DESARROLLO DEL SISTEMA DE GESTIÓN DE PROYECTOS DE INVESTIGACIÓN DE LA UNIVERSIDAD COOPERATIVA DE COLOMBIA SEDE CALI

JOSE BRANDON HENAO TUNJO
WALTER JOSE DEVIA LOZANO
WILLIAM ALEJANDRO GIRALDO ZULUAGA

UNIVERSIDAD COOPERATIVA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SANTIAGO DE CALI
2016

DESARROLLO DEL SISTEMA DE GESTIÓN INTEGRAL DE PROYECTOS DE INVESTIGACIÓN DE LA UNIVERSIDAD COOPERATIVA DE COLOMBIA SEDE CALI.

JOSE BRANDON HENAO TUNJO
WALTER JOSE DEVIA LOZANO
WILLIAM ALEJANDRO GIRALDO ZULUAGA

ANTEPROYECTO

JOSEFINA MAYRA DE LLANO FELIU
DOCENTE

UNIVERSIDAD COOPERATIVA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
SANTIAGO DE CALI
2016

CONTENIDO

pág.

GLOSARIO

BACK-END**:** En el software con arquitectura cliente-servidor, la parte que corresponde al servidor es llamada back-end, es aquella unidad con la capacidad de procesamiento pesado, contenedora de la lógica de negocio, manejadora de los datos y responsable de gestionar las peticiones del cliente.

B.I: siglas para inteligencia de negocio cuya traducción al inglés es Business Intelligence, consiste en tomar un volumen considerable de datos para luego realizar análisis que permita observar el comportamiento de la información como tendencias, falencias, virtudes, entre otros aspectos que se pueden evaluar.

DISEÑO RESPONSIVO: Tendencia de creación de páginas web que se adaptan automáticamente al tamaño de la pantalla de todo dispositivo, como pantallas de ordenadores de escritorio hasta smartphone y tablets.

FRONT-END: Es la contraparte del back-end; es el lado del cliente, corresponde las interfaces con el usuario final y su unidad de procesamiento.

1 IDENTIFICACIÓN DEL PROBLEMA

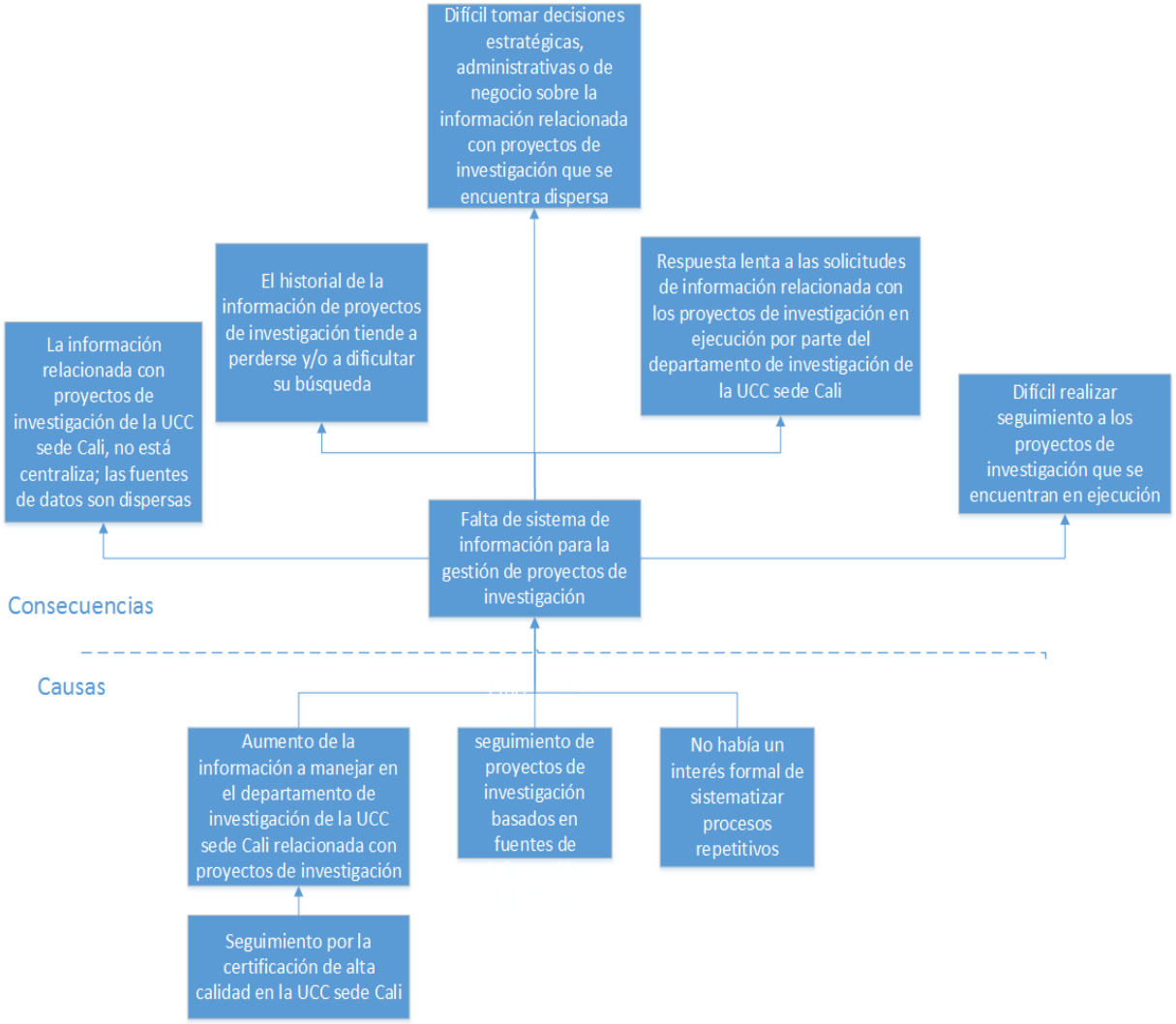
1.1 PLANTEAMIENTO DEL PROBLEMA

La Universidad Cooperativa de Colombia (UCC) sede Cali actualmente trabaja con intensidad para lograr la certificación de alta calidad; uno de los requisitos para lograrlo es presentar actividades y resultados relacionados con proyectos de investigación a las entidades reguladoras de la educación superior. Para ello dedica recursos en procesos de gestión de proyectos de investigación, donde se involucran la coordinación y la dirección del departamento de investigación de la universidad. Dichos procesos se basan hasta la fecha en distintas fuentes de información dispersas y de formato variado; con el aumento de las actividades y planes de trabajo de la gestión de proyectos de investigación junto con los procesos relacionados se presentan los siguientes inconvenientes.

- La información relacionada con los proyectos de investigación no se encuentra centralizada; las fuentes de datos son dispersas.
- El historial de la información de los proyectos de investigación tiende a perderse y/o a dificultar su búsqueda.
- Se dificulta el seguimiento a los proyectos de investigación que se encuentran en ejecución.
- Se dificulta tomar decisiones estratégicas, administrativas o de negocio sobre la información relacionada con los proyectos de investigación que se encuentra dispersa.
- Respuesta lenta a las solicitudes de información relacionada con los proyectos de investigación en ejecución por parte del departamento de investigación.

Todo lo anterior se presenta de manera resumida en la siguiente figura, en donde se presenta la identificación del problema abordada desde una perspectiva causa-consecuencia o árbol de problema.

Figura 1. Árbol del problema.



BRANDON, José; DEVIA, Walter; GIRALDO, William. SGPI.

Del árbol del problema se obtiene el punto central que identifica la problemática y al mismo tiempo la propuesta para su solución; a la fecha la UCC sede Cali no cuenta con un sistema de información en funcionamiento que soporte los procesos relacionados con la gestión y seguimiento de los proyectos de investigación de la universidad, de lo cual se propone desarrollar un sistema de gestión de proyectos de investigación (SGPI por sus siglas), que además de agilizar dichas actividades, provea un punto central para la información de los proyectos de investigación aprovechado por el mismo sistema para su procesamiento en BI.

1.2 FORMULACIÓN DEL PROBLEMA

¿Cómo sistematizar los procesos asociados a la gestión de los proyectos de investigación del departamento de investigación de la Universidad Cooperativa de Colombia sede Cali?

2 OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar el sistema de gestión de proyectos de investigación para el departamento de investigación de la Universidad Cooperativa de Colombia sede Cali.

2.2 OBJETIVOS ESPECÍFICOS

- Levantar requerimientos junto con la jefatura del departamento de investigación de la UCC sede Cali a partir de actividades de contextualización de los procesos asociados a la gestión de proyectos de investigación.

- Diseñar el sistema de información mediante diagramas de arquitectura de software a partir de los requerimientos abstraídos.

- Desarrollar el sistema de información diseñado en un lenguaje de alto nivel orientado a web siguiendo la metodología de desarrollo SCRUM.

- Realizar pruebas de software exhaustivas y correcciones finales al sistema de información desarrollado.

- Documentar el funcionamiento del sistema de información para servir como guía de usuario final, guía al desarrollador y guía de soporte técnico.

- Implementar la primera versión final del sistema de información en un entorno de producción para el departamento de investigación de la UCC sede Cali.

3 JUSTIFICACIÓN

Entre las acciones tomadas por la UCC sede Cali para la adquisición de la certificación de alta calidad se tiene un fortalecimiento en el departamento de investigación, incrementando a la vez el volumen de información de proyectos de investigación. Dado los procesos que se tienen a la fecha asociados con la gestión de los proyectos de investigación, en el departamento se generan tiempos largos de respuesta a solicitudes de información relacionada a los proyectos de investigación, sobrecarga de trabajo en los agentes administrativos y complicación en el seguimiento de los diversos proyectos de investigación.

Tanto los integrantes de los grupos de investigación como los agentes administrativos del departamento de investigación de la UCC sede Cali serán beneficiados al tener un sistema de información que soporte la gestión de sus proyectos de investigación, sistema que permitirá llevar el seguimiento constante de los proyectos en curso, ahorrar tiempo de búsqueda de información relacionada con cualquier proyecto de investigación al tener centralizada la misma y procesar y presentar dicha información de manera tal que sea útil para la toma de decisiones administrativas (BI). Con este sistema de información se promueve el crecimiento del departamento de investigación en materia de proyectos de investigación al contar con una herramienta que provee mayor tiempo de respuesta en los procesos asociados a la gestión de los proyectos de investigación.

4. MARCO REFERENCIAL

4.1 MARCO TEÓRICO Y CONCEPTUAL

Dada la naturaleza del sistema de información a desarrollar (aplicación Web) y el énfasis que tienen los desarrolladores en un patrón de trabajo guiado primero por la construcción de diseños que representan la arquitectura de la aplicación antes de escribir el código fuente, las referencias teóricas-conceptuales siguen un orden similar y orientadas a teorías y conceptos relacionados con el desarrollo Web: primero se tiene la arquitectura multicapa que indica un orden de alto nivel de abstracción a la arquitectura de la aplicación, se sigue con el patrón Modelo-Vista-Controlador que *aterriza* el concepto de la arquitectura multicapa a un patrón de trabajo moderno y finalmente se tiene el Modelo de Variabilidad Arquitectónica y Tecnológica que ayuda en la elección de las tecnologías de desarrollo a usar en función del orden de la arquitectura elegida. El marco referencial termina con la metodología de desarrollo de software elegida por el equipo desarrollador.

4.1.1 Arquitectura multicapa

Para manejar la complejidad de la aplicación se emplea arquitectura multicapa, específicamente tres capas: presentación, lógica y persistencia o de base de datos. Los beneficios de usar este tipo de arquitectura se reflejan en el bajo acoplamiento de sus componentes “debido a que esta característica permite realizar cambios en los servicios, sin tener que revisar cambios en todos los componentes” [1].

Este tipo de arquitectura facilita el desarrollo web al distribuir el código subyacente en el servidor y en el conjunto de dispositivos clientes de la misma arquitectura cliente-servidor que comprende la naturaleza Web. La primera capa que es presentación se ejecutará en los clientes (aunque su código subyacente en realidad se aloja del lado del servidor), mientras que la segunda y tercera capa que son lógica y persistencia respectivamente tendrán la ejecución de su código subyacente y alojamiento del lado del servidor. La arquitectura resultante es una arquitectura multicapa que divide el código en tres capas reduciendo el acoplamiento de las mismas para un mejor manejo de la complejidad del desarrollo de la aplicación, además posibilita la delegación de tareas por capa a los desarrolladores, por ejemplo, delegando el desarrollo de la capa de presentación al integrante del

---

[1] FLOREZ FERNANDEZ, Héctor Hugo. “Arquitectura multicapa mediante AJAX y ORM”. Revista vínculos Vol. 7 No. 1, abril 30 de 2010, p. 4. Internet: <http://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4154/5818> [consultado el 27 de agosto del 2016]

proyecto con mejores habilidades en front-end y el desarrollo de las capas 2 y 3 a los integrantes con mayor conocimiento en back-end; aunque el mayor porcentaje del desarrollo de todas las capas serán trabajadas en conjunto por todos los integrantes del proyecto al mismo tiempo. No obstante, el concepto de la arquitectura multicapa por sí solo no es suficiente en sentido práctico, solamente provee un orden de alto nivel de abstracción al código subyacente de la aplicación. A falta de tal practicidad el patrón Modelo-Vista-Controlador se presenta a continuación.

4.1.2 Modelo-Vista-Controlador (MVC)

Se añade el patrón Modelo-Vista-Controlador (MVC) a la arquitectura multicapa por la semejanza en la división de las unidades lógicas de la aplicación; “El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento” [2] dónde el Modelo contiene el código que trata la interacción con la base de datos como la interfaz comunicadora con el sistema gestor de base de datos, clases que representan las entidades de la base de datos y librerías o frameworks con fines similares; los Controladores contienen el código que trata la lógica de negocio como validaciones de los datos y comunicaciones entre Vistas y Modelo; y por último las Vistas son las interfaces gráficas a presentar al usuario. Con el patrón MVC se orienta la asignación de trabajos a las unidades lógicas, así como la comunicación entre las mismas. Nótese como cada división que comprende el patrón MVC calza con las capas de la arquitectura multicapa, de lo que se obtiene como resultado un orden en la distribución del código subyacente (carpetas, ficheros, librerías, framework) en capas definidas por la arquitectura multicapa y la delegación de trabajos a dichas capas por el patrón MVC.

Para llevar la teoría y los conceptos de la arquitectura multicapa y del patrón MVC se requiere de tecnologías que lo implementen en la práctica, es aquí donde el Modelo de Variabilidad Arquitectónica y Tecnológica se presenta como ayuda en la elección de las tecnologías que cumplan con las ideas abstractas de la arquitectura.

4.1.3 Modelo de variabilidad arquitectónica y tecnológica

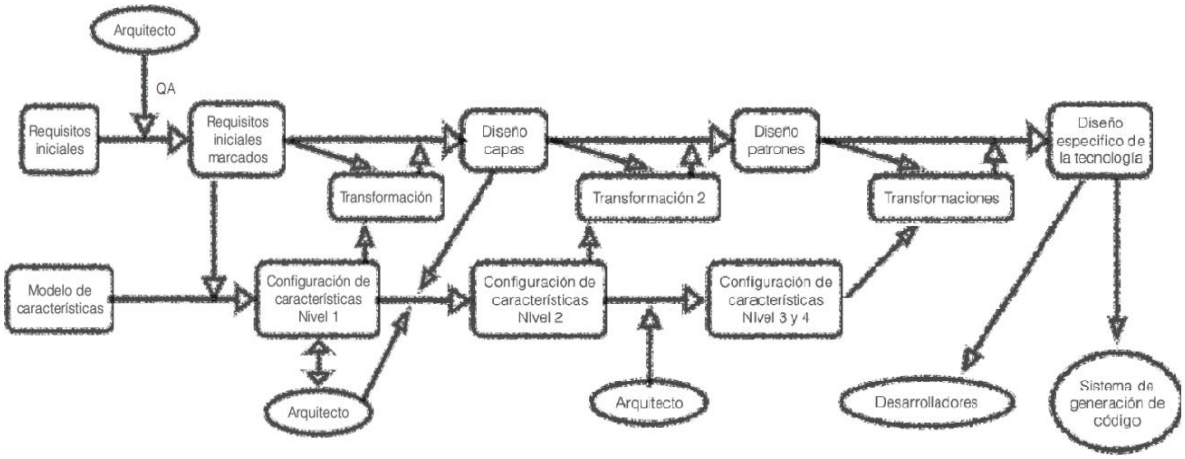
Una definición de arquitectura y de patrones de diseño no son suficientes para comenzar un desarrollo, las tecnologías que se ajusten a la arquitectura y al patrón se deben definir de la misma manera, esto es frameworks del lenguaje de programación a utilizar. Para unir ambos elementos (arquitectura y tecnología implementadora) se hace uso del *modelo de variabilidad en arquitectura*

---

[2] FERNANDEZ ROMERO, Yenisleidy, DÍAS GONZALES, Yannete. “Patrón Modelo-Vista-Controlador”. Revista Telem@tica. Vol. 11. No. 1, enero – abril del 2012, p. 47 - 57.

multicapa [3] la cual ofrece un “modelo de características que captura la variabilidad arquitectónica y tecnológica” [3], en otras palabras, un modelo para adoptar los requerimientos a la arquitectura multicapa y acoplar los patrones de diseño junto con las tecnologías que lo implementan. La figura 2 presenta dicho modelo.

Figura 2. Modelo de captura de variabilidad arquitectónica y tecnológica.



Fuente [3].

La entidad “Arquitecto” que presenta el modelo es aplicada por todos los integrantes del proyecto. El modelo puede ser apreciado como un diagrama de flujo de izquierda a derecha donde tiene sus comienzos con los requisitos levantados; se sigue con la transformación de los mismos a la arquitectura multicapa en donde se distribuye el cumplimiento de cada uno de ellos en una capa determinada, los niveles de configuración de características hacen referencia a las adaptaciones que se hacen al objeto de estudio que se trata de transformar, en este caso las adaptaciones que se les hacen a los requisitos para su cumplimiento en las capas concebidas; se sigue con la inclusión de los patrones de diseño y la adaptación de los mismos a la arquitectura multicapa, en este caso el patrón MVC calza fácilmente con las tres capas concebidas y por tanto el nivel 2 de configuración de características no representa mucho esfuerzo.

---

[3] ALONSO GRACÍA, José; BERROCAL, Javier; MURILLO, Juan Manuel. "Modelado de la variabilidad en arquitecturas multicapa". Escuela Politécnica, Universidad de Extremadura. Internet: <https://www.researchgate.net/profile/Javier_Berrocal2/publication/235408808_Modelado_de_la_Variabilidad_en_arquitecturas_Multicapas/links/0fcfd5118aa485e8ac000000.pdf> [consultado el 27 de agosto del 2016]

Se sigue con la última transformación y configuración de características de nivel 3 y 4 lo cual hace referencia al acople de las herramientas a implementar como frameworks y librerías al patrón MVC. Al final del seguimiento del modelo se tienen los requisitos adaptados a una arquitectura con tecnologías que la implementan listas para su uso práctico en la metodología de desarrollo de software, la cual en este caso es la metodología de desarrollo de software ágil SCRUM.

4.1.4 Scrum

Dentro del desarrollo de software se encuentran varias metodologías de desarrollo, los cuales proveen orden y buenas prácticas en el ciclo del desarrollo del software, en este caso se tomó la metodología de desarrollo ágil Scrum, cuyo método permite adoptar una estrategia de desarrollo incremental (i.e. requerimientos adicionales en cualquier punto del tiempo del desarrollo del software), en lugar de seguir fases separadas de planificación y ejecución resistentes a los cambios como dicta las metodologías de desarrollo ordinarias. Cabe destacar que la metodología Scrum basa la calidad del resultado más en el conocimiento integral de las personas y en los equipos que se conforman en vez de la calidad de los procesos empleados, es decir está orientado al trabajo en equipo y al crecimiento personal y colectivo. Estos dos últimos aspectos, desarrollo incremental y calidad basada en el conocimiento integral, se vuelven ideales para el equipo desarrollador dadas sus características (en el apartado 5 - Metodología se detalla más sobre las razones decisivas de Scrum).

La metodología Scrum emplea la siguiente terminología.

- Product Backlog: mejor conocido como historias de usuarios, son el conjunto de requerimientos que son descritos en un lenguaje natural.

- Sprint Planning: se refiere a una reunión en donde el Product Owner prioriza las historias de usuarios y el Scrum Team escoge o se compromete con las historias de usuario que realizarán en cada Sprint.

- Sprint Backlog: es un documento el cual contiene las historias de usuario que el Scrum Team ha seleccionado.

- Sprint: es el tiempo predefinido de duración entre 1a 4 semanas por el Scrum Team para el cumplimiento de las historias de usuario a las que se ha comprometido a realizar.

- Daily Scrum: cada día de Sprint el Scrum Team se reúne 15 minutos para rectificar el estado de las actividades del Sprint.

La metodología Scrum también incluye en su terminología denominaciones dedicadas a los roles de los agentes involucrados. Esto principalmente para establecer un nivel de entendimiento común de los roles que emplean los agentes que interactúan con el desarrollo del software tanto del lado del cliente como del equipo desarrollador. Los roles son los siguientes.

- Product Owner: tiene como oficio representar al cliente comprendiendo sus necesidades y su perspectiva del sistema, asegurando y priorizando de forma adecuada las historias de usuario.

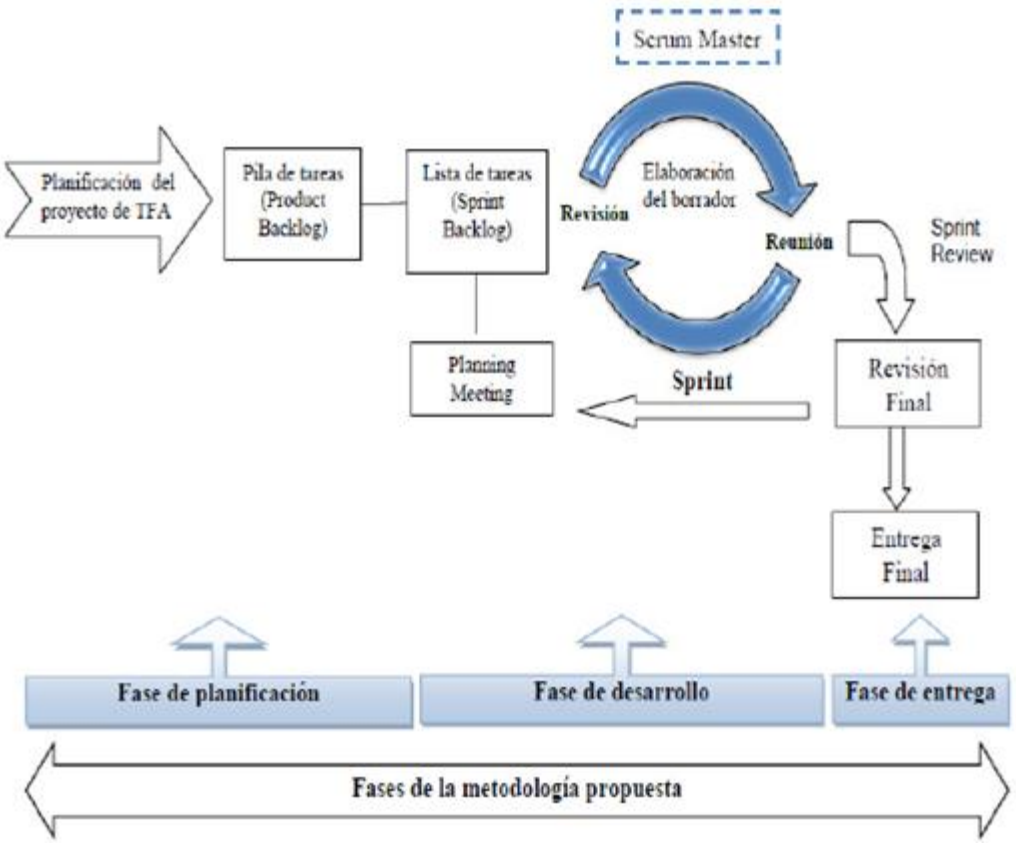
- Scrum Master: es el encargado de asegurar las buenas prácticas de la metodología Scrum y de eliminar los problemas u obstáculos que impidan el correcto desarrollo de las actividades del Sprint.

- Scrum Team: también denominado como equipo de desarrollo, es el conjunto de personas encargadas de entregar el producto de software, son grupos multidisciplinarios (análisis, diseño, desarrollo, pruebas, documentación, etc.), que por lo general están conformados por pequeños grupos de 3 a 9 personas.

4.1.4.1 Fases para el desarrollo del software con Scrum

El ciclo de vida del desarrollo del software guiado por la metodología Scrum junto con las fases que lo componen se presentan en la figura 3.

Figura 3. Fases del desarrollo del software de la metodología de desarrollo Scrum.



Fuente [4].

El ciclo de vida del desarrollo del software definido por la metodología Scrum se constituye de fases las cuales comprenden actividades que al ser desarrolladas proveen artefactos a otras fases para el continuo desarrollo de las mismas.

De la fase 1 - planeación se obtienen artefactos como el cronograma del proyecto, gestión de riesgos e historias de usuario, donde la actividad que provee las historias de usuario es aquella con mayor importancia; con un buen levantamiento de requerimientos se disminuye la adición o corrección de requerimientos iniciales.

---

[4] SONIA I., Mariño; PEDRO L., Alfonzo. "Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación". Departamento de Informática. Fac. de Ciencias Naturales y Agrimensura. Universidad Nacional del Nordeste. 19 de enero de 2014. Internet: <https://dialnet.unirioja.es/descarga/articulo/4974565.pdf> [consultado el 29 de agosto del 2016]

La fase 2 - desarrollo se realiza de manera cíclica y por tanto su duración se extiende hasta que todas las historias de usuario y reprocesos se hayan culminado. Este proceso comienza con la elección de las historias de usuario por parte del Scrum Team, el ciclo sigue con la realización de dichas historias, luego sigue la revisión del cumplimiento de las mismas para luego finalizar con su entrega.

La última fase, fase 3 - entrega, es la revisión final del sistema de información, consiste en desarrollar y ejecutar un plan de pruebas para verificar que el sistema esté funcionando correctamente, por último, el producto avalado con un manual de usuario es entregado al cliente para que el haga uso de dicho sistema.

4.2 MARCO CONTEXTUAL

El proceso que sigue todo proyecto de investigación (ya sea de investigación básica o aplicada) llevado a cabo en la UCC sede Cali sigue un flujo común desde sus inicios como anteproyecto hasta su desarrollo después de su aprobación. El inicio de un proyecto de investigación tiene lugar con la presentación de la formulación del mismo ante las facultades, dependencias y/o departamento de investigación, después se pasa por una serie de filtros siendo el primero la evaluación del consejo de facultad correspondiente (facultad más relacionada con la temática del proyecto), después se pasa por la evaluación de agente interno y externo, en función de los resultados de dichas evaluaciones el proyecto se presenta a la entidad DINAI (Dirección Nacional de Investigación de la UCC), si el proyecto es finalmente aprobado por dicha entidad el proyecto de investigación comienza su etapa de desarrollo contando con la financiación correspondiente. A partir de su comienzo el investigador principal (representante del equipo de investigación) empieza a proveer una serie de informes de avance y evidencias del proyecto de investigación ante la coordinación del departamento de investigación de la UCC sede Cali. La información que está relacionada con los proyectos desde su formulación hasta la validación de los informes de avances y evidencias del mismo se encuentra actualmente dispersa. El departamento de investigación ha decidido retomar el plan de modernizar los procesos de presentación, seguimiento y consulta de información de proyectos para contar con centralización de la información como principal propósito, con fácil acceso y alta disponibilidad.

Con el comienzo del desarrollo del proyecto de investigación se marca el comienzo del dominio del sistema. El sistema de información a desarrollar abarca los procesos asociados con la gestión de proyectos del departamento de investigación de la UCC sede Cali; los requerimientos son levantados a partir de la situación actual de dicho departamento, sin embargo, las funcionalidades del sistema permite incluir proyectos de investigación de todas las sedes de la UCC junto con sus respectivos

grupos de investigación parametrizados por la jefatura del departamento de investigación de la sede Cali (o quien tenga acceso al sistema con los privilegios requeridos), lo que permite expandir su aplicabilidad a las demás sedes de la universidad si los procesos de gestión de sus proyectos de investigación locales son similares a los procesos de la sede Cali.

5 METODOLOGÍA

5.1 METODOLOGÍA DE DESARROLLO IMPLEMENTADA

Se sigue la metodología de desarrollo Scrum ya que permite abordar las varianzas que se puedan presentar de manera ágil, lo que significa que, aunque los requerimientos cambien no generará inconvenientes para el equipo desarrollador, ya que la metodología usada tiende por sus mismas características a acomodarse a las variantes demandas del cliente y a responder con mayor certeza sus requerimientos, brindando un mayor grado de satisfacción. La manera para lograr la adaptación a los cambios en los requerimientos consiste en llevar a cabo varias iteraciones de reuniones con el cliente.

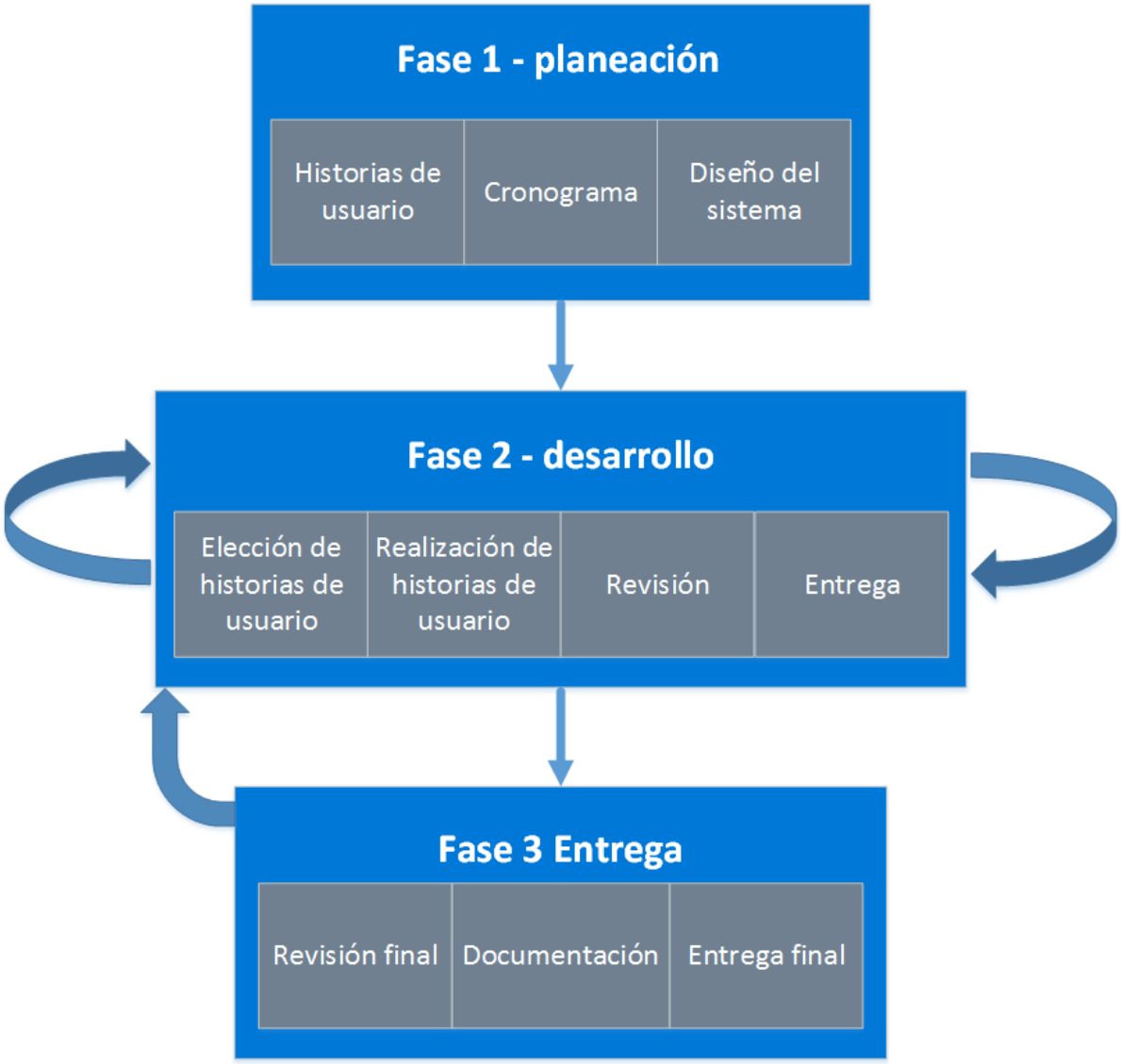
Aunque la situación relacionada con la gestión actual de la información de los proyectos de investigación posiblemente no cambie durante el desarrollo del software, el grupo desarrollador estará protegido ante un eventual cambio en los requerimientos levantados en los inicios por el uso de la metodología Scrum. Además, se aprovechará las iteraciones de reuniones con el cliente para obtener retroalimentación del avance del desarrollo de los requerimientos; la continua comunicación con el departamento de investigación permitirá al equipo desarrollador invertir todos sus esfuerzos en desarrollar requerimientos que realmente parten de la petición del cliente.

Un argumento decisivo más para la elección de la metodología Scrum está en el tamaño y distribución del grupo desarrollador. Las metodologías ágiles de desarrollo son recomendadas para grupos no mayores a 10 personas y con un espacio de trabajo común, por otro lado, las metodologías ordinarias son recomendadas para equipos grandes y distribuidos.

La presentación y retroalimentación de cada entregable de un Sprint se llevará a cabo con la jefatura del departamento de investigación y coordinación de proyectos de investigación de la UCC sede Cali. Se espera que se añadan más historias de usuario tras la entrega de las primeras, donde se incluyan requerimientos para cubrir funcionalidades de gestión y seguimiento de proyectos de grado de estudiantes de la UCC.

Las fases de la metodología Scrum originales presentadas en el marco referencial son adaptadas al uso del equipo desarrollador, de lo que se tiene las mismas tres fases, pero con artefactos por fase distintos y con misma esencia. La figura 4 presenta las fases y los artefactos de la metodología Scrum adaptada al uso del equipo desarrollador.

Figura 4. Fases del desarrollo del software Scrum adaptadas al equipo desarrollador.



BRANDON, José; DEVIA, Walter; GIRALDO, William. SGPI.

Con el levantamiento de requerimientos se tiene las historias de usuarios y a su vez, una estimación de tiempo del desarrollo del sistema. El diseño del sistema se lleva a cabo con las historias de usuario, los diagramas que se realizan son: diagrama entidad relación de base de datos, diagrama de secuencia solo para unidades de lógica de negocio claves y una combinación entre diagrama de paquetes y de topografía lógica de red para presentar a un alto nivel de abstracción la unidad de procesamiento del código fuente y los clientes en el entorno de producción.

La fase 2 se repite por cada Sprint realizado. Tras la finalización del último Sprint se sigue con la fase 3, donde su revisión final se lleva a cabo con la jefatura del departamento de investigación y la coordinación de proyectos de investigación sede Cali presentando el sistema de información culminado, aunque una fase previa de pruebas es realizada por el equipo desarrollador. Si se cuenta con aprobación en la revisión final se sigue con la elaboración de documentación que sea útil como guía de usuario, guía de desarrollador (dedicada para futuros desarrollos por parte de equipos distintos) y guía de soporte técnico. La entrega final comprende la instalación de la primera versión final del sistema de información en un entorno de producción para la UCC sede Cali, en este punto el sistema contará con todas las pruebas de calidad y funcionalidad por parte de todos los agentes interesados.

5.2 MATERIALES USADOS

Siguiendo el Modelo de Variabilidad Arquitectónica y Tecnológica para una arquitectura multicapa de entorno web y después de realizar las fases de configuración necesarias para identificar las herramientas tecnológicas a usar definidas por el mismo modelo, se determina que el framework MVC a usar será Laravel 4.2 del lenguaje de programación PHP, junto con los frameworks de diseño de interfaces Jquery UI, Angular UI y Bootstrap para diseño responsivo. Se agrega el framework AngularJS para facilitar la construcción de lógica de programación del lado del cliente usando el lenguaje de programación JavaScript. El criterio para seleccionar los lenguajes citados junto con la variedad de frameworks es el nivel de conocimiento y manejo que tienen los desarrolladores del proyecto sobre dichas tecnologías.

Para el cumplimiento de la persistencia de los datos se hará uso del sistema gestor de base de datos MySQL. Se elige este SGBD por ser gratuito y buen acople con el lenguaje de programación PHP.

Para la ejecución del lenguaje PHP y con ello la lógica de negocio del sistema se requiere un servidor con intérprete del lenguaje y con el SGBD MySQL incluido. Se hará uso del servidor con intérprete PHP (APACHE) y con servidor de base de datos MySQL gratuito que ofrece el entorno de desarrollo integrado en la nube Cloud9 [7] en la etapa del desarrollo del sistema; a posteriori en el último artefacto de la fase 3 Scrum, entrega, la aplicación será migrada a un servidor dedicado con mayores opciones de seguridad, desempeño y almacenamiento.

5.3 CRONOGRAMA DE ACTIVIDADES

No.	Actividades	Inicio	Fin	Días
1	Contextualización y levantamiento de requerimientos	15/08/2016	5/09/2016	21
3	Diseño del sistema	6/09/2016	19/09/2016	13
4	Desarrollo de las historias de usuario	20/09/2016	02/02/2017	135
5	Realización de pruebas exhaustivas del equipo desarrollador	03/02/2017	15/02/2017	12
6	Revisiones y correcciones finales	15/02/2017	20/02/2017	5

6 BIBLIOGRAFÍA

ALONSO GRACÍA, José; BERROCAL, Javier; MURILLO, Juan Manuel. "Modelado de la variabilidad en arquitecturas multicapa". Escuela Politécnica, Universidad de Extremadura. Internet: <https://www.researchgate.net/profile/Javier_Berrocal2/publication/235408808_Modelado_de_la_Variabilidad_en_arquitecturas_Multicapas/links/0fcfd5118aa485e8ac000000.pdf> [consultado el 27 de agosto del 2016]

FERNANDEZ ROMERO, Yenisleidy, DÍAS GONZALES, Yannete. “Patrón Modelo-Vista-Controlador”. Revista Telem@tica. Vol. 11. No. 1, enero – abril del 2012, p. 47 - 57.

FLOREZ FERNANDEZ, Héctor Hugo. “Arquitectura multicapa mediante AJAX y ORM”. Revista vínculos Vol. 7 No. 1, abril 30 de 2010, p. 4. Internet: <http://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4154/5818> [consultado el 27 de agosto del 2016]

SONIA I., Mariño; PEDRO L., Alfonzo. "Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación". Departamento de Informática. Fac. de Ciencias Naturales y Agrimensura. Universidad Nacional del Nordeste. 19 de enero de 2014. Internet: <https://dialnet.unirioja.es/descarga/articulo/4974565.pdf> [consultado el 29 de agosto del 2016]

PRESSMAN, Roger S. "Ingeniería del Software. Un enfoque práctico". McGraw-Hill, 6 edición. 2006.