

**AMBIENTE VIRTUAL DE APRENDIZAJE DE APOYO A LA
ENSEÑANZA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS**

Adriana Maritza Angarita Cala

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
LÍNEA DE INVESTIGACIÓN: E-LEARNING PARA EDUCACIÓN EN
INGENIERÍA
BUCARAMANGA
2010**

**AMBIENTE VIRTUAL DE APRENDIZAJE DE APOYO A LA
ENSEÑANZA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS**

ADRIANA MARITZA ANGARITA CALA

Tesis para optar al título de Magister en E-learning

**Director de Proyecto:
Dr. Cesar Darío Guerrero**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
LÍNEA DE INVESTIGACIÓN: E-LEARNING PARA EDUCACIÓN EN
INGENIERÍA
BUCARAMANGA
2010**

CONTENIDO

	Pág
GLOSARIO	5
RESUMEN	6
INTRODUCCIÓN	7
1. PLANTEAMIENTO DEL PROBLEMA	10
1.1. Objetivo General	11
1.2. Objetivos Específicos	12
2. ESTADO DEL ARTE	14
3. DESCRIPCIÓN DE LA EXPERIENCIA	20
3.1. Descripción del Proceso Investigativo	20
4. RESULTADOS ALCANZADOS	25
4.1. Diseño del Módulo	35
4.2. Implementación en Dokeos	43
4.3. Prueba Piloto	61
5. CONCLUSIONES Y RECOMENDACIONES	63
6. REFERENCIAS BIBLIOGRÁFICAS	67
LISTA DE TABLAS	69
LISTA DE FIGURAS	70
ANEXOS	71

GLOSARIO

ABSTRACCIÓN: proceso mental, que permite al individuo comprender un concepto de un objeto. Sin tener al objeto de manera tangible.

ALGORITMO: un algoritmo es una secuencia finita de operaciones realizables, no ambiguas, cuya ejecución da una solución de un problema en un tiempo finito.

BLOGS: bitácora en línea cuyo contenido se forma de los comentarios libres que todos los usuarios le hacen al autor.

COMPUTADOR: sistema digital con tecnología microelectrónica capaz de procesar datos a partir de un grupo de instrucciones denominado programa.

COMUNIDADES VIRTUALES: conjunto de personas que tienen un objetivo en común o un interés de un tema, que por medio de la motivación y la socialización apoyadas en la tecnología permite el intercambio y transferencia de información para la construcción en conocimiento.

E-LEARNING: modalidad de aprendizaje a distancia o virtual, donde se puede interactuar con los docentes por medio de internet. El usuario puede manejar los horarios y es un medio completamente autónomo.

FOROS: servicio de Internet en la que un grupo de personas, que en forma online opinan acerca de una pregunta de un tema específico intercambiando con otras personas opiniones del tema.

LMS: Los sistemas de gestión de aprendizaje (**Learning Management Systems o LMS**) son aplicaciones Web que proveen las funciones administrativas y de seguimiento necesarias para posibilitar y controlar el acceso a los contenidos, implementar recursos de comunicaciones y llevar a cabo el seguimiento de quienes utilizan la herramienta. En general, los LMS facilitan la interacción entre los docentes y los estudiantes, aportan herramientas para la gestión de contenidos académicos y permiten el seguimiento y la evaluación. Es decir, facilitan la “simulación” del modelo real en el mundo virtual, por lo que también se les conoce como Virtual Learning Environment (VLE).

PLATAFORMA EDUCATIVA: herramienta que permite la gestión de los recursos tales como documentos, videos, imágenes y contenidos propios de una asignatura.

PROGRAMACIÓN ORIENTADA A OBJETOS (POO): Es una forma especial de programar, más cercana a como se expresaría en la vida real que otros tipos de programación. Con la POO las cosas se piensan de una manera distinta, en términos de objetos, propiedades y métodos.

REDES SOCIALES: servicio de la red que permite a diferentes grupos de personas desarrollarse e interrelacionarse en el ciberespacio y realizar un intercambio dinámico según el interés que los relacionan o necesidades de información.

UML: definido por sus siglas en inglés, Unified Modeling Language, es un lenguaje de modelado de sistemas de software.

WIKI: sitio en Internet donde se puede elaborar y modificar información relacionada con un tema.

RESUMEN

Este proyecto desarrolló un Ambiente Virtual de Aprendizaje reflejo de la información recolectada en encuestas a estudiantes y docentes, como respuesta a las necesidades de propuestas didácticas e innovadoras acordes con el perfil de nuestros estudiantes actuales.

Se implementó sobre la plataforma Libre DOKEOS por sus características de trabajo colaborativo y cooperativo y con una metodología de acompañamiento a la clase presencial, donde cada tema propuesto es un Objeto de Aprendizaje.

Finalmente se realizó una prueba piloto con un grupo de estudiantes de la UNAB quienes desarrollaron el primer tema y respondieron una encuesta que nos permitió analizar su comportamiento.

Palabras claves: E-learning, Programación, Orientada, Objetos, Dokeos.

Línea de Investigación: E-Learning para educación en ingeniería.

INTRODUCCIÓN

En los últimos años se ha visto una importante reducción a nivel mundial en el ingreso de estudiantes a programas de ingeniería. Diferentes factores inciden en este problema, desde los bajos niveles en matemáticas, la reducción de la inversión en ciencia y matemáticas de algunos gobiernos (EEUU), hasta la percepción de los jóvenes que los trabajos en esta rama se agotarán, ésta última es contraria a lo que los empresarios de muchos países aseguran: para el 2020 habrá 20 millones de empleos Snider¹.

Las matemáticas no son la única razón por la cual los estudiantes no ingresan o abandonan programas de ingeniería. La dificultad en el aprendizaje de la programación, como elemento esencial en la formación de un ingeniero, agrava aún más el problema. Los cursos introductorios de programación de computadoras requieren de estrategias pedagógicas y tecnológicas que atraigan y no generen rechazo en los estudiantes, pero que a su vez fundamenten y logren realmente en ellos sembrar las bases para una buena programación, “La correcta programación no cae del cielo” Jadud². Una vez relacionada la importancia del uso de herramientas tecnológicas adecuadas, la identificación de la programación orientada a objetos como uno de los paradigmas en programación más influyentes en los últimos años Kölling³ y de identificarlo claramente como uno de los temas más álgidos, nace la intención de proponer un mecanismo de enseñanza que pueda apoyar la enseñanza de la Programación Orientada a Objetos.

¹ Jon, S. (n.d). Computer, engineering enrollment up. USA Today,

² Jadud Matthew (2005) A First look at Novice Compilation Behaviour using BlueJ.

³ Kölling Michael (1999) The Problem of teaching object-oriented programming Part I: Languages

Como complemento al objetivo de este proyecto, se busca lograr una perspectiva completamente nueva, la cual incluye las TIC, la Web 2.0 y que busca un aprendizaje totalmente autónomo en el estudiante. De igual forma se espera pueda ser utilizado desde un aula de clase como complemento de encuentros presenciales o como propuesta totalmente en línea con docencia en línea como guía. Esta solución se propone a través de un modelo de e-learning que contribuya a resolver las dificultades expuestas sobre un curso introductorio de programación de computadoras en el tema de Programación Orientada a Objetos y parte de la siguiente hipótesis: Es posible diseñar una estrategia de e-learning aplicada a la enseñanza de la programación de computadoras que contribuya a mejorar los procesos de enseñanza-aprendizaje en ésta área. Ya que han sido vistos avances en donde la diversidad de las propuestas educativas en programación tiene un resultado efectivo en los estudiantes, según estudios realizados por Prucz, Migri, y los resultados entregados por proyectos nacionales de gran trayectoria como el CUPI2 de la universidad de los Andes, quienes han avanzado en la efectividad de estas propuestas con los estudiantes Villalobos⁴, Villalobos et al⁵.

Como resultados se obtuvo un documento con el estado del arte de la enseñanza de la programación Orientada a Objetos, el diseño de un Ambiente Virtual de Aprendizaje implementado para enseñar los temas específico de la programación de computadoras, dos instrumentos de encuesta de entrada y salida, resultados de la evaluación del Ambiente Virtual de Aprendizaje implementado y finalmente un artículo de investigación que dé a conocer los resultados de la investigación y sus aportes al campo de la pedagogía y las TIC.

⁴ Villalobos Jorge (2007) Cómo enseñar a programar: un enfoque efectivo.

⁵ Villalobos Jorge et al (2005) Looking for a new approach to teach/learn a first computer-programming course

Se trabajó un cronograma que permitió cumplir los objetivos en el tiempo propuesto y que respondió a una metodología que contempla tanto la investigación realizada, el desarrollo de la propuesta, la puesta en marcha dentro de un curso real y los resultados obtenidos de ella.

1. PLANTEAMIENTO DEL PROBLEMA

Según Snider⁶ existen diferentes factores que inciden en el bajo ingreso de estudiantes a Ingenierías, desde los bajos niveles en matemáticas, la reducción de la inversión en ciencia y matemáticas de algunos gobiernos (EEUU), hasta la percepción de los jóvenes que los trabajos en esta rama se agotarán, la cual es contraria a lo que los empresarios de muchos países aseguran (para el 2020 habrá 20 millones de empleos). En particular, se ha detectado que los cursos introductorios de programación de computadoras requieren de estrategias pedagógicas y tecnológicas que atraigan y no generen rechazo en los estudiantes. Estas nuevas metodologías necesitarán una preparación por parte de los docentes en un nivel avanzado del uso de las TIC, y la concepción de las competencias pedagógicas como proyectos de formación. De tal forma que además del gusto que se puede generar en el estudiante por la conexión con el uso de TIC, se logre regular un espacio de interacción que contenga componentes básicos que determinarán la naturaleza didáctica del proceso de enseñanza aprendizaje, tales como la motivación del aprendiz, dominio por parte del enseñante de aquello a enseñar, las herramientas para la enseñanza y la cercanía con la realidad como medio de conexión del aprendizaje experimental entre otros Vargas, Gamboa⁷.

Teniendo en cuenta la problemática descrita surge la necesidad de establecer mecanismos que apoyen la enseñanza de la programación de computadoras haciendo su aprendizaje un proceso satisfactorio y motivador para el estudiante y a la vez contribuyendo a reducir la tasa de deserción en los semestres iniciales y a mejorar la imagen de estos programas para atraer más estudiantes, así como la

⁶ Jon, S. (n.d). Computer, engineering enrollment up. USA Today,

⁷ Vargas Germán, Gamboa Sarmiento Sonia Cristina (2008) Didáctica en la condición postmoderna. De las competencias a la cooperación

imperiosa necesidad de incrementar el número de estudiantes de ingeniería para nutrir una nueva generación de científicos e ingenieros tanto en el país como en el mundo.

Este proyecto responde a la pregunta de “¿Cómo mejorar la enseñanza del primer curso de programación de computadoras?” y parte de la siguiente hipótesis: es posible diseñar una estrategia de e-learning aplicada a la enseñanza de la programación de computadoras que contribuya a mejorar los procesos de enseñanza-aprendizaje en ésta área. Ya que han sido vistos avances en donde la diversidad de las propuestas educativas en programación tiene un resultado efectivo en los estudiantes, según estudios realizados por Prucz, Migri, y los resultados entregados por proyectos nacionales de gran trayectoria como el CUIP2 de la universidad de los Andes, quienes han avanzado en la efectividad de estas propuestas con los estudiantes.

Otro soporte a esta idea es la actual aparición de las tecnologías Web 2.0 las cuales han recibido gran acogida dentro de la comunidad de jóvenes, ya que ha logrado acercarlos más a la informática según Jon Swartz, permitiendo entonces crear un espacio propicio para que el e-learning pueda ser parte fundamental de ese acercamiento y terminar logrando “llegar a proveer a nuestros estudiantes currículos que vaya más allá de la programación y represente la imaginación, creatividad, colaboración y carácter complejo de la computación”

Después de entrevistar a una muestra de docentes de las universidades más representativas de la ciudad, se ha podido observar que nuestra hipótesis coincide con la opinión de los docentes en cuanto a las posibles falencias tanto en la enseñanza como la percepción de los estudiantes frente a esta temática donde la metodología es un eje central enfocado en motivación, coherencia con la realidad y la aplicabilidad a la misma. Tal como lo sustenta Salcedo P debe existir una relación entre la psicología del aprendizaje humano para un proceso de desarrollo

computacional, que permita analizar el contexto del estudiante, su entorno y sus necesidades para motivarlo a través de la relación existente entre la aplicación y la abstracción Galán, et al., de una forma efectiva y real ante los estudiantes - 90% creen que existe una relación adecuada entre la parte teórica y la parte práctica del curso, Villalobos J. et al⁸

Se propone entonces la creación de una solución para un aprendizaje activo, con desarrollo incremental de habilidades, actualidad tecnológica, basado en problemas y con equilibrio en los ejes temáticos, Villalobos Jorge⁹. Mediante la planificación y construcción de herramientas que apoyen el aprendizaje de una forma lúdica, jocosa, fresca, muy acorde con la actualidad de los jóvenes y distribuida por niveles que nos permiten garantizar la consecución de las habilidades mental de los estudiantes en cuanto al análisis, la resolución de problemas y comprensión de los conceptos. Estudio realizado por el proyecto Cupi 2¹⁰.

Siguiendo todas estas directrices se logra que la programación sea un tema motivante para estudiantes de diferentes carreras, no tan sólo ingeniería de sistemas, realizando cambios integrales generando un impacto muy positivo en todo el currículo y se garantiza la disminución radical en la deserción o pérdida de materia de los estudiantes (ej. del 43% al 0% en resultados obtenidos por el proyecto Cupi2 de la Universidad de los Andes, Villalobos J.¹¹ y un aumento exponencial en el ingreso de estudiantes a Ingeniería de Sistemas, lo cual desencadenará inevitablemente en un desarrollo importante en nuestro querido país.

⁸ Villalobos Jorge, Casallas Ruby, Marcos Katalina (2005) El reto de diseñar un Curso de Programación de Computadores

⁹ Villalobos Jorge (2007) Cómo enseñar a programar: un enfoque efectivo.

¹⁰ Villalobos Jorge (2007) Cómo enseñar a programar: un enfoque efectivo.

¹¹ Villalobos Jorge (2007) Cómo enseñar a programar: un enfoque efectivo.

1.1. OBJETIVO GENERAL

Implementar y evaluar un mecanismo basado en e-learning que apoye la enseñanza de una temática dentro de un curso introductorio a la programación de computadoras para estudiantes de Ingeniería.

1.2. OBJETIVOS ESPECÍFICOS

- Realizar un estado del arte sobre mecanismos e-learning de apoyo a la enseñanza de la programación de computadoras en Ingeniería.
- Implementar un mecanismo de e-learning para la enseñanza de un concepto abordado dentro de un curso de programación de la Universidad Autónoma de Bucaramanga.
- Determinar la efectividad del mecanismo implementado dentro de un semestre académico, logrando establecer las principales características del proceso enseñanza-aprendizaje mejoradas

2. ESTADO DEL ARTE

La enseñanza de programación ha incursionado en el campo del e-learning desde diferentes aspectos y utilizando diferentes metodologías, a continuación se menciona algunos ejemplos de los avances en este tema a nivel mundial:

Según Misoko¹² se propone aplicar la enseñanza basada en Web principalmente por la personalización del aprendizaje como medio para lograr avances en el estudio de programación utilizando el método de “ver a otros”. Para ello se crearon herramientas donde el estudiante crea un código y recibe retroalimentación por parte del docente a través de esa misma herramienta, permitiéndole producir código con el reconocimiento de sus errores y autónomamente con la guía y acompañamiento de su docente que comenta y ofrece medios para que el estudiante profundice. Se obtuvo como resultado el incremento del aprendizaje en los estudiantes. Este estudio nos muestra una necesidad solventada en programación con excelentes resultados utilizando el e-learning, por ello comprueba nuestra hipótesis inicial y objetivo de este estudio.

En la Universidad de Tecnología de Queensland en su Facultad de Tecnologías de Información, Nghi et al¹³ trabajó sobre el Ambiente para aprender a programar (EPL), el cual provee un ambiente interactivo basado en web para la enseñanza de programación en los primeros años, a través de un activo, colaborativo, en línea y constructivo ambiente para ayudar a los estudiantes de tecnología para programar exitosamente. La Universidad ha hecho más fácil el trabajo colaborativo con beneficios tanto para docentes como para estudiantes favoreciendo la retroalimentación, la práctica y permitiendo a los estudiantes progresar a su propio

¹² Heo Misoko (2003) A Learning and Assessment Tool for Web-based Distributed Education.

¹³ Truong Nghi, Bancroft Peter y Roe Paul (2003) A Web Based Environment for Learning to Program

ritmo. EPL permite la integración sencilla de notas, tutoriales y ejercicios de práctica, con posibilidad de tutoría en cualquier momento y en cualquier lugar y la cantidad que necesite. Este estudio muestra la fortaleza de la educación basada en web solucionando varios inconvenientes que suelen presentarse en el momento de enseñanza en el aula de la programación en Java, es un gran ejemplo a seguir por su éxito, objetivo acorde al nuestro y proyección.

Por otro lado, ha habido avances en la Universidad de Regina en el Departamento de Ciencias Computacionales con la creación de un Sistema tutorial Inteligente basado en Web para la programación de computadoras que ha llegado inclusive a no ser solamente una herramienta conductual, sino realmente inteligente ya que puede ayudar al estudiante a navegar a través del material en línea, recomendando metas de aprendizaje y generando secuencias de lectura apropiadas logrando así hacer muy eficiente y activo el aprendizaje. Un hecho que llevó a la realización de este proyecto y que precisamente coincide con nuestra hipótesis inicial y causa de nuestro estudio, es que los estudios empíricos muestran que la tutoría individual uno a uno es la forma más efectiva de enseñanza y aprendizaje; es una solución bien lograda que nos muestra cómo se puede abordar la enseñanza de la programación en e-learning de una forma efectiva.

Atravesando el globo nuestra problemática alcanza lugares como Taiwán, donde han unido esfuerzos 4 universidades formalizando un estudio Wu-Yuin et al¹⁴ para ofrecer una solución basada en Web llamada “WPAS” que ofrece 5 actividades de programación con varios niveles de dificultad basados en la taxonomía cognitiva de Bloom, proveyendo en línea: código, ejecuciones y herramientas para anotaciones, para conducir el entrenamiento y asignaciones a través de programación basada en Web. Nuevamente se ofrecen estadísticas muy satisfactorias de la aplicación de este

¹⁴ Hwang Wu-Yuin et al (2008) A web-based programming learning environment to support cognitive development

sistema. Es un estudio muy acorde a lo que busca este proyecto ya que se aborda no sólo una herramienta sino una metodología completa a través de actividades diseñadas para la Web, así como las expectativas que se generaron en los diferentes campos después de los resultados del proyecto.

En el Departamento de Ciencias Computacionales de la Universidad de Warwick, Jirarat y Mike¹⁵ se realizó un estudio cuyos resultados y conclusiones ofrecen un aporte muy acordes a los objetivos trazados en la construcción de la solución que se ofrecere en nuestro proyecto, ya que además de ser exitoso, se basa es una característica del trabajo colaborativo del e-learning y es la coevaluación realizada por los pares (estudiantes). Se ha encontrado que la crítica constructiva frente al trabajo de otros, motiva a los estudiantes a realizar mejores esfuerzos y permite corregir los errores propios viéndolos en otros trabajos. También presenta algunas características por mejorar, como la posibilidad de que afecten subjetivismos en los estudiantes a la hora de evaluar el trabajo de otros.

Los estudios revelan que los estudiantes pueden aprender a programar a través de 5 diferentes formas: Siguiendo, donde los estudiantes aprenden experimentando a través del seguimiento de una unidad, Generando Código: donde el aprendizaje de la programación se experimenta a través del aprendizaje de código, Entendiendo e integrando, donde el acto de programar se experimenta a través de el entendimiento y la integración de conceptos, Resolución de Problemas, donde se experimenta a través de aprender lo que se necesita para resolver un problema y Participando,

¹⁵ Sitthiworachart Jirarat y Joy Mike. (2003) Web-based Peer Assessment in Learning Computer Programming

donde se experimenta descubriendo lo que es un programador. Bruce Christine et al¹⁶

Ahora bien, teniendo claros los esfuerzos en cuanto a la mejora de la programación, es necesario conocer los avances en nuestro tema más particular: La Programación Orientada a Objetos (POO). En este tema se encuentra que presenta algunas dificultades para lograr su aprendizaje tales como desarrollar una estrategia para descomponer un problema, entender los mecanismos de programación; paso de parámetros, valores retornados de métodos y declaración de variables, emplear técnicas de programación abstractas en diferentes situaciones y finalmente entender y emplear conceptos de programación orientada a objetos tales como Herencia, abstracción, clases concretas e interfaces Keefe¹⁷.

Otra conclusión encontrada es que los problemas en este tema se deben a escogencia de lenguajes y medios de enseñanza equivocados. Kölling¹⁸. Un ambiente adecuado deberá tener siete aspectos claves: fácil uso, herramientas intregradas, Soporte de objetos, soporte para reuso de código, Soporte de aprendizaje, grupos de soporte, disponibilidad Kölling¹⁹.

De igual forma en nuestro tema en particular la enseñanza de la Programación Orientada a Objetos, existen diferentes estrategias y metodologías que han sido

¹⁶ Bruce Christine, Buckingham Lawrence, Hynd John, McMahon Camille, Roggenkamp Mike, Stoodley Ian (2004) Ways of Experiencing the Act of Learning to Program: A Phenomenographic Study of Introductory Programming Students at University.

¹⁷ Keefe Karen et al (2006) Adopting XP Practices for Teaching Object Oriented Programming

¹⁸ Kölling Michael (1999) The Problem of teaching object-oriented programming Part I: Languages

¹⁹ Kölling Michael (1999) The Problem of teaching object-oriented programming Part II: Languages

implementadas a través de estos años, logrando un porcentaje de efectividad que nos obliga a tomarlas en cuenta para elaborar el método objetivo de este estudio. Uno de ellos es el llamado “Object-first” el cual consiste en introducir a los estudiantes los conceptos, clases e instancias antes de iniciar en los elementos procedimentales de un lenguaje de programación Fang²⁰.

Adicionalmente se concluye que se ha tomado muchas veces una visión reduccionista en lugar de una holística frente al desarrollo lo cual conlleva a incluir nociones y conceptos gradualmente en medio del aprendizaje en lugar de tener una perspectiva integrada desde un comienzo. Börstler²¹.

Ahora bien adicional a estos avances es importante tener en cuenta igualmente que las condiciones para lograr ser efectivos y dado el uso de estas nuevas metodologías, los impartidores de ellas necesitan una preparación en un nivel avanzado del uso de las TIC, y la concepción de las competencias pedagógicas como proyectos de formación. Para así lograr regular un espacio de interacción que contenga componentes básicos que determinarán la naturaleza didáctica del proceso de enseñanza aprendizaje, tales como la motivación del aprendiz, dominio por parte del enseñante de aquello a enseñar, las herramientas para la enseñanza y la cercanía con la realidad como medio de conexión del aprendizaje experimental entre otros Vargas, Gamboa²².

Un estudio muy interesante relacionado con el tema visto que nos proporciona varias conclusiones muy apropiadas dada la naturaleza del mismo: Contexto Colombiano,

²⁰ Wee fang et al (2005) A student model for object-oriented design and Programming.

²¹ Börstler Jürgen and Sharp Helen (2003) Learning and Teaching Objetc Technology

²² Vargas Germán, Gamboa Sarmiento Sonia Cristina (2008) Didáctica en la condición postmoderna. De las competencias a la cooperación

análisis y diseño de un Primer Curso de Programación de Computadores y con la temática de la Programación Orientada a Objetos es el desarrollado por el proyecto CUIP2. Este proyecto consistió en un estudio inicial de todas las experiencias de la Universidad hasta el momento en sus cursos de programación, las conclusiones frente a las estrategias aplicadas hasta ahora y el estudio de los posibles factores por las cuales en algunos grupos puede ser más exitosos que en otros Villalobos et al²³.

Se tiene entonces múltiples apoyo de las herramientas y medios utilizados en línea por diferentes universidades, con sus conclusiones y éxitos para a partir de ellos vincular estas experiencias en nuestro proyecto y así lograr nuestro objetivo primordial.

Se parte de la hipótesis que dada la recepción planteada por los estudiantes de cuanto a herramientas Web 2.0 y los ambientes multimediales, que una solución planteada a través de educación virtual podría tener muy buena recepción en ellos y por ende efectos positivos en el proceso de enseñanza-aprendizaje.

²³ Villalobos Jorge et al (2005) Looking for a new approach to teach/learn a first computer-programming course

3. DESCRIPCIÓN DE LA EXPERIENCIA

3.1. Descripción del proceso Investigativo

En coherencia con los objetivos propuestos en la realización de este proyecto y teniendo como meta ofrecer un producto muy cercano a la solución del problema, se realizó una investigación a fondo de la problemática de la enseñanza de la programación de computadoras. Teniendo en mente la temática y línea de este proyecto: la enseñanza a través del e-learning, se inicia con la revisión del estado del arte de la enseñanza de programación, para así identificar los avances obtenidos en esta materia por grupos de investigación existentes. Gracias a esta búsqueda se pudo establecer contacto con diferentes Universidades principalmente nacionales que han dado pasos en este tema, tales como: La Universidad de los Andes, quienes tienen constituido un grupo llamado CUIP2 y han realizado grandes avances con ponencias a nivel nacional e internacional, en cabeza del Dr. Jorge Villalobos, así como la Universidad Francisco de Paula Santander a través de su grupo GIDIS quienes han avanzado en el desarrollo de ambientes apoyado en TIC para el aprendizaje de fundamentos de programación.

3.2. Diagnóstico Actual

Analizando toda la información recolectada, se decidió entrar de lleno en materia a través de un mecanismo que permitiera conocer de forma anónima la opinión de los estudiantes de ingenierías que han culminado su primer semestre y por ende su primer contacto con la materia de programación.

Para ello se evaluó algunos sistemas en línea analizando sus pros y contras y se decidió utilizar la herramienta para encuestas LimeSurvey. De modo que la muestra fuera muy cercana a la realidad se hizo contactos con las Universidades más

representativas de la zona en que se encuentra ubicada la sede de la investigación, entre las cuales se mencionan: Universidad Autónoma de Bucaramanga, Universidad de Santander, Universidad Pontificia Bolivariana y Universidad Industrial de Santander.

Se realizó la invitación a los principales contactos en las Universidades mencionadas, para que sus estudiantes ingresan al vínculo: <http://elearning.limequery.com/index.php?sid=33729&lang=es> (Ver Anexo1) donde podrían diligenciar la encuesta publicada a través del sistema escogido: Limesurvey, la cual contenía preguntas que nos permitieran conocer su experiencia y qué percepción tenían de los temas vistos, entre otros se puede mencionar:

- La experiencia anterior al curso en cuanto a la programación
- Experiencias laborales o personales diferentes a cursos
- Utilidad que el estudiante ve frente al tema de programación
- Grado de complejidad de algunos temas principales
- Importancia de las estrategias empleadas
- Evaluación de los aspectos del curso fundamentos de programación.

Se mantuvo abierta la encuesta durante aproximadamente 20 días al estudiantado y se obtuvo la respuesta de 50 personas, quienes la diligenciaron completamente, ofreciendo resultados que permitieron vislumbrar más claramente los puntos de mayor necesidad.

Dado que el proceso de enseñanza aprendizaje está dado entre mínimo dos actores: quien enseña y quien aprende, se decidió investigar qué opinaban los docentes que impartían esta asignatura, a través de su experiencia y conocimiento disciplinar, y cuál era su perspectiva frente al problema en cuestión. (Ver Anexo2).

Para ello se invitó nuevamente a docentes de las universidades mencionadas y de otras ubicadas fuera del área a compartir su experiencia con nosotros. Se contó con el apoyo de 7 catedráticos quienes muy amablemente resolvieron las inquietudes principales que estaban dirigidas a dos puntos en particular: los aspectos que consideran hacen que la enseñanza de programación sea compleja y los apoyos y estrategias que son usadas por ellos para la enseñanza de la misma. Se obtuvo respuestas variadas donde se destacan aspectos como la importancia entre la relación de la carrera que el estudiante curse con su motivación y entusiasmo por la programación, debido a la utilidad que el aprendiente encuentra de lo que está aprendiendo. De igual forma la necesidad de unos conocimientos básicos preexistentes a la hora de ingresar al primer curso de programación.

De este trabajo de investigación conjunto tanto disciplinar como de experiencias, se obtuvo material muy interesante que nos permite orientar nuestros pasos hacia el diseño de una metodología efectiva a implementar la cual debe ser enfocada en el estudiante, fomentando un trabajo autónomo y auto regulado por parte de él, con el apoyo de herramientas TIC y la Web 2.0; tomando como principal insumo la información proporcionada por los estudiantes sobre cómo consideran podría facilitarse su aprendizaje y lo que los docentes consideran se debe reforzar para mejorar la enseñanza.

Se derivó la escogencia de una plataforma que tuviera características como flexibilidad, facilidad, sencillez, gratuita y robusta, calificando DOKEOS como la más adecuada. Entonces se diseñó un curso de acuerdo al perfil del estudiante de ingreso, al perfil del estudiante de egreso, a la temática y aplicando las herramientas Web 2.0.

Consecutivamente se procedió a efectuar una prueba piloto con un grupo de estudiantes de Ingenierías en compañía de la docente del módulo Fundamentos de

Programación, donde ellos participarían en el desarrollo del primer tema y posteriormente calificarían a través de una encuesta su participación y aprendizaje con el uso de la herramienta desarrollada.

Se toman los resultados de la encuesta y se sintetizan las posibles conclusiones con respecto del trabajo actual y la proyección del trabajo futuro a realizarse.

4. RESULTADOS ALCANZADOS

Uno de los primeros resultados obtenidos se logró recopilar a través de una encuesta en línea, con las respuestas de 50 estudiantes quienes la completaron, ofreciéndonos resultados así:

1. ¿Qué carrera cursa actualmente?

Opción	Porcentaje
Sistemas	52.38%
Mecatrónica	7.14%
Mercados	9.52%
Financiera	11.90%
Energía	4.76%
Civil	9.52%
Ambiental	4.76%

Tabla 1 Porcentajes de respuesta a pregunta 1 Encuesta levantamiento información inicial

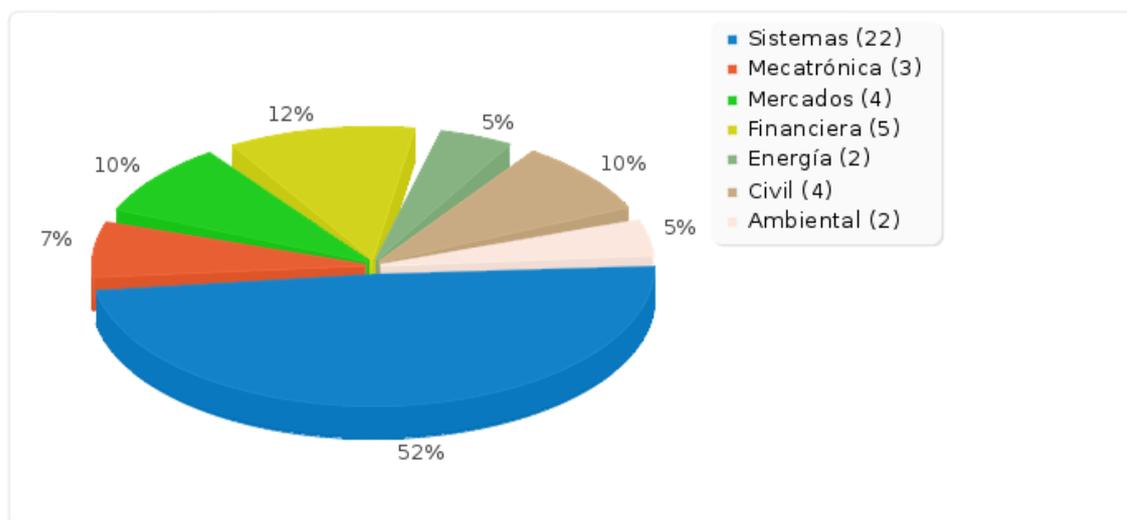


Gráfico1 Porcentajes Respuesta pregunta 1 Encuesta información inicial

2. Antes de ingresar a la universidad, en cuántos cursos de programación ha participado?

Opción	Porcentaje
0	83.33%
1	9.52%
2 o más	7.14%

Tabla 2 Porcentajes de respuesta a pregunta 2 Encuesta levantamiento información inicial

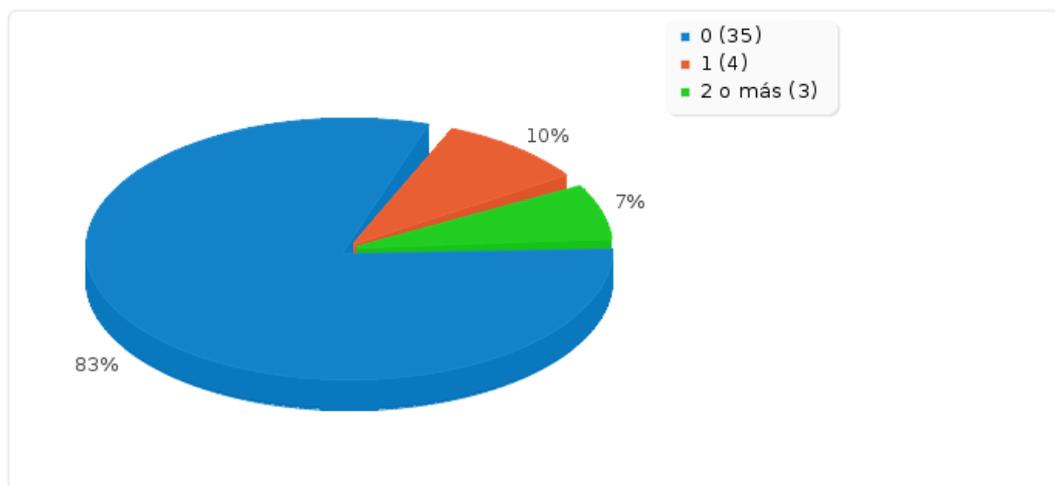


Gráfico2 Porcentajes Respuesta pregunta 2 Encuesta información inicial

3. ¿Ha tenido experiencia de tipo laboral o de interés personal con la programación de computadoras, diferente a cursos de programación?

Opción	Porcentaje
Sí	16.67%
No	83.33%

Tabla 3 Porcentajes de respuesta a pregunta 3 Encuesta levantamiento información inicial

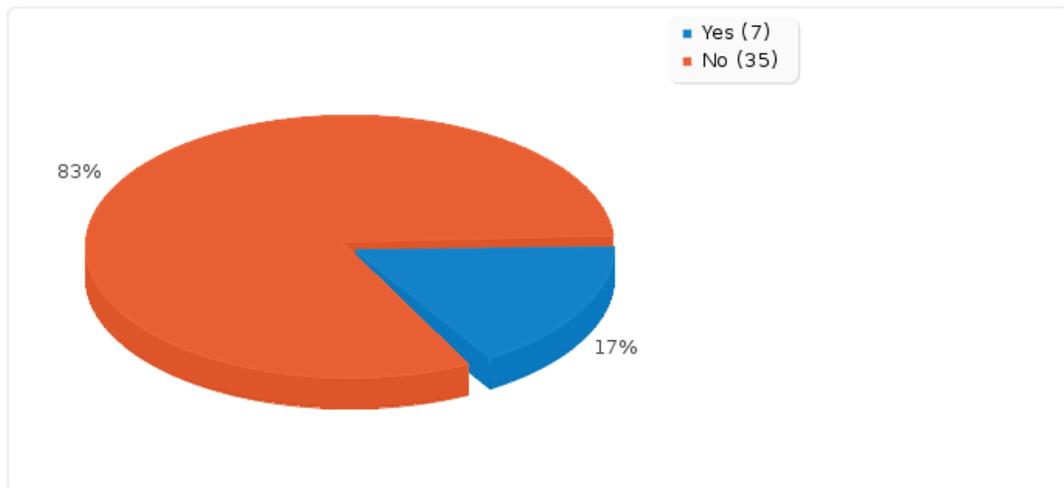


Gráfico3 Porcentajes Respuesta pregunta 3 Encuesta información inicial

4. ¿Considera que aprender a programar es útil para su profesión?

Opción	Porcentaje
Yes	85.71%
No	14.29%

Tabla 4 Porcentajes de respuesta a pregunta 4 Encuesta levantamiento información inicial

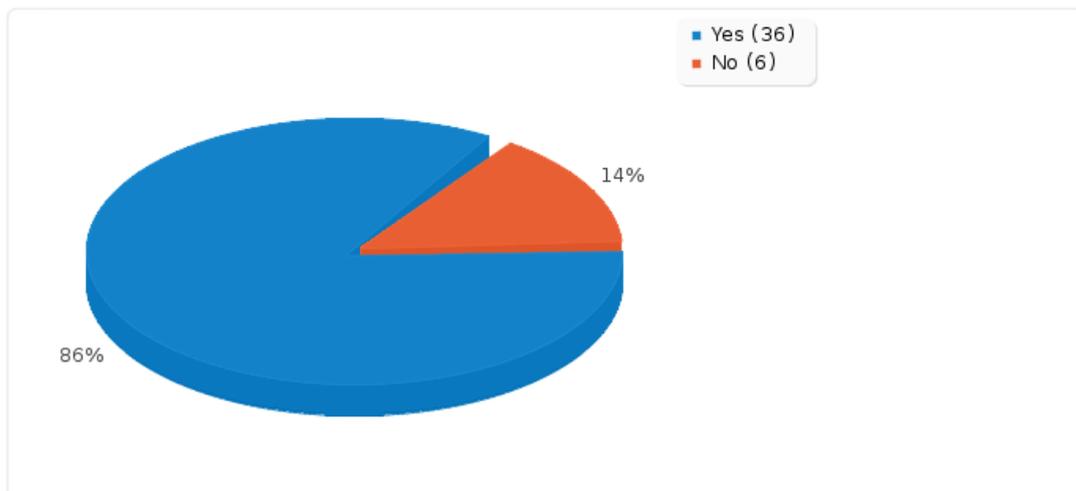


Gráfico4 Porcentajes Respuesta pregunta 4 Encuesta información inicial

5. Durante su curso de “Fundamentos de Programación” usted aprendió a: (puede seleccionar varias)

Opción	Porcentaje
Entender y analizar un problema	78.57%
Representar la solución de un problema a través de un algoritmo o diagrama de flujo	66.67%
Escribir y editar programas en una computadora	61.90%
Compilar y ejecutar programas en una computadora	64.29%
Depurar o mejorar sus programas	35.71%

Tabla 5 Porcentajes de respuesta a pregunta 5 Encuesta levantamiento información inicial

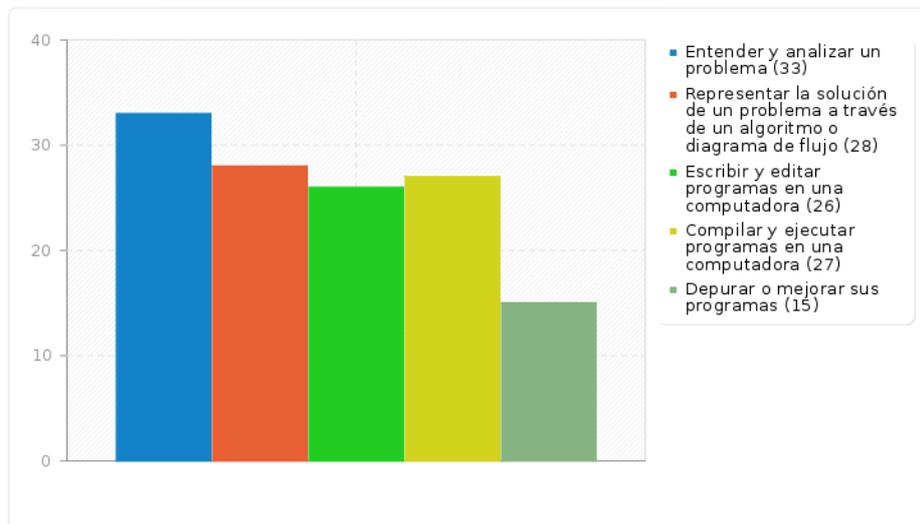


Gráfico5 Porcentajes Respuesta pregunta 5 Encuesta información inicial

6. Indique el grado de complejidad que para usted tuvo cada uno de los siguientes temas del curso de “Fundamentos de Programación.”

Tema	Muy Bajo	Bajo	Medio	Alto	Muy Alto
------	----------	------	-------	------	----------

Concepto de Algoritmo	28.57%	21.43%	30.95%	7.14%	11.90%
Construir algoritmos o diagramas de flujo	21.43%	16.67%	33.33%	14.29%	14.29%
Operadores relacionales, aritméticos y lógicos	14.29%	33.33%	35.71%	7.14%	9.52%
Estructuras de Selección Simple, Doble y Anidamiento	11.90%	21.43%	26.19%	23.81%	16.67%
Estructuras de repetición	16.67%	16.67%	35.71%	11.90%	19.05%
Sistemas de numeración y conversión	7.14%	23.81%	30.95%	14.29%	23.81%
Introducción a Java	9.52%	11.90%	35.71%	14.29%	28.57%
Constructor y Definición	11.90%	11.90%	42.86%	4.76%	28.57%
Concepto y utilización de métodos	7.14%	11.90%	42.86%	9.52%	28.57%
Operaciones básicas con arreglos	9.52%	9.52%	35.71%	16.67%	28.57%
Orientación a Objetos	9.52%	11.90%	26.19%	19.05%	33.33%
Paso de arreglos a métodos	7.14%	11.90%	28.57%	16.67%	35.71%

Tabla 6 Porcentajes de respuesta a pregunta 6 Encuesta levantamiento información inicial

7. Evalúe cada uno de los siguientes aspectos referidos a su curso de “Fundamentos de Programación

Tema	Muy Bajo	Bajo	Medio	Alto	Muy Alto
Motivación	16.67%	9.52%	30.95%	19.05%	23.81%
Número de horas Prácticas	9.52%	14.29%	28.57%	28.57%	19.05%
Cantidad de temas cubiertos	7.14%	4.76%	38.10%	30.95%	19.05%
Claridad en la exposición de los temas por parte del profesor	9.52%	16.67%	19.05%	28.57%	26.19%
Uso de las herramientas de enseñanza adecuadas para el aprendizaje de los temas del curso	7.14%	16.67%	16.67%	38.10%	21.43%

Tabla 7 Porcentajes de respuesta a pregunta 7 Encuesta levantamiento información inicial

8. Indique el grado de importancia de las siguientes estrategias para mejorar la enseñanza de la programación de computadoras

Tema	Muy Bajo	Bajo	Medio	Alto	Muy Alto
Ejercicios en el tablero	4.76%	4.76%	30.95%	16.67%	42.86%
Exámenes prácticos	0.00%	9.52%	26.19%	23.81%	40.48%
Cuestionarios en línea (por internet)	9.52%	16.67%	30.95%	21.43%	21.43%
Tutoriales en línea (por internet)	0.00%	28.57%	9.52%	30.95%	30.95%
Programar en equipos de 2 estudiantes	14.29%	19.05%	16.67%	21.43%	28.57%
Tutorías con un estudiante de semestres superiores	9.52%	9.52%	26.19%	26.19%	28.57%
Horas de consulta del docente Alice	2.38%	4.76%	9.52%	35.71%	47.62%
Videos sobre programación	48.78%	12.20%	14.63%	12.20%	12.20%
Apoyo a través de monitores	4.76%	26.19%	26.19%	14.29%	28.57%
Trabajar con Proyectos prácticos en clase a través de todo el curso	2.38%	9.52%	23.81%	26.19%	38.10%
Análisis y estudio de software ya realizado	0.00%	4.76%	19.05%	14.29%	61.90%
	0.00%	2.38%	23.81%	21.43%	52.38%

Tabla 8 Porcentajes de respuesta a pregunta 8 Encuesta levantamiento información inicial

9. ¿Qué sugeriría usted para mejorar la enseñanza de la programación en la Universidad?

- **Bases más solidas con respecto a conocimientos para usar operadores.**
- **Claridad docente.**
- **Co-ayuda a estudiantes a través de proyectos.**
- **Resolución de ejercicios por estudiantes con acompañamiento paso a paso del docente.**
- **Más clases más prácticas**
- **Uso de aplicaciones interactivas.**
- **Profundizar más en los temas vistos en clase a través de un nivel más.**

<ul style="list-style-type: none"> • Cursos de pre-programación.
<ul style="list-style-type: none"> • Tutorías de estudiantes de niveles superiores a estudiantes de niveles inferiores.
<ul style="list-style-type: none"> • Más material didáctico que permita al estudiante estudiar por cuenta propia.
<ul style="list-style-type: none"> • Mejor preparación pedagógica de profesores.
<ul style="list-style-type: none"> • El Uso de métodos más claros y didácticos a la hora de explicar los temas básicos en la Programación.
<ul style="list-style-type: none"> • Sistemas e imágenes más modernas, lo fresco y fácil de usar, optar por ser más sencillo y dinámico.
<ul style="list-style-type: none"> • Variedad de material informativo.
<ul style="list-style-type: none"> • Clases más motivadas.
<ul style="list-style-type: none"> • Mayor intensidad horaria.
<ul style="list-style-type: none"> • Hardware y Software adecuados.
<ul style="list-style-type: none"> • Actividades extra clase.
<ul style="list-style-type: none"> • Tutoría en plataformas virtuales con variedad de ejercicios en java, con foros de respuestas a preguntas frecuentes.
<ul style="list-style-type: none"> • Grado de dificultad e intensidad horaria de acuerdo a la carrera.
<ul style="list-style-type: none"> • Software más avanzado: Land survey y 3d
<ul style="list-style-type: none"> • Menor cantidad de alumnos en las aulas, logrando así un ambiente más personalizado.
<ul style="list-style-type: none"> • Aplicabilidad a través de ejercicios de acuerdo a la carrera para lograr un sentido del aprendizaje en el estudiante.

Tabla 9 Porcentajes de respuesta a pregunta 9 Encuesta levantamiento información inicial

De acuerdo a estas respuestas se postulan las siguientes afirmaciones:

- Más del 50% de estudiantes que resolvió la encuesta fueron de Ingeniería de Sistemas
- La gran mayoría de estudiantes que ingresan a este curso no han visto nunca antes nada relacionado a programación.

- El 90% de los estudiantes no ha tenido experiencia laboral ni por interés personal en programación.
- El 90% cree es útil el aprender programación para su profesión
- Los estudiantes consideran que a través de su curso de programación principalmente aprendieron a entender y resolver problemas, seguido por la representación de una solución de un problema a través de un algoritmo o diagrama de flujo, y a escribir, editar, compilar y ejecutar programas en una computadora.
- El tema que representó mayor dificultad fue el paso de arreglos a métodos seguido de la programación orientada a objetos, el concepto y utilización de métodos, las operaciones básicas con arreglos, constructores y definición y la introducción a Java.
- Al evaluar los aspectos del curso de programación, se encuentran relativamente satisfechos con las herramientas usadas al enseñar y con la claridad en la exposición de los temas por parte del profesor, pero no tanto con la cantidad de temas cubiertos en la asignatura y muy poco con el número de horas prácticas y la motivación en la asignatura.
- En la concepción con respecto al grado de importancia de las estrategias para mejorar la enseñanza los estudiantes primordialmente plantean la necesidad de trabajar con proyectos prácticos en clase a través de todo el curso como principal medida, seguido de análisis y estudio de software ya realizado, más horas de consulta con los docentes, mayor cantidad de ejercicios en el tablero, apoyo de monitores y exámenes prácticos, entre otros.
- Entre las mejoras a sugerir se esbozan temas como herramientas adecuadas a las necesidades, menor cantidad de estudiantes procurando así una enseñanza un poco más personalizada, aplicaciones interactivas, mas exigencia a los alumnos por parte de los docentes, variedad de material, actividades extraclases, virtuales y autónomas tratando de enfocarse en la utilidad que en cada carrera dará el estudiante a este tema.

Tratando de contar con un punto de vista objetivo y coherente se aplicó una encuesta a docentes de programación para conocer su punto de vista con respecto a las estrategias pedagógicas y didácticas más eficientes y su opinión frente al comportamiento de los estudiantes en esta etapa. Se entrevistaron 7 docentes y de sus respuestas (Ver anexo 3) se puede enunciar las siguientes premisas:

- El 60% de los docentes encuentra que la falta de un abordaje mínimo por parte de los estudiantes en sus niveles anteriores no permite avanzar pronta y eficazmente.
- De igual forma es claro para un 80% de los docentes que si al no haber una relación directa entre la disciplina de algunas ingenierías con este curso, se pierde la conexión aplicativa que permitiría al estudiante encontrarle una utilidad importante a la asignatura y a los conceptos allí enseñados, lo cual es clave para la apropiación del conocimiento.
- El pensamiento para el manejo de soluciones de un modo algorítmico debe ser inculcado y trabajado tanto en la asignatura como en el anterior nivel cursado por el estudiante, es una conclusión importante de más del 50% de los docentes entrevistados.
- Otra propuesta es la utilización de Diagramas de Flujo de Datos para la representación gráfica de algoritmos facilita el aprendizaje a partir de la diagramación, la cual es una forma más fácil de entender que el mismo pseudocódigo.
- La importancia de las competencias iniciales del estudiante tanto en lógica, pensamiento algorítmico, análisis y abstracción, así como en comprensión lectora y matemáticas.
- Es substancial hacer énfasis en la metodología para la resolución de problemas y el diseño del algoritmo dentro de los cursos de programación.

Estas conclusiones de las dos encuestas, los contactos realizados con los diferentes grupos mencionados y la investigación a nivel mundial realizada, nos permitieron tener un panorama más claro sobre el abordaje a este tema para el desarrollo de esta investigación, lo cual nos lleva a precisar nuestro resultado final y objetivo inicial de la misma: una propuesta metodológica que apoye el proceso enseñanza-aprendizaje de la Programación Orientada a Objetos.

4.1. DISEÑO DE LA HERRAMIENTA DIDÁCTICA PARA LA ENSEÑANZA DE PROGRAMACIÓN ORIENTADA A OBJETOS.

Para iniciar con el planteamiento de un diseño efectivo se parte de la definición clara del contexto sobre el cual se realiza esta propuesta de herramienta, de modo que se parta de un escenario real que nos permita lograr efectividad.

Una de los datos más importantes a tener en cuenta en la producción de una herramienta de enseñanza, es la identificación clara del usuario final quien utilizará nuestro producto, para nuestro caso se definen las Características del usuario teniendo en cuenta el perfil de nuestros estudiantes que tiene diferentes ventajas que deberán ser aprovechadas dentro del marco de desarrollo de nuestra propuesta.

Se propone una herramienta para estudiantes posiblemente nativos digitales para quienes es muy sencillo el uso de herramientas tecnológicas y quienes tienen una aceptación inmediata a este tipo de propuestas, adicionalmente al ser estudiantes de Ingenieras existe una posibilidad muy alta que conozcan el uso y manejo de herramientas Web 2.0 y tengan un alto grado de aceptación por ellas, lo que nos permitirá utilizarlas adecuadamente y tomar gran provecho de su aplicación y contar con una ganancia inicial por la motivación y conocimiento de entrada.

De igual forma se espera que los estudiantes al ser de primer semestre, ingresen con muchas expectativas, asertividad y motivación por participar en este nuevo paradigma de programación cuenten además de familiaridad con las redes sociales y el aprendizaje que al interior de ellas se da, lo cual posibilita el uso de las herramientas colaborativas como foros, blogs, wikis, chat y videoconferencia. Así como estaría hablando de estudiantes.

Similarmente se asume en los estudiantes un nivel básico generalizado del idioma inglés, lo que nos permitirá utilizar material de apoyo en esta lengua, el cual se encuentra en mayor cantidad y calidad en el medio. Adicionalmente se logra enriquecer la habilidad del manejo de otra lengua.

Para dar forma a la propuesta de solución se analiza la información recopilada en las encuestas de estudiantes y docentes encuentra dentro de las mismas respuestas ofrecidas para los estudiantes varias características que fueron tomadas como fundamentos en el Diseño de nuestra solución.

La primera de estas características consiste en que para iniciar un acercamiento con la Programación Orientada a Objetos es necesario que el estudiante conozca unos fundamentos de programación básicos que asegurarán nuestro éxito en el tema principal de este estudio, evitando así que se pueda dar por factores ajenos. Entonces en la comprensión de los Fundamentos de Programación Orientada a Objetos, se establecieron los siguientes temas como críticos en el aprendizaje:

- Tipos de Datos
- Variables y Constantes (Visibilidad)
- Sentencias
- Estructuras básicas de Control (Selectivas y repetitivas)
- Procedimientos y funciones
- Arrays

Secundariamente se recomienda como apoyo adicional el acompañamiento en clase presencial de herramientas como JAVA y ALICE, lo cual le permitirá tener competencias para representar en código o gráficamente los conceptos y así enfocarse más en el análisis que en la elaboración de código.

Para esta propuesta se plantean las siguientes competencias a desarrollar:

- Establece autónoma e independientemente la forma y momentos de estudio de las diferentes herramientas tecnológicas de apoyo que se proporcionarán dentro de la solución.
- Comprende los conceptos referentes a la Programación Orientada a Objetos.
- Desarrolla un proyecto aplicativo donde él será autor y constructor a través de la aplicación de los conceptos adquiridos.
- Comprensión y escucha del idioma inglés
- Utiliza y aprovecha las herramientas de la Web 2.0
- Trabaja y aprovecha el trabajo cooperativo con sus pares

Para lograr conseguir las competencias mencionadas, se propone establecer las siguientes estrategias pedagógicas:

1. Segmentar en los principales temas de la Programación Orientada a Objetos para una mejor asimilación de los contenidos que serán vistos secuencial y ordenadamente.
2. Utilizar una plataforma virtual gratuita que le permitirá al estudiante repasar cuantas veces lo necesite toda la temática que desee, con un enfoque comunicativo para el trabajo cooperativo de los grupos.
3. Realizar un test inicial que permitirá al estudiante observarse y conseguir una visión cualitativa de sí mismo e identificar las temáticas que debe reforzar, así como un medio de validación frente al conocimiento de entrada comparado con el conocimiento de salida.
4. Ofrecer material didáctico multimedial para los contenidos temáticos logrando cubrir así las diferentes formas de aprendizaje y las repeticiones que el estudiante considere necesario.

5. Proponer ejercicios prácticos enfocados principalmente a la didáctica necesaria para hacer una relación sencilla entre la realidad y la diagramación de esta
6. Dividir el aprendizaje en 5 momentos, cada uno formado por un objeto de aprendizaje que contendrá material, actividad de aprendizaje y dos test para validar el aprendizaje, uno se presentará al inicio para tener una valoración y poder hacer un comparativo que nos lleve a corroborar la efectividad de nuestro AVA y continuar mejorándolo y un test final que le permitirá comprobar efectivamente su aprendizaje en comparación con el test inicial

Teniendo en cuenta estos supuestos anteriormente descritos, se propone iniciar con el planteamiento de una segmentación de la temática general en 5 grandes temas, que permitirán realizar un seguimiento más efectivo al proceso de aprendizaje del estudiante, ofreciendo un poco más de control por parte del docente frente a las estrategias a implementar para finalmente trabajar enfocados en Objetos de Aprendizaje por cada uno de los temas, aprovechando la reutilización del material para el curso y otras propuestas posteriores. Dichos temas se manejarían así:

- **Abstracción:** Ingeniería de Software, Fase análisis y Diseño.
- **Implementación:** Declaración de atributos, métodos, ocultando información, acceso a constructores, manejo de paquetes y encapsulamiento.
- **Diseño de clases:** Herencia, sobre-escritura de métodos, control de acceso a los métodos, sobrecarga de métodos y conversión de tipo de datos.
- **Opciones avanzadas:** Atributos de clase, métodos de clase, variables finales, declaraciones estáticas, clases abstractas e interfaces.
- **Java:** Desarrollo en el lenguaje (código).

4.1.1. LMS seleccionado

Partimos entonces con esta base a la escogencia del LMS donde publicaremos nuestra propuesta de enseñanza aprendizaje. Dicha plataforma se propone sea de Uso Libre y cuente con la robustez necesaria para soportar las necesidades interactivas y de comunicación que se plantean.

Actualmente existen diversas opciones calificadas para escogencia, tales como: Moodle, Dokeos (Claroline), Autor, CHEF, Eledge etc. Los aspectos a tener en cuenta para la escogencia, tienen que ver con:

1. **Disponibilidad:** Este disponible para su uso gratuito, sin instalaciones ni procesos engorrosos por parte de docentes y estudiantes.
2. **Acceso:** Facilidad desde un navegador en cualquier tipo de computador sin fuertes requerimientos de máquina ni de conexión Web.
3. **Costo:** Deberá ser gratuita por lo menos para su fase inicial, posteriormente y cuando el proyecto se consolide puede pensarse en adquirirla en su versión profesional para perfeccionar la propuesta.
4. **Usuarios activos:** experiencia a nivel mundial por usuarios alrededor del mundo.
5. **Sistemas Operativos:** Diversidad de opciones donde puede ser ejecutada.
6. **Manejo de idiomas:** Buena cantidad de idiomas garantizan una buena cobertura y crecimiento, en nuestro caso especialmente se busca contenta el idioma español.
7. **Servidores libres:** De acceso gratuito y confiables.
8. **Enfoque en trabajo colaborativo:** Se espera contar con una plataforma que brinde el suficiente soporte colaborativo para la cooperación de los aprendientes.

9. **Facilidad para la incorporación de recursos multimedia:** Deberá contar con espacios para acceder la mayor cantidad de tipos multimedia con gran variedad de tipos de archivos permitidos. Así como plantillas que faciliten y agilicen la labor del docente que generalmente no es experto en diseño gráfico.
10. **Utilización de Estándares:** Una de las tendencias recientes en educación y con mayor acogida es el trabajo mancomunado para la construcción y el avance conjunto, ésta precisamente es la función que cumplen los estándares, así como la garantía de calidad y organización del material producido.
11. **Robusto control a través de estadísticas:** Deberá poseer un control de seguimiento tanto para el curso como para el alumno, que permita contar con información confiable y oportuna para la toma de decisiones en pro del mejoramiento continuo de la propuesta de AVA.

Teniendo en cuenta estos ítems, se ha encontrado una plataforma de uso libre con renombre en el medio académico que ofrece facilidad en la creación de la solución, así como en el uso de ella por parte de los estudiantes. Dicha plataforma es **DOKEOS**.

Una vez seleccionado el medio, la temática, nuestro usuario final y las características de nuestra propuesta se procede a diseñar la metodología sobre la cual se trabajará.

4.1.2. METODOLOGÍA

La propuesta tiene su fundamentación principal en el trabajo colaborativo y especialmente cooperativo, basado en el aprendizaje autónomo y reflexivo que concientiza al estudiante de su aprendizaje y lo contextualiza a través de un

proyecto de aula. Este último tiene su fundamentación en la respuesta de las encuestas donde los estudiantes dan una puntuación muy alta al aprovechamiento del desarrollo de proyectos dentro del aula de clase como una forma efectiva de aprendizaje.

Se plantea una secuencia o itinerario de aprendizaje en un Objeto de Aprendizaje, que el aprendiente seguirá y que estará controlado a través del acompañamiento presencial y del control de los informes presentados por la herramienta. Dicho itinerario contará con diferentes momentos que tendrán unos prerrequisitos que controlarán cada paso y que asegurarán una ruta con unas metas y objetivos.

Cumpliendo con la meta cualitativa se inicia la primera fase del proceso con el tema Abstracción, a través de la presentación de un test de 5 preguntas cerradas que se autocalificarán en la herramienta y podrá presentar cuantas veces lo desee, dando una retroalimentación clara y precisa sobre el error cometido y las razones por las cuales es equivocada dicha respuesta. Este tipo de trabajos le permitirá al estudiante aprender sobre sus errores hasta encontrar por sí mismo la solución al test presentado.

A continuación del test se propone una actividad de aprendizaje que consistirá en la realización de la primera fase del proyecto de trabajo y concerniente al tema de Abstracción, donde el estudiante a través de un planteamiento de un contexto ubicará los datos para plantear una solución desarrollada.

Para la realización de esta actividad de aprendizaje el estudiante contará con las herramientas cooperativas que DOKEOS pone a disposición de sus estudiantes, dando paso a la pluralidad de voces y a la polifonía sobre la sinfonía. Este uso de las herramientas deberá estar coordinado desde dos puntos, uno de ellos es la propuesta de utilización didáctica y pedagógica de cada uno de ellos así:

- **Foros:** Herramienta que da soporte a discusiones u opiniones en línea y que estarán enfocados a un objetivo específico de cada tema de modo que se fomente la comunicación asincrónica para ampliar, profundizar y discutir sobre los temas del curso para generar conocimiento. Esta herramienta apoya al docente en la verificación de la apropiación del conocimiento en los estudiantes, por ello ha sido creado un foro para cada tema donde el docente guiará a través de preguntas propuestas los temas que considere debe reforzar.
- **Wikis:** Herramienta de trabajo cooperativo que permite editar, agregar, modificar, borrar para finalmente construir un documento del equipo. En nuestro caso se propone un wiki por tema, donde se desarrollará un ejercicio que los estudiantes irán construyendo en grupo, puede ser determinado por el docente si es el mismo ejercicio propuesto como actividad de aprendizaje o uno paralelo, cada uno agregará o quitará aspectos según lo considere para finalmente construir una entrada por grupo, por tema, por proyecto.
- **Blogs:** Es una herramienta que permite establecer un canal de información informal entre docente y aprendiente, promoviendo la interacción social y ofreciendo al alumno un medio personal para su experimentación en su propio aprendizaje. Es por ello que se propone un blog para cada uno de los cinco temas mencionados, ya que permite controlar el nivel de apertura que se desea para facilitar así su integración en las instituciones educativas y mantiene su carácter exógeno a través de los enlaces en la Web. Se propone entonces que este blog sea desestructurado, buscando sea una práctica que trascienda el cumplimiento de este curso y que permita la elaboración de pensamiento secuencial. Entonces se espera que los alumnos expresen su opinión acerca de su aprendizaje y los aspectos que les ayudaron a afianzar sus conocimientos.

- **Chat:** Herramienta de trabajo en línea que ofrece la opción de consultoría virtual para que el docente atienda a sus estudiantes en ciertos horarios propuestos al día que le permitirán a él ofrecer este espacio a sus estudiantes en diferentes horas del día teniendo una mayor efectividad en el tiempo dispuesto para la enseñanza. De igual forma ofrece más facilidad a los estudiantes para la comunicación y postulado de preguntas, dada su tendiente disposición en el día a día de esta herramienta y por la complicitad que da una pantalla de computador que en muchos casos permite que el estudiante se decida a preguntar dudas que en clase pudiera evitar por el comentario del grupo.

Una vez se desarrollan todos los ejercicios propuestos en el tema, el estudiante presenta un test similar al inicial de 5 preguntas cerradas, donde se podrá observar un diagnóstico final que permitirá evidenciar el conocimiento adquirido a través de esta propuesta y compararlo con la validación inicial ofreciendo estadísticas frente a la eficacia de la propuesta.

Se plantean 5 momentos de trabajo coherentes con los 5 subtemas en los que se ha dividido el tema principal que se establecerán acordes con el planteamiento del plan de estudios y los momentos en que los estudiantes abordarán las temáticas propuestas en esta herramienta. Cada uno de ellos con itinerarios de aprendizaje y una duración de acuerdo a la clase presencial y el avance en temática.

Finalmente la Evaluación del Aprendizaje se da a través de la entrega de los avances del proyecto que los estudiantes concretan en la realización de cada tema y que deberá ser coherente con las calificaciones obtenidas en los test presentados.

4.1.3. SEGUIMIENTO DEL DESEMPEÑO

Se plantea un seguimiento de esta propuesta por parte del docente como apoyo a su clase presencial a través de las herramientas estadísticas que ofrece la herramienta y que permiten analizar tanto individual como grupalmente el desempeño de los estudiantes, en diferentes aspectos tales como:

- Trabajo complementario a la clase presencial que los estudiantes realizarán autónomamente en casa y que tendrá un valor porcentual en la nota definitiva.
- Trabajo colaborativo de construcción grupal que perdurará para el uso posterior de nuevos grupos.
- Refuerzo de lectura in inglés.

4.2. IMPLEMENTACIÓN EN DOKEOS

Una vez identificadas las características del contexto que guiará nuestra propuesta, se procede al respectivo montaje en la plataforma Dokeos, seleccionada en el estudio realizado.

La primera es la definición de las opciones que se estructuraran en la herramienta, para ello el docente cuenta con diferentes opciones representadas en la siguiente pantalla:

Free Campus - Dokeos

Programación Orientada a Objetos

Usuarios en línea: 152 (1 en este curso) | Vista de estudiante | Ayuda

Página principal Mis cursos Mi perfil

Programación Orientada a Objetos Salir (aangarit)

Bienvenido Estimado Estudiante

Este curso ha sido el resultado de una investigación realizada para el proyecto de tesis de la maestría en e-learning: **Mecanismo E-learning de apoyo a la enseñanza de la Programación Orientada a Objetos.**

Tomando como principio las ventajas que tiene la educación virtual o E-learning en nuestro estilo de vida diario, las múltiples herramientas Web 2.0 existentes y las ventajas que ofrecen la facilidad con que los estudiantes actuales adoptan estas tecnologías se inició una investigación en torno a la existencia de este tema en la Programación Orientada a Objetos, encontrando que la mezcla de estos dos es casi nula actualmente.

Con esta base y con el apoyo de los docentes de la Facultad de Sistemas de la UNAB se propone este curso que actuará paralelamente al estudio presencial y que le permitirá al estudiante a ciencia cierta identificar sus posibles falencias en los temas conjunturales y reforzarlos en aras de perfeccionar su aprendizaje.

Creación de contenidos

- Descripción del curso
- Lecciones
- Ejercicios
- Evaluaciones
- Abstracción
- Diseño de Calses
- Abstracción
- Documentos
- Enlaces
- Anuncios
- Glosario
- Implementación
- Opciones avanzadas
- Análisis y Diseño Orientado a Objetos Caso de estudio y requerimientos

Grafica 6 Pantalla de presentación del curso

Se informa al estudiante del planteamiento del curso como tal de modo que esté muy bien informado en la descripción del curso.

Free Campus - Dokeos Programación Orientada a Objetos
Usuarios en línea: 165 (1 en este curso) | Vista de profesor

Página principal **Mis cursos** Mi perfil Salir (aangarit)

Programación Orientada a Objetos > Descripción del curso

Descripción general

Este módulo se ofrece como herramienta de apoyo al curso de Fundamentos de Programación en la temática de la Programación Orientada de Objetos.

Objetivos

El estudiante podrá establecer criterios autónomos e independientes para el apoyo de su estudio de los conceptos de la Programación Orientada a Objetos a través del seguimiento temático y paralelo de lo visto en el aula de clase y la elaboración final de un proyecto paso a paso donde aplique cada concepto interiorizado, con la posibilidad de repetir cada apartado cuantas veces necesite y las partes que necesite.

Contenidos

1. Abstracción: Ingeniería de Software, Fase análisis y Diseño
2. Implementación: Declarar atributos, métodos, ocultando información, acceso a constructores, manejo de paquetes, encapsulamiento
3. Diseño de clases: Herencia, sobrescritura de métodos, control de acceso a los métodos, sobrecarga de métodos, conversión de tipo de datos
4. Opciones avanzadas: Atributos de clase, métodos de clase, variables finales, declaraciones estáticas, clases abstractas e interfaces.

Metodología

Se propone realizar inicialmente un test que permitirá al estudiante observarse y conseguir una visión cualitativa de si mismo e identificar las temáticas que debe reforzar, paso seguido hacer una segmentación de los principales temas de la Programación Orientada a Objetos para una mejor asimilación de los contenidos que serán vistos secuencial y ordenadamente. Contando con el acompañamiento de herramientas audiovisuales y gráficas que permiten cubrir las diferentes formas de aprendizaje y repasar cuantas veces necesite toda la temática que desee.

Adicionalmente se propone la realización ejercicios prácticos enfocados principalmente a la didáctica necesaria para mostrar una relación sencilla entre la realidad y la diagramación de esta, y el uso de videos explicativos que reforzaran el aprendizaje y le permitirán al estudiante repetirlos cuantas veces le sea necesario.

Finalmente se concluye el aprendizaje con un test que le permitirá comprobar efectivamente su aprendizaje en comparación con el test inicial.

Grafica 7 Pantalla de Descripción del curso

Se plantea entonces realizar itinerarios de aprendizaje donde el estudiante cuente con la información necesaria organizada por cada tema segmentado ubicándola fácil y rápidamente. Se presentarán entonces 5 lecciones, una para cada tema segmentado. Así:

Free Campus - Dokeos Programación Orientada a Objetos
Usuarios en línea: 159 (1 en este curso) | Vista de profesor

Página principal Mis cursos Mi perfil Salir (aangarit)

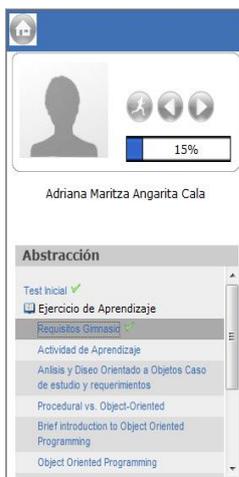
Programación Orientada a Objetos > Lecciones

Nombre	Progreso
Abstracción	<div style="width: 100%;"><div style="width: 100%;"></div></div> 100%
Implementación	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0%
Diseño de Clases	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0%
Opciones Avanzadas	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0%
JAVA	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0%

Profesor : Adriana Maritza Angarita Cala Plataforma Dokeos 1.8.6.1 © 2010

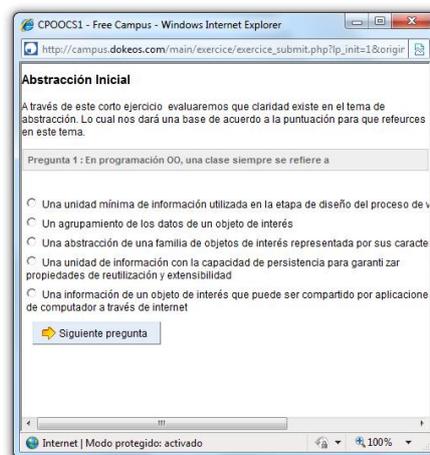
Grafica 8 Pantalla de presentación de las lecciones

Cada una de estas lecciones, tal como se planteó inicialmente, contará con diferentes momentos descritos como su itinerario de aprendizaje, de modo que el estudiante podrá observar cuál de ellos ha cumplido y en qué porcentaje del cumplimiento del itinerario se encuentra, así:



Grafica 9 Pantalla de presentación del Itinerario de Aprendizaje

El primer momento de las lecciones planteadas será la presentación de una evaluación de 5 preguntas que se resolverá automáticamente. Esta evaluación tiene un objetivo cualitativo que permitirá al estudiante ser consciente de su comprensión en el tema. Así:



Grafica 10 Pantalla de presentación del curso

Una vez el estudiante haya respondido la evaluación recibirá una retroalimentación para cada pregunta de la misma, ofreciendo así un ayuda para el aprendizaje ya que no sólo evidencia el no conocimiento respecto de un tema sino que recibe una explicación frente al análisis que ya hizo el estudiante, así:

Dentro de la experiencia del proceso investigativo se pudo encontrar que una de las formas en que los estudiantes prefieren, aprender es a través del desarrollo de proyectos en sus clases. Por ello para esta propuesta se decidió a partir del planteamiento de un ejercicio desarrollar todo un proyecto de programación. Así:

<input type="radio"/>	<input type="radio"/>	Una información de un objeto de interés que puede ser compartido por aplicaciones de computador a través de internet	
Puntuación : 0/0			
Pregunta 2 : Se está definiendo el diseño de un sistema de información para una empresa donde se cuenta con la siguiente descripción. Una clase Persona tiene un identificador, una cedula y un nombre. Un empleado es una Persona que tiene un salario y un cargo.			
Teniendo definida la clase Persona, ¿cuál de los siguientes grupos de atributos serían apropiados para incluirlos en la definición de la clase Empleado?			
Su selección	Selección correcta	Respuesta	Comentarios
<input type="radio"/>	<input type="radio"/>	Identificador, Salario y cargo	
<input checked="" type="radio"/>	<input type="radio"/>	Identificador, cedula, Salario, Cargo	Incorrecto!!! ya que incluye atributos de la clase base Persona y se constituyen en distractores si la persona que es evaluada hace una relación con la definición de llaves primarias en bases de datos y considera que identificador (a) o identificador y cedula (opción b) llaves por medio de las que se puede acceder a los atributos definidos en Persona.
<input type="radio"/>	<input type="radio"/>	Identificador, Cedula, nombre, salario, cargo	
<input type="radio"/>	<input checked="" type="radio"/>	Salario, Cargo	
Puntuación : 0/10			
Pregunta 3 : En la programación orientada a objetos el concepto de ignorar los detalles y concentrarse en las características esenciales de un objeto se conoce como:			
Su selección	Selección correcta	Respuesta	Comentarios
<input type="radio"/>	<input type="radio"/>	Abstracción	

Grafico 11 Pantalla de retroalimentación de la evaluación

Adriana Angarita

- Abstracción
 - Presentación ✓
 - Planteamiento del Desarrollo del tema ✓
 - Abstracción ✓
 - Test Inicial Abstracción
 - Actividad de Aprendizaje
 - Nuevo ejercicio
 - Requisitos Gimnasio ✓
 - CC40B - Analisis y Diseo Orientado a Objetos
 - Definiendo Clases en el momento de Abstracción

Ejercicio de aprendizaje

Tip 1

Iniciaremos esta propuesta de aprendizaje en esta primera fase: Abstracción, a través de la realización de un proyecto que iremos alimentando a medida que vayamos estudiando cada segmento.

Después de realizar el test inicial iniciaremos la primera fase del proyecto, para ello se debe leer el archivo **Requisitos de Gimnasio**, el cual nos expone el contexto de nuestro proyecto y que desarrollaremos a través de toda la herramienta. Como resultado de esta actividad de aprendizaje elabore las clases correspondientes al ejercicio con sus atributos.

Tip 4

Una vez tengas tu entregable elaborado (clases con sus atributos), envíalo a través de la herramienta tareas para revisión de tu docente de modo que compartamos este trabajo en el salón de clase junto con tus compañeros para analizar no sólo los resultados sino los inconvenientes que tuviste realizándolos individualmente y qué construcción podemos adelantar colaborativamente.

A continuación de tu envío podrás presentar el test de evaluación final que nos permitirá corroborar tu

Tip 2

Es importante recordar que este tema de Abstracción trabajaremos los subtemas de Ingeniería de Software, Fase de Análisis y Diseño.

Tip 3

Encontrarás unos tipos en los audios distribuidos alrededor de la página que te ayudaran en tu aprendizaje.

De igual forma cuentas con los enlaces y documentos que han sido seleccionados y dispuestos dentro del itinerario de aprendizaje, consistentes en ayudas didácticas y sencillas (videos, textos etc) para reforzar tu aprendizaje.

Recuerda que cuentas con la herramienta de foros, donde encontrarás preguntas y respuestas a tus posibles dudas y que serán un apoyo interesante a tu estudio. Propón allí inquietudes que surjan en tu trabajo de modo que tus compañeros o el docente traten de dar respuesta y sea útil para futuros estudiantes.

Iniciemos entonces con nuestro proyecto... adelante!!!!

Grafico 12 Pantalla de presentación del ejercicio de aprendizaje

Esta actividad de aprendizaje contará con ciertas ayudas didácticas que facilitan el aprendizaje tales como:

- Audios con “tips” proporcionados por expertos en el tema que facilitan el desarrollo y concepción del proyecto.
- Videos sencillos, claros, directos, didácticos y prácticos.
- Lecturas en inglés y español con ejemplos de cada tema que guiarán al estudiante por medio de la casuística a entender los temas con un referente.
- Enlaces temáticos en la Web seleccionados especialmente por su claridad, coherencia y pertinencia en el tema, no sólo de ambientes académicos sino inclusive de empresa (microsoft programador cinco estrellas)

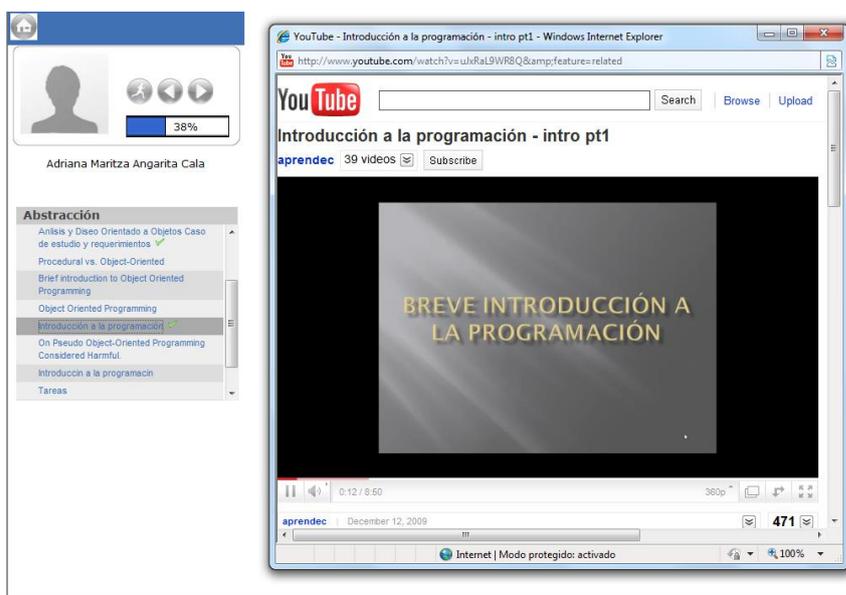


Grafico 13 Pantalla de presentación de los videos

- Documentos de apoyo seleccionados que cumplen con características tales como rigor científico y sencillez, esta es la visión del estudiante de los documentos.

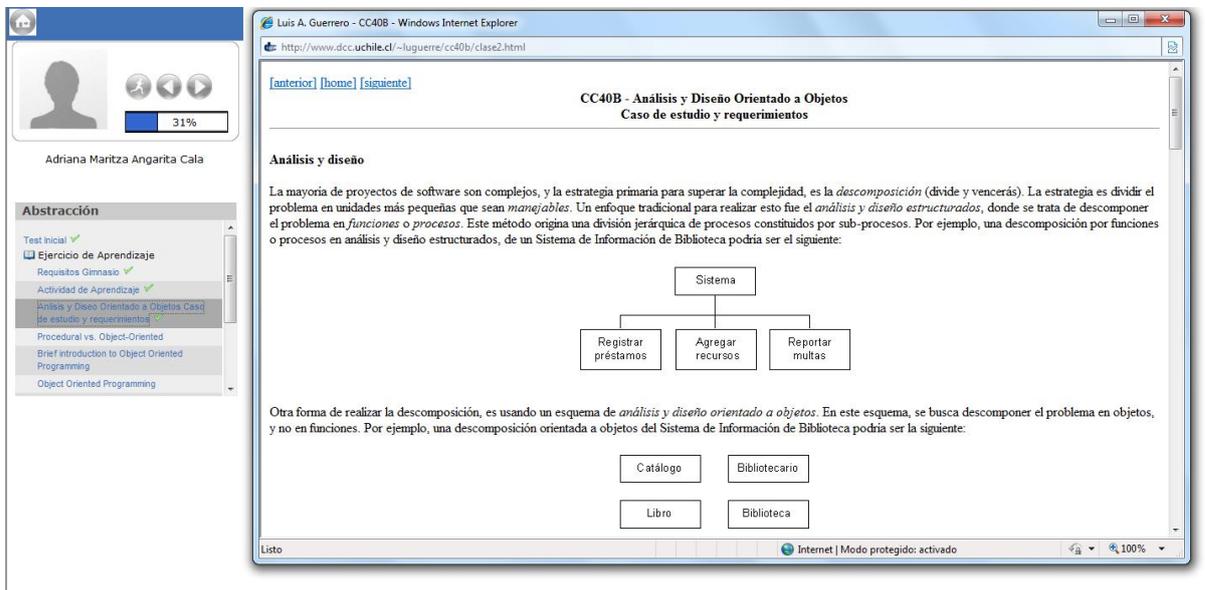


Grafico 14 Pantalla de presentación de documentos

- Enlaces temáticos que acercan otras investigaciones de grupos que han avanzado en el tema del aprendizaje de cada tema
- De modo que se aproveche el aprendizaje situado, cooperativo y el aprendizaje de grupos, se propone el uso de las siguientes herramientas:
 - Blogs: La herramienta de Blogs nos permite construir estudiar los resultados de cada uno de los participantes en cada uno de los ejercicios. La propuesta es a analizar cada una y encontrar las falencias o fortalezas permitiendo un aprendizaje completo y efectivo. Esta es la forma como los estudiantes la visualizan a través de la plataforma.

Free Campus - Dokeos Programación Orientada a Objetos
Usuarios en línea: 219 (1 en este curso) | Vista de estudiante | Ayuda

Página principal **Mis cursos** Mi perfil Salir (aangarit)

Programación Orientada a Objetos - Abstracción

Principal Nuevo artículo Administración de tareas Administración de usuarios

Abstracción

Construyendo en torno a la abstracción

Octubre 2010						
Lun	Mar	Mié	Jue	Vie	Sáb	Dom
					1	2
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Buscar

Mis tareas

Sin tareas

Bienvenido

Viernes 24 Septiembre de 2010 (05:25)

¡Este es el primer artículo en el blog! En adelante, todos los usuarios suscritos a este blog pueden participar

Autor: Angarita Cala Adriana Maritza - Comentarios: 0

Profesor: Adriana Maritza Angarita Cala Plataforma Dokeos 1.8.6.1 © 2010

Gráfico 15 Pantalla de presentación del blog

- Foros: Se plantea un foro por cada tema donde el docente motivará la discusión a través de preguntas, suposiciones y afirmaciones, logrando un debate sano y constructivo que será aprovechada por los estudiantes de posteriores niveles. A continuación

Free Campus - Dokeos Programación Orientada a Objetos
Usuarios en línea: 222 (1 en este curso) | Vista de profesor

Página principal **Mis cursos** Mi perfil Salir (aangarit)

Programación Orientada a Objetos - Grupos - Área del Grupo (Estudiantes UNAB) - Abstracción - Definiendo Clases en Abstracción

Foro

[Volver a la vista general del foro](#) [Volver al foro](#) [Responder a este tema](#) [Vista plana](#) [Vista arborescente](#) [Vista jerarquizada](#)

Definiendo Clases en Abstracción

Primer mensaje | Mensaje anterior [1 / 1] | Mensaje siguiente | Último mensaje

Dokeos Team
2010-05-24 04:43:47

[Responder a este mensaje](#)
[Citar este mensaje](#)

Definiendo Clases en Abstracción

En el momento de la creación y/o diseño de las Clases se debe buscar la forma en que realmente representen requerimientos y respuestas. Enfocándose primero en los verbos que me permitan reconocer las propiedades y comportamientos que necesita y las que no.

Qué otros aspectos te han sido útiles en este momento de definición de Clasescc on sus respectivos atributos?

Estructura

Definiendo Clases en Abstracción

Profesor: Adriana Maritza Angarita Cala Plataforma Dokeos 1.8.6.1 © 2010

Gráfico 16 Pantalla de presentación del foro

se presenta la pantalla de cómo es vista en la plataforma.

- Chat: De igual forma se cuenta con la herramienta de chat para utilizar con las personas que se encuentren en línea en el mismo momento, así:

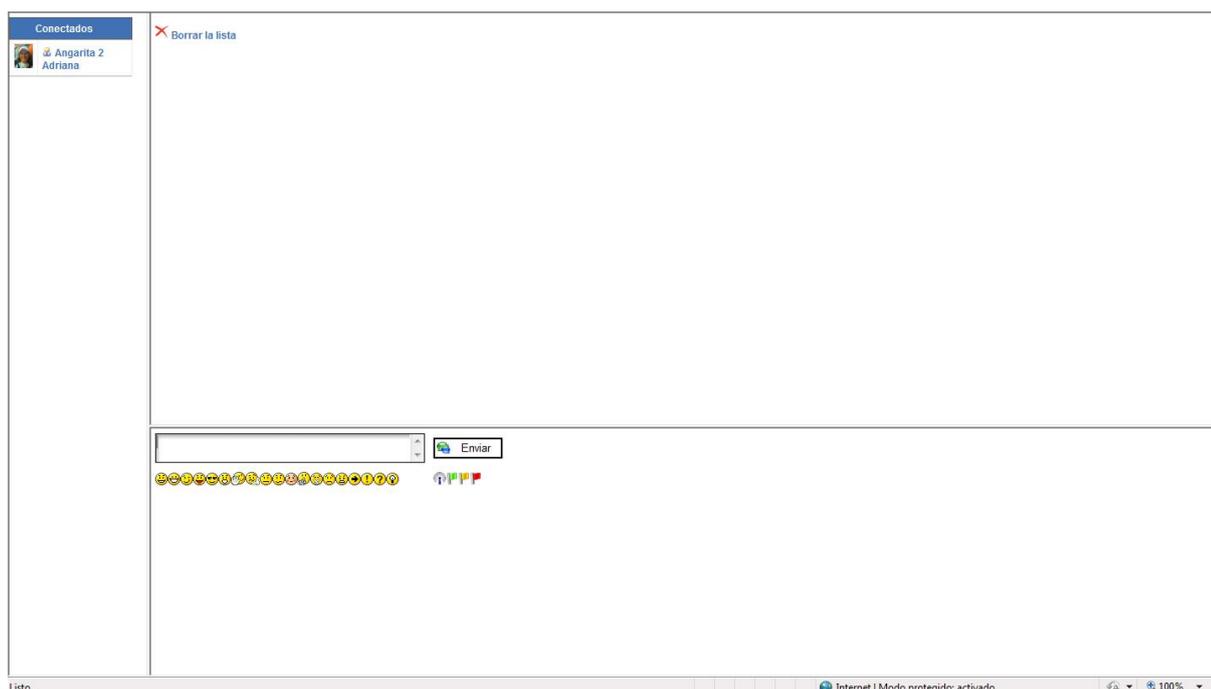


Grafico 17 Pantalla de presentación del chat

- Documentos compartidos: De igual forma se podrá compartir documentos entre los participantes del grupo, fomentando la investigación como trabajo complementario para el crecimiento del equipo, así:

Demo Campus - Dokeos Programación Orientada a Objetos angarita 2 adriana
Usuarios en línea: 464 (2 en este curso) | Vista de estudiante | Ayuda

Página principal Mis cursos Mi perfil Mi agenda Informes Salir (aangarit2)

Programación Orientada a Objetos - Compartir documentos - Archivos enviados

El archivo ha sido enviado

Añadir una categoría Enviar un archivo

Archivos enviados Archivos recibidos

1 - 1 / 1 1 / 1

Tipo	Título	Tamaño	Dirigido a	Último reenvío	Modificar
	DCE0_IntroduccionOrientacionAObjetos.pps	1966 kB	U00069782 U00069782, U00069750 U00069750, U00069708 U00069708, U00069426 U00069426, U00069411 U00069411, U00069407 U00069407, U00069396 U00069396, U00069394 U00069394, U00069358 U00069358, U00069341 U00069341, U00069137 U00069137, U00068917 U00068917, U00068431 U00068431, U00068313 U00068313, U00067514 U00067514, U00067421 U00067421, U00062305 U00062305, U00062208 U00062208, U00061539 U00061539, U00058540 U00058540, U00052057 U00052057, U00051726 U00051726, U00051303 U00051303, U00044950 U00044950, U00040312 U00040312, U00040312, U00035587 U00035587, U00015792 U00015792, 11101045 11101045, 100004461 100004461, 100001577 100001577,	Hoy 2010-10-26 00:43:41	0 Comentarios

Seleccionar todos - Anular selección todos

1 / 1

Plataforma Dokeos 1.8.6.1 © 2010

Internet | Modo protegido: activado

Grafico 18 Pantalla de presentación de documentos compartidos

- Wiki: Se presenta una Wiki del tema propuesto que tendrá un manejo tal como se explicó en la metodología y que perdurará una vez termine el grupo su paso por la herramienta de modo que las clases posteriores obtendrán un apoyo frente a la construcción conjunta realizada por unos pares suyos, así:



Grafico 19 Pantalla de presentación del wiki

- Videoconferencia: Esta herramienta es muy adecuada pues se plantea que en un momento de 1 hora bien sea presencial o totalmente virtual, los estudiantes tengan acceso a escuchar las opiniones de un experto, quien acercará la realidad y mostrará la aplicabilidad del conocimiento que están adquiriendo. Esto se espera motive a los estudiantes a continuar su aprendizaje y además permita despejar dudas que conserven. Se propone el uso de esta herramienta como respuesta a la inquietud presentada por los docentes en su encuesta inicial con respecto a la practicidad y aplicación a la realidad que los estudiantes necesitan ver en sus aprendizajes para motivarlos. De igual forma se plantea la temática y orientación de esta videoconferencia teniendo en cuenta la anotación de los docentes con respecto de la existencia de un público de múltiples disciplinas pero que tienen un solo objetivo en común al entender las ventajas de la POO.

El estudiante enviará su trabajo individual a través de la herramienta tareas presente e individualmente para cada subtema, así:

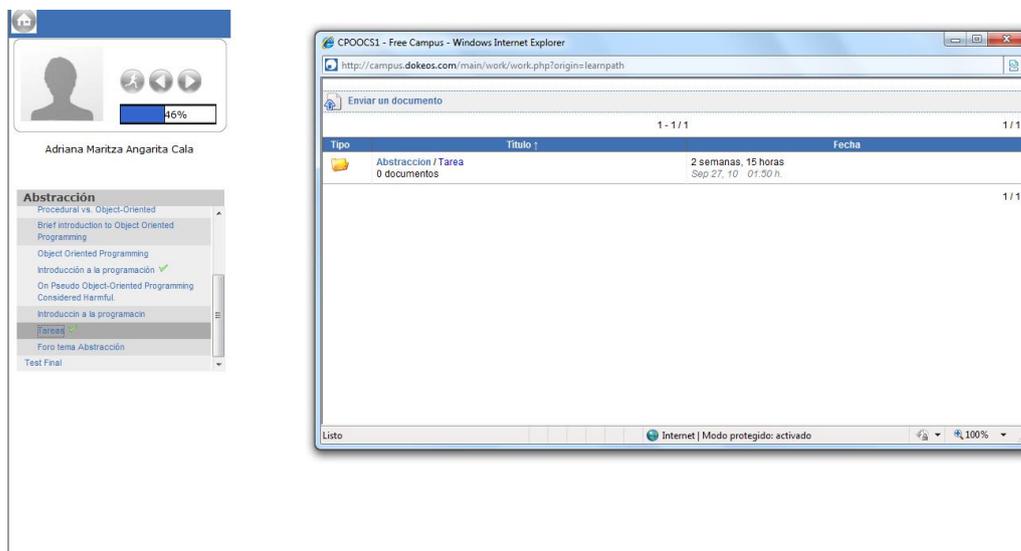


Grafico 20 Pantalla de presentación de la entrega de tareas

Se contará con una herramienta de control propio y de coordinación con su docente con respecto de las actividades y compromisos, llamada Agenda, su distribución es sencilla y lógica así como importante y muy útil. La forma como se presenta esta herramienta es:

Demo Campus - Dokeos

Programación Orientada a Objetos angarita 2 adriana
Usuarios en línea: 466 (2 en este curso) | Vista de estudiante | Ayuda

Página principal **Mis cursos** Mi perfil Mi agenda Informes Salir (aangarit2)

Programación Orientada a Objetos > Agenda

Añadir un nuevo evento Filtrar por grupos/usuarios Mostrar todos Importar un fichero iCalics Ordenar eventos (antiguos / recientes) Eventos del mes Vista mensual

Octubre 2010						
Lun	Mar	Mié	Jue	Vie	Sáb	Dom
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Próximo evento
2010-10-26 14:00:00 - Momento Final

Octubre 2010		
	Momento Final	Dirigido a: Estudiantes Ingenieras
Inicio: Martes 26 Octubre de 2010 14:00 h.	Final: Martes 26 Octubre de 2010 16:00 h.	Modificar
Revisión de las actividades realizadas independientemente y Videoconferencia: Encontrando sentido a la POO		
<hr/>		
	Inicio Prueba Piloto	Dirigido a: Dirigido a
Inicio: Martes 19 Octubre de 2010 14:00 h.	Final: Martes 19 Octubre de 2010 16:00 h.	Modificar
Se inicia la prueba piloto con los estudiantes de ingenierías de la UNAB		
<hr/>		
	Creación del curso	Dirigido a: todos los usuarios del curso
Inicio: Jueves 14 Octubre de 2010 14:10 h.	Final: Jueves 14 Octubre de 2010 14:10 h.	Modificar
El curso ha sido creado en esta fecha		
<hr/>		
	Creación del curso	Dirigido a: todos los usuarios del curso
Inicio: Jueves 14 Octubre de 2010 14:10 h.	Final: Jueves 14 Octubre de 2010 14:10 h.	Modificar

Internet | Modo protegido: activado 100%

Grafico 21 Pantalla de presentación de la agenda

Una vez el estudiante ha realizado la actividad, estudiado el material propuesto, debatido grupalmente y construido colectivamente, se procede a evaluar si ha habido una mejora en el conocimiento desde el momento en que ingreso el estudiante y al finalizar el estudio del tema con el apoyo de la herramienta, para ello el estudiante responderá una evaluación final que al igual que la inicial se autocalificará y retroalimentará robustamente.

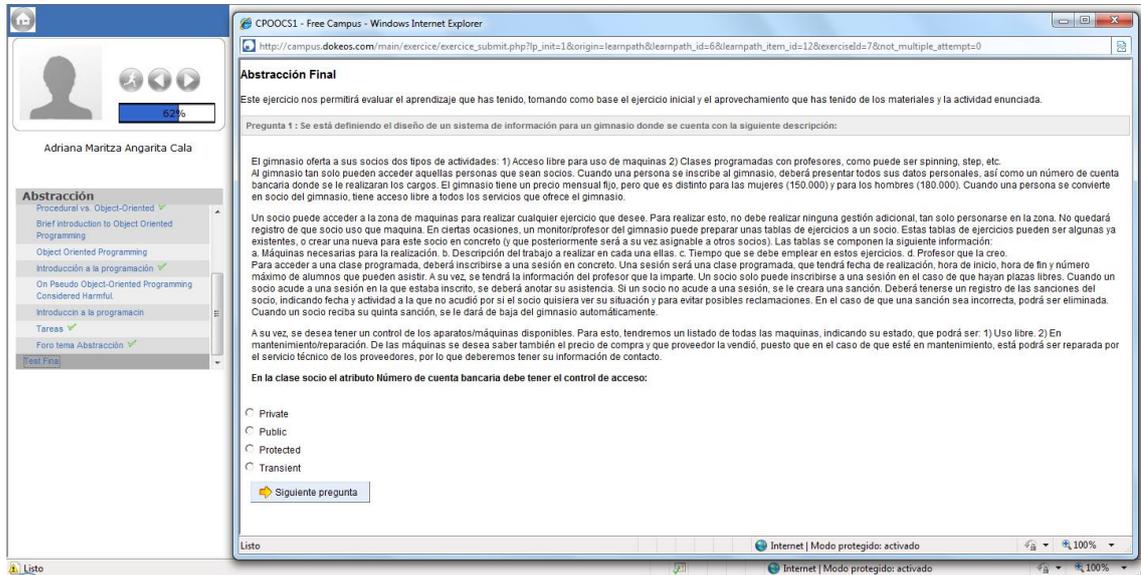


Grafico 22 Pantalla de presentación del test final

Una vez estudiados cada uno de los 5 temas y presentadas las evaluaciones pertinentes, finalmente se llega al momento de evaluación de la herramienta, la cual se realizará a través de una encuesta final que nos permitirá conocer la percepción de los estudiantes en todos los aspectos relativos a ella.

Free Campus - Dokeos Programación Orientada a Objetos
Usuarios en línea: 209 (1 en este curso) | Vista de estudiante | Ayuda

Página principal Mis cursos Mi perfil Salir (aangarit)

Programación Orientada a Objetos > Encuestas

Crear encuesta

1 - 2 / 2 1 / 1

	Nombre de la encuesta	Código de la encuesta 1	Número de preguntas	Autor	Idioma	Disponible desde	Disponible hasta	Invitados	Anónimo	Modificar
<input type="checkbox"/>	Información del grupo referente a sus inicios en POO y a la educación virtual.	001	7	Adriana Maritza Angarita Cala	-	2010-10-10	2010-10-19	1 / 36	Sí	   
<input type="checkbox"/>	Evaluando la efectividad de la herramienta	0102	0	Adriana Maritza Angarita Cala	-	2010-10-12	2010-10-13	0 / 0	No	   

Seleccionar todos - Anular seleccionar todos 1 / 1

Profesor: Adriana Maritza Angarita Cala Plataforma Dokeos 1.8.6.1 © 2010

Grafico 23 Pantalla de presentación de la encuesta

El docente cuenta con diferentes herramientas para la realización del seguimiento al curso, en forma individual y grupal. Algunos ejemplos de herramientas son:

- Manejo de información grupal: Envíos de documentos, avisos, tareas, wikis, foros y agendas para todo el grupo en un solo click facilita el trabajo docente y asegura que la información sea recibida.

Demo Campus - Dokeos Programación Orientada a Objetos angarita 2 adriana
Usuarios en línea: 467 (2 en este curso) | Vista de estudiante | Ayuda

Página principal Mis cursos Mi perfil Mi agenda Informes Salir (angarit2)

Programación Orientada a Objetos > Grupos > Área del Grupo Estudiantes Ingenieras

[Volver a la lista de grupos](#) [Modificar este grupo](#)

Herramientas:

- Foro: Estudiantes Ingenieras
- Documentos
- Agenda
- Tareas
- Anuncios
- Wiki

Tutores:

angarita 2 adriana

Miembros del grupo: 1 - 20 / 30 1 / 2 ▶ H

	Apellidos	Nombre ↑	Correo electrónico
	100001577	100001577	a@a.com
	100004461	100004461	a@a.com
	11101045	11101045	a@a.com
	U00015792	U00015792	a@a.com
	U00035587	U00035587	a@a.com
	U00040312	U00040312	a@a.com
	U00044950	U00044950	a@a.com

Listo Internet | Modo protegido: activado 100%

Grafico 24 Pantalla de presentación del grupo

- Mantenimiento del curso: la posibilidad de realizar copias de seguridad de partes o del curso completo, le permite tener control y prever la pérdida de información, así:

Demo Campus - Dokeos Programación Orientada a Objetos angarita 2 adriana
Usuarios en línea: 443 (2 en este curso) | Vista de estudiante

Página principal **Mis cursos** Mi perfil Mi agenda Informes Salir (angarit2)

Programación Orientada a Objetos > Mantenimiento

Mantenimiento

Copia de seguridad del curso

- **Crear una copia de seguridad**
Puede seleccionar los contenidos del curso que constituirán la copia de seguridad.
- **Importar una copia de seguridad**
Puede transferir una copia de seguridad desde su ordenador o bien usar una copia de seguridad ya disponible en el servidor.

Copiar el curso

Dokeos permite copiar, parcial o completamente, un curso en otro; para ello el curso de destino debe estar vacío.
La única condición es tener un curso que contenga algunos documentos, anuncios, foros... y un segundo curso que no contenga los elementos del primero. Se recomienda usar la herramienta "Reciclar este curso" para no traer futuros problemas con su contenido.

Reciclar este curso

Esta utilidad elimina de forma total o parcial los contenidos de las distintas herramientas de un curso. Suprime documentos, foros, enlaces... Esta utilidad puede ejecutarse al final de una acción formativa o de un año académico. ¡Por supuesto, antes de "reciclar", tenga la precaución de realizar una copia de seguridad completa del curso!

Suprimir el sitio web de este curso

Haga clic en este enlace para eliminar cualquier rastro del curso en el servidor...
¡Esta funcionalidad debe ser usada con extrema precaución!

Plataforma Dokeos 1.8.6.1 © 2010

Listo Internet | Modo protegido: activado 100%

Grafico 25 Pantalla de presentación de mantenimiento del curso

- Recordatorios personales del docente pueden ser mantenidos gracias a la herramienta de Notas Personales, ayudando al docente a mantener toda la información y sus compromisos al día.

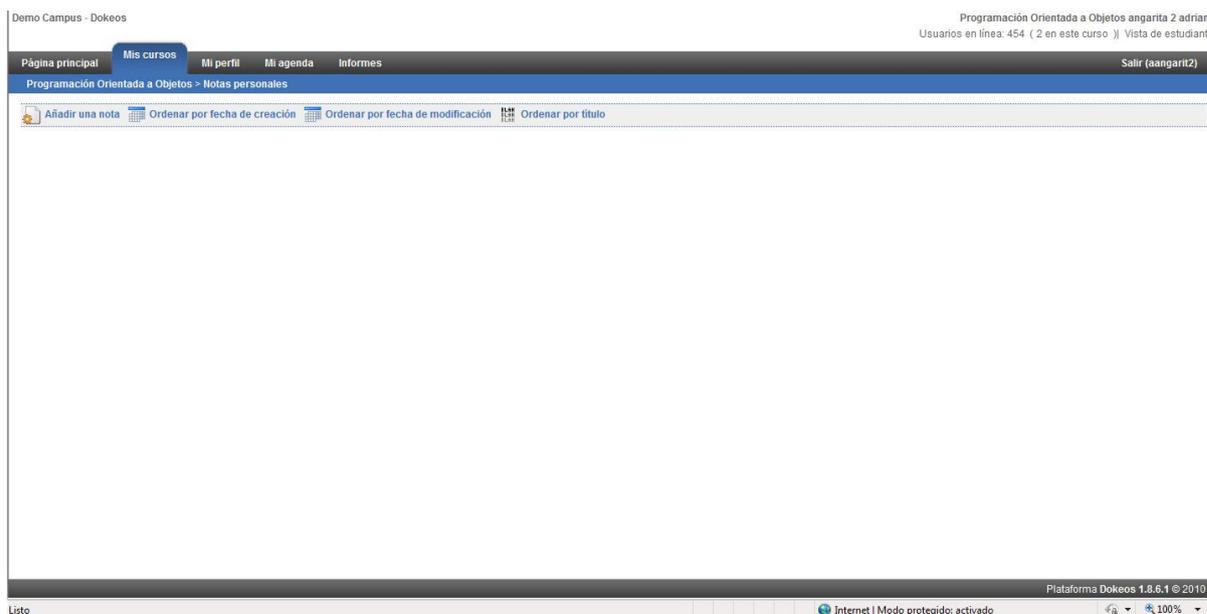


Grafico 26 Pantalla de presentación de recordatorios personales.

- Los detalles de acceso de los estudiantes le permiten al docente tener más herramientas de información con respecto a la desenvolvimiento de los estudiantes, logrando así encontrar respuesta a posibles interrogantes en los procesos de algunos estudiantes, así:

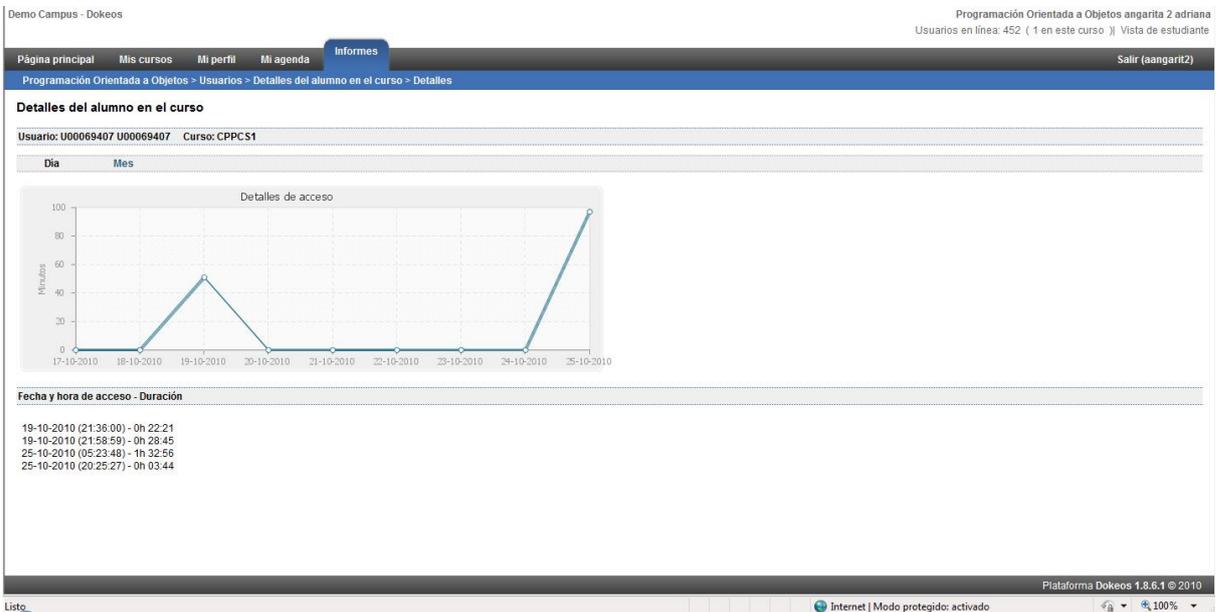


Grafico 27 Pantalla de presentación de control de acceso de alumnos

- La posibilidad de analizar el desempeño de cada estudiante en todo el curso o en cada tema permite al docente evaluar posibles mejoras sobre segmentos del ambiente virtual de aprendizaje diseñado según la efectividad evaluada en esta herramienta, así:

Democampus - Dokeos Detalles del alumno en el curso
Usuarios en línea: 456

Página principal Mis cursos Mi perfil Mi agenda **Informes** Salir (aangarit2)

Programación Orientada a Objetos > Usuarios > Detalles del alumno en el curso

Imprimir Exportar a un fichero CSV Enviar correo Detalles de acceso

Información		Seguimiento	
Nombre:	U00069407 U00069407	Primer acceso:	19 Octubre 2010
Correo electrónico:	a@a.com	Última conexión:	25 Octubre 2010
Teléfono:	90	Tiempo de permanencia en el curso:	2:27:46
Código oficial:	No Código oficial	Progreso:	18.8 %
En línea:	No	Puntuación:	0.04 %

Programación Orientada a Objetos | Número de conexiones a este curso: 13 | Tutor: aangarita 2 adriana

Lecciones	Tiempo ↑	Puntuación ↑	Progreso ↑	Última conexión ↑	Detalles
Abstracción	2:12:47	20 %	94%	25 Octubre 2010	»»
Diseño de Clases	0:00:00	/	0%	-	
Implementación	0:00:00	/	0%	-	
Java	0:00:00	/	0%	-	
Opciones Avanzadas	0:00:00	/	0%	-	

Ejercicios	Puntuación ↑	Intentos	Corregir este ejercicio
Diseño de Clases Final	/	0	
Diseño de Clases Inicial	/	0	
Implementación Final	/	0	
Implementación Inicial	/	0	
Java Final	/	0	

Internet | Modo protegido: activado

Gráfico 28 Pantalla de presentación de desempeño por alumno

- Recordatorios de inactividad o demora que llegarán al correo personal del estudiante, le permiten llegar rápida y efectivamente a ellos, así:

Democampus - Dokeos Programación Orientada a Objetos aangarita 2 adriana
Usuarios en línea: 449 (2 en este curso) | Vista de estudiante | Ayuda

Página principal Mis cursos Mi perfil Mi agenda **Informes** Salir (aangarit2)

Programación Orientada a Objetos > Informes

Seguimiento de los alumnos | Seguimiento del curso Imprimir Exportar a un fichero CSV

Recordatorio para los usuarios que han permanecido inactivos durante más de 7 días Enviar las notificaciones

20 1 - 20 / 32

Código oficial	Apellidos ↑	Nombre	Tiempo en el curso	Progreso en la lección	Puntuación	Tareas	Mensajes de foro	Primer acceso	Última conexión	Detalles
100001577		100001577	0:00:30	5 %	0 %	0	0	22 Octubre 2010	22 Octubre 2010	»»
100004461		100004461	0:00:00	0 %	0 %	0	0			»»
11101045		11101045	0:00:00	0 %	0 %	0	0			»»
Adriana Maritza		Angarita Cala	0:20:21	8 %	0.04 %	0	0	19 Octubre 2010	24 Octubre 2010	»»
Juan Camilo		Solano Estrada	0:35:48	10.6 %	0.08 %	0	0	19 Octubre 2010	19 Octubre 2010	»»
U00015792		U00015792	1:31:33	6.6 %	0.06 %	0	0	19 Octubre 2010	19 Octubre 2010	»»
U00035587		U00035587	0:47:33	8 %	0.02 %	0	0	19 Octubre 2010	19 Octubre 2010	»»
U00040312		U00040312	0:41:32	8 %	0.04 %	0	0	19 Octubre 2010	25 Octubre 2010	»»

Internet | Modo protegido: activado

Gráfico 29 Pantalla de notificaciones por inactividad de los alumnos

- Detalle de encuesta donde se recoge la opinión de las personas involucradas en los roles destinados para este ambiente, da la visión de para quienes se está haciendo esta labor y el impacto que ésta causa en ellos, por eso se cuenta con la herramienta encuesta que permite recoger estas inquietudes, así:

Grafico 30 Pantalla de presentación detallada de la encuesta

4.3. PRUEBA PILOTO

Se realizó una prueba con el grupo de Fundamentos de Programación de la Universidad Autónoma de Bucaramanga, el cual se encuentra en el primer nivel con 32 estudiantes de diferentes facultades y bajo el acompañamiento y seguimiento del docente del módulo.

Estos estudiantes cumplieron con el tema de Abstracción, el cual se encuentra acorde al planteamiento académico que dispone la Universidad para ellos en esta semana 12 de su semestre.

Se inició la prueba con la explicación en clase presencial de la herramienta y la clara exposición de cómo sería la participación de los estudiantes en esta prueba piloto. Fueron inicialmente inscritos en el curso y se realizó el test de abstracción inicial con ellos en clase.

Se hizo una inducción con respecto a la primera fase del proyecto que realizarían tanto individualmente (actividad de aprendizaje) como colaborativamente (blog, foro, wiki) y se delimitó el tiempo para el envío de la tarea correspondiente a abstracción para la siguiente semana.

En el siguiente encuentro junto con el grupo se discutió la experiencia y se revisó el material enviado. Se realizó la videoconferencia propuesta, se presentó el test final del tema y finalmente se respondió la encuesta de evaluación de la herramienta.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Se puede decir que de nuestra fase de exploración de los avances existentes, de la perspectiva que docentes y estudiantes de la asignatura de programación, de la implementación y prueba, se concluyen algunos puntos:

Los estudiantes esperan y necesitan una enseñanza mucho más práctica, enfocándose en la resolución de problemas basándose en un aprendizaje autónomo pero guiado por tanto sus docentes como monitores de semestres superiores que seguramente conozcan los problemas e inconvenientes que los estudiantes estén pasando y la forma sencilla de superarlos.

De manera particular se encuentra que a pesar de lo expuesto en el anterior párrafo dentro de los resultados de la encuesta aparece el uso de la herramienta ALICE con muy baja votación, dando una impresión de no tener mucha aceptación el uso de esta herramienta en ellos. Es preciso analizar detalladamente este tema aunque existe un precedente de y es que gran número de estudiantes no conocían ALICE al momento de la encuesta. De igual forma se resalta la gran aceptación que entre los estudiantes tienen las herramientas web 2.0 y el buen recibimiento que apoyos didácticos como los audios y los videos tuvieron en los estudiantes.

Se realizó la prueba piloto con estudiantes del Ciclo básico de Ingenierías del primer curso de Fundamentos de Programación, de acuerdo a los resultados y al desarrollo de la prueba, se concluye que para el aprovechamiento de la propuesta es necesario que los estudiantes tengan un conocimiento adelantado frente a los conceptos en la Programación Orientada a Objetos de modo que la parte inicial (Abstracción) sea un refuerzo, es decir según el grupo estudiado se propone iniciar con estudiantes de segundo semestre.

Dado que el planteamiento de la temática abarca todo el aprendizaje completo de los temas de Programación Orientada a Objetos se plantea para trabajos futuros implementar el ambiente propuesto a través de varios semestres de modo que pueda ser abordada completamente y así obtener una validación de toda la herramienta.

Las herramientas colaborativas como wikis, chat, blog y foros diseñadas para brindar acompañamiento y construcción colaborativa perdurable deberán estar acompañadas por una fuerte motivación desde las clases presenciales por parte del docente para el uso y aprovechamiento de éstas, proponiendo debates y fomentando su valor en el período académico, a través de la investigación de material externo y respuestas a inquietudes que los estudiantes pueden observar.

Se plantea una actualización y modificación de las preguntas presentadas en los test periódicamente, de tal forma que los estudiantes no pasen esta información a través de los grupos posteriores y así pueda perder validez estas pruebas. Así mismo se propone una constante indagación de material que permita actualizar la propuesta documental, teniendo siempre en cuenta las características que este tipo de material deben tener en cuanto a didáctica y multimedia.

Dentro de los resultados de la prueba piloto se arrojan algunas opiniones negativas frente a la herramienta, al realizar el análisis de los estudiantes responsables de ellas, se encuentra que no confían ni creen en la educación virtual, es decir hay un rechazo de entrada a cualquier aporte que una herramienta e-learning pueda hacer.

De igual forma se concluye que deberá permitirse un mayor espacio de tiempo para cada fase, de modo que los estudiantes puedan realizar un avance pausado y acompañado desde la clase presencial.

Adicionalmente se observa bajo desempeño en la segunda prueba inclusive menor que en la primera, lo cual se puede atribuir a una mayor apropiación de la parte conceptual que de la práctica en el tema de abstracción, ya que a que en el primer test se pretendió evaluar la parte conceptual y en el segundo ejercicio se basó en la aplicación práctica de los conceptos.

De igual forma es de anotar que el 80% de los alumnos habiendo transcurrido el 80% del tiempo propuesto para el desarrollo, no habían ingresado ni una sola vez a la plataforma para leer el material y/o estudiarlo. Esto se refleja en la recepción de tan solo 6 trabajos, equivalente al 20% del total de estudiantes.

Se presentaron algunos inconvenientes técnicos en el uso de la plataforma Adobe Connect Pro al momento de la videoconferencia, que lamentablemente afectó la recepción por parte de los estudiantes del audio y por ende la explicación del experto. Por ello inclusive algunos estudiantes calificaron la videoconferencia de aburrida.

De igual forma se desea mencionar la dificultad que se presenta en la interdisciplinariedad de los estudiantes, ya que estuvo comprobado que sólo los estudiantes de Ingeniería de Sistemas y Mecatrónica se encuentran motivados con la temática y por ello fueron quienes realizaron todas las actividades propuestas.

En general hubo una buena aceptación del acompañamiento de la plataforma, el diseño y la metodología por parte de los estudiantes así como el estímulo y la motivación para seguir adelante.

Se propone continuar con la herramienta en la Universidad y realizar al menos 3 pruebas con estudiantes de segundo semestre para encontrarlas posibles mejoras en un público amplio y ecuánime.

6. REFERENCIAS BIBLIOGRÁFICAS

Börstler Jürgen and Sharp Helen (2003) Learning and Teaching Object Technology.
Tomado el 24 de abril de 2010.

Delgado Cejudo, Sebastián (2003) Elearning. Análisis de plataformas gratuitas.
Tomado el 1 de Octubre de 2010.

Heo Misoko (2003) Florida State University. A Learning and Assessment Tool for
Web-based Distributed Education. Tomado el 18 de enero de 2010.

Hwang Wu-Yuin et al (2008) A web-based programming learning environment to
support cognitive development. Tomado el 18 de Enero de 2010.

Jadud Matthew (2005) A First look at Novice Compilation Behaviour using BlueJ.
Tomado el 24 de abril de 2010.

Keefe Karen et al (2006) Adopting XP Practices for Teaching Object Oriented
Programming. Tomado el 24 de Abril de 2010

Kölling Michael (1999) The Problem of teaching object-oriented programming Part I:
Languages. Tomado el 24 de abril de 2010.

Kölling Michael (1999) The Problem of teaching object-oriented programming Part II:
Languages. Tomado el 24 de abril de 2010.

Sitio Web Proyecto CUIP2

<http://cupi2.uniandes.edu.co/>

Sitthiworachart Jirarat y Joy Mike. (2003) Web-based Peer Assessment in Learning Computer Programming. Tomado el 18 de Enero de 2010

Truong Nghi, Bancroft Peter y Roe Paul (2003) A Web Based Environment for Learning to Program. Tomado el 18 de Enero de 2010.

Vargas Germán, Gamboa Sarmiento Sonia Cristina (2008) Didáctica en la condición postmoderna. De las competencias a la cooperación. Tomado el 13 de febrero de 2010. Disponible en:
http://dialnet.unirioja.es/servlet/dfichero_articulo?codigo=2754106&orden=0

Villalobos Jorge (2007) Cómo enseñar a programar: un enfoque efectivo. Tomado el 18 de Enero de 2010. Disponible en:
<http://cupi2.uniandes.edu.co/proyecto/presentaciones/Cupi2-2007-04-18-CCC07.PDF>

Villalobos Jorge, Casallas Ruby, Marcos Katalina (2005) El reto de diseñar un Curso de Programación de Computadores. Tomado el 24 de abril de 2010.

Villalobos Jorge et al (2005) Looking for a new approach to teach/learn a first computer-programming course. Tomado el 18 de Enero de 2010. Disponible en
<http://cupi2.uniandes.edu.co/docs/Lookingfor-Uniandes-FullPaper.PDF>

Bruce Christine, Buckingham Lawrence, Hynd John, McMahon Camille, Roggenkamp Mike, Stoodley Ian (2004) Ways of Experiencing the Act of Learning to Program: A Phenomenographic Study of Introductory Programming Students at University. Tomado el 24 de abril de 2010.

Wee fang et al (2005) A student model for object-oriented design and Programming.
Tomado el 24 de Abril de 2010.

LISTA DE DE TABLAS

	Pag
Tabla 1 Porcentajes de respuesta a pregunta 1 Encuesta levantamiento información inicial	25
Tabla 2 Porcentajes de respuesta a pregunta 2 Encuesta levantamiento información inicial	26
Tabla 3 Porcentajes de respuesta a pregunta 3 Encuesta levantamiento información inicial	26
Tabla 4 Porcentajes de respuesta a pregunta 4 Encuesta levantamiento información inicial	27
Tabla 5 Porcentajes de respuesta a pregunta 9 Encuesta levantamiento información inicial	28
Tabla 6 Porcentajes de respuesta a pregunta 9 Encuesta levantamiento información inicial	29
Tabla 7 Porcentajes de respuesta a pregunta 9 Encuesta levantamiento información inicial	29
Tabla 8 Porcentajes de respuesta a pregunta 9 Encuesta levantamiento información inicial	30
Tabla 9 Porcentajes de respuesta a pregunta 9 Encuesta levantamiento información inicial	31

LISTA DE FIGURAS

	Pág
Gráfico1 Porcentajes Respuesta pregunta 1 Encuesta información inicial	25
Gráfico2 Porcentajes Respuesta pregunta 2 Encuesta información inicial	26
Gráfico3 Porcentajes Respuesta pregunta 3 Encuesta información inicial	27
Gráfico4 Porcentajes Respuesta pregunta 4 Encuesta información inicial	27
Gráfico5 Porcentajes Respuesta pregunta 5 Encuesta información inicial	28
Gráfico 6 Pantallazo Presentación del curso	43
Gráfico 7 Pantalla de Descripción del Curso	44
Gráfico 8 Pantalla de Presentación de las lecciones	44
Gráfico 9 Pantalla de Presentación del itinerario de aprendizaje	45
Gráfico 10 Pantalla de Presentación del test inicial	46
Gráfico 11 Pantalla de retroalimentación de la evaluación	47
Gráfico 12 Pantalla de Presentación del ejercicio de aprendizaje	47
Gráfico 13 Pantalla de Presentación de los videos	48
Gráfico 14 Pantalla de presentación de documentos	49
Gráfico 15 Pantalla de Presentación del Blog	50
Gráfico 16 Pantalla de Presentación del foro	50
Gráfico 17 Pantalla de Presentación del chat	51
Gráfico 18 Pantalla de Presentación de documentos compartidos	52
Gráfico 19 Pantalla de Presentación del Wiki	53
Gráfico 20 Pantalla de presentación de entrega de tareas	54

Gráfico 21 Pantalla de presentación de la agenda	55
Gráfico 22 Pantalla de Presentación del test final	56
Gráfico 23 Pantalla de Presentación de la encuesta	57
Gráfico 24 Pantalla de Presentación del grupo	58
Gráfico 25 Pantalla de Presentación de mantenimiento del curso	58
Gráfico 26 Pantalla de Presentación de recordatorios personales	59
Gráfico 27 Pantalla de Presentación de Control de acceso de alumnos	60
Gráfico 28 Pantalla de presentación de desempeño por alumno	61
Gráfico 29 Pantalla de notificaciones por inactividad de los alumnos	61
Gráfico 30 Pantalla de Presentación detallada de la encuesta	62

ANEXO 1

ENCUESTA PARA RECOLECCIÓN DE DATOS DE ESTUDIANTES PARA EL DESARROLLO DEL AMBIENTE VIRTUAL DE APRENDIZAJE PARA LA ENSEÑANZA DE PROGRAMACIÓN ORIENTADA A OBJETOS

Entrevistas a estudiantes de ingenierías sobre sus impresiones y óptica con respecto a la asignatura de programación, como medio de recolección de datos que serán insumos para el desarrollo del proyecto que busca ofrecer una solución que facilite la enseñanza-aprendizaje de la programación en las ingenierías.

Estimado estudiante:

Esta encuesta tiene por objeto determinar los principales factores que dificultan el aprendizaje de la programación de computadoras. Su participación en este estudio es crucial para establecer mecanismos innovadores que hagan de la programación un área de estudio de interés y utilidad dentro del ejercicio de su profesión.

Hay 9 preguntas en esta encuesta

Preguntas

A continuación encontrará las preguntas a responder. Todas serán abiertas con espacio suficiente para que el encuestado pueda expresar claramente sus opiniones.

1 [0101]

1. ¿Qué carrera cursa actualmente? *

Por favor seleccione **sólo una** de las siguientes opciones:

- Sistemas
- Mecatrónica
- Mercados
- Financiera
- Energía
- Industrial
- Civil
- Electrónica

- Alimentos
- Ambiental
- Otra

2 [0102] 2. Antes de ingresar a la universidad, en cuántos cursos de programación ha participado? *

Por favor seleccione **sólo una** de las siguientes opciones:

- 0
- 1
- 2 o más

3 [0103]

3. ¿Ha tenido experiencia de tipo laboral o de interés personal con la programación de computadoras, diferente a cursos de programación? *

Por favor seleccione **sólo una** de las siguientes opciones:

- Sí
- No

4 [0104] 4. ¿Considera que aprender a programar es útil para su profesión? *

Por favor seleccione **sólo una** de las siguientes opciones:

- Sí
- No

5 [0105] 5. Durante su curso de “Fundamentos de Programación” usted aprendió a: (puede seleccionar varias) *

Por favor, marque las opciones que correspondan:

- Entender y analizar un problema
- Representar la solución de un problema a través de un algoritmo o diagrama de flujo
- Escribir y editar programas en una computadora
- Compilar y ejecutar programas en una computadora
- Depurar o mejorar sus programas

6 [0106] 6. Indique el grado de complejidad que para usted tuvo cada uno de los siguientes temas del curso de “Fundamentos de Programación. *

Por favor, seleccione la respuesta apropiada para cada concepto:

	1	2	3	4	5
Concepto de Algoritmo	<input type="radio"/>				
Construir algoritmos o diagramas de flujo	<input type="radio"/>				
Operadores relacionales, aritméticos y lógicos	<input type="radio"/>				
Estructuras de Selección Simple, Doble y Anidamiento	<input type="radio"/>				
Estructuras de repetición	<input type="radio"/>				
Sistemas de numeración y conversión	<input type="radio"/>				
Introducción a Java	<input type="radio"/>				
Constructor y Definición	<input type="radio"/>				
Concepto y utilización de métodos	<input type="radio"/>				
Operaciones básicas con arreglos	<input type="radio"/>				
Orientación a Objetos	<input type="radio"/>				
Paso de arreglos a métodos	<input type="radio"/>				

Siendo 1 el menor grado de dificultad y 5 el mayor grado de dificultad.

7 [0107] 7. Evalúe cada uno de los siguientes aspectos referidos a su curso de “Fundamentos de Programación *

Por favor, seleccione la respuesta apropiada para cada concepto:

	1	2	3	4	5
Motivación	<input type="radio"/>				
Número de horas Prácticas	<input type="radio"/>				
Cantidad de temas cubiertos	<input type="radio"/>				
Claridad en la exposición de los temas por parte del profesor	<input type="radio"/>				
Uso de las herramientas de enseñanza adecuadas para el aprendizaje de los temas del curso	<input type="radio"/>				

Evalúe de 1 a 5 siendo, 1 la puntuación mas baja y 5 la más alta

8 [0108] 8. Indique el grado de importancia de las siguientes estrategias para mejorar la enseñanza de la programación de computadoras *

Por favor, seleccione la respuesta apropiada para cada concepto:

	1	2	3	4	5
Ejercicios en el tablero	<input type="radio"/>				
Exámenes prácticos	<input type="radio"/>				
Cuestionarios en línea (por internet)	<input type="radio"/>				
Tutoriales en línea (por internet)	<input type="radio"/>				
Programar en equipos de 2 estudiantes	<input type="radio"/>				
Tutorías con un estudiante de semestres superiores	<input type="radio"/>				
Horas de consulta del docente	<input type="radio"/>				
Alice	<input type="radio"/>				
Videos sobre programación	<input type="radio"/>				
Apoyo a través de monitores	<input type="radio"/>				
Trabajar con Proyectos prácticos en clase a través de todo el curso	<input type="radio"/>				
Análisis y estudio de software ya realizado	<input type="radio"/>				

Escoja el grado de importancia dentro del rango de 1 a 5, siendo 1 Poco importante y 5 muy importante.

9 [0109] 9. ¿Qué sugeriría usted para mejorar la enseñanza de la programación en la Universidad? *

Por favor, escriba su respuesta aquí:

Trate de ser lo mas claro y explícito posible para así poder captar perfectamente su sugerencia

ANEXO 2

ENCUESTA DOCENTE EXPERIENCIAS FRENTE A LA ENSEÑANZA DE LA PROGRAMACIÓN ORIENTADA A OBJETOS

A través de una corta encuesta dirigida a los docentes de programación de las Universidades más importantes de la Ciudad de Bucaramanga, se pudo conocer sus experiencias frente a la enseñanza de programación y su perspectiva frente a las posibles causas de deserción en las ingenierías. De igual forma se conoció su forma de enseñar y las estrategias pedagógicas usadas, lo cual nos permitirá hacer una correlación directa con las respuestas proporcionadas por los estudiantes. La encuesta se presentó así:

1. ¿Cuáles son desde tu experiencia los principales aspectos que hacen compleja la enseñanza del primer curso de programación? por favor ten en cuenta aspectos relacionados con lo curricular (contenidos, estrategias pedagógicas), formación previa del estudiante, motivación hacia la programación, entre otros.
2. ¿Qué apoyos o estrategias pedagógicas utilizas para apoyar la enseñanza del primer curso de programación dentro y fuera del aula?

ANEXO 3 RESPUESTAS DOCENTES FRENTE A ENCUESTA

Se entrevistaron docentes de diferentes instituciones universitarias de Bucaramanga para de esta forma tratar de tomar una muestra representativa y variada. Se consultaron 7 docentes con el siguiente texto:

Saludo

Te cuento actualmente soy estudiante de la maestría en E-learning que ofrece el convenio entre la UOC y la UNAB y estoy realizando mi proyecto de investigación : “Mecanismo E-learning de apoyo a la enseñanza programación de computadoras”, he iniciado el proceso de investigación analizando las principales problemáticas en la enseñanza de la programación y para lograr ofrecer un mecanismo efectivo quisiera contar con el apoyo de personas con la experiencia y el conocimiento como tú, docentes de las universidades más representativas de la ciudad. Por ello quisiera muy respetuosamente solicitarte tu colaboración respondiendo a las siguientes preguntas:

1. ¿Cuáles son desde tu experiencia los principales aspectos que hacen compleja la enseñanza del primer curso de programación? por favor ten en cuenta aspectos relacionados con lo curricular (contenidos, estrategias pedagógicas), formación previa del estudiante, motivación hacia la programación, entre otros.

2. ¿Qué apoyos o estrategias pedagógicas utilizas para apoyar la enseñanza del primer curso de programación dentro y fuera del aula?

Te agradezco la colaboración prestada ya que será un apoyo crucial para la eficacia de mi proyecto de grado.

De igual forma se consultaron algunos datos personales de los encuestados para obtener la información que harían válida su escogencia y respuestas para esta encuesta.

Los datos de los encuestados y sus respuestas son descritos a continuación:

- Profesión: Ingeniero Químico
- Estudios postgrado: Doctorado en Ingeniería Química
- Tiempo docencia programación: 3 semestres

- Asignaturas relacionadas con programación que ha dictado: Ingeniería computacional (contenido: simulación de los fenómenos fisicoquímicos a escalas inferiores a la micro); bases informáticas en ingeniería química (contenido: introducción a la programación, diagramas de flujo, lenguajes de programación VBA-Excel y MATLAB, ejemplos aplicados a la ingeniería química).
- Universidades donde ha dado estas asignaturas: UIS.
- Respuestas a las preguntas:
 - a) Principales aspectos que hacen compleja la enseñanza del primer curso de programación: *Lo que he notado es que la formación previa del estudiante y la motivación por aprender una nueva materia influyen la complejidad del aprendizaje de la programación. Varios estudiantes argumentan que no les impartieron cursos de programación en el colegio, que no recuerdan temas de matemáticas vistos en clase y por eso se sienten perdidos. También, varios estudiantes dan mayor importancia a sus cursos de matemáticas, física y química los cuales están directamente relacionados con sus carreras. Por parte del contenido, estoy seguro que para disminuir la falta de interés, atención y aprehensión, la aplicación de la programación a temas propios de cada carrera debe ser una constante. En cuanto a esto, es necesario apuntar que en la mayoría de los ejercicios ingenieriles se requiere el uso de métodos numéricos, con lo cual, un curso de programación debería estar soportado por un curso previo o paralelo de métodos numéricos.*
 - b) Apoyos o estrategias pedagógicas usadas: *Ejercicios de aplicación los cuales se completan con los estudiantes a lo largo de la clase. Ejercicios en grupo para entregar al día siguiente. Problemas propuestos para que el estudiante investigue y proponga la solución para la siguiente clase. Quices sobre el tema previamente explicado. Exámenes diseñados por el profesor para asegurar que cada estudiante resuelva un problema particular. Asignaciones finales para el curso que compilen lo visto en clase.*

2. Profesión: Ingeniero de Sistemas

- Estudios postgrado: Candidato a Magister en Software Libre

- Tiempo docencia programación: cátedra
- Asignaturas relacionadas con programación que ha dictado: fundamentos de programación, programación de computadores
- Universidades donde ha dado estas asignaturas: UNAB
- Respuestas a las preguntas:

- a. Principales aspectos que hacen compleja la enseñanza del primer curso de programación: *La gran mayoría de estudiantes no tiene bases de programación y es su primera experiencia con este tipo de cursos, por lo tanto es bastante complejo para el docente llegar al punto de nivel donde todos se encuentre sintonizados con los temas impartidos.*

La motivación en muchos de los estudiantes hacia el curso se ve afectada debido a que este curso no hace parte de los cursos de línea de su correspondiente programa académico, por lo tanto lo consideran de relleno y su concepción se reduce simplemente a algo que no les va a servir para nada en la vida. Por lo cual en todo el grupo se genera una indisposición colectiva que el docente tiene que vencer. Esto se debe a que este curso hace parte del ciclo básico de las ingenierías.

Los contenidos en general están bien formulados y de todas formas son revisados cada semestre y en el transcurso del mismo.

- b. Apoyos o estrategias pedagógicas usadas: *En lo personal una de mis estrategias pedagógicas es el uso de documentales en relación al ¿qué?, al ¿cómo?, y al ¿por qué? del uso de las diferentes tecnologías que hacen parte de la vida cotidiana de la gran mayoría de personas en el mundo y que son hoy fundamentales para nuestras interrelaciones personales y laborales. Lo anterior con la intención de sensibilizar al estudiante que la base de esas tecnologías fueron construidas a partir de programas como los hechos en clase y así creará un mayor interés en el curso.*

En este último semestre los curso de programación I y II sean venido apoyando en herramientas didácticas con la filosofía de que

con el juego se vaya aprendiendo. Por ejemplo, Alice es una herramienta libre que pretende desde un mundo y una gran cantidad de objetos se pueda crear historias y juegos a través de arrastrar y soltar los mismo para luego definir una serie de comandos que en ultimas especifica las estructuras básicas y lógica de programación, y además que ayuda a esta última a desarrollarla siendo una de las mayores dificultades del estudiante.

3. Profesión: Ingeniero de Sistemas

- Estudios postgrado: Maestría en Ciencias Computacionales, Especialización en Ingeniería del Software
- Tiempo de docencia en programación: 17 años
- Asignaturas dictadas relacionadas con programación: Introducción a la programación, programación general, fundamentos de programación, programación de computadoras y estructura de datos
- Universidades donde ha dado estas asignaturas: UNAB, UIS
- Respuestas a las preguntas:
 - a. Principales aspectos que hacen compleja la enseñanza del primer curso de programación: *Enseñar a los estudiantes a estructurar soluciones de manera algorítmica, pues desde el colegio no se estimula al estudiante a crear sus propias soluciones sino a intentar buscar recetas o soluciones repetitivas. Desmotivación en muchos casos, especialmente en estudiantes de otras ingenierías (diferentes a Sistemas) que no ven clara la importancia de estas asignaturas, sino las tomas como materias impuestas. Falta usar más herramientas de apoyo, que faciliten el proceso, que los estudiantes asimilen con más facilidad, como el caso de Alice para la introducción a la programación a objetos, el uso de DFD para la representación gráfica de los algoritmos, ya que para muchos estudiantes los diagramas son más claros que las sentencias de un algoritmo en pseudocódigo en algún lenguaje de programación. La diversidad de conocimientos de entrada, pues algunos estudiantes ya vieron algún lenguaje de programación en el colegio, en cambio otros solo ven herramientas de ofimática, por lo tanto, desde el*

inicio del curso, se sienten en desventaja. En general, deficiencias en conocimientos básicos de matemáticas y lógica, por ejemplo para determinar cuándo una sentencia es verdadera o falsa, o para escribir correctamente alguna expresión aritmética.

- b. Apoyos o estrategias pedagógicas usadas: *Uso de Alice, Uso de DFD, Uso de la plataforma TEMA (Moodle) para material de estudio y ejercicios, Desarrollo de talleres en clase, donde se propicia la participación de todos, usando el tablero para aprendizaje colaborativo, construyendo soluciones con los aportes de todos (bueno, de los más participativos), Uso de material de consulta: El estudiante siempre puede tener a mano sus apuntes, libros, ejercicios, para consulta, puesto que el objetivo no es memorizar cosas, sino encontrar soluciones funcionales a distintas situaciones.*

4. Profesión: Ingeniero de Sistemas

- Estudios postgrado: Msc en Computer Science, University of South Florida (USF), Ph.D. en Computer Science, USF
- Tiempo de docencia en programación: 1.5 años
- Asignaturas dictadas relacionadas con programación: Estructura de datos, C para Ingenieros, Programación y Algoritmia
- Universidades donde ha dado estas asignaturas: University of South Florida, Universidad del Norte

a) Respuestas a las preguntas:

a. Principales aspectos que hacen compleja la enseñanza del primer curso de programación: *Dependen, en el caso de pregrado, de qué carrera se esté estudiando. Estos son los aspectos para estudiantes cuyas carreras no son en Informática: Interés en programación: esto es clave para estudiantes cuya carrera no tiene nada que ver con Informática, o Ciencias de la Computación. Hablando del caso específico de Uninorte, los estudiantes de Industrial, por ejemplo, deben tomar un curso en programación. Como muchos piensan que no lo necesitan, su poco interés por la clase hace que la enseñanza sea más compleja. Poca relación de la*

necesidad de programar, con los temas vistos por los estudiantes en sus carreras. Muchos no ven utilidad al hecho de saber programar, dado que en sus carreras no encuentran comúnmente necesidad de usar sus conocimientos en programación.

b. Apoyos o estrategias pedagógicas usadas: trato de hacer ejercicios/problemas que se relacionen con las carreras de los estudiantes. Sin embargo, es difícil hacer todos los ejercicios basados en problemas reales de cada carrera (dado que en las clases hay estudiantes de no solo una carrera, sino varias)

5. Profesión: Ingeniero de Sistemas

- Estudios postgrado: Magister en Ciencias Computacionales
- Tiempo de docencia en programación: 5 años
- Asignaturas dictadas relacionadas con programación: Fundamentos de Programación
- Universidades donde ha dado estas asignaturas: UNAB
- Respuestas a las preguntas:
 - a. Principales aspectos que hacen compleja la enseñanza del primer curso de programación: *La ubicación de ese primer curso en el plan de estudios, cuando, como en el caso de la UNAB, se encuentra en el primer semestre. El interés del estudiante por el tema, cuando en el caso de estudiantes de carreras diferentes a Ingeniería de Sistemas (en la UNAB) no encuentra una aplicación directa de la programación en su carrera, mas cuando los ejemplos no están orientados a las diferentes carreras o el estudiante no les encuentra aplicación práctica. El lenguaje de programación elegido, sobre todo cuando no ofrece una interfaz gráfica al estudiante. Las competencias iniciales del estudiante, no sólo en lógica, pensamiento algorítmico, análisis y abstracción, sino también en comprensión lectora y matemáticas.*
 - b. Apoyos o estrategias pedagógicas usadas: *Hacer más énfasis en la metodología para la resolución de problemas y el diseño del algoritmo, que en la programación. En las clases que no son en el*

laboratorio, uso de videobeam para facilitar la revisión del problema y su representación, así como la visualización de los resultados de su implementación. En los laboratorios, entrega previa de los ejercicios a desarrollar para que sea más productivo el trabajo en el laboratorio, y apoyo de un monitor que conozca el tema de manera que se pueda dar respuesta oportuna a todos los estudiantes. Autoevaluaciones periódicas, para que el estudiante identifique en qué aspectos debe mejorar. Asignación de lecturas como preparación de las clases e incentivos al "tomar la lección". Apoyo de plataforma virtual y asignación de trabajos colaborativos, programando concursos para incentivar su realización. Plan padrino para que aquellos estudiantes a los que se les facilita el tema, apoyen a uno o más compañeros. Realización de clases de refuerzo, adicionales a las establecidas en el horario normal del curso. Asignación de ejercicios relacionados con sus carreras, asignación de un proyecto de aula, con un plan de trabajo que se debe seguir durante el semestre. Realización de actividades que ayuden en el desarrollo de las competencias requeridas.

6. Profesión: Ingeniero de Sistemas

- Estudios postgrado: Maestría en Ciencias Computacionales
- Tiempo de docencia en programación: 8 años
- Asignaturas dictadas relacionadas con programación: Fundamentos de Programación, Introducción a la Programación Orientada a Objetos.
- Universidades donde ha dado estas asignaturas: UNAB, UIS.
- Respuestas a las preguntas:
 - a) Principales aspectos que hacen compleja la enseñanza del primer curso de programación: *A mi modo de ver, el primer factor de complejidad se encuentra en la materia de estudio 'per se'. La programación obliga el desarrollo de un tipo especial de inteligencia que le permita al humano manejar el lenguaje de la programación. Esto exige desarrollar competencias en la aplicación de la lógica y la algoritmia.*

El segundo factor se encuentra en la falta de competencia de los estudiantes en áreas como la lectura y la matemática. La comprensión de lectura como herramienta básica para la asimilación de los problemas y el análisis que se hace de ellos. Las matemáticas, como fundamento para la resolución de problemas para los cuales es factible desarrollar soluciones por medio de algoritmos.

Aspectos como la motivación, la organización de los contenidos y las estrategias pedagógicas deberían formar parte de la solución. Estos elementos son los que deben pensarse, probarse, evaluarse y adaptarse para constituirse en elementos de solución. Cuando no se hace, entonces contribuyen a ser parte del problema de aprendizaje.

Un factor importante en la complejidad se encuentra en el docente. En algunos casos los docentes son recién egresados con poca experiencia o en programación o en docencia. Lo ideal es que se posean las dos, aunque alguien con buena experiencia en programación puede desarrollar más fácilmente la 'pedagogía' (yo diría más bien la 'didáctica') que se requiere para elaborar un buen proceso de enseñanza-aprendizaje de la programación.

Otro aspecto que incide es la administración de la academia. Ya que aunque quienes administran lleguen a conocer cuáles son las últimas tendencias y paradigmas, ignoran aspectos como la curva de aprendizaje que posee cada paradigma. No ven que los docentes deben asimilar el paradigma, el lenguaje, el uso de la herramienta, las técnicas de programación, y aspectos como la depuración de programas y documentación, que requieren estudio y experimentación con el fin de que los docentes tengan 'capacidad de respuesta' a las múltiples inquietudes que surgen en el proceso en los estudiantes.

b) Apoyos o estrategias pedagógicas usadas: Mi estrategia general es motivar a los estudiantes mediante dos ideas principales: la primera es que el programador es irremplazable por la maquina. Se pueden construir programas que traduzcan algoritmos elaborados en diagramas de flujo que generen programas en lenguajes de alto nivel, pero no que resuelvan el problema. Es decir, la elaboración del algoritmo es donde somos irremplazables. Este hecho nos garantiza un área de desempeño importante largo placista.

La segunda es convertir la actividad en algo 'tan lúdico como sea posible'. Esto se logra mediante el desarrollo de programas para algunos juegos triviales que motiven al estudiante el deseo de ver su programa como un instrumento que se puede usar para jugar.

Existen algunos motivadores extra como las competencias de programación, donde quienes tengan habilidades extraordinarias reciben puntos extras por sus logros.

En cuanto a los apoyos, me gusta enfatizar el uso de tutoriales y libros. Los tutoriales permiten obtener puntos de vista adicionales que les ayuden a encontrar el enfoque que les funcione. Los libros les ayudan a afirmar los aspectos formales de la programación.

7. Profesión: Ingeniero de Sistemas, Docente Tiempo Completo

- Estudios de postgrado: Maestría en Ciencias de la Computación, Master in Computer Engineering y Phd in Computer Science
- Tiempo de docencia en programación: 16 años
- Asignaturas dictadas relacionadas con programación: Programacion Móvil, Programación bajo la Web, Tópicos especiales en Java, Fundamentos de Programación I, Fundamentos de Programación II, Estructura de Datos II, Proyectos de Grado.
- Universidades donde ha dado estas asignaturas: Universidad del Norte, Corporación Universitaria de la Costa (CUC), Politécnico Costa Atlántica
- Respuestas a preguntas:
 - a. Principales aspectos que hacen compleja la enseñanza del primer curso de programación: Uno de los principales problemas que he encontrado, y es en cuanto a contenidos, es que el estudiante muchas veces no ha visto asignaturas como calculo I y algebra lineal donde existen temas relacionados con sumatorias, vectores o matrices los cuales son fundamentales para el mejor entendimiento de los conceptos que se explican en la segunda parte del curso donde empiezan a trabajar con ciclos repetitivos. Por otra parte, en lo referente a estrategias pedagógicas, no se cuenta con alguna herramienta visual que le ayude al estudiante al mejor

entendimiento de los temas impartidos. Cuando hablo de una herramienta visual me refiero por ejemplo a una herramienta visual que desplegara en forma grafica por ejemplo las famosas pruebas de escritorio que se le hacen a los algoritmos para poder entender su funcionamiento.

- b. Apoyos o estrategias pedagógicas usadas: Básicamente nos apoyamos con un tutor el cual maneja 2 horas adicionales fuera de clases (aunque no es obligatorio que el estudiante las vea), en donde se coordina el desarrollo de ejercicios adicionales que le ayuden a reforzar los conceptos explicados en clases. Por otro lado, tenemos 2 horas de laboratorio en la asignatura (método tradicional en las universidades) en donde también se desarrollan ejercicios ya implementados en C++ como refuerzo a lo expuesto en clases. Dentro de las clases cada vez que se explica un nuevo tema se desarrollan ejemplos los cuales se presentan con varias soluciones (por ejemplo un algoritmo se resuelve de 3 y 4 maneras), en donde al estudiante se le explica que existen diferentes formas de enfocar la solución, con esto intentamos brindarle al estudiante varias alternativas de solución

8. Profesión: Ingeniera de Sistemas

- Estudios postgrado: estudios de maestría
- Tiempo de docencia en programación: 14 años
- Asignaturas dictadas relacionadas con programación: Introducción a la programación, fundamentos de programación, algoritmia, modelos de programación, programación I
- Universidades donde ha dado estas asignaturas: UDES, UIS, y UTS
- Respuesta a preguntas:
 - a. Principales aspectos que hacen compleja la enseñanza del primer curso de programación: *La metodología de enseñanza que traen los estudiantes es muy mecánica sólo aplicación de fórmulas y no se hace análisis del por qué, poco desarrollo de la lógica (considero que es la más importante), son muy mecánicos a pesar que saben y*

entienden como realizar las cosas, al describirlas les cuesta mucho trabajo realizar paso por paso. A pesar de que hemos venido evolucionando en los contenidos considero que todavía hay población donde no se imparte la programación con anterioridad, o sea no tienen conocimientos previos lo cual en parte para mi es positivo; porque si tuvieron experiencias negativas vienen con una indisposición hacia el tema. Se imparten los conocimientos sin aplicabilidad y eso es lo que genera incertidumbre y aparece la pregunta muy usual ¿para qué me sirve esto a mi?

- b. Apoyos o estrategias pedagógicas usadas: *En el aula: ejercicios de la vida real, ejercicios ya resueltos, seguimiento a código. Fuera del aula: Elaboración de ejercicios de la disciplina y seguimiento a código.*

ANEXO 4 RESPUESTAS DE ESTUDIANTES EN LA ENCUESTA EVALUACIÓN HERRAMIENTA

1. ¿Considera usted que la educación virtual promueve el aprendizaje autónomo?

Total global Porcentaje Gráfico

Sí	<u>21</u>	95.45 %	
No	<u>1</u>	4.55 %	
Total	22		

2. ¿Considera usted que la educación virtual es una alternativa útil para reforzar lo aprendido en clase?

Total global Porcentaje Gráfico

Sí	<u>22</u>	100 %	
No		0 %	
Total	22		

3. ¿Considera tiene igual validez el acompañamiento docente virtual que el presencial?

Total global Porcentaje Gráfico

Sí	<u>4</u>	18.18 %	
No	<u>18</u>	81.82 %	
Total	22		

4. Según la siguiente escala, ¿cuál considera que es la evaluación general de la herramienta?

Total global Porcentaje Gráfico

Malo		0 %	
Regular	<u>2</u>	9.09 %	

Bueno	<u>16</u>	72.73 %	
Excelente	<u>4</u>	18.18 %	
Total	22		

5. Según la siguiente escala, ¿cómo evalúa el apoyo de la herramienta a su aprendizaje de la programación orientada a objetos?

Total global Porcentaje Gráfico

Malo		0 %	
Regular	<u>4</u>	19.05 %	
Bueno	<u>11</u>	52.38 %	
Excelente	<u>6</u>	28.57 %	
Total	21		

6. ¿Considera que la herramienta posee suficientes estrategias pedagógicas para hacer del aprendizaje una experiencia agradable?

Total global Porcentaje Gráfico

Sí	<u>19</u>	86.36 %	
No	<u>3</u>	13.64 %	
Total	22		

7. ¿Quisiera seguir su aprendizaje con el apoyo de esta herramienta a través de todos los temas?

Total global Porcentaje Gráfico

Sí	<u>16</u>	72.73 %	
No	<u>6</u>	27.27 %	
Total	22		

8. ¿El aprendizaje con la herramienta le parece más atractivo que el aprendizaje tradicional en el aula de clase?

Total global Porcentaje Gráfico

Sí	<u>10</u>	45.45 %	
No	<u>12</u>	54.55 %	
Total	22		

9. ¿Considera usted que el examen realizado en la herramienta permite evaluar efectivamente lo que se aprendió?

Total global Porcentaje Gráfico

Sí	<u>17</u>	77.27 %	
No	<u>5</u>	22.73 %	
Total	22		

10. ¿La nota obtenida en el examen le permitió identificar debilidades en su aprendizaje?

Total global Porcentaje Gráfico

Sí	<u>15</u>	71.43 %	
No	<u>6</u>	28.57 %	
Total	21		

11. ¿La nota obtenida en el examen le permitió identificar debilidades en su aprendizaje?

Total global Porcentaje Gráfico

Sí	<u>15</u>	68.18 %	
No	<u>7</u>	31.82 %	
Total	22		

12. ¿Considera usted que el trabajo colaborativo o cooperativo propuesto a través de las herramientas de foro, blog y wiki aportan a su aprendizaje?

Total global Porcentaje Gráfico

Sí	<u>20</u>	90.91 %	
No	<u>2</u>	9.09 %	
Total			

13. ¿Utilizó éstas herramientas de trabajo colaborativo?

Total global Porcentaje Gráfico

Sí	<u>17</u>	77.27 %	
No	<u>5</u>	22.73 %	
Total	22		

14. ¿Le fue sencillo Interpretar la tarea a realizar propuesta en el ejercicio de aprendizaje propuesto?

Total global Porcentaje Gráfico

Sí	<u>11</u>	50 %	
No	<u>11</u>	50 %	
Total	22		

15. ¿Cree usted que la elaboración de un proyecto (desarrollado en su primera parte en esta etapa: Abstracción), le facilita la comprensión de cada una de las etapas de un desarrollo completo?

Total global Porcentaje Gráfico

Sí	<u>19</u>	90.48 %	
No	<u>2</u>	9.52 %	
Total	21		

16. ¿Cree usted que la elaboración de un proyecto (desarrollado en su primera parte en esta etapa: Abstracción), le facilita la comprensión de cada una de las etapas de un desarrollo completo?

Total global Porcentaje Gráfico

Sí	<u>19</u>	90.48 %	
No	<u>2</u>	9.52 %	
Total	21		

17. ¿Qué actividades considera que debería realizar el docente de manera virtual para apoyar la efectividad de la herramienta e-learning?

- E-book
- Pues ninguna me gusto como se trabajo en clase.
- mas talleres y conceptos que nos ayuden en nuestro aprendizaje
- Algo más dinámico, que incentive a los estudiantes participar.
- pienso q este último tema
- mejorar el sistema de audio.
- pues tener un seguimiento pero de igual forma la presencial es muy importante ya que se tiene mas disciplina
- diferentes ejercicios para realizar prácticas de lo enseñado
- Mas organización de la estructura, pero en general muy bien.
- Actividades divertidas como juegos interactivos que evalúan y facilitan el aprendizaje.

18. ¿Considera sencilla de utilizar el entorno de la herramienta (la plataforma)?

Total global Porcentaje Gráfico

Sí	<u>22</u>	100 %	
No		0 %	
Total	22		

19. ¿Fue rápido y fácil ubicar los componentes propuestos para su aprendizaje?

Total global Porcentaje Gráfico

Sí	<u>18</u>	85.71 %	
No	<u>3</u>	14.29 %	
Total	21		

20. ¿Considera pertinente el material seleccionado para el desarrollo del tema?

Total global Porcentaje Gráfico

Sí	<u>20</u>	95.24 %	
No	<u>1</u>	4.76 %	
Total	21		

21. ¿Cuál de los siguientes recursos didácticos tuvo mejor recepción en usted?

Total global Porcentaje Gráfico

Audios	<u>3</u>	14.29 %	
Lecturas	<u>11</u>	52.38 %	
Videos	<u>7</u>	33.33 %	
Total	21		

22. ¿Qué tipo de material hubiera preferido encontrar en la herramienta?

- Ninguna
- videos
- Todo estuvo bien, salvo a que la videoconferencia estuvo un poco "aburrida" ya que desde mi punto de vista no se entendió bien.
- todos estuvieron muy bueno no solo los videos sino lecturas y audios
- el material fue bueno a excepción del audio
- juegos didácticos con la información evaluada.
- Historietas cómicas y juegos estratégicos que me facilita mucho más el entendimiento.

23. ¿Tuvo inconvenientes con las lecturas en inglés?

Total global Porcentaje Gráfico

Sí	<u>10</u>	47.62 %	
No	<u>11</u>	52.38 %	
Total	21		