

Sistema web de control de evaluaciones por competencias

Web-based competency-based evaluation control system

DOI: 10.46932/sfjdv3n6-057

Received in: November 10th, 2022

Accepted in: December 15th, 2022

Eliud Paredes Reyes

Maestría en Sistemas Computacionales

Institución: Instituto Tecnológico Superior del Occidente del Estado de Hidalgo

Dirección: Paseo del Agrarismo, 2000, Carr. Mixquiahuala, Tula, km 2.5, Mixquiahuala de Juárez, Hidalgo, C.P. 42700

Correo electrónico: eparedes@itsoeh.edu.mx

Cristy Elizabeth Aguilar Ojeda

Maestría en Tecnologías de la Información

Institución: Instituto Tecnológico Superior del Occidente del Estado de Hidalgo

Dirección: Paseo del Agrarismo, 2000, Carr. Mixquiahuala, Tula, km 2.5, Mixquiahuala de Juárez, Hidalgo, C.P. 42700

Correo electrónico: caguilar@itsoeh.edu.mx

Lorena Mendoza Guzmán

Maestría en Tecnología Educativa

Institución: Instituto Tecnológico Superior del Occidente del Estado de Hidalgo

Dirección: Paseo del Agrarismo, 2000, Carr. Mixquiahuala, Tula, km 2.5, Mixquiahuala de Juárez, Hidalgo, C.P. 42700

Correo electrónico: lmendozag@itsoeh.edu

RESUMEN

El presente documento contiene información sobre el sistema web de control de evaluaciones por competencias, el cual facilita el control y registro de las actividades de aprendizaje estipuladas por los docentes de la carrera de Ingeniería en Sistemas Computacionales de ITSOEH, además de permitir la elaboración de rubricas o utilizar alguna que haya sido creada con anterioridad por otro docente que haya impartido la materia, lo cual contribuye a mejorar el control de las evaluaciones de las diversas materias, la metodología utilizada fue Scrum, con cinco Sprints de tres semanas de duración, las tecnologías utilizadas para el desarrollo del proyecto son las siguientes: el framework Symfony y como gestor de base de datos Mysql, como resultado se diseñaron los módulos para los perfiles de administrador, docente y alumno; facilitando la consulta de calificaciones parciales y finales tanto para el docente como para el alumno, así como los instrumentos de evaluación empleados en cada actividad de aprendizaje, además de generar reportes de dichas evaluaciones por unidad de las materias impartidas en el semestre, el sistema está diseñado para que pueda ser utilizado por las otras carreras de la institución.

Palabras clave: competencia, instrumento, scrum, evaluación, sistema.

ABSTRACT

This document contains information about the web system of control of evaluations by competencies, which facilitates the control and registration of learning activities stipulated by the teachers of the Computer Systems Engineering career of ITSOEH, in addition to allowing the development of rubrics or use one that has been created previously by another teacher who has taught the subject, which helps to

improve the control of the evaluations of the various subjects, the methodology used was Scrum, with five Sprints of three weeks duration, the technologies used for the development of the project are as follows: Symfony framework and Mysql database manager, as a result the modules were designed for the administrator, teacher and student profiles; facilitating the consultation of partial and final grades for both the teacher and the student, as well as the evaluation instruments used in each learning activity, in addition to generating reports of these evaluations per unit of the subjects taught in the semester, the system is designed so that it can be used by the other careers of the institution.

Keywords: competence, instrument, scrum, evaluation, system.

1 INTRODUCCIÓN

Con el transcurso del tiempo la forma de evaluación en la educación a nivel superior a sufrido cambios donde se requiere mayor énfasis al momento de calificar en el modelo de competencias para lo cual se definen instrumentos y procesos de evaluación de las diversas asignaturas.

Por lo tanto los pasos que se tomaron en cuenta para la construcción de un sistema web para la automatización de rubricas de evaluación, desarrollado bajo la metodología Scrum y utilizando el framework Symfony del lenguaje PHP y como gestor de base de datos Mysql, para lo cual se trabajo en la carrera de Ingeniería en Sistemas Computacionales de ITSOEH como caso de prueba para la implementación del sistema y demostrar que puede cubrir las necesidades inmediatas de cualquier carrera del instituto tecnológico de nivel superior que desee implementarlo.

Durante la implementación de la metodología Scrum se realizaron historia de usuario que permitieron priorizar las actividades para la creación del sistema y agilizar el desarrollo y entregas constantes del producto.

Dentro de los objetivos definidos para la creación del sistema se describe la funcionalidad con la que cuenta el sistema, como contar con tres perfiles de usuario administrador, profesor y alumno, el primero puede dar de alta profesores, alumnos, periodos, semestres, materias, el segundo usuario puede configurar o generar rubricas de evaluación, asignar grupos, evaluar alumnos y el tercer usuario puede consultar calificaciones y el detalle de cada evaluación ya que podrá visualizar cada rubro y herramienta con la que fue evaluado por el profesor todo esto con el fin de optimizar la evaluación continua, formativa y coordinar las tareas de los profesores de la carrera de Ingeniería en Sistemas Computacionales.

La importancia de este sistema es mejorar el control de calificaciones tanto para el docente y para el alumno para llevar acabo el seguimiento oportuno de sus evaluaciones para cada actividad asignada en cada una de las unidades de las asignaturas.

El articulo está distribuido en dos apartados: la sección del desarrollo del sistema que describe cada las fases de la metodología y la sección de resultados y conclusiones que describen los objetivos alcanzados así como las mejoras futuras al sistema de evaluación por competencias a nivel superior.

2 TEORÍA

Este apartado describe la implementación de la metodología de trabajo Scrum, donde incluye los artefactos o documentos con los que se gestionan las tareas de adquisición y suministro: requisitos, monitorización y seguimiento del avance, así como las responsabilidades y compromisos de los participantes en el proyecto (Guía SBOKTM, 2017).

La primera fase de la metodología se asignaron los roles al equipo de trabajo como se muestra tabla 1.

Tabla. 1. Personas y roles

<i>Rol</i>	<i>Persona</i>	<i>Contacto</i>
<i>Gestor de Producto P. Owen</i>	Eliud Paredes Reyes	
<i>Coordinador Scrum Master</i>	Eliud Paredes Reyes	
<i>Equipo Técnico</i>	Cristy Elizabeth Aguilar Ojeda Lorena Mendoza Guzmán	

Fuente: (Elaboración propia)

Una vez que se llevó a cabo la recopilación de requerimientos mediante la entrevista del propietario producto con el cliente se realizó la pila de producto, en la tabla 2 se puede apreciar las priorización de cada historia según su urgencia e importancia para determinar los tiempos y el esfuerzo necesarios para el desarrollo del sistema. Dicho de otra manera, se formaliza la planificación del proyecto.

- Product backlog.
- Spring backlog.

El Product Backlog es un listado priorizado de funcionalidades o requisitos que son básicamente los objetivos que el cliente tiene como expectativa del producto de software a desarrollarse (Guía SBOKTM, 2017).

El listado de historias de usuario será jerarquizado en virtud de su importancia para el negocio, dicha importancia se calcula multiplicando la urgencia por el impacto. La urgencia se mide en una escala del 1 al 5 siendo el 1 la menor urgencia. Adicionalmente el Product Backlog estima la importancia de modo que el tiempo de implementación de las tareas (medido en días).

Una vez que se definió la importancia de las tareas se asignaron cada una de ellas a una iteración. La iteración más próxima se incluyó a las tareas de mayor importancia.

Tabla 2.Pila del producto

PILA DEL PRODUCTO					
ID	HISTORIA DE USUARIO	URGENCIA	IMPACTO	IMPORTANCIA	TIEMPO ESTIMADO (días)
1	Acceso de usuarios al sistema	5	4	20	5
2	Modificar Usuario Existente	5	4	20	5
3	Eliminar Usuario Existente	2	4	8	4
4	Implementación de Perfiles	4	4	16	4
5	Creación de usuario y contraseña de (Docentes y Alumnos)	4	3	12	5
6	Modificar Contraseña de Usuario	5	3	15	5
7	Listar Usuarios (Docentes y Alumnos)	5	2	10	4
8	Modificar estatus del Usuario	3	3	9	3
9	Modificar Información Personal de los Usuarios	5	2	10	3
10	Agregar Periodos	4	3	12	4
11	Agregar Materias	4	3	12	5
12	Agregar Grupos	5	4	20	5
13	Listar Materias	5	3	15	5
14	Agregar Semestres	5	4	20	4
15	Configurar Unidades	4	4	16	5
16	Configurar Rubrica	5	4	20	6
17	Mostrar Detalle de Rubros	3	2	6	6
18	Calificar Alumnos	4	2	8	5
19	Calificaciones Finales	4	4	16	5
20	Eliminar Información del sistema	2	4	8	4
21	Crear Cuenta de Usuario	2	3	5	3
22	Modificar Cuenta de Usuario	4	4	16	5
23	Usuario Dar de Baja a un Usuario	4	4	16	3
24	Mostrar Calificaciones Finales del Alumno	5	5	25	5

Fuente:(Elaboración propia)

2.1 PLANIFICACIÓN INICIAL DE LAS ITERACIONES/ SPRINTS

Como resultado de la priorización de Historias de Usuario en el Product Backlog, se puede realizar una planificación tentativa de las iteraciones necesarias en este ciclo de desarrollo de software. En la tabla 3 se muestra una primera aproximación, la que podríamos llamar ideal, del Product Backlog dividido en Sprints o iteraciones. Los Sprint inicialmente asignados podrían ser susceptibles de modificación, como es propio en desarrollos de tipo ágil, de ocurrir estos cambios serán explicados puntualmente en el desarrollo de cada Sprint.

Como la metodología que se utilizó en este proyecto es de tipo iterativo e incremental se hace necesario un ajuste en planificación en cada siguiente iteración. Claramente el siguiente resultado dependerá del anterior. Si los tiempos estimados son los correctos, no habrá modificaciones, caso contrario deberán ser aumentados o disminuidos dependiendo el caso.

Tabla 3. División ideal del product backlog en sprints (iteraciones)

ID	HISTORIA DE USUARIO	SPRINT
1	Acceso de usuarios al sistema	1
2	Modificar Usuario Existente	2
3	Eliminar Usuario Existente	2
4	Implementación de Perfiles	1
5	Creación de usuario y contraseña de (Docentes y Alumnos)	1
6	Modificar Contraseña de Usuario	2
7	Listar Usuarios (Docentes y Alumnos)	1
8	Modificar estatus del Usuario	4
9	Modificar Información Personal de los Usuarios	2
10	Agregar Periodos	1
11	Agregar Materias	1
12	Agregar Grupos	1
13	Listar Materias	2
14	Agregar Semestres	1
15	Configurar Unidades	2
16	Configurar Rubrica	3
17	Mostrar Detalle de Rubros	3
18	Calificar Alumnos	3
19	Calificaciones Finales	3
20	Eliminar Información del sistema	4
21	Crear Cuenta de Usuario	1
22	Modificar Cuenta de Usuario	4
23	Usuario Dar de Baja a un Usuario	4
24	Mostrar Calificaciones Finales del Alumno	5

Fuente: Elaboración propia.

Las historias de usuario se realizaron con el siguiente formato como se muestra en la tabla 4 que en este caso es del administrador.

Tabla 4 Formato de historia de usuario (administrador)

Tabla 4 Formato de historia de usuario (administrador)				
Número:	<u>1</u>	Usuario:	Administrador (Rolando)	
Nombre Historia:		Módulo Administrador		
Prioridad en Negocio:		Alta	Riesgo en Desarrollo	Solo Desarrollo
Horas Estimadas:		[No. Horas – Solo Desarrollo]		
Descripción:				
En el Módulo de perfil de administrador se podrá dar de alta, consultar, modificar y eliminar datos.				
Criterios de aceptación:				
Qué es lo que quiere hacer el usuario				

- Quiero poder dar de alta Profesores
- Quiero poder dar de alta Alumnos
- Quiero poder dar de alta Materias
- Quiero poder dar de alta Periodos
- Quiero poder Configurar Retículas
- Quiero poder Ver la lista de alumnos registrados en la base de datos
- Quiero poder Ver la lista de Profesores registrados en la base de datos
- Quiero poder Ver la lista de materias registrados en la base de datos
- Quiero poder actualizar la contraseña de los profesores
- Quiero poder actualizar la contraseña de los alumnos

Qué es lo que no debe hacer el usuario

- No debo poder actualizar la contraseña si el campo nueva contraseña no coincide con el campo Confirmar contraseña.
- No debo poder actualizar la contraseña si no he agregado valores a los campos nueva contraseña y campo confirmar contraseña.
- No debo de poder dar de alta un profesor si no e ingresado los valores de los campos [Matricula, Nombre, Apellido paterno, Apellido materno]

Acciones quiere que haga el sistema

- Cuando pulse clic en la pestaña Registrar del menú Profesores debe de mostrar un formulario con los datos necesarios para dar de alta un profesor.
- Cuando pulse clic en el botón Registrar debe de guardar los datos del profesor en la base datos.
- Cuando pulse clic en la pestaña Registrar del menú Alumno debe de mostrar un formulario con los datos necesarios para dar de alta un Alumno.

Observaciones:

Fuente: (Elaboración propia.)

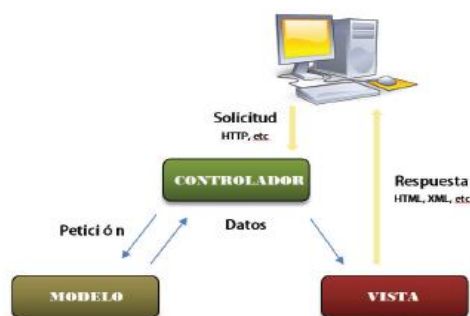
El patrón de diseño para el sistema fue el Modelo-Vista-Controlador que a continuación se describe.

2.2 DESCRIPCIÓN DEL PATRÓN DE DISEÑO

Este patrón separa el modelo o lógica del negocio de su presentación o vista, enlazándolos mediante el controlador.

Con algo más de detalle podemos decir que el Modelo envuelve el comportamiento y los datos del dominio de aplicación; la Vista despliega la Información y el Controlador interpreta las entradas generadas por el mouse o el teclado de usuario, informando al Modelo cuando es necesario un cambio de estado, en la figura 1 se muestra el ciclo de vida o funcionamiento de MVC, (Kendall, 2011).

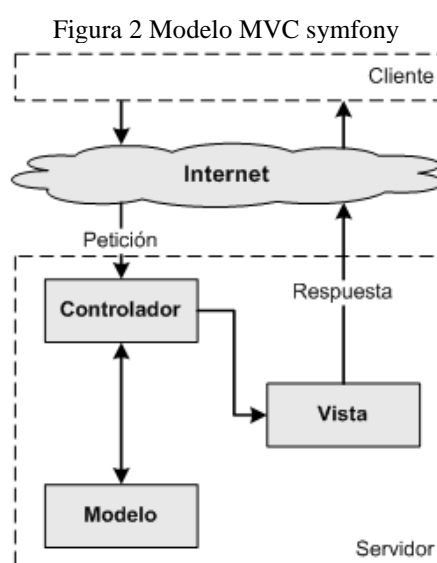
Figura 1. Funcionamiento del patrón de arquitectura modelo-vista-controlador.



Fuente:(Guía Scrum, 2017)

La Estructura MVC de symfony se muestra en la figura 2 con la estructura siguiente para el proyecto.

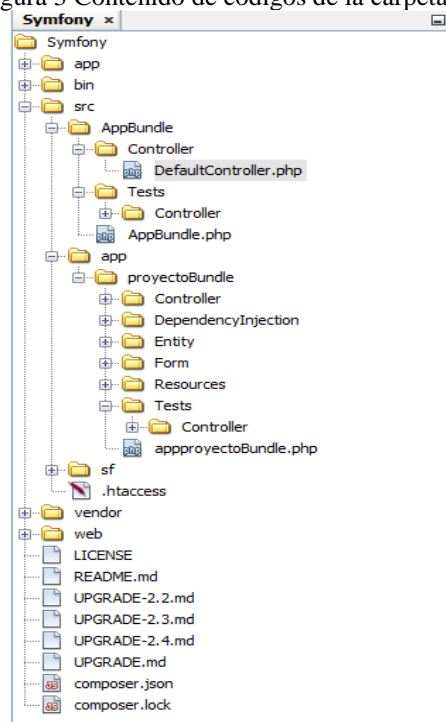
- La capa del Modelo
 - Abstracción de la base de datos
 - Acceso a los datos
- La capa de la Vista
 - Vista
 - Plantilla
 - Layout
- La capa del Controlador
 - Controlador frontal
 - Acción



Fuente:(Guía Scrum, 2017)

La carpeta SRC del proyecto (ver figura 3) contiene todo las clases del modelo de datos y el controlador para el funcionamiento del sistema.

Figura 3 Contenido de codigos de la carpeta src.



Fuente: (pantalla del proyecto)

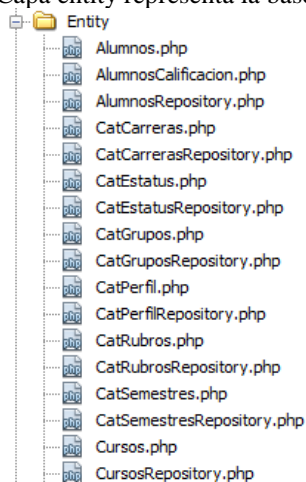
2.3 MODELO

El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

La capa del modelo se puede dividir en la capa de acceso a los datos y en la capa de abstracción de la base de datos, de esta forma, las funciones que acceden a los datos no utilizan sentencias ni consultas que dependen de una base de datos, sino que utilizan otras funciones para realizar las consultas. Así, si se cambia de sistema gestor de bases de datos, solamente es necesario actualizar la capa de abstracción de la base de datos.

La capa entity representa la base de datos mediante objetos. Como lo muestra la figura 4.

Figura 4. Capa entity representa la base de datos mediante objetos

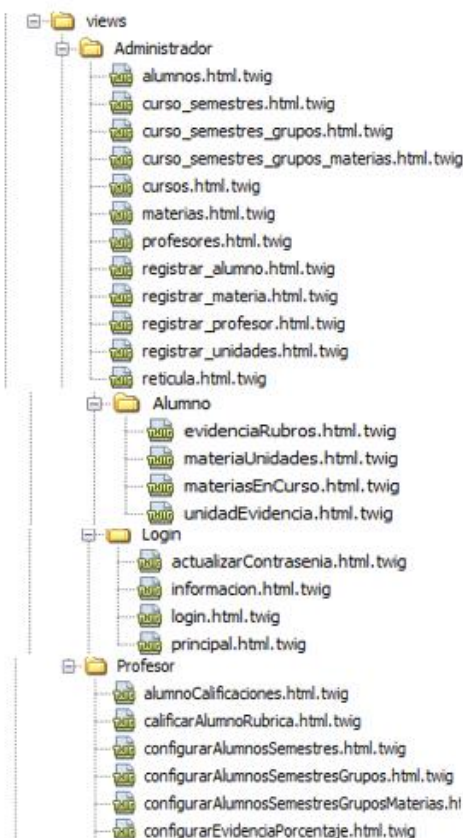


Fuente: (pantalla del proyecto)

2.4 VISTA

La vista transforma el modelo en una página web que permite al usuario interactuar con ella, normalmente sólo cambia el interior de la página, por este motivo, la vista se separa en un layout y en una plantilla donde el layout es global en toda la aplicación o al menos en un grupo de páginas. La plantilla sólo se encarga de visualizar las variables definidas en el controlador. Para que estos componentes interaccionen entre sí correctamente (ver figura 5)

Figura 5. Capa vista transforma el modelo en una página web.



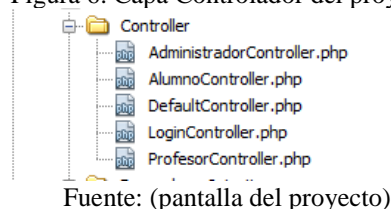
Fuente: (pantalla del proyecto)

2.5 CONTROLADOR

El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Tareas comunes se encuentran el manejo de las peticiones del usuario, el manejo de la seguridad, cargar la configuración de la aplicación y otras tareas similares. Por este motivo, el controlador normalmente se divide en un controlador frontal, que es único para cada aplicación, y las acciones, que incluyen el código específico del controlador de cada página (ver figura 6)

Figura 6. Capa Controlador del proyecto



3 RESULTADOS

El sistema se ha desarrollado con éxito se han realizado las pruebas de funcionalidad para validar el correcto funcionamiento del sistema obteniendo los resultados siguientes.

- Se diseñó un modelo de base de datos en mysql donde se guardan correctamente los datos insertados desde las interfaces diseñadas del sistema web.
- Se diseñó una interfaz amigable intuitiva para facilitar el uso de los perfiles de usuario administrador, profesor y alumno.
- Se desarrolló y diseño un módulo para el perfil administrador que puede capturar correctamente los datos de los periodos, grupos, materias, profesores y alumnos que son guardados en la base de datos correctamente y se puede tener un control más personalizado para la carrera de Ingeniería en Sistemas computacionales.
- Se desarrolló y diseño un módulo para de perfil de profesor para agilizar el proceso de creación de rubricas y evaluación del alumno.
- Se desarrolló y diseño un módulo para el perfil de alumno donde puede consultar calificaciones y detalles de la evaluación que realizo el profesor para cada unidad facilitando la consulta de calificaciones parciales y finales.

En la fase de desarrollo del sprint para el módulo de administrador se realizaron los códigos como se muestra en la figura 7 y 8 el fragmento de código para la autenticación de usuario con el framework symfony respecto al modelo y al controlador.

Figura 8. Fragmento de código del modelo (clase perfil).

```
<?php
namespace app\proyectoBundle\Entity;
use Doctrine\ORM\Mapping as ORM;

/**
 * CatPerfil
 *
 * @ORM\Table(name="cat_perfil")
 * @ORM\Entity(repositoryClass="app\proyectoBundle\Entity\CatPerfilRepository")
 */
class CatPerfil
{
    /**
     * @var integer
     *
     * @ORM\Column(name="id_perfil", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id;

    /**
     * @var string
     *
     * @ORM\Column(name="nombre", type="string", length=50)
     */

    public function getId()
    {
        return $this->id;
    }
    public function setId($id)
    {
        $this->id = $id;
        return $this;
    }
}
```

Fuente: (pantalla del proyecto)

Figura 9. Fragmento de código de controlador (clase LoginController)

```
<?php

namespace app\proyectoBundle\Controller;

use Symfony\Component\HttpFoundation\Request;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use app\proyectoBundle\Entity\Usuarios;
use Symfony\Component\Security\Core\Exception\AccessDeniedException;

class LoginController extends Controller {

    public function loginAction() {

        $authenticationUtils = $this->get('security.authentication_utils');
        $error = $authenticationUtils->getLastAuthenticationError();
        $lastUsername = $authenticationUtils->getLastUsername();

        return $this->render('appproyectoBundle:Login:login.html.twig',
            array('last_username' => $lastUsername, 'error' => $error,));
    }

    public function principalAction(Request $request) {

        if (false == $this->get('security.context')->isGranted('IS_AUTHENTICATED_FULLY'))
        || $this->getUser()->getIdEstados()->getId() == 2 {
            throw new AccessDeniedException();
        } else {

            $idPerfil = $this->getUser()->getIdPerfil()->getId();

            if ($idPerfil == 1) { //Administrador
                return $this->redirectToRoute('appproyecto_listar_profesor');
            } else if ($idPerfil == 2) { //Profesor
                return $this->redirectToRoute('appproyecto_informacion');
            } else if ($idPerfil == 3) { //Alumno
                return $this->redirectToRoute('appproyecto_informacion');
            }
        }
    }
}
```

(Pantalla del proyecto)

En la figura 10 se muestra la pantalla principal del módulo de administrador.

Figura 10. Pantalla principal del administrador.



Fuente: (pantalla del proyecto)

4 CONCLUSIONES

De lo anterior se concluye que la propuesta desarrollada es capaz de mejorar la calidad del servicio prestado por la institución, en relación con la definición de métricas de evaluación para los alumnos al llevar un control de calificaciones, donde podría llegar a convertirse en una herramienta esencial que permitirá realizar dicho proceso de manera eficiente y confiable permitiendo obtener mejores resultados, para trabajos futuros el sistema se puede modificar en la forma de evaluar en las competencias a desarrollar con respecto a las necesidades de cada docente y seguir algun estandar de evaluación.

REFERENCIAS

Sommerville, I. (2011). Modelos de proceso de software. En I. Sommerville, Ingeniería de software (pág. 792). Mexico, DF: pearson educación.

Gauchat, J. D. (2012). El gran libro de HTML5, CSS3 y Javascript. Barcelona: marcombo.

Kendall, K. E., & Kendall, J. E. (2011). Análisis y diseño de sistemas. México: pearson

Fabien Potencier (2014), Symfony 2.4, el libro oficial, LibrosWeb.

Rubin, K. s. (2014). Essential Scrum: a practical guide to the most popular agile process. United States: Pearson.

Scrumstudy, (2017), Una guía para el Cuerpo de Conocimiento de Scrum (Guía SBOK™) 3ra Edición, SCRUMstudy.