

Metodología para el desarrollo de aplicaciones web ajustada al contexto de los Productos Mínimos Viables

Methodology for web application development adapted to the context of Minimum Viable Products

Jaime Eduardo Alvarez Salazar, Jaime Eduardo Alvarez Salazar
 Ingeniería de sistemas, Universidad Autónoma de Bucaramanga, Bucaramanga, Colombia
 Correo-e: jalvarez37@unab.edu.co

Resumen— El sector de las Tecnologías de la Información y Comunicación ha sido identificado como un foco de desarrollo para la economía. Se impulsan iniciativas para la creación de negocios en este ramo, las cuales en su mayoría presentan limitaciones en recursos, experiencia y conocimiento. Es por ello que se hace necesaria una ejecución efectiva del proceso. Este artículo formula una metodología para el desarrollo de aplicaciones web como Producto Mínimo Viable (PMV), así como destaca el papel que desempeñar el Software Libre el cual, por sus características, puede soportar el proceso de manera efectiva.

Palabras clave— Gestión de Proyectos Web, Ingeniería Web, Metodología de Desarrollo, Producto Mínimo Viable, Software Libre.

Abstract— The Information and Communication Technologies sector has been identified as a development focus for the economy. Initiatives are promoted for the creation of businesses in this field, which mostly have limitations in resources, experience and knowledge. This is why an effective execution of the process is necessary. This article formulates a methodology for the development of web applications as Minimum Viable Product (MVP), as well as highlights the role of Free Software which, due to its characteristics, can effectively support the process.

Keywords— Development methodology, Free Software, Minimum Viable Product, Web Engineering, Web Projects Management.

I. INTRODUCCIÓN

El sector de las Tecnologías de la Información y Comunicación (TIC) ha sido identificado por diferentes naciones como uno de los principales generadores de desarrollo económico, de modo que los gobiernos promueven

iniciativas que contribuyan a la generación y consolidación de emprendimientos basados en este rubro.

Colombia no es ajena a esta realidad, por lo que se han llevado a cabo iniciativas como Apps.co [1], la cual es impulsada por el Ministerio de Tecnologías de la Información y las Comunicaciones, esta se suma a otros esfuerzos de carácter público y privado. El fin de estas iniciativas principalmente es el de impulsar ideas y encaminarlas hacia la creación de negocios rentables y/o de alto impacto; la contribución a esta causa se logra mediante consultoría, capacitaciones y/o el aporte de recursos económicos.

Con el fin de que dichas ideas sean llevadas a buen término, en la mayoría de ocasiones se plantea la construcción de un Producto Mínimo Viable (PMV), el cuál será un desarrollo preliminar de la aplicación que poseerá las características mínimas para que sea acogido por una cantidad considerable de usuarios.

El autor del artículo participó en una de las convocatorias de Apps.co, proceso en el cual pudo identificar que, mediante una metodología para PMV se podría optimizar el proceso de desarrollo, siendo esto fundamental ya que en este contexto suelen haber importantes limitaciones en tiempo, conocimiento y recursos económicos.

II. METODOLOGÍA

Si bien la metodología propuesta incorporará elementos de otras utilizadas como referente, especialmente las ágiles [2], el contexto en el que se plantea sea aplicada tiene las siguientes características:

- El equipo de trabajo está conformado por un número reducido de personas (entre 2 y 4), de las cuáles sólo algunas poseen conocimientos en desarrollo web.
- El equipo de trabajo no posee experiencia en proyectos previos, ni tiene conocimiento acerca de las metodologías de desarrollo.
- Se disponen de recursos económicos muy limitados, por lo que está fuera del alcance llevar a cabo acciones como contratar algún tipo de consultoría o adquirir licencias para herramientas de software propietario.
- El tiempo para desarrollar la solución es corto, por lo que es mandatorio el saber administrarlo, llevando a cabo acciones que efectivamente logren alcanzar los objetivos del proyecto.

Se plantea formular la metodología, dividiéndola en una serie de componentes que agrupan un conjunto de técnicas, actividades y herramientas. La aplicabilidad de cada uno de estos componentes está determinada por el estado actual del proyecto.

A. Componente de Levantamiento de Requerimientos.

Los requerimientos se establecerán a partir de la propuesta de valor la cuál se define en el modelo Canvas [3] del proyecto. Los requerimientos pueden definirse mediante historias de usuario [4] para que cada hecho se presente de forma breve y concisa.

Una historia de usuario se conforma de:

- Nombre breve y descriptivo.
- Especificación de la funcionalidad de acuerdo a como la describe el usuario.
- Criterio de validación que determinará la aceptación por parte del cliente de la funcionalidad descrita.

B. Componente de Análisis y Diseño.

Se recomienda en sobremedida elaborar mockups [2][5] para las diferentes interfaces que desplegará la aplicación. Este recurso visual permitirá incrementar notablemente el entendimiento del sistema por parte de los potenciales usuarios, los cuales emitirán su respectiva retroalimentación.

Cuando el proceso que se desea modelar, requiere de una serie de interacciones complejas con el sistema se puede recurrir a la elaboración de diagramas de secuencia para ilustrar con mayor claridad los pasos y el comportamiento del mismo. Como complemento, podría considerarse la elaboración de diagramas de casos de uso.

En esta etapa, también debería determinarse el conjunto de tecnologías que se adoptarán para llevar a cabo la construcción de la solución de software. Se recomienda altamente la adopción de frameworks basados en Software Libre [6] tanto para construir el entorno front (jQuery,

Bootstrap, AngularJS) como el backend (Laravel, CodeIgniter, Yii). Esto presenta las siguientes ventajas:

- Se agiliza en gran medida el proceso de desarrollo, permitiendo así realizarlo en un menor tiempo.
- Al ser Software Libre no debe incurrirse en costos de licenciamiento.
- En la construcción de estas herramientas han participado desarrolladores experimentados, los cuales han aplicado conceptos avanzados de Ingeniería de Software y Seguridad de la Información.

La decisión al momento de escoger el conjunto de herramientas para la construcción de la solución de software es importante, ya que, dependiendo de las elecciones realizadas, se tendrá un mayor acceso a documentación y a colaboración en entornos como chats o foros. La cantidad de estos recursos generalmente es proporcional al número de usuarios que hayan adoptado estas tecnologías. Para orientar este proceso de selección, se tomarán como referencia investigaciones realizadas por entes académicos y empresariales que han logrado determinar las tecnologías que actualmente tienen una mayor acogida. IEEE [7] realiza una investigación anual en el que indaga el grado de adopción de los diferentes lenguajes de programación.

Esta investigación formula una métrica denominada Spectrum Ranking, la cual es calculada mediante el análisis ponderado de 12 sitios web en donde se menciona el lenguaje de programación en cuestión. A una mayor frecuencia de menciones se entiende que es mayor el nivel de acogida.

Al filtrar por tipos de lenguajes a solo Web, se obtienen los siguientes resultados:

TABLA I

Posición	Lenguaje	Spectrum Ranking
1	Java	98,1
2	Python	98
4	PHP	82,8
5	JavaScript	82,2
6	Ruby	74,5

Por otra parte, la empresa Sitepoint [8], conduce una encuesta acerca del nivel de adopción de los Frameworks BackEnd, a continuación se muestran las 5 posiciones del ranking de acuerdo a su medición mas reciente:

TABLA II

Posición	Lenguaje
1	Laravel
2	Symfony
3	Nette
4	CodeIgniter
5	Yii

Respecto al nivel de adopción de los Frameworks FrontEnd, puede tomarse como referencia la investigación realizada por Ashley Nolan [9], de acuerdo a esta las primeras 5 posiciones del ranking son:

TABLA III

Posición	Lenguaje
1	jQuery
2	AngularJS
3	React
4	Backbone
5	Ember

C. Componente de Desarrollo.

Antes de iniciar el proceso de codificación, deben establecerse una serie de buenas prácticas y patrones [10] con el fin de asegurar que los elementos desarrollados sean efectivos, eficientes y fáciles de mantener. Dentro de estos se destacan:

- Notación PSR [11]. Establece una serie de convenciones y buenas prácticas en lo que se refiere a la escritura de código. Se recomienda al menos la adopción de los estándares PSR-1 y PSR-2.
- Enfoque DRY (Don't Repeat Yourself). Plantea el evitar desarrollar elementos de código cuya funcionalidad sea duplicada.
- Modelo Vista Controlador (MVC). Este patrón de diseño plantea la separación de los diferentes elementos de código según su funcionalidad. El modelo se encargará de interactuar directamente con la base de datos, el controlador se encargará de gestionar las peticiones y redireccionarlas bien sea al modelo o la vista, mientras que esta se encarga de presentar las diferentes interfaces con las que el usuario interactúa.
- Singleton. Se utiliza para restringir el número de veces que se crea un objeto en específico, permitiendo solamente una única instancia del objeto. Se recomienda utilizarlo por ejemplo, para la comunicación con la Base de Datos manteniendo una única conexión abierta y así mantener el rendimiento del sistema.
- Decorator. Se utiliza cuando deben alterarse ciertas funcionalidades de un objeto, sin alterar las funcionalidades del objeto original. Es útil frente a requerimientos que presenten cierto nivel de incertidumbre.

D. Componente de Pruebas.

Con el fin de asegurar un alto nivel de calidad en la solución final, se recomienda ejecutar una serie de pruebas que las que pueden identificarse dos tipos:

- Pruebas unitarias: están enfocadas hacia las unidades mas pequeñas dentro de una aplicación de software. Para llevar a cabo estas pruebas puede utilizarse la herramienta PHPUnit.
- Pruebas funcionales: el propósito de estas pruebas es validar que la característica de la aplicación está arrojando la salida esperada. Estas pruebas pueden realizarse a través de la herramienta Selenium.

III. IMPLEMENTACIÓN DE LA METODOLOGÍA EN EL DESARROLLO DE UN PMV

Se procedió a la construcción de una aplicación web, en el que el proceso estuvo guiado a los postulados enunciados en los puntos anteriores. Adicionalmente se estableció que todas las herramientas utilizadas estuviesen basadas en Software Libre. Se presenta a continuación el listado mencionando el nombre de la herramienta y su propósito en el desarrollo del proyecto:

TABLA IV

Sistema	Propósito	Componente
Ubuntu	Sistema operativo para desarrollo.	Levantamiento de Requerimientos, Análisis y Diseño, Desarrollo, Pruebas
LibreOffice	Paquete de software para creación de documentos	Levantamiento de requerimientos
Pencil Project	Herramienta para la construcción de diagramas	Análisis y Diseño
MySQL Workbench	Herramienta para el diseño de bases de datos MySQL	Análisis y Diseño
PHP	Lenguaje de Programación	Desarrollo
MySQL	Sistema de Gestión de Base de Datos	Desarrollo
Apache	Servidor Web	Desarrollo
Yii	Framework de Desarrollo BackEnd	Desarrollo
Bootstrap	Framework de Desarrollo FrontEnd	Desarrollo
GIMP	Herramienta para edición de imágenes	Desarrollo
Inkscape	Herramienta para la edición de gráficos vectoriales	Desarrollo
Debian	Sistema operativo para ambiente de pruebas y producción	Desarrollo, Pruebas
Netbeans	Entorno Integrado de Desarrollo	Desarrollo, Pruebas
PHPUnit	Pruebas Unitarias	Pruebas
Selenium	Sistema de pruebas automatizadas	Pruebas

El sistema a desarrollar estaba destinado a la Secretaría de Educación de la ciudad de Manizales el cuál debe satisfacer los siguientes requerimientos:

- Mantener un control de las matrículas correspondientes a los estudiantes de las instituciones educativas. De esta forma, mediante el sistema se puede realizar el proceso de transferir un estudiante de una institución a otra y mantener un control sobre este proceso.
- Mantener un registro actualizado y detallado de todos los elementos involucrados en el proceso educativo, estos son: Institución > Sede > Jornada > Grado > Grupo > Estudiante.

Respecto a las actividades que se han realizado durante los diferentes Sprints ejecutados se pueden mencionar:

- Levantamiento de requerimientos por medio de entrevistas a un funcionario de la Secretaría de Manizales.
- Implementación de las funcionalidades especificadas.

CONCLUSIONES Y RECOMENDACIONES

La adopción de una metodología de desarrollo es un factor determinante para asegurar el éxito de un proyecto de software.

En el contexto del desarrollo de los MVP, las acciones llevadas a cabo deben ser cuidadosamente planeadas y ejecutadas ya que los recursos de los que se disponen generalmente son limitados.

La utilización de Software Libre en este tipo de proyectos, trae grandes ventajas, ya que mas allá de su costo cero en licenciamiento, las herramientas de las que actualmente se pueden disponer favorecen a la agilidad y calidad en la construcción de la solución de software.

REFERENCIAS

- [1] (2016) Apps.co Sitio Web. [En línea]. Disponible en <http://apps.co>
- [2] J. Rivero, J. Griggera, G. Rossi, E. Robles, N. Koch. (2011). Improving Agility in Model-Driven Web Engineering [En línea]. Disponible en <http://ceur-ws.org/Vol-734/PaperVision05.pdf>.
- [3] A. Cowan (2014). The 20 minute business plan: business model canvas made easy [En línea]. Disponible en <http://www.alexandercowan.com/business-model-canvas-templates/>
- [4] M. Cohn (2016). User Stories. [En línea]. Disponible en <https://www.mountaingoatsoftware.com/agile/user-stories>
- [5] J. Zhang y J. Chung (2014). Mockup-driven Fast-prototyping Methodology for Web Application Development. [En línea]. Disponible en

<https://pdfs.semanticscholar.org/222a/fe8255d3c6e7e21ac3bf603978a7bfe06838.pdf>

- [6] Categoría:Frameworks para aplicaciones web (2016). [En línea]. Disponible en https://es.wikipedia.org/wiki/Categor%C3%ADa:Frameworks_para_aplicaciones_web
- [7] N. Diakopoulos y S. Cass (2016). Interactive: The Top Programming Languages 2016. [En línea]. Disponible en <http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2016>
- [8] B. Skvorc (2015). The Best PHP Framework for 2015: SitePoint Survey Results. [En línea]. Disponible en <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
- [9] A. Nolan (2015). The State of Front-End Tooling. [En línea]. Disponible en <https://ashleynolan.co.uk/blog/frontend-tooling-survey-2015-results>
- [10] M. Lang y B. Fitzgerald (2006). New Branches, Old Roots: A Study of Methods and Techniques in Web / Hypermedia Systems Design. Disponible en <https://aran.library.nuigalway.ie/bitstream/handle/10379/409/Information%20Systems%20Management%2023%283%29%202006.pdf?sequence=1&isAllowed=y>
- [11] (2016) PHP Standards Recommendations. [En línea]. Disponible en <http://www.php-fig.org/psr/>

GLOSARIO

- Mockup: representación gráfica de una interfaz perteneciente a un sistema a desarrollar. Esto permite acordar los elementos visuales y la disposición de los mismos.
- Backend: hace referencia a la capa de software encargada de acceder a datos y realizar operaciones con los mismos. El procesamiento es realizado por un servidor y el resultado enviado al navegador en un formato interpretable por el mismo tal como HTML, JSON o XML.
- Frontend: capa de software encargada de la presentación del sistema. Esta despliega los diferentes elementos visuales que permitirán al usuario interactuar con el sistema. En aplicaciones web todo el procesamiento es realizado por el navegador.