

Producto Mínimo Viable: Un Ajuste al Proceso de Desarrollo de Software

Minimum Viable Product: An Adjustment to the Software Development Process

Leydi Diana choque Sarmiento

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

164909@unsaac.edu.pe

Carlos Eduardo Inca Cruz

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

174912@unsaac.edu.pe

Carlos Enrique Quispe Chambilla

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

174447@unsaac.edu.pe

Erick Andrew Bustamante Flore

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

171943@unsaac.edu.pe

Alex Helder Huancara Ccolque

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

174911@unsaac.edu.pe

Ronadinho Vega Centeno Olivera

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

140934@unsaac.edu.pe

Richard Mikhael Pereira Chinchero

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

171916@unsaac.edu.pe

Daniel Eduardo Sarco Jacinto

Departamento Académico de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco
Cusco, Perú

174452@unsaac.edu.pe

Resumen — El proyecto siguiente describe el proceso de implementación de un proyecto mínimo viable de un sistema de tutorías mediante metodologías ágiles, comenzando con la captura de requisitos a través de historias de usuario, para posteriormente hacer el análisis y diseño a través de diagramas SD y SR, así como la determinación de software para el desarrollo del MVP.

Seguimos una serie de buenas prácticas de programación, gestión de costos a través de COCOMO, Gestión de riesgos con el modelo de Colofello, patrones de diseño de software, code smells, métricas de proyectos. Posteriormente en el diseño del MVP utilizamos React, bootstrap y HTML para el frontend, PHP para el backend, la base de datos se implementó en MySQL.

Palabras Clave - sistema de tutoría ; producto mínimo viable, desarrollo web; MVP.

Abstract — The following project describes the process of implementing a minimum viable project of a mentoring system using agile methodologies, starting with the capture of requirements through user stories, to later do the analysis and design through SD and SR diagrams, as well as the determination of software for the development of the MVP.

We follow a series of good programming practices, cost management through COCOMO, Risk management with the Colofello model, software design patterns, code smells, project metrics. Later in the design of the MVP we used React, bootstrap and HTML for the frontend, PHP for the backend, the database was implemented in MySQL

Keywords -tutoring system; minimum viable product, Web development, MVP

I. INTRODUCCIÓN

Frente a esta problemática de la falta de un sistema de tutorías en la escuela profesional de Ingeniería Informática y de sistemas se desea la implementación de un MVP de un sistema de tutorías con el fin de apoyar a la formación profesional, el cual servirá para orientar al estudiante, facilitar el cumplimiento de sus objetivos, apoyo en problemas académicos, personales y laborales.

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 cual será un desarrollo preliminar de la

aplicación que poseerá las características mínimas para que sea acogido por una cantidad considerable de usuarios.

Antes de la implementación de un MVP se tuvieron estudios previos sobre diferentes temas que conllevan la Ingeniería de Software, se observó que mediante una metodología para MVP 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), el contexto en el que se plantea sea aplicada tiene las siguientes características:

- El equipo de trabajo está conformado por 8 personas, de las cuáles sólo algunos poseen escasos conocimientos en desarrollo web.
- El equipo de trabajo no posee experiencia en proyectos previos de desarrollo web.
- El equipo tiene conocimiento acerca de las metodologías de desarrollo.
- El tiempo para desarrollar la solución es corto, por lo que es importante saber administrarlo, llevando a cabo acciones que efectivamente logren alcanzar los objetivos del proyecto.

Se plantea formular la metodología, dividiéndola en una serie de 4 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.

1. *Componente de Levantamiento de Requerimientos*

Los requerimientos se establecerán a partir de la propuesta de valor la cuál se define en el modelo de tutorías de la UNSAAC [1] 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.

2. *Componente de Análisis y Diseño*

Este recurso visual permitirá incrementar notablemente el entendimiento del sistema por parte de los potenciales usuarios, los cuales emitirán su respectiva retroalimentación.

En esta etapa, usaremos los diagramas SD y SR para modelar el sistema a un nuevo tipo de formato. También para tener un mejor conocimiento del sistema que queremos desarrollar modelamos los casos de usos con mayor importancia tomando como referencia la técnica de priorización de requisitos llamado MoSCoW, diagramas de análisis y un diagrama de componentes.

También se determinó 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 tanto para construir el entorno frontend (jQuery, React, Bootstrap, AngularJS) como el backend (Laravel, CodeIgniter, Yii, Nodejs)

3. *Componente de Desarrollo*

Antes de iniciar el proceso de codificación, deben establecerse una serie de buenas prácticas y patrones [2] con el fin de asegurar que los elementos desarrollados sean efectivos, eficientes y fáciles de mantener. Sin embargo también debemos mencionar algunos tipos de estudios al código del proyecto de entre los cuales tenemos:

- **Backlog** es una lista de trabajo ordenado por prioridades para el equipo de desarrollo que se obtiene de la hoja de ruta y sus requisitos. Los elementos más importantes se muestran al principio del backlog del producto para que el equipo sepa qué hay que entregar primero.
- **Cocomo** es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.
- **Code Smells** son una serie de síntomas en el código que nos vienen a indicar que tal vez no se están haciendo las cosas de una forma del todo correcta, lo que puede llevar a que haya algún problema a futuro y un problema de trasfondo.
- **Métricas de Proyectos** permiten monitorizar un producto para determinar su nivel de calidad aunque, el seguimiento que este tipo de medidas permiten llevar a cabo brinda la oportunidad de conocer muchas más cosas de una solución.
- **Requisitos no Funcionales** describen características de funcionamiento, por eso suelen denominarse Atributos de calidad de un sistema.

4. *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 más 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 DEL PMV

Se procedió a la construcción de una aplicación web para la gestión de las tutorías, 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 con el nombre de la herramienta y su propósito en el desarrollo del proyecto:

Sistema	Propósito	Componente
Window 10	Sistema operativo para desarrollo.	Levantamiento de Requerimientos, Análisis y Diseño, Desarrollo, Pruebas
Canva	Presentación de resultados	Presentación
Documentos de google	Paquete de software para creación de documentos	Levantamiento de requerimientos
Google Chrome	Buscar información	Buscador
Lucidchart, piStar tool	Herramienta para la construcción de diagramas	Análisis y Diseño y Diseño
Google meet	Realización de reuniones para coordinación	Organizador

WhatsApp Web	Comunicación y coordinación	Comunicación
Github	repositorio para trabajo en equipo	Desarrollo
MySQL	Sistema de Gestión de Base de Datos	Desarrollo
PHP	Lenguaje de Programación	Desarrollo
HTML	Lenguaje de marcado	Desarrollo
XAMPP	Gestión de bases de datos MySQL, el servidor web Apache	Desarrollo
Visual Studio Code	editor de código redefinido y optimizado	Desarrollo
React	Crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página.	Desarrollo

El sistema a desarrollar está destinado a la escuela profesional de Ingeniería Informática y de Sistemas de la Universidad Nacional de San Antonio Abad del Cusco el cuál debe satisfacer los siguientes requerimientos:

- Gestionar la asignación de tutores para cada estudiante de la escuela profesional. .
- Designar un horario y aula para las sesiones de tutorías.
- Mantener un registro actualizado y detallado de todas las sesiones de tutorías.

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

- Levantamiento de requerimientos por medio de la revisión del reglamento de tutorías de la escuela profesional.
- Levantamiento de requerimientos por medio de entrevistas a la docente del curso de Ingeniería de software escuela profesional.
- Implementación de las funcionalidades especificadas.

IV. RESULTADOS OBTENIDOS

Se logró la implementación del PMV del sistema de tutorías para la Escuela Profesional de Ingeniería Informática y de Sistemas, cumpliendo con el requisito principal de asignar un tutor a un grupo de alumnos.

La realización de un PMV del sistema web propuesto, permitió cumplir con el desarrollo del software en un plazo de 6 semanas, con un trabajo de 4 horas semanales aproximado por cada integrante del grupo.

La estimación de costos por COCOMO no fue útil para estimar el costo que tendría el proyecto antes de ser terminado.

La utilización de software libre no es impedimento para el desarrollo de aplicaciones web.

V. CONCLUSIONES

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 más 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.

VI. TRABAJO A FUTURO

Al ser un producto mínimo viable uno de los trabajos futuros es mejorar el proyecto incluyendo más funcionalidades y ampliar la base de datos. Otro de los trabajos a futuro es ampliar la cantidad de datos para que tenga conexión con el centro de cómputo y que sea global para toda la UNSAAC.

AGRADECIMIENTOS

Agradecemos a la universidad por el reglamento de dónde sacamos la mayor parte de información para elaborar nuestro proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Roxana L. Q. P. (2021) Repositorio Tutoría <http://bit.ly/c1-tutoria>:
- [2] 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>
- [3] A. Cowan (2014). The 20 minute business plan: business model canvas made easy [En línea]. Disponible en <http://www.alexandercowan.com/business-modelcanvas-templates/>
- [4] M. Cohn (2016). User Stories. [En línea]. Disponible en <https://www.mountaingoatsoftware.com/agile/usersories>
- [5] Cesar D. Alcolea (2019). Code Smells y Deuda Técnica. Disponible en <https://openwebinars.net/blog/code-smells-y-deuda-tecnica/>
- [6] Repositorio GitHub: <https://github.com/Mijnboodssj999/Grupo3-Renovados>