

PROGRAMACIÓN DE COMPUTADORES, LIDERAZGO Y FORMACIÓN DE EMPRENDEDORES EN EL AULA

LUIS ALEJANDRO BERNAL ROMERO *

RESUMEN

Los Proyectos de Programación son una estrategia docente para motivar tanto la cultura del desarrollo de software como la formación de emprendedores. Estos proyectos son llevados a cabo por un grupo de estudiantes en el contexto de una materia de programación y pueden hacerse extensivos a otras materias de la carrera de Ingeniería de Sistemas. Los proyectos fomentan las buenas prácticas de programación, el uso de la Ingeniería del Software y la creación de empresa en el sector informático.

*Ingeniero de Sistemas, Universidad Central.
Magister (candidato) en Ingeniería de Sistemas y Computación, Universidad de Los Andes.
Docente Tiempo completo, Escuela de Administración de Negocios E.A.N.
E-mail: lbernal@uvirtual.ean.edu.co

INTRODUCCIÓN

El proceso de desarrollo de software es altamente complejo. Por ello, existen técnicas, herramientas, metodologías, etc. que conjuntamente se llaman Ingeniería de Software y ayudan a reducir la complejidad de esta tarea. Tanto enseñar a desarrollar software como enseñar Ingeniería de Software de una manera efectiva no son tareas simples. El presente documento describe una metodología para aprender a hacer software haciendo software, inspirado en el método de proyectos que se ha denominado Proyectos de Programación; una manera excelente de aprender haciendo.

Como se verá, los proyectos son una buena base para el desarrollo de las competencias que todo emprendedor necesita, tales como la creación de nuevos productos (innovación), su presentación ante un público (ya sea un grupo de inversionistas o clientes), la creación de imagen corporativa y de producto, las habilidades propias de la planeación, dirección y gestión de proyectos, y por supuesto la creación de empresas.

Este documento recoge el resultado de la experiencia del autor, después de aplicar la metodología de Proyectos de Software durante varios años en materias que van desde Introducción a la Programación hasta el Diseño e Implementación de Compiladores pasando por la Programación Orientada a Objetos, materia en la que adquirió su forma actual.

1. ¿POR QUÉ?

El desarrollo de software¹ es uno de los sectores de más rápido crecimiento en el mundo y en el país. Para muchos es claro que existe una necesidad acuciante de software. Esto es porque la computación y específicamente el software es el que puede marcar la diferencia competitiva de una empresa. Pressman afirma al respecto:

En esencia, el software es normalmente el factor que marca la diferencia. La suficiencia y oportunidad de la información dada por el software diferencia a una compañía de sus competidoras. El diseño de un producto software "amigable a los humanos" lo diferencia de los productos competidores que tengan unas funciones similares. La inteligencia y función dada por el software empotrado² distingue normalmente a dos productos in-

dustriales o de consumo similares. Es el software el que marca la diferencia. [Pressman89, pag. 1]

Y esto se hace más evidente, si pensamos que estamos entrando en una era en la cual la información es la base del poder, como dice Toffler [Toffler95], y son los computadores los que nos ayudan a manejarla. El software es el alma, el centro donde gira el mundo post-moderno [Turkle95]³. Todo esto lleva a una conclusión: las oportunidades de negocios del mundo actual están en el sector Informático y fundamentalmente en el desarrollo de software.

En la Facultad de Ingeniería de la Escuela de Administración de Negocios EAN se ha pensado al egresado como un Ingeniero de Software, y específicamente como un emprendedor del sector Informático. En respuesta a esto el autor ha diseñado, a partir de su experiencia tanto en desarrollo de software como en la enseñanza de la programación, una metodología que ayuda al desarrollo de las habilidades de este emprendedor del software, la cual se denomina, Proyectos de Programación.

2. ¿EN QUÉ CONSISTE?

Los Proyectos de Programación consisten en proyectos de software que un grupo de estudiantes desarrollan a lo largo del semestre. Este desarrollo se realiza en el contexto de una materia de programación. Idealmente deberían involucrar varias materias del semestre o de la carrera que estén relacionadas con el tema del software.

¹ El desarrollo de software es la disciplina perteneciente a la Ingeniería de Software que se encarga de hacer programas de computadores.

² Software empotrado se refiere a aquellos programas que están en máquinas que requieren procesadores para su funcionamiento. Actualmente se tienen muchas máquinas de este tipo que van desde los hornos microondas hasta el transbordador espacial, pasando por cajeros electrónicos, máquinas expendedoras de bebidas y alimentos, teléfonos celulares, etc. En un futuro cercano se requiere dotar a todas nuestras máquinas de inteligencia para que estén interconectadas entre sí y a nuestros computadores; esa es la nueva Internet y el comienzo de una nueva era.

³ El libro de TURKLE, Sherry La vida en Pantalla es una excelente reflexión sobre la formación de la personalidad en la era de internet. Específicamente trata a los computadores y a Internet como la concreción física del pensamiento postmoderno.

En el proyecto se deben utilizar los métodos, recomendaciones y etapas de la Ingeniería del Software. Se hace bajo el contexto de la creación de una empresa de desarrollo de software, que es ideada desde el comienzo por cada grupo de estudiantes, lo cual se convierte en un fuerte factor motivacional si se lleva adecuadamente; de otro lado, es una estrategia para formar emprendedores en Ingeniería de Software.

3. IR PASO A PASO

Es recomendable que el proyecto se comience a realizar desde el primer día del semestre y se haga por etapas, para dividir el proceso en pequeñas etapas que por un lado sean sencillas de realizar por parte de los estudiantes y por otro, permitan la evaluación por parte del docente.

Cada etapa debe tener un punto culminante con la entrega de documentación, siendo la más importante, una puesta en común mediante una presentación pública. La entrega final tiene una sustentación en la que se evalúa el aspecto técnico del producto y un evento de lanzamiento del producto, a través del cual se motiva la formación empresarial del estudiante. También es conveniente permitir entregas parciales, para saber cómo va el proyecto.

La partición que se sugiere a continuación está basada en: Primero, el ciclo de vida clásico del software, esto se puede ver muy bien en [Pressman89, pág. 23] y en [Fairley87]. Segundo, en facilitar el desarrollo del proyecto en pequeñas etapas. Y tercero, en la experiencia que se ha tenido aplicando la metodología de Proyectos de Programación en varios cursos de programación.

3.1 Un Inicio Sencillo

Los estudiantes comienzan por la creación y publicación de una página personal en Internet. Esto con el objetivo de que aprendan empezando con algo muy sencillo. Deben poner sus datos personales, sus aficiones y en general todo lo que les interese. Se utilizan los servicios de correo electrónico gratis que hay en Internet y los servicios de publicación de páginas gratis que existen allí mismo. Pueden utilizar editores de página, pero se sugiere que aprendan HTML (Lenguaje de Marcación de Hipertextos).

Esta etapa sirve primero como una motivación a la materia, que está diseñada para que los estudiantes consideren que el futuro del software está ahí. Y no están errados, porque el auge de Internet y en especial el incremento de los productos y servicios que se ofrecen, así lo confirman. Por ello, toda la documentación del proyecto y el programa deben quedar publicados en Internet.

La página debe contener mínimo:

- Nombres completos.
- Foto.
- Intereses personales.
- Colección de enlaces preferidos.

A lo largo de todas las etapas se les hace (a los estudiantes) unos requerimientos mínimos que deben cumplir y superar según sus criterios. Este es otro aspecto que refuerza la personalidad emprendedora: Cumplir más de lo que se promete, una de las características de los líderes.

3.2 Dar a Luz una Empresa

Es vital para que el proyecto esté enmarcado en una empresa conceptuada y creada por los mismos estudiantes. Esta empresa, lógicamente, se debe ubicar en el sector informático, y específicamente como productora de software. Se motiva a los estudiantes diciéndoles que creen la empresa que sería la ideal para ellos, que sueñen cómo sería esa empresa que les gustaría que fuera de ellos. Una vez ideada, deben crearle en consecuencia una imagen corporativa, su misión, la visión, la estructura, etc. Al final se tiene un documento escrito en HTML que se publica en Internet, con la siguiente información mínima:

- Razón social.
- Objeto social.
- Imagen Corporativa.
 - Eslogan.
 - Logotipo.
- Datos de cada socio.
 - Código.
 - Nombres completos.
 - Cargo.

- Responsabilidades.
- Enlace a la página personal.

3.3 Y Ahora, ¿Qué Hago?

La siguiente etapa es escoger el tema del proyecto. El grupo de estudiantes debe ser el que escoja el tema del proyecto. Esto tiene varias razones: a) El factor motivacional: si los estudiantes no escogen su tema, si por el contrario, éste es impuesto no sentirán la misma motivación que por un tema que han seleccionado a su gusto. b) El compromiso: los estudiantes se sienten más comprometidos cuando son ellos los que escogen el tema y asumen sus propias decisiones. c) La creatividad: el ejercicio de tener que escoger un tema por su cuenta hace que comiencen a ejercitar las habilidades cognitivas relacionadas con la creatividad. Estas habilidades son condiciones necesarias para cualquier emprendedor. d) El desarrollo de la personalidad emprendedora: un emprendedor es quien toma la iniciativa, no el que espera a que le den el tema para desarrollarlo. e) La creación de software innovador para la Feria del Software, evento que realiza la Facultad de Ingeniería cada semestre.

Es necesario aclarar que no se debe dejar solo al estudiante en la selección del tema. El profesor debe coadyuvar en el proceso. Coadyuvar viene de co y del latín adiuvaré, ayudar, o sea, co-ayudar. Esto quiere decir que el profesor debe proporcionar las condiciones, las herramientas y propiciar el proceso para que el estudiante escoga el tema. Esto no quiere decir que le va a decir el tema ni siquiera a sugerírselo. Al principio es un proceso difícil y es posible que el estudiante insista en que se le dé un tema para desarrollar, porque así se le facilita la tarea y se le exime de pensar. En esos momentos es cuando el maestro debe mostrar el camino que debe recorrer el estudiante, pero nunca lo debe recorrer por él.

Otro aspecto con el que el profesor debe tener precaución es cuando los estudiantes propongan temas o demasiado fáciles o demasiado difíciles. Si propone un tema demasiado fácil no se desarrollarán las habilidades que el proyecto pretende. Y si el tema es muy difícil, el estudiante no terminará el proyecto y se sentirá decepcionado. El profesor debe hacer que el estudiante pon-

ga un proyecto en el límite de sus capacidades para ayudar a extender este límite. Por supuesto, el docente debe poseer un conocimiento detallado del estudiante y de sus capacidades; esto podría abordarse, por ejemplo, con una prueba de entrada, pero principalmente manteniendo una relación de diálogo con el estudiante.

El resultado de la selección del tema del proyecto es un documento que está escrito en lenguaje natural⁴ y explica en forma general lo que se va a hacer. Se le pide que la descripción sea sucinta y muy precisa, como ocurre siempre que se publica en Internet.

3.3.1 ¡Cuidado! No Hay que Pasarse

Esta metodología considera esencial utilizar los métodos, recomendaciones y etapas de la Ingeniería de Software, porque, el perfil del Ingeniero de Sistemas de la EAN está encaminado en esa dirección.

Los distintos métodos, recomendaciones y etapas de la Ingeniería de Software deben estar adecuados al nivel en desarrollo de software que posean los integrantes del proyecto. Esta adecuación no consiste en reducir el número de documentos para entregar, las etapas o los puntos en el documento, consiste más bien en adecuar el nivel de complejidad y profundidad de la documentación con base en el nivel de conocimientos y habilidades de los estudiantes.

Por ejemplo, en el caso de estudiantes de primer semestre, sus documentos deben tener las mismas partes, pero la diferencia está en el nivel de complejidad de los mismos: el de análisis será más una descripción del problema en lenguaje natural y sus distintos aspectos utilizando la lógica y el sentido común del estudiante. El documento de diseño, contendrá los diagramas de flujo y/o los algoritmos que, a este nivel, dada la simplicidad de los programas, son una descripción arquitectónica suficiente del software. El código

⁴ Se llama lenguaje natural a los lenguajes como el castellano o el inglés en contraposición de los lenguajes artificiales que por ejemplo se usan para la programación de computadores y la descripción formal de algoritmos.

fuentes no incluirá la partición de la solución en funciones, ni la definición de estructuras abstractas de datos, pero, eso sí, debe ir - como todos los documentos y etapas - perfectamente documentado.

3.4 Análisis. ¿Qué Hacer?

En la etapa de análisis los estudiantes deben determinar en forma exacta qué es lo que van a hacer, sin decir cómo lo van a hacer. Para ello se nutren de la fuente principal: el cliente. Deben estudiarlo, conocer su lenguaje, sus necesidades y de esta forma describir el problema que se quiere resolver. Este problema se desglosa en subproblemas que sean más fáciles de resolver. Tal como la estrategia de los antiguos romanos: Divide y vencerás.

Una vez hecho, se procede a plantear una solución software, una solución general que debe cumplir una lista de requerimientos que resuelvan los subproblemas y el problema principal como un todo.

Este proceso debe ser guiado por el profesor utilizando tanto sus estrategias pedagógicas como su experiencia personal en desarrollo de software. El autor se ha ayudado de varias estrategias que van desde charlas con los grupos, hasta talleres en donde se guía paso a paso al estudiante en el proceso. Lo importante en ésta y las siguientes etapas es que el estudiante reciba una realimentación permanente. Se debe dejar al estudiante la iniciativa en lo que se refiere a la realimentación, no hay que olvidar que estamos formando emprendedores.

El resultado de esta etapa es el Documento de Análisis que también debe publicarse en Internet.

3.4.1 Documento de Análisis

En el documento de análisis se define lo que se va a hacer pero no cómo se va a hacer y debe tener mínimo lo siguiente:

- Perfil del usuario. Cómo es el usuario, cuales son sus preferencias, cual es el lenguaje que utiliza, etc.
- Problema que pretende resolver el software:

- Descripción global del problema. Esta es una descripción en unos pocos párrafos que define el problema.

- Descripción detallada. Esta desglosa el problema en varios subproblemas y éstos a su vez en sub-sub-problemas y así sucesivamente. El resultado es una lista detallada de problemas.

- Lista de requerimientos. Esta es una lista detallada de todo lo que debe cumplir el programa terminado. Esta lista debe responder a cada uno de los problemas y subproblemas de la lista de problemas.

- Cronograma. Debe tener en cuenta las fechas de entrega especificadas en la materia de programación.

- Recursos a utilizar y costos.

- Soportes: entrevistas, evaluaciones, etc.

- Bibliografía consultada.

3.5 Diseño. ¿Cómo lo Hago?

Tomando como base el documento de análisis, el estudiante plantea cuál es la forma específica de la solución. Debe exponer en forma detallada cuál es el diseño del software que soluciona el problema.

Aquí también el docente está al lado del estudiante orientándolo en el proceso, pero cuidado, ya que el diseño y todo el proceso de creación deben correr por cuenta del estudiante. Como en la etapa anterior, el docente dará todas las condiciones para que el estudiante pueda tomar la iniciativa de realimentación.

El resultado de esta etapa son dos documentos: el documento de diseño y el Manual preliminar del usuario, ambos escritos en HTML y publicados en Internet. Con respecto al manual del usuario, es muy importante que se haga una versión preliminar en esta etapa pues es una forma de saber cómo va a quedar la aplicación al final, asunto vital desde el punto de vista del cliente y del docente. El primero puede determinar si eso

es lo que quiere y el docente puede ver si está dentro de las posibilidades de los estudiantes. En este mismo orden de ideas y dada la importancia de este paso, los estudiantes deben exponer su proyecto ante el curso y ante el profesor. La exposición es de carácter técnico: Se trata de exhibir las competencias adquiridas ante profesores y estudiantes, para autoevaluarse y evaluar las posibles sugerencias y adecuaciones. Contrastarse es la mejor forma de ampliar horizontes, incorporar ideas o desarrollar más las que se poseen.

3.5.1 Documento de Diseño

El documento de diseño debe especificar cómo se va a hacer el programa. Contendrá como mínimo los siguientes items:

➤ Diseño arquitectónico, estructura del programa:

- Diagrama de los módulos de la aplicación y su relación entre sí.

➤ Diseño detallado:

- Los encabezados de función y/o procedimientos:

1. Tipo.
2. Identificador de la función.
3. Nombre, tipo y descripción de cada uno de los parámetros.
4. Precondición y postcondición.
5. Descripción funcional de la función. Párrafo corto en lenguaje natural que describe lo que hace la función (pero no cómo lo hace) en términos de entradas y salidas, o sea, en términos de sus parámetros y el valor que retorna.
6. Representación del algoritmo de la función (o procedimiento) mediante pseudocódigo y/o diagramas de flujo (se pueden utilizar representaciones equivalentes si el proyecto lo amerita).



7. caso de prueba de la función, con entradas y salidas perfectamente definidas.

➤ Índice alfabético.

3.5.2 Manual del Usuario

El manual del usuario contiene mínimo las siguientes partes:

➤ Presentación.

➤ Requerimientos.

➤ Instalación.

➤ Tutorial: Esta es una guía que explica en forma muy sencilla al usuario cómo usar el programa.

➤ Manual de referencia: Es una guía detallada -organizada en forma

lógica y técnica-, de todos los aspectos del programa (Ej: archivos de configuración y sus distintas partes, instalación avanzada, configuración avanzada, detalles técnicos, archivos de datos y su estructura y todos los aspectos del programa que pueda requerir un usuario avanzado).

➤ Índice alfabético.

3.6 Entrega Final !Por Fin!

En la entrega final se sabe si tuvo éxito todo el proceso. Ya no es el momento de corregir. Es el momento de presentar el software funcionando y cumpliendo con los requerimientos del documento de análisis. También de entregar corregidos, los anteriores documentos, que lógicamente varían en el proceso de Implementación⁵ del software.

Dada la importancia de esta etapa, fuera de la entrega de la documentación, los estudiantes sus-

⁵ La palabra implementación es aceptada por la Real Academia Española. En su Diccionario de la Lengua Española se lee: "Implementar. tr. inform.. Poner en funcionamiento, aplicar métodos y medidas, etc., para llevar a cabo algo".

tentan su trabajo ante dos instancias: La primera, con el maestro quién evalúa el componente técnico y determina si el proyecto puede presentarse al Evento de Lanzamiento de Producto. La segunda en el Evento de Lanzamiento de Producto que hace énfasis en su aspecto empresarial y comercial, este evento se hace ante académicos, estudiantes e invitados especiales de la facultad y del mundo empresarial (posibles clientes o inversionistas).

Resumiendo, la entrega final consta de:

➤ Los documentos:

- Páginas personales de los socios.
- Página de la empresa.
- Paquete de ingeniería.
- Paquete comercial.

➤ Sustentación de la parte de ingeniería del programa:

- Requerimientos mínimos para poder presentar la sustentación (si no se cumplen, no se puede presentar):

1. El programa debe compilar con cero errores y cero warnings.
2. El programa debe funcionar, hacer lo que debe hacer de forma completa (esto se verifica contra la lista de requerimientos del documento de análisis).

➤ Evento de lanzamiento de productos.

- Requerimientos mínimos para poder participar en el evento:

1. Haber aprobado la sustentación con un promedio acordado previamente.

3.6.1 Paquete de Ingeniería

El paquete de ingeniería es el conjunto de todos los documentos que dicen como está hecho el software y los programas en sí. El paquete técnico

debe también escribirse en HTML y publicarse en Internet. Los documentos que componen el paquete técnico son:

- Descripción del software.
- Documento de análisis.
- Documento de diseño.
- Manual del usuario.
- Código fuente perfectamente documentado.
- Ejecutable del programa.
- Documento de pruebas.
- Licencia del producto.

A continuación se describen detalladamente los documentos que no han sido explicados.

3.6.1.1 Código fuente del Programa⁶.

El código fuente es vital para cualquier desarrollo software, por ello debe estar muy bien construido, y ser fácil de entender y de mantener. El código debe poder ser leído y entendido por cualquier persona que tenga conocimientos de desarrollo de software y debe también ser leído y entendido por los autores después de muchos años de haberlo creado. Por ello, se exigen una serie de prácticas y requerimientos que se consideran prácticas de buena programación:

➤ Requerimientos mínimos (deben satisfacerse para poder comenzar a evaluar el código fuente, de lo contrario, es como si no se hubiera presentado el programa):

⁶ El código fuente del programa, es el conjunto de archivos texto que tienen las instrucciones que el computador debe seguir. Este código fuente debe ser traducido para crear un código ejecutable.

- El programa debe compilar con cero errores y cero warnings, estando estos activados en su totalidad.

- El programa debe funcionar, esto es, debe hacer lo que se especificó en la lista de requerimientos del documento de análisis y debe hacerlo bien, adecuadamente y completamente (no debe haber casos de excepción que no estén especificados en el documento de análisis).

- El programa no debe presentar errores sin controlar, como divisiones, por cero, apuntadores perdidos, salidas de rango, pérdida o errores de memoria, etc.

- No debe presentar el código o el ejecutable con virus.

- Es vital que el estudiante tenga un conocimiento profundo del programa. Debe demostrar que fué él mismo quien hizo el programa.

- Debe haber sido realizado por la persona que lo presenta (no se aceptan ni copia por ningún medio, ni compra, ni regalo, ni donación, ni colaboración, ni alianzas, ni nada similar).

- No se admiten programas que no estén estructurados. p. ej.: no se admite el uso de variables globales, ni la utilización de la instrucción "goto".

- No se admite código escrito por generadores de programas, p. ej.: los hechos con entornos de programación visual.

➤ El programa debe tener:

- Un archivo de código fuente del proyecto con:

1. Un comentario de cabeza con:

Nombre de la empresa.
Nombre del proyecto.

- Nombre y número de versión del módulo.

- Integrantes del proyecto.

- Encargado del archivo.

- Fecha de creación.

- Si hay modificaciones.

1. Fecha de la modificación.

2. Descripción corta de la modificación.

3. Motivo de la modificación.

- Cada constante debe tener:

1. Identificador.

2. Valor.

3. Comentario de su función en el código.

4. No haber números mágicos en el código (esto es, variables numéricas inmersas en el código).

- Cada tipo de dato debe tener:

1. Identificador.

2. Comentario de su funcionalidad.

3. Invariante de tipo de dato.

4. Cada campo debe tener:

Tipo.

Identificador.

Comentario con la descripción de su funcionalidad dentro del tipo de dato.

5. Cada operación sobre el tipo de dato debe tener lo mismo que las funciones (ver más adelante).

- Cada función debe tener:

1. Tipo.

2. Identificador de la función que debe incluir el nombre del tipo de dato al que pertenece.

3. Nombre, tipo y descripción de cada uno de los parámetros.

4. Precondición y postcondición.

5. Descripción funcional de la función. Esto es un párrafo corto en lenguaje natural que describe lo que hace la función (pero no como lo hace) en términos de sus entradas y salidas, o sea, en términos de sus parámetros y el valor que retorna.

6. En el cuerpo de la función debe haber:

- Variables locales.
- Comentarios iniciando cada partición del problema.
- Comentario en cada ciclo indicando el invariante.
- Comentarios aclarando cualquier código oscuro.

3.6.1.2 Documento de pruebas

El documento de pruebas garantiza que el software, cumple con los requerimientos especificados en el documento de análisis y que no tiene ninguna falla. Es el certificado de calidad del programa.

- Lista de requerimientos.
- Validación de que estos requerimientos están en el programa.
- Lista de casos de prueba con su exacta especificación de entradas, salidas y resultados intermedios.
- Validación de cada caso de prueba.

3.6.1.3 Licencia del producto

Los estudiantes como empresarios del software deben proteger sus creaciones legalmente mediante licencias. Por su naturaleza académica, los Proyectos de Software, deben ser abiertos para dejar que tanto el maestro como sus pares y estudiantes puedan estudiar como está hecho el software. También permite que otros grupos de desarrollo de la EAN o externos puedan continuar un proyecto o mejorarlo en consonancia con el proyecto académico y respetando la autoría. Para resolver esta aparente contradicción, el mundo Internet ofrece una solución probada mundialmente: la licencia GNU.

La licencia GNU permite proteger la autoría de un software admitiendo que otras personas interesadas puedan estudiar el programa, copiarlo, mejorarlo, usarlo, instalarlo, etc. sin problemas legales. Este tipo de licencia es el que usan los programas más populares en Internet tales como el sistema operativo Linux y el servidor de páginas Web Apache. Y lo que es más importante: permite al docente y los estudiantes examinar y usar el software para su evaluación sin comprometer a la institución en problemas legales.

Por lo anterior, se exige que todos los Proyectos de Programación se licencien bajo la modalidad GNU. Deben incluir tres copias por escrito y una electrónica.

3.6.2 paquete comercial

- El paquete comercial del programa debe contener como mínimo:
 - Caja del producto incluyendo la imagen corporativa y la imagen del producto.
 - Descripción del software.
 - Manual del usuario.
 - Disco con los ejecutables del programa, incluyendo instalador.
 - Licencia del producto.

3.6.3 Organización del evento de lanzamiento de productos

El evento tiene como fin que los estudiantes desarrollen competencias y habilidades empresariales tales como la presentación de sus ideas ante un público, saber vender su producto, proyectar su imagen y la del producto, etc.

El proceso de organización es el siguiente:

- Se nombra un grupo organizador del evento.
- El grupo escoge la forma que tendrá el evento e invita a:

- Docentes de tiempo completo de los departamentos que están vinculados a la carrera:

1. Ciencias básicas.
2. Ciencias de la computación.
3. Ingeniería de software.
4. Comunicaciones.
5. Jefes de estos departamentos.

- Docentes del semestre.

- Decano de la facultad.

- Otros profesores interesados.

- Invitados externos.

➤ Las invitaciones deben ser entregadas con quince días de anticipación.

4. CONCLUSIONES

En los cursos en que se ha aplicado esta metodología se han desarrollado proyectos de software innovadores, gracias a que se ha alcanzado una gran motivación de los estudiantes. Los logros más significativos son:

1. Aprender a desarrollar software en equipo.
2. Sustentar el software ante un público interno y externo.

Se han presentado resultados adversos cuando los proyectos de los estudiantes han estado por debajo o por encima de sus posibilidades. Pero si el profesor ha hecho un seguimiento permanente y responsable a través de los distintos momentos de realimentación y evaluación, esta situación no se debe presentar.

En general, el autor considera que la metodología de Proyectos de Programación es una buena alternativa para la enseñanza de la programación en el contexto de la Ingeniería de Software para la formación de emprendedores en el sector informático.

BIBLIOGRAFÍA

ANGULO, María Elena. El desarrollo de la personalidad emprendedora: un programa de formación empresarial no convencional. En Revista E.A.N. No 19-20 (Mayo - Diciembre 1993); p 34-41.

DAVIS, Gary A. y SCOTT, Joseph A. Estrategias para la creatividad. Primera edición, 3a. reimpresión 1992, Ed. Paidós, Buenos Aires.

Escuela de Administración de Negocios. Medios y estrategias en la formación de empresarios : la experiencia de la EAN. En revista EAN No 2. (Enero -Abril de 1987); p. 32-47.

Escuela de Administración de Negocios E.A.N. Proyecto Educativo Institucional. Julio de 1995.

FAIRLEY, Richard E. Ingeniería de software. México, 1987 Mc Graw Hill.

GARDNER, Howard. Inteligencias Múltiples: La teoría en la práctica. Primera edición, Ed. Paidós, Barcelona, 1995.

KILPATRICK, C. The Project Method. Teachers College Record. Vol. 19, no 4 (Septiembre), págs. 230-335.

NORRIS, Mark y RIGBY, Peter. Ingeniería de software explicada. Primera edición, México, 1994.

PARRA RODRÍGUEZ, Jaime. Inspiración: Asuntos íntimos sobre creación y creadores. Primera edición, Bogotá, Ed. Magisterio, 1996.

PRESSMAN, Roger S. Ingeniería del Software: Un enfoque práctico. Segunda edición, Mc Graw-Hill, Madrid 1989.

PRESSMAN, Roger S. Ingeniería del Software : Un enfoque práctico. Cuarta edición, Mc Graw-Hill, Madrid 1998.

Real Academia Española. Diccionario de la Lengua Española. Vigésima Primera Edición, Madrid, Espasa Calpe, 1992.

RAMÍREZ CARDONA, Carlos. Empresa, empresario emprendedor o intraempresario. En revista

E.A.N. No. 12 (Septiembre - Diciembre de 1990); p. 14-17.

RODRÍGUEZ GÓMEZ, Luis Gerardo. Docencia para el logro y educación para emprender. En Revista E.A.N. No 23 (Septiembre - Diciembre de 1994); p. 51-62.

RODRÍGUEZ GÓMEZ, Luis Gerardo. Educación para emprender: una opción para el mundo en desarrollo. En Revista E.A.N. No 25-26 (Junio - Diciembre 1995); p. 55-62.

RODRÍGUEZ HERNÁNDEZ, Jairo Enrique. La formación de profesionales empresarios: reponsabilidad de quién?. En revista EAN No 11 (Mayo -Agosto de 1990); p. 5-11.

RUBIO SALAS, Alvaro. Formación profesional, espíritu emprendedor y creación de nuevas empresas. En revista E.A.N. No. 9 (Septiembre - Diciembre de 1989); p 27-28.

RUBIO SALAS , Alvaro. Formación profesional y espíritu emprendedor. No. 16-17 (1992); p. 7-13.

GUILFORD, J. P., et. al. Creatividad y educación. Primera edición, tercera reimpresión 1994, Paidós, Barcelona.

LÓPEZ GUZMÁN, Pedro A. Usted puede ser empresario!. En revista E.A.N. No. 4 (Enero - Abril de 1988); p. 55-57.

TOFFLER, Alvin. El cambio del poder. Cuarta edición: julio, 1995, Barcelona, Plaza y Janes.

TORRES SANTOMÉ, Jurjo. Globalización e interdisciplinariedad: el currículo integrado. Segunda edición, Madrid, ed. Morata, 1996.

TURKLE, Sherry. La vida en la pantalla: La construcción de la identidad en la era de Internet. Primera edición, 1997, Paidós, Barcelona.

Universidad Distrital Francisco José de Caldas. Curso de capacitación proyecto de modernización del CEMDIZOB: Lecturas. Santa Fe de Bogotá 1998.

WHITEM, Jeffrey L., et. al. Análisis y Diseño de Sistemas de Información. Primera edición, Bogotá, 1997.