

**DESARROLLO DE HERRAMIENTAS LÚDICAS PARA UN
ENTORNO VIRTUAL QUE FOMENTE EL DESARROLLO DE LA
CREATIVIDAD EN LOS ESTUDIANTES.**

CARLOS ALFONSO TARAZONA T.

SERGIO IVAN RODRIGUEZ R.

UNAB



UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA	
VENDEDOR	IDENTIFICACION
	TG/34.04 T178d
	057754

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERA DE SISTEMAS
BUCARAMANGA
2004**

**DESARROLLO DE HERRAMIENTAS LÚDICAS PARA UN
ENTORNO VIRTUAL QUE FOMENTE EL DESARROLLO DE LA
CREATIVIDAD EN LOS ESTUDIANTES.**

CARLOS ALFONSO TARAZONA T.

SERGIO IVAN RODRIGUEZ R.

**Director
SONIA CRISTINA GAMBOA
Ingeniero de Sistemas**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERIA DE SISTEMAS
BUCARAMANGA
2004**

RESUMEN

El trabajo se realizó utilizando las siguientes tecnologías: SVG, JAVA, MYSQL, UML, LINUX, las cuales abren un nuevo espacio de información y comunicación, mostrando procesos de aprendizaje en los diferentes campos del conocimiento.

Es por esto que estas herramientas funcionarán en un entorno virtual, para ser aprovechada en un ambiente Web.

La herramienta realizada consta de tres juegos, los cuales tienen en cuenta un espacio para la interacción, la memorización y el entretenimiento además permiten que cualquier persona pueda acceder la herramienta teniendo en cuenta que se ha montado en Internet.

AGRADECIMIENTOS

Sonia Cristina Gamboa, ingeniera de sistemas director del proyecto por sus valiosas orientaciones.

Lilia Nayibe Gelvez, ingeniera de sistemas y Claudia Salazar, licenciada en educación por habernos guiado en el alcance de los objetivos propuestos del proyecto.

Universidad Autónoma de Bucaramanga por facilitar los servicios en el desarrollo de la tecnología utilizada.

Tabla de figuras

	Pág.
	4
Figura 1 Orígenes del Proceso Unificado	51
Figura 2 Mapa de navegación de Usuario	52
Figura 3 Mapa del sitio de administrador	53
Figura 4. Diagrama de caso de uso: Administrador	54
Figura 5, caso de uso usuario	55
Figura 6. Diagrama de casos de uso: Creación de administrador	56
Figura 7. Diagrama de secuencia: Creación de administrador	57
Figura 8, diagrama de caso de uso: consultar juegos activos	58
figura 9, Diagrama de secuencia: consultar juegos activos	59
Figura 10, Diagrama de caso de uso: consultar juegos terminados	60
Figura 11, Diagrama de secuencia: consultar juegos terminados	61
Figura 12, Diagrama de casos de uso: Creación de usuarios	62
Figura 13, Diagrama de secuencias: Creación de usuarios	63
Figura 14, Diagrama de casos de uso: Creación de juegos	64
Figura 15 , Diagrama de secuencia : Creación de juegos	65
Figura 16, Diagrama de casos de uso: Enviar y recibir comentarios	66
Figura 17, Diagrama de secuencia: Enviar y recibir comentarios	67
Figura 18, Diagrama de caso de uso: Crear curso	67
Figura 19, Diagrama de secuencia: Crear curso	68
Figura 20, Diagrama de casos de uso : Modificar datos de entrada	69
Figura 21, Diagrama de secuencia: Modificar datos de entrada	70
Figura 22, Diagrama de casos de uso: Empezar juego	71
Figura 23, Diagrama de secuencia: Empezar juego	72
Figura 24, Diagrama de casos de uso: Consulta de ayuda.	73
Figura 25, Diagrama de secuencia: Consulta de ayuda.	74
Figura 26, diagrama de casos de uso: Juego editor grafico	75
Figura 27, Diagrama de secuencias: Juego editor grafico	

Tabla de contenido

	Pág.
INTRODUCCIÓN	2
1. MARCO TEORICO	2
1.1 Desarrollo de herramientas software	3
1.1.1 Proceso unificado de desarrollo del software.	3
1.1.1.1 Orígenes	4
1.1.1.2 Características	6
1.2 Herramientas de desarrollo.	6
1.2.1 Uml	7
1.2.1.1 Objetivos del UML	8
1.2.1.2 Arquitectura del UML	8
1.2.1.3 Áreas conceptuales de uml	10
1.2.1.4 Diagrama de casos de uso	10
1.2.1.5 Diagrama de secuencia y colaboración	11
1.2.1.6 Diagrama de clases	11
1.3 Metodología	11
1.3.1 Resolución de problemas	11
1.3.1.1 Algunas investigaciones previas en educación superior	12
1.3.1.2 Que es un problema	13
1.3.1.3 Características para la resolución de problemas.	14
1.3.1.4 Habilidades requeridas para la resolución de problemas.	14
1.3.1.4.1 Habilidades cognitivas	15
1.3.1.4.2 Habilidades cognoscitivas	18
1.3.1.5 Métodos para la resolución de problemas	
1.4 Implementación de la resolución de problemas en la herramienta	19

1.5 DISEÑO METODOLÓGICO	20
2 TECNOLOGÍAS DE DESARROLLO	25
2.1 JAVA 2, STANDRAD EDITION V 1.4	25
2.2 JAVABEANS	25
2.2.1 Características de JavaBeans	25
2.3 JAVA SERVER PAGES	26
2.3.1 Introducción	26
2.3.2 ventajas jsp	26
2.3.3 Arquitectura de jsp	27
2.3.4 Elementos de jsp	28
2.3.4.1 Directivas	28
2.3.4.2 Declaraciones	28
2.3.4.3 Expresiones	28
2.3.4.4 Scriptlets	29
2.3.4.5 Comentarios	29
2.4 JAVA DATA BASE CONNECTION	30
2.5 SVG	32
2.6 SERVIDOR WEB JAKARTA-TOMCAT	34
2.6.1 Versiones del servidor Tomcat	34
2.7 MYSQL	35
2.7.1 Introducción	35
2.7.2 Historia de mysql	36
2.7.2.1 Características de Mysql	36
2.8 LINUX	37
2.8.1 Introducción	37
2.8.2 Breve historia de Linux	37
2.8.3 Comandos para creación, gestión de archivos y directorios	39

	45
	45
3. DESARROLLO DE LA APLICACIÓN	46
3.1 Introducción a la aplicación	46
3.2 Descripción de juegos	47
3.2.1 Historias	48
3.2.2 Combinaciones	49
3.2.3 Editor Gráfico	49
3.3 Aporte de la herramienta al estudiante	49
3.4 Aporte de la herramienta al profesor	50
3.5 Aporte de la herramienta al investigador	50
3.6 Mapa de navegación por usuario	53
3.7 Mapa de sitio administrador	53
3.8 Modelamiento y diseño de la aplicación	54
3.8.1 Caso de uso administrador	55
3.8.2 Caso de uso usuario	57
3.8.3 Creación de administrador	58
3.8.4 Consulta de juego activos	60
3.8.5 Consulta de juegos terminados	62
3.8.6 Creación de usuario	64
3.8.7 Crear juego	66
3.8.8 Enviar y recibir comentarios	67
3.8.9 Crear curso	69
3.8.10 Modificar datos de entrada	71
3.8.11 Empezar juego	73
3.8.12 Consulta de ayuda	76
3.8.13 Editor gráfico	77
CONCLUSIONES	78
RECOMENDACIONES	
BIBLIOGRAFÍA	

INTRODUCCIÓN

Las nuevas tecnologías de la información y de las comunicaciones posibilitan la creación de un nuevo espacio para las relaciones humanas, porque muestran nuevos procesos de aprendizaje en los diferentes campos del conocimiento a través de las redes de comunicación.

Día a día, los entornos virtuales, han generado un enorme interés en todos los ámbitos de nuestra sociedad y se convierten en una herramienta de información y comunicación apoyada en la forma de aprender.

Es por esto que las herramientas trabajadas, estarán montadas en un entorno que sea compatible con la tecnología utilizada en el proyecto, de tal forma que puedan ser aprovechadas en un ambiente web, teniendo en cuenta el espacio para el desarrollo de las diferentes habilidades en los estudiantes.

El desarrollo de esta aplicación se presenta por medio de un prototipo, en el cual se quiere mostrar el funcionamiento de cada uno de los juegos como herramienta lúdicas y que se ofrece como servicio para fomentar el desarrollo de la creatividad en los usuarios.

1. MARCO TEORICO

La educación actual afronta múltiples retos y uno de ellos es dar respuesta a los profundos cambios en los diferentes campos que se prevén en la sociedad de la información. Y es que cuando necesitamos tener acceso a cualquier información, se hace necesario satisfacer los requerimientos en cuanto a tecnología y comunicaciones, los cuales combinan aspectos educativos y computacionales.

Los aspectos nombrados anteriormente, los hemos puesto en práctica en la solución de nuestro proyecto, pues lo que se persigue es que a través de herramientas lúdicas " *en este caso estamos hablando de juegos*", se fomente el desarrollo de la creatividad.

Estos juegos se presentarán en un ambiente web, en el que se busca que el usuario del sistema se encuentre a gusto con lo que está haciendo y que de una u otra forma sea él, el que se sienta motivado a interactuar con la herramienta por iniciativa propia más no porque se lo impongan.

1.1 DESARROLLO DE HERRAMIENTAS SOFTWARE

Cuando se quiere realizar software es necesario optar por una metodología adecuada para el cumplimiento de los requerimientos de los usuarios teniendo principalmente en cuenta lo propuesto en un plan inicial.

El proceso unificado de desarrollo de software ofrece una alternativa viable para realizar un buen diseño que cumpla con las expectativas de los clientes.

UML es la alternativa adecuada para el desarrollo de nuestra aplicación ya que está estructurado por un conjunto de vistas que buscan especificar diferentes elementos del sistema de tal manera que puedan integrarse para facilitar el análisis, el diseño, la implementación y las pruebas del software.

1.1.1 Proceso unificado de desarrollo del software.

1.1.1.1 Orígenes. ¹El antecedente más importante lo ubicamos en 1967 con la Metodología Ericsson (Ericsson Approach), ésta es una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; entre los años de 1987 a 1995 Jacobson funda la compañía "Objectory AB" y lanza el proceso de desarrollo Objectory (abreviación de Object Factory), posteriormente en 1995 "Rational Software Corporation" adquiere "Objectory AB" y es entre 1995 y 1997 que se desarrolla "Rational Objectory Process (ROP)" fruto del encuentro y evolución de Objectory 3.8 y la Metodología Rational (Rational Approach) que adopta por primera vez UML como lenguaje de modelamiento.

A principios de los noventas, la guerra de los métodos hizo evidente la necesidad de unificar criterios, es así como Grady Booch autor del método Booch y James Rumbaugh (desarrollador para General Electric) se unieron en Rational en 1994, después en 1995 se une Jacobson y gracias al esfuerzo de varias compañías y metodologistas evolucionó UML hasta ser un estándar en 1997, el cual es adoptado en todos los modelos del ROP. Desde ese entonces y a la cabeza de Booch, Jacobson y Rumbaugh, Rational ha desarrollado e incorporado diversos elementos para expandir el ROP, destacándose especialmente el flujo de trabajo conocido como modelamiento del negocio, es así como en junio del 1998 se lanza Rational Unified Process 5.0 evolucionado hasta el momento de elaboración de

¹ <http://atenea.ucauca.edu.co/~gramirez/archivos/AnotacionesRUP.pdf>, Universidad del cauca, noviembre de 2003.

este documento bajo el nombre de RUP.² La evolución y orígenes de este proceso de desarrollo se puede visualizar mejor en la figura 1(orígenes del proceso unificado).

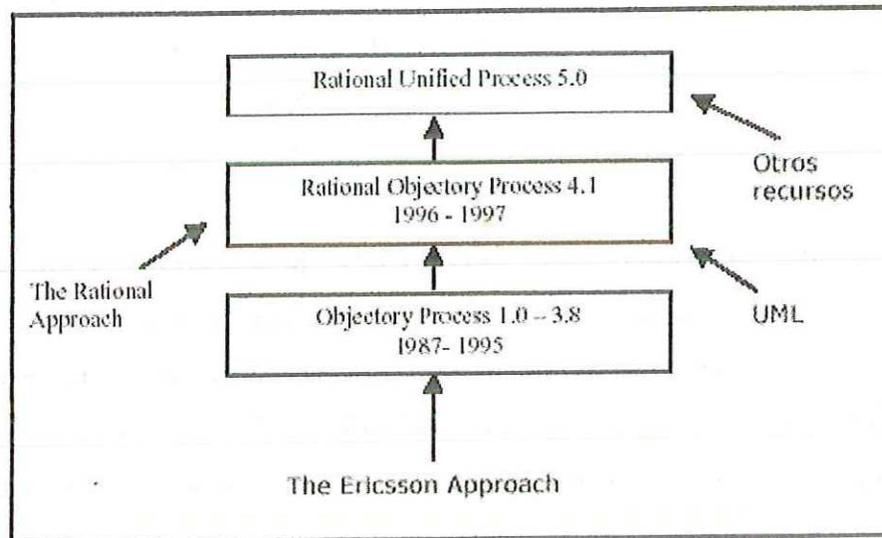


Figura 1 Orígenes del proceso Unificado[JBR99]

1.1.1.2 Características

- **Guiado / manejado por casos de uso:** La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas. Un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

² <http://atenea.ucauca.edu.co/~gramirez/archivos/AnotacionesRUP.pdf>, Universidad del cauca, noviembre de 2003

- **Proceso Integrado:** ⁴Se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración; el proceso unificado establece una estructura que integra todas estas facetas. Además esta estructura cubre a los vendedores y desarrolladores de herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo, para construir los diferentes modelos e integrar el trabajo a través del ciclo de vida y a través de todos los modelos.

1.2 HERRAMIENTAS DE DESARROLLO.

1.2.1 **Uml.** ⁵Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes. Existían diversos métodos y técnicas Orientadas a

⁴ <http://atenea.ucauca.edu.co/~gramirez/archivos/AnotacionesRUP.pdf>, Universidad del cauca, noviembre de 2003.

⁵ <http://www.creangel.com/uml/home.html>, Rational Rose, Copyright © 2000-2003.

Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que generó la creación del UML como estándar para el modelamiento de sistemas de software principalmente, pero con posibilidades de ser aplicado a todo tipo de proyectos.

1.2.1.2 Objetivos del UML

- ⁶UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.

⁶ <http://www.creangel.com/uml/intro.html>, Rational Rose, Copyright © 2000-2003.

- Imponer un estándar mundial.

1.2.1.3 *Arquitectura del UML.*

⁷Arquitectura de cuatro capas, definida a fin de cumplir con la especificación Meta Object Facility del OMG:

- *Meta-metamodelo*: Define el lenguaje para especificar meta modelos.
- *Metamodelo*: Define el lenguaje para especificar modelos.
- *Modelo*: Define el lenguaje para describir un dominio de información.
- *Objetos de usuario*: Define un dominio de información específico.

1.2.13 Áreas conceptuales de uml. Los conceptos y modelos de UML pueden agruparse en las siguientes áreas conceptuales.

- **Estructura estática**: Cualquier modelo preciso debe primero definir su universo, esto es, los conceptos clave de la aplicación, sus propiedades internas, y las relaciones entre cada una de ellas. Este conjunto de construcciones es la estructura estática. Los conceptos de la aplicación son modelados como clases, cada una de las cuales describe un conjunto de objetos que almacenan información y se comunican para implementar un comportamiento. La información que almacena es modelada como atributos; La estructura estática se expresa con diagramas de clases y puede usarse para generar la mayoría de las declaraciones de estructuras de datos en un programa.
- **Comportamiento dinámico**: Hay dos formas de modelar el comportamiento, una es la historia de la vida de un objeto y la forma como

⁷ <http://www.creangel.com/uml/intro.html>, Rational Rose, Copyright © 2000-2003.

interactúa con el resto del mundo y la otra es por los patrones de comunicación de un conjunto de objetos conectados, es decir la forma en que interactúan entre sí. La visión de un objeto aislado es una maquina de estados, muestra la forma en que el objeto responde a los eventos en función de su estado actual. La visión de la interacción de los objetos se representa con los enlaces entre objetos junto con el flujo de mensajes y los enlaces entre ellos. Este punto de vista unifica la estructura de los datos, el control de flujo y el flujo de datos.

- **Construcciones de implementación:** ⁸Los modelos UML tienen significado para el análisis lógico y para la implementación física. Un componente es una parte física reemplazable de un sistema y es capaz de responder a las peticiones descritas por un conjunto de interfaces. Un nodo es un recurso computacional que define una localización durante la ejecución de un sistema. Puede contener componentes y objetos.
- **Organización del modelo:** La información del modelo debe ser dividida en piezas coherentes, para que los equipos puedan trabajar en las diferentes partes de forma concurrente. El conocimiento humano requiere que se organice el contenido del modelo en paquetes de tamaño modesto. Los paquetes son unidades organizativas, jerárquicas y de propósito general de los modelos de UML. Pueden usarse para almacenamiento, control de acceso, gestión de la configuración y construcción de bibliotecas que contengan fragmentos de código reutilizable.
- **Mecanismos de extensión:** UML tiene una limitada capacidad de extensión pero que es suficiente para la mayoría de las extensiones que requiere. Un estereotipo es una nueva clase de elemento de modelado con

⁸ <http://www.creangel.com/uml/intro.html>, Rational Rose, Copyright © 2000-2003.

la misma estructura que un elemento existente pero con restricciones adicionales.

1.2.1.3 Diagrama de casos de uso. El modelo de casos de uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre como utilizar el sistema. La mayoría de los sistemas tiene muchos tipos de usuarios. Cada tipo de usuarios se representa por medio de un actor. Los actores utilizan el sistema al interactuar con los casos de uso. Todos los actores y casos de uso del sistema forman parte del modelo de casos de uso.

Un diagrama de casos de uso describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores (primario, secundario, material externo, otros sistemas) y sus relaciones.

Estos diagramas son importantes para especificar, visualizar como se comportan los elementos dentro de un sistema.

Los elementos conforman sistemas, subsistemas y clases son entendibles para que cualquier usuario entienda que tarea realiza.

1.2.1.4 Diagrama de secuencia y colaboración. Un diagrama de colaboración recuerda un diagrama de clases pero contiene instancias y enlaces en lugar de clases y asociaciones. Este diagrama muestra como interactúan los objetos secuencialmente o en paralelo, numerando los mensajes que se envía de uno a otro.

Además, son tipos de diagrama de iteración, los cuales muestran un iteración consistiendo de un conjunto de objetos y sus relaciones, los diagramas de iteración direccionan vistas dinámicas de un sistema.

Este diagrama enfatiza la organización de sus relaciones.

1.2.1.5 **Diagrama de clases.** ⁹Se utilizan generalmente para mostrar clases y sus relaciones, pero también pueden utilizarse para mostrar subsistemas e interfases.

Estos diagramas son los más comunes del modelado de sistemas orientado a objetos. Muestra una manera estática de la estructura de información del sistema y la viabilidad que tiene cada clase, dadas por sus relaciones con los demás modelos

1.3 METODOLOGÍA

1.3.1 Resolución de problemas

1.3.1.2 Algunas investigaciones previas en educación superior.

¹⁰Se pueden citar algunas investigaciones en educación superior en la Universidad de Carolina del Norte, investigadores como Felder y Brent (1994) han realizado estudios longitudinales adaptando y diseñando procedimientos para implementar técnicas de aprendizaje cooperativo en cursos que se basan en la resolución de problemas. Señalan además, algunas de las advertencias para la implementación del aprendizaje cooperativo y remarcan los beneficios hacia los estudiantes y los coordinadores. Johnson y Johnson (1991) también se han preocupado en sus estudios sobre la efectividad de los aprendizajes cooperativos en educación superior. Este aprendizaje puede ocurrir tanto dentro como fuera del aula. La idea básica es que los estudiantes aprendan "*haciendo*" y no sólo "*escuchando*" durante las clases.

⁹ El proceso unificado de desarrollo de software, Ivar Jacobson, Grady Booch, James Rumbaugh, Addison Wesley, Copyright © 2000.

¹⁰ <http://www.rimed.cu/Ense%F1anzas/SUPERIOR.ASP>, Ministerio de educación de la República de Cuba, agosto de 2003.

¹¹(Bonwell et al., 1991) y que durante las clases realicen trabajos cooperativos y colaborativos pudiendo incluir fuera de la clase actividades tales como: poner en marcha trabajos de investigación, completar la resolución de problemas, preparar informes, presentar informes, diseñar proyectos, etc.

Felder y Brent (1994) han evidenciado que los estudiantes después del trabajo en grupo cooperativo tienden a exhibir: un mejor rendimiento académico, un mejor nivel de razonamiento, desarrollo de habilidades de pensamiento, entendimiento más profundo del material de aprendizaje, más tiempo dedicado a las tareas y menos interrupción en clase, mayor motivación intrínseca hacia el aprendizaje y sus logros y mayor capacidad para ver situaciones desde otras perspectivas.

La falta de metodologías de estudio, falta de estrategias para resolución de problemas, las dificultades para diferenciar la información trae problemas como la falta de atención y el conocimiento frágil de los alumnos (Perkins, 1995).

De acuerdo a los problemas detectados, se puede decir que un cambio en la metodología de trabajo de los estudiantes podría mejorar su rendimiento.

1.3.1.3 **Que es un problema.** ¹²Un problema es una situación con un objetivo a lograr, que requiere del estudiante para ser cumplido, una serie de operaciones que permitan resolver la o las incógnitas contenidas en ella. Para que sea un verdadero problema, el estudiante no debe disponer de antemano del conocimiento de las estrategias a seguir para su resolución

Hay una diferencia básica entre el concepto "problema" y "ejercicio". No es lo mismo hacer un ejercicio que resolver un problema. El problema es una situación que requiere del estudiante un proceso de razonamiento que lo lleve

¹¹ <http://www.rimed.cu/Ense%F1anzas/SUPERIOR.ASP>, Ministerio de educación de la republica de cuba, agosto de 2003.

¹² <http://www.xtec.es/~jcorder1/problema.htm>, Juan Antonio Cordero, 1996-2000.

¹³a analizar la situación y desarrollar una estrategia para obtener la información que necesita y el ejercicio es resolver de una forma mecánica algún algoritmo evitando la problemática para resolver un problema.

El estudiante dedica muy poco tiempo para el desarrollo de los problemas que se ve enfrentado durante su educación. La dificultad no es un motivo para que tenga más tiempo de dedicación a resolverlo. En parte ello es consecuencia de la falta de hábitos en esforzarse por conseguir las propias metas. Es una obviedad, no sólo que no disfrutan ante los retos intelectuales sino, que no están dispuestos a "malgastar" el tiempo pensando.

Este es uno de los grandes motivos por los cuales la herramienta de juegos lúdicos intentar romper este círculo vicioso y hacerles disfrutar de los resultados logrados a través de la manipulación e ingenio que pueda tener el estudiante, para esto es necesario cumplir con características de la resolución de problemas

1.3.1.4 Características para la resolución de problemas.

- Requiere actividad mental, un pensamiento directivo. Parte de un estado de incertidumbre, de una información inconsciente.
- La meta o solución está relacionada con el grado de definición de objetivos.
- Parte de una limitación del sistema cognitivo: atención, calidad de los datos. Memoria a largo plazo, información sobre soluciones. Memoria operativa donde se aplican las estrategias de solución.
- Se caracteriza por la seriedad de las operaciones. No se puede llegar a la meta directamente desde el estado inicial.
- La tarea del solucionador, es elaborar una estructura que represente e incluya estados intermedios para alcanzar la meta.

¹³ <http://www.xtec.es/~jcorder1/problema.htm>, Juan Antonio Cordero, 1996-2000

1.3.1.5 Habilidades requeridas para la resolución de problemas.

¹⁴La enseñanza-aprendizaje por medio de habilidades cognitivas, cognoscitivas, nos han llevado a implementar estrategias de apoyo como herramientas de carácter lúdico que tienen el objetivo de infundir la creatividad en los estudiantes y llevar el control del procesamiento de la información que se suministra en las diversas asignaturas de los planes de estudio.

1.3.1.5.1 **Habilidades cognitivas.** El concepto de Habilidad Cognitiva es una idea que enfatiza que el sujeto no solo adquiere los contenidos mismos sino que también aprende el proceso que uso para hacerlo: aprende no solamente lo que aprendió sino como lo aprendió.

Análisis

Consiste básicamente en la descomposición del objeto de aprendizaje.

El análisis comprende las siguientes etapas:

- Ubicación espacio temporal del objeto de aprendizaje
- Identificación de componentes
- Comparación de componentes con respecto a ubicación, forma, estructura y función.
- Clasificación de componentes de acuerdo con su función.
- Jerarquización de los componentes que constituyen la estructura del tema global de aprendizaje.

¹⁴ <http://espanol.geocities.com/criteriosevaluacion/procesocognitivo.htm>, Instituto Técnico Superior "Dámaso Zapata, Capacitación docente, Bucaramanga, Marzo de 2003.

Síntesis

El proceso de síntesis conduce a la formación de estructuras intelectuales, con las cuales se es capaz de interpretar, valorar y transformar la realidad, de manera original y creativa, de acuerdo con el nivel de desarrollo de cada educando.

La síntesis intelectual como resultado de los procesos de comprensión y análisis, lleva al descubrimiento y construcción de principios, leyes, definiciones y patrones de comportamiento que constituyen los valores, los cuales permiten elegir, decidir y obrar por motivos racionales, antes que por pasiones o emociones.

La capacidad de síntesis puede identificarse en el alumno cuando es capaz de construir: hipótesis o alternativas de solución a problemas específicos que exijan una síntesis; elaborar ensayos teóricos de un tema dado, con base en la síntesis; y formular definiciones, principios o leyes, a partir del análisis de los componentes y operaciones de un objeto de estudio.

1.3.1.5.2 **Habilidades cognoscitivas.**¹⁵ Es la construcción del conocimiento y la formación intelectual en el proceso de aprendizaje, que se adquieren mediante la capacidad dinámica que posee el ser humano para conceptualizar, comprender, generalizar y desarrollar todas las demás acciones de carácter intelectual

Conceptualización

Es la etapa o proceso de construcción de la imagen o representación mental del objeto, situación o acción percibida por los sentidos. El concepto o idea, adquirido por la observación, la atención y la percepción, constituye el material u objeto primario, que permite iniciar el proceso mental del conocimiento humano.

¹⁵ <http://espanol.geocities.com/criteriosevaluacion/procesocognitivo.htm>, Instituto Técnico Superior "Dámaso Zapata, Capacitación docente, Bucaramanga, marzo de 2003.

Mientras una educación con base en temas o contenidos teóricos y en su memorización, termina en el concepto, sin ninguna elaboración, la educación por procesos correctamente desarrollada se inicia con el concepto, que sirve de material para que la mente construya el conocimiento a través de los demás procesos intelectuales.

El concepto se origina en la percepción y se va perfeccionando con el desarrollo cognoscitivo en estructuras conceptuales cada vez más elaboradas, abstractas y universales.

Los conceptos pueden clasificarse de acuerdo con su origen en:

- Conceptos perceptivos: Se construyen ante la presencia de los objetos.
- Conceptos imaginativos: Se construyen a partir de las imágenes de objetos ya conocidos o mediante combinación de imágenes.
- Conceptos abstractivos: Se construyen a partir de otros conceptos o estructuras mentales.

"El desarrollo conceptual se sirve de métodos que permiten la autonomía y el uso de la capacidad creadora".

Comprensión

¹⁶La comprensión se da cuando el alumno adquiere dominio forma y material sobre el objeto de aprendizaje. Se pueden diferenciar tres grados o niveles de comprensión:

¹⁶ <http://espanol.geocities.com/criteriosevaluacion/procesocognitivo.htm>, Instituto Técnico Superior "Dámaso Zapata, Capacitación docente, Bucaramanga, Marzo de 2003.

- Comprensión de traducción: cuando el alumno es capaz de pasar a sus propias palabras, o identifica el significado de cada una de las palabras del tema.
- Comprensión de interpretación: cuando puede identificar y explicar el mensaje o significado del objeto de estudio.
- Comprensión de extrapolación: cuando se transfiere o aplica el tema comprendido en otras situaciones.

La secuencia de los grados de comprensión, en relación con el proceso de aprendizaje, es la siguiente:

- Traducción
- Interpretación
- Extrapolación.

La comprensión de traducción nos explica el significado de cada uno de los términos propuestos por una historia

La interpretación nos da a conocer el mensaje tomado y la enseñanza que la herramienta nos desea transmitir.

La Extrapolación nos transmite la idea fundamental y principal que el estudiante plasmo con el juego una vez terminado.

Generalización

¹⁷La generalización consiste en transferir y aplicar en otras situaciones la síntesis elaborada, de manera que sea posible constatar, ampliar y enriquecer el conocimiento adquirido

Implica abstraer lo que es común a varias cosas, extender o ampliar la cobertura significativa de una proposición o verdad.

Si la generalización no respeta los procesos previos de conceptualización y comprensión, corre el peligro de convertirse entonces en juicio 'a priori' o utopía.

1.3.1.6 **Métodos para la resolución de problemas.** Existen métodos por los cuales podemos llegar a la resolución de los problemas como:

Preparación

- Análisis de: datos, restricciones, criterios de solución.
- Resultado: revisión del problema en otros más sencillos, elementales. Ignorar información para simplificar el problema.
- Preparación: breve cuando se tiene suficiente información o habilidad adquirida.

Producción

- Operaciones: Recuperación de la memoria a largo plazo. Exploración de la información ambiental.
- Elaboración de posibles soluciones.

¹⁷ <http://espanol.geocities.com/criteriosevaluacion/procesocognitivo.htm> , Instituto Técnico Superior "Dámaso Zapata, Capacitación docente, Bucaramanga, Marzo de 2003.

Practica de la solución

- Una única intervención
- Seriación de posibles soluciones.

Enjuiciamiento

- Evaluación de resultados.
- Contraste con el criterio inicial.
- Modificaciones.

1.4 IMPLEMENTACIÓN DE LA RESOLUCIÓN DE PROBLEMAS EN LA HERRAMIENTA.

Anteriormente se hablo sobre que es la resolución de problemas, que es un problema, características de la resolución de problemas, las habilidades requeridas para la resolución de problemas y los métodos utilizados para la resolución de problemas. Todos estos temas han dado una idea clara de cómo construir la herramienta para alcanzar el objetivo propuesto al inicio de nuestro proyecto.

La herramienta muestra el desarrollo creativo que deben tener los estudiantes para resolver un problema, buscando la forma de explotar sus habilidades básicas y específicas ante una situación que ha sido planteada y expuesta por un profesor.

1.5 DISEÑO METODOLÓGICO

Basándonos en un enfoque metodológico, la tecnología constituye la base para el desarrollo de este proyecto y el conocimiento de las herramientas que se deben trabajar.

El tipo de trabajo con el cual se identifica el proyecto es de desarrollo tecnológico, en el que se implementaran estrategias y procedimientos ayudando a fomentar la creatividad en los estudiantes.

Para la realización del proyecto, se fijan unas etapas a las cuales se le definen las tareas y actividades que se irán desarrollando.

A continuación se describen cada una de estas etapas:

1 ETAPA: RECOLECCION DE INFORMACIÓN

En esta etapa se investigó toda la información que nos dio las suficientes bases conceptuales para tener una idea sobre el proyecto que se realizó, además nos brindó la fundamentación en las diferentes áreas del conocimiento y la preparación para utilizar las herramientas que nos llevaron al desarrollo de la aplicación.

Los pasos para la realización de la primera etapa:

- *La metodología para el desarrollo de software:* se buscaba aprender lo necesario del proceso unificado del software para tener los conocimientos de un buen modelamiento para así, en el momento de pasar a otra etapa tener la menor cantidad de inconvenientes o incoherencias

- *Las herramientas para la especificación y desarrollo de software:* Se empezó a consultar los requerimientos de hardware que se necesitaban para el desarrollo por medio del director del proyecto, otros profesores, hasta tener bien claro que era lo que se podía utilizar.
- *Las tecnologías utilizadas en el proyecto:* Consultamos las tecnologías con las cuales el proyecto estaba ligado como svg, linux, dreamweaver, java, tomcat, UML (Modelamiento), jsp, por medio de Internet, libros, tutorías, profesores.
- *Actividades lúdicas para el desarrollo de la creatividad a implementar:* Estas actividades nos fueron entregadas una vez se sustentó el anteproyecto y por medio de las tecnologías estudiadas y la especificación del hardware, se empezó a trabajar en el entendimiento de cada una de ellas.

2. ETAPA: ANALISIS DE REQUERIMIENTOS PARA LA IMPLEMENTACION DE LAS HERRAMIENTAS

Esta etapa la comenzamos identificando profundamente cada uno de los requerimientos que necesitábamos tanto en hardware como en software para desarrollar la aplicación.

En la aparte de software:

- *Compatibilidad del software:* Nos percatamos de que toda la tecnología utilizada se pudiera trabajar sin presentar inconvenientes.
- *Aprendizaje de tecnologías:* Hacemos referencia al tiempo que tardaríamos en aprender o manipular cada una de las diferentes tecnologías.

4. ETAPA: IMPLEMENTACION DEL PROTOTIPO

En esta etapa se desarrollo el prototipo de las herramientas que serán ofrecidas a los estudiantes mediante un entorno virtual apoyados en los puntos que se expuestos en las etapas anteriores.

Las fases del desarrollo fueron:

- Se trabajo fuerte en el modelamiento de la base de datos.
- Se empezó a implementar lo relacionado con la programación en web como por ejemplo: JSP, JAVA2 EDITON 1.4, SCRIPTLET, SVG, JDBC, TOMCAT.
- Se utilizaron herramientas alternas como dreamwaver.
- La aplicación fue montada con direcciones privadas.

5. ETAPA: PRUEBAS

Esta etapa fue la mas destacada de nuestro proyecto, ya que cuando se pensaba que la aplicación estaba terminada, nos salían datos erróneos y fue así como se fue convirtiendo en la mas primordial de todas las etapas para el desarrollo del producto.

Para la realización de las mejoras, la aplicación fue probada por el director, los desarrolladores, los evaluadores y terceros, los cuales dieron su opinión acerca de los respectivos cambios que se le deberían hacer y es así que después de largas horas de trabajo se logra un producto con el fin de satisfacer las necesidades de los actores que interactúan con en el sistema.

6. ETAPA: DOCUMENTACIÓN

En esta etapa mostramos la documentación de todas las etapas anteriores y en el que se reúne la información completa del desarrollo del proyecto.

Esta etapa de documentación, fue elaborada continuamente gracias a la ayuda de las personas interesadas en el proyecto como lo son los evaluadores, la directora y los desarrolladores, describiendo que cosas hacen faltan en la construcción del mismo para así llegar a mostrar un documento final que sea fácil de entender para las personas que el día de mañana lo necesiten.

2. TECNOLOGÍAS DE DESARROLLO

2.1 JAVA 2, STANDRAD EDITION V 1.4

Java 2, standard edition v 1.4, es una herramienta para crear e implantar la parte cliente en aplicaciones empresariales. Hoy en día, esta es la versión mas rápida que se encuentra.

Comparada con la anterior, la versión 1.4 muestra un porcentaje más alto, superándola en un 10%.

Para los usuarios finales, la versión 1.4 es más eficiente, pues permite un uso rápido, eficiente y sobre todo sencillo de las aplicaciones que se encuentren en un ambiente web.

2.2 JAVABEANS

Un JavaBean es un componente de software reutilizable que se escribe en el lenguaje de programación de Java. El programador puede escribir JavaBeans mientras que escribe cualquier otra clase de Java. Se puede también tomar componentes, applet, o las clases existentes de Java y darles vuelta en JavaBeans.

2.2.1 **Características de JavaBeans.** Las características son análogas y los acontecimiento son análogos a los comportamientos o métodos definidos para los objetos. La diferencia principal es que las características y los acontecimientos están expuestos a las herramientas del constructor para permitir la manipulación de las diferentes opciones. Los valores asignados a las características se pueden

modificar en el tiempo del diseño. Cuando usted define características, usted las declara típicamente como tipos privados.

2.3 JAVA SERVER PAGES

2.3.1 **Introducción.** ¹⁸Existen numerosas tecnologías para la creación de paginas web, pero en realidad JSP, ha sido realmente acogido por la los desarrolladores o programadores, gracias a su facilidad de compatibilidad con otras herramientas y la formas de adaptarse a diferentes entornos.

Las páginas de JSP abarcan típicamente de:

- Componentes estáticos de HTML/XML.
- Etiquetas especiales de JSP
- Scriptlets llamados del lenguaje de programación de Java.

Por lo tanto, usted puede crear y mantener las páginas de JSP por las herramientas convencionales de HTML / XML.

JSP puede ser utilizado no solamente por programadores, sino también por los diseñadores de la páginas, que saben desempeñar un papel más directo en el ciclo vital del desarrollo.

2.3.2 ventajas jsp

- Con JSP, la lógica para generar el contenido dinámico es guardada a parte de las plantillas estáticas de la presentación, encapsulándolo dentro de los componentes externos de JavaBeans. Éstos componentes son creados y utilizados por la página de JSP usando etiquetas y scriptlets especiales.

¹⁸ <http://developer.java.sun.com/developer/onlineTraining/JSPIntro/contents.html>, Sun Microsystem septiembre de 2003.

Cuando un diseñador de la página realiza cualquier cambio a la plantilla de la presentación, la página de JSP es recopilada y recargada automáticamente en el web Server por el motor de JSP.

- Las páginas de JSP se pueden mover fácilmente a través de plataformas, y a través de los servidores, sin ningunos cambios.
- No hay nada que asigna los datos estáticos de la plantilla por mandato dentro de una página de JSP para estar de cierto formato.
- Usted puede hacer casi cualquier cosa que se puede hacer con servlets usando JSP, pero más fácilmente.
- Otra ventaja de JSP es la separación inherente de la presentación del contenido facilitado por la tecnología.

2.3.3 Arquitectura de jsp

¹⁹El propósito de JSP es proporcionar un método declarativo, de desarrollar servlets.

Típicamente, las páginas de JSP están conformadas por dos fases, una fase es la de traducción y la otra es la del proceso de la petición.

La fase de la petición: Se realiza solamente una vez. Es allí donde se asume que no existe ningún error de sintaxis dentro de la página, el resultado es un archivo de la clase de la puesta en práctica de la página de JSP que pone el interfaz de Servlet en ejecución.

¹⁹ <http://developer.java.sun.com/developer/onlineTraining/JSPIntro/contents.html>, Sun Microsystem, septiembre de 2003.

La fase de la traducción: Es realizada típicamente por el motor de JSP cuando recibe un pedido de la página de JSP.

Muchos detalles de la fase de la traducción, como la localización en donde se almacenan los archivos de la fuente y de la clase son dependientes de la puesta en práctica.

2.3.4 Elementos de jsp

2.3.4.1 **Directivas.**²⁰ Las directivas de JSP son los mensajes para el artefacto de JSP. Ellos no producen ningún rendimiento visible directamente, pero se sabe que hacer con el resto de la página de JSP. Las directivas de JSP siempre son adjuntos dentro de una etiqueta `<% @... % >`.

Las dos directivas primarios son:

- **Page:** el cual es el que le brinda la información a la pagina,
- **Include:** el cual incluye los archivos completos palabra a palabra.

NOTA (JSP 1.1 también proporciona el **taglib**, director que puede usarse por trabajar con las bibliotecas de etiqueta de costumbre aunque esto no se discute aquí.).

2.3.4.2 **Declaraciones.** Las declaraciones de JSP le permitieron definir las variables para ahorrar la información o definir métodos que pueda necesitar la pagina JSP.

Las declaraciones se encuentran dentro de la etiqueta `<%! ... % >`. Y en estas declaraciones también se encuentran los métodos.

²⁰ <http://developer.java.sun.com/developer/onlineTraining/JSPIntro/contents.html>, Sun Microsystem, septiembre de 2003.

2.3.4.3 **Expresiones.** Con las expresiones en JSP, se convierten los resultados de evaluar una expresión incluida dentro de la página.

²¹Típicamente se usan las expresiones para desplegar valores simples de variables o valores del retorno.

Las expresiones de JSP empiezan dentro de la etiqueta `<% =... % >` y no incluye los puntos y comas. Cualquier cosa que se encuentre dentro de la etiqueta, será evaluada, convertida a cadena y posteriormente mostrada en la pantalla. Esta conversión es de tipo inicial a String de manera automática.

2.3.4.4 **Scriptlets.** Este código de Java se corre cuando la demanda se repara por la página de JSP. Usted puede tener cualquier código de Java válido dentro de un scriptlet, y no se limita a una línea de código de la fuente.

Dentro de los script tenemos algunos objetos implícitos entre estos están:

- **Request:** la petición del cliente
- **Page:** forma de la pagina para su referencia.
- **Session:** es asociado a la petición del cliente
- **Application:** es lo que el servlet devuelve.
- **Page context:** encapsula los rasgos aplicación-dependientes.
- **Config:** representa el ServletConfig para el JSP.

2.3.4.5 **Comentarios.** Aunque usted siempre puede incluir los comentarios de HTML en las páginas de JSP, los usuarios pueden ver éstos si tienen la fuente de la página.

²¹ <http://developer.java.sun.com/developer/onlineTraining/JSPIntro/contents.html>, Sun Microsystem, septiembre de 2003.

Un rasgo más útil de comentarios de JSP es que ellos pueden bloquear script o etiquetas de la recopilación..

2.4 JAVA DATA BASE CONNECTION

²²JDBC es una API de Java para ejecutar declaraciones del SQL. Consiste en un sistema de clases y de interfaces escritos en el lenguaje de programación de Java.

Usando JDBC, es fácil enviar declaraciones del SQL a virtualmente cualquier base de datos emparentada. Es decir con el JDBC API, no es necesario escribir un programa diferente para tener acceso a cada base de datos. Uno puede escribir un solo programa usando el JDBC API, y el programa podrá enviar declaraciones del SQL a la base de datos apropiada. Y, con un uso escrito en el lenguaje de programación de Java, uno también no tiene que preocuparse de diversos usos de la escritura para funcionar en diversas plataformas. La combinación de Java y de JDBC deja un programador escribirlo una vez y funcionarlo dondequiera.

Java, siendo robusta, segura, fácil de utilizar, fácil entender, es una base excelente de la lengua para los usos de la base de datos.

JDBC amplía qué se puede hacer en Java. Por ejemplo, con Java y el JDBC API, es posible publicar una pagina web que contiene un applet que utilice la información obtenida de una base de datos alejada.

Que hace JDBC

JDBC permite hacer tres cosas:

²² <http://java.sun.com/products/jdbc/index.jsp>, sun microsystem, copyright 1999-2000, septiembre de 2003.

1. Establecer una conexión con una base de datos .
2. Enviar las declaraciones del SQL .
3. procesar los resultados.

²³JDBC invoca comandos del SQL directamente, pero fue diseñado también para ser una base sobre la cual construir interfaces y las herramientas de alto nivel.

Mientras que el interés en JDBC ha crecido, más reveladores han estado trabajando en las herramientas JDBC para hacer los programas complejos un poco mas fáciles para los programadores y los usuarios externos o finales.

JDBC abarca potencialmente un conjunto de sistemas de base de datos. Java con la inclusión de JDBC se puede decir que es un entorno perfectamente valido para el acceso y la explotación de un sistema de base de datos. Para que una aplicación escrita en java se pueda comunicar con una base de datos estipulada a través de JDBC debe tener un controlador JDBC el cual es el que convierte las ordenes en comandos inteligibles para esa base de datos exacta.

Existen varios tipos de controladores JDBC , pero el mas indicado es el puente entre el JDBC-ODBC, el cual sirve para que las aplicaciones java se comuniquen con sistemas de bases de datos que funciones con el estándar ODBC, Además el JDBC-ODBC garantiza el acceso a cualquier base de datos comercial.

En resumen, el JDBC API es un interfaz natural de Java a las abstracciones básicas y a los conceptos del SQL.

JDBC conserva las características fundamentales de ODBC; en hecho, ambos interfaces se basan en el X / Open SQL CLI (interfaz llano de la llamada). La diferencia grande es que las estructuras de JDBC han encendido y reforzado el estilo y las virtudes de Java, y, por supuesto, son fáciles de utilizar.

²³ <http://java.sun.com/products/jdbc/index.jsp>, sun microsystem, copyright 1999-2000

2.5 SVG

²⁴No cabe duda de que la inclusión de imágenes en las páginas web, trajo un gran avance en el posterior desarrollo y popularización de la WWW. Efectivamente, con la posibilidad de incluir ficheros gráficos, las páginas web pasaron a ser solo documentos textuales, normalmente destinados a divulgaciones científicas, a verdaderos documentos multimedia capaces de presentar al usuario final un gran mundo de color e imágenes.

Dos han sido los formatos de fichero gráfico estandarizados en la web, el formato GIF y el JPG. Cada uno presenta sus ventajas e inconvenientes, y han servido durante mucho tiempo (en la escala relativa de Internet y la WWW) a los diseñadores web para construir páginas agradables y operativas.

Varios han sido los intentos de encontrar y estandarizar un nuevo formato que supliera estas limitaciones. Cabe destacar el formato BMP, de Microsoft, que aunque presenta una buena calidad en un relativo poco peso, no se ha convertido en estándar, quedando limitado su uso correcto en las diferentes versiones del navegador propietario de esta empresa, Internet Explorer. También hay que mencionar el formato PNG, que intenta conjugar las mejores características de los gráficos GIF y los JPG en un único formato, pero que tampoco ha llegado a alcanzar la popularidad que se esperaba.

Dos son las limitaciones básicas de todos estos formatos, desde el punto de vista del diseñador-programador web actual:

Son formatos de mapa de bits, por lo que no pueden ser escalados sin una pérdida de calidad considerable.

Créan gráficos estáticos, basados en el almacenamiento interno de una tabla de valores de coloración para los diferentes pixels que forman la imagen final, valores que son fijos e inmutables una vez configurado el fichero.

²⁴ http://www.htmlweb.net/disenos/svg/svg_1.html, Luciano Moreno, del departamento de diseño web de BJS Software, octubre de 2003.

- ²⁵Con este planteamiento, es lógico pensar que las investigaciones de empresas y programadores han ido destinadas a la creación de un formato gráfico reuniera una serie de características básicas:
- Escalabilidad, con lo que la calidad de la imagen final no estaría condicionada por el grado de aumento o reducción que se le diera. Es decir, que guardara gráficos vectoriales, no de mapa de bits.
- Dinamicidad física, con lo que se podrían crear pautas de comportamiento de los gráficos en función del tiempo, permitiendo con ello la construcción de verdaderas animaciones gráficas.
- Dinamicidad lógica, con lo que los gráficos dejaría de ser una entidad con propiedades fijas e inmutables una vez guardado el fichero. Con ello se consigue crear gráficos inteligentes, capaces de adaptarse a cambios en los valores que los definen.

En este sentido empezaron a trabajar las empresas interesadas en el tema, surgiendo diferentes propuestas y alternativas, algunas de las cuales se asumieron y otras no.

Uno de los formatos gráficos surgidos de esta ola de investigación, y que ha llegado a convertirse en un estándar en la web, es el formato Shockware Flash, de la casa Macromedia. Es un formato capaz de crear gráficos vectoriales escalables de alta calidad, con animaciones configurables en el tiempo a las que se pueden aplicar múltiples de efectos y filtros. Además, los ficheros finales resultan tener un relativo poco peso, en comparación con la calidad obtenida.

Como inconveniente, el formato Flash ha sido creado con un planteamiento de creación general, capaz de crear presentaciones en formato propio, en CD-ROM y, adicionalmente, en cualquier aplicación que admita la ejecución del plugin adecuado. No es pues un formato pensado y creado por y para la web.

²⁵ http://www.htmlweb.net/disenosvg/svg_2.html, Luciano Moreno, del departamento de diseño web de BJS Software, octubre de 2003

2.6 SERVIDOR WEB JAKARTA-TOMCAT

²⁶Tomcat se desarrolla en un ambiente abierto y participante y se lanza bajo la licencia del software de Apache y puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (actualmente se soporan los servidores Apache, IIS y Netscape).

El servidor jakarta-Tomcat es la implementación de referencia para las *Java Server Pages* (JSP) y las especificaciones *Java Servlet*. Esto significa que es el servidor Java disponible que más se ajusta a los estándares.

Muchos usuarios de entre una gran área de aplicaciones descubrieron el servidor Tomcat y decidieron que se ajustaba a sus necesidades. Estos usuarios eran proveedores de *hosting* web, desarrolladores de aplicaciones *e-commerce*, desarrolladores web de compañías, estudiantes e investigadores universitarios, así como gente con intereses propios y otros usuarios potenciales.

2.6.1 Versiones del servidor Tomcat

- ²⁷Tomcat 5X: Es el Nuevo lanzamiento de tomcat
- Tomcat 4X: Pone un envase nuevo del servlet en ejecución que se base en totalmente nueva arquitectura. El 4.x lanza el instrumento el Servlet 2.3 y las especificaciones de JSP 1.2

²⁶ http://www.programacion.com/java/tutorial/tomcatintro/1/#1_intro, Java en castellano. Septiembre de 2003.

²⁷ <http://jakarta.apache.org/tomcat/>, the apache jakarta proyect ,Copyright © 1999-2004, Apache Software Foundation, noviembre de 2003

- Tomcat 3X: Todos los lanzamientos de Tomcat 3.x remontan su herencia de nuevo a las puestas en práctica originales de Servlet y de JSP que el sol donó a la fundación del software de Apache. 3.x las versiones todo ponen el Servlet 2.2 y especificaciones de JSP en ejecución 1.1.

Estas son algunas de las versiones y si no son todas, si son las mas conocidas y utilizada por las personas que trabajan con este servidor.

2.7 MYSQL

2.7.1 **Introducción.**²⁸MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

²⁸ http://www.aditel.org/~dpecos/docs/mysql_postgres/x57.html#AEN71,
noviembre de 2003.

Gustavo Ramírez,

2.7.2 **Historia de mysql.** ²⁹MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que MSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a MSQL.

No se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

2.7.3 **Características de mysql.** Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
2. Soporta gran cantidad de tipos de datos para las columnas.
3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
4. Gran portabilidad entre sistemas.
5. Soporta hasta 32 índices por tabla.
6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.

²⁹ http://www.aditel.org/~dpecos/docs/mysql_postgres/x57.html#AEN71, Gustavo Ramírez, noviembre de 2003.

2.8 LINUX

2.8.1 **Introducción.** ³⁰Linux es un sistema operativo multitarea y multiusuario. Esto es, se trata de un programa que interacciona entre el hardware de la máquina y nosotros, permitiéndonos, gracias a los programas, hacer tareas que de otro modo sería muy complicado (acceso a disco, escritura de ficheros, etc.). Decimos *multitarea* porque es capaz de tener varios programas ejecutándose al mismo tiempo, y decimos *multiusuario*, porque es capaz de admitir a varios usuarios distintos trabajando sobre la misma máquina al mismo tiempo.

Dado que se trata de un sistema multiusuario, cuando alguien quiere acceder a la máquina, debe *identificarse*, para poder ser reconocido por esta y permitirle la entrada al sistema si se trata de un usuario autorizado. Este proceso es el que se conoce como *login*. Durante este proceso, la máquina nos preguntará nuestro nombre de usuario (*login*) y nuestra contraseña (*password*). Es el administrador del sistema (*root*) quien debe crearnos una cuenta en la máquina y quien nos dará los datos. Una vez entremos a nuestra cuenta, nos encontraremos dentro de nuestro directorio HOME (el directorio que nos asigna el administrador para que guardemos nuestro trabajo), y se nos aparecerá el *prompt* del sistema: esto es un símbolo que nos indica que la máquina está lista para recibir comandos.

2.8.2 **Breve historia de linux.** El antecesor de Linux fue el sistema operativo UNICS, concebido en 1969 en los Laboratorios de la Bell Computer. En 1971, Kenneth Thompson y Dennis ³¹Ritchie presentaron la primera versión de Unix, después de 2 años de investigaciones y desarrollo.

³⁰<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?cap=1> El rincón del programador, Copyright © 2000-2004, Lola Cárdenas Luque

³¹ <http://www.computerhope.com/history/unix.htm>, linux system, Noviembre de 2003.

Una serie de situaciones acontecieron en el devenir de los años, incluyendo el licenciamiento que Microsoft obtuvo de AT&T a fines de los 70, luego Santa Cruz Operation Inc. (SCO) liberó su versión denominada Xenix, en 1983, y no fue hasta 1985 en que la declaración sobre el GNU fuese publicada en la revista Dr. Dobb, cuando el proyecto del hoy denominado "código abierto" empezó a ser puesto en práctica:

Richard Stallman inicialmente fue el creador del **Free Software Foundation** (Fundación de Software Gratuito) y mientras trabajó en el MIT (Massachusetts Institute of Technology), en la década de los 80, lanzó al mundo el proyecto GNU (GNU's not Unix).

Linux fue desarrollado en 1991 por el joven finlandés Linus Torvalds, de la universidad de Helsinki, después de un año de investigación, "hacking" y desarrollo hasta que empezó a difundirse en enero del 1992, causando toda una revolución tecnológica en el mundo de las tecnologías de información.

El kernel de linux es uno de los mas robustos y versátiles de todos los sistemas operativos conocidos e incluso ha sido mejorado en posteriores versiones todas vez que el sistema original recibe constantemente importantes aportes de miembros de las comunidades de Linux. Sin embargo, son las aplicaciones de terceros las que hasta la fecha han demostrado tener varias vulnerabilidades.

En la medida que este sistema operativo empezó a popularizarse, los creadores de virus le pusieron atención, tal como lo manifestáramos en 1986, aduciendo su poca popularidad por aquellos años. Sin embargo los tiempos cambiaron y surgieron los virus para Linux.

Existen en la actualidad muchísima versiones libres y comerciales de Linux, incluso en otros idiomas diferentes al inglés o español. Obviamente que las mas destacables han sido las que se comercializan, ya que con los ingresos obtenidos pueden sufragar la investigación, desarrollo y mejora de sus productos:

- Linux Red Hat
- Mandrake Linux
- Linux Caldera
- Linux Conectiva
- Turbo Linux

2.8.3 **Comandos para creación, gestión de archivos y directorios.** ³²Los siguientes comandos: `ls`, `cd`, `cp`, `mkdir`, `mv`, `pwd`, `rm`, `rmdir`, `chmod`, `touch`, están orientados a la gestión típica de ficheros y directorios: listar el contenido de un directorio, cambiar de directorio, borrar archivos, etc...

ls

Lista el contenido de directorios del sistema. Es lo que el usuario de MS-DOS conoce como DIR. Hay muchas opciones para este comando (`-a`, `-l`, `-d`, `-r`,...), que a su vez se pueden combinar de muchas formas. Dado que esto puede variar según el sistema Unix en el que se esté, es recomendado consultar su página de manual (*man ls*). Sin embargo, de todas éstas, las que podríamos considerar más comunes son:

`-l` (*long*): Formato de salida largo, con más información que utilizando un listado normal.

`-a` (*all*): Se muestran también archivos y directorios ocultos.

`-R` (*recursive*): Lista recursivamente los subdirectorios.

³²<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?cap=2a>
El rincón del programador, Copyright © 2000-2004, Lola Cárdenas Luque

cd

³³Con este comando podremos cambiar de directorio de trabajo. La sintaxis básica es la siguiente:

```
cd [nombre_directorio]
```

Si usamos el *shell* bash, (que es el que se instala por defecto en los sistemas Linux), simplemente tecleando cd, volvemos a nuestro directorio HOME. Si le pasamos como parámetro una ruta (que puede ser absoluta, es decir, el nombre completo desde el directorio raíz, o relativa a dónde estamos) de un directorio, cambiaremos a ese directorio, siempre y cuando tengamos permiso para entrar. Es muy posible que en nuestro Linux, al hacer cd para entrar en algún directorio, no lo veamos reflejado en el *prompt* (como sucede en MS-DOS). Para poder ver en qué directorio estamos, podemos usar el comando pwd (**print working directory**), aunque de este comando hablamos unos párrafos más adelante.

Algunos ejemplos de cd:

```
MathLand:~# cd Programas/C
```

```
MathLand:Programas/C# cd
```

```
MathLand:~#
```

```
MathLand:~# cd /usr/bin
```

```
MathLand:/usr/bin#
```

cp

Con el comando cp copiamos un archivo (origen), en otro lugar del disco (puede ser un archivo o un directorio), indicado en destino.

Su sintaxis es *cp <origen><destino>*. Si el destino es un directorio, los archivos de origen serán copiados dentro de él. Hay que decir que los *shells* de Unix aceptan *wildcards* (comodines, los trataremos en otro capítulo), que son los caracteres especiales * y ?. Así, una orden como

³³<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?cap=2a>, El rincón del programador, Copyright © 2000-2004, Lola Cárdenas Luque

³⁴MathLand:~# cp *.html Directorio_HTML

copiará todos los archivos que finalicen en .html en el directorio Directorio_HTML, si éste existe. Debemos recordar que en Unix el campo "." no separa el nombre de la extensión de un fichero, como en MS-DOS, sino que es simplemente un caracter más. No debemos olvidarlo, sobre todo con comandos como rm, si no queremos llevarnos sorpresas desagradables. Para copiar de forma recursiva (es decir, también subdirectorios) podemos usar la opción *-r*

mkdir

Tal y como su nombre parece querer decir, crea un directorio. La sintaxis será simplemente *mkdir <nombre_directorio>*. Para crear un directorio tenemos que tener en cuenta los permisos del directorio en que nos encontremos trabajando pues, si no tenemos permiso de escritura, no podremos crear el directorio.

mv

Con este comando podemos renombrar un archivo o directorio, o mover un archivo de un directorio a otro. Dependiendo del uso que hagamos, su sintaxis variará. Por ejemplo:

mv <archivo/s> <directorio> moverá los archivos especificados a un directorio, mientras que con *mv <archivo1><archivo2>* renombrará el primer fichero, asignándole el nombre indicado en *<archivo2>*. Veamos unos ejemplos:

```
MathLand:~# mv Hola_Mundo.c Copia_Hello.c
```

(Renombra el archivo Hola_Mundo.c como Copia_Hello.c)

```
MathLand:~# mv *.c Un_Directorio
```

³⁴<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?cap=2a>, El rincón del programador, Copyright © 2000-2004, Lola Cárdenas Luque

pwd

³⁵Imprime en pantalla la ruta completa del directorio de trabajo donde nos encontramos actualmente. No tiene opciones, y es un comando útil para saber en todo momento en qué punto del sistema de archivos de Unix nos encontramos.

rm

Elimina archivos o directorios. Sus tres opciones son *-r* (borrado recursivo, es decir, de subdirectorios), *-f* (no hace preguntas acerca de los modos de los archivos), y *-i* (interactivo, solicita confirmación antes de borrar cada archivo). Su sintaxis es muy sencilla: *rm [-r] [-f] [-i] <archivo>*. Hay que tener mucho cuidado con este comando cuando se usen comodines, sobre todo si no lo ejecutamos en modo interactivo.

rmdir

Borra directorios *únicamente* si están vacíos. Su sintaxis básica será *rmdir <directorio>*. Si queremos borrar directorios que no estén vacíos, hemos de utilizar el comando *rm -r <directorio>*.

chmod

Con este comando, cambiamos los permisos de acceso del archivo o del directorio que le especifiquemos como argumento. Podemos ejecutar de dos formas básicas este comando: la primera es *chmod <modo> <archivo>*, siendo *modo* un valor numérico de tres cifras octales que describe los permisos para el archivo. Cada una de estas cifras codifica los permisos para el dueño del archivo, para los usuarios que pertenecen al mismo grupo que el dueño, y para el resto de los usuarios, en este orden.

³⁵<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?cap=2a> El rincón del programador, Copyright © 2000-2004, Lola Cárdenas Luque

Veamos un ejemplo: supongamos que queremos cambiar los permisos de un archivo llamado Algo.txt (por poner algún nombre), de manera que nosotros podamos leerlo y modificarlo, pero no ejecutarlo, los usuarios de nuestro grupo³⁶ puedan leerlo y el resto de usuarios no pueda ni leer el fichero. La terna de permisos que describe esta situación es:

```
rw- r-- ---
```

Si escribimos un 1 en el lugar del permiso activado y un 0 en el del permiso desactivado, tenemos:

```
rw- r-- ---  
110 100 000
```

Y si ahora codificamos los triplete en binario a octal, tenemos que el permiso será 640, por tanto, tendremos que escribir el comando
chmod 640 Algo.txt

La segunda forma es algo más complicada:

```
chmod <who> +/- <permiso> <archivo>
```

Indicaremos, en el parámetro *who*, la identidad del usuario/s cuyos permisos queremos modificar (*u-user*, *g-group*, *o-others*); a continuación irá un + o un -, dependiendo de si reseteamos el bit correspondiente o lo activamos, y en permiso debemos colocar el permiso a modificar (*r-read*, *w-write*, *x-exec*).

Veamos unos ejemplos de ambos tipos de cambio de permisos:

```
MathLand:~# chmod 123 Ejercicio2.txt  
MathLand:~# chmod 765 Practica.tex  
MathLand:~# chmod g+r Un_programa.o  
MathLand:~# chmod o+rx Otro_programa.p
```

³⁶<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?cap=2a>, El rincón del programador, Copyright © 2000-2004, Lola Cárdenas Luque.

touch

Actualiza la fecha de modificación de un archivo, o crea un archivo vacío si el fichero pasado como parámetro no existe. Con la opción `-c` no crea este archivo vacío. Su sintaxis es `touch [-c] <archivo>`.

3. DESARROLLO DE LA APLICACIÓN

3.1 INTRODUCCIÓN A LA APLICACIÓN

Las herramientas desarrolladas permiten a los estudiantes fomentar el desarrollo de la creatividad, apoyada en la metodología de resolución de problemas.

Los juegos realizados dan una síntesis sobre la capacidad de entendimiento y resolución que posee el estudiante frente a un posible problema presentado, es por eso que se desarrollaron juegos con diferentes aplicaciones en el campo de la programación.

La estructura que presentan los juegos es la siguiente:

Un usuario administrador es el encargado de incluir y administrar todo lo referente a los juegos, al igual permite crear los perfiles que van a tener los usuarios. Una vez creado el usuario (**profesor**) asigna una contraseña y un login a cada estudiante que se integre al curso, cada estudiante hará sus jugadas hasta terminar el juego y para así el profesor dar un juicio valorativo referente al objetivo del juego, el cual es el desarrollo de la creatividad.

Se realizaron diagramas de casos de uso y diagrama de secuencias para el modelamiento, los cuales fueron de gran ayuda para la aplicación.

3.2 DESCRIPCIÓN DE JUEGOS

3.2.1 **Historias.** Quien quiere jugar se llama usuario, cuando ingresa al juego deberá digitar su password. El sistema (Figuras 2 y 3) le brindará una situación específica, cuya característica principal es que será una historia poco común, producto de la imaginación de alguien. El sistema además le sugiere una ayuda general para navegar en el juego. A partir de la herramienta el sistema le brindará elementos que serán el punto de partida de su camino, por ejemplo: un nombre de historia, una frase inicial y una descripción. El usuario hará la construcción del camino hacia la situación final con la colaboración de una segunda persona, que será un compañero de curso escogido al azar por el sistema, quien recibirá el producto inicial del primer usuario y la ayudará en la construcción del camino. Los dos usuarios no conocerán aspectos de su compañero, a excepción del *nickname* que no debería dar pista de los rasgos personales. Cada usuario recibirá el producto con las adiciones que haga su compañero de juego, por lo que no podrá tener acceso al producto hasta tanto su compañero no lo haya modificado. El juego terminará cuando cada usuario haya realizado 5 modificaciones al producto final. Cuando esto suceda cada uno de ellos recibirá en el correo un juicio valorativo por parte del profesor.

Objetivos del Juego

- El juego tiene como objetivo que los estudiantes que intervienen en él lleven a cabo procesos creativos. El hacer que el estudiante lleve a cabo las actividades en momentos diferentes propicia que realice una creación paso a paso, dándole tiempo de reflexionar alrededor de lo que ha estado diseñando. Dado que el juego estará disponible en un entorno, se aprovechan las herramientas de las que este presenta y se aprovechan

algunas de las características principales que surgen alrededor del uso de las tecnologías de información y comunicación.

- Buscar que el estudiante por medio de la herramienta construya frases o historias coherentes con base en la frase dada por el profesor.
- El estudiante desarrolle su habilidad cognoscitiva a partir de la conceptualización observando y construyendo frases.
- Poner en practica los métodos de resolución de problemas, para este caso los mas oportunos son enjuiciamiento para realizar las modificaciones y producción para dar la solución mas apropiada

3.2.2 Combinaciones. Muchos inventos han surgido de la combinación de factores que nadie antes había imaginado. Este ejercicio puede ser bastante divertido si se relacionan al azar sustantivo y el estudiante genera un nuevo producto a partir de tal combinación. El estudiante deberá realizar sus jugadas por medio de una frase en la cual tendrá que combinar los sustantivos que el sistema le da para crear su frase. El juego terminara cuando el estudiante haya realizado tres modificaciones a los sustantivos expuestos inicialmente

Objetivos del juego

- Nuevamente el objetivo es jugar con la imaginación del estudiante, de manera que pueda asumir como reales elementos que nunca han existido.
- Fortalecer los conocimientos por medio de sustantivos combinándolos para la creación de la historia.
- El estudiante desarrolle su habilidad cognoscitiva a partir de la comprensión y conceptualización adquiriendo material de aprendizaje.

- Tratar de utilizar el método de practica de solución en donde el usuario a través de una intervención escribe posibles soluciones

3.2.3 Editor Gráfico. Cuando el usuario ingresa al juego deberá digitar su login y password. El sistema le brindará un editor, cuya característica principal será la creación de una imagen, producto de su imaginación o la imaginación de alguien. El sistema. A partir de la herramienta el sistema le brindará elementos que serán el punto de partida de su camino, por ejemplo: path abierto, circulo, cuadrado, línea y path cerrado. El usuario hará la construcción de su imagen hacia la situación final con la colaboración de una segunda persona, que será un compañero de curso escogido al azar por el sistema, quien recibirá el producto inicial del primer usuario y la ayudará en la construcción de la imagen. Los dos usuarios no conocerán aspectos de su compañero, a excepción del *nickname*. Cada usuario recibirá el producto con las adiciones que haga su compañero de juego, por lo que no podrá tener acceso al producto hasta tanto su compañero no lo haya modificado. El juego terminará cuando cada usuario haya realizado 5 modificaciones al producto final. Cuando esto suceda cada uno de ellos recibirá en el correo un juicio valorativo por parte del profesor.

Objetivos del Juego

- El juego tiene como objetivo que los estudiantes que intervienen en él lleven a cabo procesos creativos. El hacer que el estudiante lleve a cabo las actividades en momentos diferentes propicia que realice una creación paso a paso, dándole tiempo de reflexionar alrededor de lo que ha estado diseñando. Dado que el juego estará disponible en un entorno, se aprovechan las herramientas de las que este presenta y se aprovechan algunas de las características principales que surgen alrededor del uso de las tecnologías de información y comunicación
- Busca que el estudiante tenga un mejor razonamiento ante una situación expuesta por otro usuario.

- El estudiante desarrolle las habilidades cognitivas por medio de la identificación y clasificación de componentes de acuerdo a su función y cognoscitivas por medio de la generalización para ampliar y enriquecer el juego
- Se pondrá en practica el método de enjuiciamiento por medio de modificaciones y criterios de otros usuarios para obtener la creación de un dibujo utilizando las cinco herramientas que le brinda el editor

3.2 APORTES DE LA HERRAMIENTA AL ESTUDIANTE.

Esta aplicación le brinda al estudiante la capacidad de fomentar el desarrollo de la creatividad a través de los juegos que hemos implementado en la herramienta. Los juegos son expuestos por el profesor y son ellos los encargados de sacarle el mejor provecho a la herramienta y explotar por medio de las jugadas creadas la capacidad del estudiante, además también es el encargado por medio de sus ideas que el estudiante se sienta a gusto y manipule la herramienta con mas frecuencia.

3.3 APORTES DE LA HERRAMIENTA AL PROFESOR

El objetivo de la herramienta desarrollada es fomentar la creatividad en los estudiantes, pero para esto, el profesor debe tener un plan de trabajo expuesto por medio de imágenes, freces y palabras las cuales plasmará en la aplicación de tal forma que pueda obtener los resultados que necesita para cumplir su objetivo.

3.4 APORTE DE LA HERRAMIENTA AL INVESTIGADOR

La aplicación desarrollada brinda gran variedad de actividades las cuales se desarrollaron bajo lenguajes de programación siendo compatibles con cualquier sistema de información y buscando ser montadas en un entorno virtual, quedando un prototipo sujeto a cualquier cambio para trabajar las capacidades síntesis, análisis de información y transferencia de conocimientos.

3.6 NAVEGACIÓN DE LA HERRAMIENTA POR EL ALUMNO

El alumno podrá navegar en la herramienta hasta ciertos sitios empezando primero que todo por su registro en el sistema, después comenzara un juego propuesto por el administrador o profesor como lo indica la figura 2

Ver Figura 2 Mapa de navegación de usuario

3.7 NAVEGACIÓN DE LA HERRAMIENTA POR EL ADMINISTRADOR

El administrador tendrá que hacer su registro en el sistema a través de un login y un password entrando a un menú administrador en el cual creara los juegos de historias, editor gráfico y combinaciones para que sus alumnos puedan empezar sus jugadas sin restricciones.

Ver Figura 3 Mapa de navegación de administrador

Figura 2 Mapa de navegación de usuario

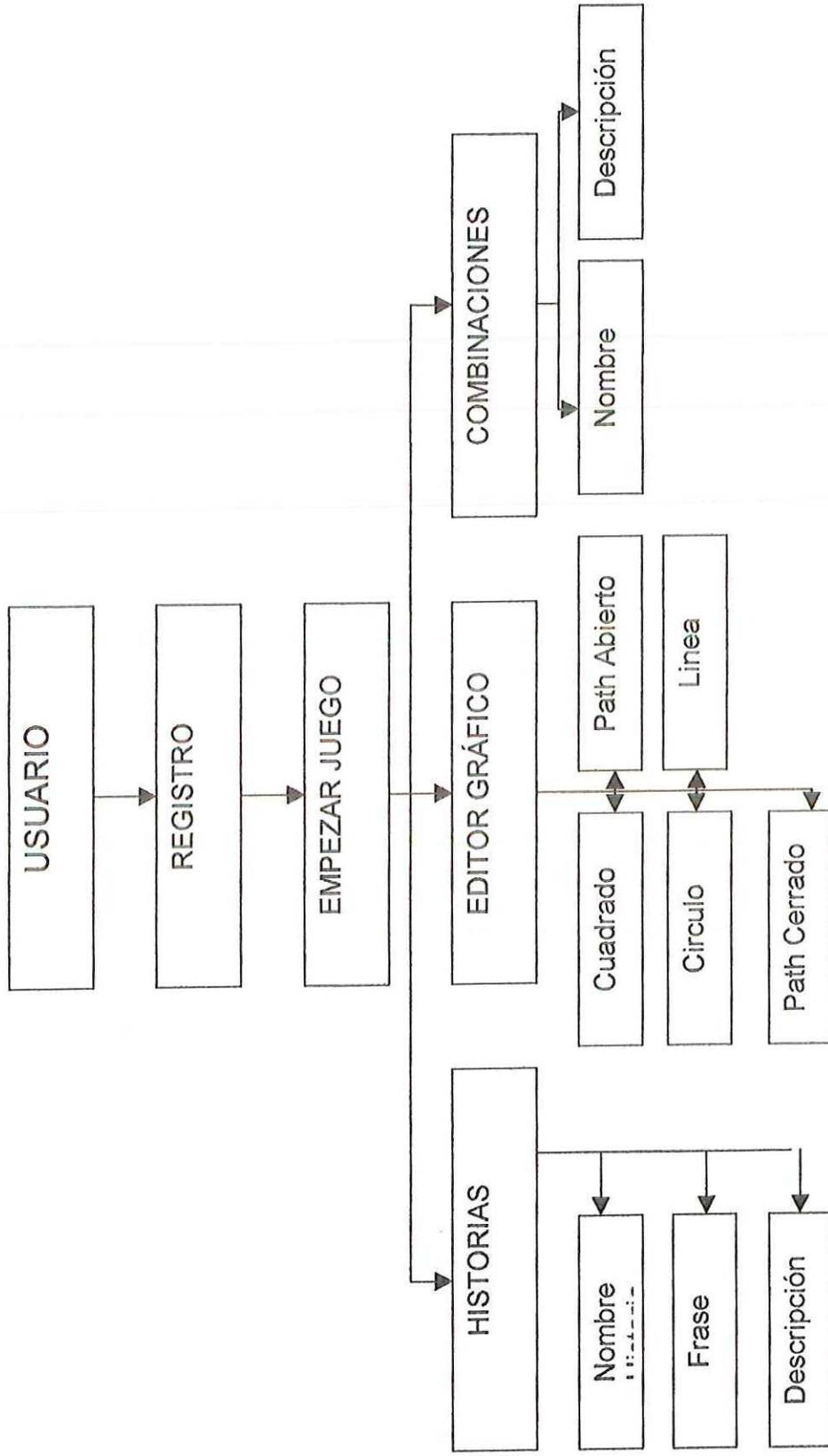
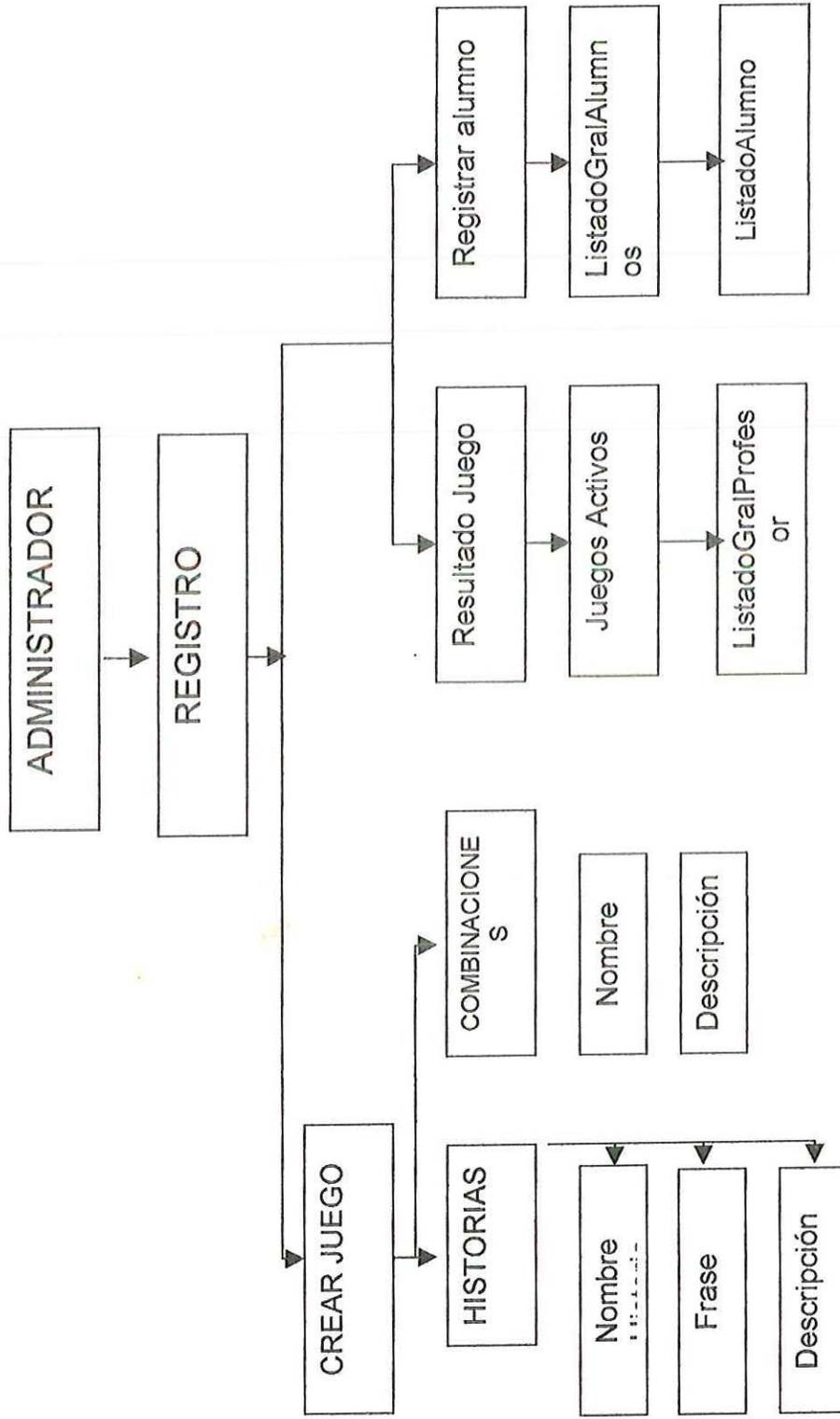


figura 3 mapa de navegación por administrador



3.8 MODELAMIENTO Y DISEÑO DE LA APLICACIÓN

3.8.1 Caso de uso administrador

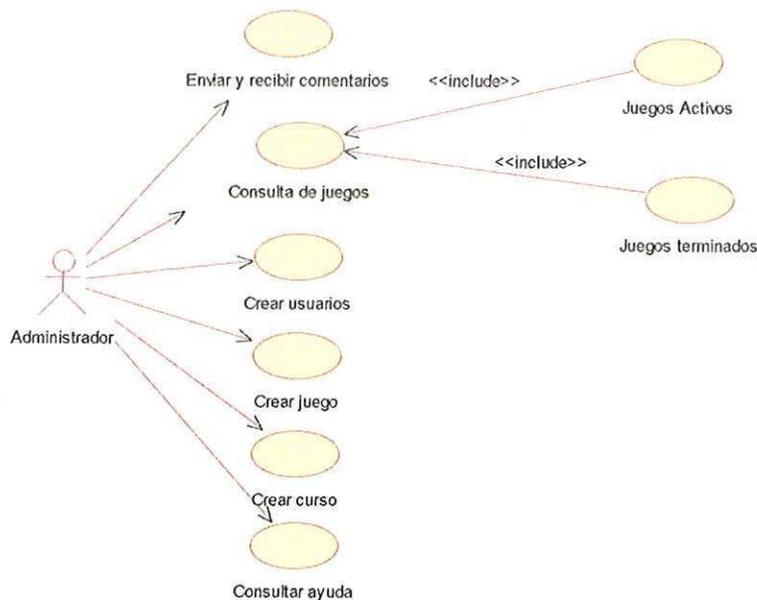
Requerimientos

En la *figura 4*, muestra el diagrama de caso de uso del administrador. Este caso de uso admite ver el perfil del administrador el cual le permite al administrador la creación de juegos, usuarios, cursos, las consultas de juegos al igual que enviar y recibir comentarios.

Análisis

Este caso de uso comienza una vez el administrador ingresa al sistema y comienza a interactuar con cada una de las opciones que el sistema le brinda, como se muestra en la *figura 4*.

Figura 3. Diagrama de caso de uso: Administrador



3.8.2 Caso de uso usuario

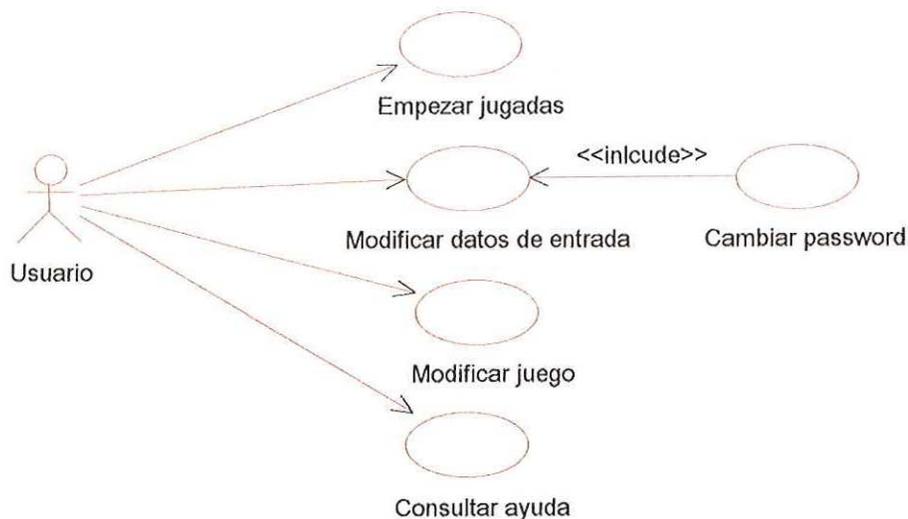
Requerimientos

En la figura 5, muestra el caso de uso de usuario del sistema, este caso de uso permite ver el perfil del estudiante en donde se encuentran los juegos creados por el administrador, los juegos que tiene activos un jugador y consultar las jugadas.

Análisis

Este caso de uso comienza cuando el estudiante por medio de un login y un password que es dado por el administrador, ingresa a interactuar con el sistema haciendo las jugadas respectivos de acuerdo con el juego que el administrador creo.

Figura 5, caso de uso usuario



3.8.3 Creación de Administrador

Requerimientos

En la figura 6, se muestra el diagrama de casos de uso crear administrador. Este caso de uso le permite al nuevo administrador poder tener acceso al sistema y realizar la creación de su curso.

Análisis

En la figura 7, se muestra el diagrama de secuencia para la creación del administrador de acuerdo a las siguientes pasos.

El usuario administrador por medio de un enlace encontrara una nueva ventana que pedirá sus datos para registrarse en el sistema.

A continuación el usuario deberá llenar los campos que el sistema le presenta los cuales son obligatorios para poder ingresar a este.

Para que la creación del administrador se haga efectivamente deberá oprimir el botón de registrar administrador.

Figura 6. Diagrama de casos de uso: Creación de administrador

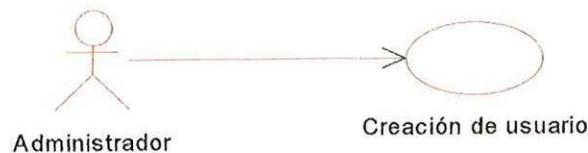
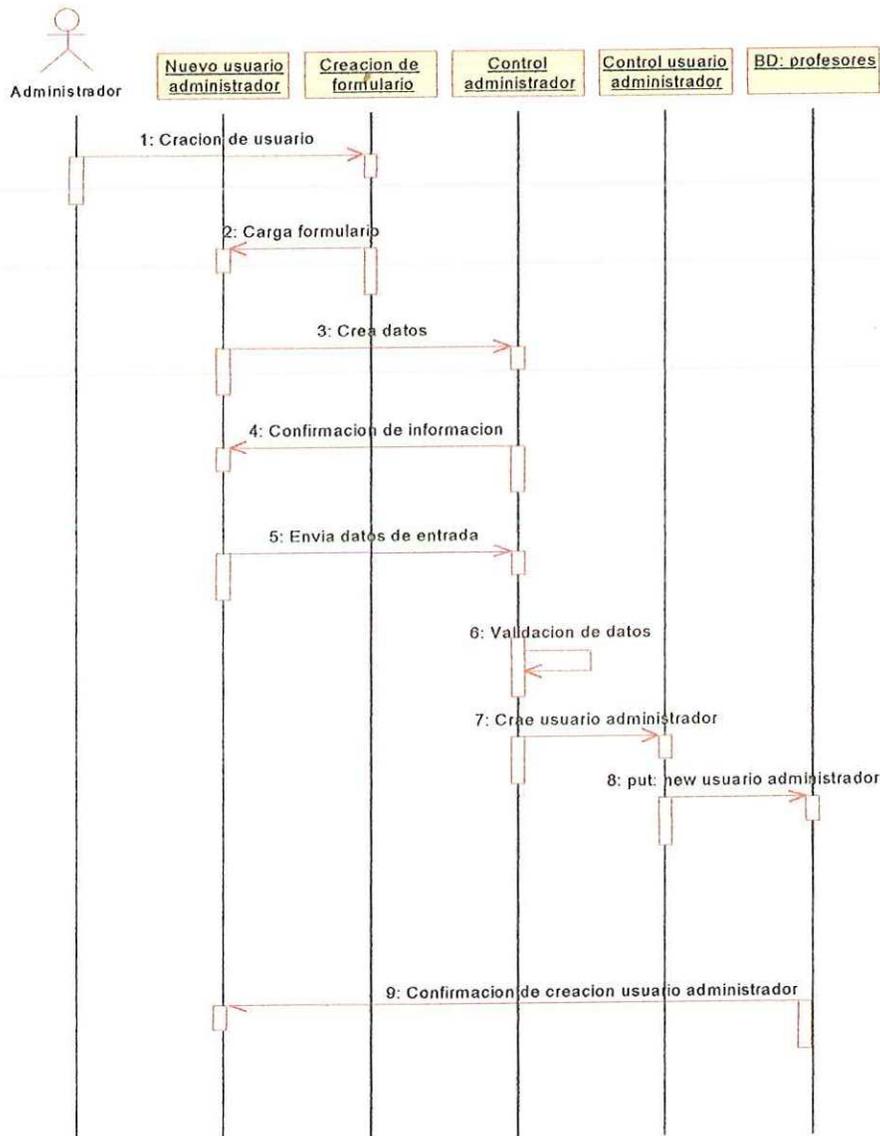


Figura 7. Diagrama de secuencia: Creación de administrador



3.8.4 Consulta de juegos activos

Requerimientos. En la figura 8, muestra el diagrama de casos de uso de consultar juegos activos, el cual trata de brindar una ayuda al profesor y así tener una guía o idea de quienes son los usuarios que se encuentran en los diferentes juegos que él como administrador les ofrece.

Análisis

En la figura 9, se muestra a través del diagrama de secuencias, la forma como se realiza el caso de uso.

Una vez el administrador u profesor ha ingresado al sistema, podrá consultar a través de un link, los juegos que en ese momento se estén jugando.

El sistema generará a través de una ventana, el reporte de los juegos que se encuentren activos.

Figura 8, diagrama de caso de uso: consultar juegos activos

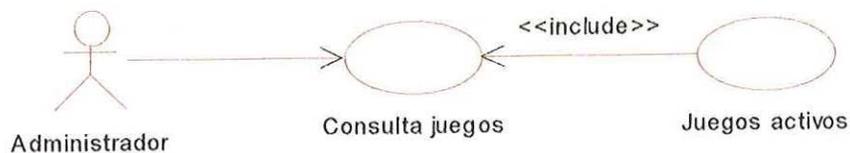
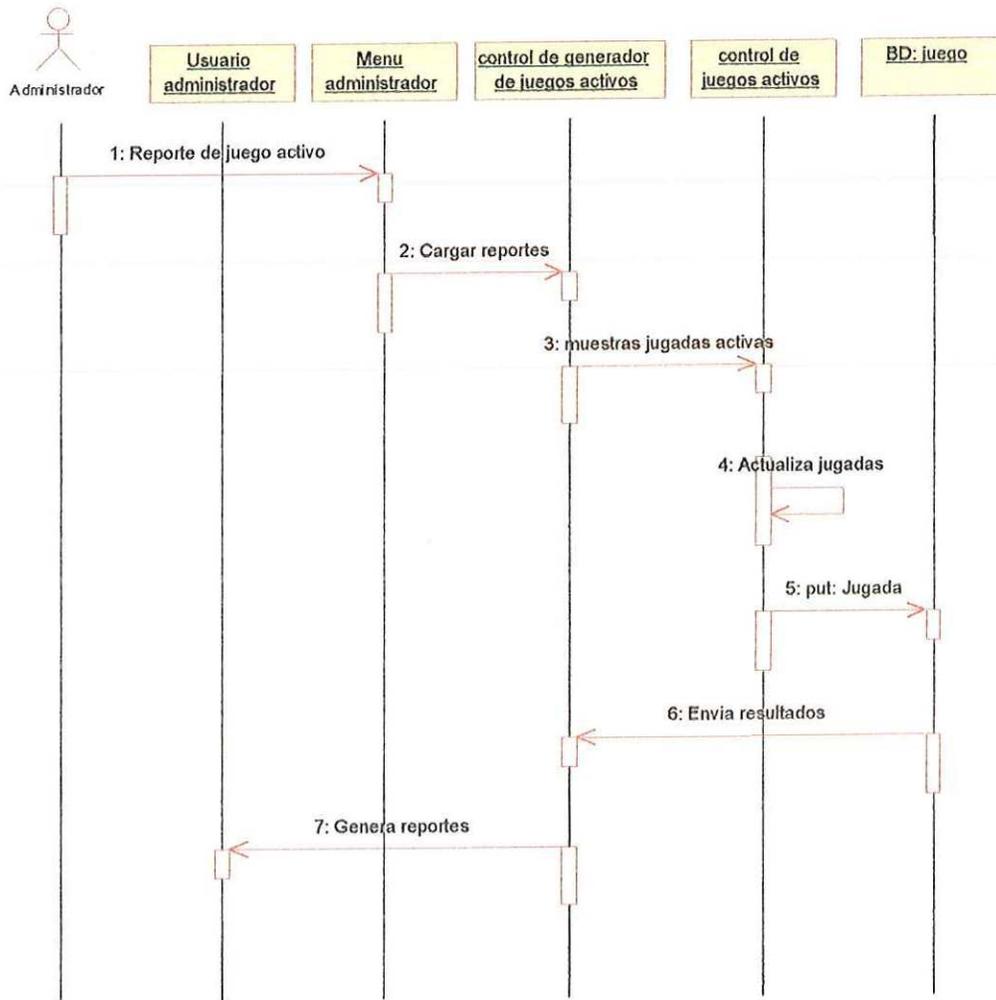


figura 9, Diagrama de secuencia: consultar juegos activos



3.8.5 Consulta de juegos terminados

Requerimientos

En la figura 10, muestra el diagrama de casos de uso de consultar juegos terminados, el cual trata de brindar una ayuda al profesor y así tener una guía o

idea de quienes son los usuarios que ya han jugado para así asignarles otro juego o consultar lo que el estudiante creo.

Análisis

En la figura 11, se muestra a través del diagrama de secuencias, la forma como se realiza el caso de uso.

Una vez el administrador u profesor ha ingresado al sistema, podrá consultar a través de un link, los juegos que en ese momento se estén terminados.

El sistema generará a través de una ventana, el reporte de los juegos que se encuentren terminados.

Figura 10, Diagrama de caso de uso: consultar juegos terminados

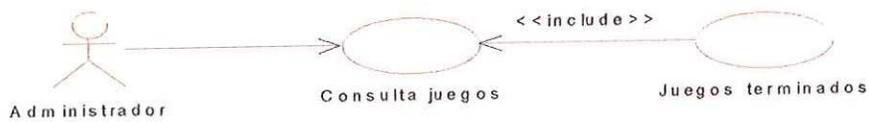
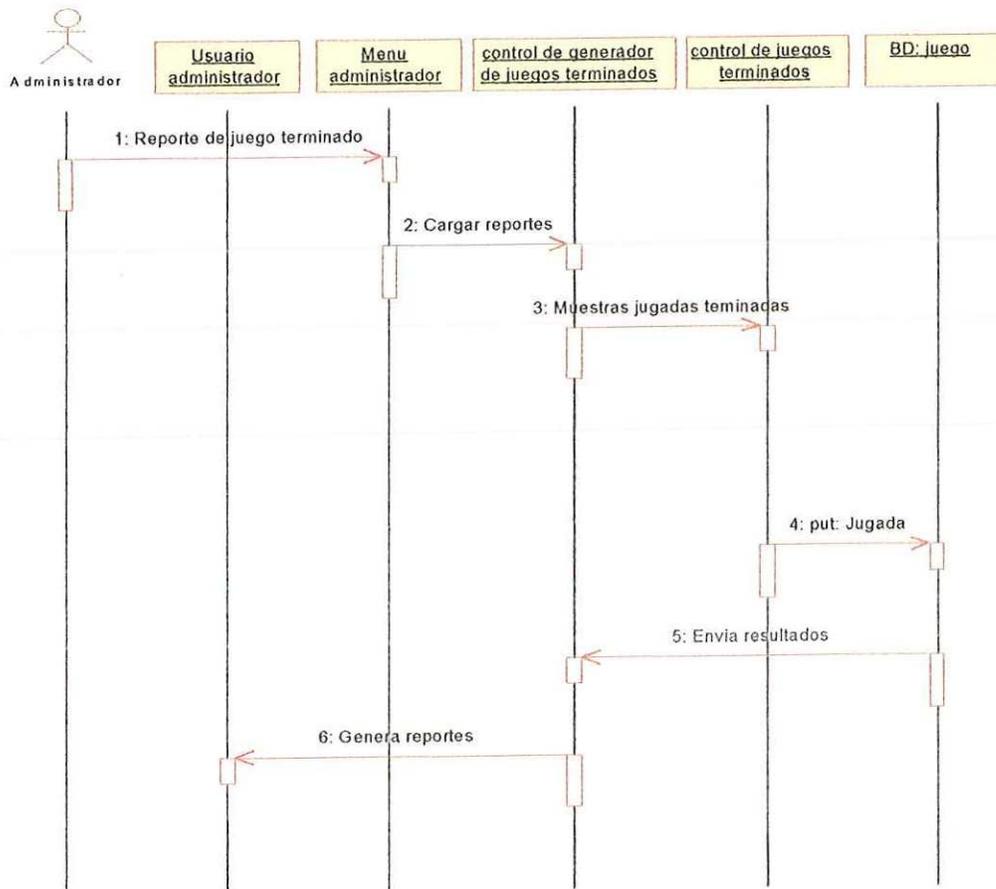


Figura 11, Diagrama de secuencia: consultar juegos terminados



3.8.6 Creación de usuarios

Requerimientos

En la figura 12, se muestra el diagrama de casos de uso, el cual pretende mostrar la creación de los usuarios por administrador y así llegar a formar cursos.

Análisis

En la figura 13, se muestra el diagrama de secuencia de creación de usuarios el cual se da de la siguiente forma .

La creación de usuario es realizada por el administrador del sistema, el cual le asigna un login y un password al usuario creado.

El administrador es el encargado de llenar los campos que el sistema pide en la creación de usuarios .

A continuación el sistema mostrara un mensaje de confirmación o creación una vez oprimido el botón guardar.

Figura 12, Diagrama de casos de uso: Creación de usuarios

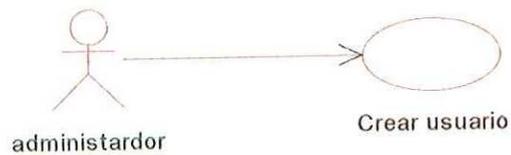
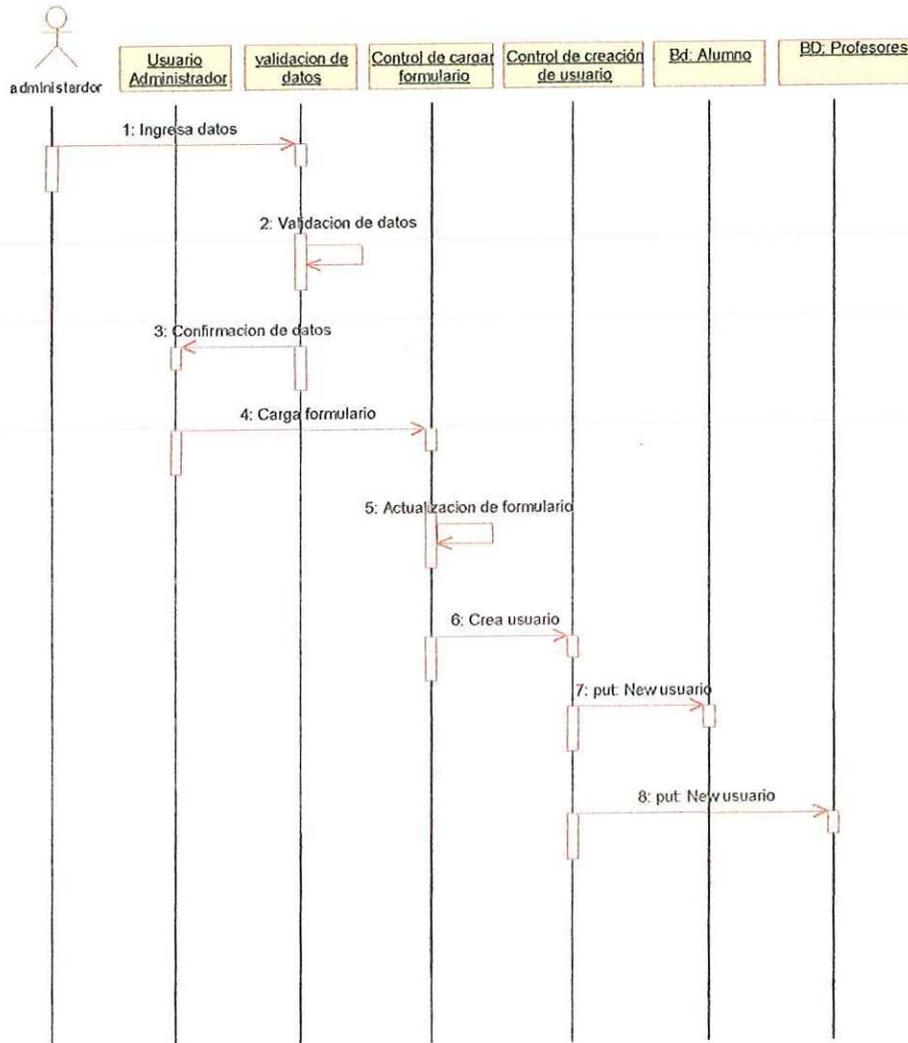


Figura 13, Diagrama de secuencias: Creación de usuarios



3.8.7 Crear juegos

Requerimientos

En la figura 14, se muestra el diagrama de casos de uso crear juego. Este caso de uso permite al administrador generar los juegos que le impone al estudiante

cuando este ingresa al sistema. El administrador puede crear un o mas juegos, dependiendo de los que desee que el estudiante desarrolle.

Análisis

En la figura 15, se muestra el diagrama de secuencia, este comienza cuando el administrador ingresa al sistema y escoge cualquiera de los juegos que se quiera jugar. A continuación y dependiendo del juego, el profesor o en este caso el administrador, empieza a llenar la base de datos con frases, sustantivos, gráficos o situaciones hipotéticas, para que el estudiante a la hora de entrar al sistema tenga juegos para empezar a realizar sus jugadas.

Figura 14, Diagrama de casos de uso: Creación de juegos

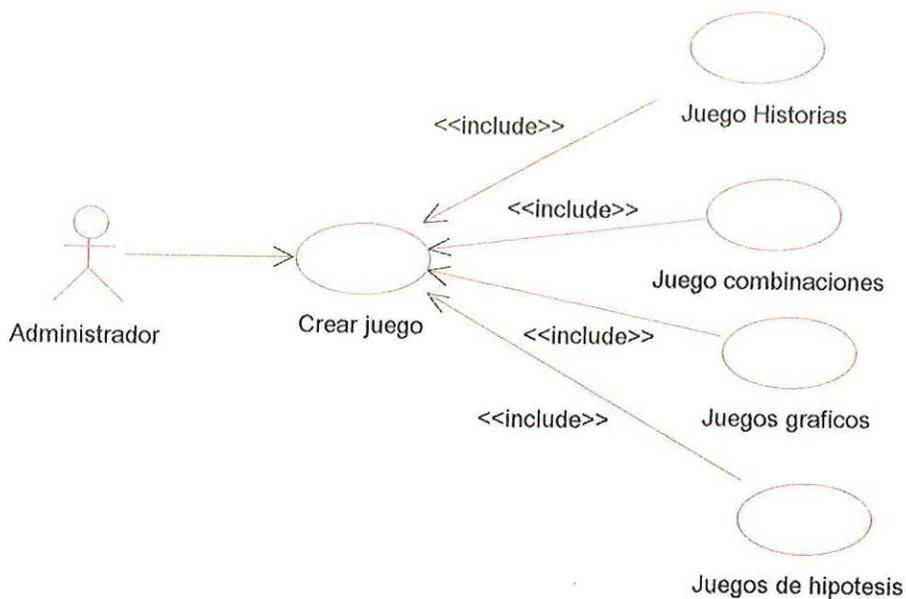
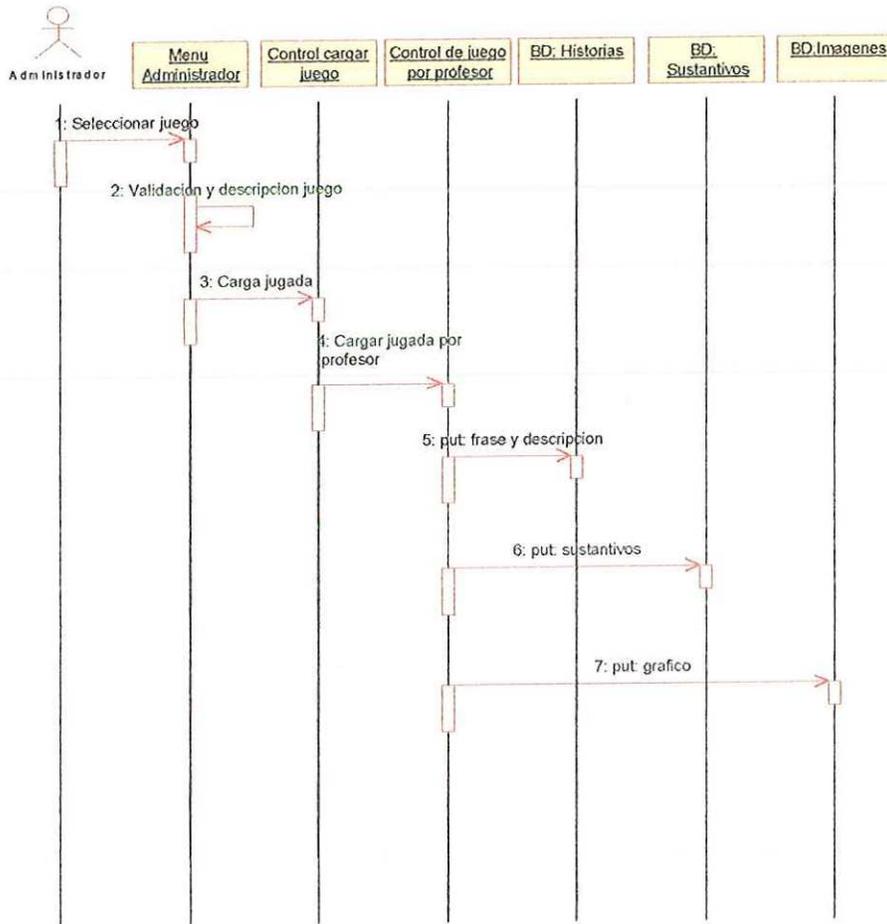


Figura 15 , Diagrama de secuencia : Creación de juegos



3.8.8 Enviar y recibir comentarios

Requerimientos

En la figura 16, se muestra el caso de uso, enviar y recibir comentarios, en este caso de uso es donde el administrador puede ver cual fue el trabajo del estudiante a través de los reportes generados por el sistema, de los juegos terminados, además, el administrador podrá enviarle a un correo externo, dando su opinión sobre el trabajo realizado, este correo externo, estará almacenado en un campo de la tabla estudiantes.

Análisis

En la figura 17, se muestra el diagrama de secuencia, enviar y recibir comentarios. Este comienza cuando el administrador quiere ver el resultado de los juegos que se han terminado como ya lo habíamos visto, además se da cuando el profesor por medio de su correo le envía al estudiante su opinión al respecto del juego.

Figura 16, Diagrama de casos de uso: Enviar y recibir comentarios

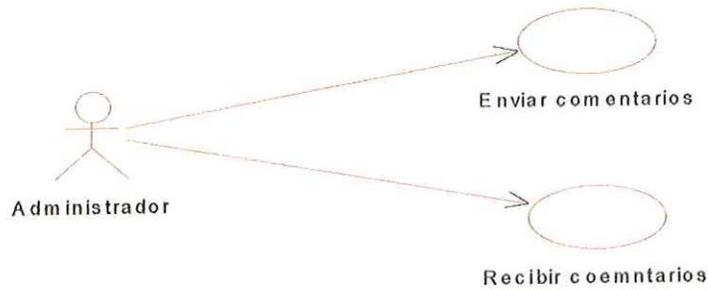
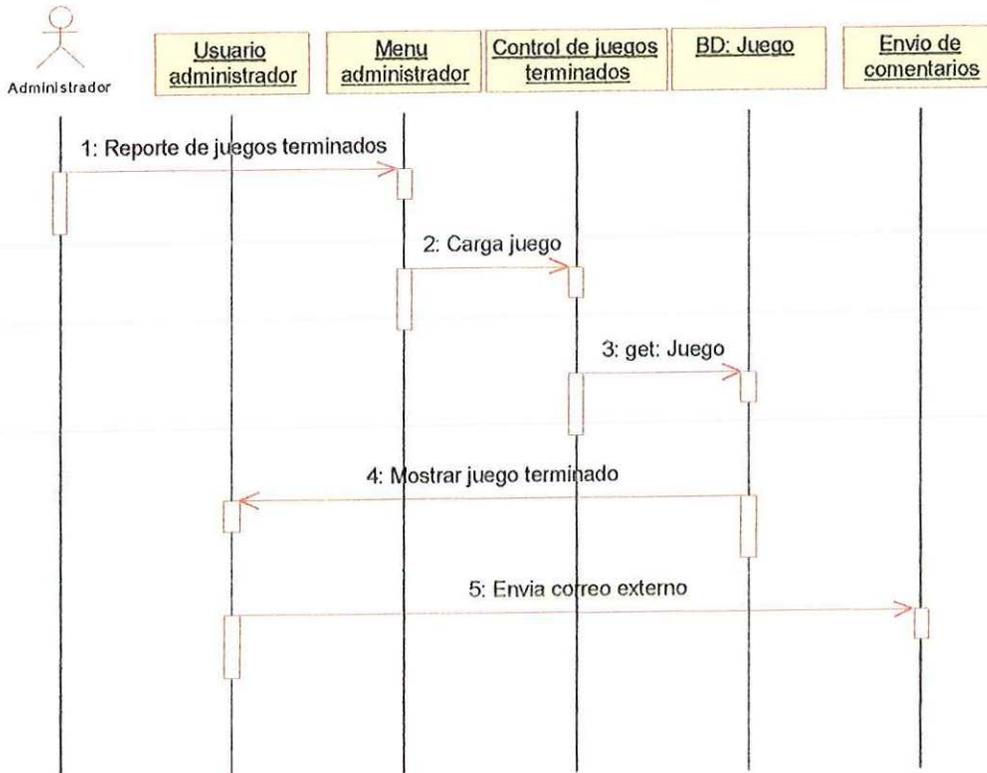


Figura 17, Diagrama de secuencia: Enviar y recibir comentarios



3.8.9 Crear curso

Requerimientos

En la figura 18, se muestra el diagrama de casos de uso Crear curso, Este caso de uso es donde el administrador vincula a un grupo de estudiante al curso que el quiera.

Análisis

En la figura 19, se muestra el diagrama de secuencia de crear un curso en el que el administrador ingresa los datos y el sistema se los valida. Si el usuario administrador es valido se carga el formulario del estudiante que el profesor quiere inscribir y le asigna su respectivo curso.

Figura 18, Diagrama de caso de uso: Crear curso

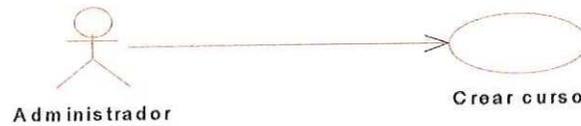
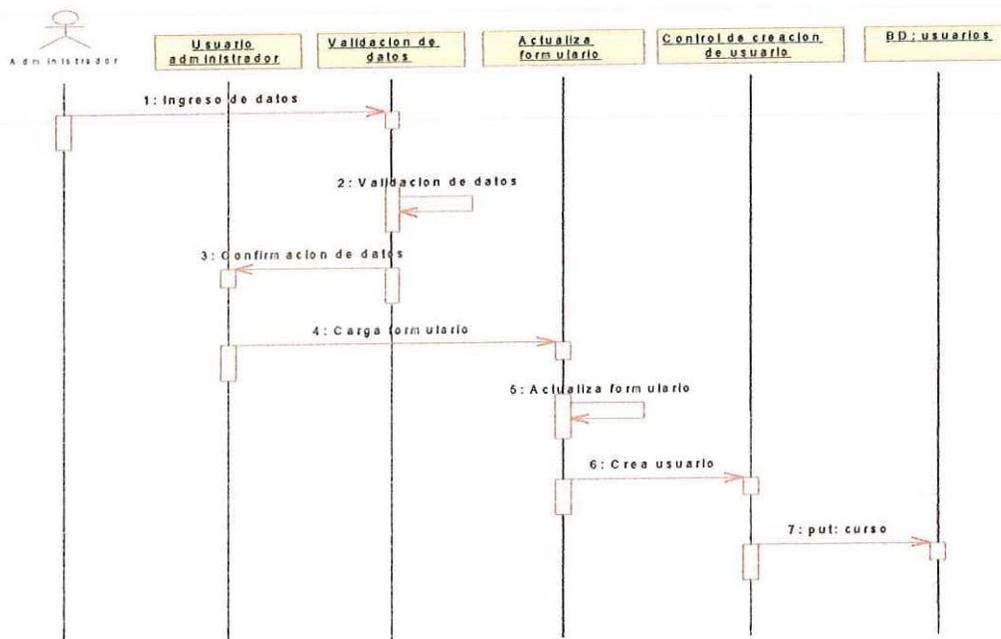


Figura 19, Diagrama de secuencia: Crear curso



3.8.10 Modificar datos de entrada

Requerimientos

En la figura 20, se muestra el caso de uso Modificar datos de entrada, en este caso de uso el estudiante podrá cambiar el password que el profesor le ha asignado por el que quiera.

Análisis

Una vez el usuario ingrese al sistema, tendrá la opción de cambiar su password o dejar el que tiene por defecto.

Si el usuario cambia su password, el sistema le pedirá escribir su password anterior, su nuevo password y la confirmación de este.

Cada vez que el usuario ingrese al sistema, deberá escribir su login y su nuevo password.

Figura 20, Diagrama de casos de uso : Modificar datos de entrada

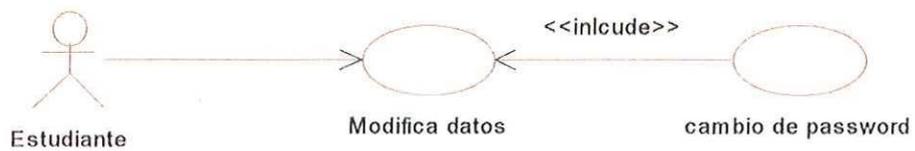
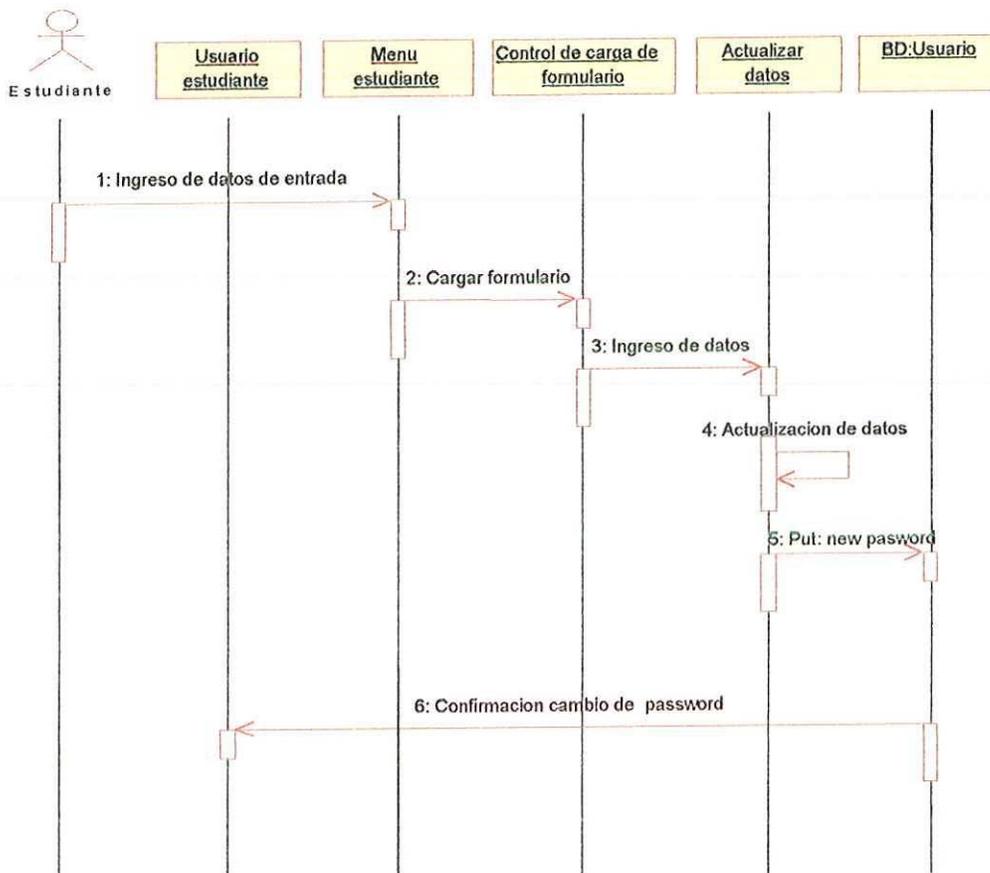


Figura 21, Diagrama de secuencia: Modificar datos de entrada



3.8.11 Empezar juego

Requerimientos

En la figura 22, se muestra el caso de uso Empezar juego, Ente caso de uso, muestra la forma como el estudiante comienza a jugar el juego que le ha asignado el profesor o el juego que el halla querido escoger para comenzar a realizar sus jugadas.

Análisis

El actor tendrá la opción de escoger el juego entre los que se encuentran creación de historias, hipótesis y combinaciones.

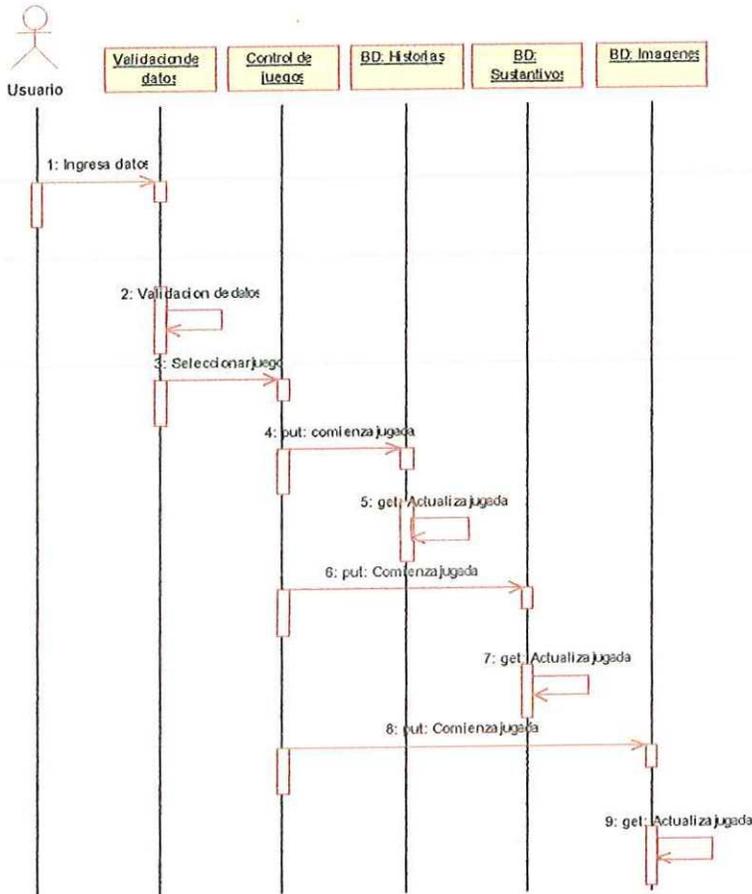
Mientras el usuario se encuentre esperando que se realice alguna jugada, este podrá empezar un juego nuevo.

Si el juego es combinaciones o hipótesis, el jugador deberá terminar el juego para empezar otro.

Figura 22, Diagrama de casos de uso: Empezar juego



Figura 23, Diagrama de secuencia: Empezar juego



3.8.12 Consulta de ayuda

Requerimientos

En la figura 24, mostramos el caso de uso consulta de ayuda, este caso de uso pretende que el estudiante a la hora de empezar un juego se de cuenta como es el procedimiento para poder jugar.

Análisis

Los actores del sistema podrán ver las instrucciones del juego por medio de un archivo texto el cual aparecerá oprimiendo el botón de ayuda.

Dentro de las instrucciones encontrara las diferentes formas de realizar jugadas las cuales serán de guía para un juego mas optimo

Figura 24, Diagrama de casos de uso: Consulta de ayuda.

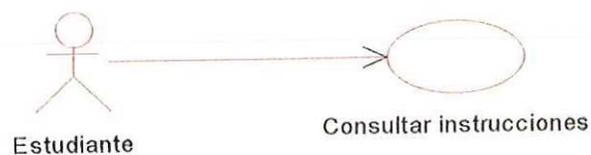
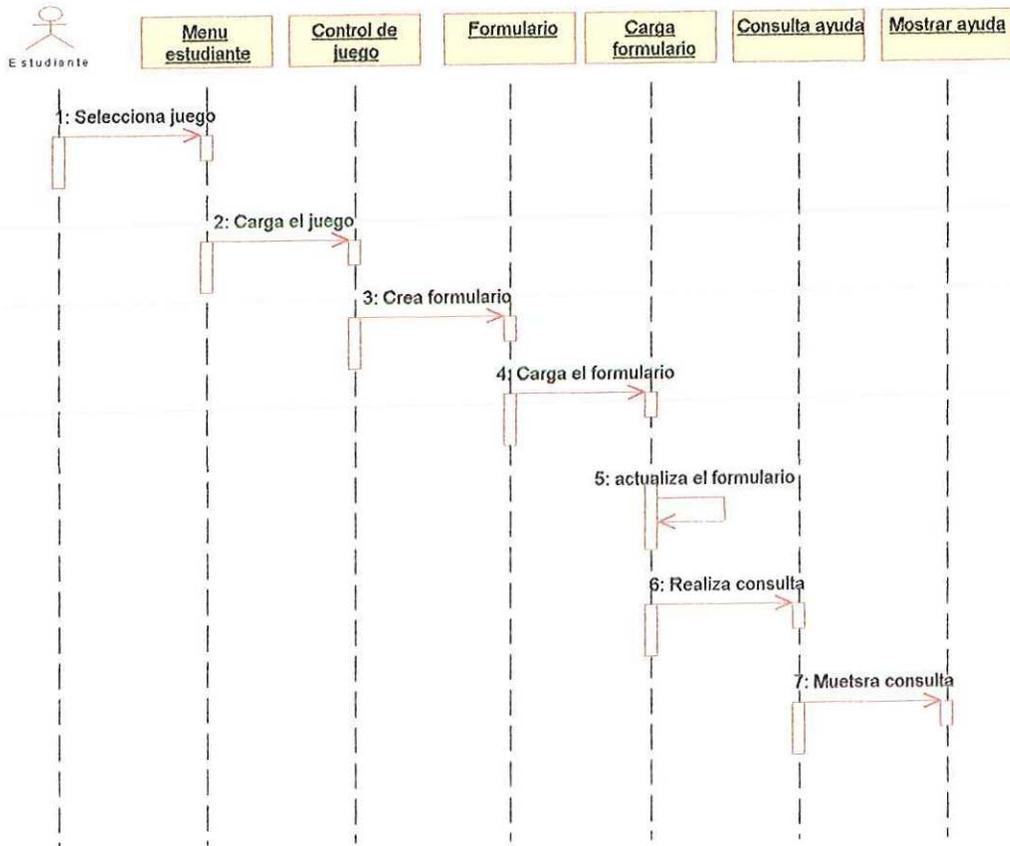


Figura 25, Diagrama de secuencia: Consulta de ayuda.



3.8.13 Editor grafico

Requerimientos

En la figura 26, mostramos el caso de uso del editor grafico, este caso de uso pretende que el estudiante fomente su creatividad por medio de figuras o imágenes.

Análisis

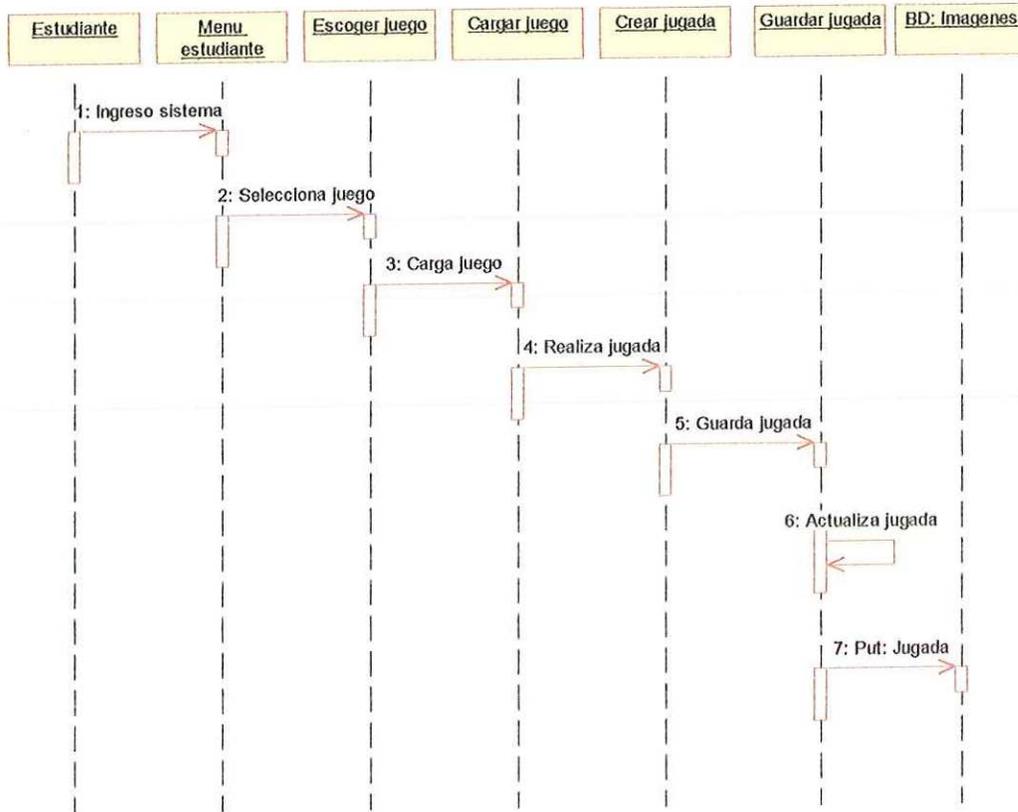
Los actores del sistema podrán ver realizar su jugada por medio de un archivo grafico en el cual aparecerán las diferentes figuras con las cuales podrá interactuar.

Dentro de las instrucciones encontrara las diferentes formas de realizar jugadas las cuales serán de guía.

Figura 26, diagrama de casos de uso: Juego editor grafico



Figura 27, Diagrama de secuencias: Juego editor grafico



CONCLUSIONES

Al comenzar el proyecto se seleccionaron ciertas actividades para desarrollar un prototipo de la herramienta y fomentar la creatividad en los estudiantes. La herramienta realizada consta de tres juegos, los cuales tienen en cuenta un espacio para la interacción, la memorización y el entretenimiento además permiten que cualquier persona pueda acceder la herramienta teniendo en cuenta que se ha montado en Internet.

El principal desafío que se encontró en el desarrollo de este proyecto se dio en gran parte a las nuevas tecnologías que se utilizaron porque requirió un aprendizaje previo, además todo el software que se utilizó es libre por lo tanto las ventajas en cuanto a disminución de costos, uso y distribución, salen a relucir.

Otro reto importante fue la búsqueda de personas que entendieran y capacitaran en las herramientas de programación el momento de dudas, dando así un paso agigantado en la enseñanza y el autoaprendizaje de las tecnologías trabajadas.

Con el desarrollo de esta aplicación se alcanzaron los objetivos planteados inicialmente, implementar actividades para la web, montarlas en algún entorno virtual y que ofrezca a estudiantes algún fin educativo trabajos como este motivan a seguir aprendiendo sobre tecnologías y métodos de desarrollo para implementar aplicaciones lúdico-educativas.

Se fortalecieron los conocimientos en la parte de programación, ya que se utilizaron tecnologías nunca antes vistas como lo fueron SVG, JAVA . Además se hizo énfasis en la pedagogía a través de la metodología de resolución de problemas la cual nos ayudo para atacar el problema que se expuso en un principio.

BIBLIOGRAFIA

<http://atenea.ucauca.edu.co/~gramirez/archivos/AnotacionesRUP.pdf>, universidad del cauca, Noviembre de 2003.

<http://www.creangel.com/uml/home.html>, Rational Rose, Copyright © 2000-2003

<http://www.creangel.com/uml/intro.html>, Rational Rose, Copyright © 2000-2003

<http://www.xtec.es/~jcorder1/problema.htm>, Juan Antonio Cordero, 1996-2000.

El proceso unificado de desarrollo de software, Ivar Jacobson, Grandy Booch, James Rumbaugh, Addison Wesley

<http://www.rimed.cu/Ense%F1anzas/SUPERIOR.ASP>, Ministerio de educación de la republica de cuba, Agosto de 2003

<http://espanol.geocities.com/criteriosevaluacion/procesocognitivo.htm>, Instituto Técnico Superior "Dámaso Zapata, Capacitación docente, Bucaramanga, Marzo de 2003.

<http://developer.java.sun.com/developer/onlineTraining/JSPIntro/contents.html> Sun Microsystems, Copyright 1996-2000

<http://java.sun.com/products/jdbc/index.jsp>, Sun Microsystems, copyright 1999-2000

<http://jakarta.apache.org/tomcat/>, the apache Jakarta Project ,Copyright © 1999-2004, Apache Software Foundation

http://www.programacion.com/java/tutorial/tomcatintro/1/#1_intro, java en castellano, Septiembre de 2003

<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?cap=1>, El rincón del programador, Lola Cárdenas Luque, Copyright © 2000-2004.

<http://rinconprog.metropoliglobal.com/CursosProg/Linux/Introduccion/index.php?ca p=2a> El rincón del programador, Lola Cárdenas Luque, Copyright © 2000-2004.

http://www.htmlweb.net/disenosvg/svg_1.html Luciano Moreno, del departamento de diseño web de BJS Software, Octubre de 2003.

http://www.htmlweb.net/disenosvg/svg_2.html, Luciano Moreno, del departamento de diseño web de BJS Software, Octubre de 2003.