

Diseño de una aplicación para enseñar programación a estudiantes universitarios usando la metodología incremental y el modelo de diseño instruccional ADDIE

[Designing an application that teaches programming to college students using incremental methodology and instructional design model ADDIE]

René López Caballero, María Guadalupe Medina Barrera, and José Juan Hernández Mora

Maestría en Sistemas Computacionales,
Instituto Tecnológico de Apizaco,
Apizaco, Tlaxcala, México

Copyright © 2016 ISSR Journals. This is an open access article distributed under the *Creative Commons Attribution License*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: The aim of this paper is to describe a methodology used to create an application that teaches programming to college students. The instructional design model and incremental software development model are used as a base. The instructional design model is a general methodology for the development of educational materials. This methodology is composed by 5 phases: Analysis Design, Development, Implementation and Evaluation, commonly referred as ADDIE. We use the software incremental methodology, this software methodology divide software development into increments, in each increment, a functional software is developed and can be tested, for each increment will use the 5 phases of instructional design presented before.

The software teaches programming to college students, detecting the students learning style and providing a personalized experience, we use the visual programming paradigm. This application is developed in a web environment, using as tool set: HTML, Javascript, Linux, Apache, MySQL and PHP.

At the end of this work we have realized that the incremental software development methodology and the incremental development model is a good combination in the development of educational software focused on the web.

KEYWORDS: software, methodology, programming, incremental, development.

1 INTRODUCCIÓN

En la enseñanza se ha pasado del conductivismo al constructivismo, donde la educación se basa en que el alumno genere su propio conocimiento y no en que el sea solo un ente pasivo que recibe la información del profesor. “Hay que pasar del enfoque en donde el alumno solo memoriza la información, ya que en la actualidad los conocimientos se actualizan constantemente, al enfoque donde los estudiantes son capaces de aprender por ellos mismos y así siempre estar a la par de los cambios tecnológicos.”[1]

Como consecuencia de esto, surge la educación basada en las competencias. Las competencias de acuerdo a la UNESCO son: “el conjunto de comportamientos socioafectivos y habilidades cognoscitivas, psicológicas, sensoriales y motoras que permiten llevar a cabo adecuadamente un desempeño, una función una actitud o una tarea”[2]. Esto hace que tengamos la necesidad de capacitar alumnos que sean competentes, en las carreras de tecnologías de la información una de las competencias mas importantes es la de la programación.

No todos las personas aprenden de la misma manera ya que estamos influenciados por diversos factores como pueden ser las condiciones culturales, la edad, el bagage cultural, si nos gusta trabajar solos o acompañados, etc. Debido a esto se supone que una educación personalizada nos ayudará a que el alumno tenga mejor aprovechamiento. Esto nos hace

plantearnos la necesidad de crear un programa que enseñe programación de forma personalizada a los alumnos. Existen varias formas de clasificar estos estilos de aprendizaje dependiendo de diversos autores. Para este trabajo nos basaremos en el modelo bipolar creado por Felder y Silverman, en donde se dividen los estilos de aprendizaje en 5 duplas contrapuestas: Activos o reflexivos, sensitivos o intuitivos, visuales o verbales, secuenciales o globales e inductivos o deductivos: [3].

Con la llegada del internet a nuestras vidas, la educación ha cambiado para bien. El conocimiento que antes era restringido a aquellas personas que tenían acceso a las bibliotecas, ahora está disponible a cualquier persona que use internet desde una computadora, una tableta o un teléfono celular. En internet se pueden acceder a una infinidad de contenidos, lo que en sus inicios eran solo documentos de texto, con algunas imágenes y un formato básico, se han transformado en paginas multimedia, wikis, blogs, vídeos, teleconferencias. Podemos encontrar desde información básica como inflar una llanta, aprender calculo, infinidad libros electrónicos, transiciones en vivo, cursos, conferencias, hasta acceder todo el contenido de cursos de reconocidas universidades como Standford y MIT.

Jenkins, Browne y Walker describen a una plataforma e-learning, plataforma educativa web o entorno virtual de enseñanza y aprendizaje como una aplicación web que integra un conjunto de herramientas para la enseñanza-aprendizaje en línea, permitiendo una enseñanza no presencial (e-learning) y/o una enseñanza mixta (b-learning), donde se combina la enseñanza en Internet con experiencias en la clase presencial [4].

Debido a la cercanía con los Estados Unidos, México ese un lugar ideal para desarrollar software para el país vecino y tenemos la oportunidad de ser competitivos en esta área.

De acuerdo al A.T. Kearney Global Services Location Index 2014 [5] México ha pasado del sexto lugar en 2011 al cuarto lugar en 2014 en servicios de Offshoring con los Estados Unidos, detrás de India, China y Malasia. Si México cuenta con un profesionales altamente calificados en el área de programación estos pueden aprovechar las ofertas laborales con los Estados Unidos.

En la actualidad muchos alumnos egresan con un conocimiento básico en el área de programación, por lo que existe la necesidad de generar profesionistas con competencias desarrolladas en el área de programación de computadoras y el desarrollo de sistemas automáticos. Esto logrará que la región sea más competitiva y productiva en las TIC y esto beneficiara a la población.

Diversos métodos han sido utilizado para enseñar programación a las personas: usando diagramas de flujo [6], enseñando a los niños a crear sus propios videojuegos [7], manipulando un robot virtual [8], usando facebook como medio de resolver dudas [9], etc. El metodo mas usado actualmente es el que usa el paradigma de la programación visual, sitios web como *scratch*, *code.org*, *tinker*, *app inventor*, hacen uso de este, pero no encontramos ninguno que ofrezca enseñanza personalizada dependiendo del estilo de aprendizaje del usuario.

El presente trabajo surge, debido a la necesidad de desarrollar un software educativo, que sea un auxiliar en la enseñanza de la programación para estudiantes universitarios. Se desarrolló como una aplicación web, con la finalidad de que pueda ser utilizado por cualquier estudiante con acceso a internet. El software será instalado en el Instituto Tecnológico de Apizaco y del Instituto Tecnológico del Altiplano de Tlaxcala para que pueda ser utilizado por sus alumnos y profesores.

El software detecta el estilo de aprendizaje del usuario y dependiendo de esto le proporciona una enseñanza personalizada, para esto utilizaremos el paradigma de programación visual, permitiendo que el usuario, pueda arrastrar bloques para realizar sus programas. La programación visual nos permite que el usuario aprenda la lógica de la programación sin preocuparse de los errores de sintaxis. La programación visual es una excelente primera aproximación a la programación antes de aprender algún lenguajes de programación tradicional.

Ya que nuestro software, es un software educativo, se utilizó como base a la metodología de diseño instruccional ADDIE (Análisis, diseño, desarrollo, implementación y evaluación) [10], la cual esta diseñada para desarrollo de material educativo.

En cuanto a la metodología de software, utilizamos la metodología de de desarrollo incremental. En ésta, el desarrollo del software está dividido en incrementos. Dentro de cada una de estos incrementos se utiliza la metodología ADDIE (Análisis, diseño, desarrollo, implementación y evaluación) adaptada al desarrollo de software.

2 MODELO ADDIE

“Para Berger y Kam el diseño instruccional es la ciencia de creación de especificaciones detalladas para el desarrollo, implementación, evaluación, y mantenimiento de situaciones que facilitan el aprendizaje de pequeñas y grandes unidades de contenidos, en diferentes niveles de complejidad.” [10]

El modelo ADDIE es un acrónimo generado a partir sus cinco fases, las cuales son: Análisis, Diseño, Desarrollo, Implementación y Evaluación:

1. Análisis: En esta etapa se analiza las necesidades de los alumnos
2. Diseño: En esta etapa se define la serie de materiales y como se darán
3. Desarrollo: En esta fase lleva a cabo la creación del software
4. Implementación: Se implementa el software para ser usado por los alumnos
5. Evaluación: Se hace una evaluación del producto

Este proceso al que a veces nos referimos como ADDIE puede modificarse para adaptarse a cualquier situación, tema, público y modelo de formación. [11]

3 METODOLOGÍA DESARROLLO INCREMENTAL

En la metodología de desarrollo incremental, se crea una primera versión del software, después se le pide retroalimentación al usuario y se van realizando entregas incrementales, hasta llegar a la versión final del software.

Durante cada versión del software se van añadiendo las características que el cliente necesita, por lo general el primer incremento consta de funcionalidades básicas mas importantes o las mas urgentes.

Cada incremento en esta metodología de software es un producto que funciona y puede ser probado.

Algunos beneficios de desarrollo incremental son:

- Se reduce el costo de adaptar nuevos requerimientos. La cantidad de documentos que tiene que rehacerse es mucho menor
- Es mas fácil obtener retroalimentación del cliente, ya que puede resultar engorroso para el cliente leer manuales técnicos
- Se puede entregar versiones funcionales al cliente, antes de que el producto este 100% terminado

Los modelos de desarrollo incremental no son recomendados para software grande y complejo, pero sí para la mayoría de sistemas empresariales, de comercio electrónico y personales.[12],[13]

4 MODELO PROPUESTO

Nuestra propuesta de metodología, es tomar las 5 fases del modelo ADDIE: Análisis, diseño, desarrollo, implementación y evaluación e implementarlas en cada en cada uno de los incrementos, de una metodología tradicional de desarrollo de software incremental.

En esta metodología podemos dividir nuestro software en diversos incrementos, en donde cada uno, sera un producto funcional, que podremos probar, antes de continua con el siguiente incremento.

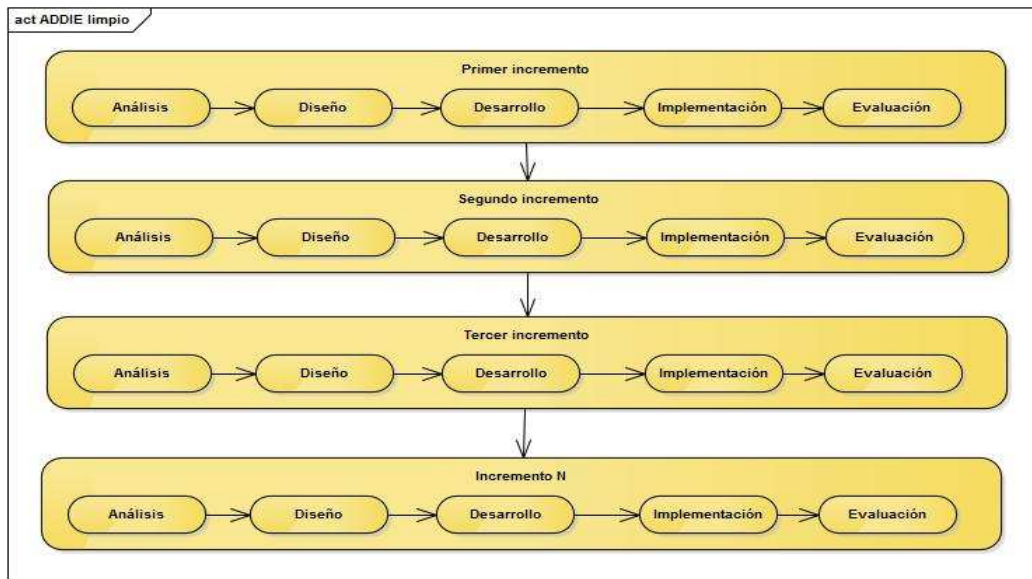


Fig. 1. Modelo de desarrollo propuesto

5 DISEÑO DE LA APLICACIÓN PARA ENSEÑAR PROGRAMACIÓN A ESTUDIANTES UNIVERSITARIOS

Nuestro software, como ya lo hemos mencionado, es un software educativo, auxiliar en la enseñanza de la programación para los alumnos del Instituto Tecnológico de Apizaco y del Instituto Tecnológico del Altiplano de Tlaxcala. En este capítulo describiremos la metodología de desarrollo usada en nuestro software.

Ya que nuestro software, es un software educativo, tomaremos como base a la metodología de diseño instruccional ADDIE (Análisis, diseño, desarrollo, implementación y evaluación), la cual está diseñada para desarrollo de material educativo.

En cuanto a la metodología de software, usaremos la metodología de desarrollo incremental. En ésta el desarrollo del software está dividido en incrementos, dentro de cada una de estos incrementos usaremos la metodología ADDIE (Análisis, diseño, desarrollo, implementación y evaluación) adaptada al desarrollo de software.

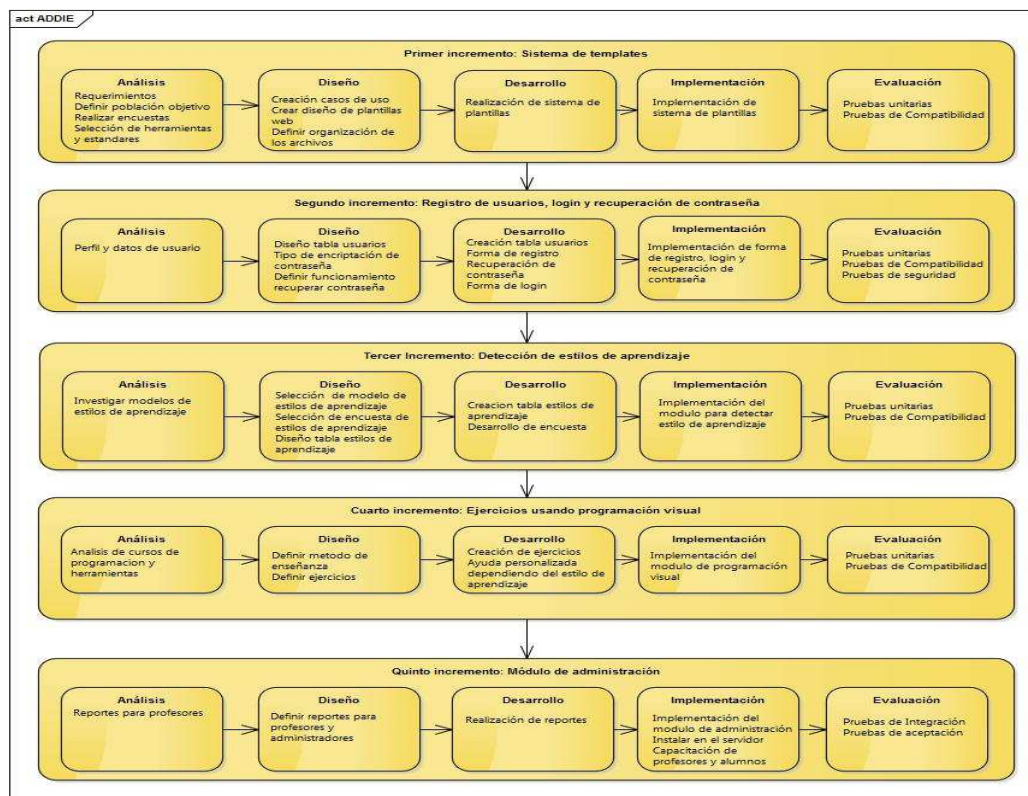


Fig. 2. Metodología de la aplicación para enseñar programación a estudiantes universitarios

5.1 PRIMER INCREMENTO

En la metodología ADDIE una de las partes mas importantes, es determinar si es que nuestro programa informático será adecuado para nuestra población objetivo, con motivo de esto, como primer paso se realizó una encuesta para ver si nuestros usuarios: encuentran difícil la programación, así como para ver si cuentan con los medios tecnológicos para usar el sistema.

Se encuestaron un total de 132 estudiantes, que van de los 17 a los 24 años, de los cuales 74 fueron hombres y 58 mujeres, se encuestó a alumnos que se encuentran entre el primero y el séptimo semestre, los resultados de esta encuesta, se muestran en la Tabla 1.

Tabla 1: Percepción de la programación para los alumnos

	Muy Fácil	Fácil	Normal	Difícil	Muy Difícil
Programación	0%	8%	47%	43%	2%
Variables	4%	14%	45%	35%	2%
Condicionales	6%	25%	40%	27%	2%
Ciclos	5%	17%	42%	33%	4%
Matrices	1%	8%	24%	60%	7%
Funciones	1%	11%	36%	46%	7%
Objetos	0%	15%	30%	48%	8%

Después de validar con encuestas la viabilidad de nuestro software, se procedió a recabar los requerimientos de nuestro programa, tanto funcionales como no funcionales. Con base en nuestros requerimientos se realizó un diagrama de los casos de uso de nuestro software.

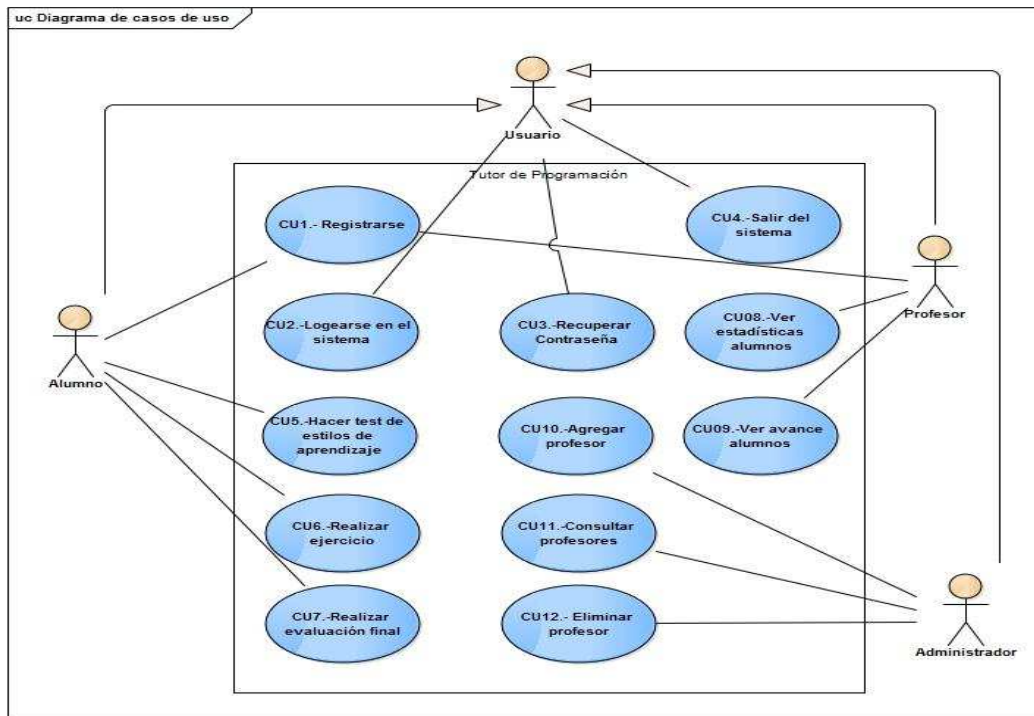


Fig. 3. Casos de uso de la aplicación

Para poder llevar a cabo un correcto desarrollo, se definió qué tipo de aplicación desarrollaremos y qué herramientas utilizaremos. Nuestra aplicación es una webapp, para poder ser accesada fácilmente por un mayor número de alumnos. Las herramientas elegidas para el desarrollo son: Linux, Apache, Mysql y PHP, conjunto de herramientas mejor conocido como LAMP, también utilizamos HTML, Javascript así como las librerías Bootstrap, librería de CSS, Blockly librería de programación visual y AngularJS librería para creación de HTML dinámico.

Para que nuestro código sea legible y estandarizado se definieron los estándares de codificación a seguir, como se nombran archivos, como nombrar de variables, clases, indentación etc.

Para tener una mejor organización de los archivos se definió la estructura de directorios, cada archivo se agrupo de acuerdo a su tipo, img (imagenes), js (javascript), templates (templates html) y css (hojas de estilo en cascada).

Para poder implementar el mismo software en dos universidades, pero cada una con propio *look and feel*, se creo un sistema de *templates*, donde con el cambio de una simple variable de la configuración, se puede cambiar el encabezado y pie de nuestro sitio.

5.2 SEGUNDO INCREMENTO

Como segundo incremento se desarrolló el sistema de registro de usuarios, recuperación de la contraseña del usuarios y la autenticación de usuario al sistema.

El primer paso fue el de establecer que datos se requieren almacenar para cada usuario de nuestro sistema. Con esto se creo la tabla usuarios, la cual almacenar toda la información relacionada con estos.

Antes de realizar la programación, creamos un diagrama de actividad para cada una de las funcionalidades mencionadas previamente. En la figura 4 mostramos el diagrama de actividad del registro de usuarios, este tipo de diagramas se realizaron para cada una de las funcionalidades del sistema, en este documento solo mostramos uno como ejemplo.

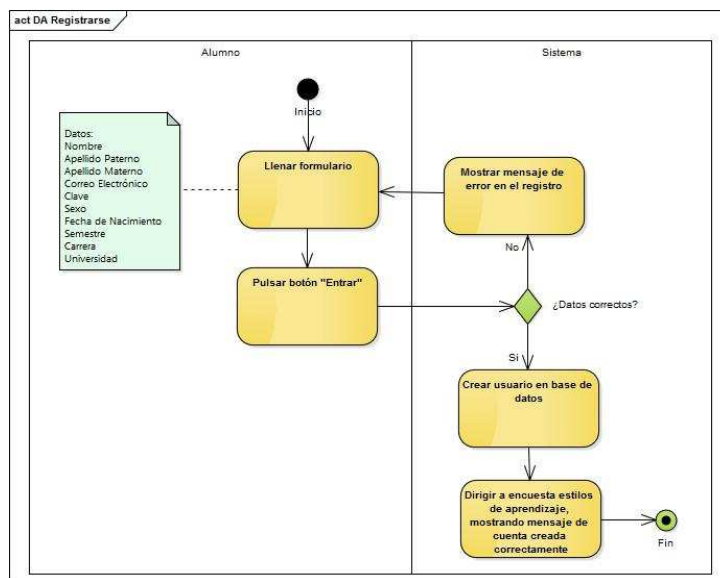


Fig. 4. Diagrama de actividad del registro de alumnos

Primero se llevó a cabo el registro de alumnos, ya que un requerimiento del sistema es que para poder usar el sistema el alumno deberá primero obtener una cuenta. Después de eso se realizó el sistema de recuperación de contraseña, el cual permitirá a un usuario que haya perdido su clave acceder al sistema, el usuario ingresara su contraseña, el sistema enviara un correo que contendrá una liga especial, la cual el usuario deberá hacer click, para verificar la pertenencia de el correo electrónico, una vez hecho esto, el usuario podrá establecer una nueva contraseña. Como ultimo paso de este incremento se desarrolló el sistema de login, ya que para poder acceder al sistema el usuario debe de proporcionar su correo electrónico y contraseña.

5.3 TERCER INCREMENTO

El tercer incremento de nuestra aplicación es el modulo de detección de estilos de aprendizaje. Como primer paso se realizó una investigación de los diferentes estilos de aprendizaje, que cuestionarios existen y si son libres de usar en nuestra aplicación. De los modelos existentes analizamos: el modelo PNL, el modelo de David Kolb, El modelo de Felder y Silverman y el modelo de Nad Herman. Hemos elegido el modelo de Fielder-Silverman, ya que este modelo, divide los estilos de aprendizaje en cinco: Activos o reflexivos, sensitivos o intuitivos, visuales o verbales, secuenciales o globales, inductivos o deductivos. De estos estilos utilizaremos el de visuales o verbales para saber que tipo de ayuda proporcionar al usuario y el de secuenciales o globales, para saber como prefieren la información los usuarios, si es que prefieren la información poco a poco, o toda a la vez. Este modelo aparte nos provee de un cuestionario ya diseñado el cual es libre de uso para aplicación educativas, por lo que lo pudimos utilizar sin problemas en nuestra aplicación.

Una vez elegido el modelo de estilos de aprendizaje a utilizar, se creó la base de datos, la cual tiene que almacenar las respuestas del usuario, y el estilo de aprendizaje calculado, y se procedió a la realización del modulo, que aplicara el cuestionario Felder-Silverman al usuario antes de que este empiece a usar el sistema

5.4 CUARTO INCREMENTO

Para nuestro cuarto incremento, analizamos diversos cursos en linea gratuitos de la plataforma MIT Open Courseware, para darnos cuenta cual es la forma actual en la que se esta enseñando programación, de esta manera, pudimos obtener, los ejercicios que aplicaremos a los alumnos, para que ellos aprendan practicando.

Una vez definidos los ejercicios para ser realizados por los alumnos, procedimos al modulo de ejercicios el cual utiliza el paradigma de programación visual, esto lo logramos gracias a la librería Blockly, una librería de código libre, desarrollada por google, que nos permite implementar la programación visual de una manera sencilla. Blockly nos permite generar código Javascript, PHP y Python a partir de la programación visual, de esta manera los estudiantes podrán ir observando la analogía de la programación visual, y la programación tradicional e ir aprendiendo a su ritmo.

Nuestro sistema muestra ayuda personalizada dependiendo del estilo de aprendizaje del alumnos, para los alumnos verbales se utiliza texto y para los visuales se utiliza vídeo.

5.5 QUINTO INCREMENTO

Como ultimo incremento de nuestra aplicación se definieron los reportes necesarios para los profesores. Se realizo un reporte general donde el profesor pueda ver el avance general de su grupo y un reporte por usuario con información mas detallada, en esto los profesores, pueden ver si los alumnos están utilizando la herramienta, que ejercicios han realizando, y observar si los alumnos tienen dificultad en alguno en especifico.

6 CONCLUSIONES

- La metodología de diseño incremental, nos permite dividir el desarrollo de nuestro software en diferentes módulos, cuando el modulo es independiente de los otros, se puede trabajar en paralelo en diferentes módulos y de esa forma agilizar el proceso de desarrollo.
- La metodología de desarrollo incremental resulta un modelo que puede ser adaptado fácilmente a cada uno de los incrementos de una metodología de desarrollo incremental.
- Existe una gran cantidad de programas que enseñan programación, utilizando el paradigma de programación visual, pero no encontramos alguno que sea enfocado a estudiantes universitarios o que detecte el estilo de aprendizaje.
- La enseñanza personalizada es muy difícil de lograr en el salón de clases, por la gran cantidad de alumnos que pueden tener los grupos, pero utilizando software podemos cubrir esta necesidad.
- Para futuros trabajo, se sugiere una aplicación que utilice un lenguaje como processing, el cual se basa en JAVA, pero con una sintaxis simplificada.
- Las aplicaciones web resultan útiles cuando el software debe ser utilizado por personas con diferentes dispositivos y sistemas operativos.

REFERENCIAS

- [1] Maribel Aragón García y Yasmín Ivette Jiménez Galán, "Diagnóstico de los estilos de aprendizaje en los estudiantes: Estrategia docente para elevar la calidad educativa," *Revista de Investigación Educativa*, 2009.
- [2] Bertha Guadalupe Paredes Zepeda and Gabriel Márquez Ramírez, "Educación Superior basada en Competencias," MEMORIAS DEL VI FORO DE ESTUDIOS EN LENGUAS INTERNACIONAL, 2010
- [3] Aman Consulting, "Felder and Silverman's Index of Learning Styles," 2013.
- [4] Ana Fernández-Pampillón Cesteros, "Las plataformas e-learning para la enseñanza y el aprendizaje universitario en Internet," *Universidad Complutense de Madrid*, 2009.
- [5] "The 2014 A.T. Kearney Global Services Location Index," 2014.
- [6] Terry Wilson, Martin C. Carlisle, Jeff Humphries y Jason Moore "RAPTOR - Flowchart Interpreter," 2006.
- [7] Krishna Vedati, "Tinker," 2013.
- [8] André Roberge, "Learning Python: Child's Play with RUR-PLE!," 2004.
- [9] Chien-Hung Lai, Wei-Ching Lin, Bin-Shyan Jong y Yen-Teh Hsia, "Java Assist Learning System for Assisted Learning on Facebook," 2013.
- [10] Consuelo Belloch, "Diseño Instruccional," *Unidad de Tecnología Educativa(UTE). Universidad de Valencia*, 2010.
- [11] Peter Williams, Lynne Schrum, Albert Sangrà y Lourdes Guàrdia, "Fundamentos del diseño técnico-pedagógico en e-learning," 2004.
- [12] Roger S. Pressman, "Ingeniería del software, UN ENFOQUE PRÁCTICO," 2010.
- [13] Ian Sommerville, "Ingeniería de software," *9nd Ed.* 2011.