

BUENAS PRÁCTICAS APLICADAS A LA IMPLEMENTACION COLABORATIVO DE APLICATIVOS WEB



Universidad de Pamplona

Manuel Guillermo García Sandoval¹, Héctor David Ariza Torrado², Martha Lucia Pinzón³, Anderson Smith Flórez Fuentes⁴

Resumen

La implementación de buenas prácticas para el desarrollo de software, son una compilación de métodos o técnicas que permiten llevar a cabo de manera óptima el conjunto de actividades que comprenden el desarrollo de un sistema de información. En el campo de desarrollo de software colaborativo, estas prácticas son empleadas con el fin de que las actividades realizadas por cada miembro de un equipo de trabajo, y esté tengan concordancia con las actividades realizadas por los demás miembros. Esto se realiza mediante el seguimiento de las normas correspondientes, que al ser implementadas, generan un proyecto comprensible por quienes conforman el equipo de trabajo. El no seguimiento de las mismas, puede generar retrasos e incluso fracaso en un proyecto de desarrollo colaborativo, debido que al no establecer una metodología específica para la solución de un problema, cada miembro del equipo puede tomar diferentes caminos para la solución del mismo, pudiendo las discrepancias entre los mismos, provocar futuros problemas de compatibilidad dentro del sistema a implementar. Este proyecto pretende llevar a cabo el establecimiento de una metodología de desarrollo, que involucre una serie de actividades para la implementación colaborativa de un aplicativo web en el entorno colaborativo cloud9, basada en las buenas prácticas, con la posterior aplicación y evaluación de la misma, de acuerdo con los resultados obtenidos.

Palabras Clave: buenas prácticas, desarrollo colaborativo, entorno colaborativo, metodología, software.

Abstract

The implementation of best practices for software development, is a compilation of methods or techniques to perform optimally the set of activities that include the development of an information system. In the field of collaborative software development, these practices are used in order that the activities of each member of a team, and is having consistent with the activities of the other members. This is done by tracking the relevant rules, which when implemented, produce a comprehensive project who make up the team. Failure to follow them, can cause delays and even failure in a collaborative development project, because by not establishing a specific methodology for solving a problem, each team member can take different paths to the solution thereof, may discrepancies between them, causing future problems of compatibility within the system to be implemented. This project aims to carry out the establishment of a development methodology that involves a series of activities for collaborative implementation of a web application in cloud9 collaborative environment based on best practices, with subsequent implementation and evaluation of it, according to the results.

Keywords: best practices, collaborative environment, collaborative development, , methodology, software.

¹ Universidad de Pamplona, Facultad de Ingenierías y Arquitecturas, Ingeniería de Sistemas (Villa del Rosario) gerardo61@gmail.com

² Universidad de Pamplona, Facultad de Ingenierías y Arquitecturas, Ingeniería de Sistemas (Villa del Rosario) hector1150304@gmail.com

³ Universidad de Pamplona, Facultad de Ingenierías y Arquitecturas, Ingeniería de Sistemas (Villa del Rosario) marthaluciapinzon@gmail.com

⁴ Universidad de Pamplona, Facultad de Ingenierías y Arquitecturas, Ingeniería de Sistemas (Villa del Rosario) andersonflorezf@unipamplona.edu.co

BUENAS PRÁCTICAS APLICADAS A LA IMPLEMENTACION COLABORATIVO DE APLICATIVOS WEB

1. INTRODUCCIÓN

Desarrollar un software significa construirlo simplemente mediante su descripción. Está es una muy buena razón por la que se considera la actividad de desarrollo de software como una ingeniería. Una de las mayores deficiencias en la práctica de construcción de software es la poca atención que se presta a la hora de la discusión del problema.

En general los desarrolladores se centran en la solución dejando el problema inexplorado. Esto se debe a que el desarrollador o el equipo de desarrolladores no tiene definido desde el momento en que se empieza a desglosar el problema una metodología a implementar durante el proceso de desarrollo del software y es por eso que la mayoría de los proyectos de software fallan, se prolongan o se salen de presupuesto.

Al analizar el por qué sucede esto, generalmente es por las mismas razones de no seguir las mejores prácticas de desarrollo de software o seguirlas con fallas al no saber cómo implementarlas. Las buenas prácticas de desarrollo de software son un conjunto de métodos y técnicas compiladas a partir de la experiencia en miles de proyectos de software con los que es posible eliminar de tajo los problemas del desarrollo de software. Se denominan mejores prácticas porque se han identificado como el factor común que caracteriza a organizaciones exitosas de software.

Para hacer un buen desarrollo de software debemos tener en cuenta los siguientes factores que nos permitirán que cualquier software aumente las posibilidades de acabar un proyecto exitosamente:

1. Administración: Establecer un plan de trabajo, darle seguimiento para ver que se cumpla y en caso de que no, resolver los problemas que detengan el avance del proyecto.

2. Requerimientos: Hay que establecer y acordar los requerimientos en una etapa temprana del proyecto, aunque no se puedan definir en su totalidad, al menos hay que buscar tener definida la mayor parte del proyecto o bien, tener bien definida la primera parte que se va a desarrollar.

3. Diseño: Mantener el diseño simple y útil, realizando únicamente lo necesario para ayudar a entender lo que se desarrollará, apoyándose de diagramas de UML. Existen muchos tipos de diagramas UML, pero en realidad no es necesario utilizarlos todos, sobre todo en un desarrollo ágil, muchas veces con los diagramas de flujo, de secuencia o ambos bien diseñados es más que suficiente como base para la codificación.

4. Arquitectura: Escoger la arquitectura adecuada es un punto más que importante, si no existe una arquitectura dentro de la empresa, sin lugar a dudas empezaría a diseñarla antes de desarrollar cualquier proyecto trascendente.

5. Programación: La programación es únicamente una parte del proyecto, el error más común en las empresas pequeñas es aventarse a programar antes de seguir las prácticas de los puntos anteriores. Existen muchas prácticas de programación que hay que seguir, como son el uso de nomenclaturas, pruebas de humo, revisión en pares y las propias de cada lenguaje y herramienta en la que se programe.

6. Administración de la configuración: Consiste en administrar las versiones del proyecto y de todos los artefactos relacionados al proyecto, por ejemplo la documentación del proyecto.

7. Testing: El testing es en sí toda un área de conocimiento un tanto separada del desarrollo, ya que existen muchos tipos de pruebas, como puede ser de interacción con otras aplicaciones ajenas al

proyecto o de performance, pero que nos puede dar visibilidad de que el proyecto funcione o no. El error más común al realizar las pruebas, es encontrar uno o muchos errores y no haber programado previamente tiempo para regresar al desarrollo y corregir esos problemas.

8. Dependencias: Todos los proyectos tienen dependencias, por lo general los proyectos de desarrollo de software suelen seguir el mismo patrón de dependencias, por lo que es fácil prevenirlas, por ejemplo, por lo regular los desarrollos de software requieren de infraestructura tecnológica, contar con información para realizar testing, permisos de otros componentes o base de datos, interdependencias con otras aplicaciones, por mencionar algunas.

2. MATERIALES Y MÉTODOS

Para este proyecto se implementó un sistema web de préstamo de recurso para la Universidad de Pamplona extensión Villa del Rosario que permitirá al tanto docente sacar prestado un recurso para tener un apoyo a la hora de dictar su clase como a un administrativo en caso de la realización de un evento.

El desarrollo de esta aplicación web nos apoyaremos en las siguientes herramientas:

Toad Data Modeler: es una herramienta que nos ayuda a crear modelos de datos de alta calidad. Esta herramienta de modelado de datos permite implementar fácilmente los cambios exactos a las estructuras de datos en más de 20 plataformas diferentes.

Base de datos MySQL: es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

phpMyAdmin: es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.

PHP: es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Cloud 9 IDE: es un entorno de desarrollo integrado online. Soporta cientos de lenguajes de progra-

mación como PHP, Ruby, Python, JavaScript con node.js, entre otros. Permite a los desarrolladores comenzar inmediatamente con la codificación con pre-configuración de espacios de trabajo, colaborar con sus compañeros con funciones de codificación de colaboración y desarrollo web características como vista previa en vivo y pruebas de compatibilidad del navegador.

Al momento de crear nuestro diagrama del modelo relacional de la base de datos, el nombre de la tabla, ya sea de una (fig 1) o dos entidades (fig 2), se escribe todo en minúscula.

Figura 1. Nombre de la tabla con una sola entidad.



Figura 2. Nombre de la tabla con dos entidades.



Ahora para los atributos de cada entidad la escribiremos de la siguiente manera:

Si el nombre de la tabla hace referencia a una sola entidad se toman las cuatro primeras letras, seguidos de barra al piso (_) y el nombre del atributo (Figura 3).

Si el nombre de la tabla hace referencia a dos entidades, se toman las dos primeras letras de cada entidad, seguido de barra al piso (_) y el nombre del atributo (Figura 4).

Figura 3. Nombre los atributos que pertenecen a una sola entidad.

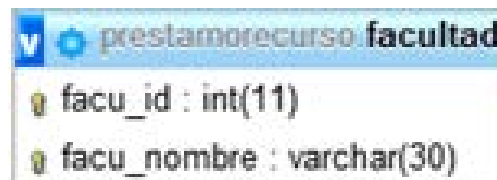


Figura 4. Nombre los atributos que pertenecen a dos entidades.



Una vez definida y armada la base de datos, nos iremos a nuestro IDE. Para esto primero debemos registrarnos y creamos nuestro espacio de trabajo (workspace). Después abrimos una terminal e instalamos phpMyAdmin con el siguiente comando:

PHPMYADMIN-CTL INSTALL

Para la etapa de desarrollo e implementación se dividió el trabajo de forma equitativa de acuerdo a los requerimientos que se especificaron en la etapa de análisis.

3.RESULTADOS

Después de un desarrollo arduo y colaborativo, y de repaso y aprendizaje en el lenguaje de HTML, JavaScript y PHP, se logró diseñar una aplicación web de préstamo de recurso prototipo de acuerdo a los requerimientos discutidos por parte de bienestar universitario de la Universidad de Pamplona extensión Villa del Rosario.

Figura 5. Diseño de la aplicación, Logueo.

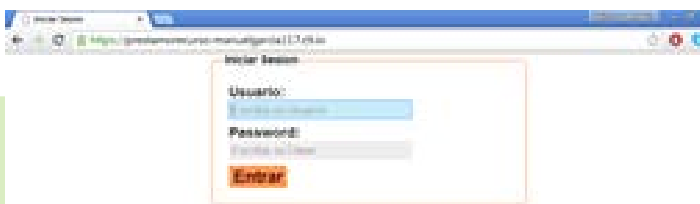


Figura 6. Página principal de la aplicación.



4.CONCLUSIONES

Para desarrollar un producto de software se requiere de todo un proceso. El cual esta predefinido por una diversidad de modelos. Es elemental tener en cuenta los modelos que se adapten para un proyecto requerido, para obtener como producto de ello una funcionalidad óptima del producto. Las mejores prácticas para un desarrollo de software se encontraran ligadas hacia la finalidad del proyecto que requiera de dicho producto.

5.BIBLIOGRÁFIA

Aplicativo web de Prestamos, <https://prestamorecurso-manuelgarcia117.c9.io/>

Modelos Administrativos , <http://www.virtual.unal.edu.co/cursos/sedes/manizales/4010014/Contenidos/Capitulos%20PDF/CAPITULOS%201%20AL%204.pdf>

Mobile Web Application Best Practices, <http://www.w3.org/TR/mwabp/>

Mejores prácticas para el establecimiento y aseguramiento de la calidad de software Por: Vega Lebrún Carlos Rivera Prieto Laura Susana García Santillán Arturo Serie Libros y Manuales: Informática Unidad Multidisciplinaria: CIET Libros de Texto/02/2008

Narvaez, L, Jurado, J. 2011, Implantación de buenas prácticas a un proceso de Desarrollo de Software - Una mirada Empresarial <http://jci.uni autonomia.edu.co/2011/2011-6.pdf>

Student Branch Development: Best Practices, https://www.ieee.org/membership_services/membership/students/st_best_practices_flyer.pdf