

# Modelando el juego tradicional la Cuarta, practicado por los jóvenes de la región Caribe colombiana

Autor:

Luis Gabriel Turizo Martínez\*  
lturizo@coruniamericana.edu.co

Coautores:

Sergio Carlos Orozco Torres\*\*  
orozcosergio@coruniamericana.edu.co

David Enrique Sánchez Sierra\*\*\*  
sanchezdavid@coruniamericana.edu.co

## Resumen:

La Cuarta, es un juego tradicional practicado por jóvenes de la región Caribe colombiana, el cual tiene como acción principal lanzar una moneda y medir con la mano, encerrando un alto valor educativo, apto para ser llevado a una simulación y convertirlo en juego virtual. La idea principal es utilizar el modelo de las leyes físicas del movimiento parabólico y codificarlo en el lenguaje de programación Java, en un sistema que simule lo mejor posible lo practicado por los chicos, sirviendo como material didáctico para las enseñanzas de la física y las matemáticas, y una excelente opción para que los ingenieros de sistemas y programadores vean un recurso en que trabajar, utilizando situaciones propias de su entorno: los juegos tradicionales. El artículo muestra el proceso como estamos modelando las observaciones directas del juego, el evento físico presente (movimiento en el plano) y la forma como lo codificamos en el lenguaje Java.

## Palabras claves:

La Cuarta, movimiento parabólico, programación en Java, simulación.

## Abstract

The Fourth is a traditional game practiced by young Colombian Caribbean region, which has the main action and measure flipping a coin with his hand, enclosing a high educational value, suitable to be taken to make a simulation and virtual game. The main idea is to use the model of the physical laws of parabolic motion and encode in Java programming language, in a system that simulates best practiced by the boys, serving as teaching materials for teaching physics and mathematics, and an excellent choice for systems engineers and programmers see a resource in which to work, using situations within its environment: traditional games. The article shows the process as we are modeling the direct observations of the game, this physical event (plane motion) and the way they coded in Java language.

## Keywords:

The fourth, parabolic motion, Java programming, simulation.

## Introducción

El juego tradicional la Cuarta practicado por jóvenes de la región Caribe colombiana, consiste en lanzar una moneda sobre una pared y tratar de llegar a la de su adversario con una distancia de una cuarta. Cuando las monedas están

en el suelo, se mide la cuarta, que es la distancia entre el meñique y el pulgar extendidos. Por lo general, la cuarta tiene medidas entre 20 y 21 cm. Si las distancias de las monedas son iguales o menores a la cuarta, el jugador gana, de lo contrario cede el turno. Llevar a la virtualidad este juego es nuestra tarea.

\* Licenciado en Matemáticas y Física. Especialista en Pedagogía e Investigación en el Aula. Docente Tiempo Completo de la Corporación Universitaria Americana en el Programa de Ingeniería de Sistemas.

\*\* Sergio Carlos Orozco Torres. Estudiante de Sexto Semestre de Ingeniería de Sistemas en Coruniamericana. \*\*\* David Enrique Sánchez Sierra. Estudiante de Octavo Semestre de Ingeniería de Sistemas en Coruniamericana.

Artículo recibido: Diciembre 16/2011. Aceptado: Enero 28/2012.

Para el desarrollo de herramientas didácticas y tecnológicas, desde una perspectiva investigativa, es muy importante el concepto de modelo, el cual se afianza más en situaciones propias de las ingenierías, asumiendo diferentes facetas, dentro de las cuales en el caso de la ingeniería de sistemas se convierte en su máxima expresión cuando se habla de simulación.

En tal sentido, observar el juego la Cuarta es un primer paso para convertirlo en un modelo y luego simularlo.

El Ministerio de Educación Nacional en los estándares define un modelo: Un modelo puede entenderse como un sistema figurativo mental, gráfico o tridimensional que reproduce o representa la realidad en forma esquemática para hacerla más comprensible. Es una construcción o artefacto material o mental, un sistema –a veces se dice también “una estructura”– que puede usarse como referencia para lo que se trata de comprender; una imagen analógica que permite volver cercana y concreta una idea o un concepto para su apropiación y manejo.

Un modelo se produce para poder operar transformaciones o procedimientos experimentales sobre un conjunto de situaciones o un cierto número de objetos reales o imaginados, sin necesidad de manipularlos o dañarlos, para apoyar la formulación de conjeturas y razonamientos y dar pistas para avanzar hacia las demostraciones. (Ministerio de Educación Nacional, Estándares básicos de competencias en matemáticas, artículos-116042, 2004).

En los modelos de ingeniería, la simulación es una de las mejores opciones de modelamiento. El modelamiento del juego la Cuarta y posterior simulación, indica acotarlo en un sistema matemático y físico, para después poderlo amoldar a un sistema informático de códigos.

“Simulación es el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar el sistema” (Coos Bu, 1999).

### Desarrollo

En pleno siglo XXI, cuando el desarrollo del software avanza a tanta velocidad que no terminamos de adaptarnos a una herramienta informática específica cuando ya tenemos al alcance una mejor de su mismo tipo, irónicamente nos facilita el trabajo minimizando la complejidad de algunos componentes del sistema que hasta el momento representaban mucha dedicación y trabajo. Abriéndonos así un nuevo abanico de posibilidades de desarrollo de un software robusto y evolutivo.

Uno de esos grandes logros alcanzado por expertos en desarrollo del software y hoy soportados por la comunidad mundial de desarrollo de software libre es Java y es precisamente con este lenguaje de programación de alto nivel que el equipo investigador está desarrollando el aplicativo informático del juego la Cuarta.

Debido a la facilidad de ejecución en múltiples plataformas, optimizamos el trabajo de los miembros del grupo investigador que trabajan sobre plataformas distintas a Microsoft Windows, por ejemplo GNU/Linux.

El IDE NetBeans proporciona una interfaz gráfica agradable y fácil de manipular para el lenguaje de programación Java (aplica para otros lenguajes de programación), existen otras alternativas, por ejemplo eclipse (multilenguaje), qtcreator (c++/qt), codeblocks (c++), visual studio (c#, c++ / Windows forms, WPF), se escogió NetBeans debido a que el equipo conocía su modo de trabajo, es multiplataforma, genera los ejecutables (jar) al momento de la compilación.

Visual studio se descartó, debido a que es plataforma propietaria, algo que no va con la filosofía de código libre que se está manejando con el proyecto. Los demás se descartaron democráticamente, la mayoría del equipo en ese momento usaba NetBeans.

El desarrollo del prototipo lo comenzamos con C++ y más tarde encaminamos todo a Java usando el IDE NetBeans.

Dentro de estas razones tenemos:

- Flexibiliza el trabajo para el equipo investigador.
- La reutilización de código.
- Con Java eliminamos la dependencia entre aplicación con el sistema operativo.
- Optimizamos y volvemos más versátil y eficiente el desarrollo del código de la aplicación.
- El que mejor se ajusta a las necesidades del equipo desarrollador de la aplicación. Java proporciona una interfaz gráfica agradable y fácil junto al IDE NetBeans.

Al realizar observaciones directas del juego, delimitarlo y contextualizarlo, de manera explicativa-descriptiva, desde los principios físicos de la mecánica y sus respectivas ecuaciones matemáticas y para facilitar el proceso consideraremos únicamente el primer toque de la moneda en el suelo, no trataremos los movimientos posteriores, es decir, que la moneda cuando toque el suelo, quede estática, además tampoco tendremos en cuenta: la resistencia del aire, la rotación de la moneda y la rigurosidad de la pared.

Para abordar esta situación empezamos observando a los jóvenes jugar la Cuarta.

En el proceso pudimos deducir que prácticamente los estudiantes lanzan la moneda de tres maneras diferentes, las cuales fueron modeladas matemáticamente utilizando las leyes del movimiento parabólico.

Cuando hablamos de movimiento parabólico estamos utilizando el término físico que acoge a todos aquellos movimientos que están sujetos o bajo los efectos de la aceleración de la gravedad y que describen una curva, en forma de parábola, originado por la combinación de un movimiento con velocidad horizontal constante y otro con aceleración constante vertical.

Estas ecuaciones pueden llevarse al juego la Cuarta, sirviendo como preámbulo para considerar que la aceleración constante es la aceleración de la gravedad ( $g = -9.8 \text{ m/s}^2$ ), incluida en el movimiento vertical que hace bajar la moneda, estos movimientos modelan el comportamiento de la moneda.

En los modelos de ingeniería, son muy comunes los modelos gráficos y esquemáticos, tenemos el modelo de casos de usos y el modelo de dominio. En UML (Unified Modeling Language, Lenguaje Unificado de Modelado) se utiliza para este proceso dos términos técnicos claves: actores y casos de uso.

“Un actor es un usuario del sistema que tiene un rol particular...Un caso de uso es una tarea que un actor necesita ejecutar con la ayuda del sistema” (Stevens y Pooley, 2007).

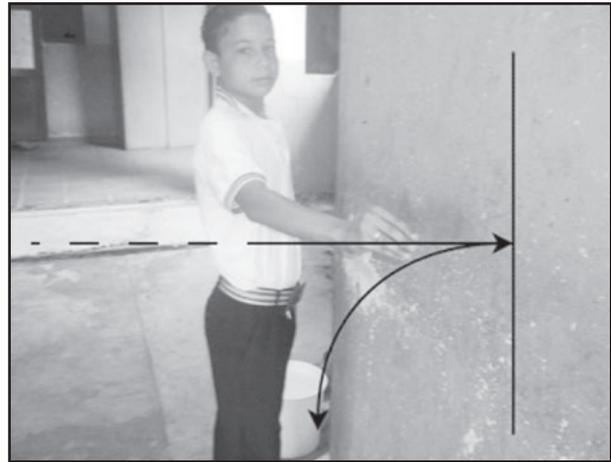
Se define como: “Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés” (Larman, 2003).

El modelo de casos de uso, nos ayudarán al análisis y descripción sistemática de lo que se hará y se debe hacer, entre el usuario (jugador) y el servidor (administrador) es decir, de una manera más amplia nos ayuda a comprender la forma como interactuarán lo externo con lo interno dentro de la simulación: jugador (es) y administrador.

De igual manera los Modelos de Dominio permitirán comprender básicamente la realidad física que debemos capturar para luego expresar en el lenguaje de simulación, en nuestro juego: jugador, monedas, partidas, lanzamiento, pared, entre otros.

Observando detenidamente el juego dedujimos los siguientes tres tipos de lanzamientos expresados para el alcance (Distancia sobre terreno horizontal, medido entre el punto de lanzamiento y el de caída o pique de la moneda):

### 1. Lanzamiento de la moneda perpendicularmente sobre la pared.



Esta jugada sucede cuando la moneda es lanzada por el jugador formando un ángulo recto contra la pared desde una altura escogida, conduciendo a la ecuación para el alcance  $X: v_0 \sqrt{2h/g}$

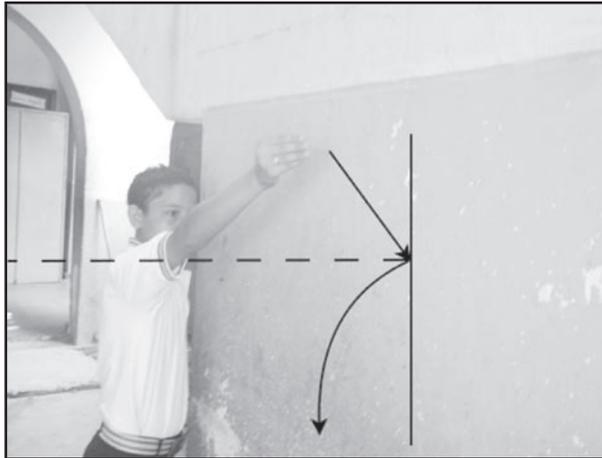
La demostración nos conduce a que, el alcance de la moneda estará regido por  $x = v_0 \sqrt{2h/g}$  el producto entre la velocidad  $v_0$  con que sale la moneda de la pared y la raíz cuadrada de dos veces la altura de lanzamiento  $h$  dividido por la aceleración de la gravedad  $g$  (considerándola de  $9,8 \text{ m/s}^2$ ). Es decir, a mayor velocidad o a mayor altura la moneda se alejará de la pared.



### 2. Lanzamiento de la moneda por debajo de la línea de perpendicularidad:

Esta jugada genera una trayectoria parabólica completa y es realizada por debajo de la línea de referencia oblicua, es decir, forma un ángulo agudo. La demostración conduce a la siguiente ecuación para el alcance, donde  $v_0$  velocidad de salida de la moneda,  $\Phi$  es el ángulo agudo que forma con la línea

de referencia,  $\sin \Phi$  y  $\cos \Phi$  son las funciones trigonométricas presentes en las componentes de la velocidad  $v_0$ :  $X=v_0 \cos\theta(-v_0 \sin\theta+\sqrt{(9v_0^2 \sin^2\theta+2gh/g)}$



### 3. Lanzamiento de la moneda por encima de la línea de perpendicularidad:

Este lanzamiento es un poco raro pero se observa, es una jugada de pique sutil que no genera mucho alcance, haciéndose por encima de la línea de referencia, de igual manera formándose un ángulo agudo. La ecuación que describe este comportamiento es:

$$X=v_0 \cos\theta(v_0 \sin\theta+\sqrt{(v_0^2 \sin^2\theta+2gh/g)}$$

Esto lo estamos llevando a un lenguaje informático, extrayendo un fragmento:

```
public Lanzamiento(Double h, double v, int ángulo,
Lanzador panel){
this(h, v, ángulo, panel, 100, 20);
}

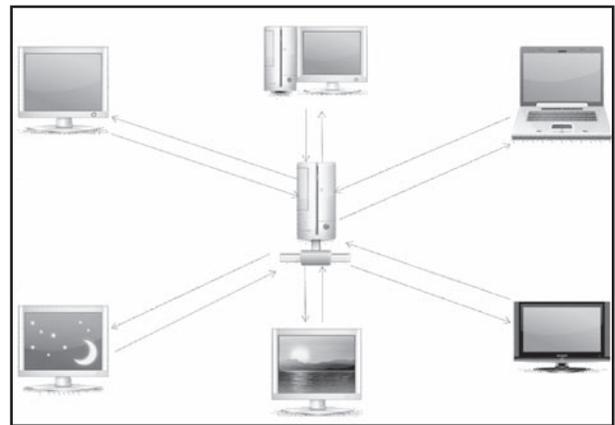
/**
 * Constructor del Lanzamiento
 * @param h: altura de la moneda
 * @param v: velocidad de la moneda
 * @param angulo: angulo desde el que es lanzada la moneda
 * @param panel: panel que se encargará de graficar
 * @param escala: escala del dibujo
 * @param hiperTiempo: escala de tiempo para el hilo
 */
public Lanzamiento(Double h, double v, int ángulo,
Lanzador panel, int escala, int hiperTiempo){
double cos_a, sen_a;
this.panel=panel;
this.escala=escala;
this.altura=h;
this.hiperTiempo=hiperTiempo;
sen_a = Math.sin(Math.toRadians(angulo));
cos_a = (angulo==90||angulo===-90)?0:Math.
cos(Math.toRadians(angulo));
Vy = v * sen_a;
Vx = v * cos_a;
```

```
tiempoTotal=(v*sen_a+Math.sqrt(v*v*sen_a*sen_a+19.6*h))/9.8;
alcance = v*cos_a*tiempoTotal;
double alturaMax=v*v*sen_a*sen_a/(-2*g);
System.out.println(alturaMax);
panel.setValores(h, escala, alcance, alturaMax>0?alturaMax:0);
}
```

Otro aspecto muy importante es el relacionado con la arquitectura de cliente-servidor. Para el juego la cuarta aplicar la arquitectura de cliente servidor, consiste en las múltiples labores atendidas por un servidor, haciendo posible que el juego se proyecte a plataformas en línea o de acceso remoto.

“Esta forma de comunicación está orientada a soportar los roles y el intercambio de mensajes de las interacciones típicas cliente-servidor. En el caso normal, la comunicación petición-respuesta es síncrona, ya que el proceso cliente se bloquea hasta que llega la respuesta servidor.

Esta comunicación también puede ser fiable ya que la respuesta al servidor es, un efecto, un acuse de recibo para el cliente. La comunicación cliente servidor asíncrona es una alternativa que puede ser útil en situaciones donde los clientes pueden recuperar las respuestas más tarde”. (Coulouris y Kindberg 2005).



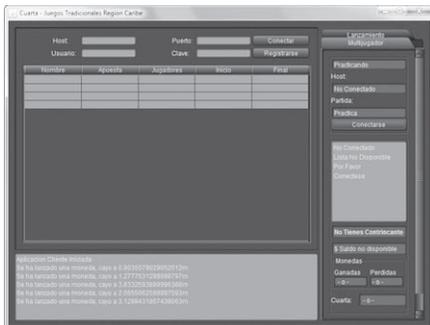
En nuestro aplicativo en la GUI (Graphic User Interface - Interfaz Gráfica de Usuario) del cliente, usamos un panel (espacio delimitado en pantalla) donde se colocan las monedas, este simula el piso donde se está jugando.

Primero, el jugador lanza la moneda, el aplicativo cliente encargado de realizar los cálculos, los lanzamientos para saber dónde cae la moneda según la información dada por el jugador, esto se logra usando las fórmulas del movimiento parabólico y posteriormente las del impulso, para encontrar el punto exacto donde caerá la moneda. Ahora, la Cuarta es jugada por más de un jugador, por tanto se hace necesario

buscar una forma de hacer que un jugador lance la moneda y que los demás sepan que esta se lanzó y donde cayó la misma. Consideramos construir el sistema en donde todos los jugadores interactúan con una única ventana en el mismo equipo computacional, pero de esta forma se genera un problema el cual consiste en que un jugador sabrá como fué el lanzamiento de sus contrincantes, conocerá los datos usados en el tiro y colocará valores muy cercanos o los mismos lo cual es considerado trampa, ahora usando la arquitectura cliente-servidor logramos que cada persona lance estando en su propio aplicativo guardando así privacidad en los datos de sus lanzamientos.

El software planteado ofrece la posibilidad de juego en un PC Servidor donde se establece la partida y recibe cada uno de los puntos donde cae la moneda, luego envía la información pertinente a todos los equipos cliente, con el fin de dibujar en los demás paneles la posición en donde han caído las monedas. Con base a lo dicho anteriormente, lo que se pretende es hacer que la lógica esté distribuida, entre el cliente y el servidor. Podemos centralizar el juego con esta técnica, permitiendo saber qué ha sucedido durante la partida (almacenado). Como la información permanece en un solo equipo (servidor), para tener acceso a ella sólo debemos solicitarla.

Para nuestro prototipo, planeamos mantener información estadística que sirva para comparar jugadores, analizar cómo se realiza el juego. La información que almacenamos de los jugadores son: el nombre, un identificativo dentro del juego, su clave, su fecha de nacimiento (para saber su edad), su Cuarta. Sólo faltaría una cosa, aunque sea un juego, queremos que sea robusto. Lo normal es que los jugadores se conozcan, y por supuesto, queremos mantener información estadística, debemos almacenar algunos datos de los clientes como son el nombre, la identificación dentro y fuera del juego, las estadísticas de cada uno de los tiros del mismo, y el tamaño de la Cuarta del jugador es relevante para saber en qué momentos gana monedas al oponente y que tan bueno es jugando. Para finalizar escogimos los colores: rojo Cardín tipo ladrillo y gris oscuro tipo cemento, todo esto para que se sienta que se está jugando en el suelo.



Gracias a la portabilidad del lenguaje Java podemos afirmar que usamos el lenguaje como plataforma para mostrar la simulación virtual del juego con la garantía de ejecución independientemente al entorno operacional donde se encuentre la aplicación. En nuestro proyecto estamos brindando una interfaz gráfica vectorial, amigable y con presentación de cálculos lo más exacto posible para recrear la realidad de los lanzamientos de las monedas que tiren los jugadores en plena competencia.

Encontramos en Java bibliotecas:

-Que nos ayudan en la comunicación entre el cliente y el servidor; facilitando el desarrollo del módulo multijugador, es decir, la parte del software encargada de administrar los mensajes entre los jugadores y el servidor.

-De código las cuales hasta el momento contienen las bases lógicas para gran parte del desarrollo de nuestro proyecto. Una librería es un conjunto de código externo prediseñado necesario para el desarrollo de las aplicaciones usando el principio de reutilización de código.

**Conclusiones:**

Lo presentado anteriormente en una manera concisa de nuestro trabajo en el modelado del juego tradicional la Cuarta, es lo que llevamos desarrollado hasta el momento, sin embargo hemos realizado avances y cambios:

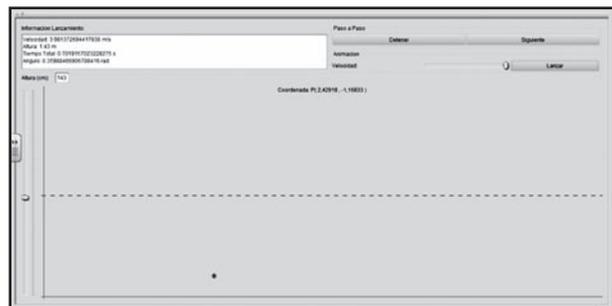
- Añadir la funcionalidad de encriptación de contraseñas para el acceso de los jugadores.

Por ejemplo: Para la contraseña 007 se envía: e94b15ed312fa42232fd87a55db0d39

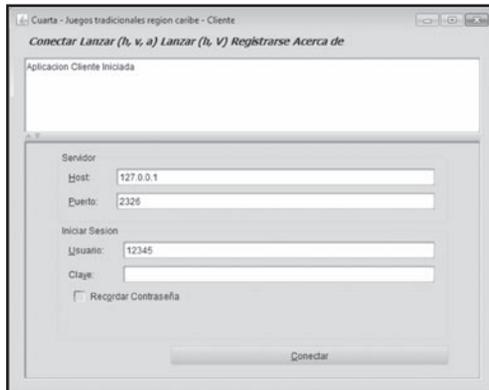
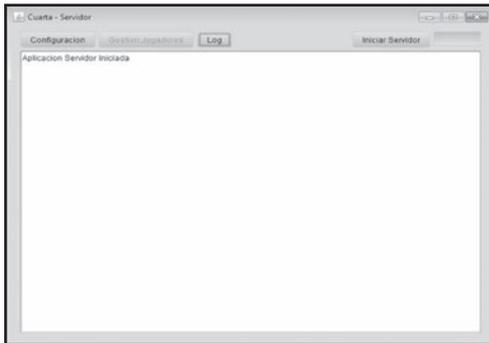
- Añadida una barra de herramientas enlazando a las funciones principales del juego.

*Conectar Lanzar (h, v, a) Lanzar (h, V) Registrarse Acerca de*

- Añadida una nueva opción para lanzamientos usando el vector velocidad y la altura con respecto al piso, implementando un panel especial para ello.



- Cambio de colores, servidor (gris, azul oceánico) cliente (azul cielo, dorado).



- Modificada la posición del panel del log, añadida la posibilidad de ocultarla u ocultar el panel principal.



## Bibliografía

BUNGE, Mario. (1997). La ciencia, su método y su filosofía. Ediciones Sudamericanas. Buenos Aires, Argentina.

COOS BÚ, Raúl, 1999. Simulación un enfoque práctico. Limusa Noriega Editores, México, pág. 12.

Estándares Básicos de calidad en Matemáticas y Lenguaje (2004). ¿Conoce usted lo que sus hijos deben saber y saber hacer con lo que aprenden? Ministe-

rio de Educación Nacional, Santa Fe de Bogotá pág. 52.

Estándares Básicos de Competencias en Ciencias Naturales y Ciencias Sociales (2004). Formar en Ciencias: el desafío! Ministerio de Educación Nacional, Santa Fe de Bogotá.

HERNÁNDEZ Sampieri, Roberto. Fernández Collado, Carlos y Baptista Lucio, Pilar. (2007). Fundamentos de Metodología de la Investigación. McGraw-Hill Interamericana.

JIMÉNEZ Valdez, Carlos Alberto (2003). Neuropedagogía, lúdica y competencias Cooperativa Editorial Magisterio, pág. 79.

LARMAN Graig, 2003. Uml y Patrones Fowler M-:Prentice Hall, México, pág. 122.

Lineamientos Curriculares, Matemáticas Lineamientos Curriculares. Cooperativa Editorial Magisterio. Ministerio de Educación Nacional, Santa Fe de Bogotá, 1998.

Lineamientos Curriculares, Ciencias Naturales y Educación Ambiental. Cooperativa Editorial Magisterio. Ministerio de Educación Nacional, Santa Fe de Bogotá, 2000.

Manual de Convivencia de la Institución Educativa Distrital Nuestra Señora del Rosario, Distrito de Barranquilla, sin página.

MONTENEGRO A., Ignacio Abdón (2004). Evaluemos Competencias Matemática. Cooperativa Editorial Magisterio.

NEGRET J. (1996). El Constructivismo, un modelo de enseñanza o un modelo de aprendizaje. Revista el Educador Frente al Cambio.

Planes de mejoramiento (2004). Y ahora...¿cómo mejoramos? Ministerio de Educación Nacional, Santa Fe de Bogotá.

QUICENO H. (1998). Corrientes Pedagógicas en el s. XX en Colombia. Revista El educador frente al Cambio.

SABINO, Carlos A. (1994). El proceso de Investigación. El Cid Editor. Bogotá, Colombia.

STEVENS Perdita, Pooley Rob (2007). Utilización de UML en Ingeniería del Software con Objetos y Componentes, 2da. Edición, Pearson, Addison Wesley, Madrid.

Vigésima Segunda Edición del Diccionario de la Real Academia de la Lengua Española. <http://buscon.rae.es/drae/>