



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO**

---

---

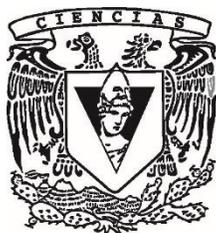
**FACULTAD DE CIENCIAS**

**CREACIÓN Y DOCUMENTACIÓN DE UN SISTEMA  
WEB PARA LA SECRETARÍA DE EDUCACIÓN  
ABIERTA Y CONTINUA**

**ACTIVIDAD DE APOYO A LA DOCENCIA**

**QUE PARA OBTENER EL TÍTULO DE:  
LICENCIADO EN CIENCIAS DE LA  
COMPUTACIÓN  
P R E S E N T A:**

**FERNANDO ANTONIO SÁNCHEZ MONTOYA**



**DIRECTOR DE TESIS:**

**M. EN C. MARÍA GUADALUPE ELENA  
IBARGÜENGOITIA GONZÁLEZ**

**CIUDAD UNIVERSITARIA, CD. MX., 2018**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno.  
Sánchez  
Montoya  
Fernando Antonio  
5561606770  
Universidad Nacional Autónoma de México  
Facultad de Ciencias  
Ciencias de la Computación  
309160568
2. Datos del asesor.  
M en C.  
Ibargüengoitia  
González  
María Guadalupe Elena
3. Datos del sinodal 1  
Dra.  
Oktaba  
Jadwiga  
Hanna
4. Datos del sinodal 2  
Dr.  
Valdés  
Souto  
Francisco
5. Datos del sinodal 3  
Dra.  
Martínez  
Ramírez  
Selene Marisol
6. Datos del sinodal 4  
M. en C.  
Márquez  
Flores  
Gustavo Arturo
7. Datos de la Tesis  
Creación y documentación de un sistema web para la Secretaría de Educación Abierta y Continua  
Manual para Ingeniería de Software II  
153 p.  
2018

1.	Introducción	6
	Estructura del manual.	7
2.	Forma de trabajar.	8
	Objetivo del curso.	8
3.	SCRUM	10
	Introducción	10
	Principios de SCRUM	11
	Control empírico de procesos:	11
	Auto-organización:	12
	Colaboración:	12
	Priorización basada en el valor:	12
	Time-boxing:	12
	Desarrollo Iterativo:	13
	Organización en SCRUM	14
	Justificación de negocios	14
	Calidad	15
	Cambio	15
	Riesgo	15
	Procesos de SCRUM	16
	Fase de iniciación	16
	Fase de plan y estimación	16
	Fase de implementación	17
	Fase de revisión y de la retrospectiva de trabajo	17
	Fase de entrega	17
4.	Trello	18
	Introducción.	18
	Uso de Trello.	18
	Creación de una sesión en Trello.	18
	Creación y uso de un tablero en Trello.	19
	Creación y propiedades de tareas en un tablero en Trello.	22
	Ventajas del uso de un tablero de control de tareas	28
5.	Construcción del Sistema Web.	30
	Planteamiento del Problema de la Secretaría de Educación Abierta y continua de la Facultad de Ciencias.	30
	Desarrollo de la primera, segunda y sexta iteración.	31

Fase de iniciación:	31	
Product backlog inicial.		33
Fase de plan y estimación	35	
Plan de la primera iteración.		36
Plan de la segunda iteración.		37
Fase de implementación	37	
Product backlog primera iteración.		38
Product backlog segunda iteración.		38
Fase de revisión y de la retrospectiva de trabajo	40	
Product backlog sexta iteración.		41
Fase de entrega	43	
<b>6. Resultado del Sistema.</b>	<b>44</b>	
Página de inicio.	45	
1. ¿TE INTERESA ALGÚN CURSO? REGÍSTRATE		46
2. REGISTRAR PONENTE		50
Rol de Administrador.		53
Exportar a PDF información del Ponente		55
Rechazar Ponente		57
Registrar curso		59
Exportar a PDF información del Curso		65
Editar Curso		69
Rechazar Curso		70
Aprobar Curso		71
3. CATÁLOGO DE CURSOS		72
Iniciando sesión		74
<b>7. Conclusiones.</b>	<b>76</b>	
<b>8. Bibliografía</b>	<b>77</b>	
<b>9. ANEXO A. DOCUMENTOS INICIALES.</b>	<b>78</b>	
Anexo A. A1. Requerimientos generales.		79
Anexo A. A2. Plan del Proyecto.		85
Anexo A. A3. Product Backlog inicial.		93
Anexo A. A4. Arquitectura.		96
<b>10. ANEXO B. DOCUMENTOS PRIMERA ITERACIÓN.</b>	<b>102</b>	
Anexo B. B1. Requerimientos de la primera iteración.		103
Anexo B. B2. Plan de la primera iteración.		107

Anexo B. B3. Arquitectura primera iteración.	109
Anexo B. B4. Retrospectiva.	114
Anexo B. B5. Product Backlog modificado. (Primera iteración)	117
<b>11. ANEXO C. DOCUMENTOS SEGUNDA ITERACIÓN.</b>	<b>119</b>
Anexo C. C1. Requerimientos de la segunda iteración	120
Anexo C. C2. Plan de la segunda iteración.	124
Anexo C. C3. Arquitectura segunda iteración.	126
Anexo C. C4. Retrospectiva.	131
Anexo C. C5. Product Backlog modificado. (Segunda iteración)	135
<b>12. ANEXO D. DOCUMENTOS SEXTA ITERACIÓN.</b>	<b>137</b>
Anexo D. D1. Requerimientos de la sexta iteración.	138
Anexo D. D2. Plan de la sexta iteración.	141
Anexo D. D3. Arquitectura sexta iteración.	143
Anexo D. D4. Retrospectiva	148
Anexo D. D5. Product Backlog finalizado.	151

# 1.Introducción

El presente trabajo pretende mostrar al estudiante, de la materia de Ingeniería de software II, la importancia de trabajar con una metodología ágil y herramientas, que servirán de ayuda para la creación de un sistema web.

Una problemática común es la falta de documentación de un proyecto, que va a ser desarrollado, que está siendo desarrollado o que ya fue desarrollado.

La falta de elaboración de documentos basada en estándares de notación UML, como los diagramas de casos de uso, descripción detallada de casos de uso, diagrama de la base de datos, arquitectura del sistema, por mencionar algunos; puede perjudicar al proyecto en muchos aspectos. Si se requiere dar mantenimiento o extender sus funcionalidades, o tal vez sea necesario hacer una medición de tamaño funcional e incluso, si se desean hacer correcciones, es necesario contar con un mínimo de documentación.

La metodología ágil, en este caso será SCRUM, cuenta con procesos específicos, en estas notas se realiza un esbozo de éstos y se busca que el alumno los ponga en práctica para mejorar la calidad del trabajo, cumpliendo el objetivo del curso de una manera eficaz.

La metodología ágil provee al proyecto, con una gama de documentos que son importantes para describir un sistema, antes de su creación, durante el desarrollo y al final de este.

Por estas razones, este trabajo presenta un manual para la creación y documentación de un sistema web. En este caso, se ejemplifica con un sistema web creado para la Secretaría de Educación Abierta y Continua de la Facultad de Ciencias, con los documentos que deben realizarse, al hacer un proyecto en el curso de Ingeniería de Software II o al usar una metodología ágil como SCRUM.

Este trabajo, presenta ejemplos de los productos que un alumno de Ingeniería de Software II, tiene que realizar durante el curso. Se ejemplifica con el caso de un proyecto de desarrollo que se hizo en el año 2015 [Realización de la página para la Secretaría de Educación Abierta y Continua de la Facultad de Ciencias de la Universidad Nacional Autónoma de México].

Durante el curso de Ingeniería de Software II, el alumno deberá reconocer las propuestas existentes de las buenas prácticas de la ingeniería de software, con la finalidad de aplicarlas en el desarrollo de proyectos.

## Estructura del manual.

El manual, al inicio, cuenta con una introducción que relata en aspectos generales lo que es el manual y su función dentro de la materia. Después, se describe el objetivo del curso de Ingeniería de Software II.

**Capítulo 2.** Explica la forma en que se trabaja durante el curso de Ingeniería de Software II y da a conocer los aspectos más relevantes con los que se trabajará.

**Capítulo 3.** En este capítulo se describe la metodología ágil que se usará en el proyecto, SCRUM, sus prácticas y principios; también los documentos necesarios antes de iniciar con el proyecto, los documentos correspondientes de cada iteración y los documentos al finalizar el proyecto.

**Capítulo 4.** Trello, es una herramienta de organización y gestión de tareas útil para los miembros del equipo SCRUM, en el desarrollo del sistema web, por lo que se presenta un manual para su uso.

**Capítulo 5.** Presenta un caso de ejemplo, en el que se mostrará el desarrollo de las primeras dos iteraciones y la última, con el fin de mostrar la evolución de la documentación.

**Capítulo 6.** En este capítulo del manual, se muestran los resultados del funcionamiento de algunos casos de prueba, del ejemplo del sistema web creado para la SEAYC.

Las conclusiones relatan la importancia de cursar esta materia en los semestres finales de la carrera de Ciencias de la Computación, para que la incursión del alumno en la ingeniería de software, en un ambiente laboral, sea eficaz y de forma profesional.

Los anexos muestran la documentación mínima requerida para iniciar el proyecto y los documentos que se generan por cada iteración, con el fin de ilustrar, al alumno, en la creación de la propia documentación para el sistema.

## 2. Forma de trabajar.

### Objetivo del curso.

El objetivo del curso de la materia de Ingeniería de software II es: brindar una experiencia real al alumno de la utilización de la ingeniería de software. Por lo tanto, la Profesora del curso María Guadalupe Elena Ibargüengoitia González, se da a la tarea de encontrar “Clientes Reales”, para la realización de un proyecto; que aportará un beneficio a la comunidad universitaria. De tal forma, que permita, siendo estudiantes, vivir una experiencia del ámbito laboral aplicando los conocimientos adquiridos durante la Licenciatura de Ciencias de la Computación, como lo son Bases de Datos, Ingeniería de Software y Modelado y Programación, por mencionar algunos.

Los alumnos interactúan con los “Clientes Reales”, con la profesora; como intermediaria para la realización de un sistema web, de complejidad moderada, que cumpla con las necesidades propuestas por los “Clientes Reales” en un periodo de un semestre escolar (tres meses aproximadamente).

Para la realización del proyecto de la Ingeniería de Software II, el equipo, de tres miembros, se basa en la metodología ágil llamada SCRUM.

La realización del proyecto se hace en iteraciones, las cuales constan de aproximadamente dos semanas cada una y son seis iteraciones en el curso, cumpliendo los requerimientos establecidos desde un principio con los “Clientes Reales”. Además, de integrar el código creado, preparar los entregables y la documentación del sistema web.

Durante cada iteración, se hace la recomendación del uso de Trello para la asignación y control de tareas que cada integrante del equipo debe realizar, al igual que la descripción de éstas y su fecha de entrega.

Al final de cada iteración, en la fecha establecida, se entrega el código y la documentación. En este manual se ejemplificarán las iteraciones 1, 2 y 6, con los respectivos documentos; así como los documentos iniciales que SCRUM requiere.

Los documentos que se generan antes de iniciar el desarrollo y durante las iteraciones son los siguientes:

- **Plan del proyecto.** En este documento se establecen: un resumen del enunciado del problema del proyecto, objetivo, alcance, los entregables y los recursos que se usarán para el desarrollo del proyecto.  
A partir de estos textos se construye el diagrama de casos de uso, y se les asignan prioridades a las tareas y actividades. También se describen los riesgos que se pueden presentar durante el desarrollo del sistema y un plan de contingencia para minimizar estos y un plan de control de versiones sobre los entregables.
- **Product backlog.** Este documento describe los casos de uso a realizar, en una tabla a manera de itinerario, donde se establecen tres factores importantes para el inicio del desarrollo del sistema web, basado aún, en estimaciones hechas por los miembros del equipo. Primero, al caso de uso, se le asigna un número de iteración, en la cual, este se desarrollará. También se le otorga una prioridad (baja, media o alta) dependiendo su importancia con respecto a otros

casos de uso. Al final, se hace una estimación de tiempo que se considera idóneo, por los alumnos, para su realización, de acuerdo a la dificultad del caso de uso.

- **Arquitectura de la iteración.** Este documento muestra los atributos de calidad, el patrón arquitectónico que guiará el desarrollo, el ambiente de implementación, el diagrama de componentes y sus interfaces y la base de datos.
- **Requerimientos por iteración.** El contenido de este documento contiene una parte del total de los casos de uso y tareas, que se deben cumplir para la iteración correspondiente. También cuenta con un plan de pruebas mínimas al haber terminado la iteración.
- **Plan de la iteración.** Se muestra el desglose de las tareas a realizar en la iteración correspondiente y a qué miembro del equipo le fueron asignadas.
- **Retrospectiva de la iteración.** Este documento busca ser una retroalimentación de cómo se desarrolló el equipo durante la iteración correspondiente. El objetivo de este documento es que el equipo reconozca los logros obtenidos y los problemas que se presentaron durante la iteración, para que estos aspectos en la práctica, funcionen de mejor manera para la siguiente iteración.

La administración de tiempos en cuestión de tareas durante la iteración se hace a través de Trello, cada integrante puede ver las tareas que le corresponden e indicar el avance de éstas.

Durante las reuniones, obligatorias de 15 minutos al día, los miembros del equipo establecen las problemáticas a resolver y se revisa en conjunto el software para verificar la funcionalidad y detectar si hay cosas pendientes o está completo. También se realiza la retrospectiva de la iteración que sirve para identificar y resaltar aspectos que deben implementarse durante la iteración y mejorar estos en la siguiente.

Al término de cada iteración se agenda una cita con los “Clientes Reales”, para mostrar los avances del sistema web hasta ese momento. Ellos por su parte, hacen observaciones en cuanto a detalles estéticos, técnicos o de funcionalidad del software y si desean agregar al sistema algunos puntos que consideran importantes, estos aspectos son considerados hasta la iteración final.

Dentro de este manual, también se encuentra una sección de Anexos para ejemplificar los productos y que el alumno realice sin problemas su proyecto.

Con esta forma de trabajo estipulada, el proyecto se desarrollará lo largo del semestre. La forma de evaluación del curso de Ingeniería de Software II, está constituida con la entrega del desarrollo web con los debidos documentos, en tiempo y forma establecidos.

Hay otro trabajo de titulación que es el manual que contiene la parte técnica (complementario al presente) se mostrará a los alumnos a generar el código del proyecto en Ruby on Rails.

# 3.SCRUM

## Introducción

SCRUM, es un marco de trabajo ágil muy popular en el mundo de la ingeniería de software ya que es rápido, flexible y eficaz. Su implementación, a lo largo de un proyecto de software, es a través de iteraciones. SCRUM está basado en una comunicación confiable y trabajo responsable en equipo para un fin común: El desarrollo de proyectos en ingeniería de software.

El equipo, lo más importante en la metodología de SCRUM, debe ser multifuncional, es decir, los integrantes del equipo deben realizar tareas coordinadas en periodos de trabajo cortos, lo que permite que el flujo de trabajo sea constante y alcanzable. Si se realiza una planeación incorrecta y un reparto de tareas inequitativo en una iteración, el equipo se verá obligado a tomar más tiempo del establecido y la calidad del proyecto puede verse afectada.

La primera fase de la metodología comienza con una junta entre las partes interesadas (cliente – desarrolladores) para establecer la visión del proyecto que se tiene que realizar. El cliente presenta la problemática que suscita la necesidad de crear un proyecto de software y presenta los puntos con mayor prioridad a realizar de acuerdo a las necesidades del sistema, de esta situación se genera el documento llamado Product Backlog.

El equipo, a partir del Product Backlog, empieza a dividir las tareas con mayor prioridad en iteraciones o sprints y se reparten las tareas equitativamente entre los miembros del equipo.

Para llevar el control de las tareas asignadas a cada miembro del equipo se usa un tablero, donde se especifica, la fecha límite de su entrega. Esto es útil para el equipo ya que se lleva un control para evitar retrasos al presentar avances al cliente. La descripción del uso de un tablero de tareas se detalla en el capítulo Trello de este mismo manual.

Cuando se han cumplido todas las tareas asignadas en la iteración, es preciso convocar una junta con los clientes para presentar los avances y realizar una debida retroalimentación.

Si el cliente pide correcciones o agregar nuevas tareas, al final o a la mitad de una iteración, éstas se consideran al final del Product Backlog para tener un control específico del tiempo y de las tareas con prioridades.

Una regla importante de SCRUM, es no complacer al cliente cuando él lo pida.

En este capítulo se detallarán las prácticas de SCRUM, para que el alumno pueda ponerlas en práctica en el desarrollo de su proyecto de Ingeniería de Software II.

# Principios de SCRUM

SCRUM cuenta con seis principios:

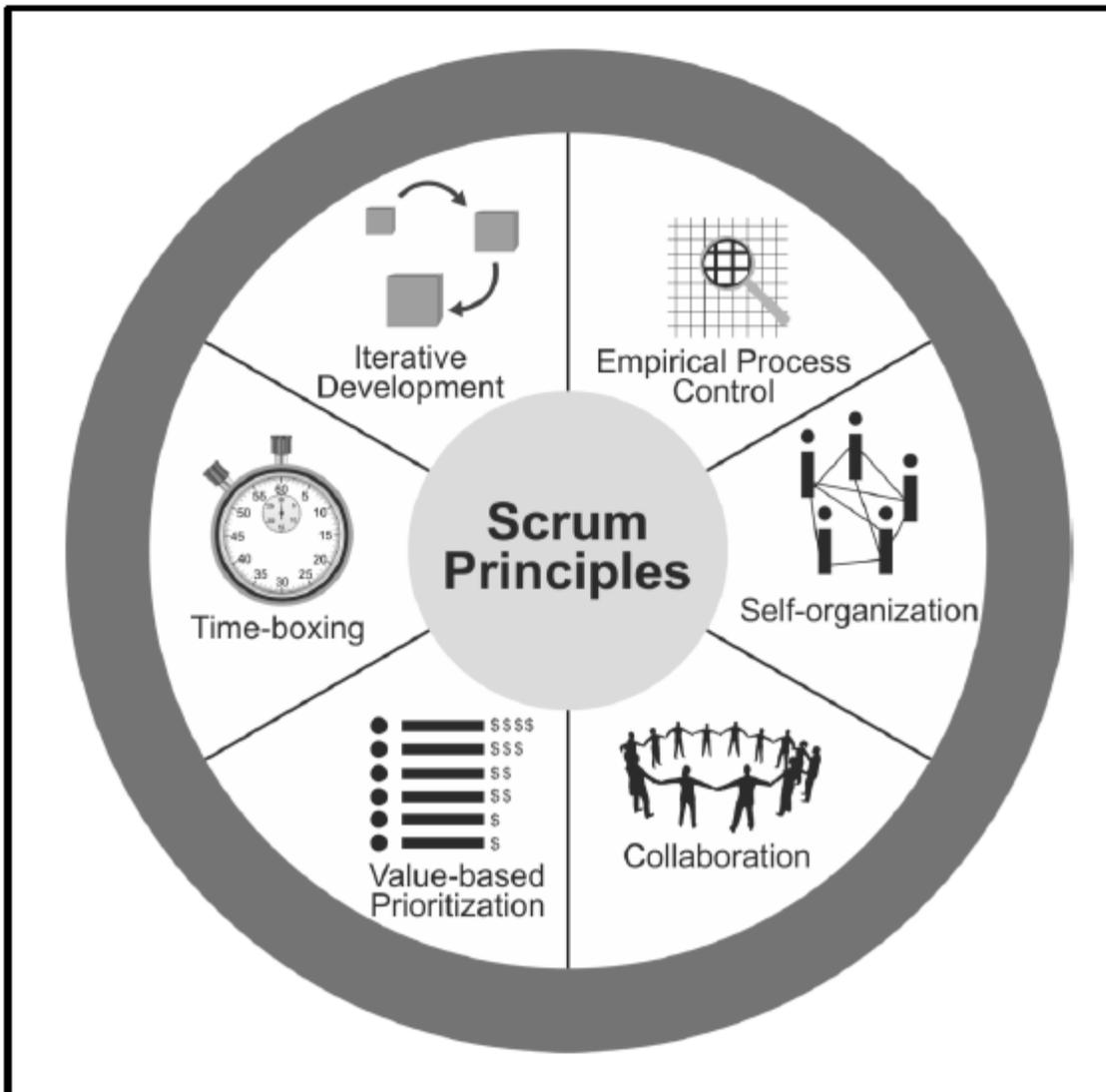


Figura 1. Principios de SCRUM. [A Guide to the Scrum Body of Knowledge (SBOK™ Guide)]

## Control empírico de procesos:

- **Transparencia:** La transparencia en SCRUM es un factor de importancia ya que permite a cualquiera observar de manera parcial durante el proceso de trabajo. Es decir, cualquier miembro del equipo, supervisor de trabajo, miembros de la parte interesada, tienen que tener una visión sin límites sobre el trabajo (Product Backlog, tableros organizacionales de tareas, documentación, información de las juntas, etc...).
- **Inspección:** Como consecuencia de la transparencia, la inspección, sobre todo cuando hay límite de entrega, se debe llevar un monitoreo sobre el progreso del equipo y sobre los avances generados hasta el momento y verificar que todo vaya de acuerdo al documento Product Backlog y a las fechas establecidas.

- **Adaptación:** Es la manera en que se da el acoplamiento a las reuniones diarias de 15 minutos, a la inspección de tareas, así como las juntas con los clientes al final de cada iteración, al igual que la retrospectiva y evaluación de riesgos.

### **Auto-organización:**

Este principio de SCRUM se basa en la creencia de que los miembros se conciben como un equipo por lo tanto hay una motivación grupal para conseguir el éxito como un todo, pero esto no sería así si los miembros no fueran responsables y organizados.

### **Colaboración:**

Principio en el cual se debe hacer énfasis, la colaboración del equipo SCRUM es multidireccional: equipo SCRUM- equipo SCRUM, equipo SCRUM-Parte Interesada, esta interacción es esencial para la creación y validación del producto en todas sus fases para alcanzar la meta exitosamente. La colaboración nace cuando los miembros del equipo aportan ideas y trabajan en conjunto, con ayuda de la parte interesada y los supervisores para transformar el trabajo generado y llegar al objetivo planeado.

Consta de tres dimensiones fundamentales que son:

- **Concientización:** en el trabajo en equipo se debe tener conciencia del trabajo del otro.
- **Articulación:** de la generación de trabajo dividido entre los miembros del equipo y al finalizar las tareas individuales, integrarlas en un todo.
- **Apropiación:** adaptación de las tecnologías por parte del equipo para homogenizar el trabajo lo máximo posible.

### **Priorización basada en el valor:**

En este principio el concepto de la priorización determina la división de las tareas, a realizar por el equipo SCRUM, de las más importantes o urgentes a las que no son urgentes o dependen de una más importante. Además de asignar fechas a las tareas. La priorización de las tareas debe correr a cargo de la parte interesada en conjunto con el equipo SCRUM.

### **Time-boxing:**

Una de las cosas más importantes, en la gestión de un proyecto, es el tiempo, pues en consecuencia todo está aunado a él. En SCRUM el nombre de Time-boxing se le da al manejo adecuado del tiempo al establecer periodos justos (ni más, ni menos tiempo) para la realización de cualquier tipo de tareas asignadas a los miembros del equipo SCRUM. Time-boxing en SCRUM es el máximo aprovechamiento del tiempo en beneficio del proyecto, y se ve reflejado en actividades o prácticas como lo son las juntas diarias de 15 minutos entre los integrantes del equipo, las iteraciones definidas las juntas donde se hace la retrospectiva del trabajo al final de alguna iteración, etc.

## **Desarrollo Iterativo:**

Como se mencionó en la parte inicial de este capítulo una práctica de SCRUM en la parte del desarrollo del proyecto se divide en iteraciones que constan de una secuencia de tareas a desarrollar en un periodo de tiempo previamente definido.

## Organización en SCRUM

SCRUM cuenta con una sólida organización, tiene implementados roles bien definidos lo cual permite llegar a los resultados deseados de una manera eficiente.

Algunos de los roles que dicta SCRUM son los siguientes:

Roles principales: Son roles esenciales que SCRUM necesita para concretar un proyecto exitosamente.

- SCRUM master: Líder y responsable de facilitar al equipo SCRUM el ambiente propicio para concretar un proyecto exitosamente.
- Dueño del producto: Persona responsable de especificar los requerimientos del cliente y comunicarlos al equipo SCRUM.
- Equipo SCRUM: Grupo responsable de desarrollar los requerimientos que expresa el dueño del producto.

Roles no principales: Son roles externos a las necesidades de SCRUM (parte interesada, vendedores, consejeros, etc.).

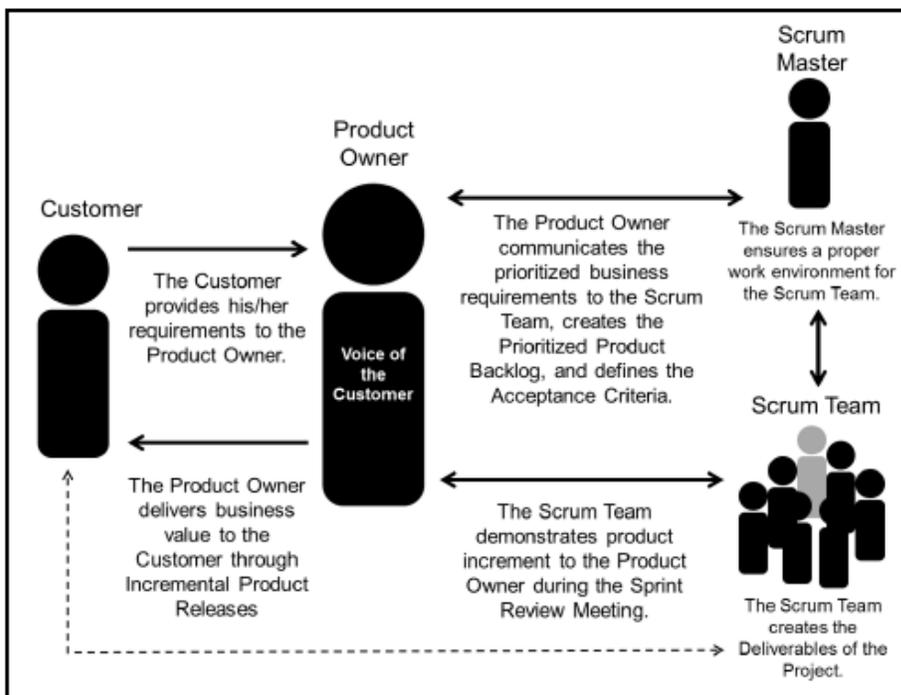


Figura 2. Organización en SCRUM. [A Guide to the Scrum Body of Knowledge (SBOK™ Guide)]

## Justificación de negocios

La justificación de negocios se refiere a la razón por la cual el proyecto a realizar o que se está en desarrollo debe seguir evolucionando, creciendo y si es siendo viable.

## **Calidad**

Es la medida de la capacidad de un producto para realizar las tareas para el cual fue creado y que estas tareas cumplan con ciertos criterios de aceptación. SCRUM al establecer una práctica de colaboración entre el equipo SCRUM y la parte interesada, garantiza, a través de retrospectivas, retroalimentación en diferentes direcciones (equipo SCRUM-Parte interesada, equipo SCRUM-equipo SCRUM) y los documentos donde se establecen prioridades (Product backlog), que se alcance un nivel de calidad aceptable.

## **Cambio**

El cambio es un factor inevitable en la Ingeniería de software, cualquier proyecto vive cambios, el equipo SCRUM tiene que tener la capacidad de minimizar los riesgos y amplificar los beneficios que vienen con el cambio.

## **Riesgo**

Factor que pone en peligro los objetivos del proyecto, éstos deben de ser identificados, evaluados y respondidos. Para estar preparado al identificar un riesgo, se debe identificar la probabilidad de que suceda y su posible impacto dentro del proyecto. Una práctica de SCRUM es identificar los riesgos posibles y más comunes que pueden pasar al trabajar en un equipo, con clientes, con personas desconocidas, entonces es factible establecer una gama de soluciones para evitar que estos riesgos impacten fuertemente a la estructura del proyecto y al equipo SCRUM en general.

## Procesos de SCRUM

SCRUM cuenta con 5 fases y un total de 19 procesos, los cuales se describirán.

Fase	Procesos
Iniciación	<ol style="list-style-type: none"><li>1. Crear la visión del proyecto.</li><li>2. Identificar SCRUM Master e interesados.</li><li>3. Formación del equipo SCRUM.</li><li>4. Creación del Product Backlog.</li><li>5. Planificación.</li></ol>
Plan y Estimación	<ol style="list-style-type: none"><li>6. Definición de los requerimientos</li><li>7. Aprobar, estimar y acordarlos requerimientos.</li><li>8. Creación de tareas.</li><li>9. Estimar tareas.</li><li>10. Crear Backlog de Sprint.</li></ol>
Implementación	<ol style="list-style-type: none"><li>11. Creación de entregables.</li><li>12. Juntas diarias.</li><li>13. Actualización del Product Backlog de prioridades</li></ol>
Revisión y la retrospectiva del trabajo	<ol style="list-style-type: none"><li>14. Convocar SCRUM.</li><li>15. Demostrar y validar Sprint.</li><li>16. Retrospectiva.</li></ol>
Entrega	<ol style="list-style-type: none"><li>17. Entregables.</li><li>18. Retrospectiva del Proyecto.</li></ol>

Figura 3. Los procesos de las cinco fases en SCRUM. [A Guide to the Scrum Body of Knowledge (SBOK™ Guide)]

### Fase de iniciación

1. Visión del proyecto. El dueño del producto da a conocer lo que quiere y es comunicado al equipo de desarrollo para dar una semblanza de hacia donde se tiene que dirigir el proyecto.
2. Identificar al SCRUM Master e interesados. Se identifican de acuerdo a sus habilidades al SCRUM master y los interesados en el proyecto.
3. Formación del Equipo SCRUM. Selección de que personal formará parte del equipo SCRUM.
4. Creación del Product Backlog. Se genera el documento Product Backlog donde se establecerán los requerimientos y su prioridad.
5. Planificación. Se empieza a dar tiempos a las iteraciones.

### Fase de plan y estimación

6. Definición de los requerimientos. Se aseguran que los requerimientos del cliente al momento de ser presentados sean entendidos.
7. Aprobar, estimar y acordarlos requerimientos. Se aprueban los requerimientos por parte del dueño del producto y, de acuerdo con la prioridad de cada requerimiento, se va dando forma a la primera iteración. Además, el SCRUM master y equipo asignan una estimación en esfuerzo a cada requerimiento
8. Creación de tareas. A partir de los requerimientos se inicia una creación de tareas de manera detallada.
9. Estimar Tareas. Después de la creación detallada de cada tarea es más sencillo darle una estimación más precisa de esfuerzo a cada tarea.

10. Crear Backlog de Sprint. Documento en el cual se especifican que tareas serán desarrolladas en la iteración.

### **Fase de implementación**

11. Creación de entregables. Conforme se van finalizando tareas se van integrando en un todo para generar los entregables correspondientes. Aquí se hace uso de los tableros para verificar los avances y darse cuenta si se está administrando el tiempo de manera favorable o, todo lo contrario.
12. Juntas diarias. Se hacen reuniones diarias del equipo SCRUM de una duración de 15 minutos para expresar dudas, problemas, avances, etc.
13. Actualización del Product Backlog de prioridades. Dado el avance que se tenga durante la iteración el product backlog de prioridades debe ser actualizado de acuerdo a la situación (desarrollo normal o con retrasos).

### **Fase de revisión y de la retrospectiva de trabajo**

14. Convocar SCRUM. El equipo SCRUM se reúne para reuniones diaria para colaborar y rastrear sus respectivos progresos, impedimentos y dependencias entre equipos. Esto generalmente cuando son proyectos muy grandes.
15. Demostrar y validar Sprint. El equipo SCRUM se reúne con la parte interesada para mostrar los avances de la iteración de acuerdo al Product backlog y esperar la validación de la parte interesada y recabar información que den para hacer mejoras al producto o en su caso entender de una mejor manera y a detalle la forma de ver de los clientes.
16. Retrospectiva. Se hace una reunión entre los miembros del equipo SCRUM para comentar las cosas que salieron exitosamente, para recalcar problemáticas y ofrecer soluciones, aspectos a mejorar o conservar para el buen funcionamiento del equipo y hacerse recomendaciones.

### **Fase de entrega**

17. Entregables. Se convoca a una reunión para presentar el paquete de entregables y si así se requiere hacer uso de las tecnologías para la instalación del sistema en las propiedades del cliente y que esté listo para su uso.
18. Retrospectiva del Proyecto. Hacer una retrospectiva general de todo el proyecto (es como una retrospectiva al final de una iteración).

# 4.Trello

## Introducción.

Trello es una plataforma dinámica de gestión de tareas usada en cualquier ámbito laboral y académico, con fines individuales o grupales. Es una forma de organización cronológica de las tareas que se tienen que desarrollar durante algún periodo a través de un tablero. Básicamente es una simulación de un tablero físico donde se usan los post-it's para verificar las tareas que tenemos pendientes.

Para el curso de Ingeniería de Software II es indispensable la creación de un tablero para cada iteración para poder verificar que todo el equipo está trabajando en el proyecto y que las tareas hayan sido equitativamente distribuidas, además de permitir hacer anotaciones sobre éstas (comentarios, prioridad, tipo de tarea, etc), responsable de realizarla y fecha de entrega. Trello avisará, vía e-mail, de las actualizaciones a actividades de los otros miembros del equipo o tareas pendientes que se acerca la fecha de vencimiento.

Una de las finalidades más importantes del uso de Trello es que los integrantes estén al tanto de la realización de sus tareas, de la fecha de entrega y la prioridad de su parte correspondiente, además poder ver el avance del resto del equipo con la finalidad de que sea transparente y coordinado, que será de ayuda también, para las revisiones con el profesor y ayudantes debido que pueden verificar el avance del equipo y comparar la entrega con el tablero en Trello (por lo cual es importante que cada miembro actualice el tablero de acuerdo a sus actividades).

Aunque existen más opciones como lo son freedcamp, basecamp, se recomienda el uso de Trello por la facilidad de uso y la experiencia que se ha tenido en cursos anteriores.

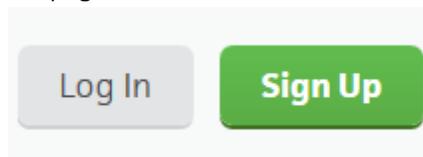
En este capítulo se tratará el manejo de Trello para mejorar la gestión de tareas entre el equipo de Ingeniería de Software II.

## Uso de Trello.

Se explicará paso por paso desde la creación de una cuenta hasta la creación de un tablero con sus respectivas tareas y propiedades que se le pueden asignar.

### Creación de una sesión en Trello.

1. Dirigirse al siguiente enlace <https://trello.com/>, para después hacer clic en "Sign up" en la esquina superior derecha de la página.



Aparecerá la siguiente sección en dónde se tendrán que ingresar datos personales (Nombre, correo, contraseña y aceptar términos y condiciones).

## Create a Trello Account

Name

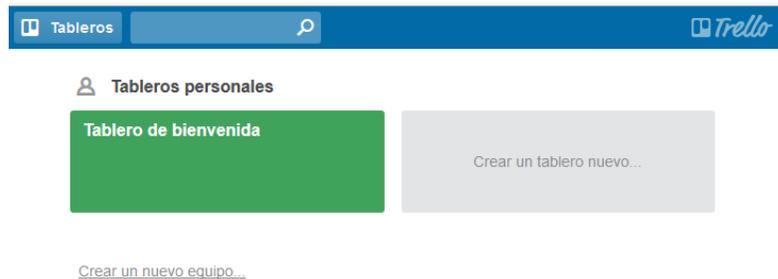
Email

Password

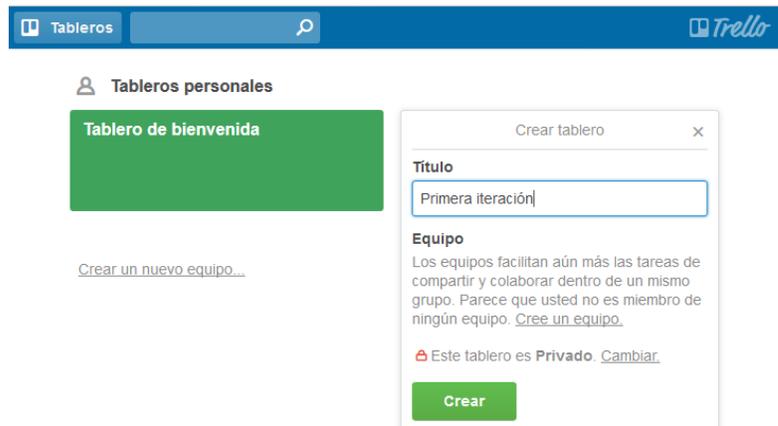
I accept the [Terms of Service](#) and [Privacy Policy](#)

Previamente llenados estos campos se dará clic en “Create New Account”. Y aparecerá la siguiente página.

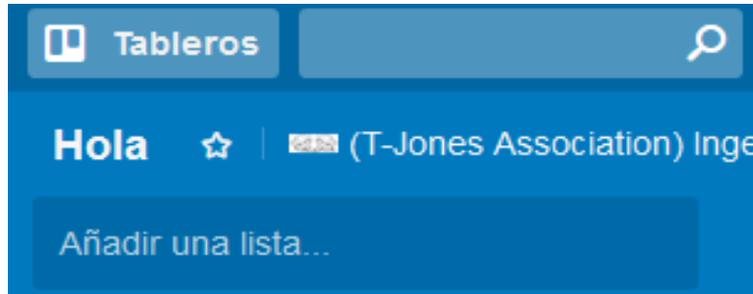
## Creación y uso de un tablero en Trello.



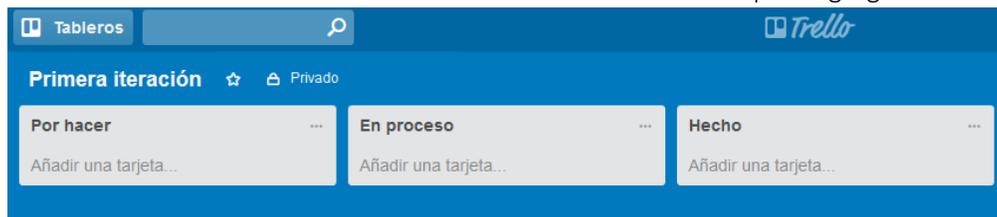
2. Para crear nuestro primer tablero es necesario dar clic en “Crear un tablero nuevo”. Y darle un título. Por ejemplo “Primera iteración” y de preferencia deberá ser privado para que sólo los miembros con invitación puedan verlo y/o editarlo. Y dar clic en “Crear”



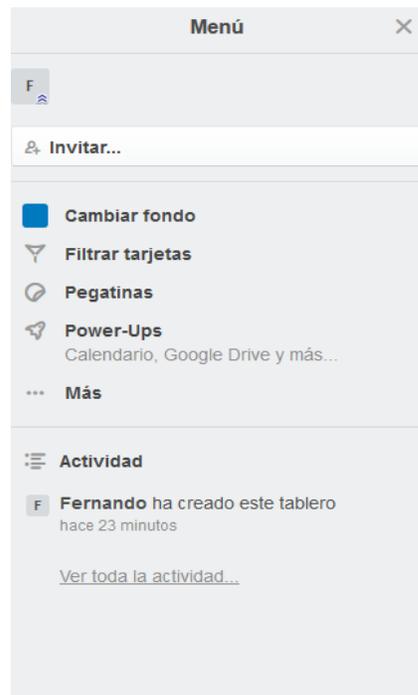
3. Empezar a hacer uso del tablero con fines de que nuestro proyecto sea bien coordinado se crean tres listas, dando clic en “Añadir una lista”. La primera tendrá el nombre de “Por hacer”; la segunda será nombrada “En proceso”; la tercera “Hecho”.



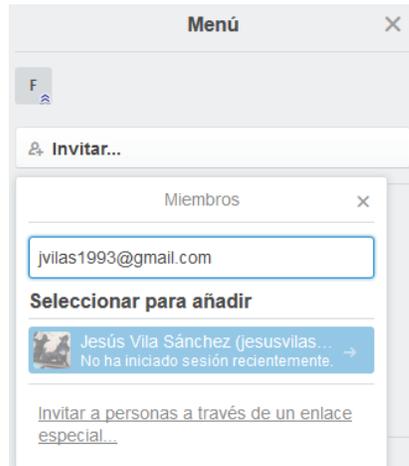
4. Ahora nuestro tablero esta seccionado en tres columnas listo para agregar nuevas tareas.



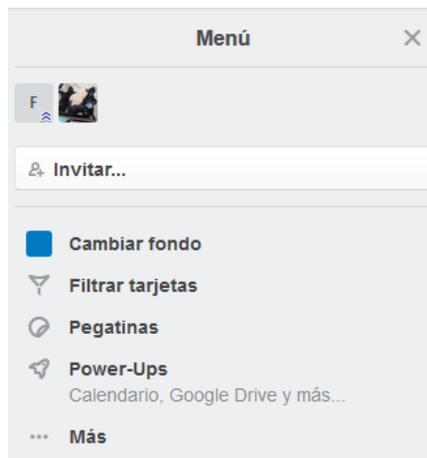
5. Invitar a los miembros del equipo a ser parte del tablero para la coordinación de tareas.



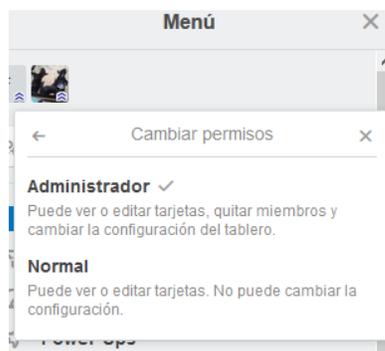
Damos clic en “Invitar...”. Después introducimos un correo electrónico para enviar invitación y en dado caso de que el usuario ya este registrado aparecerá lo siguiente.



Y damos clic en el usuario a agregar.



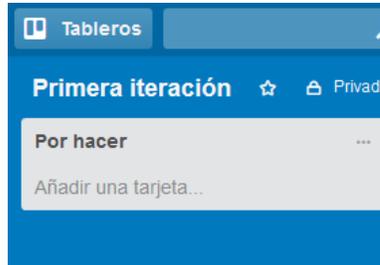
Y ya aparecerá en los usuarios y tenemos que cambiar sus permisos a “Administrador” haciendo clic en la foto del miembro y dando clic en “Administrador”.



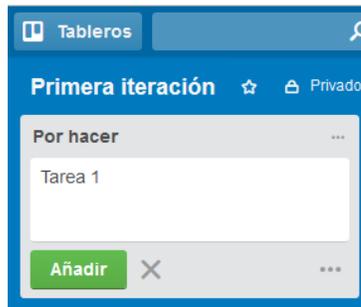
Así se pueden agregar los miembros de todo el equipo.

## Creación y propiedades de tareas en un tablero en Trello.

6. Para empezar a asignar y registrar las tareas entre todo el equipo se habilita la columna de “Por hacer” y se da clic en “Añadir una tarjeta”. Para cada tarea que se tiene que hacer en el proyecto se debe poner una tarjeta, inicialmente en la columna “Por hacer”



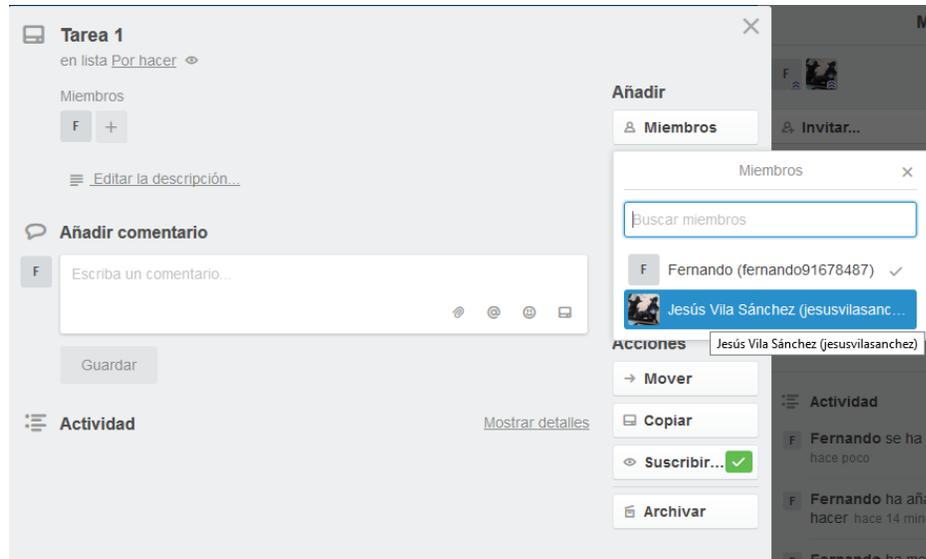
Se le asigna un nombre a la tarjeta.



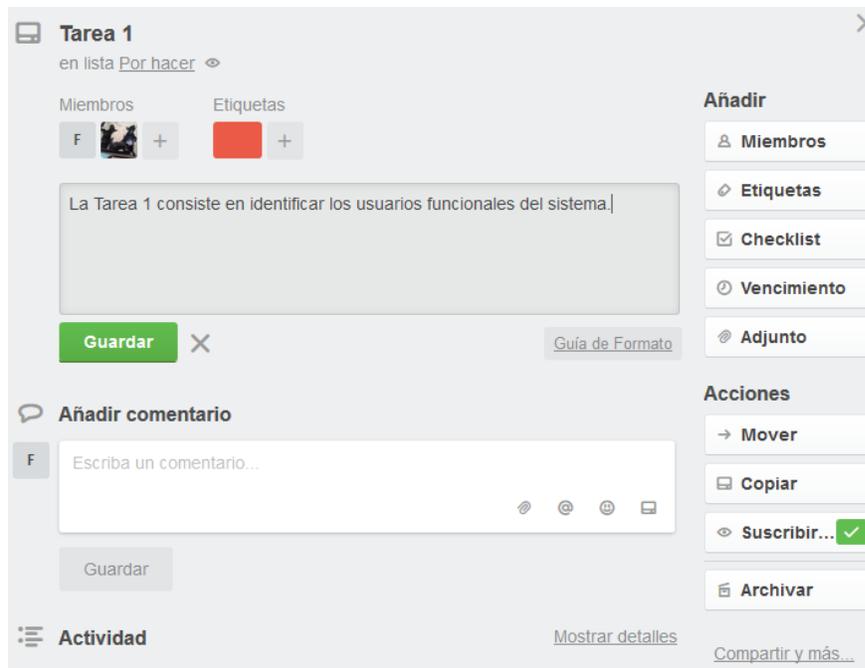
Se selecciona “Añadir” y se selecciona haciendo clic sobre la tarea para editar sus atributos.



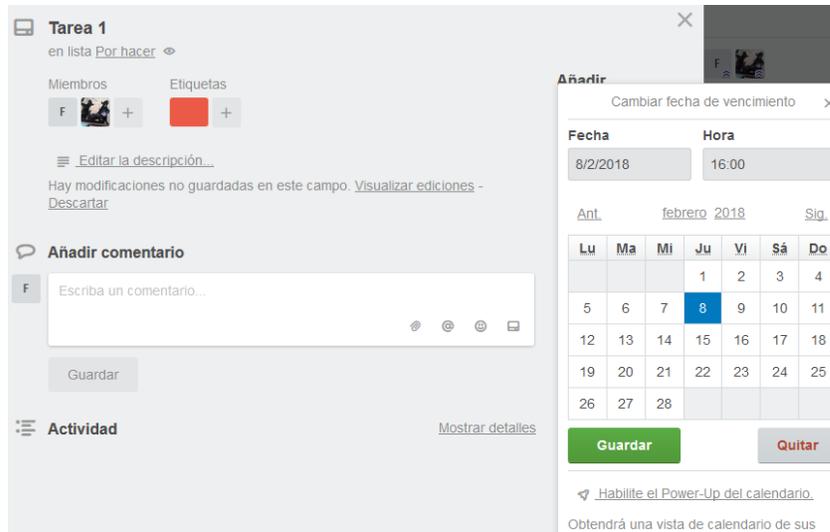
Se agregan el miembro(s) responsable(s) de cumplir la Tarea 1.



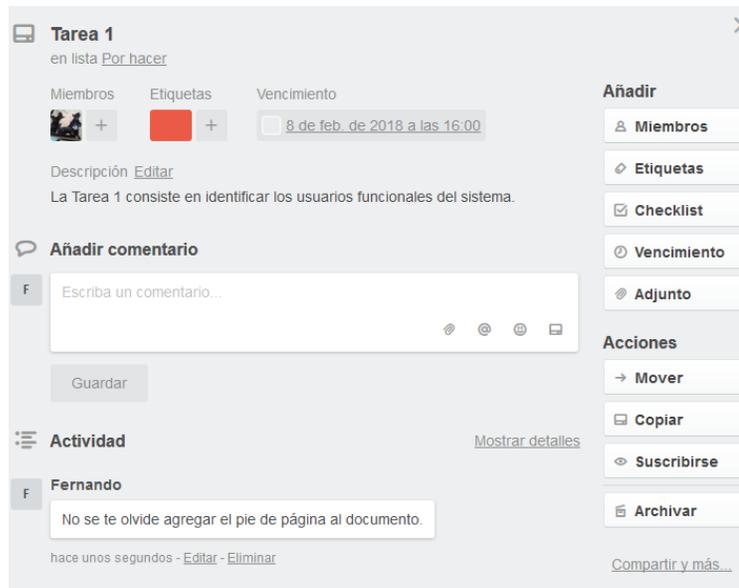
Se le asigna una descripción a la Tarea 1 y un color que nos serviría para identificar si es tarea correspondiente a documentación (rojo) o a código (azul, por ejemplo).



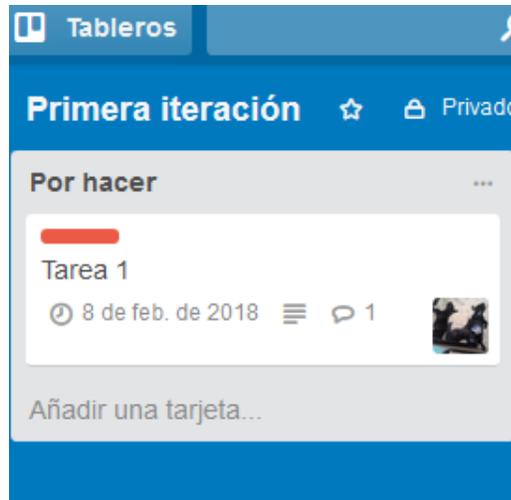
Después, para finalizar la asignación de atributos a la Tarea 1 se le asigna una fecha de vencimiento.



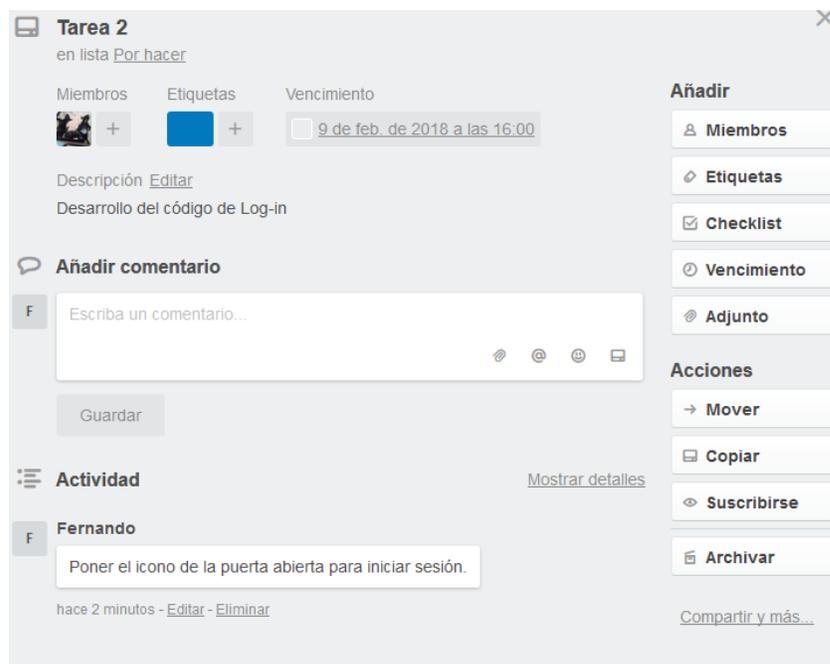
Se pueden agregar comentarios que se registrarán en la “Actividad” que ha tenido la tarea, como anotaciones, pequeños recordatorios o detalles a modificar.



La Tarea 1 estará en la columna de “Por hacer”

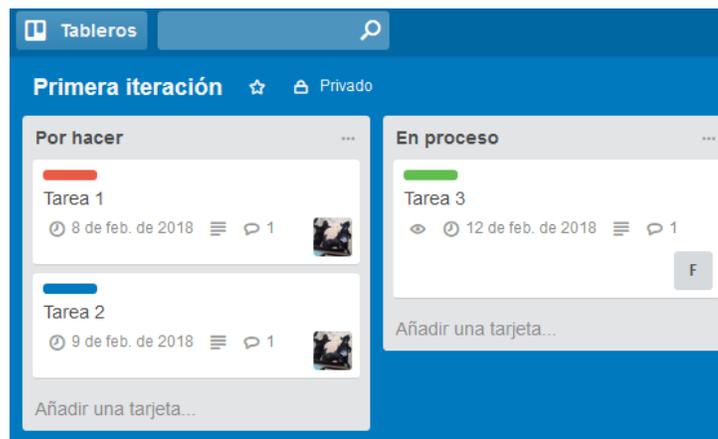


Y se pueden agregar las tareas a cumplir en la iteración uno.

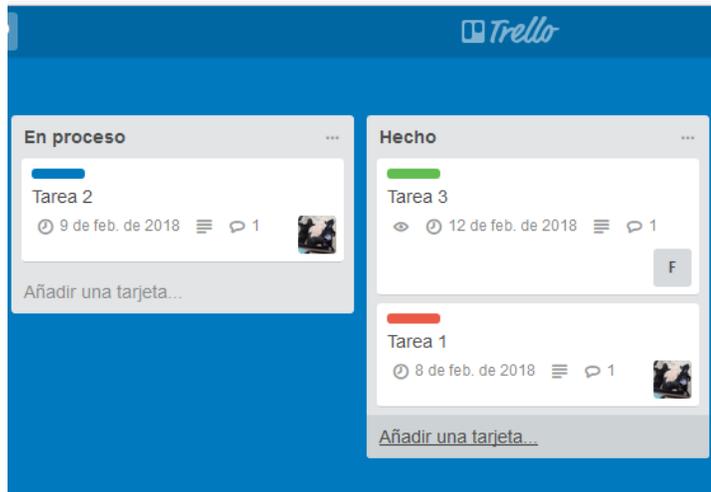




Si alguna tarea ya se está realizando a hacer se mueve a la columna “En proceso”



Si alguna tarea ya fue terminada se colocan en la columna de “Hecho”.



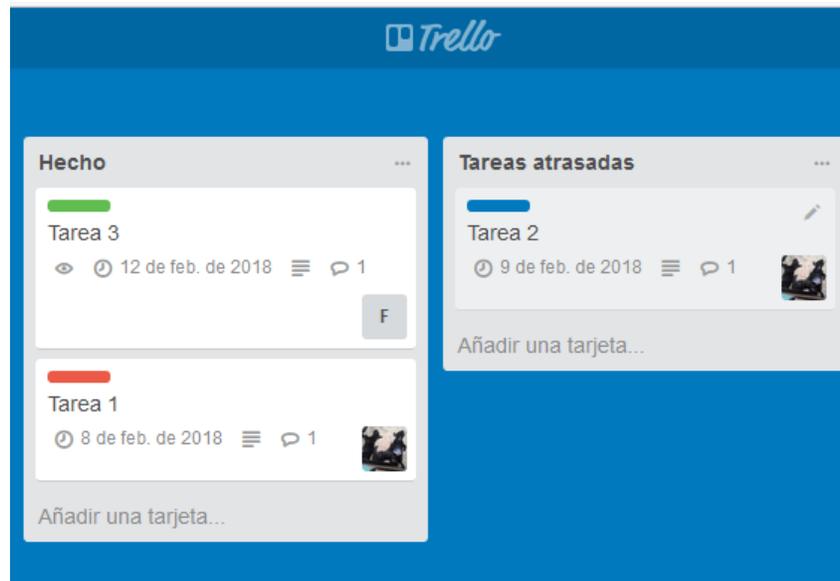
## Ventajas del uso de un tablero de control de tareas

El uso y actualización del tablero en cualquier momento del proyecto es necesario para mantenerse al corriente del avance del proyecto, así como en casos de emergencia, es decir, si algún miembro decide abandonar la clase de Ingeniería de Software II a la mitad del semestre, los demás miembros del equipo pueden identificar las tareas que tenía asignadas, este miembro, para reasignarlas y cumplir responsablemente con la entrega.

Muchas de las ventajas son evidentes al usar este tipo de herramientas ya que brindan un orden y coordinación, lo cual es elemental para la realización de un proyecto sin importar la magnitud y ámbito en que se le ocupe.

A la hora de hacer la retrospectiva al final de la iteración, se puede identificar rápidamente, si se ha cumplido con el cien por ciento de las tareas de la iteración. En caso contrario, si alguna tarea no fue acabada, se puede acomodar en una columna especial de “Tareas atrasadas” y hacer mención de este caso en la entrega y/o en junta con los clientes y llegar a un acuerdo para incluirla en la siguiente iteración.

Ejemplo:



Otra de las ventajas, es que Trello notifica por medio de correo electrónico conforme la fecha de vencimiento se acerca y aún pasada esta fecha te notifica que hay tareas pendientes que ya “caducaron”. Es decir, Trello hace una insistencia tal, que es imposible ignorar o hacer caso omiso y por lo tanto tener en mente las fechas, en las cuales hay que entregar las tareas correspondientes.

Este tipo de herramientas (tableros) deja en claro el trabajo y el responsable que debe cumplirlo así se puede evitar la mala organización y la asignación no equitativa de trabajo, pues la manera de trabajo se vuelve transparente y si algún miembro del equipo no se siente conforme con la carga de trabajo correspondiente, puede verificar la carga de trabajo asignada a los demás miembros y proponer un cambio de tareas entre sí.

Un tip para hacer un uso precavido con ayuda de Trello o cualquier herramienta similar, en cuestión de fechas es lo siguiente:

Supongamos que la primera iteración tiene como fecha límite de entrega el 16 de octubre y la fecha del presente es 2 de octubre. Para evitar contratiempos a la hora de unir bloques de código o varias partes de un documento encargado a más de un miembro del equipo en una sola versión y también, descartar que haya más de una versión final, la fecha límite para que todos los miembros del equipo reporten sus tareas como finalizadas se establecería al menos 3 o 4 días antes de la fecha de entrega, es decir, el 12 de octubre sería la entrega entre miembros del equipo y se convocaría a una junta para juntar todos los módulos que forman parte de la entrega y previamente quede todo dispuesto para la entrega de la iteración.

## 5. Construcción del Sistema Web.

El Planteamiento del problema o Enunciado del problema expone, de manera detallada, la descripción del producto en base a las necesidades del cliente para que el proyecto cumpla con el cien por ciento de las funcionalidades y las expectativas sean exitosamente cubiertas.

Generalmente el cliente debe brindar toda la información para que este documento pueda ser generado.

Se ejemplificará el planteamiento de un problema real resuelto en el caso de Ingeniería de Software II (esta sólo es una parte de la descripción del problema que los clientes brindaron para el inicio del proyecto. El documento completo se puede consultar en el **Anexo A. A2. Plan del Proyecto.**):

### **Planteamiento del Problema de la Secretaría de Educación Abierta y continua de la Facultad de Ciencias.**

La Secretaría de Educación Abierta y Continua (SEAYC) de la Facultad de Ciencias tiene la función de promover y organizar cursos, diplomados, talleres, conferencias y seminarios afines a las áreas del conocimiento de la Facultad de Ciencias. Dicha secretaria desea dar impulso a los cursos de educación continua que profesores de la Facultad o profesionales con experiencia probada proponen para el público en general. Los cursos y los ponentes son sometidos a revisión por un Comité Académico para realizar observaciones y autorizar o no el curso o al ponente.

Actualmente, las propuestas se registran mediante un formato en Word (descargable) donde se describe el curso, y otro formato para el currículum del coordinador del curso y de los ponentes. Después, la SEAYC pasa las propuestas al Comité Académico para realizar observaciones que deberán ser atendidas por aquellos que proponen los cursos, además se lleva un historial de modificaciones de las propuestas.

Tanto si las propuestas como los ponentes son aceptados por el Comité Académico, entonces la SEAYC solicita la autorización del Consejo Técnico de la Facultad para enviarla (manualmente) al departamento de contabilidad, de manera que se puedan recibir los pagos de los inscritos.

Por otro lado, se quiere tener una base de datos de los interesados a los cursos mediante un registro a través de la página de la SEAYC. Donde se lleva el registro del número de interesados para cada uno de los cursos que se ofrecen, así cuando un curso ya tiene un número importante de interesados, la encargada de los cursos envía de manera manual, al coordinador del área el reporte de interesados para cada curso registrado. Con este reporte, se toman las decisiones sobre qué cursos se deben organizar, comunicar a los ponentes para definir fechas y enviar una notificación a los interesados para invitarlos a inscribirse a los cursos.

El catálogo de cursos y diplomados, no se encuentra clasificado por área de conocimiento para difundir la oferta académica de la Facultad de Ciencias. El catálogo de cursos propone a aquellos que están interesados en algún curso a registrarse en la página de la SEAYC.

El problema con este sistema es que no se cuentan con alertas que indiquen al coordinador del área si hay un curso que tenga ya el número de interesados el cual valga la pena tomar en cuenta para proceder a organizarlo y eventualmente ponerlo en marcha. El proceso se tiene que hacer a petición del coordinador y la revisión del número de interesados, para cada curso, se debe hacer de manera manual.

Por esto la SEAYC necesita un Sistema de apoyo a las funciones de registro de interesados, de ponentes, de cursos y que proporcione una alerta global de notificaciones a los interesados y a la secretaria para alertar al coordinador del área cuando los cursos ya tienen suficientes interesados para iniciar su organización y posteriormente su promoción.

## **Desarrollo de la primera, segunda y sexta iteración.**

En esta sección se mostrará el desarrollo, en específico de algunas iteraciones, de la primera, segunda, y última iteración, con el propósito de mostrar la evolución y la forma de trabajo implementada y previamente detallada en los capítulos anteriores. Se pretende explicar a detalle, la parte teórica, para después ejemplificar y mostrar la teoría puesta en práctica.

El desarrollo del proyecto se basará en los procesos de SCRUM por etapa y las plantillas para cada documento que se brindarán durante el curso.

### **Fase de iniciación:**

1. Crear la visión del proyecto: Los alumnos acudirán a una junta, con los “Clientes Reales” y la profesora en la cual los “Clientes Reales” establecerán de manera detallada las necesidades del curso, estas se transformarán en el “Planteamiento del Problema” (el cual está explicado y se muestra un ejemplo en el Capítulo 5).
2. Identificar SCRUM Master e interesados: Al finalizar la junta los “Clientes Reales” deberán asignar a alguien que los represente, quien estará en contacto, y al pendiente de correos y otras juntas, con los alumnos. Éstos a su vez, en su equipo, elegirán a un miembro como SCRUM master.
3. Formación del equipo SCRUM: En un equipo debe existir una organización sólida para un flujo de trabajo armónico y eficaz, por lo tanto, las responsabilidades de un proyecto de tal magnitud tienen que ser repartidas equitativamente entre el equipo. Este tipo de responsabilidades pueden ser del tipo técnicas, las cuales tienen que ver con el desarrollo de software y correcta instalación de las plataformas y su versión necesaria para el adecuado trabajo del equipo; las responsabilidades sobre la creación de la documentación con calidad para llevar un orden por escrito de las tareas que se utilizaron así como diagramas y modelos utilizados durante el desarrollo del proyecto y repartición de tareas a los miembros del equipo; finalmente, el seguimiento de la planeación del proyecto a través del tablero para la verificación de avances, responsables y fechas.
4. Creación del Product Backlog: En una reunión con los “Clientes Reales” se identifican en una lista las necesidades asignándoles prioridad, una estimación horas y a qué iteración pertenecerá. (Al final del punto 5, de esta fase, se mostrará un ejemplo del Product Backlog inicial).
5. Planificación: Se empiezan a identificar las tareas con mayor prioridad, según las necesidades de los clientes.



### Product backlog inicial.

Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs	0
02	Registrar CV e información de los ponentes	Alta	6hrs.	1
03	Modificar CV e información de los ponentes	Media	2hrs.	1
04	Registrar propuesta de curso	Alta	6hrs.	1
05	Modificar propuesta de curso	Media	2hrs.	1
06	Borrar curso	Media	4hrs.	1
07	Documentación de la primera iteración	Alta	11.75 hrs.	1
08	Aprobar a los ponentes	Baja	1hrs.	2
09	Agregar Administradores	Baja	4hrs.	2
10	Autorizar Curso	Alta	4hrs.	2
11	Buscar curso	Media	4hrs..	2
12	Documentación de la segunda iteración	Alta	9.75hrs.	2
13	Registrar en la página de SEAYC	Media	5hrs.	3
14	Interesarse en un curso	Baja	4hrs.	3
15	Documentación de la tercera iteración	Alta	6.5hrs	3
16	Enviar notificación	Media	2hrs.	4
17	Documentación de la cuarta iteración	Alta	8hrs.	4

18	Exportar a formato PDF la información de los ponentes	Alta	6hrs.	5
19	Exportar a formato PDF la información de los cursos	Alta	2hrs.	5
20	Documentación de la quinta iteración	Alta	5.75hrs.	5
21	Documentación de la sexta iteración	Alta	19hrs	6

## Fase de plan y estimación

- Definición de los requerimientos: El equipo SCRUM debe describir los requerimientos de manera entendible para evitar ambigüedad en estos, además si existen dudas de la funcionalidad de algún requerimiento, inmediatamente se acude con el representante de los “Clientes Reales” para la aclaración de este. Al final del punto 10 se muestra con un ejemplo, el diagrama de casos de uso final, el cual describen las tareas que los usuarios pueden realizar dentro del sistema.
- Aprobar, estimar y acordarlos requerimientos: Cuando se tengan bien entendidos y desarrollados los requerimientos del proyecto, se reúnen, el equipo SCRUM y los “Clientes Reales” para que se les dé el visto bueno y se empieza a diseñar la primera iteración.
- Creación de tareas: El equipo SCRUM se coordina para separar los requerimientos en tareas.
- Estimar Tareas. Después de la creación detallada de cada tarea es más sencillo darle una estimación más precisa de esfuerzo a cada tarea.
- Crear Plan de la iteración: Se asignan las tareas a desarrollar en la iteración correspondiente (Se mostrará el plan de la primera y segunda iteración en las siguientes).

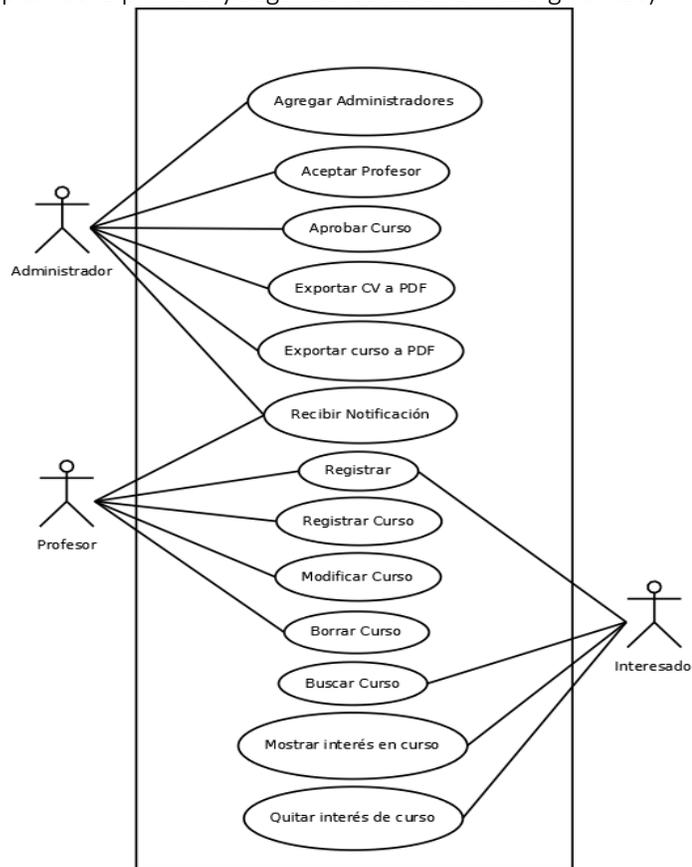


Ilustración 4. Diagrama de casos de uso. Extraído del Anexo A. A1. Requerimientos Generales.

La documentación completa en este punto reside en el **Anexo A. A1. Requerimientos Generales**.

### Plan de la primera iteración.

ID	Funcionalidad	Responsable	Tiempo	Estado
1.1	Registrar propuesta de curso	Luisita	1hr.	Por hacer
1.2	Implementar una plantilla para registrar cursos, ponentes (CV e información académica)	Pedrito	1hr.	Por hacer
1.3	Diseñar base de datos	Sotanito	2hrs.	Por hacer
1.4	Crear diagrama relacional	Sotanito	1hr.	Por hacer
1.5	Diseñar la vista principal	Pedrito	3hrs.	Por hacer
1.6	Crear página principal	Pedrito	1hr.	Por hacer
1.7	Crear vista para registrar curso	Pedrito	2hrs.	Por hacer
1.8	Crear vista para registrar CV e información académica del Ponente	Sotanito	1hr.	Por hacer
2.1	Implementar la agregación a la base de datos de Cursos	Luisita	1hr.	Por hacer
2.2	Implementar la eliminación a la base de datos de Cursos	Luisita	1hr.	Por hacer
2.3	Implementar la consulta a la base de datos de Cursos	Luisita	45min.	Por hacer
2.4	Implementar la agregación a la base de datos de Ponente	Pedrito	45min.	Por hacer
2.5	Implementar la eliminación a la base de datos de Ponente	Pedrito	45min.	Por hacer
2.6	Implementar la consulta a la base de datos de Ponente	Pedrito	45min	Por hacer
3.1	Editar datos de los Cursos	Sotanito	2hrs.	Por hacer
3.2	Editar CV e información académica de los Ponentes	Sotanito	1hr.	Por hacer
4.1	Minutas de 15 min diarias	Todo el equipo	1.5hrs.	Por hacer
4.2	Documento de requerimientos de la primera iteración	Pedrito	1hr.	Por hacer
4.3	Documento del plan de la primera iteración	Pedrito	2hrs.	Por hacer
4.4	Documento de arquitectura de la primera iteración	Sotanito	2hrs.	Por hacer
4.5	Retrospectiva de la primera iteración	Luisita	1hrs.	Por hacer
4.6	Product Backlog de la primera iteración modificado.	Sotanito	30min.	Por hacer
4.7	Actualización del tablero en Trello	Todo el equipo	15min.	Por hacer
4.8	Realizar presentación de los avances	Luisita	2hrs.	Por hacer
4.9	Junta con los clientes para presentar avances	Todo el equipo	1.5hrs.	Por hacer

## Plan de la segunda iteración.

ID	Funcionalidad	Responsable	Tiempo	Estado
1.1	Implementar la vista de la página con el modelo de la SEAYC (marcos y estilo)	Luisita	2hrs.	Por hacer
2.1	Aceptar/Rechazar propuesta de Ponente	Pedrito	2hr.	Por hacer
2.2	Aceptar/Rechazar propuesta de Curso	Pedrito	1hr.	Por hacer
2.3	Crear Súper Administrador	Sotanito	1hr.	Por hacer
2.4	Agregar más administradores (Súper Administrador o administradores)	Sotanito	1hr.	Por hacer
3.1	Manejo de sesiones en el sistema	Pedrito	2hrs.	Por hacer
3.2	Mostrar mensajes de error/confirmación	Sotanito	1hr.	Por hacer
4.1	Crear vista para registrar Interesado	Luisita	1hr.	Por hacer
4.2	Vista de la búsqueda de cursos para el Interesado	Luisita	2hrs.	Por hacer
5.1	Minutas de 15 min diarias	Todo el equipo	1hr.	Por hacer
5.2	Documento de requerimientos de la segunda iteración	Luisita	45min.	Por hacer
5.3	Documento del plan de la segunda iteración	Pedrito	45min.	Por hacer
5.4	Documento de arquitectura de la segunda iteración y actualización de la base de datos.	Pedrito	1hr.	Por hacer
5.5	Retrospectiva de la segunda iteración	Pedrito	45min	Por hacer
5.6	Product Backlog de la segunda iteración modificado.	Sotanito	2hrs.	Por hacer
5.7	Actualización del tablero en Trello	Sotanito	1hr.	Por hacer
5.8	Realizar presentación de los avances	Todo el equipo	1.5hrs.	Por hacer
5.9	Junta con los clientes para presentar avances	Todo el equipo	1hr.	Por hacer

La documentación completa en este punto reside en el **Anexo B. B2. Plan de la primera iteración.** y en el **Anexo C. C2. Plan de la segunda iteración.**, respectivamente.

## Fase de implementación

11. Creación de entregables: Se van creando los productos de trabajo y conforme se acerca el tiempo de entrega se van alistando para entregarlos y presentarlos a los clientes.
12. Juntas diarias: El equipo SCRUM se debe de reunir diariamente a una sesión de 15 minutos para discutir problemas, avances, etc. O expresar dudas y ser escuchado.
13. Actualización del Product Backlog: En una reunión entre los miembros del equipo se realiza una revisión para cotejar las actividades de mayor prioridad establecidas en el Product Backlog y establecer el estado de estas actividades para la entrega de la iteración, así como para integrar nuevas tareas que no fueron consideradas con una alta prioridad y definir cambios en las tareas realizadas (Se mostrará con ejemplos, el product backlog de la primera y segunda iteración de acuerdo al avance obtenido en cada una de las iteraciones, basados en el proyecto de ejemplo).

### Product backlog primera iteración.

Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs.	0
02	Registrar CV e información de los ponentes	Alta	6hrs.	1
03	Modificar CV e información de los ponentes	Media	2hrs.	1
04	Registrar propuesta de curso	Alta	6hrs.	1
05	Modificar propuesta de curso	Media	2hrs.	1
06	Borrar curso	Media	4hrs.	1
07	Generación de documentos, juntas, minutas, etc.	Alta	11.75 hrs.	1

### Product backlog segunda iteración.

Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs.	0
02	Registrar CV e información de los ponentes	Alta	6hrs.	1
03	Modificar CV e información de los ponentes	Media	2hrs.	1
04	Registrar propuesta de curso	Alta	6hrs.	1
05	Modificar propuesta de curso	Media	2hrs.	1
06	Borrar curso	Media	4hrs.	1
07	Documentación de la primera iteración	Alta	11.75 hrs.	1

08	Aprobar a los ponentes	Baja	1hrs.	2
09	Agregar Administradores	Baja	4hrs.	2
10	Autorizar Curso	Alta	4hrs.	2
11	Buscar curso	Media	4hrs..	2
12	Documentación de la segunda iteración	Alta	9.75hrs.	2

## Fase de revisión y de la retrospectiva de trabajo

14. Convocar SCRUM: El equipo SCRUM se reúne para colaborar y rastrear sus respectivos progresos, impedimentos y dependencias.
15. Demostrar y validar Sprint. El equipo SCRUM se reúne con la parte interesada para mostrar los avances de la iteración de acuerdo al Product backlog. Se da retroalimentación de ambas partes para identificar lo que queda pendiente para el trabajo de la siguiente iteración (se muestra el product backlog con todas las actividades marcadas en realizado, al final del punto 16 aparece la tabla).
16. Retrospectiva. Los miembros del equipo SCRUM se reúnen a hacer la retrospectiva de la iteración, esto es, para platicar la experiencia de trabajo y declarar los éxitos y fracasos o problemas que se tuvieron durante la iteración y se dan ideas para mejorar y corregir problemas para ser implementados en la siguiente iteración. El documento de retrospectiva se muestra en el **Anexo B. B4. Retrospectiva.** y **Anexo C. C4. Retrospectiva.** para primera y segunda iteración, respectivamente.

### Product backlog sexta iteración.

Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs	0
02	Registrar CV e información de los ponentes	Alta	6hrs.	1
03	Modificar CV e información de los ponentes	Media	2hrs.	1
04	Registrar propuesta de curso	Alta	6hrs.	1
05	Modificar propuesta de curso	Media	2hrs.	1
06	Borrar curso	Media	4hrs.	1
07	Documentación de la primera iteración	Alta	11.75 hrs.	1
08	Aprobar a los ponentes	Baja	1hrs.	2
09	Agregar Administradores	Baja	4hrs.	2
10	Autorizar Curso	Alta	4hrs.	2
11	Buscar curso	Media	4hrs..	2
12	Documentación de la segunda iteración	Alta	9.75hrs.	2
13	Registrar en la página de SEAYC	Media	5hrs.	3
14	Interesarse en un curso	Baja	4hrs.	3
15	Documentación de la tercera iteración	Alta	6.5hrs	3
16	Enviar notificación	Media	2hrs.	4
17	Documentación de la cuarta iteración	Alta	8hrs.	4

18	Exportar a formato PDF la información de los ponentes	Alta	6hrs.	5
19	Exportar a formato PDF la información de los cursos	Alta	2hrs.	5
20	Documentación de la quinta iteración	Alta	5.75hrs.	5
21	Documentación de la sexta iteración	Alta	19hrs	6

## Fase de entrega

17. Entregables. Se convoca a una reunión para presentar el paquete de entregables y si así se requiere hacer uso de las tecnologías para la instalación del sistema en las propiedades del cliente y que esté listo para su uso.
18. Retrospectiva del Proyecto. Al terminar todo el proyecto se hace una retrospectiva general de todo el proyecto (es como una retrospectiva al final de una iteración).

## 6.Resultado del Sistema.

Después de acabar con las iteraciones de trabajo de la materia, es preciso convocar una reunión con los miembros del equipo SCRUM, con la profesora y con los “Clientes Reales”. Todo esto para presentar el sistema que fue el resultado de un semestre de trabajo en la materia de Ingeniería de software II.

En el presente capítulo se mostrará el resultado, como manual el sistema creado para la SEAyC.

## Página de inicio.

La página de inicio se muestra con la misma forma a la página oficial de la SEAYC, además cuenta con varias secciones:

1. Barra amarilla:  
Se presenta el icono de “home” que nos redirige a este mismo sitio (<http://seayc.herokuapp.com/>)  
Muestra la fecha (Día, mes y año).  
Icono de ingresar (Ponente o administrador).
2. Barra azul:  
Esta el nombre de la SEAYC y de la FC (nos lleva a estos sitios respectivamente).
3. Menú  
Muestra las opciones: Registrar ponente; Catálogo de Cursos; ¿Te interesa algún curso? Regístrate.



Las funcionalidades de este proyecto residen en este menú:

[REGISTRAR PONENTE](#)

[CATÁLOGO DE CURSOS](#)

[¿TE INTERESA ALGÚN CURSO?  
REGISTRATE](#)

# 1. ¿TE INTERESA ALGÚN CURSO? REGÍSTRATE

[REGISTRAR PONENTE](#)

[CATÁLOGO DE CURSOS](#)

[¿TE INTERESA ALGÚN CURSO?  
REGÍSTRATE](#)

Se despliega un formulario con datos necesarios solamente para que algún interesado se registre.

Secretaría de Educación Abierta y Continua Facultad de Ciencias

[REGISTRAR PONENTE](#)  
[CATÁLOGO DE CURSOS](#)  
[¿TE INTERESA ALGÚN CURSO?  
REGÍSTRATE](#)

### Registro Estudiante

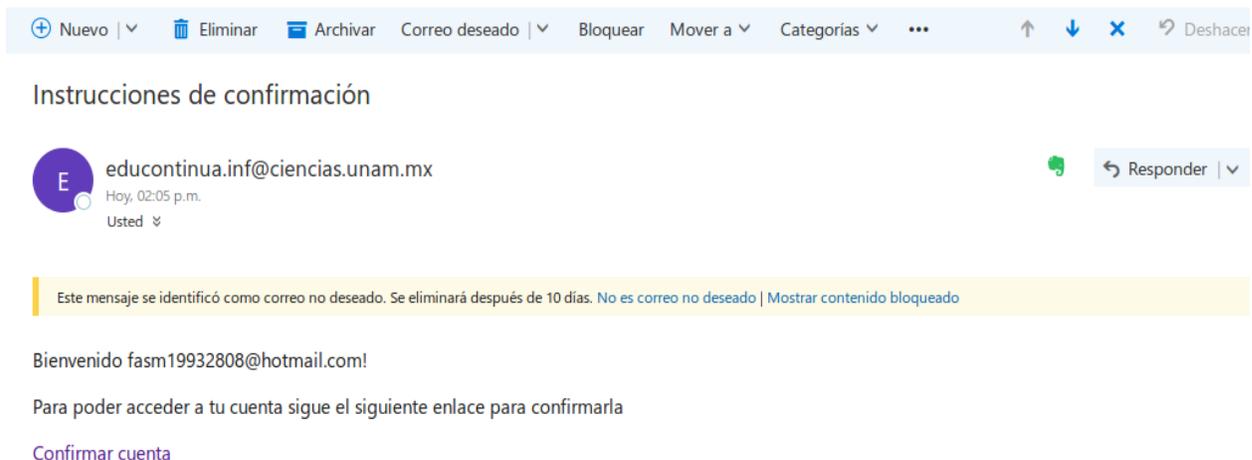
AVISO DE PRIVACIDAD La Secretaría de Educación Abierta y Continua de la Facultad de Ciencias de la UNAM está comprometida con la protección de sus datos personales, al ser responsable de su uso, manejo y confidencialidad, por lo que no compartimos sus datos personales con ninguna otra instancia. Para leer el aviso de privacidad completo [haga click aquí](#)

**Datos Personales** (Todos los campos son obligatorios)

* Nombre	* Apellido Paterno	* Apellido Materno
<input type="text" value="Fernando"/>	<input type="text" value="Sánchez"/>	<input type="text" value="Montoya"/>
* Correo electrónico	* R.F.C	
<input type="text" value="fasm19932808@hotmail.com"/>	<input type="text" value="fasm28081993"/>	
* Teléfono local	* Teléfono móvil	
<input type="text" value="5156165552"/>	<input type="text" value="1565454515"/>	
Contraseña	<input type="password" value="*****"/>	
Confirmar contraseña	<input type="password" value="*****"/>	

Cuando se ha terminado de llenar el formulario, y este sea correcto (todos los campos requeridos y 8 caracteres mínimo para la contraseña). Se da clic en “Registrarse” y llegara un correo para confirmar la cuenta a la dirección de correo proporcionada.

Si se abre el correo de confirmación, y se desea seguir con la confirmación de la cuenta se da clic en “Confirmar cuenta”



Volvemos al sitio para confirmar y acceder a nuestro espacio.



Al confirmar, la contraseña se accede correctamente a la sesión.


**Universidad Nacional Autónoma de México**



5 de Octubre del 2016 Fernando 

**Secretaría de Educación Abierta y Continua** Facultad de Ciencias

Sesión Iniciada. ✕

[CATÁLOGO DE CURSOS](#)

[MIS CURSOS](#)

### Instructivo para solicitar impartir Cursos, Diplomados, Talleres o Seminarios de Educación Continua

**Estimado profesor:**

Si tiene interés en impartir algún curso, Diplomado o Taller especializado con temas de su dominio, lo invitamos a proponerlo a esta Secretaría entregando la documentación necesaria de acuerdo a los siguientes formatos:

Aquí podemos checar la lista de cursos disponibles y ver los cursos en los cuales te has interesado (de acuerdo con el menú que se encuentra en la parte izquierda de la pantalla). Se puede realizar una búsqueda de los cursos dados de alta en la SEAYC.


**Universidad Nacional Autónoma de México**



5 de Octubre del 2016 Fernando 

**Secretaría de Educación Abierta y Continua** Facultad de Ciencias

[CATÁLOGO DE CURSOS](#)

[MIS CURSOS](#)

## Cursos disponibles

[Mostrar todos los cursos](#)

O si el alumno ya se registró en algún curso, se puede hacer clic en “Mis cursos”


**Universidad Nacional Autónoma de México**  



5 de Octubre del 2016 Fernando  Salir

**Secretaría de Educación Abierta y Continua** Facultad de Ciencias

[CATÁLOGO DE CURSOS](#)  
[MIS CURSOS](#)

## Mis cursos

Cuando se hayan hecho todas las consultas necesarias es sencillo el cerrar la sesión dando clic al recuadro de “salir” en la Barra Amarilla.

5 de Octubre del 2016  Ingresar

**Secretaría de Educación Abierta y Continua** Facultad de Ciencias

Sesión finalizada. ✕

[REGISTRAR PONENTE](#)  
[CATÁLOGO DE CURSOS](#)  
[¿TE INTERESA ALGÚN CURSO? REGISTRATE](#)

### Instructivo para solicitar impartir Cursos, Diplomados, Talleres o Seminarios de Educación Continua

Estimado profesor:

Si tiene interés en impartir algún curso, Diplomado o Taller especializado con temas de su dominio, lo invitamos a proponerlo a esta Secretaría entregando la documentación necesaria de acuerdo a los siguientes formatos:

Lo cual nos redirige a la Página Principal.

## 2. REGISTRAR PONENTE

[REGISTRAR PONENTE](#)

[CATÁLOGO DE CURSOS](#)

[¿TE INTERESA ALGÚN CURSO?  
REGISTRATE](#)

Aparece el formulario con Datos Personales, Formación Académica, Experiencia Académica, Experiencia Profesional y Reconocimientos. Se llenan los apartados.

### Registro Ponente Currículum vitae

**AVISO DE PRIVACIDAD** La Secretaría de Educación Abierta y Continua de la Facultad de Ciencias de la UNAM está comprometida con la protección de sus datos personales, al ser responsable de su uso, manejo y confidencialidad, por lo que no compartimos sus datos personales con ninguna otra instancia. Para leer el aviso de privacidad completo [haga click aquí](#)

#### Datos Personales (Todos los campos son obligatorios)

\* Nombre

Fernando

\* Apellido Paterno

Sánchez

\* Apellido Materno

Montoya

\* Correo electrónico

cafetafer@ciencias.unam.mx

¿Publicar correo?

Sí No

\* R.F.C

fasm19932808

\* Teléfono local

12121212121

\* Teléfono móvil

4545124512

#### Dirección

Estado

Estado de México

\* Municipio

Lerma

\* Colonia

Centro

\* Calle

Av. 2 de Marzo

\* Número exterior

1

\* Número interior

1

## Formación académica

\* Último grado obtenido

Licenciatura

Área de especialidad

Ciencias de la Compu

Año

2015

Entidad que lo otorgó

UNAM

Cursos de actualización de los últimos 5 años

Ingeniería de Software

Métricas de Software

Borrar

Agregar más

## Experiencia profesional

Entidad de adscripción

Ninguna

Experiencia en el área del curso

Ninguna

Agregar más

## Reconocimientos

Ninguno

Agregar más

Contraseña

\*\*\*\*\*

Confirmar contraseña

\*\*\*\*\*

Registrarse

Limpiar

Se da clic en "Registrar" después de llenar correctamente cada campo.

Para llenar correctamente los campos existen pistas que ayudan si existe alguna duda. Por ejemplo.

## Experiencia académica (Últimos 5 años)

Experiencia docente ?

Introduce solo un trabajo por entrada, en caso de querer agregar varios presiona el botón 'Agregar más'

Agregar más

Ahora, es turno del ponente entrar a su correo y confirmar la cuenta, pero no podrá acceder a su cuenta hasta que el administrador lo haya aceptado.

5 de Octubre del 2016 Ingresar

Secretaría de Educación Abierta y Continua Facultad de Ciencias

Te has registrado correctamente, pero no has podido iniciar sesión porque tu cuenta no ha sido aprobada por el consejo académico. x

[REGISTRAR PONENTE](#)

[CATÁLOGO DE CURSOS](#)

[¿TE INTERESA ALGÚN CURSO? REGISTRATE](#)

**Instructivo para solicitar impartir Cursos, Diplomados, Talleres o Seminarios de Educación Continua**

Estimado profesor:

Si tiene interés en impartir algún curso, Diplomado o Taller especializado con temas de su dominio, lo invitamos a proponerlo a esta Secretaría entregando la documentación necesaria de acuerdo a los siguientes formatos:

Abrimos el correo y hacemos la verificación de la cuenta

educontinua.inf Instrucciones de confirmación - Bienvenido cafetafer@ciencias.unam.mx! Para poder acceder a tu cuenta sigue el siguiente enlace para cor 14:53

Se abre el correo de confirmación para seguir el procedimiento de registrar.

Instrucciones de confirmación Recibidos x

 **educontinua.inf@ciencias.unam.mx** a través de sendgrid.net 14:53 (hace 11 minutos) ☆ ↶ ⌵

para mí ▾

Bienvenido [cafetafer@ciencias.unam.mx](mailto:cafetafer@ciencias.unam.mx)

Para poder acceder a tu cuenta sigue el siguiente enlace para confirmarla

[Confirmar cuenta](#)

## Rol de Administrador.

Entramos como administrador del sistema) y damos clic en “Ver ponentes” para ver la lista de ponentes que se han registrado.



5 de Octubre del 2016 Salir

**Secretaría de Educación Abierta y Continua** Facultad de Ciencias

Sesión iniciada.

- [VER PONENTES](#)
- [VER CURSOS](#)
- [AGREGAR ADMINISTRADOR](#)

**Instructivo para solicitar impartir Cursos, Diplomados, Talleres o Se de Educación Continua**

Estimado profesor:

Si tiene interés en impartir algún curso, Diplomado o Taller especializado con temas de su dominio, lo invitamos a proponer a la Secretaría entregando la documentación necesaria de acuerdo a los siguientes formatos:

Se selecciona un ponente y la información del ponente se despliega en una tabla.

## Aprobar Ponente

### Datos Personales

Nombre	Fernando Sánchez Montoya
Correo	cafetafer@ciencias.unam.mx
R.F.C	fasm19932808
Teléfono local	12121212121
Teléfono movil	4545124512

### Dirección

Estado	Estado de México
Municipio	Lerma
Colonia	Centro
Calle	Av. 2 de Marzo
Número exterior	1
Número interior	1

## Dirección

<b>Estado</b>	<b>Estado de México</b>
<b>Municipio</b>	<b>Lerma</b>
<b>Colonia</b>	<b>Centro</b>
<b>Calle</b>	<b>Av. 2 de Marzo</b>
<b>Número exterior</b>	<b>1</b>
<b>Número interior</b>	<b>1</b>

## Formación académica

<b>Último grado obtenido</b>	<b>Licenciatura</b>
<b>Área de especialidad</b>	<b>Ciencias de la Computación</b>
<b>Año que lo obtuvo</b>	<b>2015</b>
<b>Entidad que lo otorgó</b>	<b>UNAM</b>
<b>Actualizaciones</b>	<b>Ingeniería de Software Métricas de Software</b>

Aceptar Ponente

Exportar a PDF

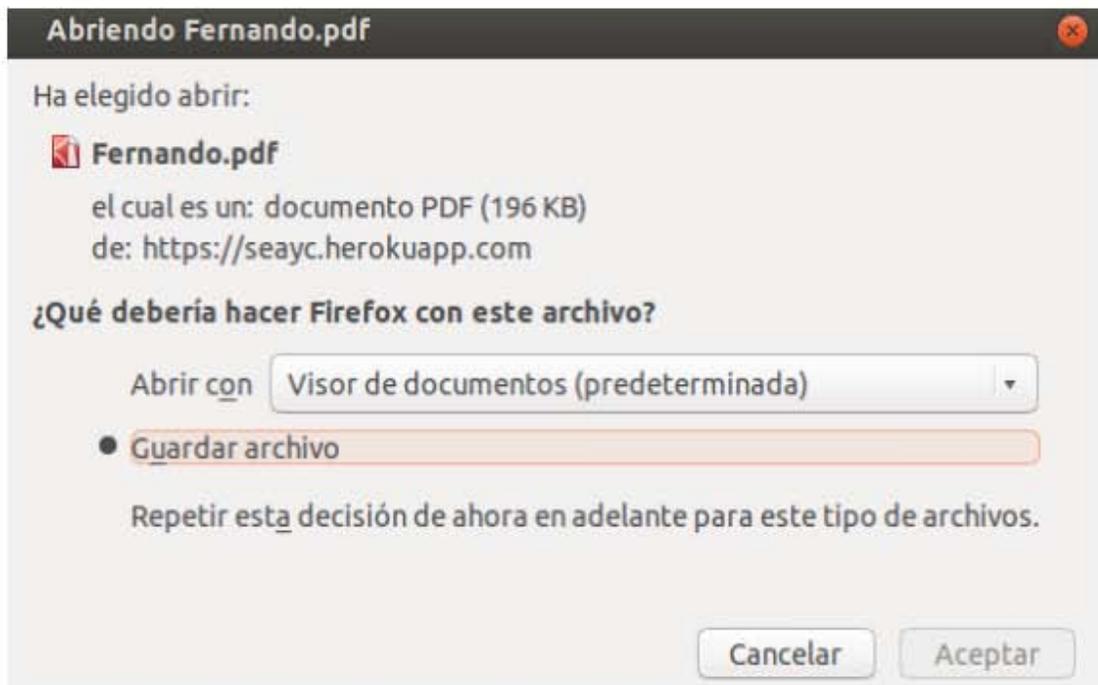
Rechazar Ponente

Tenemos tres opciones:

1. Exportar a PDF: la información del ponente se descargará en un PDF y podrá ser impreso.
2. Rechazar Ponente: no se dará de alta al ponente.
3. Aceptar Ponente: Se dará de alta al ponente que realizó el registro.

## Exportar a PDF información del Ponente

Si se selecciona la opción “Exportar a PDF” se despliega la ventana siguiente y se descarga el formato de tabla en PDF.



El formato en PDF es el siguiente:



Secretaría de Educación Abierta y Continua



Facultad de Ciencias

**CURRICULUM VITAE - SÍNTESIS**

**DATOS PERSONALES**

<b>Nombre completo</b>
Fernando Sánchez Montoya
<b>Teléfono</b>
Local 121212121 Móvil 4545124512
<b>Correo electrónico</b>
cafetafer@ciencias.unam.mx
Deseo que se publique en internet como contacto de información académica para los interesados Sí / <input type="checkbox"/> No <input checked="" type="checkbox"/>
<b>Dirección</b>
Av. 2 de Marzo 1, Centro Lerma, Estado de México
<b>RFC</b>
fasm19932808

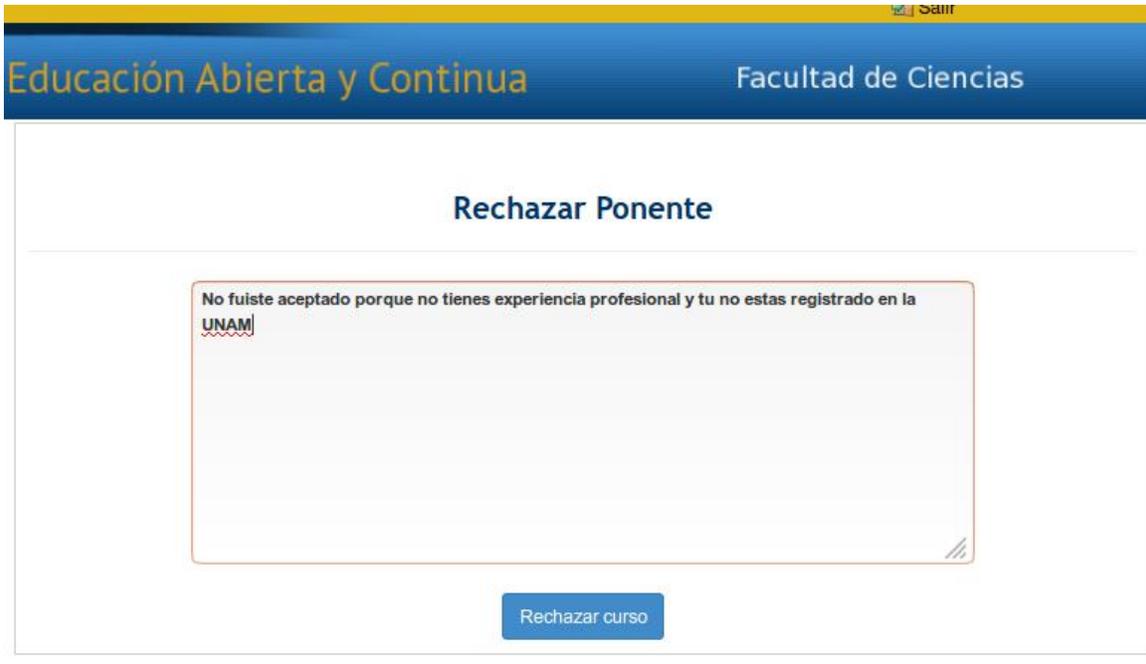
**ÁREAS DE ESPECIALIZACIÓN**

<b>Formación académica</b>
Licenciatura Ciencias de la Computación 2015 UNAM Ingeniería de Software Métricas de Software
<b>Experiencia académica</b>
Ninguna
<b>Experiencia profesional</b>
Ninguna, Ninguna
<b>Reconocimientos</b>
Ninguno

## Rechazar Ponente

Si se seleccionó la opción “Rechazar ponente”

Se tendrá que llenar un espacio con la justificación, que los administradores consideren pertinentes, de por qué se ha rechazado, al ponente en cuestión, y serán enviadas a su correo.



The screenshot shows a web interface for rejecting a speaker. At the top, there is a yellow navigation bar with a home icon, the date "5 de Octubre del 2016", and a "Salir" button. Below this is a blue header with "Educación Abierta y Continua" on the left and "Facultad de Ciencias" on the right. The main content area is titled "Rechazar Ponente" and contains a large text input field with a light blue border. The text inside the field reads: "No fuiste aceptado porque no tienes experiencia profesional y tu no estas registrado en la UNAM". Below the text field is a blue button labeled "Rechazar curso".

Se emite un mensaje exitoso de que, el correo, ha sido enviado a la dirección del ponente rechazado.



The screenshot shows the same web interface after the rejection. The yellow navigation bar now shows the date "5 de Octubre del 2016" and the "Salir" button. The blue header now displays "Secretaría de Educación Abierta y Continua" on the left and "Facultad de Ciencias" on the right. A green success message box is visible, containing the text: "El ponente caletafer@ciencias.unam.mx ha sido rechazado". Below the message box is a blue button labeled "VER PONENTES".

En el correo del ponente, estará en la bandeja el correo con las razones por las cuales ha sido rechazado.

educontinua.inf Tu cuenta no fue aprobada - Tu cuenta no pudo ser activada por la siguiente razón: No fuiste aceptado porque no tienes experiencia profesio 15:41

Si el ponente abre el correo de la SEAYC, se encontrará con el siguiente texto (ejemplo):

Tu cuenta no fue aprobada Recibidos x

educontinua.inf@ciencias.unam.mx a través de sendgrid.net 15:41 (hace 0 minutos)

para mí

Tu cuenta no pudo ser activada por la siguiente razón:

No fuiste aceptado porque no tienes experiencia profesional y tu no estas registrado en la UNAM

[SEAYC](#)

Si el ponente es aceptado se da un aviso y al igual, si es rechazado, le llegará un correo a su dirección electrónica.

El ponente cafelafer@ciencias.unam.mx ha sido aceptado

[VER PONENTES](#)

[VER CURSOS](#)

[AGREGAR ADMINISTRADOR](#)

**Instructivo para solicitar impartir Cursos, Diplomados, Talleres o Seminarios de Educación Continua**

Estimado profesor:

Con un texto (ejemplo) de alta exitosa.

Cuenta activada Recibidos x

educontinua.inf@ciencias.unam.mx a través de sendgrid.net 15:46 (hace 0 minutos)

para mí

**Fernando Sánchez,**

Tu curriculum ha sido verificado por el consejo académico y ha activado tu cuenta para que puedas crear cursos.

[SEAYC](#)

## Registrar curso

Si el Ponente fue aceptado exitosamente puede agregar un curso para impartir, dándolo de alta en la SEAYC.

### Registro de nuevo curso

#### Información para la operación y logística

**Tipo de actividad académica**

**Modalidad**

**Días de clase**

- Lunes
- Martes
- Miércoles
- Jueves
- Viernes**
- Sábado**
- Domingo

Para seleccionar más presiona Ctrl y da click en otra opción.

**Hora inicio**

**Hora fin**

**Fecha de inicio**

**Fecha final**

**Lugar**

**Horas por sesión**

**Número de sesiones**

**Pago por hora**

**Cupo mínimo**

**Cupo máximo**

**Costo público en general**

**Costo alumnos, tesis y personal de la UNAM**

## Información académica

### Descripción del curso

#### Antecedentes

Breve curso de métricas de software

#### Justificación

todo

### Objetivos del curso

#### Objetivo general

Ser un master en métricas

#### Objetivo específico

Ser master

#### Campo conocimiento

Software

#### Dirigido a

Estudiantes

#### Metas

10km

#### Procesos de evaluación

Exámenes cada sábado

#### Requisitos de ingreso

ir en penúltimo semestre

#### Requisitos de permanencia

ser constante

#### Requisitos de egreso

8 de promedio

## Contenido y calendarización

Tema	Horas	Ponentes	Fechas
<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="2"/>

Subtemas

[Borrar](#)

Tema	Horas	Ponentes	Fechas
<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="2"/>

Subtemas

[Borrar](#)

[Agregar Tema](#)

Material de apoyo requerido durante las clases	Equipo de cómputo y software requerido
<input type="text" value="Pizarrón, proyector y luces"/>	<input type="text" value="Laptop, cámara y acción"/>

Material requerido a los alumnos	Material de apoyo para los alumnos
<input type="text" value="Laptop"/>	<input type="text"/>

Bibliografía

[Registrar curso](#) [Limpiar](#)

Después de haber llenado los campos correctos se registra curso donde después el ponente puede ver, editar o borrar, la información de este.

[REGISTRAR CURSO](#)

[VER CURSOS](#)

**Cursos disponibles**

---

**Métricas de Software Express** [Editar curso](#) [Borrar curso](#)

Después se esperará la notificación de que el administrador ha aceptado el curso. Aquí es donde el administrador juega de nuevo.

El administrador da clic en **VER CURSOS**

**VER CURSOS**

Y se despliega la lista de cursos por aprobar y aprobados

**VER PONENTES**

**VER CURSOS**

**AGREGAR ADMINISTRADOR**

	<b>Cursos sin aprobar</b> ⓘ
<b>1</b> Primer Curso	
<b>2</b> Métricas de Software Express	
	<b>Cursos aprobados</b> ⓘ
No hay cursos aprobados.	

Selecciona cualquiera que tenga que aprobar y se despliega la siguiente información

## Información del curso

### Métricas de Software Express

Fecha Inicio	28/10/2016
Fecha Final	17/12/2016
Horario	Vi,Sa de 5:00 PM a 6:30 PM
Número de sesiones	12
Número de horas por sesión	2
Tipo de curso	3
Dirigido a	Estudiantes
Modalidad de curso	2
Costo al público en general	\$10000.0
Costo alumnos, tesistas y personal de la UNAM	\$5000.0
Evaluación	Exámenes cada sábado
Requisitos de ingreso	Ir en penúltimo semestre
Requisitos de permanencia	ser constante
Requisitos de egreso	8 de promedio
Mínimo de alumnos requeridos	1
Máximo número de alumnos permitidos	10
Bibliografía	Como 5 libros
Campo de conocimiento	Software

Pago	1500.0
------	--------

## Profesor

El curso será impartido por el profesor [Fernando Sánchez Montoya](#).

## Objetivo general, objetivo específico y metas

**Ser un master en métricas**

**Ser master**

**10km**

## Antecedentes y justificación

**Breve curso de métricas de software**

**todo**

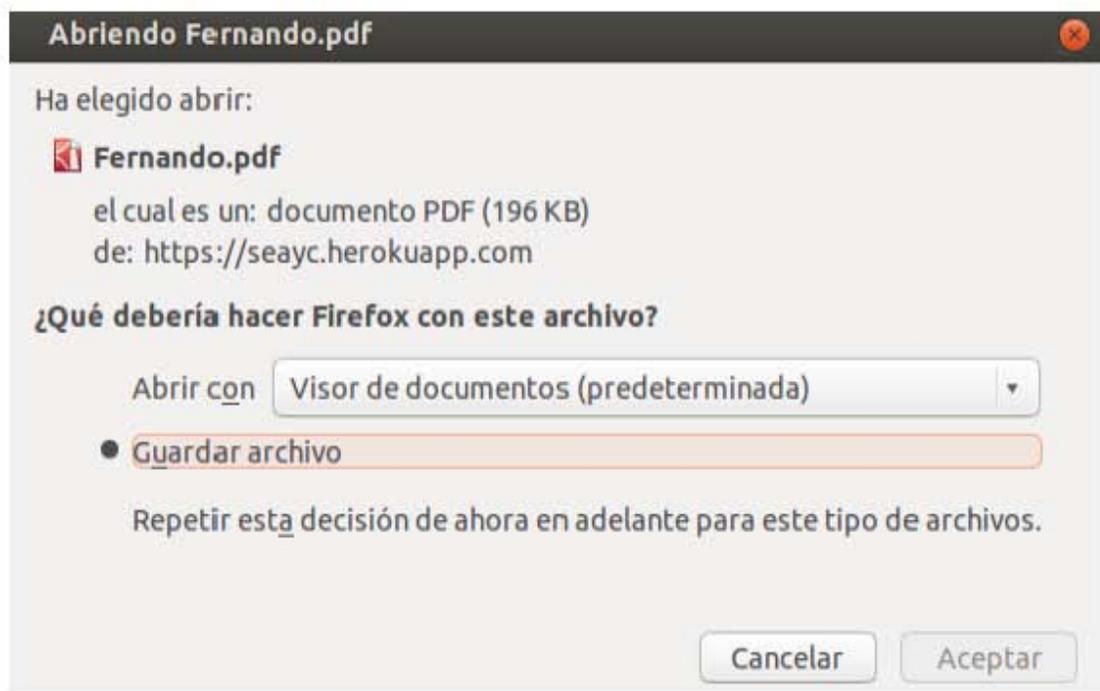
[Aprobar curso](#) [Editar curso](#) [Exportar a PDF](#) [Rechazar curso](#)

Hay 4 opciones que el administrador puede realizar con la propuesta de curso

1. Exportar a PDF: Se exportará la información en PDF y se descargará en la computadora.
2. Editar Curso: El administrador puede editar información del curso, si hay errores en existencia.
3. Rechazar Curso: Rechazar el curso y dar justificaciones que legaran por correo al ponente.
4. Aprobar Curso: se dará de alta el curso.

## Exportar a PDF información del Curso

Si se selecciona la opción exportar a PDF aparece la siguiente ventana:



Después, si se dio en guardar o abrir, se despliega la información del curso en un formato PDF:



Secretaría de Educación Abierta y Continua



Facultad de Ciencias

## Métricas de Software Express

### 1. INFORMACIÓN PARA LA OPERACIÓN Y LOGÍSTICA

#### DATOS GENERALES

<b>Tipo de actividad académica</b>	Diplomado
<b>Modalidad</b>	Semi-presencial
<b>Fecha de inicio</b>	28/10/2016
<b>Fecha de término</b>	17/12/2016
<b>Días y horarios</b>	Vi,Sa de 5:00 PM a 6:30 PM
<b>Lugar</b>	Facultad de Ciencias
<b>Número de sesiones</b>	12
<b>Número de horas por sesión</b>	2
<b>Número total de horas</b>	24
<b>Máximo de inscritos</b>	1 - 10

#### REQUERIMIENTOS

<b>Pago propuesto a docentes (Horas)</b>	\$ 1500.0
<b>Material de apoyo requerido durante las clases</b>	Pizarrón, proyector y luces

<b>Equipo de cómputo y software requerido</b>
Laptop, cámara y acción
<b>Material requerido a los alumnos</b>
Laptop
<b>Material de apoyo para los alumnos</b>

#### COSTO DE INSCRIPCIÓN PROPUESTO

<b>Para público en general</b>
\$ 10000.0
<b>Para alumnos, tesis y personal de la UNAM</b>
\$ 5000.0

#### PUBLICIDAD

#### IL INFORMACIÓN ACADÉMICA

### Descripción del curso

<b>Antecedentes y justificación</b>
Breve curso de métricas de software todo

### Objetivos

<b>Objetivo general</b>
Ser un master en métricas
<b>Objetivo específico</b>
Ser master
<b>Metas</b>
10km
<b>Campo de conocimiento</b>
Software
<b>Dirigido a</b>
Estudiantes

**Docente y entidades participantes**  
**Requisitos académicos para alumnos**

**Requisitos de ingreso**

ir en penúltimo semestre

**Requisitos de permanencia**

ser constante

**Requisitos de egreso**

8 de promedio

**Contenido y calendarización**

TEMA	No de horas	Ponente(s)	Fechas	Subtemas
1	1	1	2	4
2	1	1	2	5

**Bibliografía**

Como 5 libros

## Editar Curso

Si se desea hacer una edición en la información, antes proporcionada, del curso.

Ejemplo: Para hacer una prueba se cambia el precio

<b>Cupo mínimo</b>	<b>Cupo máximo</b>	<b>Costo público en general</b>	<b>Costo alumnos, tesistas y personal de la UNAM</b>
1	10	15000	8000

se da clic en “Actualizar Curso”

**Actualizar curso** **Limpiar**

y cambia en la información desplegada, al volverse a verificar.

<b>Costo al público en general</b>	<b>\$15000.0</b>
<b>Costo alumnos, tesistas y personal de la UNAM</b>	<b>\$8000.0</b>

## Rechazar Curso

Si la opción seleccionada es “Rechazar curso”. Se muestra un mensaje exitoso de que el curso ha sido rechazado

El curso Métricas de Software Express ha sido rechazado ✕

[VER PONENTES](#)

[VER CURSOS](#)

[AGREGAR ADMINISTRADOR](#)

**Cursos sin aprobar** ?

1 Primer Curso

**Cursos aprobados** ?

No hay cursos aprobados.

Notifica al Ponente que su curso ha sido rechazado y los motivos de este rechazo.

educontinua.inf **Curso rechazado** - Fernando Sánchez, Lo sentimos el curso @curso.nombre fue rechazado debido a que no cumpla con los requerimientos a **16:29**

Al abrir el correo se muestra el texto(ejemplo) de la justificación del rechazo.

Curso rechazado Recibidos x 🖨 🗑

 **educontinua.inf@ciencias.unam.mx** a través de sendgrid.net 16:29 (hace 4 minutos) ☆ ↶ ⌵

para mí ⌵

**Fernando Sánchez,**

Lo sentimos el curso @curso.nombre fue rechazado debido a que no cumpla con los requerimientos adecuados.

[SEAYC](#)

## Aprobar Curso

Se aprueba el curso y se muestra un mensaje satisfactorio y pasa a la sección de cursos aprobados.

El curso Primer Curso ha sido aceptado ✕

- [VER PONENTES](#)
- [VER CURSOS](#)
- [AGREGAR ADMINISTRADOR](#)

**Cursos sin aprobar** ⓘ

No se han registrado nuevos cursos.

**Cursos aprobados** ⓘ

1 Primer Curso Historial

### 3. CATÁLOGO DE CURSOS

[REGISTRAR PONENTE](#)

[CATÁLOGO DE CURSOS](#)

[¿TE INTERESA ALGÚN CURSO?  
REGISTRATE](#)

Muestra los cursos disponibles dada la búsqueda.

Secretaría de Educación Abierta y Continua Facultad de Ciencias

---

[REGISTRAR PONENTE](#)  
[CATÁLOGO DE CURSOS](#)  
[¿TE INTERESA ALGÚN CURSO?  
REGISTRATE](#)

## Cursos disponibles

[Mostrar todos los cursos](#)

O el total de todos los cursos existentes.

[REGISTRAR PONENTE](#)  
[CATÁLOGO DE CURSOS](#)  
[¿TE INTERESA ALGÚN CURSO?  
REGISTRATE](#)

## Cursos disponibles

[Mostrar todos los cursos](#)

[Primer Curso](#)

[REGISTRAR PONENTE](#)  
[CATÁLOGO DE CURSOS](#)  
[¿TE INTERESA ALGÚN CURSO?  
REGISTRATE](#)

## Cursos disponibles

[Mostrar todos los cursos](#)

[Primer Curso](#)

Si se desea ver la información del curso se da clic sobre el nombre de este.

## Primer curso

Fecha Inicio	22/09/2016
Fecha Final	03/10/2016
Horario	Martes,Jueves,Sábado de 10:00 AM a 4:00 PM
Número de sesiones	7
Número de horas por sesión	4
Tipo de curso	1
Dirigido a	Publico en general
Modalidad de curso	1
Costo al público en general	\$11.0
Costo alumnos, tesistas y personal de la UNAM	\$5.0
Evaluación	Ninguno
Requisitos de ingreso	Ninguno
Requisitos de permanencia	Ninguno
Requisitos de egreso	Ninguno
Bibliografía	Ningún libro

## Profesor

El curso será impartido por el profesor Jorge Ascencio Espindola.

## Objetivo general, objetivo específico y metas

Ninguno

Ninguno

Ninguna

Me interesa el curso

Nueva búsqueda

Regresando al rol de Interesado al iniciar sesión podemos ver que ahora ya hay cursos disponibles y podemos interesarnos en alguno.

### Iniciando sesión

## Acceder

Correo electrónico

fasm19932808@hotmail.com

Contraseña

••••••••

[¿Olvidaste tu contraseña?](#)

Acceder

Ingresamos en Catálogo de Cursos

Secretaría de Educación Abierta y Continua Facultad de Ciencias

[REGISTRAR PONENTE](#)

[CATÁLOGO DE CURSOS](#)

[¿TE INTERESA ALGÚN CURSO? REGISTRATE](#)

### Cursos disponibles

Introduce el nombre del curso que buscas

[Mostrar todos los cursos](#)

Realizamos una búsqueda y seleccionamos algún curso que sea de nuestro interés y damos clic en “Me interesa el curso”

Este aparecerá en “Mis Cursos”

**CATÁLOGO DE CURSOS**

---

**MIS CURSOS**

---

**Mis cursos**

---

Primer Curso

Cuando algún curso ha sido de interés del al menos el cupo mínimo llega notificación por correo al Ponente para decirle que ya tiene el cupo mínimo de interesados y que puede proceder a impartirlo.

El Administrador puede observar el número de interesados para tener en cuenta que se puede impartir el curso o no.

## Antecedentes y justificación

Todos

Porque si

## Reporte de interesados

# Interesado	Nombre del interesado	Fecha que se apunto
1	Fernando Sánchez Montoya	05/10/2016

## 7. Conclusiones.

El objetivo de este manual fue presentar la metodología de trabajo que debe seguir un alumno de Ingeniería de Software II durante el curso.

El proyecto, llevado a cabo en la materia de Ingeniería de Software II, para ser desarrollado durante el semestre fue siguiendo las metodologías SCRUM y Kanban. Los productos que contiene este manual, de ejemplo, podrían ser usados por cualquier persona que use una metodología ágil para el desarrollo de software y no solo para los alumnos.

Los ejemplos de productos que se incluyen en este manual fueron resultado de un proyecto que se realizó llevando la materia de Ingeniería de Software II, por lo tanto, son adecuados y brindan la ayuda necesaria que requiere el alumno, antes de iniciar con el desarrollo del proyecto, durante el desarrollo de las iteraciones y al termino del proyecto, en alguna situación de problema que pueda llegar a experimentar.

Además, da la debida importancia a la metodología ágil, a Trello y a la creación de la documentación pertinente de un sistema de tal magnitud, es decir, los productos de trabajo documentales pueden servir a cualquier profesional en su ámbito académico o laboral, pues sirven de guía de buenas prácticas para la documentación de un proyecto web.

En conjunto con el manual para la realización de un sistema web en Ruby on Rails, sirven para apoyar al alumno en el desarrollo práctico y documental que se requiere al cursar la materia de Ingeniería de Software II y en futuras experiencias profesionales.

Un objetivo más de esta materia es: Que los conocimientos del alumno en otras áreas de las Ciencias de la Computación contribuyan en el uso de la ingeniería de software, permitiendo que el alumno desarrolle y aproveche sus multicapacidades y considere, la ingeniería de software a largo o corto plazo, como una fuente de empleo y todo esto se puede facilitar haciendo uso de las herramientas mencionadas y de los ejemplos de productos de este manual.

## 8. Bibliografía

Carmen Lasa Gómez, A. Á. (2017). *Métodos Ágiles Scrum, Kanban, Lean*. Madrid: Anaya.

Fog Creek Software. (2016). *Trello*. Obtenido de <https://trello.com/>

Guadalupe Ibargüengoitia, H. O. (2016). *Primer Curso de Ingeniería de Software*. Ciudad de México.

Lifecycle profiles for Very Small Entities (VSEs). (2011). *Estándar Internacional ISO/IEC 29110 Software engineering* -.

Palacios, J. (2018). *Jerónimo Palacios*. Obtenido de <https://jeronimopalacios.com/scrum/>

Pressman., R. S. (2010). *Ingeniería de Software. Un enfoque Práctico*. México. D.F.: Mc Graw Hill.

SCRUMstudy™, a brand of VMEdU, Inc. (2016). *A Guide to the SCRUM BODY OF KNOWLEDGE (SBOOK™GUIDE)*. Phoenix, Arizona 85008 USA: Library of Congress Cataloging-in-Publication Data.

T-Jones Technologies. (2016). *SEAYC*. Obtenido de <https://seayc.herokuapp.com/>

## 9.ANEXO A. DOCUMENTOS INICIALES.

**A1. Requerimientos generales.**

**A2. Plan del proyecto.**

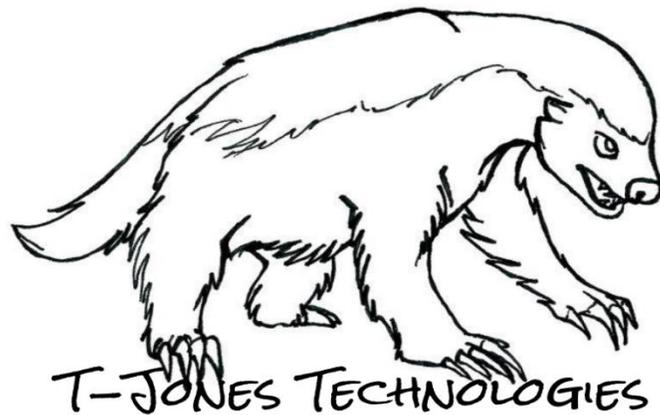
**A3. Product Backlog inicial.**

**A4. Arquitectura.**

Anexo A. A1. Requerimientos generales.

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Requerimientos Generales**

**Pedrito**

**Sotanito**

**Luisita**

# Descripción General.

## Objetivo

Se requiere un sistema web para poder registrar las propuestas de cursos con su coordinador y los ponentes; revisar en términos académicos las propuestas de los cursos; registrar a los interesados; llevar el número de interesados para cada uno de los cursos del catálogo; clasificar los cursos en el catálogo y que se actualice periódicamente; realizar un reporte de interesados a cada curso y enviar una notificación a los interesados para invitarlos a inscribirse a los cursos y por último enviar notificaciones los ponentes y a la Secretaría de los cursos que están prontos a abrirse.

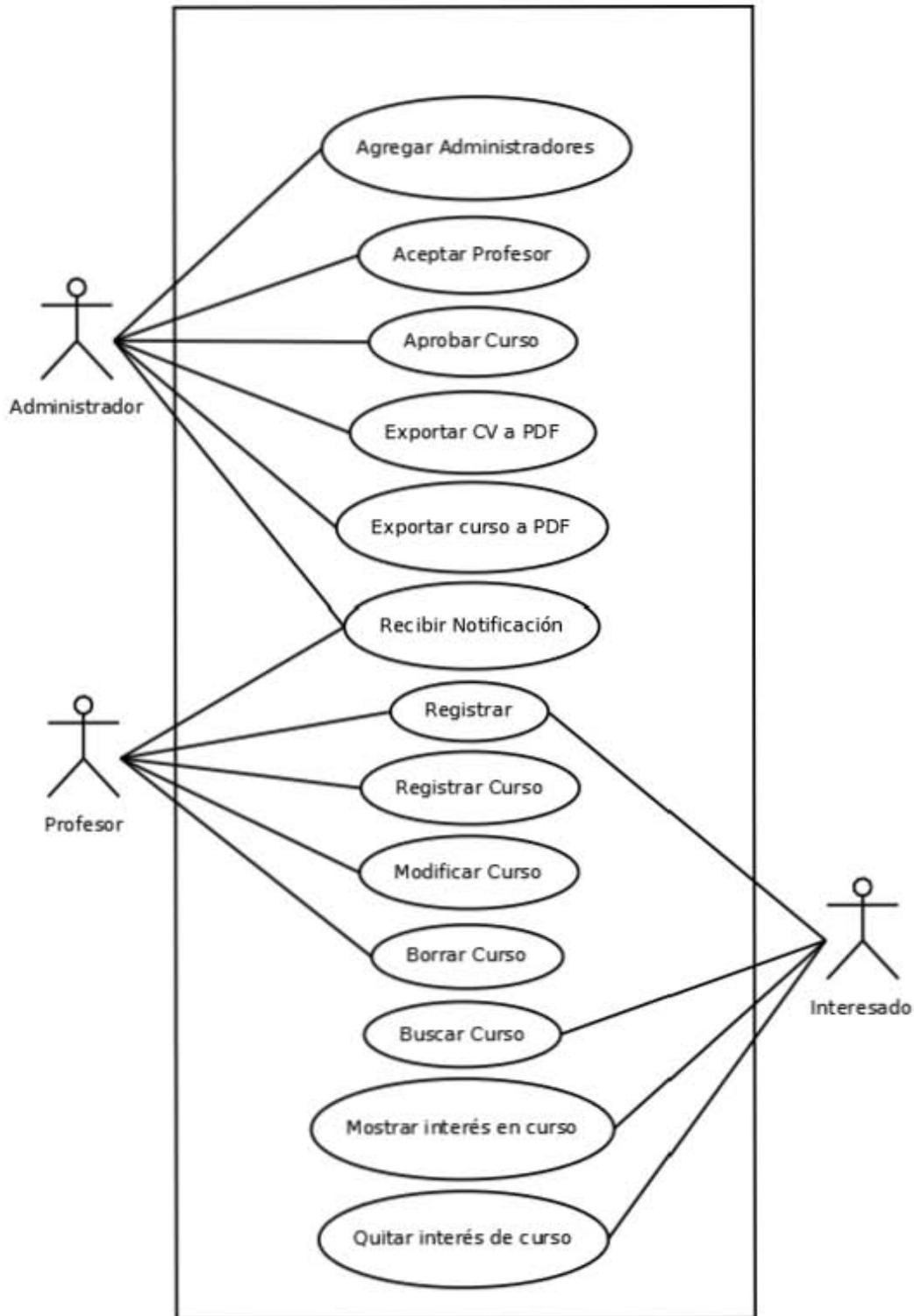
Rediseñar el sistema de la Secretaría de Educación a Distancia Abierta y Continua de la Facultad de Ciencias principalmente para agilizar la promoción de cursos y diplomados, también agilizar los procesos de inscripción e implementar acciones que sustituyan parcialmente el “papeleo” físico.

## Glosario de términos

Administrador:	Actor que tiene permisos para aprobar, rechazar, editar y exportar a PDF en lo que refiere a cursos y ponentes
Curso:	Curso, taller o diplomado que un ponente puede proponer
CV:	Currículum Vitae
Interesado:	Actor que puede buscar cursos en el sitio y puede mostrar su interés sobre alguno.
Iteración/sprint (s):	Periodo de tiempo en el cual se establecen objetivos y se trabaja para alcanzarlos.
Metodología ágil:	Conjunto de métodos y técnicas en la ingeniería de software para agilizar los procesos y lograr los objetivos en tiempo, mejorando la calidad del producto y del trabajo en equipo.
MVC:	Modelo-Vista-Controlador
Ponente:	Actor que desea impartir un curso, taller o diplomado.
Product Backlog:	Documento en el cual se describen las tareas y sus prioridades de acuerdo a las necesidades del sistema
Propuesta:	Propuesta de Curso o Propuesta de Ponente que espera ser aceptada o rechazada por el Administrador
Requerimientos:	Funcionalidades características que se desean implementar en un proyecto de software. Generalmente descritas como tareas que el sistema va a realizar
SCRUM Master:	Líder y responsable de facilitar al equipo SCRUM el ambiente propicio para concretar un proyecto exitosamente
SCRUM:	Metodología Ágil
SEAYC:	Secretaría de Educación Abierta y Continua de la Facultad de Ciencias
Stakeholders:	Conjunto de personas interesadas, que básicamente son los clientes, en la realización del proyecto de software
Trello:	Gestor de tareas en línea

# Requisitos funcionales

Identificación de los casos de uso



## Ponente

- **Registrar propuesta de curso.** El ponente manda una descripción de curso que desea impartir llenando un formulario.
- **Registrar CV y datos académicos.** El ponente ingresa información correspondiente a su currículum vitae y datos académicos.
- **Modificar la propuesta de curso e información.** Después de que el Administrador realiza las observaciones de la propuesta, el ponente debe realizar las correcciones del documento que mandó.
- **Borrar Curso.** El ponente puede eliminar algún curso que el haya propuesto.
- **Recibir Notificación.** Cuando los cursos tienen un número considerable de interesados, el ponente recibe una alerta de que su curso cumple con el mínimo de interesados necesarios para su realización.

## Administrador (SEAYC)

- **Aceptar ponentes.** Si la solicitud de algún ponente cumple con los requisitos el administrador puede aceptar al ponente, si no es el caso puede rechazarlo.
- **Aprobar curso.** Si el curso registrado por algún ponente está en las condiciones y cumple con los requisitos, el administrador puede aprobar el curso, si no es el caso puede rechazarlo.
- **Exportar información del ponente a PDF.** Se exportará la información de un ponente a formato PDF, con fines de la SEAYC.
- **Exportar información del curso a PDF.** Se exportará la información de un curso a formato PDF, con fines de la SEAYC.
- **Agregar Administradores.** El "Súper" Administrador tiene la facultad de agregar más administradores que puedan administrar el sistema de la SEAYC, así como eliminarlos.
- **Recibir notificación.** Cuando un curso tiene un número importante de interesados, se recibirá una alerta.

## Interesado

- **Registrar información de contacto.** A Través de un formato de captura de datos el Interesado se registra en la página de SEAYC, con datos para ser fácilmente contactado.
- **Interesarse/desinteresarse en un curso.** Si el interesado se interesa en un curso específico puede declararlo o si algún curso deja de ser de su interés también puede hacerlo.
- **Buscar Curso.** El interesado puede buscar algún curso que sea de su interés.

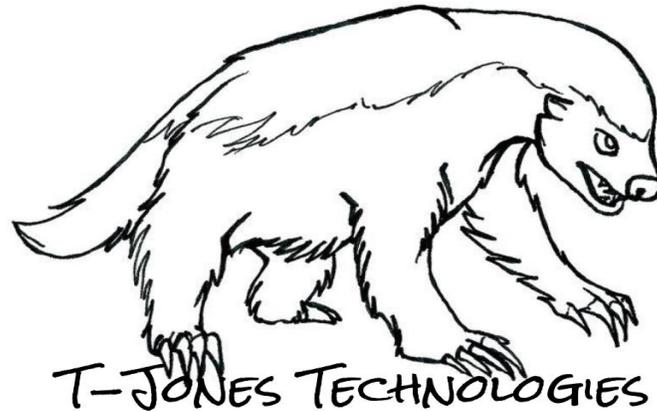
## Requisitos no funcionales

- **Confiabilidad:** Asegurar la veracidad de la información de los **Cursos** que se impartirán y los datos de los **Ponentes**. La página del sitio del sistema tendrá las especificaciones que deben seguir las páginas que pertenecen a la UNAM (Aviso de privacidad).
- **Eficiencia:** La respuesta del sistema corresponde al comportamiento deseado por el usuario del sistema.
- **Mantenimiento:** Se estableció un acuerdo de “garantía” con los “Clientes Reales” de dar mantenimiento al sistema por el siguiente medio año después de su instalación.
- **Portabilidad:** Al ser un sistema web el paquete de documentación, así como el de código se entregará y se instalará en el servidor determinado por la SEAYC por lo tanto se considera un sistema portable.
- **Restricciones de diseño y construcción:** Este proyecto a desarrollar se limita durante el proceso de desarrollo al ambiente de implementación expuesto en el documento de Arquitectura y Diseño. Si el cliente desea, después algunos cambios en los cuales se necesiten diferentes ambientes, se generará la documentación correspondiente como parte de otro proyecto.
- **Reusabilidad:** Ninguna herramienta se comprará para el desarrollo de este proyecto, todas son herramientas libres de licencia.
- **Interfaces:**
  1. **Interfaces de Usuario:** Interfaz de usuario será desarrollada en Ruby on Rails.
  2. **Interfaces de Hardware:** No se ocupará ninguna interfaz de hardware en especial.
  3. **Interfaces de Software:** El sistema en sí es la única interfaz de software, ya que no necesita de aplicaciones u otros sistemas fuera de este.
  4. **Interfaces de Comunicación:** No se necesita de interfaz de comunicación en especial para el sistema ya que es un sistema web.

Anexo A. A2. Plan del Proyecto.

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Plan del Proyecto**

**Pedrito**

**Sotanito**

**Luisita**

# Plan del Proyecto

La Secretaría de Educación Abierta y Continua (SEAYC) de la Facultad de Ciencias tiene la función de promover y organizar cursos, diplomados, talleres, conferencias y seminarios afines a las áreas del conocimiento de la Facultad de Ciencias. Dicha secretaria desea dar impulso a los cursos de educación continua que profesores de la Facultad o profesionales con experiencia probada proponen para el público en general. Los cursos y los ponentes son sometidos a revisión por un Comité Académico para realizar observaciones y autorizar o no el curso o al ponente.

Actualmente, las propuestas se registran mediante un formato en Word (descargable) donde se describe el curso, y otro formato para el currículum del coordinador del curso y de los ponentes. Después, la SEAYC pasa las propuestas al Comité Académico para realizar observaciones que deberán ser atendidas por aquellos que proponen los cursos, además se lleva un historial de modificaciones de las propuestas.

Tanto si las propuestas como los ponentes son aceptados por el Comité Académico, entonces la SEAYC solicita la autorización del Consejo Técnico de la Facultad para enviarla (manualmente) al departamento de contabilidad, de manera que se puedan recibir los pagos de los inscritos.

Por otro lado, se quiere tener una base de datos de los interesados a los cursos mediante un registro a través de la página de la SEAYC. Donde se lleva el registro del número de interesados para cada uno de los cursos que se ofrecen, así cuando un curso ya tiene un número importante de interesados, la encargada de los cursos envía de manera manual, al coordinador del área el reporte de interesados para cada curso registrado. Con este reporte, se toman las decisiones sobre qué cursos se deben organizar, comunicar a los ponentes para definir fechas y enviar una notificación a los interesados para invitarlos a inscribirse a los cursos.

El catálogo de cursos y diplomados, no se encuentra clasificado por área de conocimiento para difundir la oferta académica de la Facultad de Ciencias. El catálogo de cursos propone a aquellos que están interesados en algún curso a registrarse en la página de la SEAYC.

El problema con este sistema es que no se cuentan con alertas que indiquen al coordinador del área si hay un curso que tenga ya el número de interesados el cual valga la pena tomar en cuenta para proceder a organizarlo y eventualmente ponerlo en marcha. El proceso se tiene que hacer a petición del coordinador y la revisión del número de interesados, para cada curso, se debe hacer de manera manual.

Por esto la SEAYC necesita un Sistema apoyo a las funciones de registro de interesados, de ponentes, de cursos y que proporcione una alerta global de notificaciones a los interesados y a la secretaria para alertar al coordinador del área cuando los cursos ya tienen suficientes interesados para iniciar su organización y posteriormente su promoción.

## Objetivos

Se requiere un sistema web para poder registrar las propuestas de cursos con su coordinador y los ponentes; revisar en términos académicos las propuestas de los cursos; registrar a los interesados; llevar el número de interesados para cada uno de los cursos del catálogo; clasificar los cursos en el catálogo y que se actualice periódicamente; realizar un reporte de interesados a cada curso y enviar una notificación a los interesados para invitarlos a inscribirse a los cursos y por último enviar notificaciones los ponentes y a la Secretaría de los cursos que están prontos a abrirse.

Rediseñar el sistema de la Secretaría de Educación a Distancia Abierta y Continua de la Facultad de Ciencias principalmente para agilizar la promoción de cursos y diplomados, también agilizar los procesos de inscripción e implementar acciones que sustituyan parcialmente el “papeleo” físico.

## Alcance

Rediseñar el sistema de la Secretaría de Educación a Distancia Abierta y Continua de la Facultad de Ciencias principalmente para agilizar la promoción de cursos y diplomados, también agilizar los procesos de inscripción e implementar acciones que sustituyan parcialmente el “papeleo” físico.

## Entregables

- Código del sistema
- Guía rápida de Instalación

# Recursos Necesarios

## Humanos:

- Pedrito
- Sotanita
- Luisita

## Materiales:

- Taller de Ingeniería de Software
- Teléfonos inteligentes (smartphones)

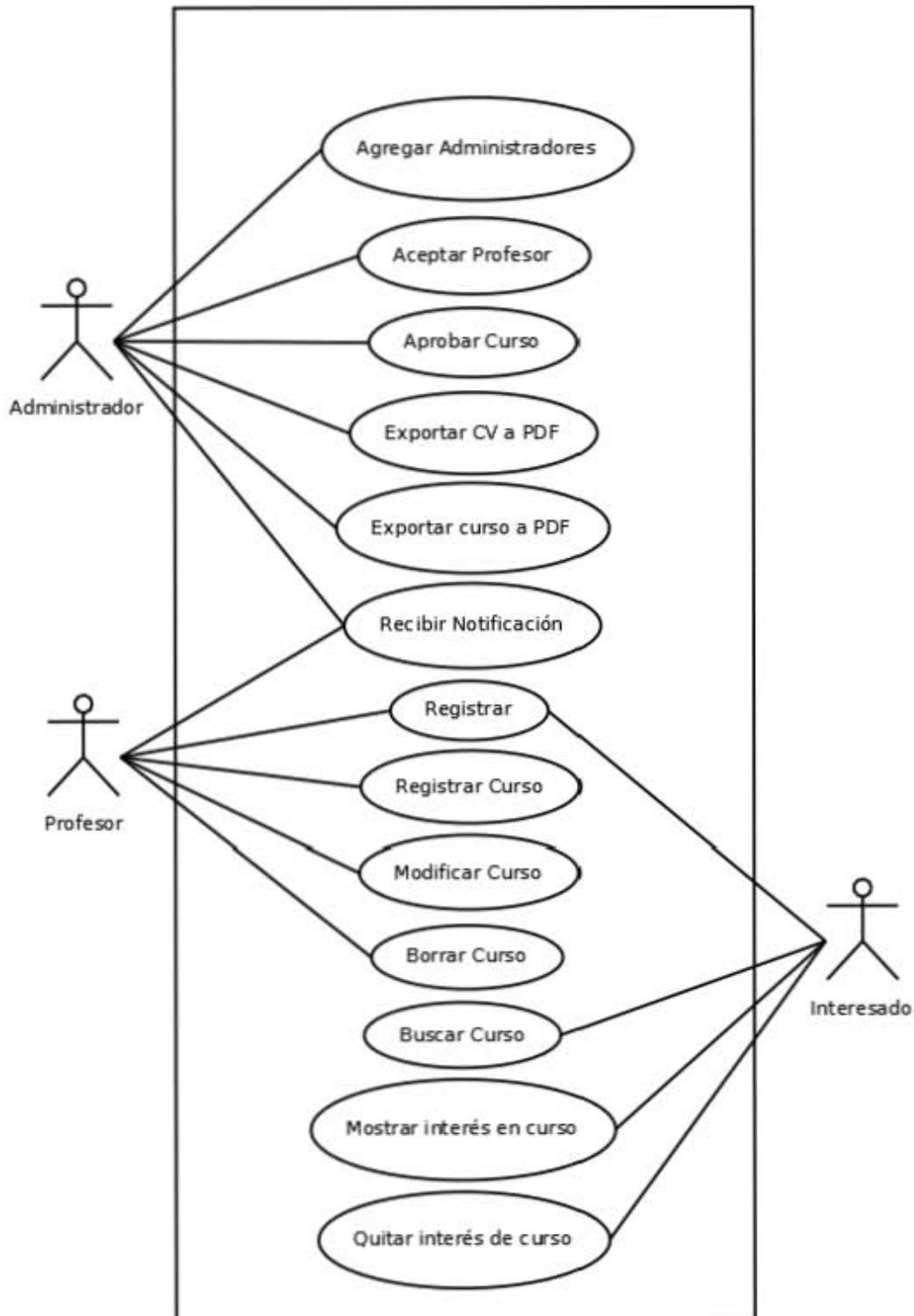
## Equipos:

- Computadoras portátiles
- Computadoras de escritorio

## Herramientas:

- Sublime Text
- Trello
- Drive
- Correo electrónico
- Hangouts
- Git
- Ruby on rails
- Dia
- PostgreSQL
- Heroku

## Diagrama General de Casos de Uso



## Definición de funcionalidades.

Id	Funcionalidad o Caso de Uso	Prioridad	Responsables	Estimación	Iteración
01	Registrar CV de los ponentes	Alta	Pedrito	6hrs.	1
02	Modificar CV de los ponentes	Media	Luisita	2hrs.	1
03	Registrar propuesta de curso	Alta	Luisita	6hrs.	1
04	Modificar propuesta de curso	Media	Pedrito	2hrs.	1
05	Borrar curso	Media	Pedrito	4hrs.	1
06	Aprobar a los ponentes	Baja	Sotanito	1hrs.	1
07	Agregar Administradores	Baja	Sotanito	4hrs.	2
08	Autorizar Curso	Alta	Luisita	4hrs.	2
09	Buscar curso	Media	Sotanito	4hrs..	2
10	Registrar en la página de SEAYC	Media	Pedrito	5hrs.	3
11	Interesarse en un curso	Baja	Sotanito	4hrs.	3
12	Enviar notificación	Media	Pedrito	2hrs.	5
13	Exportar a formato PDF la información de los ponentes	Alta	Luisita	6hrs.	5
14	Exportar a formato PDF la información de los cursos	Alta	Luisita	2hrs.	5

## Identificación de riesgos

- Fallas de red.
- Fallas eléctricas.
- Falta de asistencia por cuestiones de trabajo.
- Olvidar laptop en casa.
- Olvidar contraseñas (trello, gmail, etc)
- Falta de comunicación.
- Enfermarse

## Plan de riesgos

- Buscar formas de comunicación cuando haya imprevistos
- Ir a un café Internet
- Avisar con tiempo si no es posible asistir
- Tener un respaldo de todo lo posible en la nube
- Escribir las contraseñas en papel y compartir si es necesario

## **Estrategia de control de versiones.**

Por cada iteración que modifique el código del producto la versión se modificará en una unidad.  
Ejemplo:

Iteración 1(SEAyC v1.0) → Iteración 2 (SEAyC v2.0)

Por cada actualización a cualquier documento la versión aumentará en una centésima (aumentará del 0.01 hasta el 0.09) si rebasa 0.09, se aumentará una décima (del 0.09 al 0.10). Ejemplo:

ProductBacklog v0.01 → actualización → ProductBacklog v0.02

ProductBacklog v0.09 → actualización → ProductBacklog v0.10 → actualización → ProductBacklog v0.11.

## **Instrucciones de entrega al cliente de cada entregable**

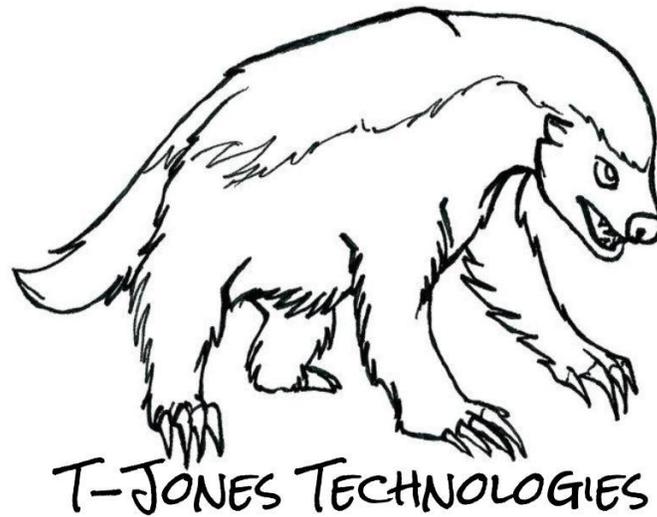
Se entregará un paquete que contiene el código dispuesto para ser instalado en el servidor correspondiente de la SEAyC, en conjunto de un manual con las instrucciones y versiones adecuadas para una instalación satisfactoria y funcional del producto web. (Véase manual de prácticas)

El paquete también contendrá una presentación con los casos de uso ejemplificado en pruebas para mostrarle el correcto uso y respuesta del sistema en cada caso al personal de la SEAyC y si es necesario se programará una sesión de capacitación para el uso del sistema web.

Anexo A. A3. Product Backlog inicial.

# **T-Jones Technologies**

*“Honey badger never gives up”*



**Product Backlog inicial**

**Pedrito**

**Sotanito**

**Luisita**

## Product Backlog

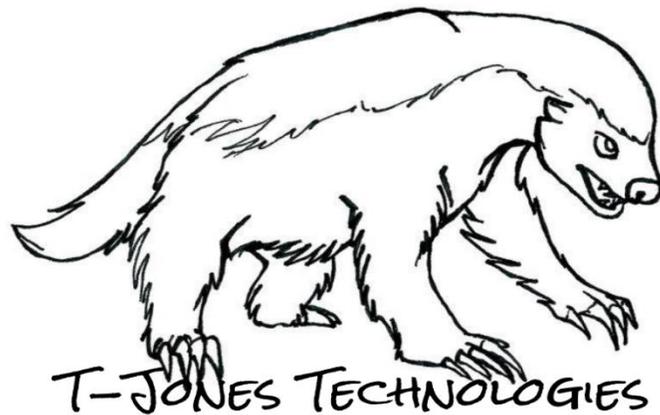
Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs	0
02	Registrar CV e información de los ponentes	Alta	6hrs.	1
03	Modificar CV e información de los ponentes	Media	2hrs.	1
04	Registrar propuesta de curso	Alta	6hrs.	1
05	Modificar propuesta de curso	Media	2hrs.	1
06	Borrar curso	Media	4hrs.	1
07	Documentación de la primera iteración	Alta	11.75 hrs.	1
08	Aprobar a los ponentes	Baja	1hrs.	2
09	Agregar Administradores	Baja	4hrs.	2
10	Autorizar Curso	Alta	4hrs.	2
11	Buscar curso	Media	4hrs..	2
12	Documentación de la segunda iteración	Alta	9.75hrs.	2
13	Registrar en la página de SEAyC	Media	5hrs.	3
14	Interesarse en un curso	Baja	4hrs.	3
15	Documentación de la tercera iteración	Alta	6.5hrs	3
16	Enviar notificación	Media	2hrs.	4

17	Documentación de la cuarta iteración	Alta	8hrs.	4
18	Exportar a formato PDF la información de los ponentes	Alta	6hrs.	5
19	Exportar a formato PDF la información de los cursos	Alta	2hrs.	5
20	Documentación de la quinta iteración	Alta	5.75hrs.	5
21	Documentación de la sexta iteración	Alta	19hrs	6

Anexo A. A4. Arquitectura.

# **T-Jones Technologies**

*“Honey badger never gives up”*



**Arquitectura**

**Pedrito**

**Sotanito**

**Luisita**

# Introducción

Es importante para todo Proyecto de Software definir el conjunto de decisiones sobre la organización del Sistema que se desea desarrollar. Se definen los Atributos de Calidad según los requerimientos que el cliente/usuario considera para evaluar el producto; la elección de un patrón o estilo para modelar la Arquitectura que contiene la composición, comportamiento e implementación de los elementos estructurales del sistema y subsistemas de forma detallada. También se especifica el ambiente de implementación donde se realizará el proyecto que mejor se adapte al modelo arquitectónico escogido y aceptado por el cliente.

Este conjunto de decisiones, pueden tener mucho peso con respecto al desarrollo y mantenimiento del sistema pues permiten administrar los cambios a medida que el sistema evoluciona.

La arquitectura que se muestra a continuación pertenece al Sistema de Apoyo al Manejo de Cursos de la Secretaría de Educación Abierta y Continua que busca promover y organizar cursos afines a las áreas del conocimiento de la Facultad de Ciencias.

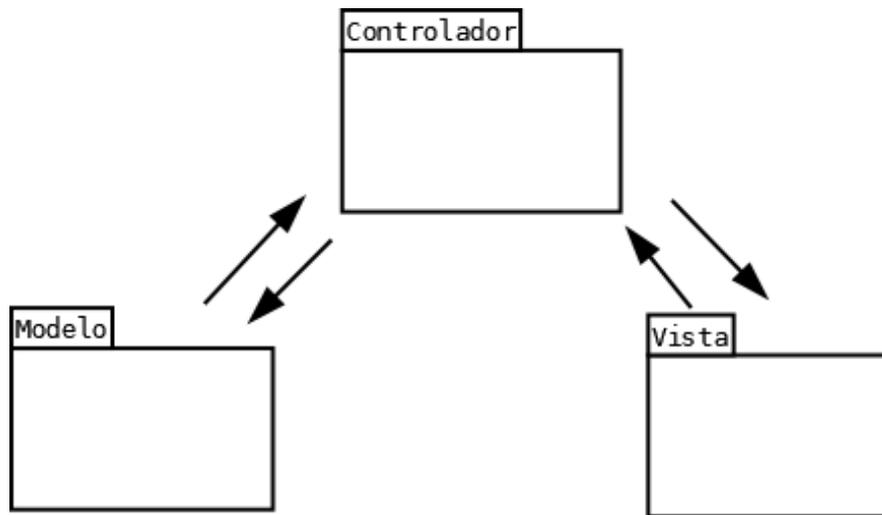
## Atributos de Calidad

Los atributos de calidad que se buscan satisfacer en cuanto a las necesidades de la Secretaría respecto al Registro de Propuestas y Ponentes son:

- **Confiabilidad:** Asegurar la veracidad de la información de los **Cursos** que se impartirán y los datos de los **Ponentes**. La página del sitio del sistema tendrá las especificaciones que deben seguir las páginas que pertenecen a la UNAM (Aviso de privacidad).
- **Entendible:** Un sistema en el cual el **Registro de Propuestas de Ponentes y Cursos**, el interesarse a un **Curso**, la revisión de **Propuestas de Ponentes y de Cursos** sea intuitivo. Es decir, un sistema en el que los datos se ingresan de manera fácil y rápida.
- **Separar Funcionalidades:** Con el modelo arquitectónico Modelo-Vista-Controlador (MVC) se garantiza la separación de funcionalidades.
- **Eficiencia:** La respuesta del sistema corresponde al comportamiento deseado por el usuario del sistema.

# Patrón Arquitectónico

El modelo arquitectónico utilizado para la creación del sistema web de la SEAYC fue el Modelo- Vista –Controlador (MVC) ya que se desea tener un sistema web robusto y este modelo es el adecuado para este tipo de sistemas que nos permite tener la menor dependencia entre cada módulo. Su estructura es la siguiente:



El MVC es un patrón arquitectónico esencialmente creado para sistemas de software robustos, como el sistema web para SEAYC, ya que existe el menor grado de dependencia entre los módulos facilitando su mantenimiento. El modelo MVC se fundamenta en la separación del código en tres capas distintas:

1. Modelo: Es la capa que trabaja con los datos con los cuales el sistema va a trabajar, ya que contiene los mecanismos para acceder a estos datos.
2. Vista: Es la capa donde se desarrolla la visualización de las interfaces adecuadas para interactuar con el sistema.
3. Controlador: Es la capa donde se encuentra la respuesta a todos los eventos que son solicitados al sistema, solicita datos a la capa Modelo y puede hacer visualizar un elemento.

# Ambiente de Implementación

En el ambiente de implementación se permite experimentar con nuevas tecnologías y herramientas, nos permitió usar y aprender Ruby on Rails. A comparación de la materia de Ingeniería de Software I, teníamos que hacer uso de un servidor que no fuera local, así llegamos a utilizar Heroku.

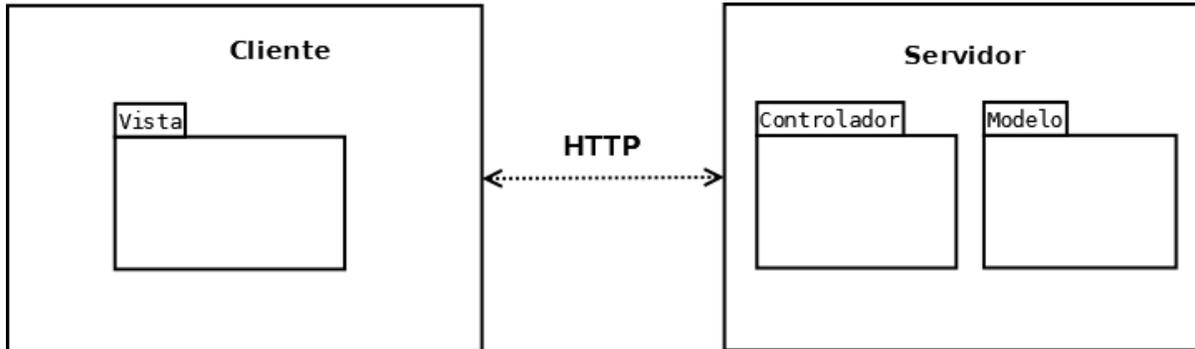
Además, Ruby on Rails, Trello y Git, son parte de los requerimientos del curso de Ingeniería de Software II.

Trello es ocupado para el seguimiento del proyecto, al igual que las reuniones diarias con SCRUM.

Las herramientas como Dia, PostgreSQL, Heroku y Sublime Text, son escogidas libremente por el equipo de desarrollo.

Concepto	Herramienta	Versión
Framework	Ruby on Rails.	4.2.3.
Diagramador UML	Dia.	0.97.2.
Manejador de bases de datos	PostgreSQL.	9.3.9.
Servidor de aplicaciones	Heroku.	Online.
IDE	Sublime Text.	4.0
Control de versiones	Git.	2.5.2.
Seguimiento del proyecto	Trello	Online

## Diagrama de distribución



El diagrama de distribución representa los nodos y sus componentes del sistema web. Al ser un sistema web no es necesario hacer la instalación de algún componente en cada máquina que tendrá acceso al sistema.

En el nodo del cliente, representada dada la arquitectura MVC, se encuentra la vista, la cual está compuesta por las interfaces gráficas y templates diseñados para la interacción del cliente con el sistema. Al ingresar los datos, el cliente al sistema, el nodo Servidor con el Controlador y el Modelo hacen conexiones a la base de datos y guardan, actualiza o eliminan datos y se ejecuta la coordinación y envío de correos, etc.

**NOTA:** En el presente documento se pide al estudiante incluir los diagramas de componentes y sus interfaces y el de diseño de base de datos. Con fines de ilustrar un avance, se omiten esos diagramas en el documento inicial de arquitectura, se creará un documento de arquitectura con estos diagramas para cada iteración ejemplificada en este manual (**Anexo B. B3, Anexo C. C3 y Anexo D. D3** para la primera, segunda y sexta iteración, respectivamente).

## **10. ANEXO B. DOCUMENTOS PRIMERA ITERACIÓN.**

**B1. Requerimientos de la primera iteración.**

**B2. Plan de la primera iteración.**

**B3. Arquitectura de la primera iteración.**

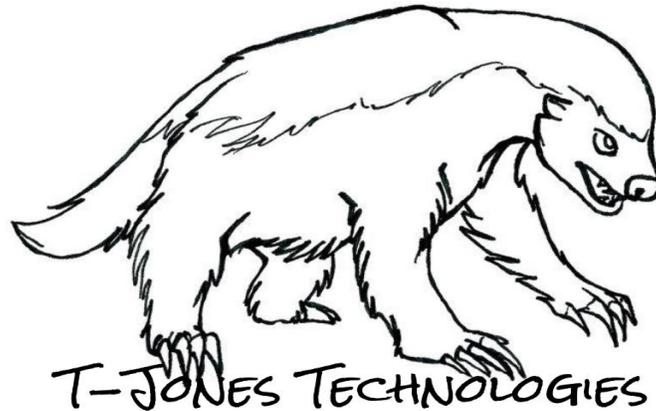
**B4. Retrospectiva.**

**B5. Product Backlog modificado. (Primera iteración)**

Anexo B. B1. Requerimientos de la primera iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Documento de Requerimientos de Software en la Primera Iteración**

**Pedrito**

**Sotanito**

**Luisita**

# Objetivos del software en la primera iteración.

El objetivo de la primera iteración es dar inicio a la creación del software empezando por el diseño de la base de datos, esto incluye la creación, consultas, eliminación y edición de datos. Además, se busca establecer un diseño para la vista y se pone orden a los elementos, de la página principal y de los formularios, que los componen. En complemento con la debida documentación que debe ser generada (Diagramas, descripción de funcionalidades, etc.) y uso de herramientas (Trello).

Lista de funcionalidades y de la primera iteración:

Funcionalidad
Registrar propuesta de curso
Implementar una plantilla para registrar cursos, ponentes (CV e información académica)
Diseñar base de datos
Crear diagrama relacional
Diseñar la vista principal
Crear página principal
Crear vista para registrar curso
Crear vista para registrar CV e información académica del Ponente
Implementar la agregación a la base de datos de Cursos
Implementar la eliminación a la base de datos de Cursos
Implementar la consulta a la base de datos de Cursos
Implementar la agregación a la base de datos de Ponente
Implementar la eliminación a la base de datos de Ponente
Implementar la consulta a la base de datos de Ponente
Editar datos de los Cursos
Editar CV e información académica de los Ponentes
Minutas de 15 min diarias
Documento de requerimientos de la primera iteración
Documento del plan de la primera iteración
Documento de arquitectura de la primera iteración
Retrospectiva de la primera iteración
Product Backlog de la primera iteración modificado.
Actualización del tablero en Trello
Realizar presentación de los avances
Junta con los clientes para presentar avances

## Identificación de los casos de uso

1. **Registrar propuesta de curso.** El ponente manda una descripción de curso que desea impartir llenando un formulario.
2. **Registrar CV y datos académicos del Ponente.** El ponente ingresa información correspondiente a su currículum vitae y datos académicos.
3. **Modificar la propuesta de curso e información.** Después de que el Administrador realiza las observaciones de la propuesta, el ponente debe realizar las correcciones del documento que mandó.
4. **Modificar la propuesta de Ponente** El Administrador realiza las observaciones de la propuesta del ponente para que este realice las correcciones del documento que mandó.
5. **Borrar Curso.** El ponente puede eliminar algún curso que el haya propuesto.

## Plan de pruebas de la primera iteración.

### Descripción:

#### 1. Registrar CV y datos académicos del Ponente.

Se ingresan datos para llenar los campos de información académica y currículum vitae y se envían, más tarde llegará un correo del servidor pidiendo confirmar la cuenta. Este nos redirige al sitio para iniciar sesión. No podremos iniciar sesión hasta que nos den de alta al sitio (la petición es revisada por el consejo académico).

### Casos de prueba:

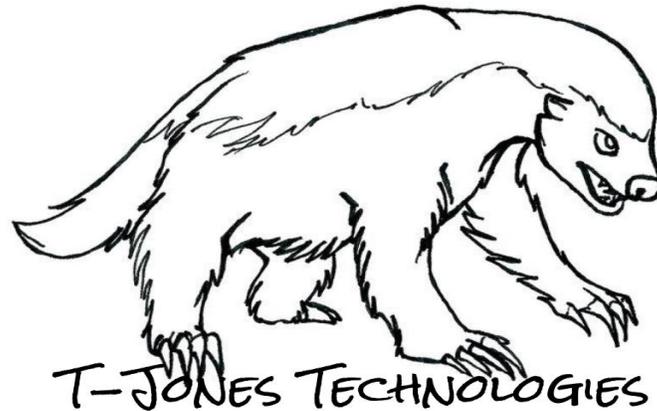
#### 1. Registrar CV y datos académicos del Ponente.

Casos de prueba con datos válidos		
Caso de Uso	Entradas	Resultados Esperados
1	Información académica y currículum. Todos los datos son válidos y el registro está completo	Registro exitoso.
Casos de prueba con datos inválidos		
Caso de Uso	Entradas	Resultados Esperados
1	Información académica y currículum. Algún campo tiene una entrada invalida o falta algún campo por llenar	Error en el registro.
Casos de prueba para las reglas del negocio		
Caso de Uso	Entradas	Resultados Esperados
1	Ingreso de CURP (expresión regular) $^{[A-Z]{1}[AEIOU]{1}[A-Z]{2}[0-9]{2}[01-9] 1[02]}(0[1-9] 1[0-9] 2[0-9] 3[01])[HM]{1}(AS BC BS CC CS CH CL CM DF DG GT GR HG JC MC MN MS NT NL OC PL QT QR SP SL SR TC TS TL VZ YN ZS NE)[B-DF-HJ-NP-TV-Z]{3}[0-9A-Z]{1}[0-9]{1}$$	Válido/inválido

Anexo B. B2. Plan de la primera iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Plan Primera Iteración**

**Pedrito**

**Sotanito**

**Luisita**

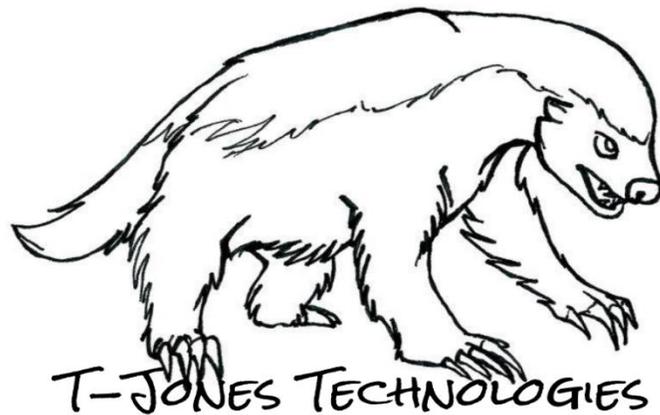
# Plan primera iteración.

ID	Funcionalidad	Responsable	Tiempo	Estado
1.1	Registrar propuesta de curso	Luisita	1hr.	Por hacer
1.2	Implementar una plantilla para registrar cursos, ponentes (CV e información académica)	Pedrito	1hr.	Por hacer
1.3	Diseñar base de datos	Sotanito	2hrs.	Por hacer
1.4	Crear diagrama relacional	Sotanito	1hr.	Por hacer
1.5	Diseñar la vista principal	Pedrito	3hrs.	Por hacer
1.6	Crear página principal	Pedrito	1hr.	Por hacer
1.7	Crear vista para registrar curso	Pedrito	2hrs.	Por hacer
1.8	Crear vista para registrar CV e información académica del Ponente	Sotanito	1hr.	Por hacer
2.1	Implementar la agregación a la base de datos de Cursos	Luisita	1hr.	Por hacer
2.2	Implementar la eliminación a la base de datos de Cursos	Luisita	1hr.	Por hacer
2.3	Implementar la consulta a la base de datos de Cursos	Luisita	45min.	Por hacer
2.4	Implementar la agregación a la base de datos de Ponente	Pedrito	45min.	Por hacer
2.5	Implementar la eliminación a la base de datos de Ponente	Pedrito	45min.	Por hacer
2.6	Implementar la consulta a la base de datos de Ponente	Pedrito	45min	Por hacer
3.1	Editar datos de los Cursos	Sotanito	2hrs.	Por hacer
3.2	Editar CV e información académica de los Ponentes	Sotanito	1hr.	Por hacer
4.1	Minutas de 15 min diarias	Todo el equipo	1.5hrs.	Por hacer
4.2	Documento de requerimientos de la primera iteración	Pedrito	1hr.	Por hacer
4.3	Documento del plan de la primera iteración	Pedrito	2hrs.	Por hacer
4.4	Documento de arquitectura de la primera iteración	Sotanito	2hrs.	Por hacer
4.5	Retrospectiva de la primera iteración	Luisita	1hrs.	Por hacer
4.6	Product Backlog de la primera iteración modificado.	Sotanito	30min.	Por hacer
4.7	Actualización del tablero en Trello	Todo el equipo	15min.	Por hacer
4.8	Realizar presentación de los avances	Luisita	2hrs.	Por hacer
4.9	Junta con los clientes para presentar avances	Todo el equipo	1.5hrs.	Por hacer

Anexo B. B3. Arquitectura primera iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



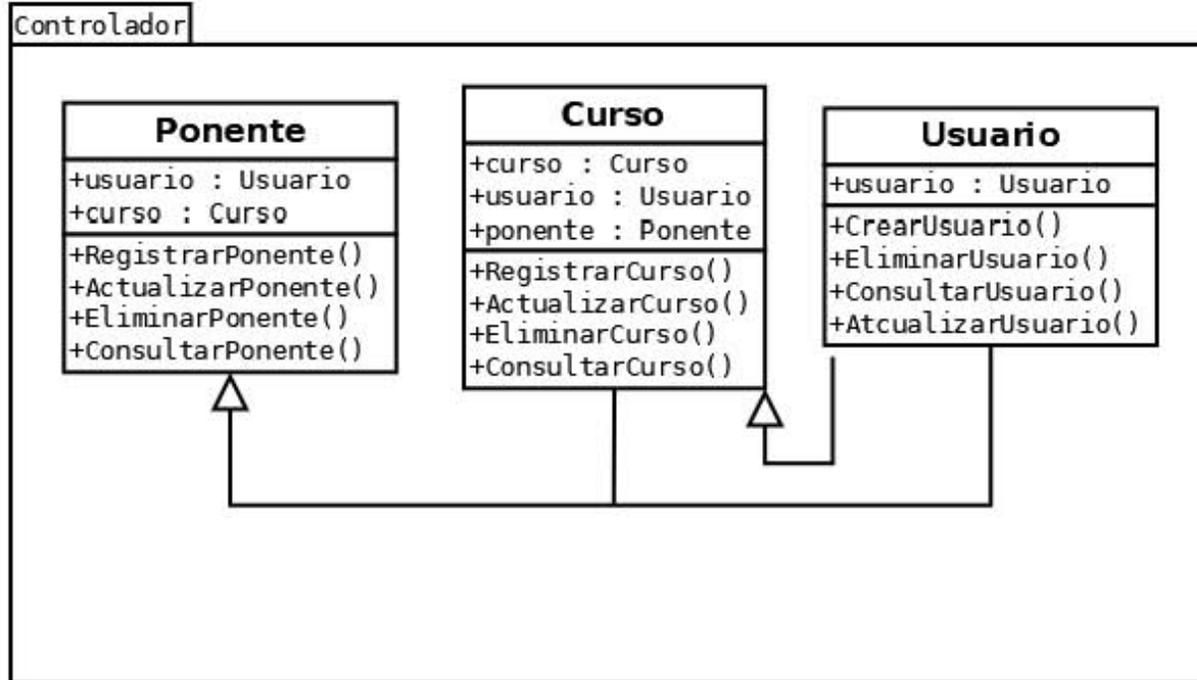
**Arquitectura primera iteración**

**Pedrito**

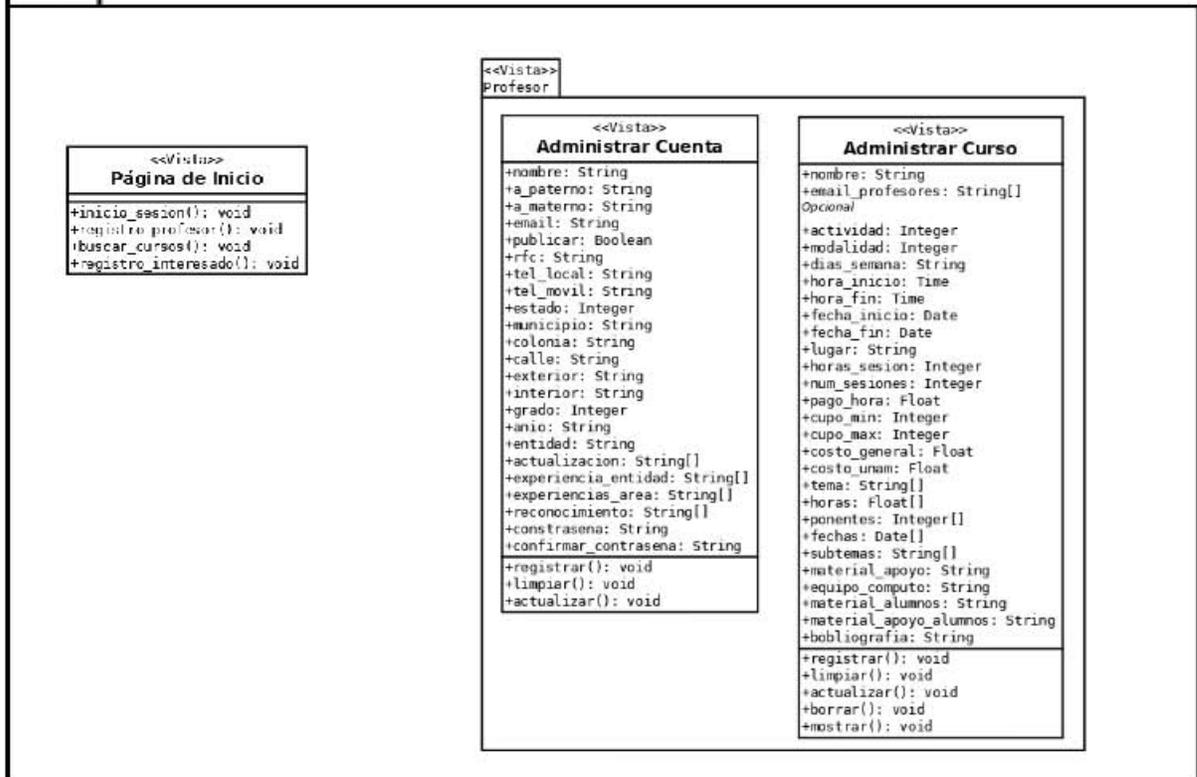
**Sotanito**

**Luisita**

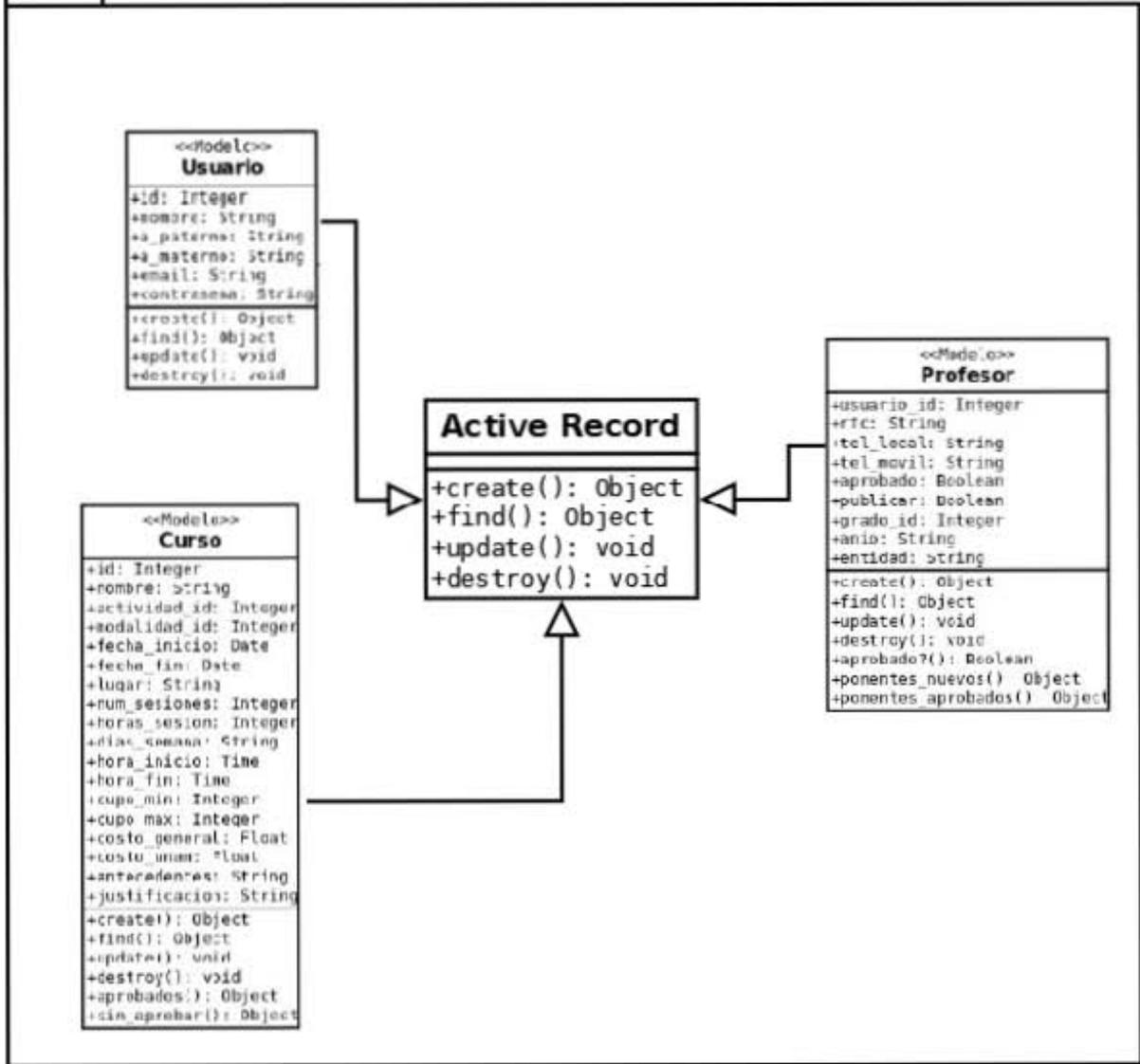
## Diagrama de componentes y sus interfaces.



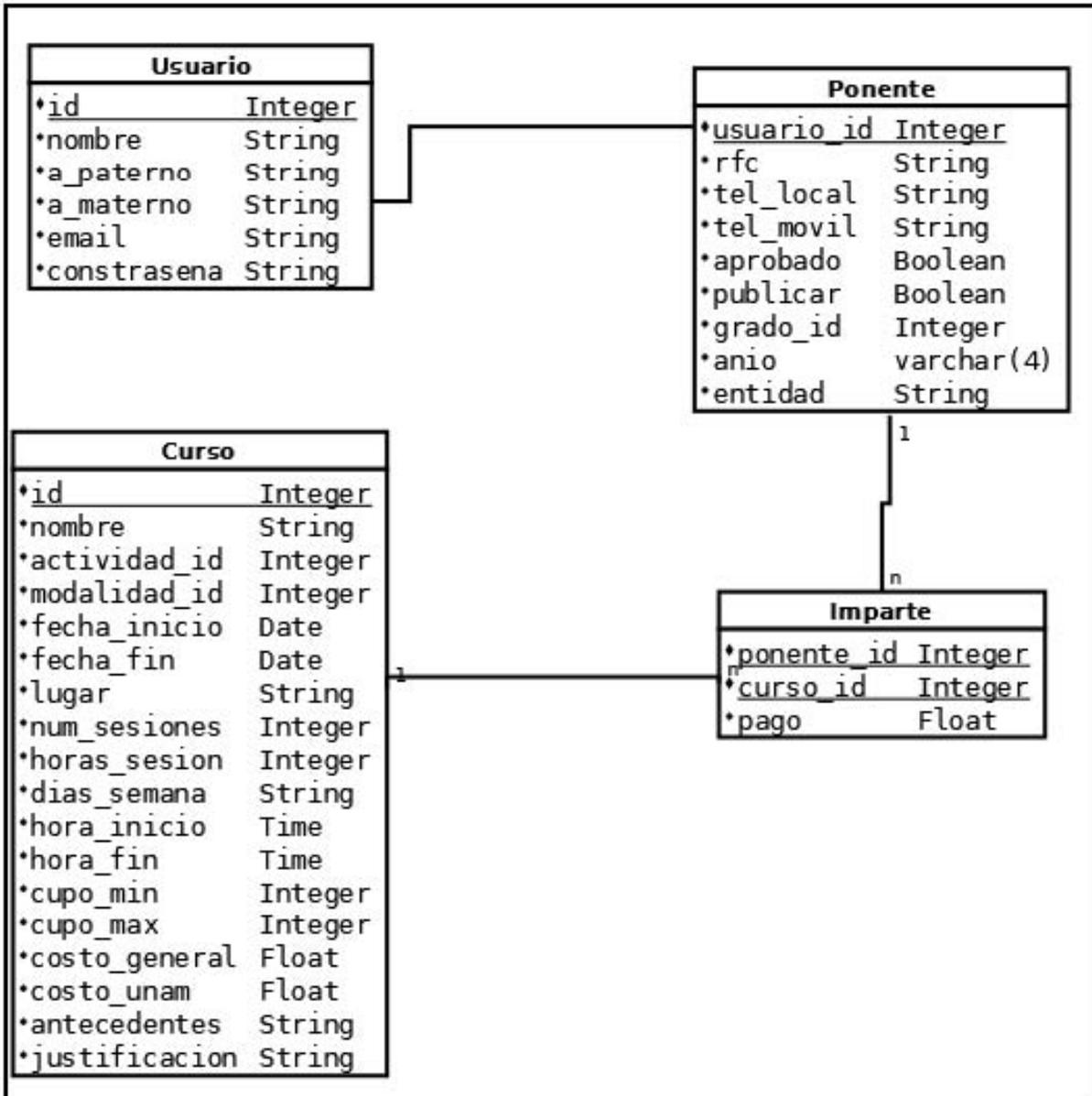
## Vista



Modelo



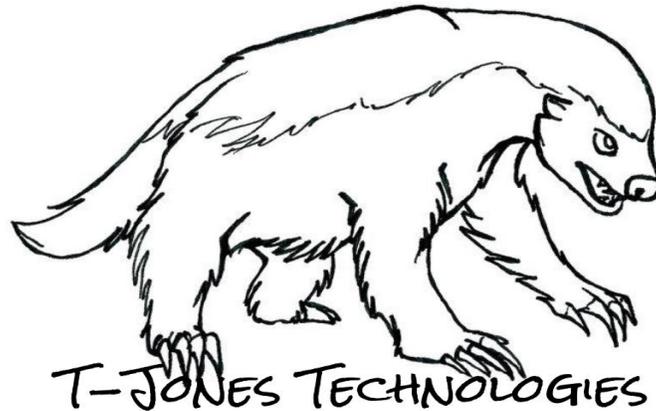
## Diseño de la base de datos.



Anexo B. B4. Retrospectiva.

# T-Jones Technologies

*“Honey badger never gives up”*



## Retrospectiva de la Primera Iteración

**Pedrito**

**Sotanito**

**Luisita**

## Retrospectiva de la Primera Iteración

Actividades pendientes en la iteración.

ID	Actividades Pendientes	Prioridad
Null	Null-Si no existen actividades de la iteración que hayan quedado pendientes el campo queda vacío-	Null

Actividades que se hicieron bien.

#Votos	Actividades que se hicieron bien
2	Creación de la página del sitio estuvo bien, parecida a la del cliente.
3	Los casos de uso fueron realizados con éxito.
3	El tablero funcionó muy bien al marcar las actividades y quién iba a realizarlas.
3	La interacción entre miembros del equipo fue buena.
0	Todo corrió perfectamente y no tuvimos contratiempos a excepción de la entrega final por falta de plataforma en las computadoras donde se mostraría el avance.

Necesidades de mejora.

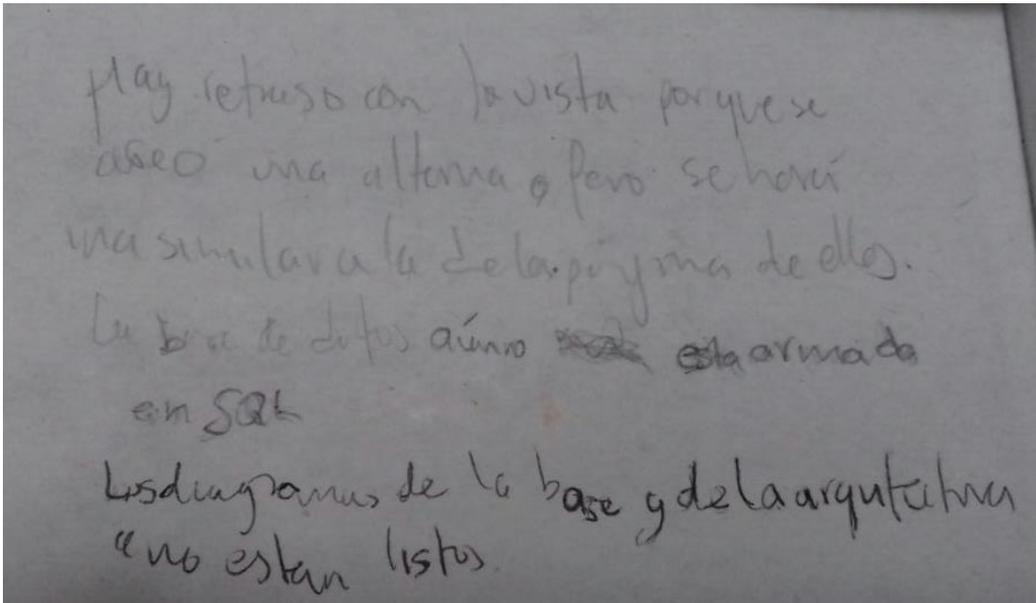
#Votos	Necesidades de mejora	Soluciones propuestas
2	Mejorar la comunicación	Estar al pendiente de los horarios de los demás y ocupar al máximo los tiempos libres.
3	Ponerse de acuerdo en las versiones que se suben a los repositorios.	Tener en cuenta que los cambios más significativos deben de ser registrados en una nueva versión.
2	El tiempo no nos alcanzó para lo que planeamos (tuvimos que acortar las actividades).	Proponer menos actividades para la segunda iteración.
3	Visitar a los clientes frecuentemente.	Tener mayor contacto personal o vía mail.
2	Al mostrar la versión final nos dimos cuenta que no había instalación de Ruby on Rails ni servidores de base de datos en las computadoras del laboratorio.	Tener previamente instalados los requerimientos básicos o llevar computadoras personales para mostrar los avances

Métricas de calidad

Iteración	#Casos de prueba	#Defectos encontrados	#Defectos corregidos
1	3	2	2

## Resultado de las minutas entre los integrantes del equipo.

### 1° Evidencia.



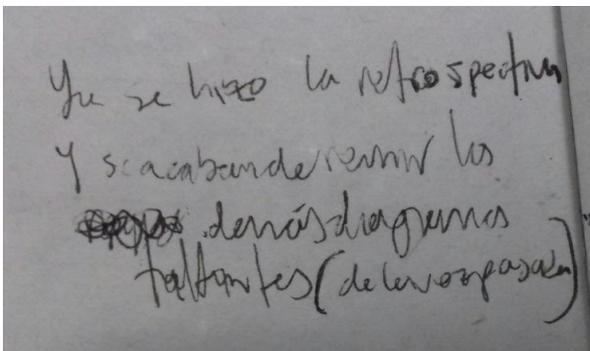
#### Comentarios:

"Hay retraso con la vista porque se creó una alterna, pero se hará una similar a la de ellos(SEAyC)"

"La base de datos aún no está armada (realizada) en SQL"

"Los diagramas de la base de datos y de la arquitectura no están listos"

### 2° Evidencia.



#### Comentarios:

"Ya se hizo la retrospectiva y se acaban de reunir los demás diagramas faltantes"

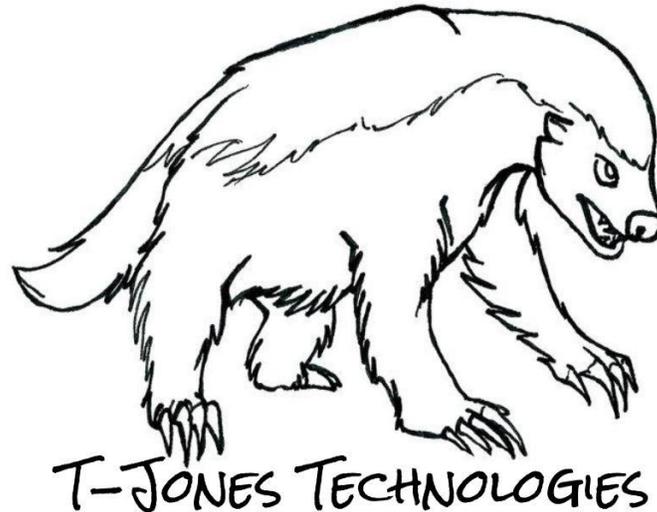
## Resultado de las juntas con los clientes.

No hubo recomendaciones por parte de los clientes, el resultado de esta junta y muestra de avances fue satisfactorio.

Anexo B. B5. Product Backlog modificado. (Primera iteración)

# **T-Jones Technologies**

*“Honey badger never gives up”*



**Product Backlog modificado.  
(primera iteración)**

**Pedrito**

**Sotanito**

**Luisita**

## Product Backlog modificado de la primera iteración.

Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración	Realizado
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs.	0	Si
02	Registrar CV e información de los ponentes	Alta	6hrs.	1	Si
03	Modificar CV e información de los ponentes	Media	2hrs.	1	Si
04	Registrar propuesta de curso	Alta	6hrs.	1	Si
05	Modificar propuesta de curso	Media	2hrs.	1	Si
06	Borrar curso	Media	4hrs.	1	Si
07	Generación de documentos, juntas, minutas, etc.	Alta	11.75 hrs.	1	Si

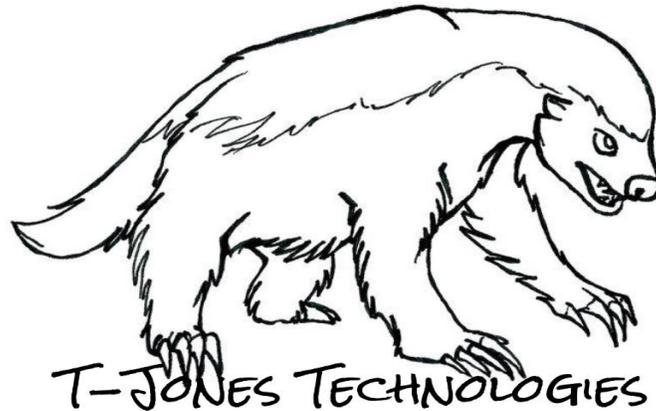
# 11. ANEXO C. DOCUMENTOS SEGUNDA ITERACIÓN.

- C1. Requerimientos de la segunda iteración.
- C2. Plan de la segunda iteración.
- C3. Arquitectura de la segunda iteración.
- C4. Retrospectiva.
- C5. Product Backlog modificado. (Segunda iteración)

Anexo C. C1. Requerimientos de la segunda iteración

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Documento de Requerimientos de Software en la Segunda Iteración**

**Pedrito**

**Sotanito**

**Luisita**

# Objetivos del software en la segunda iteración.

El objetivo de la segunda iteración es la creación de un actor importante del sistema, que es el Administrador y establecer las principales funciones o tareas que realizará, como: aceptar/rechazar las propuestas de ponentes y a su vez las propuestas de curso.

Lista de funcionalidades de la segunda iteración:

Funcionalidad
Implementar la vista de la página con el modelo de la SEAYC (marcos y estilo)
Aceptar/Rechazar propuesta de Ponente
Aceptar/Rechazar propuesta de Curso
Crear Súper Administrador
Agregar más administradores (Súper Administrador o administradores)
Manejo de sesiones en el sistema
Mostrar mensajes de error/confirmación
Crear vista para registrar Interesado
Vista de la búsqueda de cursos para el Interesado
Minutas de 15 min diarias
Documento de requerimientos de la segunda iteración
Documento del plan de la segunda iteración
Documento de arquitectura de la segunda iteración y actualización de la base de datos.
Retrospectiva de la segunda iteración
Product Backlog de la segunda iteración modificado.
Actualización del tablero en Trello
Realizar presentación de los avances
Junta con los clientes para presentar avances

## Identificación de los casos de uso

1. **Aceptar ponentes.** Si la solicitud de algún ponente cumple con los requisitos el administrador puede aceptar al ponente, si no es el caso puede rechazarlo.
2. **Aprobar curso.** Si el curso registrado por algún ponente está en las condiciones y cumple con los requisitos, el administrador puede aprobar el curso, si no es el caso puede rechazarlo.
3. **Agregar Administradores.** El "Súper" Administrador tiene la facultad de agregar más administradores que puedan administrar el sistema de la SEAYC, así como eliminarlos.
4. **Buscar Curso.** El interesado puede buscar algún curso que sea de su interés.

# Plan de pruebas de la segunda iteración.

## Descripción:

### Aprobar curso.

Dada una propuesta registrada de un curso, hecha por algún Ponente previamente registrado y aceptado, y esta propuesta cuenta con todos los requisitos, el administrador tiene el poder de aprobarla o de rechazarla.

## Casos de prueba:

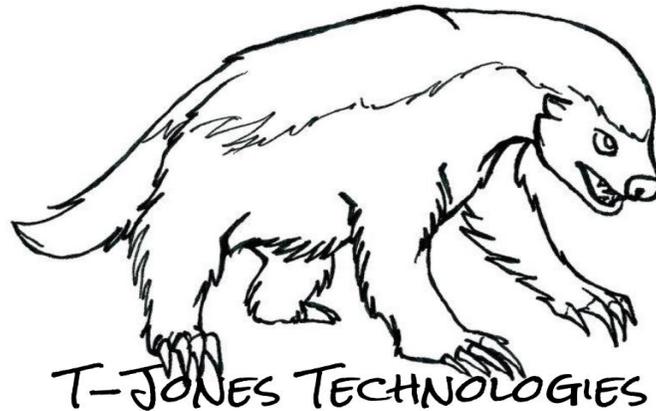
### Aprobar curso.

Casos de prueba con datos válidos		
Caso de Uso	Entradas	Resultados Esperados
2	Información descriptiva del curso. Todos los datos son válidos y el registro está completo	Propuesta aceptada.
Casos de prueba con datos inválidos		
Caso de Uso	Entradas	Resultados Esperados
2	Información descriptiva del curso. Algún campo tiene una entrada invalida, información no concisa o que el administrador considere ambigua.	Propuesta rechazada.
Casos de prueba para las reglas del negocio		
Caso de Uso	Entradas	Resultados Esperados
2	Costo del curso en forma numérica (\$)	Válido/inválido

Anexo C. C2. Plan de la segunda iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Plan Segunda Iteración**

**Pedrito**

**Sotanito**

**Luisita**

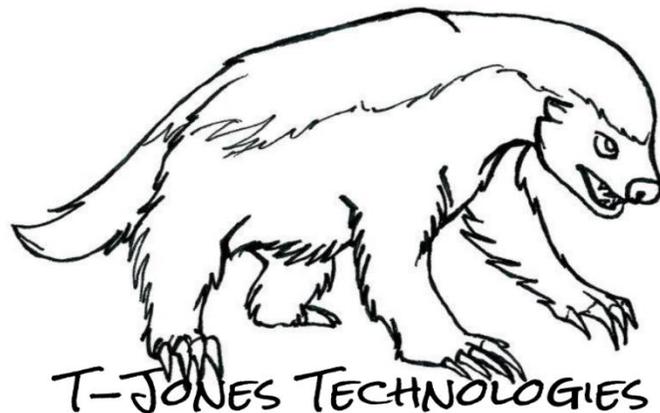
# Plan Segunda Iteración

ID	Funcionalidad	Responsable	Tiempo	Estado
1.1	Implementar la vista de la página con el modelo de la SEAYC (marcos y estilo)	Luisita	2hrs.	Por hacer
2.1	Aceptar/Rechazar propuesta de Ponente	Pedrito	2hr.	Por hacer
2.2	Aceptar/Rechazar propuesta de Curso	Pedrito	1hr.	Por hacer
2.3	Crear Súper Administrador	Sotanito	1hr.	Por hacer
2.4	Agregar más administradores (Súper Administrador o administradores)	Sotanito	1hr.	Por hacer
3.1	Manejo de sesiones en el sistema	Pedrito	2hrs.	Por hacer
3.2	Mostrar mensajes de error/confirmación	Sotanito	1hr.	Por hacer
4.1	Crear vista para registrar Interesado	Luisita	1hr.	Por hacer
4.2	Vista de la búsqueda de cursos para el Interesado	Luisita	2hrs.	Por hacer
5.1	Minutas de 15 min diarias	Todo el equipo	1hr.	Por hacer
5.2	Documento de requerimientos de la segunda iteración	Luisita	45min.	Por hacer
5.3	Documento del plan de la segunda iteración	Pedrito	45min.	Por hacer
5.4	Documento de arquitectura de la segunda iteración y actualización de la base de datos.	Pedrito	1hr.	Por hacer
5.5	Retrospectiva de la segunda iteración	Pedrito	45min	Por hacer
5.6	Product Backlog de la segunda iteración modificado.	Sotanito	2hrs.	Por hacer
5.7	Actualización del tablero en Trello	Sotanito	1hr.	Por hacer
5.8	Realizar presentación de los avances	Todo el equipo	1.5hrs.	Por hacer
5.9	Junta con los clientes para presentar avances	Todo el equipo	1hr.	Por hacer

Anexo C. C3. Arquitectura segunda iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



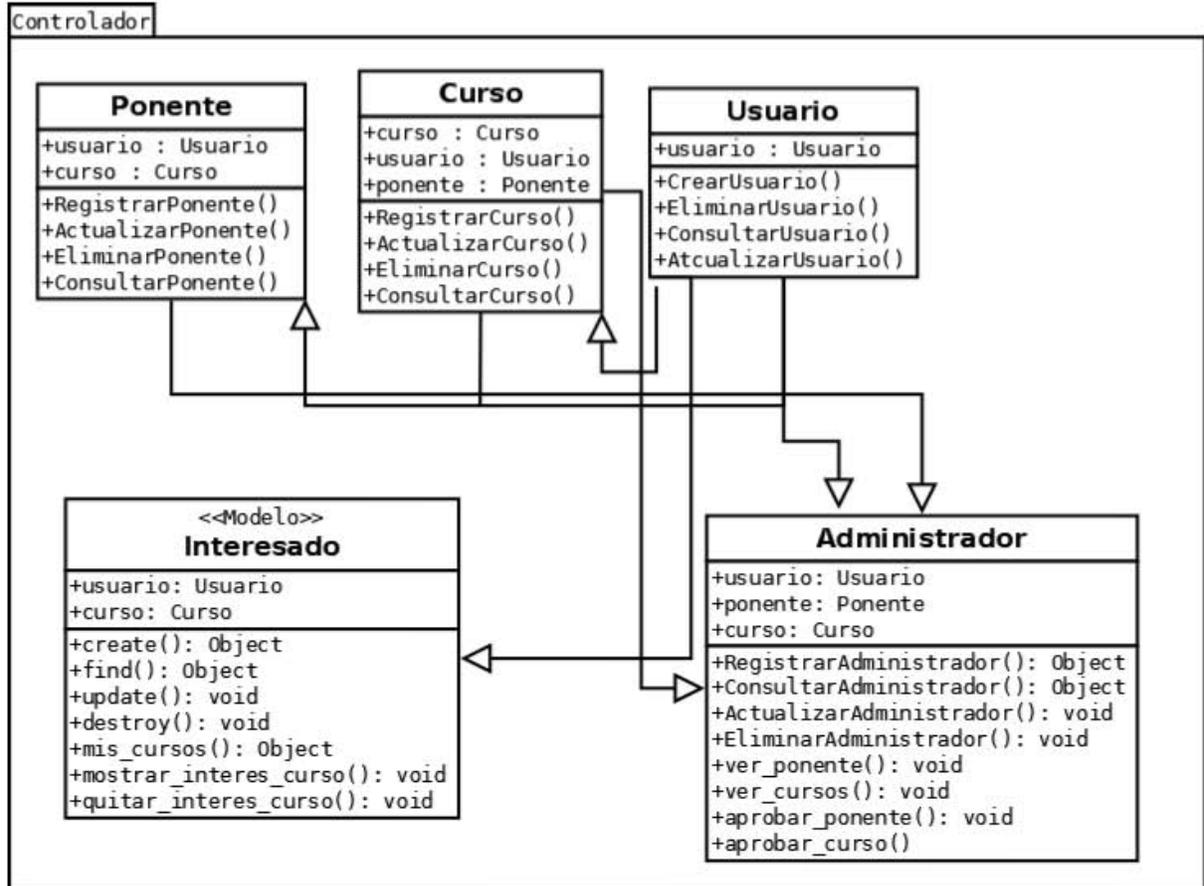
**Arquitectura segunda iteración**

**Pedrito**

**Sotanito**

**Luisita**

## Diagrama de componentes y sus interfaces



Vista

<<Vista>>  
**Página de Inicio**  
+inicio\_sesion(): void  
+registro\_profesor(): void  
+buscar\_cursos(): void  
+registro\_interesado(): void

<<Vista>>  
**Inicio de Sesión**  
+correo: String  
+contrasena: String  
+iniciar\_sesion(): void  
+recuperar\_contrasena(): void

<<Vista>>  
Administrador

<<Vista>>  
**Página de Inicio**  
+ver\_ponentes(): void  
+ver\_cursos(): void

<<Vista>>  
Interesado

<<Vista>>  
**Registro**  
+nombre: String  
+a\_paterno: String  
+a\_materno: String  
+email: String  
+rfc: String  
+tel\_local: String  
+tel\_movil: String  
+contrasena: String  
+confirmar\_contrasena: String  
+registrar(): void

<<Vista>>  
**Página de Inicio**  
+buscar\_cursos(): void  
+mis\_cursos(): void

<<Vista>>  
**Buscar Cursos**  
+curso: String  
+buscar(curso:String): void  
+ver\_curso(curso:Integer): void

<<Vista>>  
**Curso**  
+mostrar\_interes\_curso(curso:Integer): void  
+quitar\_interes\_curso(curso:Integer): void  
+buscar\_cursos(): void

<<Vista>>  
**Curso**  
+cursos\_sin\_aprobar(): void  
+aprobar\_curso(): void  
+exportar\_curso\_a\_pdf(): void  
+rechazar\_curso(): void  
+ver\_curso(curso:Integer): void

<<Vista>>  
**Ponente**  
+Ponentes\_sin\_aprobar(): void  
+aprobar\_Ponente(): void  
+exportar\_cv\_a\_pdf(): void  
+rechazar\_Ponente(): void  
+ver\_Ponente(Ponente:Integer): void

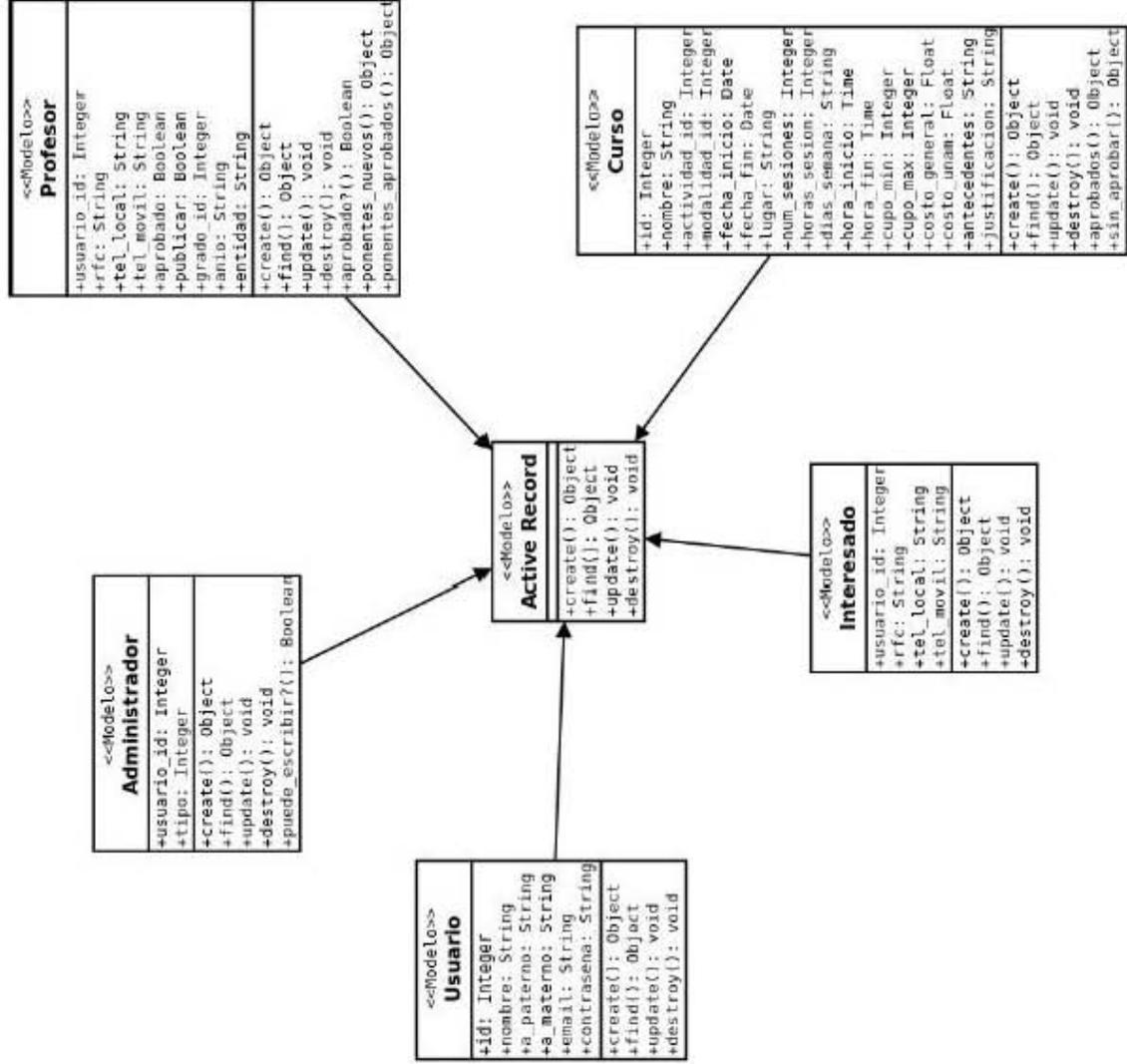
<<Vista>>  
**Rechazar Ponente**  
+mensaje: String  
+rechazar(): void

<<Vista>>  
Ponente

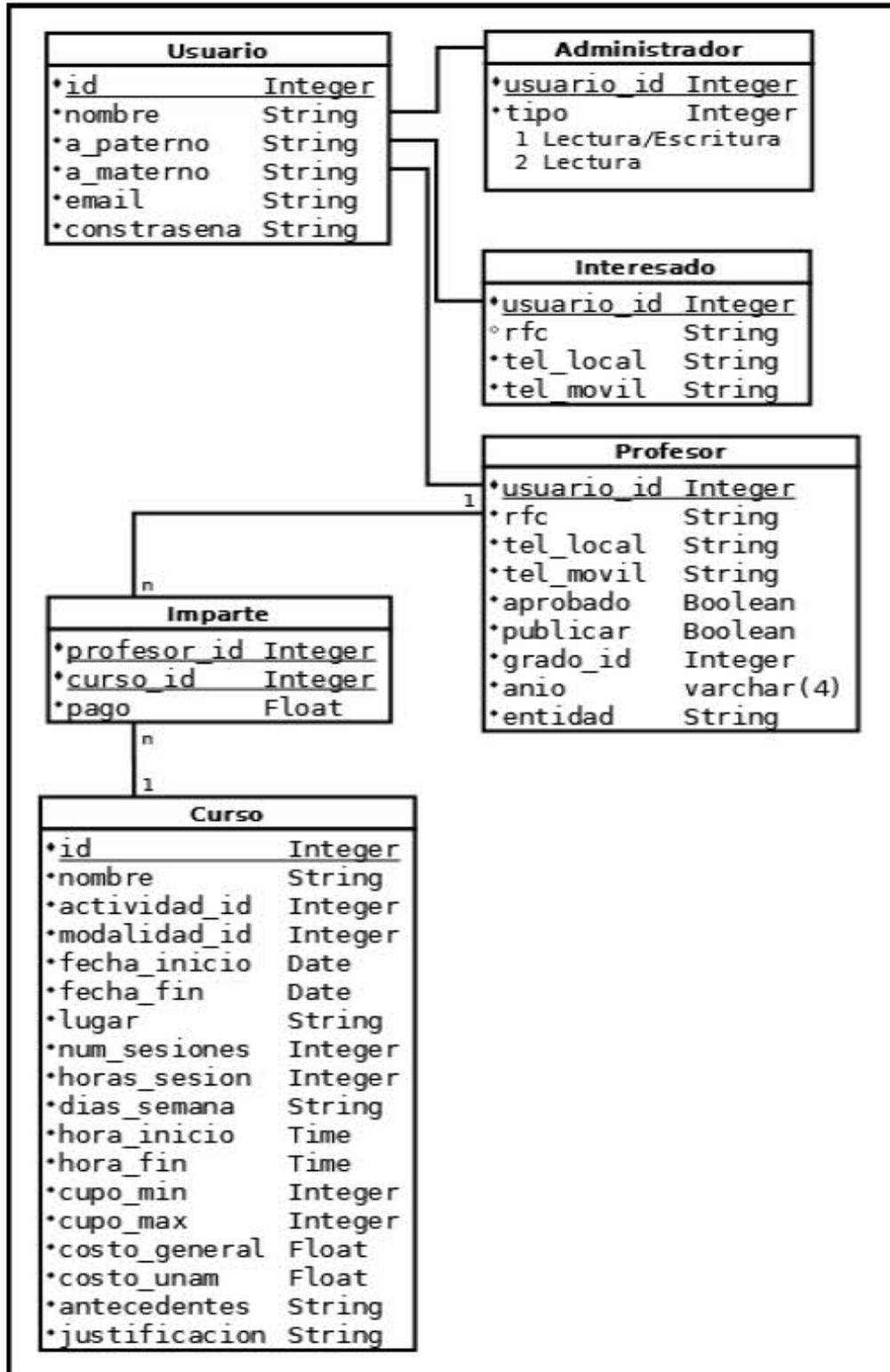
<<Vista>>  
**Administrar Cuenta**  
+nombre: String  
+a\_paterno: String  
+a\_materno: String  
+email: String  
+publicar: Boolean  
+rfc: String  
+tel\_local: String  
+tel\_movil: String  
+estado: Integer  
+municipio: String  
+colonia: String  
+calle: String  
+exterior: String  
+interior: String  
+grado: Integer  
+anio: String  
+entidad: String  
+actualizacion: String[]  
+experiencia\_entidad: String[]  
+experiencias\_area: String[]  
+reconocimiento: String[]  
+contrasena: String  
+confirmar\_contrasena: String  
+registrar(): void  
+limpiar(): void  
+actualizar(): void

<<Vista>>  
**Administrar Curso**  
+nombre: String  
+email\_Ponentes: String[]  
Opcional  
+actividad: Integer  
+modalidad: Integer  
+dias\_semana: String  
+hora\_inicio: Time  
+hora\_fin: Time  
+fecha\_inicio: Date  
+fecha\_fin: Date  
+lugar: String  
+horas\_sesion: Integer  
+num\_sesiones: Integer  
+pago\_hora: Float  
+cupo\_min: Integer  
+cupo\_max: Integer  
+costo\_general: Float  
+costo\_unam: Float  
+tema: String[]  
+horas: Float[]  
+ponentes: Integer[]  
+fechas: Date[]  
+subtemas: String[]  
+material\_apoyo: String  
+equipo\_computo: String  
+material\_alumnos: String  
+material\_apoyo\_alumnos: String  
+bibliografia: String  
+registrar(): void  
+limpiar(): void  
+actualizar(): void  
+borrar(): void  
+mostrar(): void

Modelo



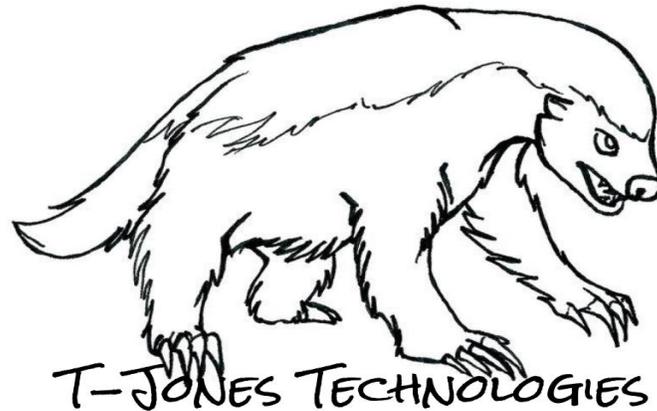
Diseño de la base de datos.



Anexo C. C4. Retrospectiva.

# T-Jones Technologies

*“Honey badger never gives up”*



## Retrospectiva de la Segunda Iteración

**Pedrito**

**Sotanito**

**Luisita**

## Retrospectiva de la Segunda Iteración

Actividades pendientes en la iteración.

ID	Actividades Pendientes	Prioridad
1	Agregar más Administradores (Súper administrador o administrador) a partir de un Súper Administrador	Media
2	Algunos mensajes de confirmación aún no aparecen en color verde	baja

Actividades que se hicieron bien.

#Votos	Actividades que se hicieron bien
3	Se realizó correctamente las actividades marcadas en el plan de la segunda iteración.
3	La actualización y manejo correcto del tablero de trello.
3	Las reuniones diarias y con los clientes ocurrieron satisfactoriamente.

Necesidades de mejora.

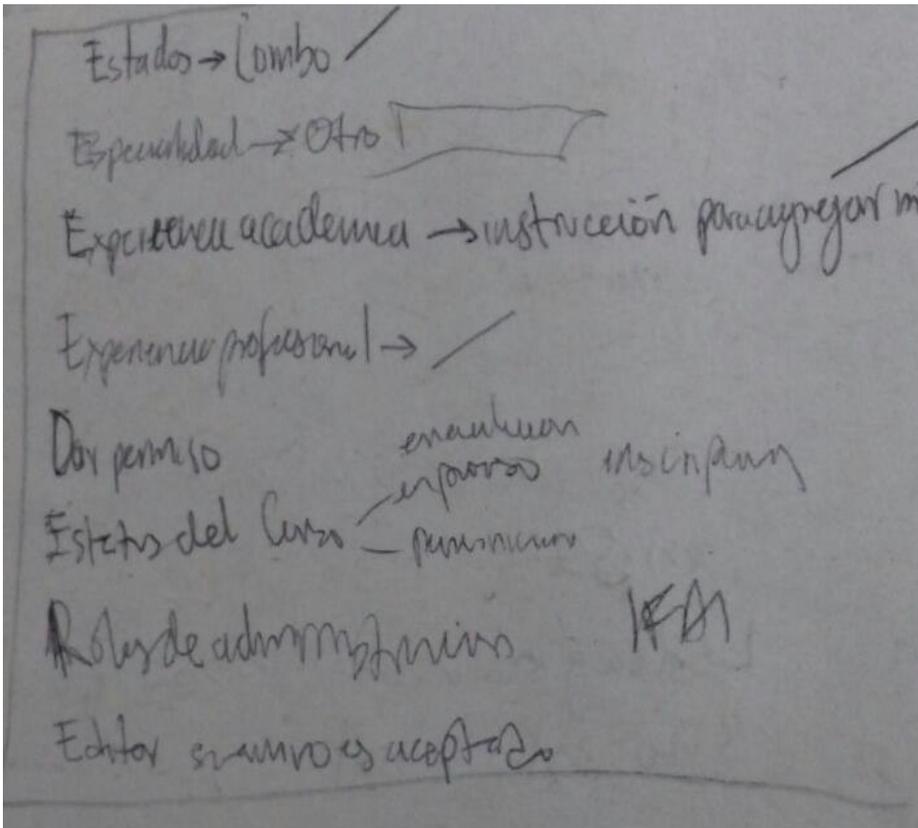
#Votos	Necesidades de mejora	Soluciones propuestas
3	Visitar al cliente tardíamente	Visitarlos más seguido
2	Faltar a clases	Ir más seguido.
2	Documento de arquitectura faltante	Terminarlo.

Métricas de calidad

Iteración	#Casos de prueba	#Defectos encontrados	#Defectos corregidos
2	2	0	0

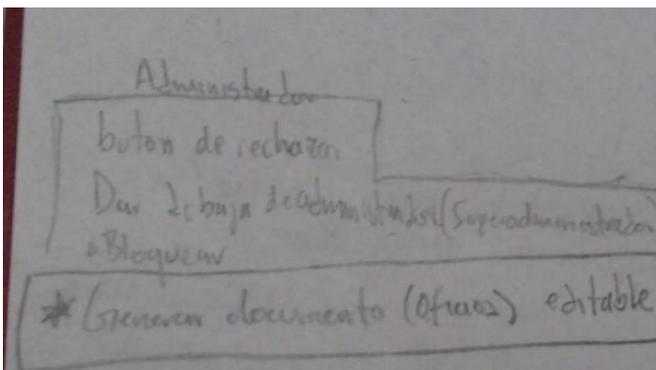
## Resultado de las minutas entre los integrantes del equipo.

### 1° Evidencia.



Se hicieron algunas observaciones y recomendaciones, de los miembros del equipo para las vistas y facilitar el uso de estas.

### 2° Evidencia.

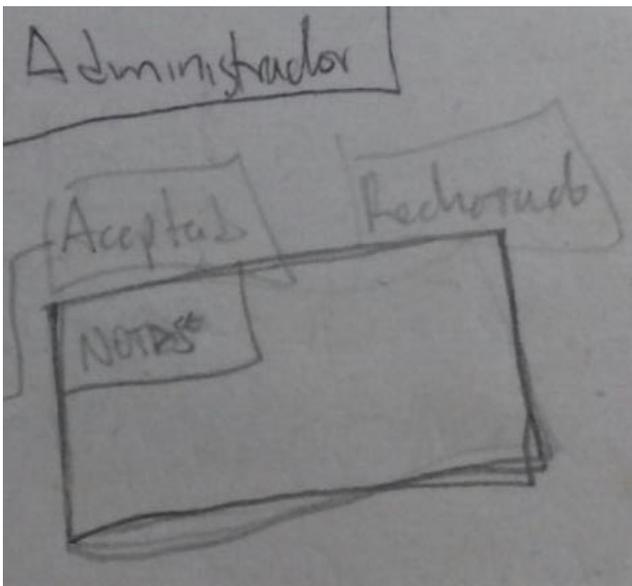


Se empezó a aclarar la idea de un “súper” administrador.

“administrador, botón de rechazar, dar de baja de administrador (súper administrador) o bloquear”

## Resultado de las juntas con los clientes.

### 1° Evidencia

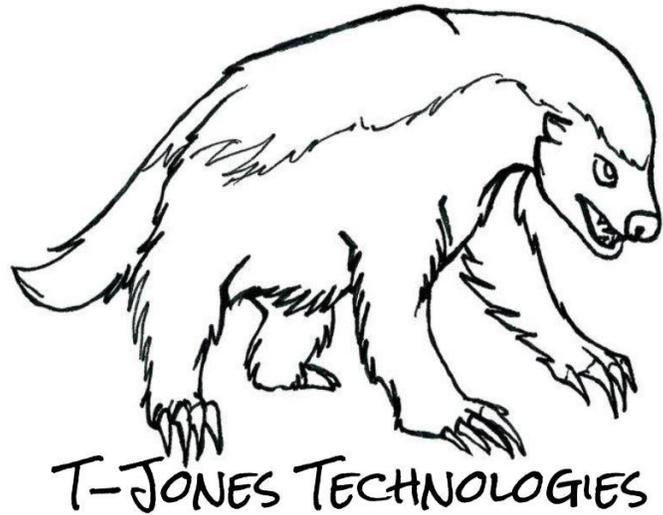


Los clientes nos hicieron la recomendación de que “los administradores tenían que tener un botón para aceptar o rechazar (Ponentes o Cursos) y además poder agregar un mensaje de recomendación o de aceptación (Notas).

Anexo C. C5. Product Backlog modificado. (Segunda iteración)

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Product Backlog (Segunda iteración)**

**Pedrito**

**Sotanito**

**Luisita**

## Product Backlog modificado de la segunda iteración.

Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs.	0
02	Registrar CV e información de los ponentes	Alta	6hrs.	1
03	Modificar CV e información de los ponentes	Media	2hrs.	1
04	Registrar propuesta de curso	Alta	6hrs.	1
05	Modificar propuesta de curso	Media	2hrs.	1
06	Borrar curso	Media	4hrs.	1
07	Documentación de la primera iteración	Alta	11.75 hrs.	1
08	Aprobar a los ponentes	Baja	1hrs.	2
09	Agregar Administradores	Baja	4hrs.	2
10	Autorizar Curso	Alta	4hrs.	2
11	Buscar curso	Media	4hrs..	2
12	Documentación de la segunda iteración	Alta	9.75hrs.	2

## 12. ANEXO D. DOCUMENTOS SEXTA ITERACIÓN.

D1. Requerimientos de la sexta iteración.

D2. Plan de la sexta iteración.

D3. Arquitectura de la sexta iteración.

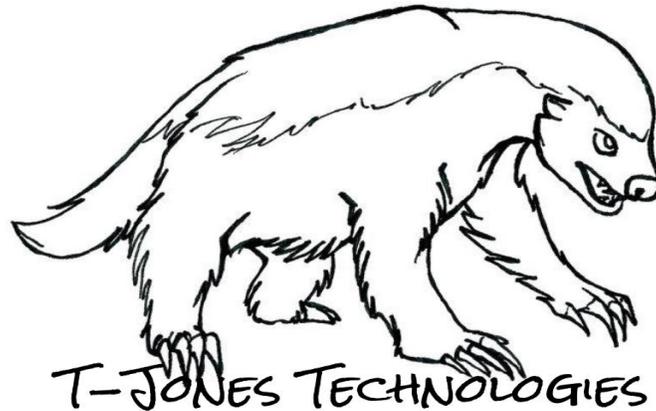
D4. Retrospectiva.

D5. Product Backlog Finalizado.

Anexo D. D1. Requerimientos de la sexta iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Documento de Requerimientos de Software en la Sexta Iteración**

**Pedrito**

**Sotanito**

**Luisita**

## Objetivos del software en la sexta iteración.

El objetivo de la sexta iteración es terminar detalles técnicos y de documentación para tener completo el proyecto, es una iteración que permite concluir lo que fue postergado o lo que quedo incompleto durante las iteraciones pasadas.

Lista de funcionalidades de la primera iteración:

Funcionalidad
Funcionalidades incompletas (Detalles en los formularios)
Documento de Requisitos y Casos de uso completados
Diseño de la base de datos (completo)
Plan de Pruebas mínimo (Casos de Uso que lo ameriten)
Software Ensamblado
Manual de Instalación
Presentación final del sistema.
Minutas de 15 min diarias
Documento de requerimientos de la segunda iteración
Documento del plan de la sexta iteración
Documento de arquitectura de la sexta iteración
Retrospectiva de la sexta iteración
Product Backlog de la sexta iteración modificado.
Actualización del tablero en Trello
Junta con los clientes para presentar el sistema web completo y entregarle entregables.

## Identificación de los casos de uso

En este caso, los casos de usos ya estaban finalizados, sólo faltaban detalles específicos y técnicos. Con respecto a la documentación, se finalizaron algunos pendientes como la base de datos y documentos de requerimientos y casos de uso. Además, se realizó una presentación para el cliente, así como el manual de instalación.

## Plan de pruebas de la sexta iteración.

Descripción:

Casos de prueba:

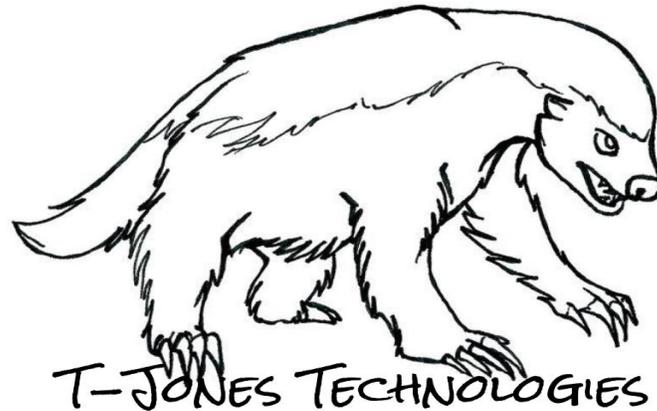
1.

<b>Casos de prueba con datos válidos</b>		
<b>Caso de Uso</b>	<b>Entradas</b>	<b>Resultados Esperados</b>
null	null	null
<b>Casos de prueba con datos inválidos</b>		
<b>Caso de Uso</b>	<b>Entradas</b>	<b>Resultados Esperados</b>
Null	Null	null
<b>Casos de prueba para las reglas del negocio</b>		
<b>Caso de Uso</b>	<b>Entradas</b>	<b>Resultados Esperados</b>
Null	null	null

Anexo D. D2. Plan de la sexta iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



## **Plan Sexta Iteración**

**Pedrito**

**Sotanito**

**Luisita**

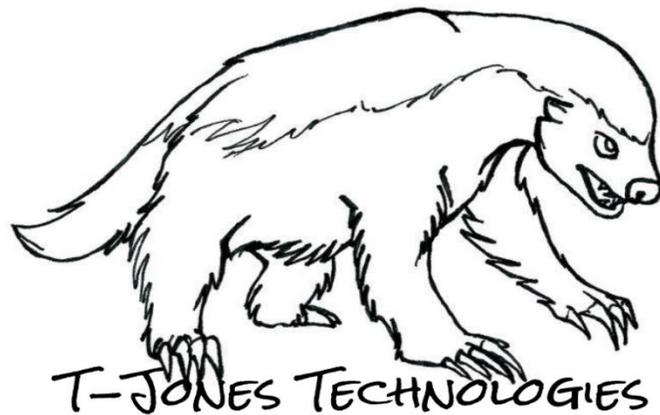
## Plan de la sexta iteración

ID	Funcionalidad	Responsable	Tiempo	Estado
1.1	Funcionalidades incompletas (Detalles en los formularios)	Pedrito	45min.	Por hacer
2.1	Documento de Requisitos y Casos de uso completados	Sotanito	1hr.	Por hacer
2.2	Diseño de la base de datos (completo)	Luisita	45min.	Por hacer
3.1	Plan de Pruebas mínimo (Casos de Uso que lo ameriten)	Sotanito	1hr.	Por hacer
4.1	Software Ensamblado	Pedrito	2hrs.	Por hacer
5.1	Manual de Instalación	Luisita	2hrs.	Por hacer
6.1	Presentación final del sistema.	Todo el equipo	3hrs.	Por hacer
7.1	Minutas de 15 min diarias	Todo el equipo	1hr.	Por hacer
7.2	Documento de requerimientos de la segunda iteración	Luisita	1hr.	Por hacer
7.3	Documento del plan de la sexta iteración	Pedrito	45min.	Por hacer
7.4	Documento de arquitectura de la sexta iteración	Pedrito	1hr.	Por hacer
7.5	Retrospectiva de la sexta iteración	Pedrito	45min	Por hacer
7.6	Product Backlog de la sexta iteración modificado.	Sotanito	2hrs.	Por hacer
5.7	Actualización del tablero en Trello	Sotanito	1hr.	Por hacer
5.8	Junta con los clientes para presentar el sistema web completo y entregarle entregables.	Todo el equipo	1hr.	Por hacer

Anexo D. D3. Arquitectura sexta iteración.

# **T-Jones Technologies**

*“Honey badger never gives up”*



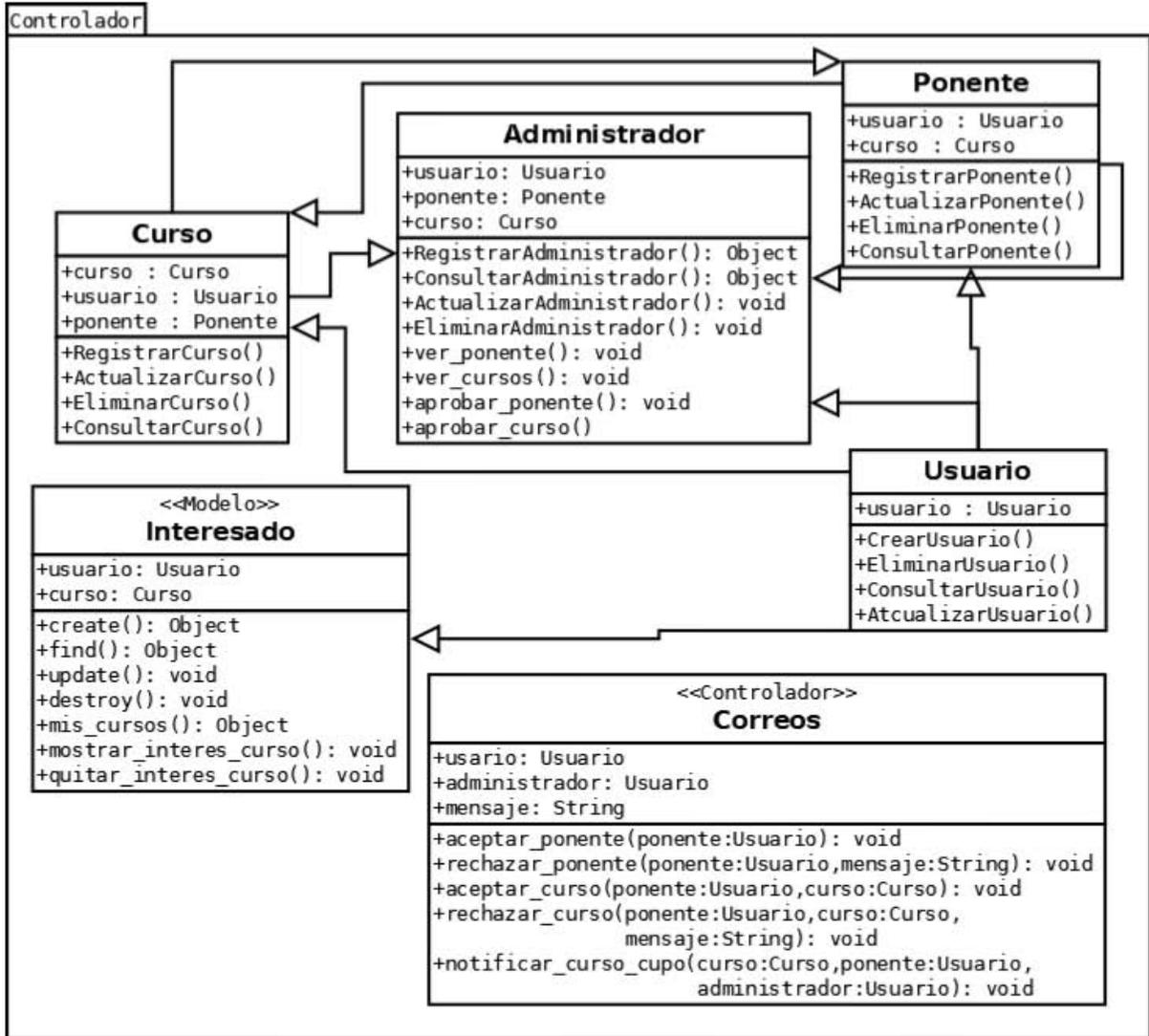
## **Arquitectura sexta iteración**

**Pedrito**

**Sotanito**

**Luisita**

## Diagrama de componentes y sus interfaces



Vista

```
<<Vista>>
Página de Inicio
+inicio_sesion(): void
+registro_profesor(): void
+buscar_cursos(): void
+registro_interesado(): void
```

```
<<Vista>>
Inicio de Sesión
+correo: String
+contrasena: String
+iniciar_sesion(): void
+recuperar_contrasena(): void
```

```
<<Vista>>
Administrador
```

```
<<Vista>>
Página de Inicio
+ver_ponentes(): void
+ver_cursos(): void
```

```
<<Vista>>
Curso
+cursos_sin_aprobar(): void
+aprobar_curso(): void
+exportar_curso_a_pdf(): void
+rechazar_curso(): void
+ver_curso(curso:Integer): void
```

```
<<Vista>>
Ponente
+Ponentes_sin_aprobar(): void
+aprobar_Ponente(): void
+exportar_cv_a_pdf(): void
+rechazar_Ponente(): void
+ver_Ponente(Ponente:Integer): void
```

```
<<Vista>>
Rechazar Ponente
+mensaje: String
+rechazar(): void
```

```
<<Vista>>
Rechazar Curso
+mensaje: String
+rechazar(): void
```

```
<<Vista>>
Interesado
```

```
<<Vista>>
Registro
+nombre: String
+a_paterno: String
+a_materno: String
+email: String
+rfc: String
+tel_local: String
+tel_movil: String
+contrasena: String
+confirmar_contrasena: String
+registrar(): void
```

```
<<Vista>>
Página de Inicio
+buscar_cursos(): void
+mis_cursos(): void
```

```
<<Vista>>
Buscar Cursos
+curso: String
+buscar(curso:String): void
+ver_curso(curso:Integer): void
```

```
<<Vista>>
Curso
+mostrar_interes_curso(curso:Integer): void
+quitar_interes_curso(curso:Integer): void
+buscar_cursos(): void
```

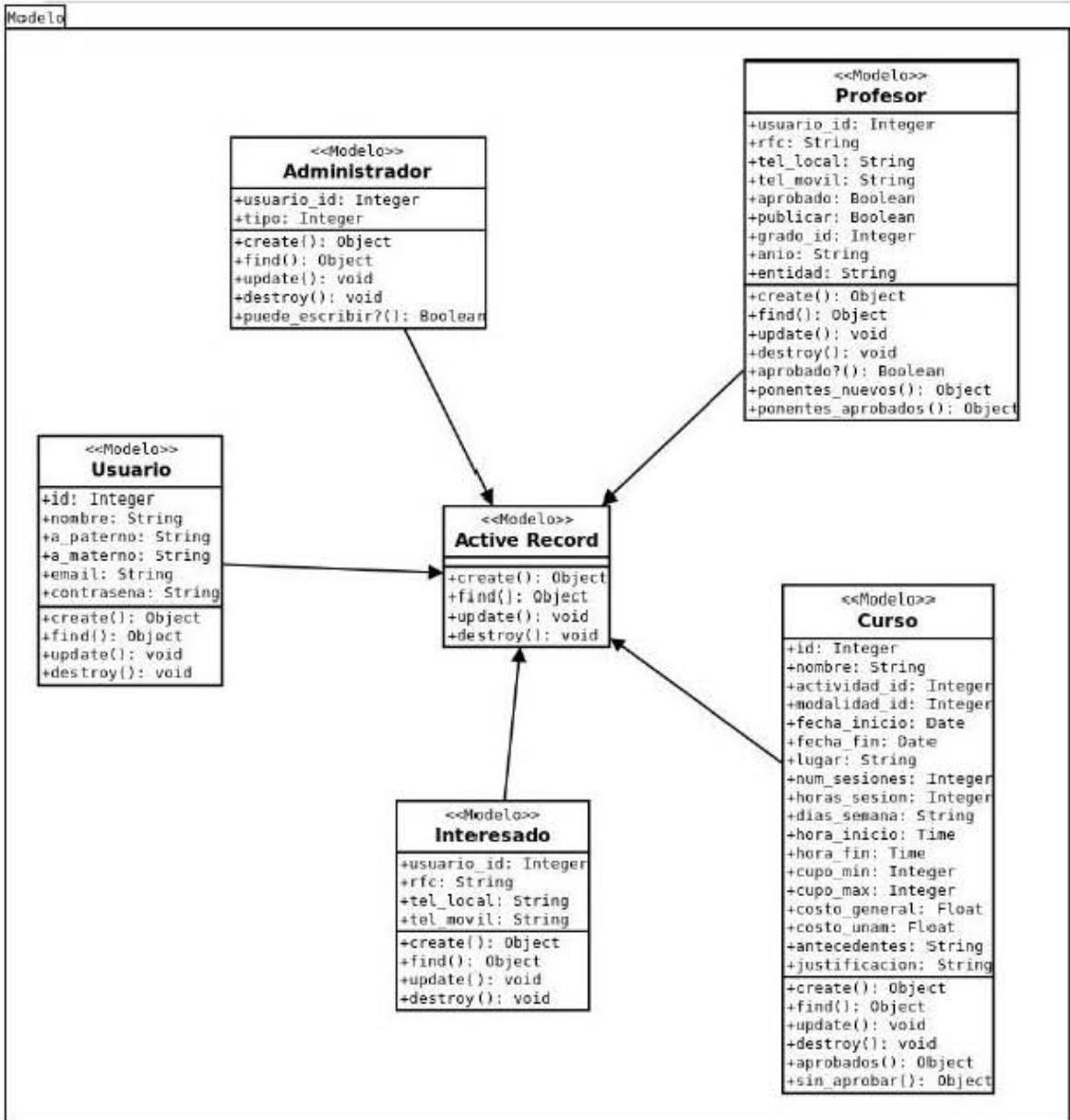
```
<<Vista>>
Ponente
```

```
<<Vista>>
Administrar Cuenta
+nombre: String
+a_paterno: String
+a_materno: String
+email: String
+publicar: Boolean
+rfc: String
+tel_local: String
+tel_movil: String
+estado: Integer
+municipio: String
+colonia: String
+calle: String
+exterior: String
+interior: String
+grado: Integer
+anio: String
+entidad: String
+actualizacion: String[]
+experiencia_entidad: String[]
+experiencias_area: String[]
+reconocimiento: String[]
+contrasena: String
+confirmar_contrasena: String
+registrar(): void
+limpiar(): void
+actualizar(): void
```

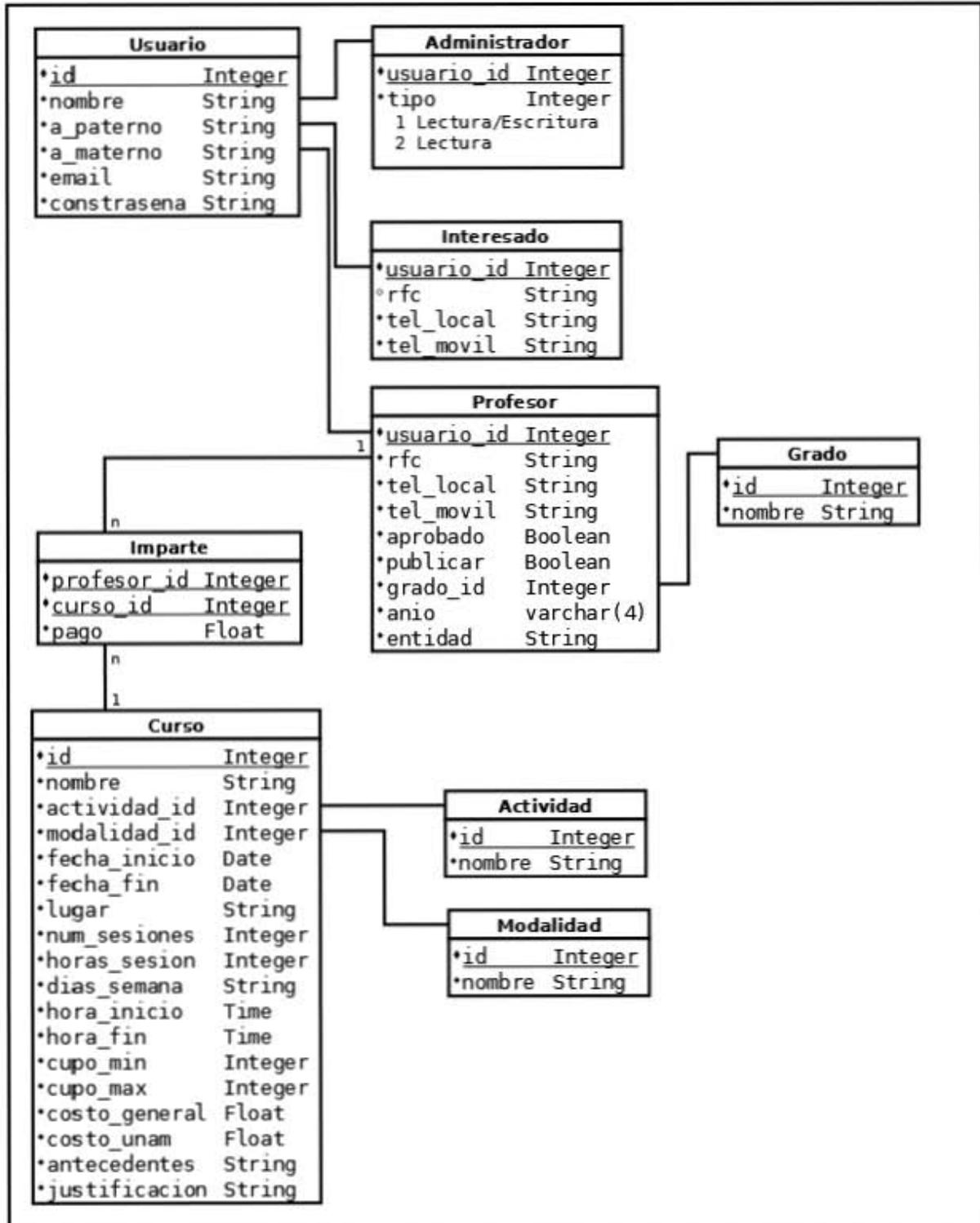
```
<<Vista>>
Administrar Curso
+nombre: String
+email_Ponentes: String[]
Opcional
+actividad: Integer
+modalidad: Integer
+dias_semana: String
+hora_inicio: Time
+hora_fin: Time
+fecha_inicio: Date
+fecha_fin: Date
+lugar: String
+horas_sesion: Integer
+num_sesiones: Integer
+pago_hora: Float
+cupo_min: Integer
+cupo_max: Integer
+costo_general: Float
+costo_unam: Float
+tema: String[]
+horas: Float[]
+ponentes: Integer[]
+fechas: Date[]
+subtemas: String[]
+material_apoyo: String
+equipo_computo: String
+material_alumnos: String
+material_apoyo_alumnos: String
+bibliografia: String
+registrar(): void
+limpiar(): void
+actualizar(): void
+borrar(): void
+mostrar(): void
```

```
<<Vista>>
Recuperar Contraseña
+email: String
+recuperar_contrasena(): void
```

```
<<Vista>>
Buscar Cursos
+curso: String
+buscar(curso:String): void
+ver_curso(curso:Integer): void
```

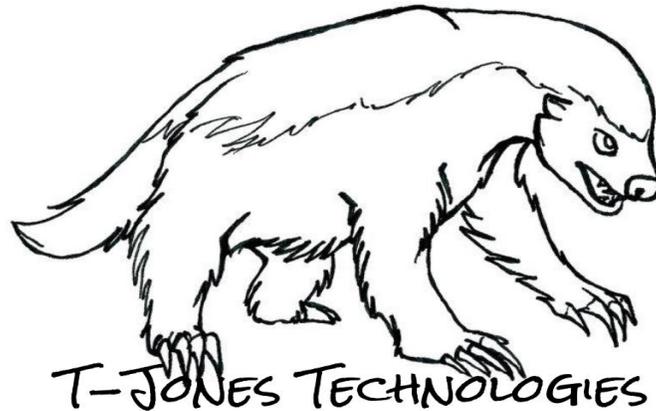


## Diseño de la base de datos.



# T-Jones Technologies

*“Honey badger never gives up”*



## Retrospectiva de la Sexta Iteración

**Pedrito**

**Sotanito**

**Luisita**

# Retrospectiva de la Sexta Iteración

Actividades pendientes en la iteración.

ID	Actividades Pendientes	Prioridad
01	Realizar la instalación del producto para el cliente	Importante.

Actividades que se hicieron bien.

#Votos	Actividades que se hicieron bien
3	Buena coordinación entre los miembros del equipo para afinar los últimos detalles.
2	Últimas reuniones con los "Clientes Reales" antes de la entrega final.
3	Las juntas diarias.
3	Asistir a clases.
3	Comunicación dentro del equipo.

Necesidades de mejora.

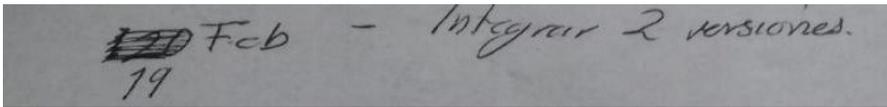
#Votos	Necesidades de mejora	Soluciones propuestas
3	Se juntaron varios pendientes por ser la última iteración.	Hacer buena administración de nuestro tiempo.

Métricas de calidad

Iteración	#Casos de prueba	#Defectos encontrados	#Defectos corregidos
5	5	2	2

## Resultado de las minutas entre los integrantes del equipo.

### 1° Evidencia

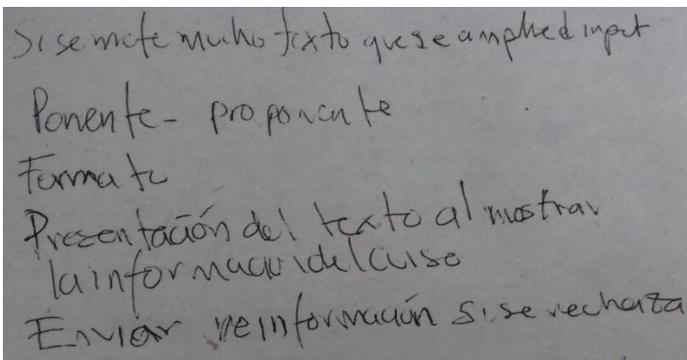


#### Comentarios:

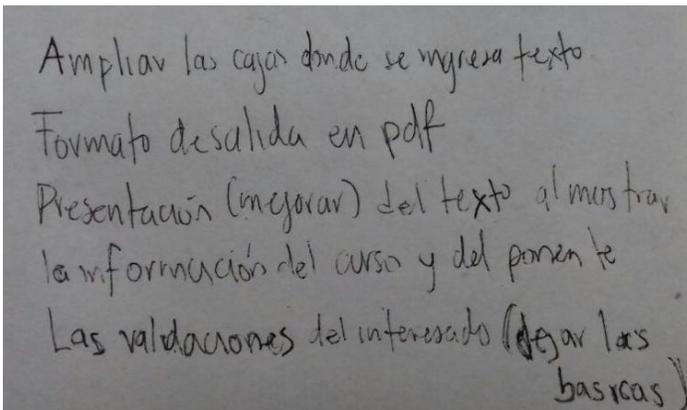
"19 de Feb-Integrar 2 versiones" Se programó la integración de dos versiones que se tenían para la fecha indicada.

## Resultado de las juntas con los clientes.

### 1° Evidencia



### 2° Evidencia.



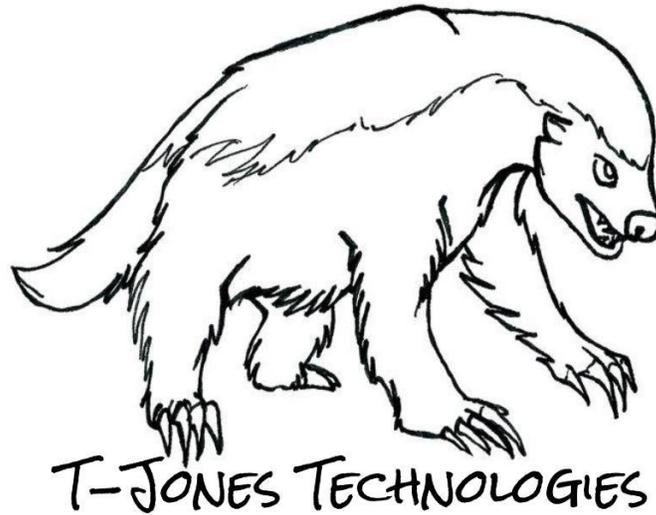
#### Comentarios:

Los clientes nos hicieron últimas recomendaciones sobre las vistas. En especial dónde se llenan formularios.

Anexo D. D5. Product Backlog finalizado.

# **T-Jones Technologies**

*“Honey badger never gives up”*



**Product Backlog finalizado**

**Pedrito**

**Sotanito**

**Luisita**

## Product Backlog finalizado.

Id	Funcionalidad o Caso de Uso	Prioridad	Estimación	Iteración
01	Documentos iniciales y diseño de formatos e imagen del equipo	Alta	10hrs	0
02	Registrar CV e información de los ponentes	Alta	6hrs.	1
03	Modificar CV e información de los ponentes	Media	2hrs.	1
04	Registrar propuesta de curso	Alta	6hrs.	1
05	Modificar propuesta de curso	Media	2hrs.	1
06	Borrar curso	Media	4hrs.	1
07	Documentación de la primera iteración	Alta	11.75 hrs.	1
08	Aprobar a los ponentes	Baja	1hrs.	2
09	Agregar Administradores	Baja	4hrs.	2
10	Autorizar Curso	Alta	4hrs.	2
11	Buscar curso	Media	4hrs..	2
12	Documentación de la segunda iteración	Alta	9.75hrs.	2
13	Registrar en la página de SEAyC	Media	5hrs.	3
14	Interesarse en un curso	Baja	4hrs.	3
15	Documentación de la tercera iteración	Alta	6.5hrs	3
16	Enviar notificación	Media	2hrs.	4

17	Documentación de la cuarta iteración	Alta	8hrs.	4
18	Exportar a formato PDF la información de los ponentes	Alta	6hrs.	5
19	Exportar a formato PDF la información de los cursos	Alta	2hrs.	5
20	Documentación de la quinta iteración	Alta	5.75hrs.	5
21	Sexta iteración	Alta	19hrs	6