COTS-intensive systems.
SEI Interactive, 1998.

Beneficios del Desarrollo de Software basado en Componentes

- n Reutilización de Software
- n Simplifica las pruebas
- n Simplifica el mantenimiento
- n Mayor calidad
- n Ciclos de desarrollo más cortos
- n Mejor ROI
- n Funcionalidad mejorada
- n Independencia SW y HW



Desventajas del Uso de Componentes

- n No existe control sobre las nuevas versiones
- n Costos asociados
 - Costos de licenciamiento
 - Costos de integración
 - Costos de soporte
- n Licencias y propiedad intelectual
- n Dependencia del fabricante
- n La integración no es trivial



Integración / Adaptación / Parametrización

- n Tipos de parametrización
 - Personalización
 - Rendimiento
 - Simplemente poner el componente a trabajar
- n La mayoría de componentes requieren algún tipo de parametrización:
 - MS-Office: un par de tardes
 - Sistemas ERP: varios meses

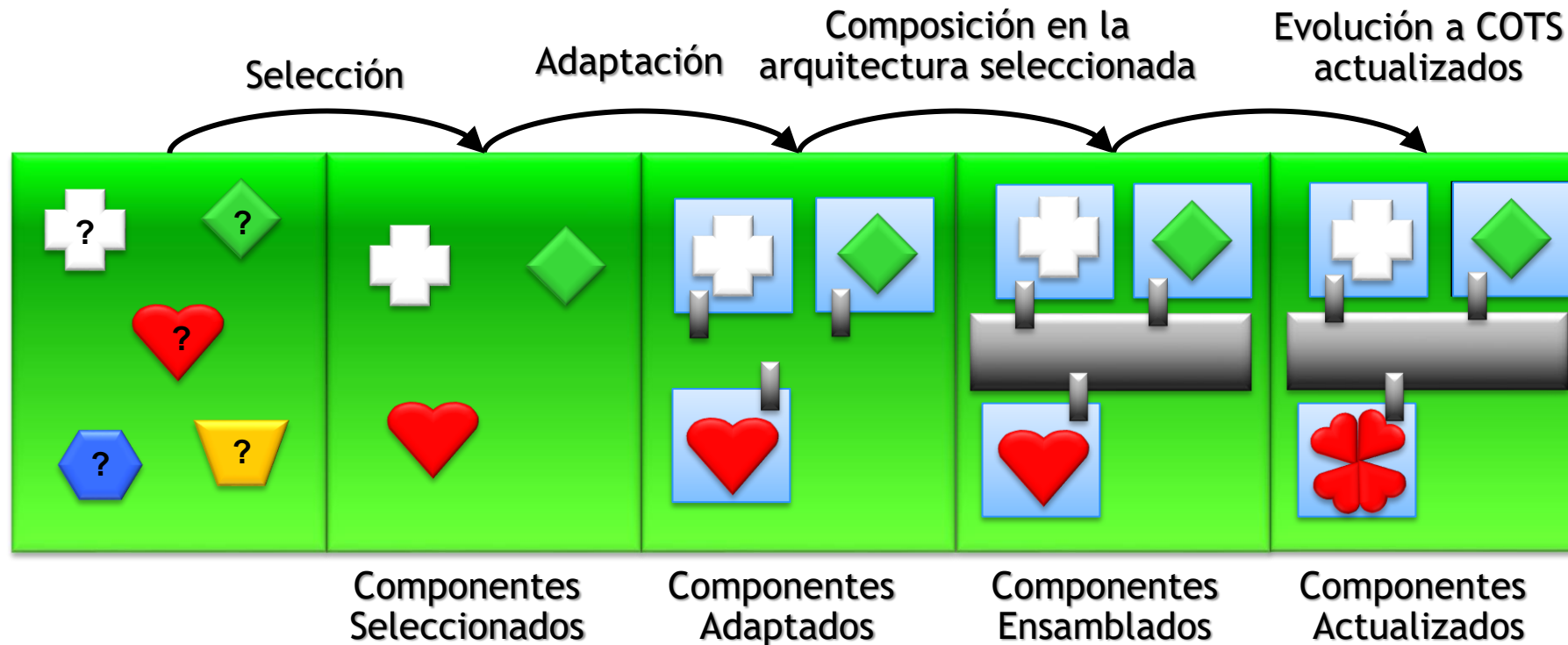


Dificultades

- n Especificación de requisitos
- n Selección de componentes adecuados
- n Adaptación e integración a una arquitectura común
- n Identificación de las necesidades estratégicas para las cuales es demandado el software
- n Agrupación de servicios relacionados en dominios atómicos que estructuran la arquitectura genérica del sistema y describen la funcionalidad mínima cubierta por cada componente



Ciclo de Vida y Retos



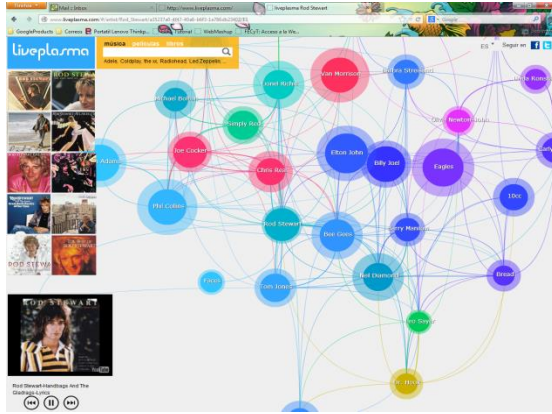
- Mercado en continua evolución:
 - Cientos de nuevos productos y actualizaciones
 - Nuevos productos no son idénticos a los anteriores
 - Reemplazo de componentes puede ser muy difícil
 - Reentrenamiento y testeo pueden ser requeridos

Mashups

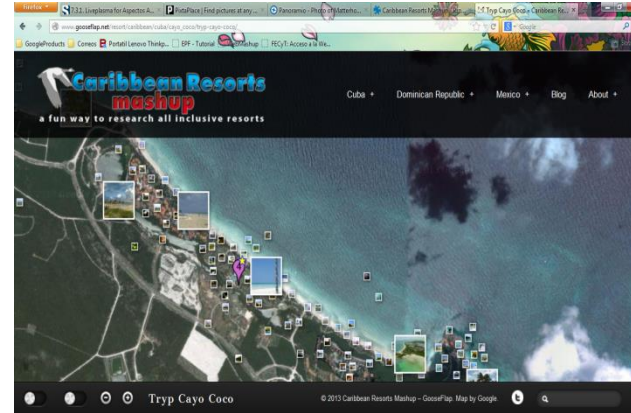
- n El término mashup tiene su origen en el dominio musical refiriéndose a artistas que mezclan algunas piezas de música, usualmente desde diferentes estilos musicales, en un simple registro.
- n Los mashups son aplicaciones desarrolladas para integrar contenido y funcionalidad cuya fuente es la web. Integran elementos heterogéneos disponibles en la web tales como RSS | Atom feeds, Web Services, APIs, contenido traído desde sitios de terceros o widgets (tales como Google maps).



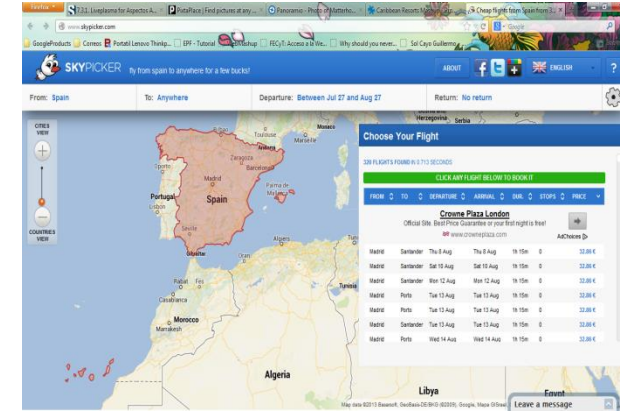
Aplicación del Método: Casos de Estudio



Liveplasma



Caribbean Resorts



Skypicker

- Se han escogido del repositorio de Programmable Web.
- Pertenecen a la lista de los Mashups más populares.
- Presentan características que permiten mostrar el método de evaluación más claramente, por la variedad de recursos que los componen.

Ejercicio en Clase 2

- n Explicar el concepto de Orquestación vs Coreografía en la composición de servicios
- n Investigar 1 caso de sistema híbrido y comentar lo siguiente:
 - Servicios o componentes integrados
 - Forma de integración
 - Pros y Cons de las fuentes utilizadas
- n Buscar 2 mashups y analizar cómo se ha realizado su composición



Bibliografía

- n <https://msdn.microsoft.com/es-es/library/bb972268.aspx>
- n J.P. Carvallo, X. Franch, “*Descubriendo la arquitectura de sistemas de software híbridos: Un enfoque basado en modelos i**” E-prints UPC -Universitat Politecnica de Catalunya-. Recuperada en Abril 10, 2012, del sitio Web temoa : Portal de Recursos Educativos Abiertos (REA) en <http://www.temoa.info/es/node/139907>
- n Carvallo J. P., Presentaciones, 2014-2015
- n L. Fuentes, J.M. Troya, A. Vallecillo, *Desarrollo de Software Basado en Componentes*, Universidad de Málaga, España en <http://www.lcc.uma.es/~av/Docencia/Doctorado/tema1.pdf>
- n P. Cedillo, A. Fernandez, E. Insfran, S. Abrahao, *Quality of Web Mashups: A Systematic Mapping Study*, ICWE 2013 International Workshops ComposableWeb, QWE, MDWE, DMSSW, EMotions, CSE, SSN, and PhD Symposium, Aalborg, Denmark, July 8-12, 2013. http://link.springer.com/chapter/10.1007/978-3-319-04244-2_8

