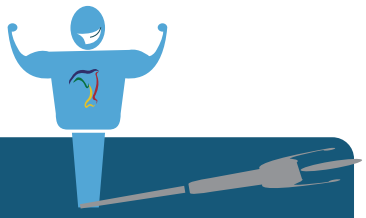
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

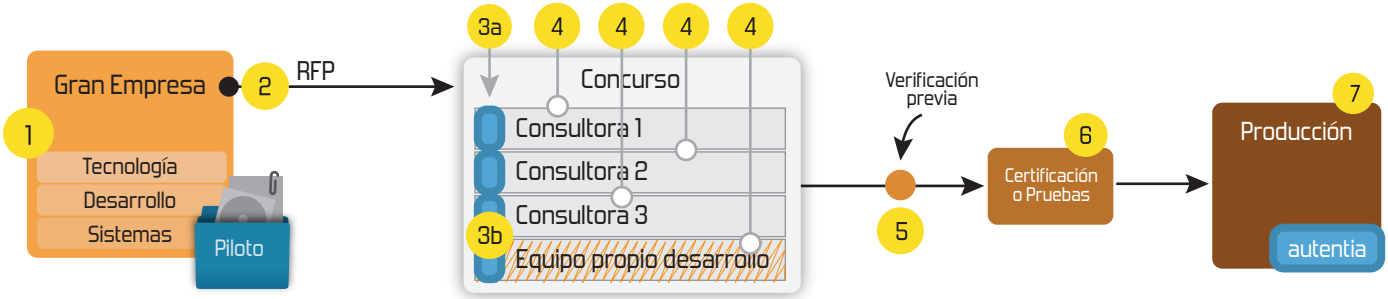
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

- Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery
- Gestor portales (Liferay)
- Gestor de contenidos (Alfresco)
- Aplicaciones híbridas
- Control de autenticación y acceso (Spring Security)
- UDDI
- Web Services
- Rest Services
- Social SSO
- SSO (Cas)
- JPA-Hibernate, MyBatis
- Motor de búsqueda empresarial (Solr)
- ETL (Talend)
- Dirección de Proyectos Informáticos.
- Metodologías ágiles
- Patrones de diseño
- TDD
- Tareas programadas (Quartz)
- Gestor documental (Alfresco)
- Inversión de control (Spring)
- BPM (jBPM o Bonita)
- Generación de informes (JasperReport)
- ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Samuel Martín Gómez-Calcerrada

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Ver todos los tutoriales del autor

Fecha de publicación del tutorial: 2014-10-23

Tutorial visitado 748 veces Descargar en PDF

Open-Closed Principle / Principio Abierto-Cerrado

0. Índice de contenidos.

1. Introducción

Este es el segundo tutorial de la batería de tutoriales S.O.L.I.D.
El principio Abierto-Cerrado es probablemente el más fácil de aplicar de los 5.
Bertrand Meyer es el responsable de acuñar el término open/closed principle en su libro “Object Oriented Software Construction” de 1988 aunque fue popularizado junto con los otros principios del acrónimo SOLID por Martin Fowler una década más tarde.

2. Open-Closed Principle / Principio Abierto-Cerrado

La definición de OCP es:
Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification.
La idea de este principio es que las clases están abiertas a la extensión pero cerradas a la modificación, lo que significa que ante peticiones de cambio en nuestro programa, hay que ser capaces de añadir funcionalidad sin modificar la existente (siempre que sea posible).

Por lo general, en un mal diseño, modificar una funcionalidad durante el ciclo de vida, suele conllevar una cadena de cambios en módulos dependientes, y este efecto puede propagarse en cascada si el código está muy acoplado.

La forma más común de seguir el principio OCP es usar interfaces o clases abstractas de las que dependen implementaciones concretas.
De esta forma puede cambiarse la implementación de la clase concreta manteniéndose la interfaz intacta.

3. Ejemplo

Tenemos una empresa que desde sus comienzos ha estado vendiendo agua embotellada, y su programa informático tenía esta forma.

Últimas Noticias

» Curso JBoss de Red Hat

» Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris

» Portales, gestores de contenidos documentales y desarrollos a medida

» Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer

» Screencasts de programación narrados en Español

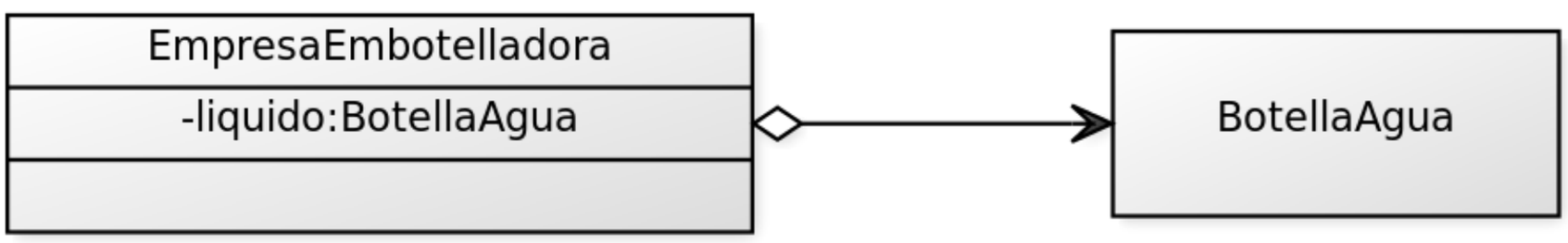
Histórico de noticias

Últimos Tutoriales

» Creación paso a paso de un webscript Alfresco

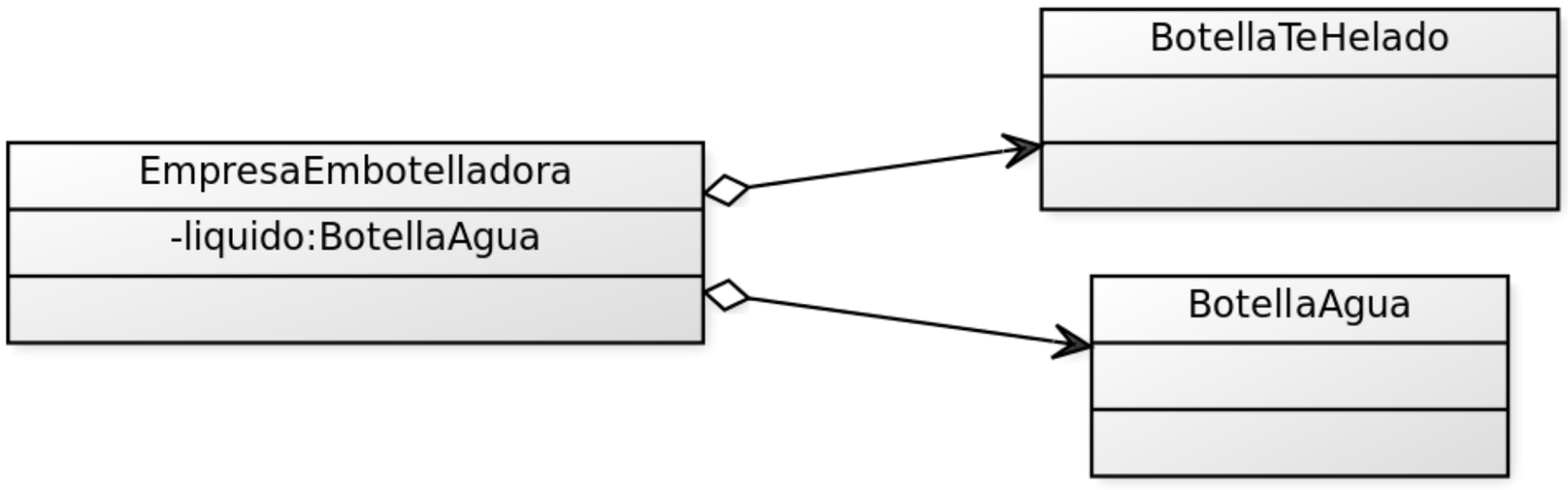
» Integración de MonkeyTalk en iOS

» Soporte de Redis con Spring: RedisTemplate



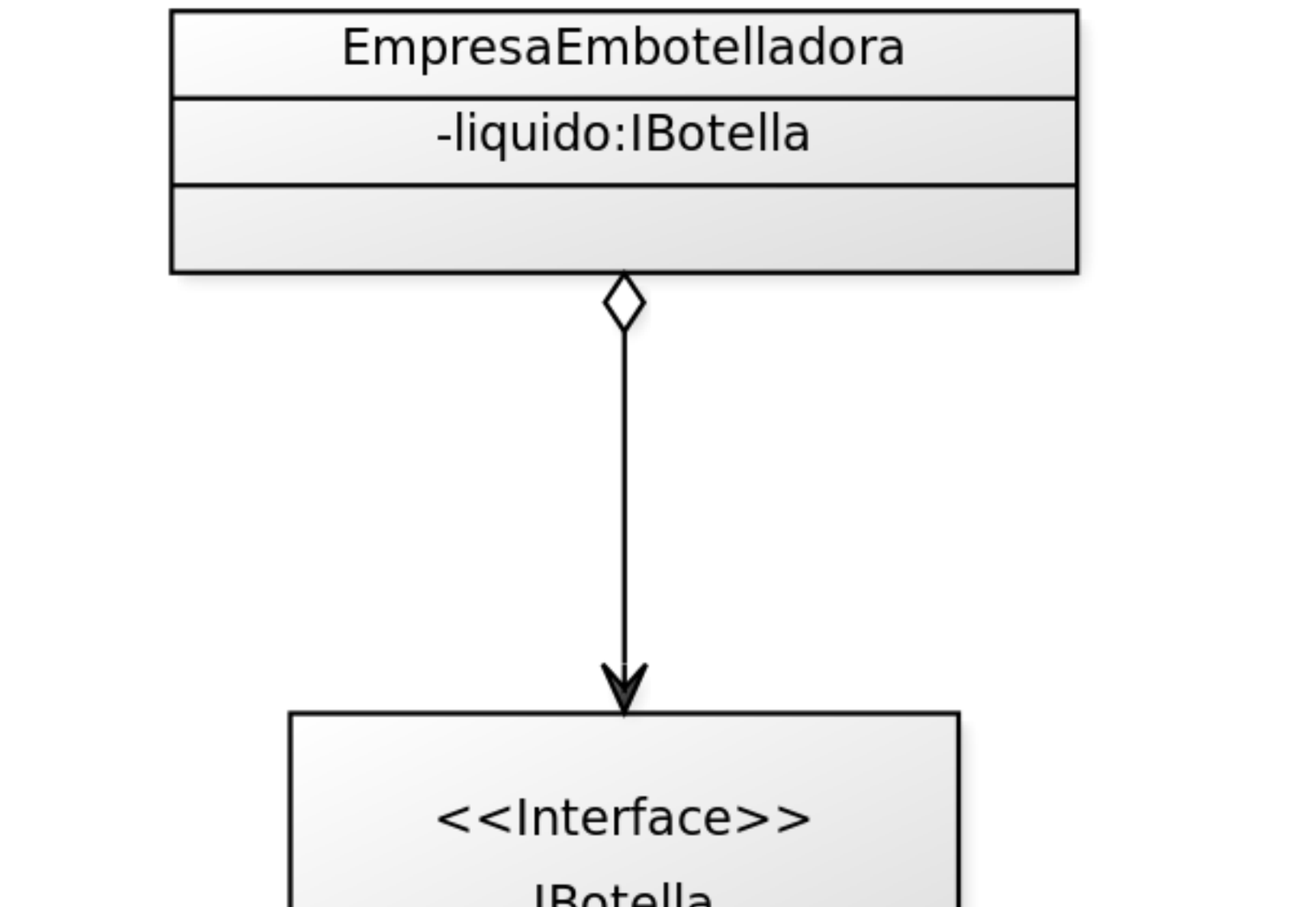
Ahora ha surgido una oportunidad de negocio y quiere empezar a vender botellas de té helado, por lo que necesita un cambio en su modelo.

Una primera aproximación sería algo similar a esto:



Pero implicaría realizar cambios en la empresa, que tiene que aprender a comunicarse con el nuevo tipo de botella, que probablemente sea muy parecida a la anterior.

Si hubieran seguido el principio Abierto-Cerrado su diagrama de clases habrían utilizado una interfaz que comunicaría el resto del sistema con las botellas, por lo que si no se cambia el api, el resto del sistema puede trabajar ajeno al cambio. Quedaría de esta forma.



4. Conclusiones

Seguir este principio ayuda a que nuestros sistemas sean más mantenibles en el tiempo y soporten mejor los cambios, aunque ante todo hay que usar el sentido común. Un uso excesivo de interfaces reducirá la productividad y la legibilidad del código.
Espero que a los que no conocierais este principio os haya parecido interesante este concepto y os ayude a mejorar

» Embeber vídeo en MailChimp

» Tutorial VIPER en Swift

Últimos Tutoriales del Autor

» [S.O.L.I.D.] Dependency inversion principle / Principio de inversión de dependencias

» [S.O.L.I.D.] Interface Segregation Principle / Principio de segregación de interfaz

» [S.O.L.I.D.] Liskov substitution

» [S.O.L.I.D.] Single responsibility principle / Principio de Responsabilidad Única

» Emmet.io - el toolkit esencial para los desarrolladores Web

vuestra forma de programar.

5. Principio S.O.L.I.D.

- Single Responsibility Principle / Principio de Responsabilidad Única
- Open-Closed Principle / Principio Abierto-Cerrado
- Liskov Substitution
- Interface Segregation Principle / Principio de segregación de interfaz
- Dependency inversion principle / Principio de inversión de dependencias

A continuación puedes evaluarlo:

Regístrate para evaluarlo

Por favor, vota +1 o compártelo si te pareció interesante

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

Fecha publicación: **2014-10-23-17:23:57**

Autor: **smartin**

Muchas gracias lalin1982, efectivamente las imágenes estaban descolocadas :)

Fecha publicación: **2014-10-23-14:22:31**

Autor: **lalin1982**

Muy buen artículo. Creo que las imágenes no están en el orden adecuado.

Un saludo.