

Representación Gráfica y Manipulación Matemática de Vectores a través de ROOT*

Andrés Mateus Vargas¹

¹Grupo de Investigación en Física e Informática (Fisinform)

Recibido: 20 de Octubre 2012 Aceptado: 5 de Noviembre 2012
Todos los derechos reservados-SEF © 2012

Abstract. By means of vectorial diagram construction is presented the ROOT framework's capacity, ease of use, and its potencial application in teaching and software development for physics teaching.

Keywords: ROOT, Vectorial Diagram, Physics Education, Software Development, Graphical User Interfaces.

Resumen. Se presenta mediante la construcción de un diagrama de vectores la capacidad y facilidad de uso de la infraestructura de desarrollo ROOT y su potencial de aplicación en la enseñanza y en el desarrollo de software para la enseñanza de la física.

Descriptores: ROOT, Diagrama Vectorial, Enseñanza de la Física, Desarrollo de Software, Interfaces Gráficas de Usuario.

1. Introducción

La multiplicidad y diversidad de herramientas utilizadas en el proceso de enseñanza-aprendizaje de las ciencias no solo facilita la construcción y reestructuración del conocimiento, sino que amplía la visión de los estudiantes respecto al amplio rango de disciplinas para el estudio y ejercicio profesional en los diferentes niveles de profundización, la incorporación de soluciones a nivel de software en el área educativa ofrece además un mayor nivel de autonomía, al proporcionar la comunidad académica al estudiante la posibilidad de trasladar el laboratorio de física del lugar de estudios al lugar de preferencia y autoformación

*ROOT: Infraestructura de Tecnología Rápida orientada a objetos para el análisis de datos, basada en el lenguaje C++. Del inglés "Rapid Object-Oriented Technology, a data analysis framework" (ver <http://root.cern.ch/root/roottalk/roottalk98/0718.html>).

amvargash@correo.udistrital.edu.co

donde existe un mayor grado de libertad, además de la posibilidad de interacción directa y de investigación sobre los modelos que el software logra implementar.

En este marco, el conocimiento acerca de cómo llevar a cabo y qué características debe tener determinado software para la enseñanza de ciencias como la física, corresponde a los docentes, teniendo en cuenta que éstos últimos son los directamente involucrados y los que tienen conocimiento de primera mano y contextualizado acerca de las dificultades presentadas en el tratamiento de determinadas temáticas, aunque existen parámetros objetivos que deben ser tenidos en cuenta a la hora de evaluarlo (ver (Navarro, Juez, Infante, y Hurtado, 2012)), el desarrollo de software es un elemento de profesionalización docente ya que proporciona mecanismos mediante los cuales es posible superar inconvenientes que hallan sido observados, adaptando software previamente desarrollado a necesidades y contextos específicos u ofreciendo alternativas mediante sus propuestas de diseño e implementación.

Es bien conocido que el lenguaje de programación que mayor presencia ha tenido en el desarrollo de aplicativos donde se reproducen modelos físicos enfocados a la visualización de experiencias de laboratorio ha sido Java, en gran parte debido a su incorporación y expansión en entornos web, no obstante, Java es un lenguaje de propósito general, lo que no dificulta pero prolonga la elaboración de software especializado, a diferencia de Java, ROOT es una infraestructura diseñada por y para su uso en el área de la física y es debido a esto que facilita y agiliza la escritura de prototipos funcionales para la enseñanza de ésta, si bien pueden encontrarse otras herramientas como Easy Java Simulations (ver (Esquembre, 2004)) que también simplifican la escritura de software para el desarrollo de simulaciones.

La infraestructura de desarrollo de software (originalmente categorizada como infraestructura para análisis de datos) ROOT, permite crear de una manera rápida prototipos de software, ofreciendo por un lado la flexibilidad de un interprete (CINT), la potencia de un lenguaje de programación de alto nivel como C++, y un conjunto de más de 2000 clases desarrolladas y adaptadas para ofrecer la máxima utilidad al campo de la física.

2. Objetivo

Mediante la realización del taller se busca presentar parte de la sección gráfica de ROOT, básicamente la utilización de líneas, flechas y ejes coordenados, donde el estudiante mediante la escritura de un prototipo de software logre comprender la estructura de éste logrando un acercamiento directo y familiarización con el desarrollo de interfaces gráficas que permitan crear inquietud en el estudiante y motivarlo para la realización de un estudio más profundo que le permita desarrollar aplicaciones cuya utilidad se vea reflejada en su ejercicio docente y en su proceso de formación.

3. Características del Taller

- **Duración:** 40 min
- **Nivel:** Básico. Estudiantes que hayan cursado la asignatura de Sistemas Computacionales I y/o personas con conocimientos básicos en programación.
- **Número mínimo de participantes:** 5;
- **Número máximo de participantes:** 15;

4. Estructura del Taller

El prototipo de software realizará la representación gráfica de las operaciones básicas con vectores bidimensionales en coordenadas rectangulares, para lo cual se hará uso de las clases TCanvas, TGaxis, TVector2, TArrow y TLine, el objetivo del uso de estas clases es mostrar la facilidad de construcción de una interfaz gráfica en ROOT (TCanvas, TGaxis, TArrow y TLine) y la utilidad del paradigma de la programación orientada a objetos mediante la clase TVector2, observando cómo al crear instancias de ésta clase, se puede realizar las operaciones de suma, resta y rotación de una manera natural e intuitiva ya que se hace uso de métodos y operadores definidos en ésta clase.

Se iniciará explorando la necesidad de un espacio determinado para realizar la representación de los vectores, su creación y configuración a través de métodos de la clase TCanvas, lo cual es análogo a la disposición del tablero del salón de clases mediante las siguientes líneas:

```
Tablero = new TCanvas();
Tablero->SetWindowSize(500,500);
Tablero->Range(-10,-10,10,10);
```

Se explicará la estructura sintáctica y los argumentos de cada método utilizado así como la utilidad en la aplicación de la clase.

El paso siguiente es la realización de los ejes coordenados, lo cual se logra a través de:

```
EjeX = new TGaxis(-10,0,10,0,-10,10);
EjeY = new TGaxis(0,-10,0,10,-10,10);
EjeX->Draw();
EjeY->Draw();
```

Líneas que permiten introducir el concepto de constructor, y la diferenciación entre la existencia de un objeto en memoria y la representación en pantalla mediante el método Draw(), una vez introducidos los constructores se escriben los vectores:

```

R1 = TVector2(1,5);
R2 = TVector2(-4,3);
R3 = TVector2();
R3 = R1 + R2;

```

Éstas líneas permiten presentar la gran utilidad que representan los objetos a la hora de programar, la definición de operadores en la clase TVector2, permite al usuario/programador abstraerse de la realización de las operaciones matemáticas que conducen a la adición de vectores, debido a que al incluir las clases (precargadas en el interprete) se le enseña al software cómo llevar a cabo las operaciones matemáticas con vectores.

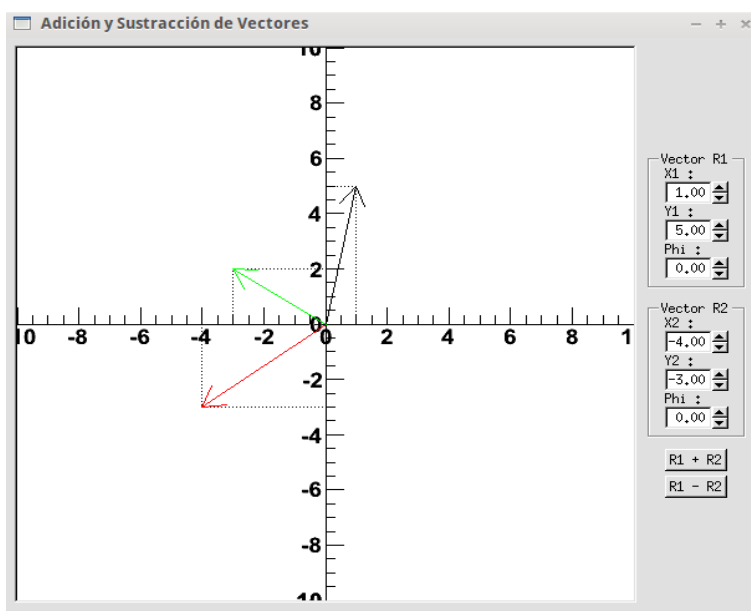


Figura 1: Interfaz gráfica del prototipo a reproducir en el taller

Los vectores al representar entidades matemáticas abstractas (matrices, sumas de componentes en vectores unitarios de un sistema coordenado), debe mediar un procedimiento mediante el cual puedan visualizarse, la representación gráfica de los vectores (en el caso particular de 2 dimensiones) se realiza mediante segmentos de recta dirigidos:

```

A1 = new TArrow(0,0,R1->X(),R1->Y());
A1->SetLineColor(1);
A1->Draw();

```

Éstas sentencias permiten detallar las características de los objetos y la comunicación entre éstos (constructor de A1), para finalizar se añaden mediante instancias de TLine las proyecciones, y se da formato a dichas proyecciones.

La interfaz gráfica de usuario del prototipo esperado se muestra en la figura 1.

5. Recursos Logísticos

La realización del taller demanda los siguientes recursos físicos:

- Aula de informática con un número de computadores equivalente al número de participantes. Es necesaria la plataforma GNU/Linux así como la instalación de ROOT (paquete root-system) en cada uno de los equipos utilizados.
- Video Beam o Pantalla de video para la presentación de información.

Referencias

- Esquembre, F. (2004). Easy java simulations: a software tool to create scientific simulations in java. *Computer Physics Communications*, 156, 199-204.
- Esquembre, F. (2005). *Creación de simulaciones interactivas en java: Aplicación a la enseñanza de la física*. Pearson educación.
- Navarro, J., Juez, J., Infante, E., y Hurtado, A. (2012). Elementos fundamentales para la evaluación objetiva de software empleado en la enseñanza de la física. *Revista Góndola*, 7(1), 66-275.
- The ROOT team. (2009). *Root an oriented-object data analysis framework, users guide 5.26*. Descargado 19 Noviembre 2011, de http://root.cern.ch/download/doc/Users_Guide_5_26_TwoInOne.pdf