

ENSEÑANZA DE APLICACIONES WEB ENRIQUECIDAS BAJO UN ENFOQUE PEDAGÓGICO DE APRENDIZAJE ACTIVO

Juan Pablo Lozano
Universidad de los Andes
Bogotá, Colombia

Juan Carlos González
Universidad de los Andes
Bogotá, Colombia

juan-loz@uniandes.edu.co

juan-go1@uniandes.edu.co

ABSTRACT

En este artículo se explica el proceso desarrollado para la reestructuración y crecimiento del curso de Clientes Web Enriquecidos del Departamento de Ingeniería de Sistemas de la Universidad de los Andes. Este proceso toma como punto de partida el análisis de la web actual, el usuario de esta y el análisis de los cursos de programación existentes, su metodología y sus dificultades para así poder estructurar el curso a partir de una metodología exitosa, soportada por unas bases que le permiten a los estudiantes obtener los conocimientos y las habilidades para el desarrollo de aplicaciones web independientes de la tecnología. Estas bases deben incluir unos ejes conceptuales que den al desarrollador los conocimientos necesarios para desarrollar la aplicación y que deben ser enseñadas mediante una metodología que les permita siempre estar reforzando cada eje. Para ello no se hace énfasis en temas muy específicos sino en temas que hacen parte de toda aplicación web y que si no se tratan dejan en la aplicación un vacío. Este proceso concluye con la evaluación de cada eje y cada parte de la metodología mediante la creación de material relacionado a cada eje conceptual utilizando la metodología adoptada así como la evaluación del mismo.

Palabras Claves

Aprendizaje Activo, Aprendizaje incremental, Aprendizaje basado en Problemas, Aprendizaje basado en ejemplos. Aplicación Web, Clientes, Servidor. MVC (Modelo – Vista - Controlador), Seguridad, Testing, Usabilidad, Programación, Arquitectura.

1. INTRODUCCIÓN

Las aplicaciones Web han cambiado en muchos aspectos en los últimos años, presentando un nuevo contexto para el desarrollo web, contenido web y los usuarios web. Ahora hay un control mínimo sobre el usuario de manera que es el usuario el que controla la aplicación y su contenido; esto es posible ya que el usuario ha adquirido ciertas características (es activo,

experto y colaborativo) que le permiten ser el protagonista de la web, no solo un espectador. Este cambio ha llevado a tres aspectos importantes. Primero tenemos que la interacción actual que existe entre usuarios web ha causado una reacción en cadena y un crecimiento de la red que también ha llevado a la creación de sistemas abiertos, hoy llamados **Redes Sociales**. Segundo el contenido y el conocimiento que presenta una aplicación o sitio web ya no están basados en la inteligencia de una sola persona, por el contrario estos son construidos mediante la **Inteligencia Colectiva**. Finalmente la web actual ofrece una serie de servicios como blogs, comunidades online, etc., todos estos basados en interacciones e individualización haciendo que la web se vea como una **Plataforma de Servicios**. Estos aspectos a su vez le presentan al usuario un nuevo canal de distribución para la información y causan un gran impacto en la **Educación, la Investigación y los Negocios**.

Basado en todos estos cambios se ha considerado la necesidad de tener un curso que genere habilidades y conocimientos en los estudiantes para desarrollo web [1].

Sin embargo construir un curso de Ciencias de la Computación y Programación implica una serie de **dificultades generales** [2] presentes en diferentes actores de estos cursos. Por ejemplo, una parte de los estudiantes muestran una baja motivación al afrontar estos cursos, no logran ver la necesidad, utilidad y ganancia del trabajo realizado, encuentran dificultades para desarrollar habilidades y conocimiento cuando programan y presentan una constante frustración ante las fallas del trabajo realizado y lo atribuyen a un factor de suerte. Igualmente existen dificultades presentes en los profesores como es la implementación de su metodología, el acoplamiento de esta a la metodología del grupo de profesores y al material de apoyo que se le entrega. Adicional a las dificultades generales de un curso de programación, estos también presentan **dificultades específicas** [3] según el tema en el que profundiza el curso. En el caso de un curso de Desarrollo Web se presentan dificultades como el alto número de tecnologías disponibles, su constante cambio, la gran cantidad de conocimiento disponible y lograr que el diseño este en función del usuario, los tipos de usuario, su información y sus necesidades.

Teniendo en cuenta que la **metodología de Cupi2** [4], basada en cuatro ramas de aprendizaje como Aprendizaje Activo, Aprendizaje Incremental, Aprendizaje mediante Ejemplos y Aprendizaje Basado en Problemas, ha resuelto y mitigado las dificultades generales en los cursos de programación, el curso que se está estructurando debe encargarse de resolver las dificultades específicas que surgen del Desarrollo Web. Para esto se ha realizado una **propuesta** basada en el uso de la metodología de CUIP2 que implica la necesidad de definir y aplicar una **arquitectura de referencia**, basada en un modelo cliente enriquecido-servidor, que sirva de soporte para aplicaciones de internet enriquecidas (RIAS) y no dependa de la tecnología. Igual que en Cupi2, surge la necesidad de definir y aplicar una nueva serie de ejes conceptuales (para aplicaciones web en este caso) que le permitan a los estudiantes adquirir **nuevos** conocimientos y las habilidades necesarias para desarrollar aplicaciones web, como son usabilidad, testing, seguridad y uso de herramientas y lenguajes de programación Web.

Entonces a partir de la anterior nueva propuesta se ha diseñado el curso de Clientes Web Enriquecidos basado en actividades donde se aplica el aprendizaje activo y en una serie de ejercicios y ejemplos donde se aplican las 4 ramas de aprendizaje de la metodología de Cupi2 y los cuales corresponden a un problema, están soportados por la arquitectura de referencia diseñada y son resueltos mediante el aprendizaje incremental de los nuevos ejes conceptuales definidos; todo esto con el fin de que se vean resueltas las dificultades específicas de diseñar un curso de RIAS antes mencionadas, ya que las dificultades generales de enseñar un curso de Programación y Ciencias de la Computación ya han sido resueltas mediante la metodología de Cupi2.

La estructura de este artículo es la siguiente:

- La Sección 2 explica el objetivo y las ramas de aprendizaje de la metodología Cupi2, cómo esta ha sido exitosa a la hora de diseñar un curso de Ciencias de Computación y Programación y como se puede aplicar para estructurar un curso de desarrollo web, que es el objetivo principal.
- La Sección 3 explica la arquitectura de referencia diseñada a partir de la anterior metodología y porque es necesaria para dar soporte a aplicaciones web enriquecidas y explica los diferentes ejes conceptuales que serán la base del nuevo curso y las razones que llevaron a definirlos.
- La Sección 4 muestra la estructura del curso diseñado, la aplicación de la metodología de Cupi2, de la arquitectura de referencia diseñada y de los ejes conceptuales en los diferentes ejercicios y ejemplos en que se basa el nuevo curso y las métricas del trabajo hecho para este.

- La Sección 5 explica el trabajo relacionado existente y las ventajas o nuevos aportes que tiene nuestro trabajo con respecto a otros.
- Finalmente en la Sección 6 se dan las conclusiones de nuestro trabajo y se explica el trabajo futuro que viene para el curso.

2. CUIP2. BUSCANDO NUEVAS MANERAS DE APRENDER A PROGRAMAR

Cupi2 es un proyecto creado en el año de 2003, en Bogotá Colombia, por el grupo de construcción de software del departamento de Ingeniería de Sistemas y Computación de la Universidad de los Andes. Este proyecto fue creado con el objetivo de encontrar y desarrollar soluciones a los inconvenientes y a las dificultades existentes en el proceso de enseñar un curso de programación, haciendo énfasis en las dificultades que afectan a estudiantes, profesores y al departamento que da estos cursos. El resultado de este proceso es una metodología exitosa, aplicada en múltiples universidades de Colombia en la actualidad y basada en un conjunto de ideas, recursos y experiencias que dan soporte al proceso de aprendizaje y enseñanza de un curso de programación.

El proceso desarrollado en este proyecto comenzó con un diagnóstico de la situación, para luego pasar a la creación de una clase piloto y a la evaluación de esta misma y las estadísticas que esta dejó. Tras el análisis y la conclusión de esta etapas el proyecto dio como resultados un modelo pedagógico basado en 7 ejes conceptuales que dan las bases, habilidades y conocimientos necesarios para la programación. Los ejes que plantea este modelo son:

1. Solution Modeling to problems
2. Algorithmic
3. Technology and programming
4. Programming tools
5. Software Processes
6. Programming techniques and Methodology
7. Structural Elements and Architecture

Sin embargo el modelo no plantea solamente 7 ejes conceptuales sino que además plantea 4 ideas o conceptos sobre los cuales se basa el proceso de enseñanza y aprendizaje de estos 7 ejes. Estas son las 4 ideas:

- **Aprendizaje Activo:** Este concepto propone que el proceso de enseñanza debe involucrar al estudiante como actor principal del proceso a través de actividades e

interacción entre los mismos estudiantes y el profesor, mas no en actividades donde el único que participe sea el profesor. Con este proceder para los cursos se busca involucrar más al estudiante y lograr en este una afinidad por el curso y facilitarle así el aprendizaje.

- **Aprendizaje incremental:** Este concepto plantea que dado el número de ejes conceptuales que se tratan en un curso de programación y teniendo en cuenta que varios de estos tienen distintos niveles de complejidad, estos se deben introducir a través de niveles de manera gradual donde en cada nivel se introduzca ciertos conceptos y se evalúen, y donde estos se refuercen en los siguientes niveles. El objetivo introducir de una manera más pausada los conceptos, evaluarlos y asegurarse que se han comprendido para luego pasar a los conceptos nuevos.
- **Aprendizaje basado en problemas:** Esta idea propone que los cursos de programación se debe acompañar de problemas reales, realizables y sobre todo afines a los estudiantes para que estos puedan sentirse identificados con el desarrollo del problema, se motiven a desarrollarlo, lo asocien con sus carreras y por ende vean la utilidad de lo que están haciendo.
- **Aprendiendo con Ejemplos:** Por último, la metodología propone que el desarrollo del curso, de los problemas y actividades debe ir acompañado por una extensa lista de ejercicios donde los estudiantes puedan buscar ejemplos a soluciones de problemas similares y así llegar a disolver las dudas existentes al desarrollar los problemas.

El resultado de la implementación de este proceso en los últimos años ha sido muy positivo y se refleja en la reducción del 16% del número de estudiantes que han perdidos los cursos de programación en la Universidad, así como en el incremento que ha tenido la calificación dada por los estudiantes tanto a profesores como a los cursos [4] y en el premio ACIS 2007 obtenido por el proyecto.

El éxito que ha tenido este proyecto nos muestra una metodología exitosa y aplicable para cursos de programación, como el curso que se pretende estructurar. Esta metodología, y su aplicación en cursos pasados, le entregan a este nuevo curso estudiantes con unas bases sólidas en programación y su aplicación en este curso implica, como se dijo antes, que es necesario definir los ejes conceptuales específicos del área de Desarrollo Web y que para desarrollarlos con esta metodología se debe definir una arquitectura de referencia que soporte los ejercicios, problemas, actividades y afiance el conocimiento entregado en cada nivel.

3. EJES CONCEPTUALES Y ARQUITECTURA DE REFERENCIA – NUEVAS BASES PARA APLICACIONES WEB

El desarrollo web es un área que incluye una lista muy grande de tecnologías y conceptos, muchos de estos en constante cambio. Debido a esto, el proceso de aprendizaje y enseñanza en un curso de Desarrollo Web requiere que se defina una lista de conceptos base o ejes conceptuales que le den a los estudiantes las habilidades y el conocimiento necesario para adaptarse, cuando sea necesario, a tecnologías específicas y a conceptos específicos y avanzados del área. En otras palabras, los ejes conceptuales de este curso deben ser independientes de la tecnología y deben entregar el conocimiento necesario para adaptarse a temas avanzados del área. Estos son los ejes que le permitirán al curso cumplir con este objetivo:

3.1 Ejes Conceptuales

1. **Arquitectura:** Este eje define y explica la estructura de una aplicación web basada en los conceptos de cliente y servidor. Este eje le dará a los estudiantes las bases para entender cómo funciona una aplicación web y como implementarlas con cualquier tecnología partiendo de una misma estructura.
2. **Usabilidad:** Este eje estudia el concepto de usabilidad que se puede definir como la facilidad de uso de una aplicación o, como lo define Powell, el grado hasta el que un sitio u aplicación web puede ser utilizado por un grupo de usuarios para cumplir ciertas metas en un contexto específico de manera efectiva, eficiente y satisfactoria [5]. Es muy importante que un curso de desarrollo web incluya un eje de usabilidad y temas como estándares de usabilidad, patrones de usabilidad y el nivel de experiencia y las necesidades de un usuario ya que todo desarrollador web debería no tener en cuenta su fácil adaptación a un interfaz web y colocarse en el papel del usuario final de la interfaz para así entender que este necesita ciertas guías para la adaptación, guías basadas en sus necesidades, en patrones, en estándares y en los objetos visuales que el usuario identifica fácilmente, todo esto con el fin de lograr una experiencia agradable para el usuario. Este eje busca enseñarle a los estudiantes a diseñar aplicaciones teniendo en cuenta a los usuarios, sus necesidades, nivel de experiencia en la web y las guías, estándares y patrones que aplicaciones web exitosas han establecido.

3. **Testing:** Este eje es uno de los más importantes y estudia el concepto de testing en aplicaciones web, que es un aspecto que permite la búsqueda de errores y la mejora de una aplicación web tanto en su parte funcional como en su estructura y su rendimiento en cuanto al acceso de recursos. El trabajo realizado para este eje se hizo de acuerdo a [12] que establece que se debe tener una estrategia dual para Testing de aplicaciones web que se base en probar la compatibilidad de la aplicación web con el ambiente de operación (pruebas de carga o rendimiento) y probar la funcionalidad de la aplicación web misma (pruebas unitarias funcionales y no funcionales). Para esto el testing funcional de aplicaciones web se desarrolla con base en una de las técnicas propuestas por Ricca and Tonella en [13] que es la técnica de testing funcional que soportan testing basado en requerimientos. De acuerdo con esto este eje le permitirá generar a los estudiantes la habilidad de poder probar una aplicación web en cuanto a su rendimiento y en cuanto a su funcionalidad teniendo un gran cubrimiento de testing sobre casi toda una aplicación web, soportado este proceso por herramientas para testing Web como lo son HttpUnit y JSUnit en cuanto a testing funcional y OpenLoad y YSlow para testing de rendimiento y carga.
4. **Seguridad:** La seguridad en una aplicación web es un aspecto muy importante especialmente si se tiene en cuenta que en países como los Estados Unidos donde, en una encuesta realizada por el FBI en el 2006, diferentes persona u organizaciones reportaron que de los incidentes de seguridad sufridos con su información 6% fueron por causados por desfiguración del sitio web y otro 6% fueron causados por una utilización mal intencionada del sitio web. [6]. Estos ataques son la consecuencia de acciones, por parte de hackers u otras personas, que involucran técnicas o ataques como *Cross Site Scripting*, *SQL injection*, etc, aplicadas en las malas implementaciones de seguridad de sitios corporativos, sitios web de entidades financieras, sitios de comercio electrónico, etc, siendo estos los blancos preferidos de esta gente [7]. Por ende, y teniendo en cuenta que hoy en día los sitios web están pensados para los usuarios, para que estos construyan el contenido del sitio, la interacción del sitio y para que manejan grandes cantidades de información dinámica, es muy importante que un curso de desarrollo web cree las habilidades necesarias y buenas prácticas de desarrollo que tengan en cuenta los riesgos de seguridad actuales, su funcionamiento y como evitarlos, no solo con soluciones puntuales sino partiendo de un proceso

de desarrollo que tenga la seguridad del sitio desde el comienzo.

5. **Herramientas, frameworks, librerías y tecnología:** Las aplicaciones web han sufrido muchos cambios en los últimos años convirtiéndose en aplicaciones con mayor funcionalidad, dinamismo y que buscan ofrecer una experiencia similar a la de una aplicación de escritorio [] donde la actualización del contenido y de la información no siempre depende de las acciones del usuario. Sin embargo todo esto trae consigo mayores tiempos de implementación, nuevas tecnologías, nuevos diseños para diferentes componentes, entre otras cosas, razón por la cual se vuelve necesario buscar formar para volver más eficiente el proceso de desarrollo. Es por esto último que se vuelve necesario que el curso tenga un eje conceptual donde el estudiante trabaje y comprenda de librerías y frameworks que le permiten la reutilización de componentes y le reducen los tiempos de implementación, logrando una aplicación con la funcionalidad deseada.
6. **Lenguajes y Programación:** Como en todo curso de programación, es necesario trabajar con los lenguajes de programación y en el caso de este curso debe existir un eje donde se trabajen diferentes lenguajes de programación en diferentes niveles y ejercicios para así demostrarle a los estudiantes que puede existir una independencia entre el desarrollo y la aplicación si se tiene una buena arquitectura.

Teniendo los ejes conceptuales definidos es claro que se necesita de una metodología apropiada para llegar a un proceso de enseñanza y aprendizaje exitoso de estos, y es por eso que se propone el curso bajo el enfoque y la metodología de CUIP2. Sin embargo las 4 ideas sobre las que se basa el modelo de CUIP2 no se pueden desarrollar por sí solas si no existe una arquitectura de referencia que de soporte al material (problemas, ejercicios, ejemplos, etc) con el que se desarrollan. Por ejemplo, para la aplicación del aprendizaje activo es necesario que los escenarios de trabajo y sus ejercicios sean soportados siempre con la misma arquitectura evitando posibles confusiones en el conocimiento y las habilidades adquiridas, cuando se realicen cambios entre los niveles, los ejercicios y demás actividades. Por el lado del aprendizaje basado en ejemplos y en problemas es necesario mantener una misma arquitectura que afiance con cada ejercicio el eje de arquitectura así como los diferentes conceptos adquiridos en cada nivel.

3.2 Arquitectura de Referencia

La arquitectura de referencia que se plantea para el curso parte de ver esta estructura a través de dos componentes, un cliente y un servidor, y un canal de comunicación que envía pedidos y respuestas entre estos dos componentes cumpliendo con un protocolo (HTTP 1.2). Esta estructura a nivel global plantea un patrón MVC (Modelo-Vista-Controlador) donde el modelo se representa con el modelo de datos de la aplicación, la vista se representa con la interfaz de la aplicación y el controlador está representado por la lógica de aplicación presente tanto en el lado cliente como en el lado servidor.

Entrando en más detalle, el lado cliente se construye con la interacción de tres componentes formando también un patrón MVC. En primera instancia se tiene una vista que es el resultado de la interpretación del código html que recibe el cliente, código que por lo general sigue ciertos patrones de usabilidad en cuanto a sus componentes y a su diagramación y por ende casi siempre presenta un encabezado, un menú, un menú lateral, un bloque de contenido y un posible pie de página (Ver Figura 1). El segundo componente del cliente es el controlador, el cual se encarga de capturar y procesar las acciones y eventos que genera la vista y manejar las conexiones asincrónicas con el servidor. Y por último está el modelo de datos del cliente, representado por los datos que se almacenan en el cliente, tanto en cookies como en variables, y por el modelo de objetos del documento (DOM) y las hojas de estilo que modelan parte de lo que se visualiza en la vista. La unión de estos componentes y su interacción explican la estructura de una aplicación web en el lado cliente y muestran una independencia de la tecnología, de los lenguajes y los diferentes navegadores. Trabajando los diferentes ejes con ejercicios, evaluaciones e introduciendo nuevos conceptos en cada nivel, usando siempre esta estructura en cada nivel, se busca afianzar conceptos como las hojas de estilo, el funcionamiento del navegador, DOM y Ajax, entre otros. El siguiente diagrama muestra el cliente y sus componentes:

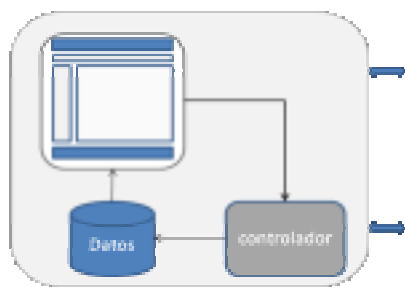


Figura 1. Arquitectura de Referencia en el Cliente

Por otro lado tenemos al servidor que con este planteamiento se presenta completamente independiente del cliente y por ende el lenguaje en el que este implementado y la tecnología que lo soporta no afectan de ninguna manera la implementación del lado cliente. Este bloque de la arquitectura de referencia

consiste primero en una serie de servicios que se ofrecen para que una implementación en el lado cliente los utilice según lo que se quiera hacer. Estos servicios utilizan una lógica de negocio que interactúa a su vez con un modelo de datos de la aplicación, con unos recursos (imágenes, archivos, etc) y con un conjunto de variables que se utilizan para manejar las sesiones sobre la aplicación. Estos componentes están soportados por un servidor web que se encarga de procesar e interpretar los pedidos y las respuestas. Con esta estructura o arquitectura de referencia en el lado cliente se busca plantear un modelo que es independiente de la tecnología (Java, PHP, .NET, etc), que sea extensible para el manejo de diferentes frameworks y librerías y que sobre todo sea independiente del lado cliente en cuanto a su tecnología. La siguiente imagen muestra el lado servidor de una aplicación web:



Figura 2. Arquitectura de Referencia en el Servidor

4. CLIENTES WEB ENRIQUECIDOS – UN CURSO DE ALTO NIVEL PARA EL APRENDIZAJE DE RIA’S

Por una parte, ante la necesidad de construir un curso que permita a los profesores crear un ambiente de aprendizaje y que a su vez le permita a los estudiantes aprender y desarrollar las habilidades necesarias para Desarrollo Web, se requiere del uso de una metodología para alcanzar estos objetivos; por esta razón el nuevo curso diseñado hace uso de las cuatro ramas de aprendizaje de la metodología de Cupi2. El curso de Clientes Web Enriquecidos está basado en un **aprendizaje activo** por parte de los estudiantes a través de una serie de actividades como debates y foros que permiten discutir acerca de la importancia del curso y sus temas y generan motivación en ellos para construir y compartir conocimiento en Desarrollo Web.

Por otra parte, ante las dificultades que surgen del Desarrollo Web, como el hecho de que las tecnologías utilizadas cambien rápidamente y que se necesite demasiado conocimiento, se requiere de una arquitectura de referencia para las aplicaciones

web independiente de la tecnología y se requiere de unos ejes conceptuales que generen los conocimientos y habilidades necesarias en los estudiantes para desarrollar aplicaciones web. El curso de Clientes Web Enriquecidos es desarrollado a través de una serie de ejercicios y ejemplos (haciendo uso del **aprendizaje mediante ejemplos**) correspondientes a un problema (haciendo uso del **aprendizaje basado en problemas**), soportados por la **arquitectura de referencia** desarrollada y resueltos con el conocimiento adquirido mediante el **aprendizaje incremental** de los nuevos **ejes conceptuales** propuestos que a su vez definen niveles en el curso (Ver Figura 3).

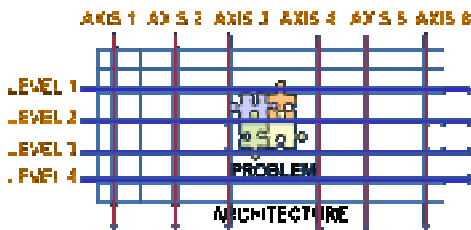


Figura 3. Estructura del nuevo curso basada en ejercicios y ejemplos correspondientes a un problema

4.1 Ejercicios basados en un problema

Lo que sigue es la aplicación de la arquitectura de referencia diseñada y el aprendizaje incremental de los nuevos ejes conceptuales definidos, en el ejercicio de Colegio Web (uno de los varios problemas que se diseñaron para el curso) basado en el problema de agregar estudiantes en un colegio, encontrar estudiantes en el colegio a través de diferentes criterios, modificar la información de cualquier estudiante y eliminar estudiantes del colegio. La aplicación de esta arquitectura y la metodología a este ejercicio se realizó con el fin de ver evidenciado en la práctica la arquitectura de referencia diseñada y de los nuevos ejes conceptuales definidos para el curso y de evaluar su eficacia para el aprendizaje de los estudiantes.

4.1.1 Arquitectura

El modelo de arquitectura que se propuso está basado en un cliente enriquecido que pide servicios a un servidor, teniendo en cuenta que el curso diseñado se orienta a la enseñanza y aprendizaje de aplicaciones de internet enriquecidas (RIAS).

La aplicación de esta arquitectura de referencia en los ejercicios de Colegio Web es desarrollada y aprendida incrementalmente, de manera que se muestra la arquitectura completa de Colegio Web después de que se han resuelto todos los ejercicios basados en este problema. Esta arquitectura de referencia refleja las bases de lo que se considera una arquitectura de una aplicación web enriquecida:

Se tiene entonces un **Cliente** que está compuesto de un **controlador** que es representado por scripts JS (llamados controladores) y tiene la función de capturar eventos generados por los componentes en el cliente (específicamente la vista) y por consiguiente hacer acciones en el mismo cliente, en el servidor o en el cliente y el servidor como una cadena de acciones; también se tiene una vista que es representada por componentes (que son divs HTML) compuestos de otros componentes HTML formando lo que se llama una página web o un template web; por otro lado se tiene un **Servidor** que está compuesto de un **controlador** que a su vez está compuesto por **servicios** (que son representados por Java Servlets para este caso de Colegio Web) y tienen la función de realizar operaciones sobre el modelo de datos y está compuesto por **recursos** (que son representados por imágenes JPG, estilos CSS, y templates JSP para este caso de Colegio Web) que son accedidos por el cliente; el Servidor está compuesto también de un **modelo** de datos (que es representado por archivos Java para este caso de Colegio Web).

4.1.2 Usabilidad

Parte del trabajo realizado fue generar una librería para el curso donde hubiera una lista de patrones de usabilidad clasificados de acuerdo al valor que generan a los usuarios y expresar una medida de que tanto una aplicación web está satisfaciendo las necesidades del usuario. Esta lista es ejemplificada incrementalmente en Colegio Web mediante el desarrollo de los diferentes ejercicios prácticos basados en este problema, donde es reflejada la aplicación de estos patrones a través de diferentes tecnologías como Ajax, JavaScript, DOM, etc. Lo que sigue es la aplicación de algunos patrones de usabilidad a los ejercicios basados en el problema de Colegio Web:



Figura 4. Patrones de usabilidad Breadcrumbs, ToolTips y Mapa del Sitio

4.1.3 Testing

Uno de los ejes conceptuales más importantes que se definieron es Testing. Lo que se hizo en este eje para los ejercicios de Colegio Web fue definir una serie de casos de prueba para requisitos funcionales (como agregar un

4.2 Métricas del trabajo hecho

Aquí se muestran algunas métricas del trabajo hecho con el curso, que a su vez sirven de soporte para este:

Cursos	Prerrequisitos	Secciones
CS1 RIAs.1	CS1, CS2, CS3, Ingeniería de Software	2

Variable	Valor
Ejercicios	27
Ejemplos	27
Presentaciones (investigación)	26

Variable	Valor
Créditos	3
Tamaño (estudiantes)	32
Semanas	15

□ POR SEMESTRE

Figura 9. Métricas del trabajo hecho con el curso

Como se puede ver hubo un aumento en la demanda del curso como resultado del trabajo realizado: un incremento del 100% en el número de estudiantes comparado con el primer semestre del año 2008; más unidades de conocimiento fueron cubiertas, haciendo 27 ejercicios, 27 ejemplos y 26 presentaciones donde se desarrolló y aplicó una arquitectura de referencia (independiente de la tecnología), 6 ejes conceptuales cubriendo gran parte de lo que es Testing Web (Pruebas Unitarias y Pruebas de Carga), Patrones de Usabilidad Web (30 patrones), Seguridad Web, Herramientas Web (más de 6 herramientas) y Lenguajes Web (más de 9 lenguajes o tecnologías) y la metodología de Cupi2 con sus 4 ramas de aprendizaje; finalmente se puede ver que hay un uso significativo de metodologías y patrones de Software en aplicaciones web.

5. TRABAJO RELACIONADO

Hay un gran número de trabajos que se han enfocado en el diseño de cursos para enseñar y aprender a desarrollar Aplicaciones Web al igual que este que se presenta en este paper. Algunos de estos se enfocan en las tecnologías web, otros en la metodología para enseñar web, otros en el uso de herramientas para facilitar el aprendizaje de programación web.

Boroni en [8] señala que la eliminación de la dependencia de plataforma para producir buenos sistemas de software es necesaria a la hora de diseñar una aplicación web; en este trabajo se hace uso de esta idea para justificar el diseño de una arquitectura de referencia que no depende de la tecnología y permite resolver la dificultad del cambio constante de tecnologías en las aplicaciones web actuales.

Hoganson establece en [9] que la arquitectura de aplicación es un concepto que encapsula el conocimiento básico y las habilidades de Ciencias de la Computación con un conocimiento y análisis de alto nivel. Esto se aplica en el trabajo realizado en cuanto a que la arquitectura de referencia diseñada soporta las cuatro ramas de aprendizaje que establece la metodología de Cupi2 y permite la aplicación de los nuevos ejes conceptuales que permiten generar habilidades y conocimientos básicos y de alto nivel en los estudiantes en cuanto a Desarrollo de Aplicaciones Web.

The Teaching Tool ETE [10] resuelve el problema de enseñar y aprender a programar en Web incorporando la Web y sus tecnologías en la clase. A diferencia de la anterior herramienta, el presente trabajo hace uso de una metodología exitosa para enseñar y aprender a programar (Cupi2) basada en cuatro ramas de aprendizaje, aplicándola para un curso de programación Web como el diseñado.

Yue y Ding en [3] establecen que enseñar un curso de desarrollo web es un reto debido al gran conocimiento que se requiere y a las tecnologías cambiantes rápidamente. En su trabajo proponen que para enfrentar este reto el curso debe estar enfocado en cubrir cuantas tecnologías web se pueda y todo el contenido que se pueda en cada una de estas. Parte del trabajo que se presenta en este paper muestra una solución diferente para estas dificultades con respecto al trabajo mencionado: el diseño de la arquitectura de referencia independiente de la tecnología y la definición de los nuevos ejes conceptuales para aplicaciones web le evitan al estudiante integrar tecnologías (como tendría que hacer un estudiante en un curso como el que presentan Yue y Ding) y le permite generar una gran cantidad de habilidades y conocimientos a la hora de desarrollar una aplicación web.

6. CONCLUSIONES

En este artículo se ha presentado la estructuración y diseño de un curso de Desarrollo de Aplicaciones Web para el currículo de Ingeniería de Sistemas de la Universidad de los Andes, resolviendo las dificultades que se presentan en la implementación de un curso de Programación como este, mediante la aplicación de las cuatro ramas de aprendizaje de la metodología de Cupi2, y logrando generar los conocimientos y habilidades necesarias para poder construir aplicaciones que no dependan de una tecnología específica y den soporte al rol actual que cumplen los usuarios en la web y a su forma de interacción.

Para poder implementar la metodología de Cupi2 en este curso, se definió una arquitectura de referencia independiente de la tecnología, que da soporte a los diferentes ejercicios, actividades, escenarios de trabajo y problemas con los que se realiza la enseñanza y se desarrolla el proceso de aprendizaje de los nuevos ejes conceptuales definidos para el curso. Estos ejes permiten generar las habilidades y conocimientos necesarios para poder desarrollar aplicaciones web bajo cualquier tecnología y adaptarse a cualquier especialización en esta área.

Adicional a todo esto se ha estructurado un curso basado en ejercicios y ejemplos que corresponden a un problema, soportados por la arquitectura de referencia diseñada y que permiten el aprendizaje incremental de los ejes conceptuales del curso.

A futuro, se pretende aplicar nuevas tecnologías web sobre las bases construidas en el presente trabajo, para evidenciar la independencia de las tecnologías y darles más libertad a los estudiantes en el desarrollo de los problemas y ejercicios del curso, generando más conocimientos y habilidades en desarrollo web.

7. REFERENCIAS

5. TRABAJO RELACIONADO

Hay un gran número de trabajos que se han enfocado en el diseño de cursos para enseñar y aprender a desarrollar Aplicaciones Web al igual que este que se presenta en este paper. Algunos de estos se enfocan en las tecnologías web, otros en la metodología para enseñar web, otros en el uso de herramientas para facilitar el aprendizaje de programación web.

Boroni en [8] señala que la eliminación de la dependencia de plataforma para producir buenos sistemas de software es necesaria a la hora de diseñar una aplicación web; en este trabajo se hace uso de esta idea para justificar el diseño de una arquitectura de referencia que no depende de la tecnología y permite resolver la dificultad del cambio constante de tecnologías en las aplicaciones web actuales.

Hoganson establece en [9] que la arquitectura de aplicación es un concepto que encapsula el conocimiento básico y las habilidades de Ciencias de la Computación con un conocimiento y análisis de alto nivel. Esto se aplica en el trabajo realizado en cuanto a que la arquitectura de referencia diseñada soporta las cuatro ramas de aprendizaje que establece la metodología de Cupi2 y permite la aplicación de los nuevos ejes conceptuales que permiten generar habilidades y conocimientos básicos y de alto nivel en los estudiantes en cuanto a Desarrollo de Aplicaciones Web.

The Teaching Tool ETE [10] resuelve el problema de enseñar y aprender a programar en Web incorporando la Web y sus tecnologías en la clase. A diferencia de la anterior herramienta, el presente trabajo hace uso de una metodología exitosa para enseñar y aprender a programar (Cupi2) basada en cuatro ramas de aprendizaje, aplicándola para un curso de programación Web como el diseñado.

Yue y Ding en [3] establecen que enseñar un curso de desarrollo web es un reto debido al gran conocimiento que se requiere y a las tecnologías cambiantes rápidamente. En su trabajo proponen que para enfrentar este reto el curso debe estar enfocado en cubrir cuantas tecnologías web se pueda y todo el contenido que se pueda en cada una de estas. Parte del trabajo que se presenta en este paper muestra una solución diferente para estas dificultades con respecto al trabajo mencionado: el diseño de la arquitectura de referencia independiente de la tecnología y la definición de los nuevos ejes conceptuales para aplicaciones web le evitan al estudiante integrar tecnologías (como tendría que hacer un estudiante en un curso como el que presentan Yue y Ding) y le permite generar una gran cantidad de habilidades y conocimientos a la hora de desarrollar una aplicación web.

6. CONCLUSIONES

En este artículo se ha presentado la estructuración y diseño de un curso de Desarrollo de Aplicaciones Web para el currículo de Ingeniería de Sistemas de la Universidad de los Andes, resolviendo las dificultades que se presentan en la implementación de un curso de Programación como este, mediante la aplicación de las cuatro ramas de aprendizaje de la metodología de Cupi2, y logrando generar los conocimientos y habilidades necesarias para poder construir aplicaciones que no dependan de una tecnología específica y den soporte al rol actual que cumplen los usuarios en la web y a su forma de interacción.

Para poder implementar la metodología de Cupi2 en este curso, se definió una arquitectura de referencia independiente de la tecnología, que da soporte a los diferentes ejercicios, actividades, escenarios de trabajo y problemas con los que se realiza la enseñanza y se desarrolla el proceso de aprendizaje de los nuevos ejes conceptuales definidos para el curso. Estos ejes permiten generar las habilidades y conocimientos necesarios para poder desarrollar aplicaciones web bajo cualquier tecnología y adaptarse a cualquier especialización en esta área.

Adicional a todo esto se ha estructurado un curso basado en ejercicios y ejemplos que corresponden a un problema, soportados por la arquitectura de referencia diseñada y que permiten el aprendizaje incremental de los ejes conceptuales del curso.

A futuro, se pretende aplicar nuevas tecnologías web sobre las bases construidas en el presente trabajo, para evidenciar la independencia de las tecnologías y darles más libertad a los estudiantes en el desarrollo de los problemas y ejercicios del curso, generando más conocimientos y habilidades en desarrollo web.

7. REFERENCIAS

- [1] Chirstina Dörge, Carsten Schulte. 2008. What are Information Techonology's Key Qualifications? Proceedings of the 13th annual conference on Innovation and technology in computer science education
- [2] J. Villalobos, R. Casallas, K. Marcos. "El Reto de Diseñar un Primer Curso de Programación de Computadores". XIII Congreso Iberoamericano de Educación Superior en Computación, Cali, Colombia, Octubre 2005.
- [3] Kwok-Bun Yue, Wei Ding. Design and evolution of an undergraduate course on web application development, Proceedings of the 9th annual SIGCSE

conference on Innovation and technology in computer science education, June 28-30, 2004, Leeds, United Kingdom

- [4] J. Villalobos, M. Vela. June 2008. "CUPI2-An Active Learning and ProblemBased Learning Approach to Teaching Programming", 8th ALE International Workshop, Bogotá - Colombia.
- [5] Powell, T. A. Web Design: The Complete Reference. Berkeley, CA: Osborne/McGraw-Hill, 2000.
- [6] Gordon, L. A., Loeb, M. P., Lucyshyn, W. and Richardson, R., 2006 CSI/FBI Computer Crime Survey, in Eleventh Annual. 2006, Computer Security Institute.
- [7] Nalneesh Gaur. 2000. Assessing the Security of Your Web Applications. Linux Journal, Volume 2000, Issue 72es (April 2000), Article 3
- [8] Christopher M. Boroni , Frances W. Goosey , Michael T. Grinder , Rockford J. Ross, A paradigm shift! The Internet, the Web, browsers, Java and the future of computer science education, ACM SIGCSE Bulletin, v.30 n.1, p.145-152, Mar. 1998.
- [9] Kenneth Hoganson, Computer science curricula in a global competitive environment, Journal of Computing Sciences in Colleges, v.20 n.1, p.168-177, October 2004.
- [10] James H. Hu, Grace M. Hu. Teaching web design and web programming using an enhanced teaching environment, v.61, p.148-154, 2004.
- [11] Engin Bozdag, Ali Mesbah, Arie van Deursen. 2007. A Comparison of Push and Pull Techniques for Ajax. Web Site Evolution, 2007. WSE 2007. 9th IEEE International Workshop.
- [12] Harry M. Sneed, Testing a Web Application, Proceedings of the Web Site Evolution, Sixth IEEE International Workshop, p.3-10, September 11-11, 2004
- [13] Filippo Ricca , Paolo Tonella, Web Testing: a Roadmap for the Empirical Research, Proceedings of the Seventh IEEE International Symposium on Web Site Evolution, p.63-70, September 26-26, 2005 [doi>10.1109/WSE.2005.23].