

Haciendo software libre en el trabajo

Enrique Zamudio
@chochosmx



¿Quién?

- 25 años programando profesionalmente
- Contribuidor en Ceylon 2012-2017
- Autor de j8583 y jAlarms

j8583

- Biblioteca en Java para crear y leer mensajes ISO8583
- Es un protocolo muy tedioso de manejar
- Es aburrido
- NADIE basa su negocio en algo así
 - Es algo que usas porque tienes que usar

j8583: ¿por qué?

- Implementé ISO8583 en ObjC en 2002
- Lo volví a implementar en Java en 2004, como cliente
- Lo volví a implementar en Java en 2006, como server

BORING

BORING

BORING

BORING





¿Y si lo libero?



Hora de liberarlo

- ¿Existe una alternativa similar?
- ¿Es parte del core business?
- ¿Nos da una ventaja significativa sobre nuestros competidores?
- ¿Le puede ayudar a nuestros clientes y proveedores?

¿Existe alternativa similar?

- Más o menos...
- jPOS
 - muy complicado
 - sí lo intenté
 - al parecer es un switch completo
 - los foros están llenos de gente con frustración de no poder echarlo a andar

¿Es parte de mi core business?

- LOL no
 - Si lo sacamos como FOSS de hecho podemos tenerlo como dependencia en vez de un proyecto interno
 - Casi no lo modificamos
 - Ya era configurable pero no tanto
 - Tendríamos nosotros mismos algo mejor que usar



¿Nos da una ventaja significativa sobre nuestros competidores?

- LOL no
- Por lo que hemos visto, cada quién implementa ISO8583 como puede
- Hay vendedores de middleware que sí están haciendo dinero aprovechando la flojera/ignorancia/incompetencia de las empresas/devs
 - Sus soluciones van mucho más allá del protocolo, es un switch completo



¿Le puede ayudar a nuestros clientes y proveedores?

- ¡Sí!
 - Siempre y cuando usen Java...
 - Qué coincidencia, el primer candidato usa Java
 - Tan simple como pedirles que bajen un jar, lean unos docs y lo usen con una configuración que les damos (esa ya es propietaria)



A woman with dark hair, wearing a white sleeveless top and a black skirt, stands in a room. She has her hands behind her back. In the background, a man is sitting on a couch reading a newspaper. The room has a brick wall, a coat rack with a purple coat, and a framed picture on the wall.

**I WANT TO
BREAK
FREE**

¿Cómo liberarlo?

- Separarlo del proyecto actual
- CERO acoplamiento: proyecto depende de lib, pero lib no depende de proyecto
- Refactoring en proyecto
- Publicar artefactos en Maven Central, repo en SourceForce / github / bitbucket

¿Qué beneficios nos dio?

- Mantenerlo como algo separado del resto del proyecto ayuda a simplificar el proyecto, no complicarlo
- Las mejoras se propagan para todos
 - Un cliente/proveedor reporta un bug, lo arreglamos, todos obtienen la corrección
 - Aplica para gente que lo usa que ni siquiera conocemos

Desventajas

- Lo mejoramos como necesitamos
- Nuestros proyectos han causado versiones nuevas
- Proyectos ajenos, no tanto
 - Usuarios que piden, piden, piden
 - Casos muy específicos
 - El soporte es gratis

Pull Requests

- ¿Afecta la funcionalidad existente, o sólo agrega nueva?
- ¿Incluyeron pruebas?
- ¿Es un caso lo suficientemente general?
- ¿Hay una mejor manera de resolver el problema?

¿Vale la pena?



I have several questions.

 @chochosmx

<http://bitbucket.org/chochos/j8583>

<http://bitbucket.org/chochos/jAlarms>

<https://bitbucket.org/chochos/ScalaSQL>

CC  SS