**UNIVERSIDAD CENTRAL DEL ECUADOR**

**FACULTAD DE INGENIERIA CIENCIAS FÍSICAS Y MATEMÁTICAS**

**CARRERA DE INGENIERÍA INFORMÁTICA**

**PLAN DE PROYECTO INTEGRADOR**

**TEMA: “Automatización de solicitudes, informes y documentos técnicos para requerimientos de programas de doctorados y maestrías para la Unidad de Gestión de Doctorados de la Universidad Central del Ecuador”**

TRABAJO DE GRADUACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO INFORMÁTICA

AUTOR:

HANMILTON JHOEL BERREZUETA CARMONA

TUTOR:

ING. SANTIAGO MORALES

QUITO, 2017

DEDICATORIA

AGRADECIMIENTOS

**AUTORIZACIÓN DE LA AUTORIA INTELECTUAL**

Yo, HANMILTON JHOEL BERREZUETA CARMONA en calidad de autor del trabajo de titulación, modalidad proyecto integrador:

AUTOMATIZACIÓN DE SOLICITUDES, INFORMES Y DOCUMENTOS TÉCNICOS PARA REQUERIMIENTOS DE PROGRAMAS DE DOCTORADOS Y MAESTRÍAS PARA LA UNIDAD DE GESTIÓN DE DOCTORADOS DE LA UNIVERSIDAD CENTRAL DEL ECUADOR, autorizo a la Universidad Central del Ecuador hacer uso de todos los contenidos que me pertenecen o parte de los que contiene esta obra, con fines estrictamente académicos o de investigación.

Los derechos que como autor que me corresponden, con excepción de la presente autorización, seguirán vigentes a mi favor, de conformidad con lo establecido en los artículos 5, 6, 8, 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento.

Asimismo, autorizo a la Universidad Central del Ecuador para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

En la ciudad de Quito, a los ……………………………..

Hanmilton Jhoel Berrezueta Carmona

CC.171804823-2

Teléfono:2648123

Celular: 0992683848

hanmilton\_12@hotmail.com

**CERTIFICACIÓN DEL TUTOR**

Yo, Santiago Morales en calidad de tutor del trabajo de titulación: **Automatización de solicitudes, informes y documentos técnicos para requerimientos de programas de doctorados y maestrías para la Unidad de Gestión de Doctorados de la Universidad Central del Ecuador**, elaborado porel estudiante Hanmilton Jhoel Berrezueta Carmona de la Carrera de Informática, Facultad de Ingeniería Ciencias Físicas y Matemática de la Universidad Central del Ecuador, considero que el mismo reúne los requisitos y méritos necesarios en el campo metodológico y en el campo epistemológico, para ser sometidos a la evaluación por parte del jurado examinador que se designe, por lo que lo APRUEBO, a fin de que el proyecto integrador sea habilitado para continuar con el proceso de titulación determinado por la Universidad Central del Ecuador.

En la ciudad de Quito, a los ……………………………….

Ing.Santiago Morales

CC: …………….

Certificados

certificados

certificados

certificados

**CONTENIDO**

**CONTENIDO**

**LISTA DE ANEXOS**

**LISTA DE FIGURAS**

**LISTA DE FIGURAS**

**LISTA DE TABLAS**

**RESUMEN**

**TEMA:** “Automatización de solicitudes, informes y documentos técnicos para requerimientos de programas de doctorados y maestrías para la Unidad de Gestión de Doctorados de la Universidad Central del Ecuador”

**Autor:** Hanmiton Jhoel Berrezueta Carmona

**Tutor:** Santiago Morales.

Con el objetivo agilizar el proceso de gestión de solicitudes, informes y documentos para la incorporación de docentes a programas de doctorados y maestrías Unidad de Gestión de Doctorados de la Universidad Central del Ecuador.

Para la automatizacin se desarrollou una aplicación web que se encontrará alojada en un servidor web, por lo cual se realizo un análisis de los procesos que tiene la Unidad de Gestion, acontinuacion se evaluaron distintas metodologías de desarrollo y se selecciono metodología XP y el stack de tecnologías que se usó es MEAN este acrónimo viene de Mongo + Express + Angular + Node con esto se usara el lenguaje javascript en todos los componentes de la aplicación web.

Con el desarrollo de esta aplicación se permitio mejorar el importante servicio que brinda la Unidad de Gestión de Doctorados para los docentes.

**PALABRAS** **CLAVE:** / SERVIDOR WEB/ MEAN / MONGO / EXPRESS / ANGULAR / NODE

**ABSTRACT**

**TOPIC:**

**INTRODUCCIÓN**

La Unidad de Gestión de Doctorados de la Universidad Central del Ecuador no cuenta con un software para agilizar el proceso de gestión de solicitudes, informes y documentos para la incorporación de docentes a programas de doctorados y maestrías los mismos que deben ser puestos a consideración del Rector para su aprobación y disposición a las unidades respectivas.

El desarrollo de esta aplicación permitirá mejorar el importante servicio que brinda la Unidad de Gestión de Doctorados para los docentes, ya que la automatización agilitará el tiempo de gestión de los procesos de la Unidad.

Para la automatizacin se desarrollou una aplicación web que se encontrará alojada en un servidor web, por lo cual se realizo un análisis de los procesos que tiene la Unidad de Gestion, acontinuacion se evaluaron distintas metodologías de desarrollo y se selecciono metodología XP y el stack de tecnologías que se usó es MEAN este acrónimo viene de Mongo + Express + Angular + Node con esto se usara el lenguaje javascript en todos los componentes de la aplicación web.

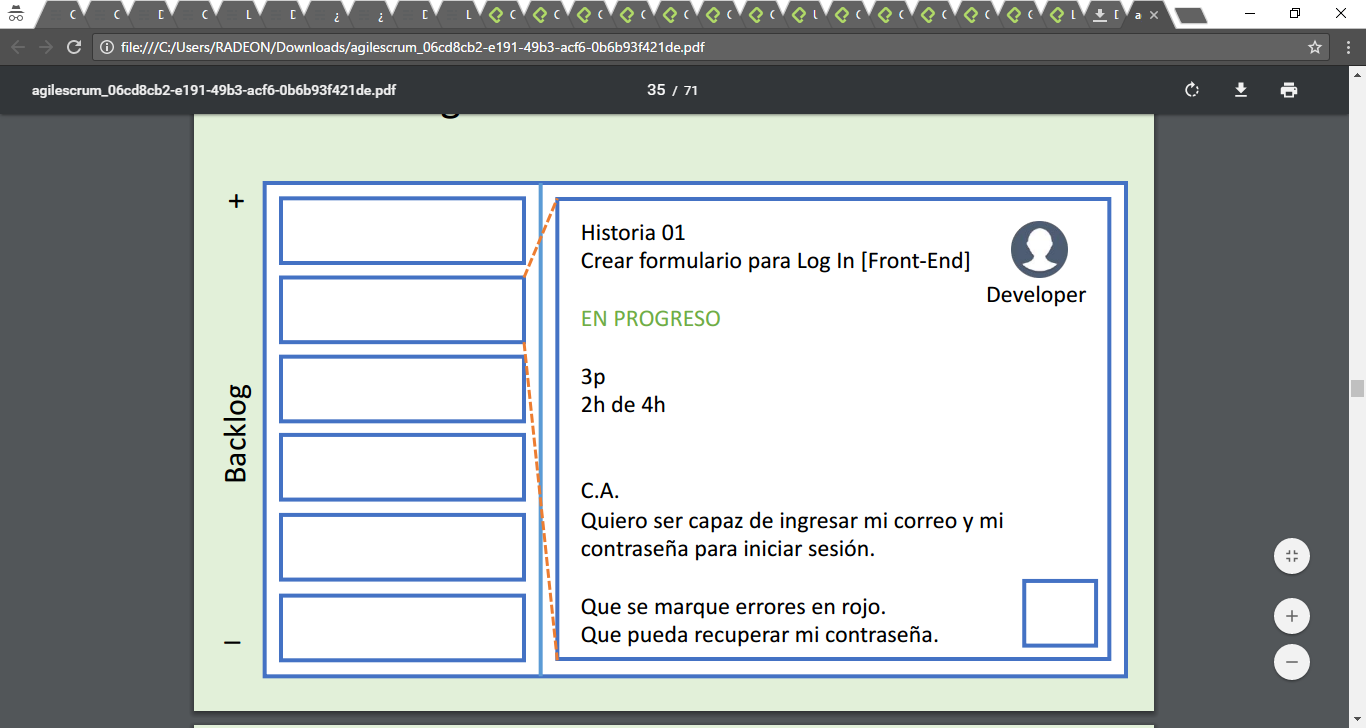
Como paso final se darán las conclusiones y recomendaciones pertinentes obtenidas en el desarrollo de la aplicación web para mejorar el servicio que brinda la Unidad de Gestión de Doctorados para los docentes.

1. **PRESENTACIÓN DEL PROBLEMA**
   1. **ANTECEDENTES**
   2. **PLANTEAMIENTO DEL PROBLEMA**
   3. **OBJETIVOS**
      1. **OBJETIVO GENERAL**
      2. **OBJETIVOS ESPECIFICOS**
      3. **JUSTIFICACIÓN**
      4. **ALCANCE**
2. **MARCO TEÓRICO**
   1. **CONTROL DE VERSIONES**
      1. **GIT**
         1. **DEPLOY CON GIT**
         2. **GIT FLOW**
      2. **GITHUB**
      3. **COLABORACIÓN EN EQUIPO**
   2. **DESARROLLO WEB**
      1. **HTML5**
      2. **CSS3**
      3. **FLEXBOX**
      4. **TIPOGRAFÍA WEB**
      5. **RESPONSIVE DESIGN**
         1. **PATRONES DISEÑO ADAPTATIVO**
         2. **OPTIMIZACIÓN CARGA SITIO WEB**
3. **METODOLOGÍA DE DESARROLLO**
   1. **COMUNICACIÓN**
      1. **TIPOS DE COMUNICACIÓN**

• Verbal, Escrita.

• Metodología.

• Backlog



• Historias de usuario

• Wireframes

Representación gráfica conceptual del producto



• Personas

Usuarios finales

• Iteraciones

Planeación

• Votaciones

• Estimación de puntos

• Stand Up

• ¿Qué hice?

• ¿Qué haré hoy? •

Si tengo problemas, ¿cuáles?

• Retrospectiva

• ¿Qué funcionó?

• ¿Qué no funcionó?

• ¿Qué podemos mejorar?

* 1. **DEFINICIÓN DE AGILE**

Es una manera de administrar proyectos de software que tiene las siguientes características:

* Se encarga de trabajar en pequeñas entregas funcionales.
* Minimiza así los problemas y defectos que puedan ocurrir.
* Organiza el trabajo bajo iteraciones o ciclos de trabajo (sprints en el caso de la metodología llamada Scrum).
* Domina la comunicación permanente tanto entre los miembros del equipo entre sí como con los miembros del negocio nuestros clientes.
* Permite entender los requerimientos funcionales o no funcionales de lo que queramos lograr.
  1. **MANIFIESTO DESARROLLO ÁGIL DE SOFTWARE**

[El manifiesto ágil](http://agilemanifesto.org/iso/es/manifesto.html) es un documento con más de 15 años (creado en el 2001) que reúne **principios y prácticas/procesos** para crear productos de software de una mejor manera.

* **Individuos e interacciones** sobre procesos y herramientas: confiar en las habilidades de cada miembro del equipo, valorar sus conocimientos y experiencia que pueda aportar, establecer una comunicación eficiente entre los miembros del equipo
* **Software funcionando** sobre documentación extensiva: pequeñas porciones de funcionalidad junto al código exclusivamente necesario para correr esas funcionalidades
* **Colaboración con el cliente** sobre negociación de contratos: una comunicación permanente y transparente con el cliente, en la cual el cliente transmitirá lo que espera del proyecto y el equipo siempre responderá con lo que se ha completado hasta el momento.
* **Respuesta ante el cambio** sobre seguir un plan: mientras más pasa el tiempo, lo que se entrega al cliente cada vez es más grande, cambiando sólo lo necesario.
  + 1. **LOS 12 PRINCIPIOS DEL MANIFIESTO PARA EL DESARROLLO DE SOFTWARE**
* Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
* Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
* Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
* Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
* Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
* El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
* El software funcionando es la medida principal de progreso.
* Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
* La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
* La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
* Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
* A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.
  1. **SCRUM**
  2. **KANBAN**
  3. **XP**

1. **HERRAMIENTAS DE PROGRAMACIÓN** 
   1. **JAVASCRIPT**
      1. **SINGLE PAGE APPLICATION**
      2. **GULP**
      3. **YO-YO**
      4. **MOMENTJS**
      5. **SUPERAGENT**
      6. **MULTER**
   2. **STACK MEAN**
      1. **MONGO**
      2. **EXPRESS**
      3. **ANGULAR**
      4. **NODE**
         1. **NODE Y NPM**
   3. **FRONTEND**
      1. **BOOTSTRAP**
      2. **FOUNDATION**
      3. **SASS**
   4. **ZENHUB**

**Projecto managemente mas completo y de manera nativa desde github y proyecot tipo scrum**

**Issues**

**Todos los probelams que la app debe trabajar**

**icebox**

**todas las ideas que se pueden ir ejecutando**

**no es prioridad**

**backlog**

**es donde se va priorizando que es lo mas importante a desarrollar**

**principales features y bugs**

**In progress**

**cuando ya se empieza a trabajar en el feature o en el bug**

**Review/QA**

**se revisa para hacer una funcion a master rama**

**Done**

**cuando es aprobado y fusionado con la rama master**

**Close**

**cerrado y solucionado**

**nota**

**feature: Nueva función o mejora**

**bugfix: bug de baja prioridad**

**hotfix: bug de alta prioridad**

**Millestones**

**burndown**

**como se va avanzando y si se va cumplientdo con los deadlines a nivel de proyecto**

Milestones ->Epics -> Issues(ToDo's) -> Tasks

1. **GLOSARIO**

* **Agile: (Ágil)** Nombre genérico de las metodologías de desarrollo modernas para administrar proyectos de software, aunque no exclusivo.
* **SCRUM**: Framework iterativo e incremental de Agile para la administración de desarrollo de productos.
* **KANBAN**: Framework de flujo de trabajo para administración de desarrollo de productos.
* **XP**: (eXtreme Programming) Framework de trabajo para la administración de desarrollo de productos caracterizado por su énfasis en procesos definidos y cortos.
* **Manifesto**: (Manifiesto) Una declaración pública de políticas y enfoques referente a algo en particular.
* **Product Owner**: (“Dueño del producto”) Persona(s) encargadas de administrar requerimientos y evaluar resultados de un equipo de desarrollo ágil.
* **Final Users (Usuarios Finales)**: Persona(s) que tienen un contacto directo con el producto en última instancia.
* **Domain Users: (Usuarios de Dominio)** Persona(s) que agrupan usuarios finales en diferentes tipos o funciones.
* **Solutions Architect: (Arquitectos de soluciones)** Rol técnico referente a estructura e implementación de tecnologías y procesos en un desarrollo de producto.
* **Iteraciones**: Repeticiones de procesos con la finalidad de alcanzar una meta deseada.
* **Sprint**: Nombre con el que se le conoce a las iteraciones en SCRUM.
* **Retrospectiva**: Revisión hacia atrás desde un punto de vista temporal, de las acciones y procesos desarrollado en un equipo para una mejora continua.
* **Stand-Up: (De pie, pararse)** Es uno de los rituales de Agile que consiste en una reunión para sincronizarse entre miembros de los equipos de manera diaria y periódica.
* **Spike**: Tipo de Sprint para realizar experimentos, investigación o documentación avanzada. Dentro de los proyectos no suelen ser más del 20% del total de iteraciones y pueden o no agregar valor al producto.
* **Stakeholder**: Todas aquellas terceras personas que tienen un interés (monetario o no) en una compañía u organización y pueden afectar o ser afectados por el negocio.
* **Backlog**: Contenedor lógico de las tareas que aún no se consideran en un proceso o estado activo para el equipo. Pueden o no estar priorizadas.
* **Sprint backlog**: Son las tareas ya priorizadas y apiladas que forman parte del sprint actual y que no han sido comenzadas a trabajar
* **User Stories**: (Historias de usuario) Es la unidad de trabajo de Agile. Es la representación de la interacción de un usuario con el producto y tiene como mínimo: Nombre e identificador, descripción (“Siendo usuario x ¿Qué espero hacer?”), prioridad, criterio de aceptación (“Si lo puedo hacer, ¿Cuáles son las características mínimas para lograrlo?”), persona asignada y estimación.
* **Wireframes**: Como en UX, es una representación visual del producto, pero en relación a las interacciones de los usuarios y roles.
* **MVP: (Minimum Valuable Product)** Mínimo Producto Valuable. Es la funcionalidad básica con la cual el Product Owner puede justificar inversiones y puede medir el ROI.
* **Epic**: Modo de granular requerimientos que se basa en las funcionalidades básicas del producto o MVP y funcionalidades extras que requieran varias iteraciones y que puedan dar un valor agregado al producto.
* **ROI**: (Return Of Investment) Retorno De Inversión. Es una métrica para evaluar el costo/beneficio del equipo en función del producto a lo largo del tiempo.
* **Bugs**: (Defectos) Fallas de funcionalidad que pueden presentarse en cualquier capa o fase del producto y son causados por mal uso o desarrollo ineficiente o que no sigue los estándares de código.
* **BurnDown** Chart: Gráfica para medir a lo largo de periodos de tiempo, el trabajo restante o que aún no se considera.
* **BurnUp Chart**: Gráfica para medir de forma incremental la funcionalidad y requerimientos entregados a lo largo de periodos de tiempo.
* **Feature**: Funcionalidad que se basa en roles o elementos del negocio para la realización de tareas específicas con fines de generar ingresos.