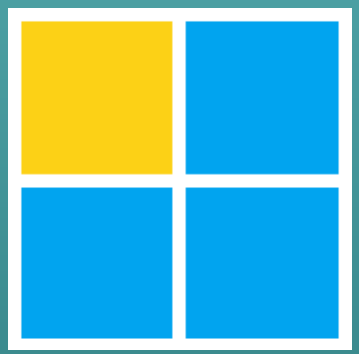
Desarrollo de Software Basado en Componentes

Diego Cardozo



Agenda

1. Motivación
2. ¿Qué es el DSBC?
3. Componentes vs Objetos
4. Aplicaciones conocidas
5. ¿Como encajan los CMS?
6. Bajar a tierra conceptos
   - WebMatrix
   - OrchardCMS

Advertencia: Esta es una charla de Ingeniería de Software

Motivación (1)

Si General Motors evolucionara como la industria de la computación, hoy tendríamos autos que saldrían $100 consumiendo 1 litro de combustible cada 1 millón de kilómetros.

Motivación (2)

Pero...

- Chocarías al menos 2 veces al día
- Comprar un auto nuevo cada vez que se cambia una señal de tránsito
- Cada vez que comprás un modelo nuevo, tenés que aprender a manejar nuevamente.

¿Conclusión?

- Otras industrias utilizan enfoques distintos
- Obtienen muy buenos resultados en áreas donde nosotros tenemos problemas.
  - Automotriz: procesos ágiles, producto complejo que se construye de forma rápida.
  - Construcción: tienen costos asociados al cambio mucho mayores que nosotros, pero se arreglan mejor.

¿Qué es lo que otras industrias tienen en común?

Trabajan utilizando componentes

- Mayor nivel de re-utilización
- Simplifica las pruebas
- Simplifica el mantenimiento
- Mayor calidad

Componentes desarrollados por terceros

- Ciclos de desarrollo mas cortos
- Mejor ROI

¿Que es un componente en SW?

"Un componentes es una unidad de composición de aplicaciones de software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistemas y compuesto con otros componentes de forma independiente, en tiempo y espacio."

Szyperski, 1998

7 Criterios

1. Puede ser usado por otros elementos de SW
2. Puede ser usado por los clientes sin la necesidad de intervención del desarrollador (CMS).
3. Incluye las especificaciones de todas las dependencias.
4. Incluye documentación de las funcionalidades que ofrece.
5. Se puede entender su funcionamiento en base a las especificaciones.
6. Se puede acoplar a otros componentes.
7. Puede ser incorporado a un sistema de manera suave y rápida.

Desventajas

- Clarividencia: diseñar un componente sin conocer quien lo utilizará.
- Particularización: es difícil "customizar" un componente sin acceder a su implementación.
- Falta de soporte: una vez que un componente es creado y sellado, ¿como y quien lo mantiene?

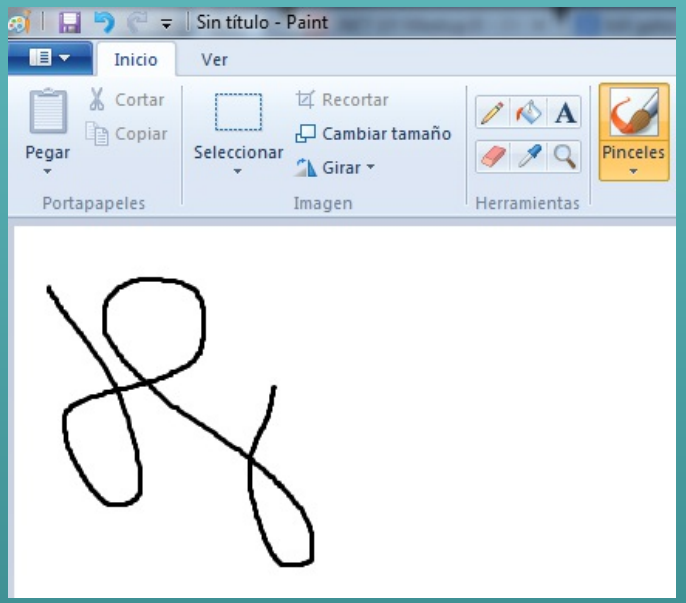
Componentes vs Objetos

	Objeto	Componente
Polimorfismo	Si	No
Instanciación	Algo tardía	Lo mas tarde posible
Encapsulación	Algo	Real y forzada
Herencia	Si	Por interfaz y reuso de binarios

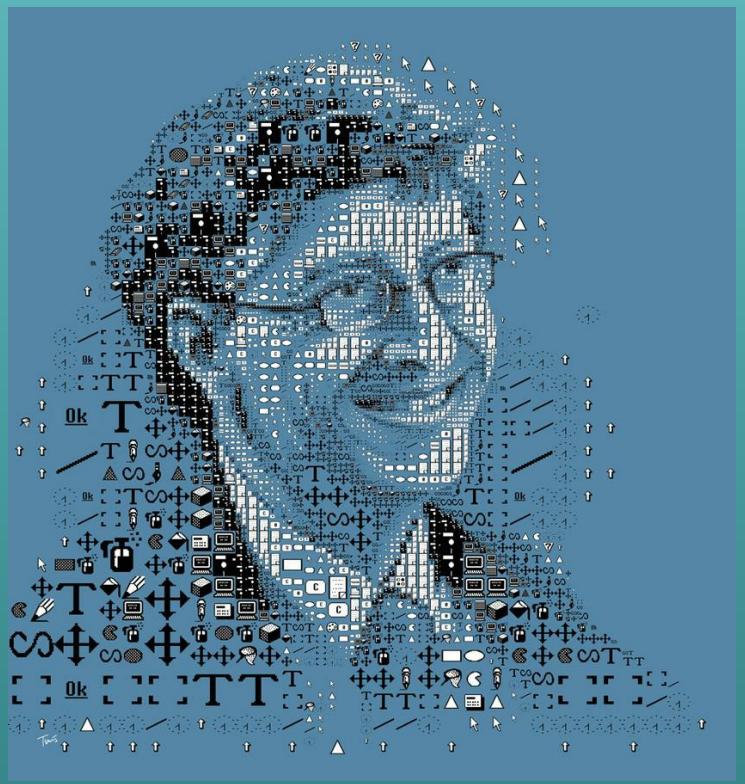
Aplicaciones conocidas

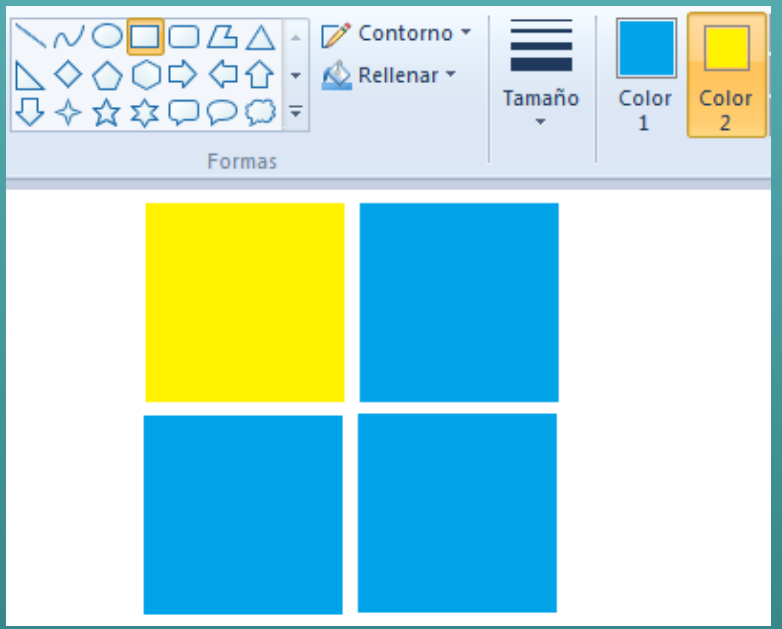
- Bajo nivel: COM (Component Object Model)
  - Comunicación de procesos en distintos lenguajes
  - Precursores de .NET
- CORBA
  - Estándar creado por el OMG
  - Usado dentro del mundo Java
- Otras aplicaciones
  - Programación para diseño gráfico

Modelo del pintor



Modelo de composición





Basta de filosofía...

Ejemplo concreto:

- Crear un sitio web para la comunidad de .NET Uruguay
- Incluír un foro

¿Por qué con un CMS?

- Se basan en la idea de DSBC.
- Tanto la estructura como el contenido son componentes
  - Páginas, imágenes, posts, widgets, módulos, etc.
- Cuentan con una comunidad y un catálogo de componentes existentes enorme.

Microsoft WebMatrix (1)

- Ambiente de desarrollo gratuito de Microsoft
- Liviano: 40MB recién instalado comparado con varios gigas de VS
- Pensado para la nube
- Soporta varios lenguajes y plataformas
  - ASP.NET
  - PHP
  - Node.js
- Integración con GIT y TFS

Microsoft WebMatrix (2)

- VS se siente como un laboratorio para crear
- WebMatrix es ideal para DSBC
  - Proporciona componentes:
    - Galería de frameworks
    - Incorpora NuGet package manager
    - Extensiones útiles
  - Aisla al framework de los componentes

OrchardCMS (1)

- Completamente Open Source
- Creciendo rápidamente
- Arquitectura MVC
- Eso significa que todos los componentes siguen una arquitectura MVC.
  - Crear nuevos componentes es sencillo
  - Los componentes existentes son fáciles de comprender y extender

Orchard CMS (2)

Componentes

- Contenido
- Módulo
  - Features
- Temas
- Templates
  - Formas (shapes)
- Widgets
- Usuarios, roles, permisos

Recursos

- Charla y código:
  - github.com/diegocard/CBSD-presentation
- DSBC en MSDN:
  - http://msdn.microsoft.com/es-es/library/bb972268.aspx#ref07back
- Excelente presentación relacionada:
  - http://www.slideshare.net/ulicruz/desarrollo-de-software-basado-en-componentes
- Paper
  - Component-Based Software Engineering – New Paradigm of Software Development (Crnkovic, Larsson)

Fin

"La Revolución Industrial del software está finalmente ante nosotros. La especialización de recursos, estándares para partes intercambiables, y herramientas de ensamblaje de última generación han sido usadas en otras industrias por cientos de años para acelerar el desarrollo de productos altamente complejos.
A pesar de su ubicuidad, la aplicación de estos conceptos a la industria moderna del software solamente ha empezando."

Bill Gates, 1997