



MAESTRÍA EN INGENIERÍA DE SOFTWARE

SimSE

**Simulación en el proceso enseñanza-aprendizaje a
nivel superior**

ESTUDIO GUIADO

Investigador líder: Dr. Cuauhtémoc Lemus Olalde

Presenta: Ing. Ma. Estrellita González Radilla

CIMAT

Julio 2010

Contenido

Introducción	4
Antecedentes.....	6
Planteamiento del problema	7
Justificación	9
Objetivos	10
Capítulo I Herramienta SimSE.....	11
1.1 Motivación de la creación de SimSE	11
1.2 Background	12
1.3 Vista general de la herramienta.....	13
Capítulo II Actividades realizadas	19
2.1 Detección de problemática dentro del aula de clases.....	19
2.1.1 Segundo semestre	19
2.1.2 Cuarto semestre	19
2.1.3 Quinto semestre	20
2.1.4 Séptimo semestre.....	20
2.1.8 Octavo semestre.....	21
2.2 Implementación de la herramienta	22
Capítulo III Resultados obtenidos	23
3.1 Beneficios obtenidos de la herramienta.....	23
3.2 Beneficios obtenidos por materia.....	24
3.3 Resultados por parte de los alumnos.....	26
Conclusiones y trabajo futuro	28
Bibliografía	30

Índice de figuras

Figura 1: Pantalla de inicio modelo cascada	15
Figura 2: Pantalla para adquirir herramientas	17
Figura 3: Pantalla de tiempo.....	18
Figura 4: Mapa conceptual de la implementación de SimSE	23

Introducción

"Hay quienes creen que el que las máquinas puedan simular cualquier cosa con tanta fidelidad que pueda percibirse como reales sólo es cuestión de tiempo"

- Dertourzos – (Revorio)

La educación debe renovarse para aprovechar nuevas oportunidades y recursos que actualmente ofrecen los sistemas de información en materia de aprendizaje. El estudiante será capaz no sólo de aprender sino de tomar decisiones, como lo haría en su ambiente de trabajo y aprender de la experiencia, reduciendo el costo de capacitación y riesgos que una empresa tendría al contratarlo si dentro de su formación académica utiliza sistemas expertos como la simulación. (Revorio)

El presente informe está basado en una investigación aplicando la metodología de experimentación, ya que fue realizado con el objetivo de recopilar resultados de los estudiantes de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior Zacatecas Sur (I.T.S.Za.S.) en los diferentes semestres y con diferentes materias, resultados basados en el uso de la simulación como instrumento didáctico en la impartición de clases, específicamente mediante el uso de la herramienta SimSE.

En el primer capítulo se habla de la herramienta SimSE, con qué objetivo se creó, quienes la desarrollaron y se da una breve explicación de cómo funciona, con el fin de que el lector tenga una idea de como es la interfaz entre el usuario y el simulador.

En el segundo capítulo se da una descripción de como se empezó a desarrollar esta investigación, desde la detección de problemática dentro de las aulas de clase hasta la manera de como se realizó el experimento paso por paso.

En el tercer capítulo se muestran los resultados obtenidos de la investigación, esto se dividieron en tres partes: una que muestra un diagrama general de los beneficios obtenidos de la herramienta por semestres y por materias, otra de las

perspectivas son los resultados observados por el docente en los diferentes semestres y por otro lado resultados observados desde el punto de vista del alumno, lo que permite hacer una comparación entre las partes involucradas en el proceso enseñanza-aprendizaje.

Antecedentes

Durante casi 7 años he dado clases en el nivel superior, en la carrera de Ingeniería en Sistemas Computacionales, en esos años he observado una serie de problemáticas en mis alumnos que en el transcurso de su carrera se ve reflejado en el carente uso de trabajar bajo procesos en el desarrollo de Software.

Desde el primer semestre de la carrera se les enseña a los alumnos el ciclo de vida del desarrollo de software, no en una forma detallada pero al menos se les explica el proceso que se debe de seguir a la hora de realizar un proyecto de software. Por desgracia, no se ha encontrado la forma de motivar al alumno a seguir este proceso durante toda la carrera, ellos mismos optan por saltarse las primeras etapas de desarrollo e irse directamente a la programación, lo que implica estar trabajando bajo prueba y error.

Se les explica que el compilador no es un buscador de errores sino un traductor entre sus programas y la computadora, pero ellos no lo ven de esta manera. Desgraciadamente, siguen esta misma forma de trabajar hasta la residencia profesional, donde realizan un cronograma de tiempo el cual está hecho al modo ingenieril, y ellos mismos no se apegan a este calendario, lo que muy a menudo por no decir siempre, los sistemas son entregados fuera de tiempo.

Otra cosa que los alumnos no toman en cuenta es la inserción de defectos, ellos lo ven como algo natural y como algo no costoso, no les toman la importancia que estos tienen, es más ni siquiera les llama la atención “compilar a la primera, probar a la primera”, tienen la falsa idea de que la inserción de defectos es casi obligatoria.

Estos problemas se encuentran desde primer semestre hasta noveno semestre, pues me ha tocado dar clases en los mismos, y hasta la fecha no he encontrado la manera de transmitirles a los alumnos la importancia de seguir un proceso en cualquier proyecto que se emprenda, de planear un calendario y apegarse a él es de mucha importancia.

Planteamiento del problema

Uno de los trabajos más difíciles para los ingenieros en sistemas computacionales es el desarrollo de proyectos de software, pero lo que lo complica aún más es que no se sigue el ciclo de vida del software de una manera adecuada, por lo que los resultados no son los esperados, la mayoría del tiempo no se termina el proyecto en la fecha planeada, los defectos que surgen en pruebas son numerosos, y la ganancia que iba a generar el producto se ve gravemente disminuida, sin contar los incumplimientos de contratos que surgen con el cliente.

Todo esto ha llevado a que surjan nuevas tecnologías o procesos que ayuden a los desarrolladores a disminuir todos los problemas que acarrea el realizar software.

Muchos de los procesos, métricas y disciplinas que se ven en las empresas no son vistas por la academia, por lo que se trata de que los temarios estén lo más actualizados posible. Existe una discrepancia en la impartición de clases, los alumnos no aceptan el proceso de desarrollo de software como tal, están acostumbrados a tener el planteamiento del problema e irse a programar inmediatamente, muchas veces sin entender cuál es el problema a resolver. Esto les ocasiona no entregar sus trabajos a tiempo o pasarse horas probando el sistema y quitando los defectos que insertaron. No es posible hacerles entender teóricamente que gran parte de los problemas que tienen en sus proyectos se podrían solucionar si siguieran el ciclo de vida de forma correcta, haciendo su análisis, diseño, codificación y pruebas.

Es por ello, que se plantea el uso de la herramienta SimSE (Simulation Software Engineering) (SimSE) como apoyo en la impartición de algunas clases de la carrera de ingeniería en sistemas computacionales, para que de esta manera se le muestre al alumno un ambiente de trabajo ilustrativo a la realidad, donde cada defecto insertado en el producto de trabajo cuesta dinero y tiempo en el desarrollo de productos de software. La ventaja del uso de SimSE es que el alumno podrá

vivir de forma simulada los problemas reales que tienen las empresas dedicadas al desarrollo de sistemas de software.

Justificación

Durante los años de experiencia que tengo como docente de nivel superior, se han detectado algunos problemas, los cuales llevaron a elegir este tema de investigación, entre la problemática detectada se encuentran:

- ✚ En el nivel superior, hablando específicamente de la carrera de sistemas, las materias que implican programación no le exigen al alumno seguir el proceso de desarrollo de software, por lo que hacen sus programas codificando directamente, sin diseñar ni mucho menos entender bien los requerimientos, lo que lleva a atrasos en las entregas de proyecto, trabajos con defectos, entre otras.
- ✚ Otro problema detectado, fue que los alumnos no están acostumbrados a documentar el proceso, ni llevar métricas de ningún tipo, esto hasta cierto punto no es tan grave, ya que no se han visto en la necesidad de hacerlo, lo que sí es preocupante es que cuando llevan la materia de PSP, que es precisamente donde se les pide la recolección de métricas, y llevar un proceso documentado, los alumnos toman una actitud negativa en contra del docente, incluso se ha llegado a reclamos fuertes.

Nunca antes se había pensado en utilizar la simulación para aplicarla en el proceso de enseñanza-aprendizaje en la carrera de ISC, por lo que al conocer la herramienta y jugar con ella se empezaron a detectar características del juego que podrían ser útiles dentro de la impartición de algunas clases y sobre todo, podrían ayudar a atacar los problemas antes mencionados.

Objetivos

Utilizar la herramienta SimSE dentro de algunas clases como apoyo en el aprendizaje de los alumnos, tiene 3 objetivos primordiales:

- ✚ Lograr un cambio de actitud en los alumnos que cursan la materia de PSP.
- ✚ Cambiar la forma en que los alumnos desarrollan sus programas, es decir, que sigan un proceso de software determinado.
- ✚ Desarrollar habilidades de líderes de proyectos en los alumnos.

Capítulo I Herramienta SimSE

1.1 Motivación de la creación de SimSE

SimSE (SimSE) es un juego de simulación de ingeniería de software interactivo, gráfico y educacional diseñado para enseñar a los alumnos los procesos de ingeniería de software. (SimSE)

SimSE direcciona el gran vacío que existe en las técnicas tradicionales de ingeniería de software educativo, los alumnos están expuestos a varios conceptos de ingeniería de software y las teorías en las clases, pero tienen pocas oportunidades para poner en práctica estas ideas en un típico proyecto pequeño de ingeniería de software. (SimSE)

SimSE pretende llenar este vacío proporcionando a los alumnos experiencias virtuales de participar en los procesos de ingeniería de software a gran escala. (SimSE)

Uno de los objetivos fundamentales del proyecto SimSE es permitir la personalización de los modelos de procesos de software que simula. (SimSE)

Algunos resultados que se observaron en la utilización de SimSE en la universidad de Irvine fueron los siguientes: (SimSE)

- ✚ Los estudiantes quienes jugaron SimSE aprendieron satisfactoriamente los conceptos para lo que fue diseñada.
- ✚ Los estudiantes encuentran que es una experiencia satisfactoria jugar con SimSE.
- ✚ Algunos estudiantes se frustraban con los resultados obtenidos en el juego, sentían que no habían recibido los temas suficientes en clase como para tener éxito.

- ✚ El proceso de aprendizaje de un jugador de SimSE implica las teorías del aprendizaje por descubrimiento, aprendizaje a través del fracaso, el constructivismo, aprender haciendo, aprendizaje situado y ARCS (Atención, Relevancia, Confianza y Satisfacción) (Keller, 2006) de Keller.
- ✚ SimSE educativo es más efectivo cuando se utiliza como un componente complementario de otros métodos de enseñanza.

1.2 Background

SimSE es un ambiente basado en computadora que facilita la creación y la simulación de los modelos de procesos de software basado en juego-modelos que involucran componentes del mundo real no presentes en la clase típica de proyectos, tales como grandes equipos de personas, hacer decisiones críticas, cuestiones de personal, presupuesto y eventos inesperados.

De este modo, su objetivo es proporcionar a los estudiantes una plataforma a través de la cual se pueden experimentar muchos aspectos diferentes del proceso de software de una manera práctica sin el énfasis en la creación de productos que es inherente en el desarrollo de software real.

El equipo que desarrollo esta herramienta se menciona a continuación:

Lead Developer: Emily Navarro

Supervising Faculty : André van der Hoek

Contributing Developers: Calvin Lee, Ethan Lee

Contributing Model Developers: Alex Baker, Beverly Chan, Terry Mingstun Fog, Barbara Zhu

Artist: Calvin Lee

1.3 Vista general de la herramienta

La herramienta SimSE consta de 7 juegos diseñados:

- a) Modelo incremental
- b) Modelo Cascada
- c) Modelo Prototipo
- d) Juego usando RUP
- e) Juego usando Extreme Programming
- f) Juego de simulación de Inspecciones de Código

Cada uno de los juegos prediseñados ataca diferentes problemas dentro del desarrollo de software, estos obligan al jugador a tomar decisiones para lograr terminar el proyecto a tiempo y bajo el presupuesto dado, el estudiante simula ser líder de proyecto, con requerimientos reales, presupuesto, personal, y tiempo límite para terminar de desarrollar el proyecto propuesto. El juego simula el proceso de desarrollo de software: requerimientos, diseño, código y pruebas. Se insertan defectos en cada etapa, que el jugador deberá quitar conforme pasa el tiempo, el jugador se enfrentara a problemas tales como: ausencia de personal por enfermedad, o por cansancio, a cambios de requerimientos, y sobre todo a la presión del tiempo y presupuesto.

El máximo de puntos a obtener es 100, los cuales se ganan de acuerdo a la administración y estrategias implementadas que haga el jugador para lograr terminar el proyecto en tiempo y presupuesto.

Lo primero que se debe hacer es elegir el juego con el que se quiere empezar a jugar, por ejemplo: supongamos que se elige el ciclo de vida de cascada.

Los requerimientos que pide este juego son:

Crear un sistema basado en Web que permitirá a las personas ordenar por Internet comestibles que serán entregados en sus hogares. El cliente es Grocery Home Delivery Service, una compañía quien hasta ahora solo tomaba ordenes de

comestibles por teléfono, pero ahora quiere dar un paso más en la era de la información.

El presupuesto de este proyecto es 280,000 dólares y un tiempo de 1,350 ticks para completar el proyecto. Sin embargo, se debe estar revisando la información del proyecto- el cliente tiene la tendencia de introducir nuevos requerimientos, y algunas veces dará más tiempo y/o dinero con estos nuevos requerimientos.

El puntaje final será de 100 puntos, y serán calculados en base a código completo y libre de errores, si el código está integrado o no, y que tan bien se apego al presupuesto y calendario.

Dos notas:

- ✚ A cada trabajador contratado se le paga en cada tick de reloj que se marque independientemente de si está ocupado o no. Por lo tanto, júsalos sabiamente!*

- ✚ Si desea utilizar una herramienta que usted ha comprado en una tarea, debe especificar cuando se le puede asignar la tarea. Si ya han empezado la tarea sin la herramienta, usted debe detener y reiniciar la tarea con la herramienta si desea que esta sea utilizada.*



Figura 1: Pantalla de inicio modelo cascada

La Figura 1, muestra la pantalla principal del juego SimSE, simula una oficina donde se encuentran los empleados disponibles en el ciclo de vida cascada. De lado derecho de la imagen aparecen los nombres de los personajes y al asignarles una tarea aparece en esta parte.

El juego contiene:

- a) Artefactos: en esta sección se encuentran los artefactos involucrados en el proyecto como son: requerimientos, diseño, implementación y pruebas. Cuando se da clic en alguno de ellos en la parte inferior aparecen las características:

- ✚ Tipo
- ✚ Nombre
- ✚ Errores conocidos
- ✚ Porcentaje completo

b) Clientes: Aparece la información de nuestro cliente.

c) Empleados: cuando se da un clic en uno de los empleados aparecen en la parte inferior las características de dicho empleado tales como:

- + Tipo de carrera
- + Nombre
- + Energía
- + Experiencia en requerimientos
- + Experiencia en diseño
- + Experiencia en código
- + Experiencia en pruebas
- + Sueldo

d) Proyectos: en la parte inferior aparece la información con respecto al proyecto que se está desarrollando como:

- + Tipo
- + Descripción
- + Presupuesto
- + Dinero gastado
- + Tiempo disponible
- + Tiempo usado

e) Herramientas: muestra un listado de las herramientas disponibles para el proyecto actual, y si se da clic en cualquiera de ellas en la parte inferior de la pantalla aparecen las características:

- + Tipo
- + Nombre
- + Costo
- + Si fue comprada o no

Para empezar a jugar a cada empleado se le debe asignar una tarea que debe cubrir durante el juego, una tarea puede ser cubierta por uno o más empleados. Para asignar las tareas basta con dar un clic derecho al empleado, y en el menú desplegable se elige la tarea que se le quiere asignar.

Cuando se desea comprar una herramienta para ejecutar una de las tareas, basta con dar clic derecho al empleado, elegir *purchase tools* y se desplegará un menú como en la Figura 2, se le da un clic a la herramienta que se quiere adquirir, y el empleado hará su actividad con la herramienta comprada, cabe mencionar que algunas herramientas tienen costo por lo que hay que tomarlas en cuenta en el presupuesto.

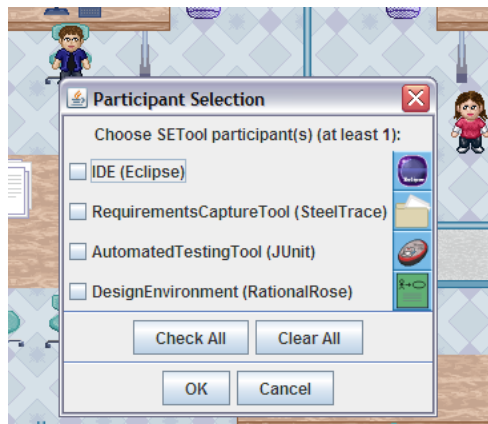


Figura 2: Pantalla para adquirir herramientas

Una vez que se tiene a cada empleado con su(s) tarea(s), se empieza a correr el tiempo. La Figura 3 muestra que hay dos opciones de hacerlo: la primera es por cada tick de reloj, donde se avanza uno por uno, y la otra forma de ejecutarlo es por eventos siguientes, donde el reloj avanza y se detiene cuando surge algún evento como podría ser algún empleado enfermo o cansado.



Figura 3: Pantalla de tiempo

Es importante que el jugador este al pendiente del presupuesto que va gastando, del tiempo que se ha invertido y de los errores que se van insertando en cada etapa del desarrollo.

Al principio podría parecer fácil jugar y terminar todo en tiempo y forma, pero conforme avanza el tiempo y el dinero se va gastando van surgiendo defectos en cada una de las etapas o peor aún se agregan nuevos requerimientos lo que a menudo nos ocasiona más tiempo y dinero, para esto el cliente algunas veces proporciona más dinero o tiempo para cubrir estos requerimientos.

Otra cosa que nos permite el juego de SimSE es ver un análisis de los resultados que se llevan, analizar la energía de cada uno de los empleados, avance de cada una de las etapas de desarrollo, algunas reglas de cada una de las fases, entre otras cosas.

Capítulo II Actividades realizadas

2.1 Detección de problemática dentro del aula de clases

En 7 años de experiencia dando clases a nivel superior se ha detectado una serie de problemática dentro del aula, esta se puede clasificar de la siguiente manera:

2.1.1 Segundo semestre

Programación orientada a objetos, es una de las materias básicas de carrera, el alumno conoce el paradigma orientado a objetos y lo implementa. Los problemas detectados en esta materia son:

- ✚ El alumno no está interesado en comprender los requerimientos del problema planteado.
- ✚ La fase de diseño es algo que no se desarrolla.
- ✚ La programación la ejecutan a prueba y error.

2.1.2 Cuarto semestre

Teoría de la computación, es la materia base para el desarrollo de programas base, como por ejemplo un compilador, en ella se ven autómatas, máquinas de Turing entre otros temas. En esta asignatura se le plantean 4 proyectos de programación al alumno, y se ha observado cierta problemática como:

- ✚ Los proyectos se desarrollan en equipo, los líderes son elegidos por ellos mismos sin contemplar nada para hacerlo.
- ✚ Los alumnos solo se enfocan en el programa, esto es, solo codificar.
- ✚ No se sigue el proceso de análisis, diseño, código y pruebas.

- ✚ Como docente sólo me interesaba la entrega del programa sin documentación.

2.1.3 Quinto semestre

Programación de sistemas, en esta asignatura el alumno conoce e implementa las fases de un compilador (análisis léxico, sintáctico, semántico, generación de código intermedio, optimización y generación de código objeto), aquí los problemas son similares a otras materias.

- ✚ Los proyectos se desarrollan en equipo, los líderes son elegidos por ellos mismos sin contemplar nada para hacerlo.
- ✚ Los alumnos solo se enfocan en el programa, esto es, solo codificar.
- ✚ No se sigue el proceso de análisis, diseño, código y pruebas.
- ✚ Como docente sólo me interesaba la entrega del programa sin documentación.

Análisis y diseño orientado a objetos, en esta asignatura el alumno conoce lo diversos diagramas relacionados con la orientación a objetos, tales como: casos de uso, diagrama de estados, diagrama de secuencia, diagrama de colaboración entre otros.

En esta materia la problemática detectada es:

- ✚ Los alumnos consideran el diseño como una pérdida de tiempo.
- ✚ Se realiza el diseño sin hacer un buen análisis del problema.

2.1.4 Séptimo semestre

Proceso personal de desarrollo de software (PSP), es una materia difícil, no sólo de impartir, sino también al alumno se le dificulta asimilar los procesos que involucrados en PSP, la disciplina que exige y la documentación que contiene.

En esta materia la problemática es aún más notoria:

- ✚ El alumno no comprende la importancia de seguir un proceso.
- ✚ Los estudiantes toman una actitud negativa ante el maestro y la materia.
- ✚ Se molestan porque se les exige hacer la entrega de tareas en tiempo y forma.
- ✚ Les molesta ser penalizados por cada defecto que encuentra el docente en las entregas de tareas.
- ✚ Muestran actitud negativa en la recolección de métricas de tiempo y defectos.

2.1.8 Octavo semestre

Desarrollo de proyectos de software, esta asignatura es la integración de varias materias de la carrera, tales como: modelado de negocios e ingeniería de requerimientos, análisis y diseño orientado a objetos, PSP, fundamentos de ingeniería de software entre otras.

- ✚ Se trabaja por equipos de trabajo donde cada uno toma un rol, este rol es asignado de forma aleatoria sin tomar en cuenta nada.
- ✚ Los equipos de trabajo no tienen una manera de simular la organización de sus tiempos antes de llevarlo a la práctica.

2.2 Implementación de la herramienta

La herramienta se implementó en las materias antes mencionadas con el propósito de resolver completa o parcialmente la problemática detectada en los alumnos, ya que permite simular en un ambiente virtual lo que difícilmente se podría hacer en una empresa.

SimSE se trabajó de la siguiente manera:

- ✚ Se detectó la problemática que se pretendía resolver, esto se hizo por materia.
- ✚ Se platicó con los alumnos que iban a utilizar la herramienta, con qué objetivo se había realizado y que resultados se habían obtenido en la Universidad de Irvine con la implementación de la misma.
- ✚ Se les hizo una presentación a los alumnos una demostración del funcionamiento de la herramienta.
- ✚ Se les explicó de forma breve cada uno de los ciclos de vida, para que de esta manera ellos comprendieran cada uno de los juegos que tiene SimSE.
- ✚ Uno de los principales problemas a los que se enfrentaron los estudiantes con SimSE, es que viene en inglés, por lo que se les concientizó de la importancia de este idioma dentro de la carrera, no sólo para jugar si no para cursar algunas materias y sobre todo, para el campo profesional.
- ✚ Después, dependiendo de la materia fue como se implementó la herramienta, en algunos casos se jugó de forma individual y en otros de manera grupal.
- ✚ En una primera fase, sólo se les pidió jugar, sin documentar nada, esto para despertar interés en los alumnos.
- ✚ Después, en una segunda fase se les pidió a los estudiantes jugar, pero ahora documentando sus estrategias, esto para conocer como se desempeñaba el alumno como líder de un proyecto.
- ✚ Cabe mencionar, que en algunos casos se motivó al alumno a jugar mediante puntos extras en su calificación.

Capítulo III Resultados obtenidos

3.1 Beneficios obtenidos de la herramienta

La Figura 4 muestra de una manera general los beneficios obtenidos con la implementación de la herramienta, se muestran sólo algunos de los problemas detectados, los semestres con las correspondientes materias y por último el beneficio adicional que se descubrió con la herramienta.

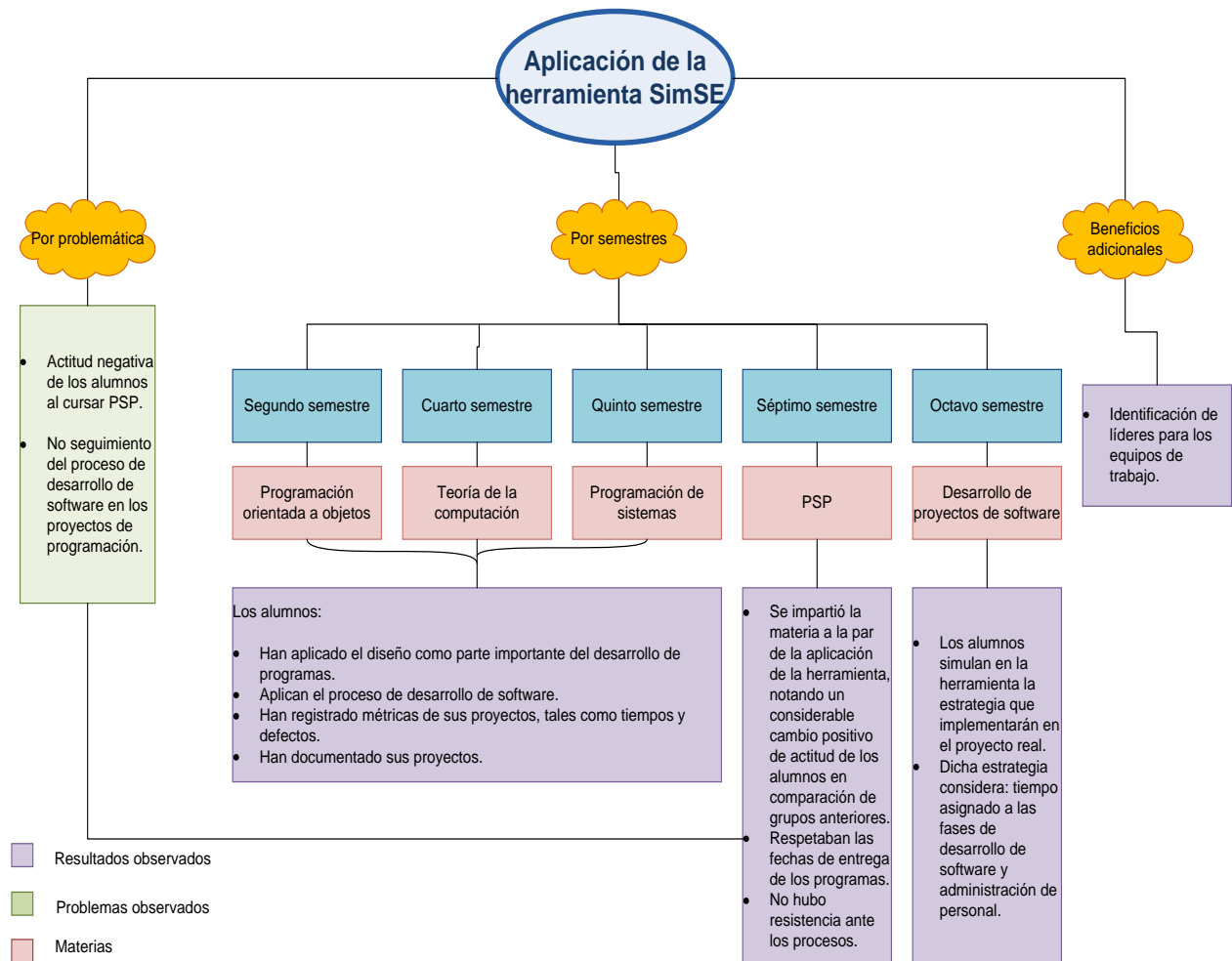


Figura 4: Mapa conceptual de la implementación de SimSE

3.2 Beneficios obtenidos por materia

Después de un año y medio de uso de la herramienta en mis clases, observé los siguientes resultados:

Programación orientada a objetos

- ✚ Los alumnos se preocuparon en entender más a fondo los requerimientos, antes de empezar a desarrollar sus proyectos.
- ✚ Se observó que los alumnos empezaron a implementar el diseño como parte del desarrollo de sus programas.

Teoría de la computación

- ✚ En los equipos de trabajo, los líderes se asignaron de acuerdo al mayor puntaje obtenido en la herramienta, ya no se hizo de forma aleatoria.
- ✚ Los alumnos elaboran el portafolio de evidencia con toda la documentación del proyecto, tales como: análisis, diseño, código, pruebas y métricas recolectadas.
- ✚ Los alumnos reconocen que seguir el proceso de desarrollo y documentarlo es laborioso, pero ven que tiene beneficios, ya que tienen la idea más clara de qué hacer y cómo hacerlo.

Programación de sistemas

- ✚ En los equipos de trabajo, los líderes se asignaron de acuerdo al mayor puntaje obtenido en la herramienta, ya no se hizo de forma aleatoria.
- ✚ Dentro del equipo de trabajo se asignaron roles, y los asignaron de acuerdo a como jugaron en la herramienta, tomando en cuenta experiencia en las fases de desarrollo y sobre todo el gusto del alumno en cada fase.

- ✚ Los alumnos elaboran el portafolio de evidencia con toda la documentación del proyecto, tales como: análisis, diseño, código, pruebas y métricas recolectadas.
- ✚ Los alumnos reconocen que seguir el proceso de desarrollo y documentarlo es laborioso, pero ven que tiene beneficios, ya que tienen la idea más clara de qué hacer y cómo hacerlo.

Análisis y diseño orientado a objetos

- ✚ En los equipos de trabajo, los líderes se asignaron de acuerdo al mayor puntaje obtenido en la herramienta, ya no se hizo de forma aleatoria.
- ✚ Comprendieron que el diseño es importante, pero creen que se le invierte mucho tiempo.

Proceso personal de desarrollo de software

- ✚ Los estudiantes por primera vez no mostraron tanta renuencia a los procesos.
- ✚ Comprendieron de una forma más clara el por qué la recolección de métricas.
- ✚ Los estudiantes se dieron cuenta que si un proyecto no se entrega a tiempo les cuesta dinero y esfuerzo.
- ✚ Se dieron cuenta que los defectos cuestan tiempo y dinero.

Desarrollo de proyectos de software

- ✚ Utilizaron la herramienta para buscar la mejor manera de organizarse en equipo en la vida real, esto en cuanto a roles.

3.3 Resultados por parte de los alumnos

Es importante mencionar los resultados observados no sólo por parte del docente, si no también ver los puntos de vista que tienen los alumnos en cuanto al uso de la herramienta.

Estos resultados son recopilados de alumnos de cuarto y sexto semestre:

- ✚ Es necesario realizar cada una de las fases del desarrollo de software, y que todas sean realizadas en el orden debido para terminar un buen producto e invertir menos tiempo.
- ✚ No es recomendable asignar una fase completa del proceso a una sola persona, ya que esto hace más lento el desarrollo del producto y con más errores.
- ✚ Cuando se dan los requerimientos y se va directo a la codificación se invierte mucho más tiempo en corregir defectos.
- ✚ Es motivador administrar un grupo de personas.
- ✚ Se deben documentar los requerimientos que da el cliente e ir actualizándolos de acuerdo a lo que diga el cliente.
- ✚ Tratar de hacer una buena asignación de tiempo en cada una de las etapas, ya que se podría caer en el error de asignar demasiado tiempo a cierta etapa y dejar a otras escasas del mismo.
- ✚ Se debe de realizar el diseño de cualquier producto de software.
- ✚ Es importante documentar todo: requerimientos, diseño, defectos, ya que todo es importante.
- ✚ Se debe de respetar el tiempo que se asigna al proyecto.
- ✚ Hace falta tener un tutorial de la herramienta.
- ✚ Asignar los líderes de requerimientos, diseño, código y pruebas tomando en cuenta la experiencia de cada integrante, sus gustos en el desarrollo de software entre otras.

- ✚ La herramienta ayuda a vernos como líderes de proyecto y esto es bueno porque si algún día tenemos la oportunidad de dirigir a un equipo desarrollador tendremos una idea de cómo hacerlo.
- ✚ Se empieza a vivir la experiencia de trabajar bajo un tiempo asignado y bajo un presupuesto.
- ✚ Una ventaja importante del uso de la herramienta es que si cometo errores graves no hay problema, en el siguiente juego los evito y así sucesivamente y en la vida real eso no puedo hacerlo.
- ✚ Nos permite conocernos como líderes y saber si podemos serlo en la vida real, en los equipos que se forman en clase.
- ✚ Fue un problema que la herramienta este en inglés.
- ✚ Es frustrante obtener puntuación baja después de varios juegos y cambiando las estrategias de juego.
- ✚ Una vez que se entiende como jugar con la herramienta, es motivador seguir jugando con ella.
- ✚ Resulta enfadoso jugar después de que el cliente cambia los requerimientos una y otra vez, ya que esto afecta todas las etapas del proceso.
- ✚ Estaría bien que la herramienta fuera implementada por toda la carrera, ya que es muy interesante y se aprenden conceptos nuevos de ella.
- ✚ La herramienta debería de implementarse en todas las materias con las que tuviera relación, y tomar en cuenta los resultados sean tomados en cuenta para tomar decisiones en la vida real.

Conclusiones y trabajo futuro

Después de haber implementado la herramienta SimSE durante tres semestres dentro del I.T.S.Za.S., en la carrera de I.S.C., se han observado resultados significativos en los alumnos, al inicio de este proyecto se plantearon 3 objetivos, los cuales se cumplieron.

El cambio de actitud de los alumnos en la materia de PSP fue muy bueno, ya que no se vivieron los problemas que se habían experimentado anteriormente, los estudiantes respondieron bien ante la materia.

Se logró el objetivo de que seguir un proceso en el desarrollo de software trae beneficios, pues ahorra esfuerzo en el desarrollo de los proyectos de las materias.

El tercer objetivo no se cumplió por completo, ya que no se lograba administrar los equipos de trabajo de las materias de la manera ideal, todavía surgían problemas dentro del equipo, lo que implicaba cambios de rol o bien de líderes.

Con esta buena experiencia obtenida en este tiempo con el uso de la simulación para el proceso de enseñanza-aprendizaje, se planea seguir trabajando no sólo con esta herramienta, si no continuar con la búsqueda de juegos que simulen la realidad que tendrán nuestros futuros profesionistas.

El hecho de simular un ambiente real, les dejó buenas enseñanzas a los alumnos, ya que siempre se quejan de no llevar mucha práctica en su vida de estudiante, y estos juegos les permiten vivir por un momento la realidad, sin ser penalizados o sin ser despedidos de un trabajo real.

Como docente también recibí recompensas, me preocupe más por los artefactos entregados por mis alumnos, no sólo por el código del programa sino por lo que había detrás de este, ahora trabajo con ellos los portafolios de evidencias, donde los alumnos documentan todo el proceso que siguieron en la elaboración de los proyectos de las diferentes materias, ahora se les pide entregar un análisis, diseño, código y pruebas, también se les pide recolectar métricas, no tan

detalladas como las de PSP, pero sí, que midan los tiempos de cada fase de desarrollo, y defectos que cometieron en su proyecto.

Fue una satisfacción también escuchar comentarios positivos de los alumnos, tales como: “es bueno diseñar, porque eso nos permite saber cómo vamos a resolver el problema”, “es mucho trabajo documentar, pero vale la pena” o “es bueno trabajar en equipo por asignación de roles, así cada quien toma un papel dentro del proyecto y se hace responsable de coordinarlo”, esto me da pauta para continuar con lo que estoy haciendo, ya que me indica que voy por el camino correcto.

El plan a futuro es seguir trabajando con la herramienta, y ahora será más interesante, ya que en agosto de este año inicia un nuevo modelo educativo, ahora es por competencias, donde el alumno más que memorizar teoría, debe demostrar que tiene los conocimientos que pide cada materia, es decir, se enfoca más a la práctica, más al saber hacer, y la implementación de la herramienta me permitirá que los estudiantes me demuestren ciertos conocimientos, esto la convierte en un buen recurso didáctico.

Además, a partir del siguiente semestre se empezaran a recolectar resultados de los juegos de los alumnos, pero ahora con métricas, y estas ayudaran a dar fundamento a las hipótesis que se planteen en la siguiente etapa de la investigación, ahora se pretende dar un giro estadístico a este proyecto y con esto lograr resultados cuantificables.

Bibliografía

Keller, J. M. (2006). *Keller's ARCS Model of Motivational Design*. Recuperado el 20 de Julio de 2010, de <http://www.arcsmodel.com/>

Revorio, A. J. (s.f.). *Contexto Educativo*. Recuperado el 18 de Julio de 2009, de <http://contexto-educativo.com.ar/2001/3/nota-07.htm>

SimSE, U. o. (s.f.). *An Educational, Game-Based Software Engineering Simulation Environment*. Recuperado el Julio de 2009, de <http://www.ics.uci.edu/~emilyo/SimSE/details.html>