



Estudio del framework Symfony2 para el desarrollo de aplicaciones empresariales

María Cristina Valle

Estudio del framework Symfony2 para el desarrollo de aplicaciones empresariales

María Cristina Valle Dávila
e-mail: crisvalle35@gmail.com

RESUMEN: El presente documento detalla todos los procesos que se siguieron para el desarrollo del sistema web de facturación de productos de ferretería para la microempresa GRADA, aplicando el estudio del framework¹ Symfony2, el cual permite al programador centrarse en la lógica del negocio gracias a que reutiliza conceptos y desarrollos de terceros como es el framework ORM² que se encarga de la comunicación con la base de datos. Se investigó la metodología de desarrollo ágil Extreme Programming para la elaboración del sistema, que se basa en cuatro principios como son comunicación, valor, realimentación y simplicidad, logrando disminuir el tiempo de desarrollo y adaptación a los cambios de requisitos que surgieron durante cualquier punto de la vida del sistema de facturación. Se desarrolló los módulos de compras, ventas y reportes, para lo cual fue necesario conocer el proceso de comercialización, tipos de contribuyentes que establece el SRI³ y los respectivos documentos contables que intervienen.

PALABRAS CLAVE: Framework, Symfony2.

ABSTRACT. This document details all the processes that were followed for the development of web invoicing system hardware products microenterprise GRADA, applying the study of Symfony2 framework, which allows the developer to focus on business logic thank you to third concepts and developments such as ORM framework that handles communication with the database. Agile development methodology for Extreme Programming system development, which is based on four principles as Communication, courage, simplicity and feedback, decreasing development time and adaptation to changing requirements that emerged during most any point investigated life billing system. development modules are purchasing, sales reports and, where was necessary meet the marketing process, types

¹ **Framework:** Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular

² **ORM:** Mapeo objeto-relacional, técnica para convertir datos entre el lenguaje de programación orientado a objetos y el de una base de datos relacional.

³ **SRI:** Servicio de rentas internas del Ecuador

of taxpayers establishing the SRI and the respective accounting documents involved.

Keywords. Framework, Symfony2.

INTRODUCCIÓN

Actualmente existen varios frameworks de desarrollo de aplicaciones web, que disminuyen el tiempo de producción de las aplicaciones para permitir al desarrollador dedicarse a las tareas más importantes sobre la lógica del negocio en cada aplicación. Symfony es un framework para desarrollar aplicaciones web con PHP basado en modelo vista controlador. Fue creado por una gran comunidad liderada por Fabien Potencier, quién actualmente sigue al frente de este proyecto con una visión orientada hacia las mejores prácticas que forman parte del estándar de desarrollo de software. El presente artículo tiene la finalidad de estudiar el framework Symfony2 para demostrar sus ventajas a través de la implementación de un sistema web de facturación de productos de ferretería para la microempresa GRADA usando la metodología Extreme Programming para su desarrollo.

FRAMEWORK SYMFONY2

1.1 Características

- Utiliza PHP 5.3.2
- **Rendimiento rápido:** Incluye un acelerador web creado con PHP, que guarda en la cache⁴ páginas completas. El acelerador minimiza las veces que se ejecuta la aplicación
- **Completo:** Proporciona una selección de componentes Symfony2 y algunas librerías de terceros, por ejemplo, SwiftMailer para enviar mensajes de correo electrónico.
- **Estándar:** Symfony2 se adapta perfectamente al modelo de petición más respuesta HTTP. Toda su arquitectura se ha diseñado para

⁴**Cache:** Memoria que almacena datos temporalmente para reducir el ancho de banda consumido



Estudio del framework Symfony2 para el desarrollo de aplicaciones empresariales

María Cristina Valle

facilitar la creación de un objeto de tipo Response a partir de un objeto de tipo Request. Fuera de este ámbito, Symfony2 desaparece y deja que sea el programador el que realice su trabajo como quiera.

- **Modular:** Symfony2 da la flexibilidad de quitar o incorporar cualquier componente, librería o bundle según las necesidades del usuario.
- **Acceso a herramientas de software libre** tales como Doctrine. También plantillas, seguridad, formularios, validación y traducción, entre otras a través de los diferentes bundles desarrollados por la comunidad de symfony2.
- **Sencillo de aprender:** La aplicación cuenta con código claro y bien organizado. Esto facilita la reutilización de código y permite a los nuevos desarrolladores ser productivos en el proyecto con mayor rapidez. [1]

1.2 Componentes

Symfony2 tiene veinte y dos componentes independientes que se pueden utilizar dentro de cualquier proyecto PHP y contienen algo útil para casi cualquier situación, independientemente de cómo se desarrolle la aplicación

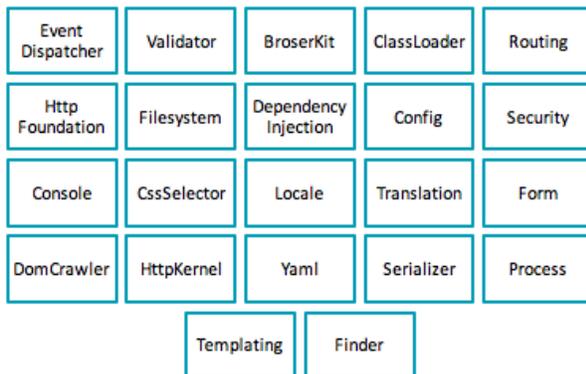


Figura 1: Componentes de Symfony2

- **HttpFoundation:** Contiene las clases Request⁵ y Response⁶ así como otras

clases para manejar sesiones y cargar archivos

- **Routing:** Potente y rápido sistema de enrutamiento que permite asociar una URL específica por ejemplo /contacto a cierta información acerca de cómo se debe manejar esa petición por ejemplo, ejecutando el método contactoAction()
- **Form:** Una completa y flexible plataforma para crear formularios y procesar los datos presentados en ellos
- **Validator:** Un sistema para crear reglas sobre datos y cuando el usuario presenta los datos comprobar si son válidos o no siguiendo esas reglas
- **ClassLoader:** Una biblioteca de carga automática que permite utilizar clases PHP.
- **Templating:** Un juego de herramientas para reproducir plantillas, manejar la herencia de la decoración de las plantillas y realizar otras tareas comunes de las plantillas
- **Security:** Una poderosa biblioteca para manejar todo tipo de seguridad.
- **Translation:** Una plataforma para traducir cadenas en la aplicación. [2]

1.3 Estructura de directorios

Por defecto todas las aplicaciones Symfony2 tienen la misma estructura de directorios sencilla y recomendada:



Figura 2: Directorio de Symfony2. [3]

⁵**Request:** clase php que representa una petición HTTP

⁶**Response:** clase php que representa una respuesta HTTP



Estudio del framework Symfony2 para el desarrollo de aplicaciones empresariales

María Cristina Valle

3 METODOLOGÍA PARA EL DESARROLLO DEL PROYECTO

Una metodología que permita lograr un código sin errores, con alta funcionalidad, manteniendo al cliente al tanto del proyecto y en plazos de tiempo cómodos, siempre han sido objetivos ideales que en todo proyecto se pretende alcanzar. La Metodología de "Programación Extrema" (XP) propone la manera de alcanzar esos objetivos.

Esta metodología de desarrollo de software consiste en un conjunto de prácticas basadas en cuatro principios, como son comunicación, simplicidad, retroalimentación y valor entre los miembros del equipo de desarrollo para que el proyecto tenga éxito. El presente capítulo, desarrolla las etapas de la Metodología de Programación Extrema.

3.1 Fase de Exploración

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, el programador realiza una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

La clave para el éxito en desarrollo de software para la metodología XP radica en promover el trabajo en equipo y propiciar un buen ambiente de trabajo, pues continuamente debe haber una retroalimentación de entre el cliente y el equipo de desarrollo, destacando comunicación entre todos los participantes y simplicidad en las soluciones implementadas.

Programador (Programmer)

- Escribe las tareas en base a las historias de usuario
- Produce el código del sistema.

- Debe existir una comunicación y coordinación adecuada entre los programadores y otros miembros del equipo

Cliente (Customer)

- Escribe las historias de usuario que describen el funcionamiento del sistema
- Asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.
- El cliente es sólo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

Encargado de pruebas (Tester)

- Ayuda al cliente a escribir los casos de pruebas de aceptación
- Ejecuta las pruebas regularmente
- Difunde los resultados en el equipo

Encargado de seguimiento o Rastreador (Tracker)

- Proporciona realimentación al equipo en el proceso XP.
- Su responsabilidad es verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones.
- Realiza el seguimiento del progreso de cada iteración y evalúa si los objetivos son alcanzables con las restricciones de tiempo y recursos presentes. Determina



Estudio del framework Symfony2 para el desarrollo de aplicaciones empresariales

María Cristina Valle

cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.

Entrenador (Coach)

- Es responsable del proceso global.
- Es necesario que conozca a fondo el proceso XP para proveer guías a los miembros del equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Gestor (Manager)

- Es el vínculo entre clientes y programadores
- Ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas
- Su labor esencial es de coordinación.

Consultor

- Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto.
- Guía al equipo para resolver un problema específico. [4]

Tabla 1: Integrantes del Equipo XP

| Nombre | Descripción | Rol XP |
|--------------------------|-----------------------------------|---------------------------------|
| Dra. Gabriela Manzo | Contadora de microempresa GRADA | Consultor y Probador |
| Graciela Dávila | Propietaria de microempresa GRADA | Cliente y Probador |
| María Cristina Valle | Tesista | Programador |
| Ing. José Luis Rodríguez | Director de tesis | Gestor, Entrenador y Rastreador |

3.2 Fase de Planificación

En la fase de planificación se describen el número y orden de las iteraciones que se necesitan para cumplir con cada historia de usuario que se recopiló en la fase de exploración. Se entregarán diferentes versiones del sistema de acuerdo a la realización de las historias de usuario, las fechas de entrega de cada versión del sistema se coordina con el Rastreador.

La velocidad del equipo se calcula de la siguiente forma:

Programadores en el equipo: 1 persona.

Velocidad del equipo al mes:

$$\frac{4 \text{ semanas}}{1 \text{ persona}} = 4 \text{ semanas al mes}$$

El resultado es 4 semanas ideales de desarrollo al mes para el equipo de desarrollo

En la sección de estimaciones de historias de usuario en tiempo ideal se obtuvo un total de 27.0 semanas ideales, para implementar la solución propuesta por el equipo de desarrollo, el cual sirve para calcular el número de iteraciones, que se indica a continuación:

Semanas ideales totales para implementar la solución: 27.0 semanas.

Velocidad del equipo de desarrollo al mes: 4 semanas

Número de iteraciones para la entrega:

$$\frac{27.0 \text{ semanas}}{4 \text{ semanas}} = 6.75 \text{ iteraciones}$$

Una iteración corresponde a un mes ideal, para este caso se trabaja de lunes a viernes, cuatro horas diarias con un programador.

Se obtiene un resultado de 7 iteraciones.

3.3 Fase de Iteración

En la fase de iteración se describe detalladamente la ejecución de las tareas que se realizan para concluir cada funcionalidad del sistema planteado a desarrollar.

Dentro de cada iteración, se realizan actividades como: diseñar pruebas de aceptación, programar historias de



Estudio del framework Symfony2 para el desarrollo de aplicaciones empresariales

María Cristina Valle

usuario, refactorización⁷ de código, refactorización de diseño cuando sea necesario y ejecutar pruebas de aceptación.

5 CONCLUSIONES

- ✓ Se logró concluir con la implementación del sistema de facturación para la microempresa GRADA con la siguiente funcionalidad: creación y emisión de Facturas y Comprobantes de retención para los clientes, almacenamiento de Cartas de devolución de mercadería, también Registro de Facturas, Notas RISE y Comprobantes de retención de los proveedores, Lista de clientes, Lista de proveedores, Lista de productos, Reportes.
- ✓ El funcionamiento de Symfony2 se basa en la arquitectura Modelo Vista Controlador, en la Vista se usa el motor de plantillas Twig, en el Modelo el ORM Doctrine2, en el Controlador intervienen veinte y dos componentes de Symfony2, además la filosofía de la segunda versión de Symfony radica en que todo es un bundle desde la funcionalidad básica, un bundle es un directorio que contiene todo tipo de archivos dentro de una estructura jerarquizada de directorios con una característica específica de funcionalidad.
- ✓ Se identificó las siguientes ventajas del framework Symfony2:
 - Permite implementar herramientas de software libre de alta calidad a través de bundles de terceros desarrollados por la comunidad de symfony2.
 - Tiene una comunidad activa en inglés y español que brinda soporte, mantiene una documentación actualizada y se resuelven dudas sobre la implementación de soluciones en symfony2
 - Se puede empaquetar funcionalidades propias de desarrollo en un bundle.

- Posee un potente motor y lenguaje de plantillas para la presentación de datos llamado Twig que permite aplicar herencia en las plantillas.
 - Ofrece una barra de depuración en el entorno de desarrollo que ayuda a encontrar la solución a los errores que se presentan durante el desarrollo de un proyecto
 - Tiene el código bien organizado con la finalidad de facilitar la reutilización de código y permite a los nuevos desarrolladores ser productivos con mayor rapidez
 - Ofrece un sistema de seguridad para evitar que un usuario pueda acceder a un determinado recurso, basado en dos etapas, autenticación mediante firewalls y autorización según los privilegios que se posea el usuario.
- ✓ Se pudo conocer el proceso de comercialización, tipos de contribuyentes que establece el SRI y los respectivos documentos contables que intervienen en el proceso de comercialización, dentro de los documentos para las ventas se manejan Facturas que constituyen un documento legal de la prestación de un bien o servicio, cobrando el 12% de impuesto IVA en los productos y 0% de impuesto IVA en los productos que están exentos de acuerdo con las normas del SRI, además Comprobantes de retención con el 1% de impuesto a la renta en calidad de que la microempresa GRADA es contribuyente obligado a llevar contabilidad y Cambios de mercadería.
- Se ha clasificado a los proveedores y clientes en los siguientes tipos de contribuyentes: Obligado a llevar contabilidad, Contribuyente de régimen simplificado y Contribuyente especial, pudiendo ser persona natural o sociedad. En el proceso de Compras se registra Facturas o Notas RISE dependiendo del tipo de contribuyente que sea el proveedor, se almacena Comprobantes de retención con Impuesto a la renta del 1% o 30% según el tipo de contribuyente, además se realiza Cambio de mercadería para el proveedor en caso de ser necesario.

⁷**Refactorización:** es el proceso que consiste en mejorar el código una vez escrito cambiando su estructura interna sin modificar su comportamiento externo.



Estudio del framework Symfony2 para el desarrollo de aplicaciones empresariales

María Cristina Valle

- ✓ Se investigó la metodología Extreme Programming para la elaboración del sistema de facturación en la microempresa GRADA, que se basa en cuatro principios comunicación, valor, realimentación y simplicidad con la finalidad de disminuir tiempo de desarrollo en los proyectos de software, adaptarse a cambios de requisitos en cualquier punto de vida del proyecto, promover el trabajo en equipo, simplicidad en las soluciones implementadas y valor para enfrentar los cambios como clave para el éxito en desarrollo de software.

6 RECOMENDACIONES

- ✓ Todas las empresas por pequeñas o grandes que sean es aconsejable que tengan un sistema de facturación que les permita controlar la entradas y salida de la mercadería y mantener segura la información en una base de datos con el propósito de crecer en el sector empresarial y brindar un mejor servicio, pues toda empresa grande comenzó como una microempresa o un pequeño negocio.
- ✓ La instalación de bundles de terceros es recomendable hacerlo mediante la consola Composer para asegurar la funcionalidad, pues antes de instalar un bundle es necesario instalar todas sus dependencias que son librerías con estructura de directorio de bundle, un bundle puede necesitar una o varias dependencias.
- ✓ Generar las entidades mediante ingeniería inversa es un proceso sencillo ya que Doctrine2 tiene en cuenta la información de las tablas, columnas, índices, claves foráneas y primarias, pero cuando el esquema de la base de datos es muy complejo Doctrine2 solamente puede obtener el 80% de la información, herencia de entidades y eventos y otra información avanzada es aconsejable configurarlo manualmente.
- ✓ Brindar la capacitación necesaria al personal de la microempresa GRADA que está involucrado en el manejo del sistema de facturación para que estén familiarizados y no tengan problema.

frequencies", IEEE Trans. On Aut. Control. Vol. 47. No 5. pp 857-863. May 2002.

- [2] H. Khalil, "Nonlinear Systems", 2nd. ed., Prentice Hall, NJ, pp. 50-56, 1996.
- [3] Francis. B. A. and W. M. Wonham, "The internal model principle of control theory", Automatica. Vol. 12. pp. 457-465. 1976.
- [4] E. H. Miller, "A note on reflector arrays", IEEE Trans. Antennas Propagat., Aceptado para su publicación.

REFERENCIAS

- [1] G. Obregón-Pulido, B. Castillo-Toledo and A. Loukianov, "A globally convergent estimator for n