

**DESARROLLO DE CLASES Y COMPONENTES REUTILIZABLES PARA EL
PROYECTO ENGINEER+ DEDICADOS A LA
TERMINAL RPN E INTERPRETER**

**ANGÉLICA YAZMIN MUÑOZ BUSTACARA
DIEGO MAURICIO RACHEN MORENO**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA, REDES Y ELECTRÓNICA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ
I-2011**

**DESARROLLO DE CLASES Y COMPONENTES REUTILIZABLES PARA EL
PROYECTO ENGINEER+ DEDICADOS A LA
TERMINAL RPN E INTERPRETER**

**ANGÉLICA YAZMIN MUÑOZ BUSTACARA
DIEGO MAURICIO RACHEN MORENO**

**Trabajo de grado
Para optar al título de
Tecnólogo en informática**

**Asesor: Carlos Armando López Solano
Director: Carlos Armando López
Solano
Ingeniero civil
Tecnólogo en Informática**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA, REDES Y ELECTRÓNICA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ
I-2011**

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Bogotá D.C. Julio de 2011

DEDICATORIA

Dedicado a Dios por darnos la oportunidad de culminar la primera fase de los estudios de educación superior, y a nuestros padres por apoyarnos incondicionalmente a lo largo del desempeño académico que hemos logrado, haciendo de nosotros personas responsables para servir a la sociedad.

AGRADECIMIENTOS ANGELICA MUÑOZ

Agradezco principalmente a Dios, porque me ha permitido alcanzar mis logros académicos, a mi madre quien me apoyo desde un principio en mi formación personal y académica, también agradezco a mi abuela quien me formo con valores en mi niñez para convertirme en quien soy ahora.

AGRADECIMIENTOS DIEGO RACHEN

Agradezco enormemente a Dios por todas las bendiciones recibidas a lo largo de mi vida académica, ya que he logrado cumplir todos mis sueños, a mi madre quien me apoyo incondicionalmente con todos los retos propuestos, a mi tía María Lucrecia Rachen ya que sin ella me hubiera sido imposible empezar mi formación académica superior y a mi abuelita quien me apoyo durante todos los años de mi vida.

AGRADECIMIENTOS GENERALES

Agradecemos al profesor Carlos Armando López, por el constante apoyo, recibido a lo largo de la carrera y la orientación prestada en la Tesis. A la universidad Minuto de Dios que nos brindó la posibilidad de estudio contando con una excelente formación

TABLA DE CONTENIDO

	Pág.
INTRODUCCIÓN	8
1.1. Título	9
1.2. Planteamiento del Problema	9
1.3 Alcances y Justificación.	10
1.4. Objetivos	11
1.4.1. Objetivo general.....	11
1.4.2. Objetivos específicos:.....	12
2. INGENIERÍA DEL PROYECTO.....	13
2.1. Metodología y Modelo de Desarrollo.....	13
2.1.2. Patrón Arquitectural.....	15
2.1.3 Comportamiento del Sistema.....	17
3. ANÁLISIS Y DISEÑO	20
3.1. Definición de Requerimientos.....	20
3.1.1. Requerimientos Funcionales	20
3.1.2. Requerimientos No Funcionales:.....	21
3.2. Descripción del sistema propuesto	22
3.3 Diseño del sistema propuesto.....	23
3.3.1 Diagramas Estáticos:.....	23
4. DESARROLLO.....	41
4.1 Especificaciones Técnicas.....	41
4.1.2. Hardware	41
5. GLOSARIO	43
6. CONCLUSIONES	45
7. BIBLIOGRAFÍA	46

LISTA DE GRÁFICOS ESPECIALES

	Pág.
Gráfica 1: Etapas de desarrollo del proyecto	15
Gráfica 2: Modelo Vista Controlador	16
Gráfica 3: Comportamiento del sistema	18
Gráfica 4: Especificaciones del proyecto Engineer+	19
Gráfica 5: Especificaciones de los requerimientos funcionales.....	21
Gráfica 6:Diagrama de clases Terminal	24
Gráfica 7: Diagrama de clases Interpreter	26
Gráfica 8: Comportamiento de la Terminal	27
Gráfica 9: Comunicación con Interpreter.....	27
Gráfica 10: Operaciones básicas.	28
Gráfica 11: Selección de un componente.	29
Gráfica 12: Captura de datos.....	30
Gráfica 13: Edición de datos.	31
Gráfica 14: Reinicio de la Terminal.	32
Gráfica 15: Presentación de resultados.	33

INTRODUCCIÓN

El desarrollo de software libre especializado en ingeniería ha tenido una evolución lenta y en el ámbito de la programación web no ha tenido la relevancia que debería tener, actualmente, se encuentra software libre para las calculadoras HP 48gx y otras herramientas como AUTOCAD o ETABS pero su licencia es costosa, y no es orientado a la web. Es por ello que desde la comunidad de software libre, Arca-csl, de UNIMINUTO, se ha planteado la necesidad de desarrollar proyectos libres orientados a la web, que involucren áreas de conocimiento como las ingenierías.

Así, Arca-csl, ha promovido el desarrollo del proyecto Engineer+, el cual consiste en la creación, desde cero, de una aplicación modular orientada a la web que permita el desarrollo de cálculos ingenieriles, en especial, cálculos en el ámbito de la ingeniería civil.

El proyecto está compuesto por varios módulos que a su vez contienen diversos componentes que en conjunto, permiten a los usuarios añadir y utilizar scripts de cálculos hechos en el lenguaje SCILAB. Sin embargo, se espera que en el futuro, a través del desarrollo evolutivo que se ha venido utilizando, se permita la creación de un lenguaje de propósito específico¹ que facilite a los estudiantes y profesionales ajenos a la programación el desarrollo de scripts en un lenguaje familiar para ellos.

El presente trabajo, tiene como objetivo presentar el desarrollo del módulo Core de Engineer+, en especial, el desarrollo de sus componentes Terminal e Interpreter, así como el diseño y desarrollo del primer prototipo que permitirá a los futuros miembros del proyecto avanzar en la evolución de la aplicación.

Para el diseño de estos componentes se han utilizado los diagramas estándar de UML² (Unified Modeling Language) con los cuales se analiza el funcionamiento e interacción entre el usuario/aplicación, y la estructura del mismo.

Los componentes Terminal e Interpreter se han desarrollado usando el lenguaje de programación PHP, acompañado de JavaScript, aprovechando la

¹Los lenguajes de propósito específico (DSL, Domain Specific Language, por sus siglas en inglés)

²Lenguaje de modelado de sistemas de software y se utiliza para visualizar, especificar, construir y documentar un sistema.

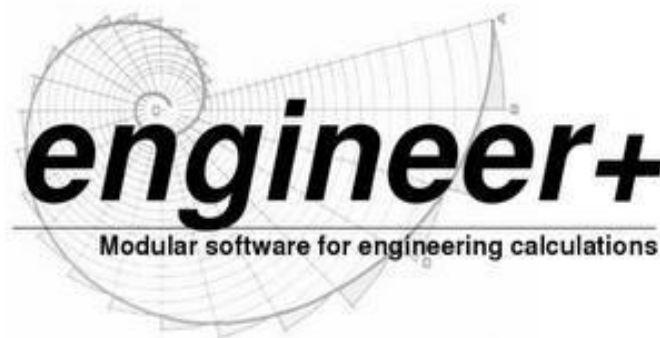
librería de clases MApache³ desarrollada por la comunidad Arca-csl. Adicionalmente, el prototipo desarrollado incluye el uso de bases de datos en MySQL para la administración de usuarios del sistema.

Como modelo de desarrollo se usara UP⁴ debido a que durante las fases de dicha aplicación es necesario ejecutar pruebas para evitar errores ya que la Terminal está pensada de manera evolutiva, al igual se utiliza el patrón modelo-vista-controlador, debido a que la aplicación es orientada a la Web.

A continuación se realiza una descripción más profunda de la funcionabilidad de los componentes, su relación con el módulo Core del proyecto Engineer+ y de los elementos que los conforman.

1.1. Título

El título del proyecto que fue definido para describir la idea principal del proyecto fue el siguiente:



DESARROLLO DE CLASES Y COMPONENTES REUTILIZABLES PARA EL
PROYECTO ENGINEER+ DEDICADOS A LA
TERMINAL RPN – INTERPRETER

1.2. Planteamiento del Problema

El desarrollo de aplicaciones libres especializadas orientadas a la web, en áreas de ingeniería es escaso, y este tipo de aplicativos son de gran apoyo para la formación y desempeño de las personas involucradas en dicha área,

³MApache es un proyecto de la comunidad de software libre Arca-csl de UNIMINUTO que permite la creación rápida de interfaces gráficas en html.

⁴UP hace referencia a una metodología evolutiva mayormente conocida como RUP (Rational Unified Process) de IBM.

debido a que facilitan el desarrollo de cálculos ingenieriles y disminuyen los procesos requeridos para la solución de un ejercicio o problema. El uso de estas herramientas actualmente es poco conocido, su manejo se debe estandarizar y conceptualizar para la utilización de ingenieros y/o estudiantes.

Adicionalmente, existen aplicaciones privativas cuyas licencias son de alto costo para las empresas acentuando los problemas de piratería ya existentes en la venta legal de software, y disminuyendo la utilización y conocimiento de las mismas. En el caso del software libre, estos inconvenientes son menores, lo cual permite una mayor distribución y utilización de estas herramientas especializadas. Por otro lado, actualmente no es común recolectar y distribuir el conocimiento entre ingenieros, impidiendo que ellos puedan participar en el desarrollo de cálculos y que estos, puedan ser usados por todos aquellos que lo requieran, permitiendo una constante actualización del software a través de mejoras y aumento del funcionamiento.

Pensando en esto, se decide desarrollar un aplicativo, el cual tendrá características que dependerán de las necesidades de los usuarios, es decir, que puede ser complementado y modificado por ellos. Sin embargo, aún no se cuenta con los diseños de un prototipo que permita presentar una solución viable.

1.3 Alcances y Justificación.

El proyecto Engineer+ busca generar una herramienta libre orientada a la web, que permita la adquisición y distribución de conocimiento entre ingenieros, facilitando el proceso de desarrollo de soluciones de cálculo, caracterizándolo como una aplicación flexible y escalable. Engineer+ busca crear una red social donde los ingenieros puedan participar en la creación y/o modificación de funcionalidades del sistema.

En general, la herramienta hará posible la solución de diferentes problemas en áreas ingenieriles que involucran procesos matemáticos, físicos, de topografía y geometría inicialmente, su interfaz gráfica tendrá una apariencia similar al de las pilas de datos de las calculadoras ingenieriles. El aplicativo está diseñado de manera evolutiva, por lo cual, el presente proyecto está encargado del prototipo inicial de dos de los principales componentes, los cuales son Terminal e Interpreter.

La Terminal es la presentación gráfica del software, se encarga de recibir las solicitudes dadas por el usuario y presentar los resultados. El usuario puede realizar varias solicitudes como: seleccionar un componente, enviar datos o las operaciones requeridas, pedir información sobre el componente, decidir cambios sobre sus datos ingresados y cambiar la presentación de la interfaz según su propio gusto.

Interpreter es un componente⁵ lógico, que permite la comunicación entre ellos, es decir, es el encargado de enviar las solicitudes requeridas desde la Terminal hacia los otros componentes, éste mismo recibe la respuesta y la envía hacia la Terminal quien notificará el resultado al usuario. Interpreter debe ser adaptable para la incorporación de componentes posteriores, esto permitirá que la aplicación sea flexible, por lo cual su funcionalidad será bastante amplia y en menos tiempo que un aplicativo normal.

Terminal e Interpreter, se presentarán en forma de prototipo que presenta una apariencia similar al de las pilas de datos de las calculadoras ingenieriles y con un soporte de comandos con los que es posible realizar algunas operaciones básicas, enfocando el proyecto hacia el ámbito investigativo y de diseño.

Este proyecto no incluye el mantenimiento posterior a la puesta en marcha del sistema. Esta limitación se debe a la poca disponibilidad de tiempo que ofrece el periodo académico.

1.4. Objetivos

1.4.1. Objetivo general

Generar una herramienta libre especializada, orientada a la web, que permita hacer cálculos ingenieriles, con un alto grado de flexibilidad. Adicionalmente, desarrollar los componentes Terminal e Interpreter para el proyecto Engineer+ con el fin de permitir la transmisión de solicitudes de los usuarios a los demás componentes del proyecto.

⁵Es una unidad ejecutable que puede ser implantada independientemente, es decir, es una pieza de software que puede ser reutilizada.

1.4.2. Objetivos específicos:

Los objetivos específicos de la Terminal e Interpreter son los siguientes:

- Permitir el acceso de usuarios de forma controlada.
- Facilitar el manejo de la aplicación a través de una interfaz gráfica basada en la notación polaca inversa (RPN).
- Suministrar la interacción y comunicación entre los distintos módulos del proyecto a través de Interpreter.
- Aportar cálculos matemáticos básicos (Operaciones básicas aritméticas, de ordenamiento y de solicitud de servicios a Interpreter), en el componente Terminal.

2. INGENIERÍA DEL PROYECTO

Esta sección provee una completa descripción del diseño del proyecto de software. Aquí se describen todos los datos, arquitecturas e interfaces y sus componentes a nivel de diseño.

2.1. Metodología y Modelo de Desarrollo

Para Engineer+ el modelo que se utiliza es UP (Proceso Unificado), ya que es un marco de desarrollo de software que se caracteriza por estar centrado en la arquitectura y por ser iterativo e incremental, esto es bastante útil para el proyecto, ya que es una aplicación evolutiva y su desarrollo debe evitar al máximo errores funcionales, para continuar con otra fase de implementación.

Existen cuatro fases en UP denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones, estas iteraciones ofrecen como resultado un incremento del proyecto que añade o mejora las funcionalidades del sistema en desarrollo.

En la etapa inicial del proyecto, se establecen las necesidades que podrían tener los usuarios de la aplicación con lo cual se generan los requerimientos del sistema, la manera adecuada de presentación y se define la utilización de software libre para suplir una necesidad actual de los ingenieros y estudiantes de ingeniería. Además, se define la utilización de comandos, su comportamiento RPN y la manera de presentación hacia los usuarios.

En la etapa de Elaboración, se determinaron los siguientes ítems:

- Se elaboran los esquemas generales que muestran el comportamiento general del sistema, observando la interacción entre los componentes Terminal e Interpreter.
- Se elaboran los diagramas UML para definir las clases, atributos, métodos, distribución, relaciones y funcionalidad que debe tener la aplicación.

- Se define la presentación de la Terminal, el uso de comandos, captura y presentación de datos basado en Notación Polaca Inversa⁶, y su funcionalidad general, diseñando la interfaz apropiada y adaptable según lo requiera el usuario.
- Se define el registro del usuario, para controlar el acceso al aplicativo.

En la etapa de Construcción se llevó a cabo lo siguiente:

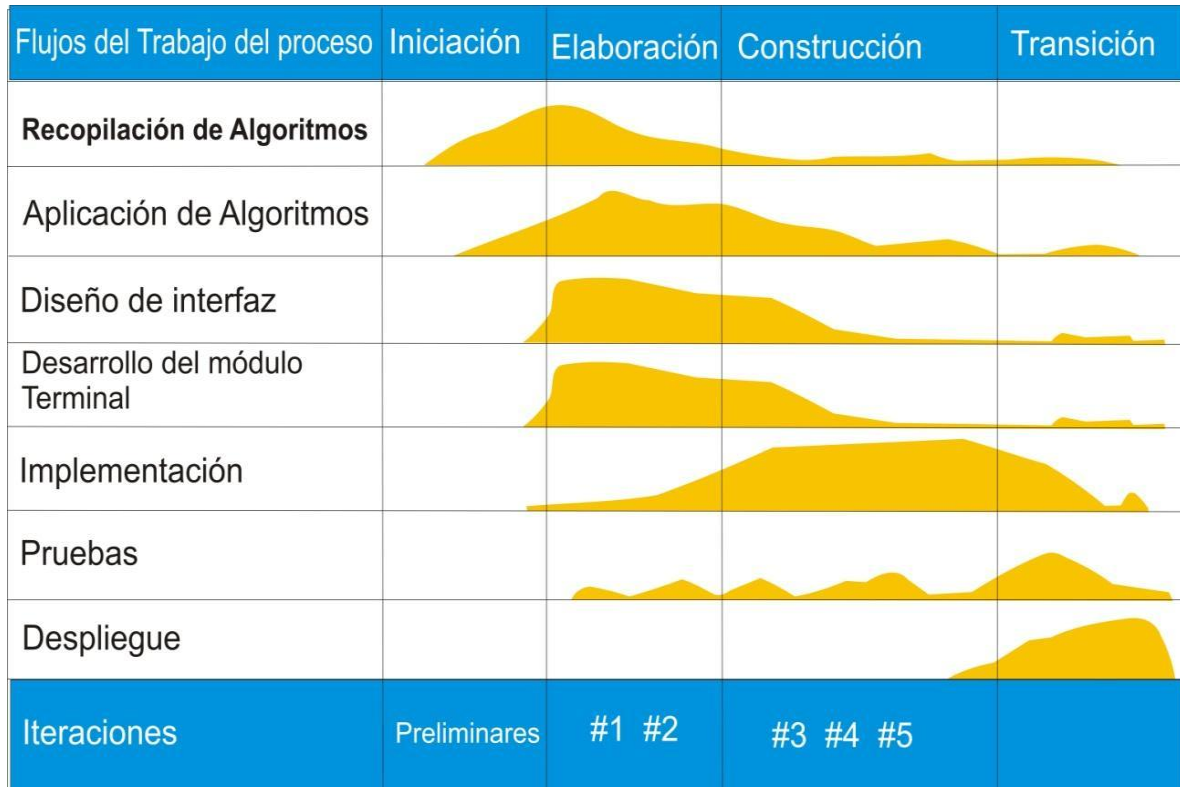
- Se implementa el componente Terminal, basado en librerías del proyecto MApache y se define su presentación.
- Se desarrollan los principales scripts para observar el funcionamiento de la Terminal, realizando las pruebas de errores necesarias para seguir en el proceso de desarrollo.
- Se realizan las pruebas correspondientes sobre el componente, para comprobar su reutilización.
- Se implementará Interpreter, se realiza la conexión entre componentes, por medio de web services y posteriormente se observará el comportamiento de la comunicación entre los componentes, realizando comprobación sobre tiempos de respuesta y flexibilidad en la adición de scripts.
- Se desarrollará la conexión con SCILAB y se realizarán las pruebas de errores para observar su funcionalidad y la de Interpreter.

Durante la etapa de transición, se aumentarán los scripts en SCILAB, para generar más funcionalidades al sistema, siguiendo los mismos estándares, a través de pruebas de errores de la aplicación y en términos de tiempos de respuesta, observando la correcta funcionalidad para evitar falencias.

En el siguiente gráfico se mostrará las etapas de evolución en la implementación del proyecto:

⁶Reverse Polish Notation por sus siglas en inglés, es un método algebraico alternativo de introducción de datos.

Etapas de desarrollo del proyecto.

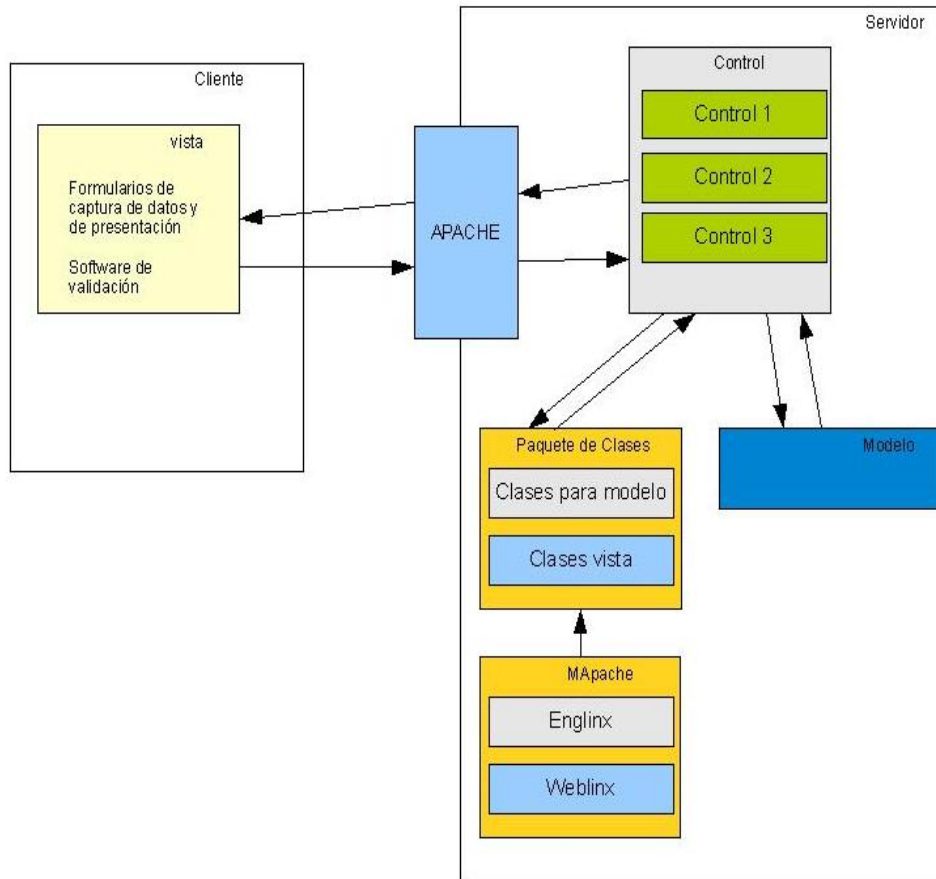


Gráfica 1: Etapas de desarrollo del proyecto
Fuente: Los autores

2.1.2. Patrón Arquitectural

El patrón arquitectural que se usará en el desarrollo de Engineer+ es el Modelo Vista Controlador (MVC - ver gráfico 2), del cual se separan tres componentes distintos, la interfaz de usuario, la lógica de control y los datos de la aplicación, ya que es un software orientado 100% a la Web. En el siguiente esquema se visualizará el patrón arquitectural del proyecto.

Modelo Vista Controlador



Gráfica 2: Modelo Vista Controlador
Fuente: <http://sites.google.com/site/englinx/>

Convenciones:

A continuación se mostrarán las especificaciones del gráfico 2: Modelo Vista Controlador:

Cliente: Es el usuario que interactúa con el software.

Vista: Formularios de presentación.

Apache: servidor Web donde se alojará la aplicación y es el encargado de realizar las operaciones indicadas en los scripts.

Control: Encargado de gestionar el comportamiento de la aplicación.

Paquete de clases: Donde se alojarán las clases tanto para el modelo como para la vista.

MApache: paquete de librerías usadas en el desarrollo.

Modelo: Sistema de gestión y lógica del proyecto.

2.1.3 Comportamiento del Sistema

A continuación veremos una descripción general del funcionamiento del proyecto Engineer +:

- Comportamiento del sistema:

En la gráfica 3 se muestra el procedimiento general del proyecto, la Terminal, quien en primera instancia interactúa con el usuario, recibe las órdenes y/o comandos, Interpreter analiza lo solicitado y elige el componente que soluciona dicha solicitud, a quien envía los datos requeridos y este mismo componente regresa una respuesta al usuario, los gráficos en rojo, muestran los componentes internos de Engineer+, quienes a través del tiempo irán incrementando.

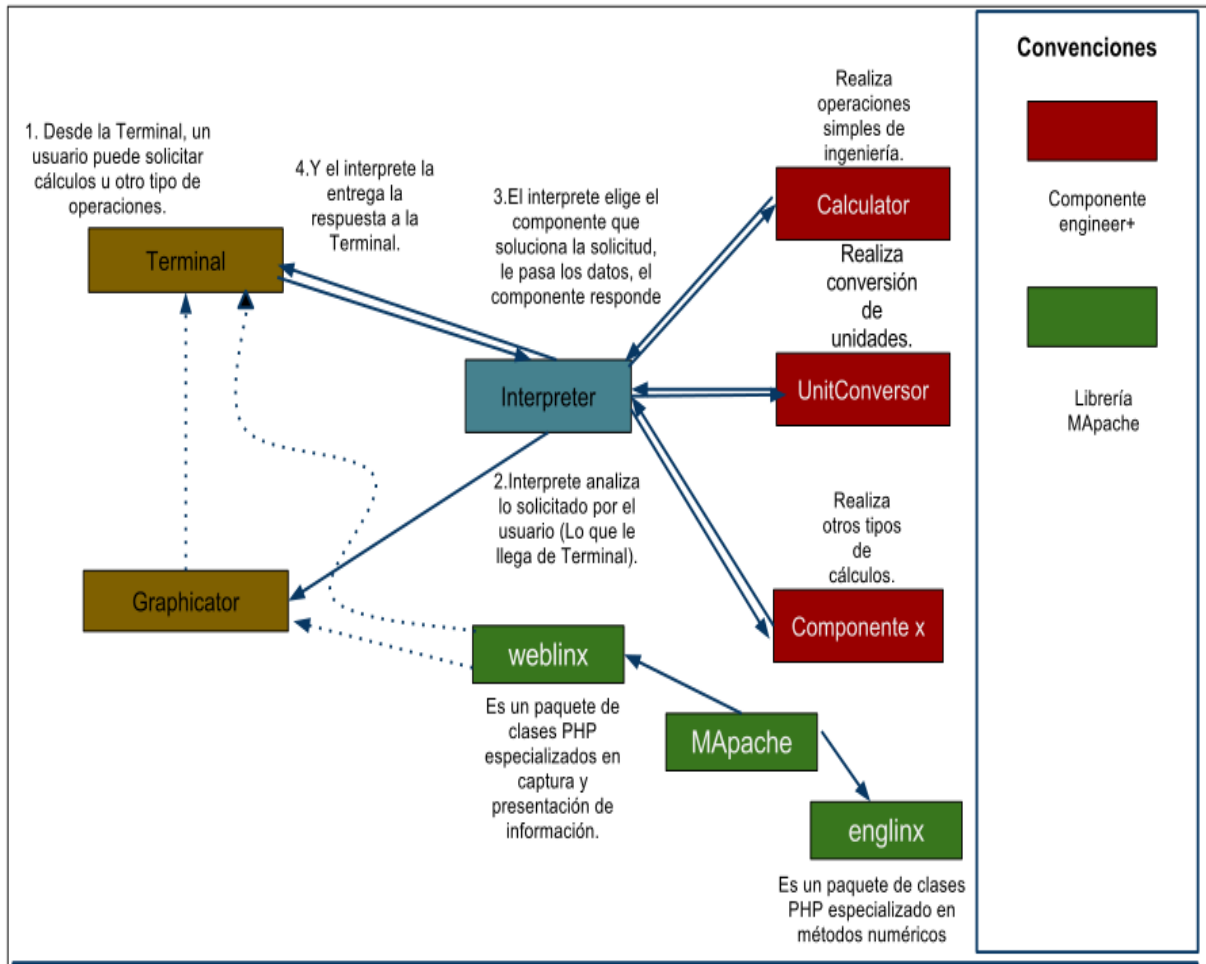
- Especificación del proyecto Engineer+:

En el gráfico 4 se visualiza los diferentes componentes que hacen parte de Engineer+ con sus Tópicos⁷ y clases correspondientes a la distribución del proyecto general.

A continuación se observarán los gráficos mencionados anteriormente:

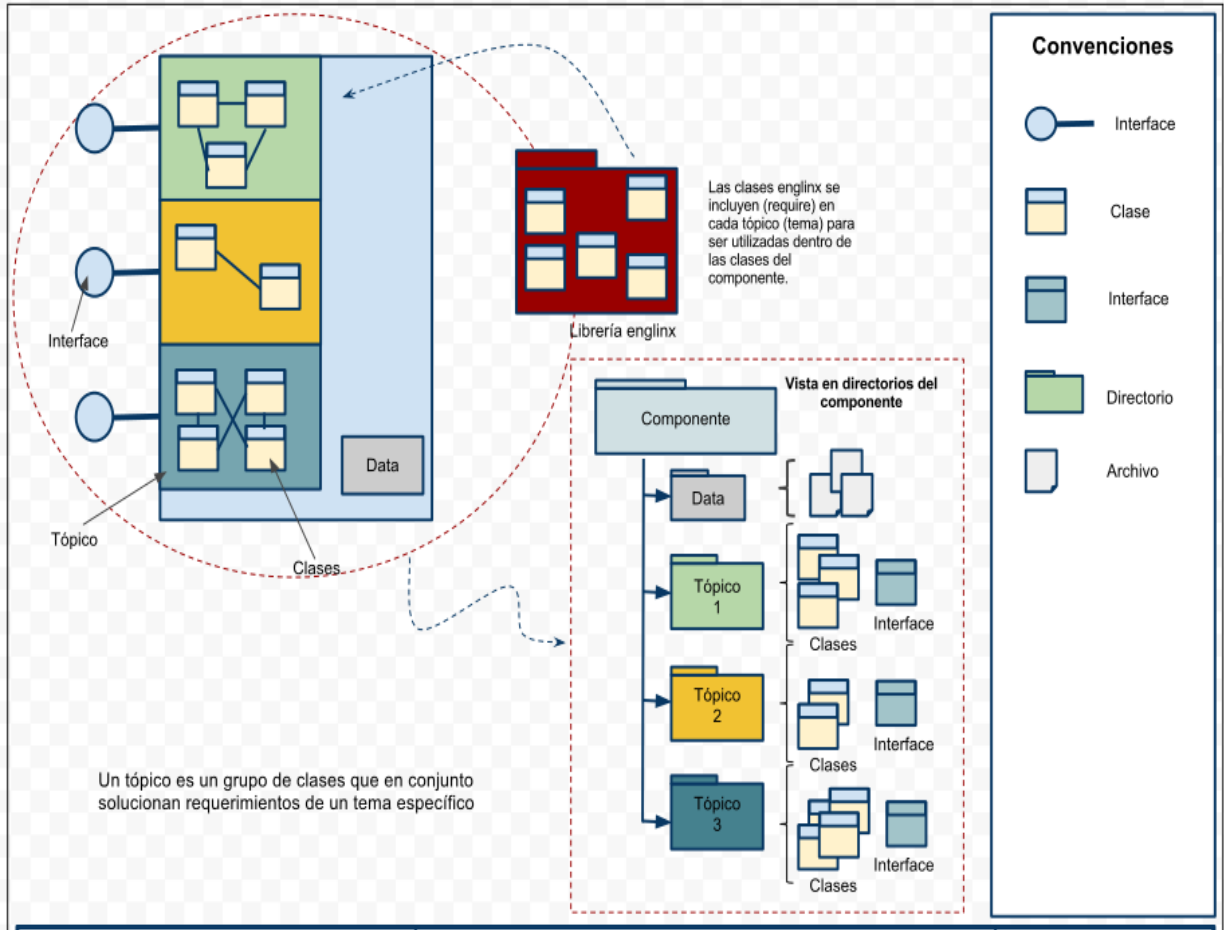
⁷Grupo de clases que en conjunto solucionan requerimientos de un tema específico.

Comportamiento del sistema



Gráfica 3: Comportamiento del sistema
Fuente: Los autores.

Especificaciones del proyecto Engineer+



Gráfica 4: Especificaciones del proyecto Engineer+
 Fuente: <http://sites.google.com/site/englinx/Home/diseño>

3. ANÁLISIS Y DISEÑO

El análisis y diseño de un software es un proceso en el cual se define la estructura del aplicativo y su forma de desarrollo, la fase de análisis del proyecto está compuesto por las siguientes especificaciones:

3.1. Definición de Requerimientos

A continuación se describen los requerimientos funcionales y no funcionales para el desarrollo de la Terminal e Interpreter del proyecto Engineer+.

3.1.1. Requerimientos Funcionales

Son aquellos que permiten determinar la funcionalidad del sistema. Para el presente proyecto se fijaron los siguientes:

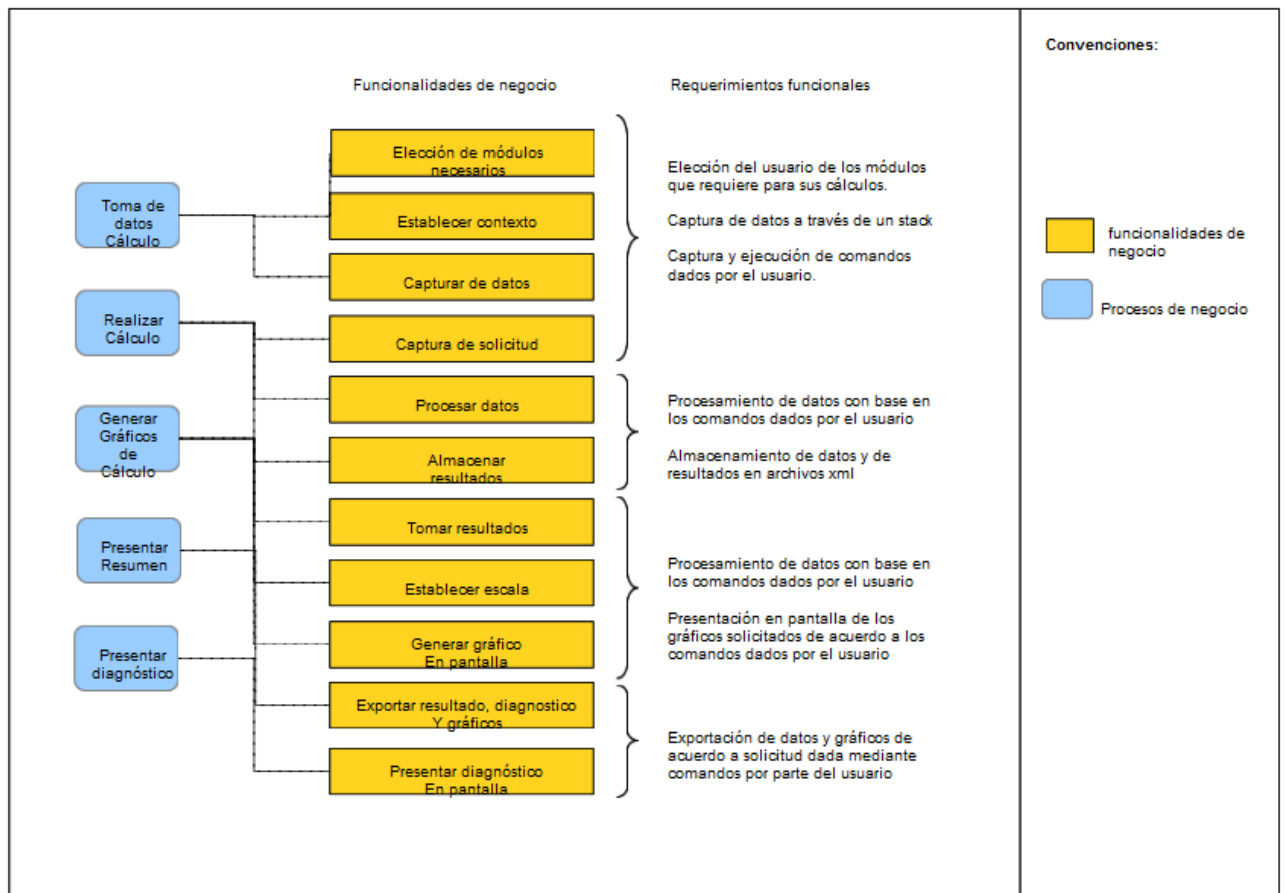
1. La Terminal utilizará la notación polaca inversa (RPN) para facilitar la captura de grandes volúmenes de datos.
2. La Terminal deberá transmitir las solicitudes del usuario (Que se hacen a través de comandos) al componente Interpreter.
3. La Terminal tendrá las funciones matemáticas básicas para el uso de los usuarios.
4. La Terminal debe permitir al usuario seleccionar los componentes en los cuales desea trabajar.
5. La Terminal admitirá la captura de datos y los guardará en un Stack⁸.
6. La Terminal permitirá la edición de datos almacenados en el Stack.
7. La Terminal posibilitará el reinicio del Stack a un estado inicial a través de una instrucción dada por el usuario.

⁸ Pila de datos ordenada.

8. La Terminal presentará los distintos resultados de las operaciones solicitadas por el usuario.

9. Interpreter envía solicitudes para la comunicación entre la Terminal y diferentes módulos del proyecto actuales o posteriores, de acuerdo a las solicitudes hechas por el usuario.

A continuación se observa gráficamente la estructura de los requerimientos del sistema.



Gráfica 5: Especificaciones de los requerimientos funcionales

Fuente: <http://sites.google.com/site/englinx/Home/disenio>

3.1.2. Requerimientos No Funcionales:

1. El usuario deberá estar on-line para la correcta funcionalidad del sistema
2. La aplicación debe estar montada en un servidor web.

3. La aplicación debe tener tiempos de respuesta menores a 5 segundos
4. La aplicación manejará un sistema de roles para los usuarios.
5. La aplicación deberá permitir el acceso a la ayuda de comandos.

3.2. Descripción del sistema propuesto

Engineer+ es una aplicación orientada a la web, distribuida, que permite el desarrollo de cálculos ingenieriles y además permite la integración de métodos numéricos de quienes así lo requieran, sigue el patrón Modelo Vista Controlador (MVC), su desarrollo hace uso del paradigma de programación orientada a componentes, lenguajes de desarrollo como PHP (usando el grupo de librerías MApache de Arca-csl), javascript y SCILAB. El sistema será completamente libre y con alto grado de flexibilidad para el crecimiento del mismo. Los ingenieros tendrán la posibilidad de aumentar su funcionalidad dependiendo de sus propios requerimientos. El proyecto está dividido en componentes los cuales cumplen funciones específicas, para llevar a cabo el objetivo general del sistema. El presente proyecto está encargado del prototipo de dos componentes: Terminal e Interpreter.

Terminal es la interfaz gráfica del proyecto, la cual comunica al usuario con el sistema. La presentación y captura de datos está basada en la Notación Polaca Inversa (RPN), la cual facilita la obtención de grandes cantidades de información de forma sencilla y rápida. Las operaciones se realizan a través de comandos enviados por el usuario y los tiempos de respuesta serán óptimos. Además, presentará datos solicitados como ayuda de comandos y desarrolladores de scripts, según lo desee el usuario.

Interpreter está encargado de comunicar los componentes del sistema con la Terminal, tramitando las solicitudes enviadas por el usuario, para esto se utiliza web services, lo cual permite una arquitectura cliente-servidor. Web services es una pieza de software que utiliza un conjunto de protocolos estándares que sirven para intercambiar datos entre aplicaciones. Con esto se logrará realizar un sistema distribuido para permitir el posterior avance del proyecto Engineer+.

En general, el prototipo del sistema permite la realización de cálculos ingenieriles, inicialmente básicos como operaciones matemáticas y cálculos en SCILAB principalmente, se presentará la funcionalidad de Engineer+, con el objetivo de mostrar su uso en el área académica y el alcance que puede lograr

el proyecto a través del tiempo, debido a que es un sistema distribuido y los componentes registrados posteriormente pueden ser usados por los usuarios del Engineer+, el cual posteriormente será una red social para las personas interesadas como ingenieros y estudiantes de dicha área.

3.3 Diseño del sistema propuesto

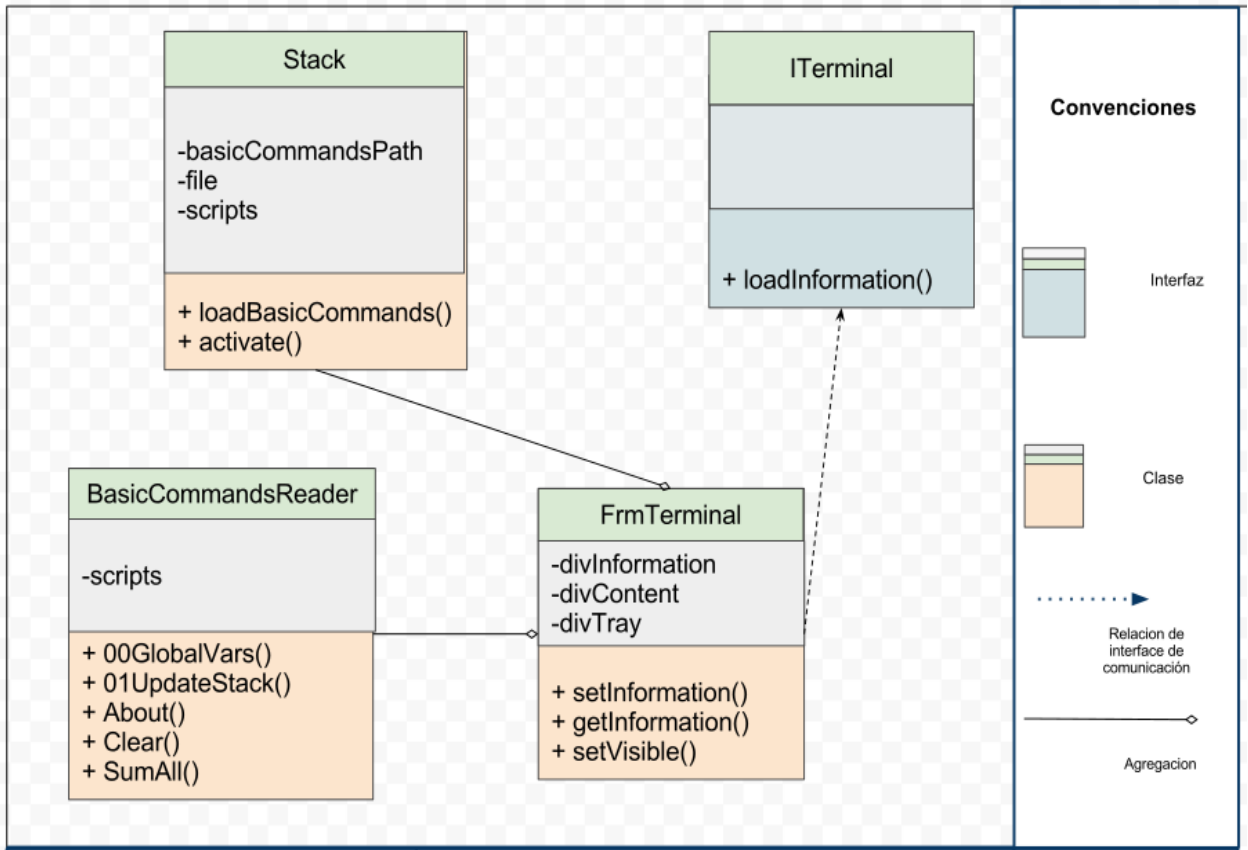
El diseño de un sistema se refiere a la definición y estructura de los requerimientos después de un respectivo análisis, durante este proceso se utiliza UML (Lenguaje unificado de Modelado), ya que es el lenguaje de modelado de sistemas que ofrece un estándar para describirlo, incluyendo aspectos conceptuales tales como procesos de negocio y funciones de la aplicación. Siguiendo esta estructura en Engineer+ se implementan los tipos de diagramas estáticos y dinámicos que se explicarán a continuación.

3.3.1 Diagramas Estáticos:

Se encargan de presentar qué componentes (entidades, objetos, áreas, clases, etc.) se van a implementar en cada uno de los aspectos del sistema a desarrollar. También se encargan de especificar cómo deben de estar estructurados estos elementos. Para tal efecto se determinaron los siguientes diagramas:

- Diagrama de clases

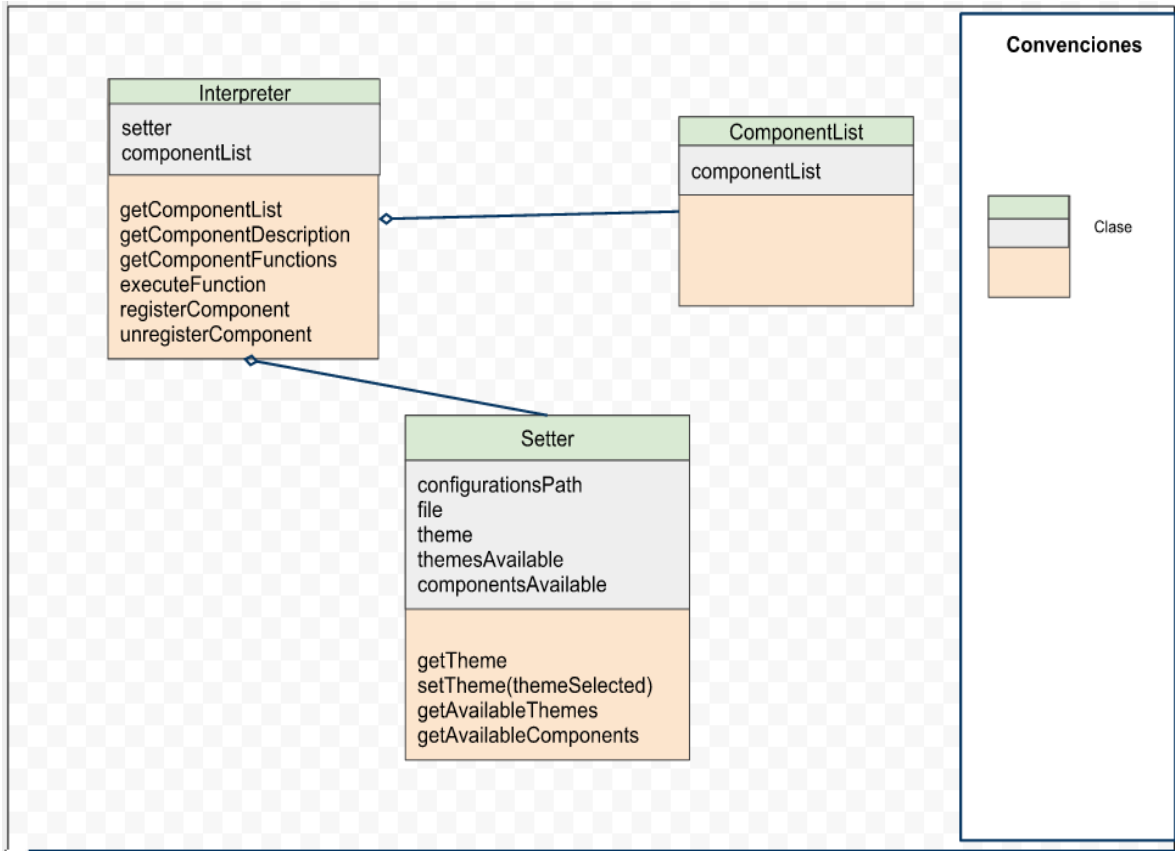
En los siguientes diagramas se encontrará la distribución de las clases de los componentes Terminal e Interpreter respectivamente, describen la estructura del sistema presentando sus atributos, relaciones y el funcionamiento del sistema.



Gráfica 6: Diagrama de clases Terminal
Fuente: Los autores.

Clases	Variables	Métodos
Stack	basicCommandsPath - > almacena una orden enviada por el usuario. dirList -> array en donde se almacenan los datos ingresados por el usuario scripts -> variable que almacena el script que se va a ejecutar.	loadBasicCommands() -> se encarga de cargar los comandos almacenados en Terminal. actívate() -> ejecuta el script solicitado y lo imprime en Terminal
ITerminal		loadInformation() -> carga los datos requeridos.

BasicCommandsReader	Scripts -> Es el encargado de leer todos los archivos JS (javascript) los cuales están contenidos en el paquete Data del componente Terminal	GlobalVars() -> array en el cual se cargan los resultados de las operaciones básicas. About() -> Información sobre el desarrollo de Engineer+ Clear() -> Reinicia la Terminal. Sort() -> ordenamiento de los datos almacenados en el Stack.
FrmTerminal	divInformation -> cuadro de texto encargado de presentar la información del usuario y fecha divContent -> cuadro de texto encargado de presentar los datos almacenados en el Stack. divTray -> cuadro de texto en el cual se muestran los resultados de las operaciones	setVisible() -> muestra los datos actuales de la Terminal

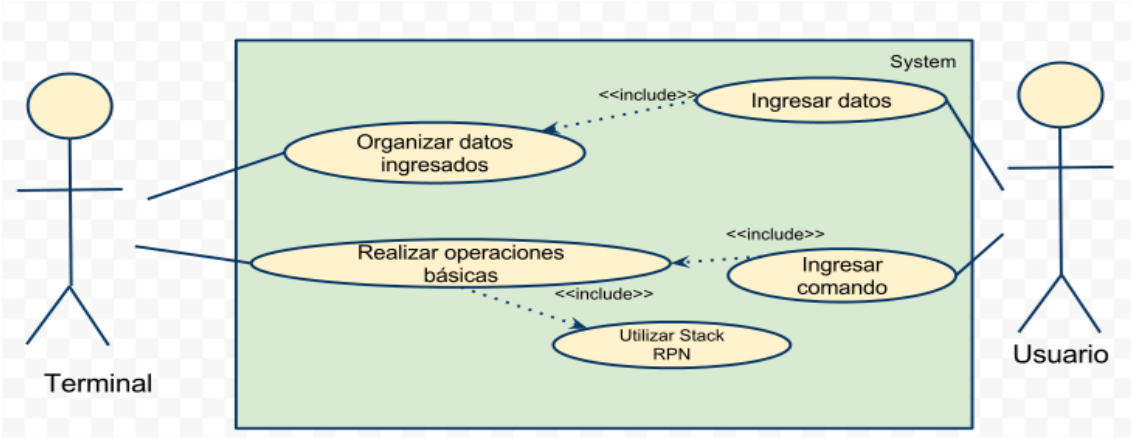


Gráfica 7: Diagrama de clases Interpreter
Fuente: Los autores.

Diagramas de casos de uso.

- Son diagramas que presentan el comportamiento e interacción entre los actores del sistema.

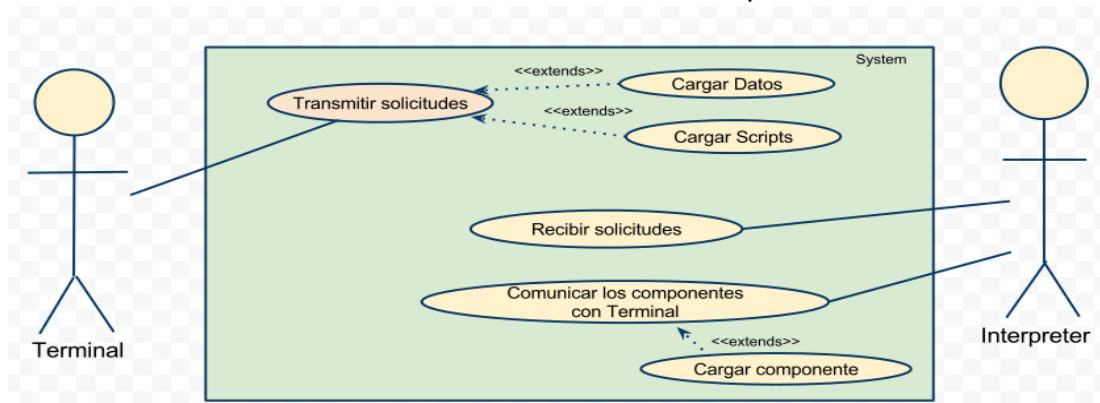
Caso de uso 1: Comportamiento de la Terminal



Gráfica 8: Comportamiento de la Terminal
Fuente: Los autores.

Nombre	Requerimiento 1: Uso de la notación polaca inversa	
Descripción	La Terminal utilizará la notación polaca inversa (RPN) para facilitar la captura de grandes volúmenes de datos.	
Entradas	Valores numéricos ingresados a la pila.	
Resultados	Organización de los datos en forma de stack	
Actores:	Terminal - Usuario	
Precondiciones:	1.Ingreso de datos numéricos a la Terminal	
Flujo Normal	Actor	Sistema
	1.Insercion de datos	2.Organizacion de datos ingresados 3.Comportamiento de Pila (stack –RPN)
Flujo Alternativo:		

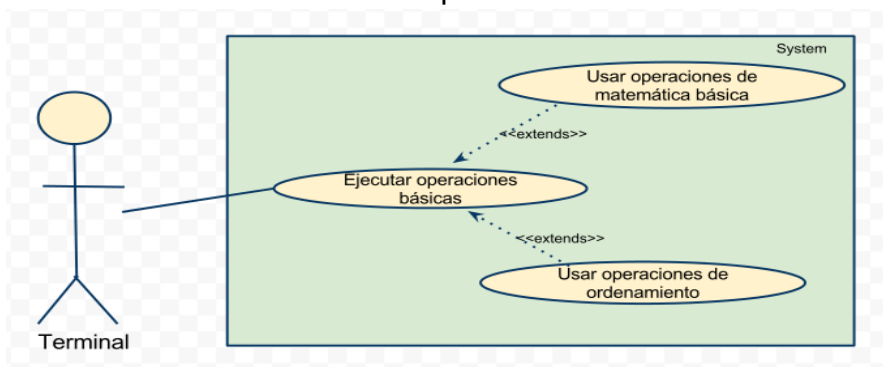
Caso 2: Comunicación con Interpreter



Gráfica 9: Comunicación con Interpreter
Fuente: Los autores.

Nombre	Requerimiento 2: Comunicación con el Componente Interpreter	
Descripción	La Terminal deberá transmitir las solicitudes del usuario (Que se hacen a través de comandos) al componente Interpreter.	
Entradas	Comandos que serán transmitidos como datos u órdenes.	
Resultados	Recepción de comandos por parte del componente Interpreter y la posterior comunicación entre los distintos módulos permitiendo la carga del componente	
Actores:	Terminal , Interpreter	
Precondiciones:	1.Ingreso de datos y ordenes a la Terminal	
Flujo Normal	Actor	Sistema
	1.Ingreso de datos y ordenes a través de la Terminal 2.Transmite las solicitudes	3.Recepcion de solicitudes 4.Comunicacion entre módulos 5.Carga el componente seleccionado
Flujo Alternativo:	1.Usuario no ingresa datos y ordenes a la Terminal 2.Usuario ingresa un dato que no existe dentro del sistema	
Pos condiciones:	Sistema registra los datos y órdenes que se le ingresaron y el componente Interpreter procesa la solicitud cargando el módulo requerido	

Caso de uso 3: Operaciones básicas

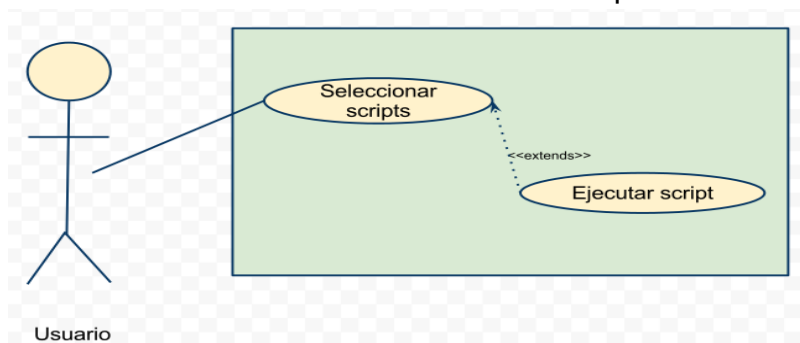


Gráfica 10: Operaciones básicas.

Fuente: Los autores.

Nombre	Requerimiento 3: Funciones Básicas	
Descripción	La Terminal tendrá las funciones matemáticas básicas para el uso de los usuarios.	
Entradas	Valores numéricos ingresados al Stack Órdenes o comandos para el procesamiento de las operaciones.	
Resultados	Valores finales de las operaciones requeridas.	
Actores:	Terminal	
Precondiciones:	1.Ingreso de datos numéricos a la Terminal 2. Ingreso de la instrucción de la operación seleccionada.	
Flujo Normal	Actor	Sistema
	1. Ingreso de datos numéricos a la Terminal 2. Ingreso de la instrucción de la operación.	3. Sistema presenta el resultado de la operación que se seleccionó.
Flujo Alternativo:	1. Usuario no ingresa datos a la Terminal 2. Usuario no ingresa el correspondiente comando de la operación.	
Pos condiciones:	Sistema registra los datos que se le ingresaron , luego procesa la instrucción de la operación seleccionada y devuelve el valor correspondiente	

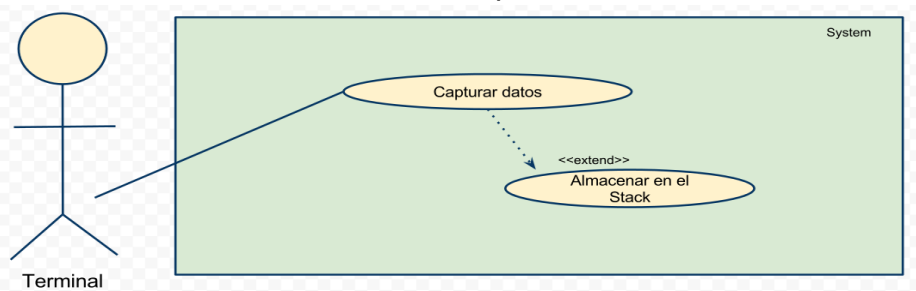
Caso de uso 4: Selección de un componente



Gráfica 11: Selección de un componente.

Nombre	Requerimiento 4: Selección de un componente.	
Descripción	La Terminal debe permitir al usuario seleccionar los componentes en los cuales desea trabajar.	
Entradas		
Instrucción que permite la selección del script a ejecutar.		
Resultados		
Sistema carga el script solicitado.		
Actores:		
Usuario		
Precondiciones:		
1.Ingreso de la instrucción a la Terminal		
Flujo Normal		
	Actor	Sistema
1. Ingreso de la instrucción a la Terminal		2. Sistema carga el script
Flujo Alternativo:		
1. Usuario no ingresa la instrucción correcta		
Pos condiciones:		
Sistema registra los datos que se le ingresaron, luego procesa la instrucción ingresada y posteriormente carga el componente.		

Caso de uso 5: Captura de datos

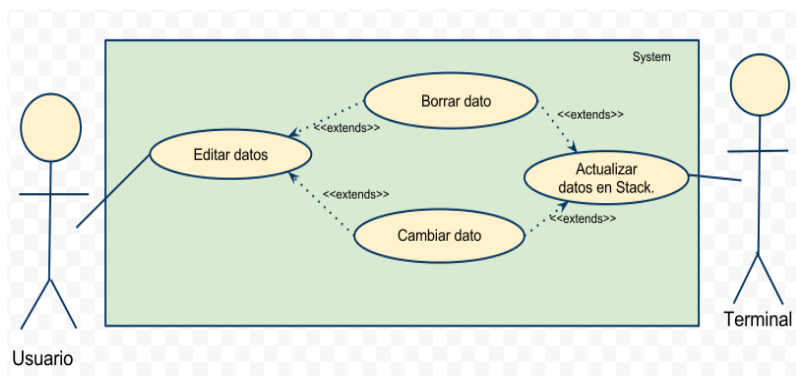


Gráfica 12: Captura de datos.

Fuente: Los autores.

Nombre	Requerimiento 5: Captura de datos y posterior almacenamiento	
Descripción	La Terminal admitirá la captura de datos y los guardará en un stack.	
Entradas		
Valores numéricos ingresados por medio de la Terminal		
Resultados		
Almacenamiento de los datos en un stack		
Actores:		
Terminal		
Precondiciones:		
1.Ingreso de datos numéricos a la terminal		
Flujo Normal		
	Actor	Sistema
	1.Incersion de datos	2.Almacenamiento de los datos ingresados en un stack
Flujo Alternativo:		
1.Usuario no ingresa datos a la terminal		
Pos condiciones:		
Sistema registra los datos que se le ingresaron y posteriormente almacena los valores ingresados en un stack		

Caso de uso 6: Edición de datos

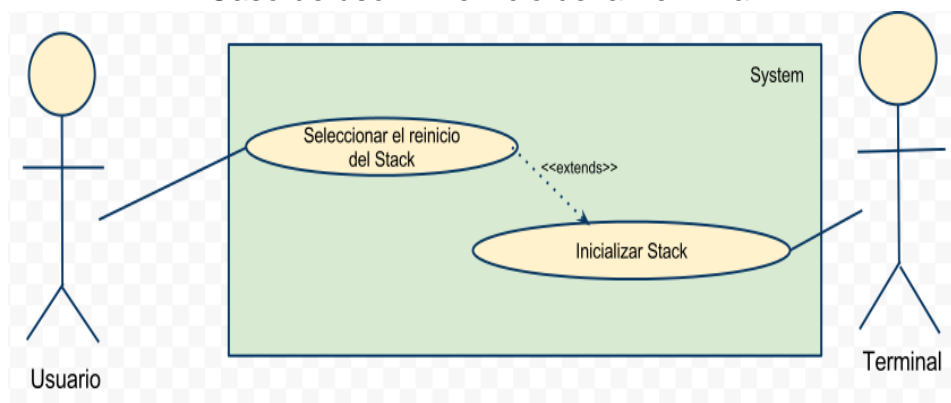


Gráfica 13: Edición de datos.

Fuente: Los autores.

Nombre	Requerimiento 6: Edición de datos en el stack	
Descripción	La Terminal permitirá la edición de datos almacenados en el stack	
Entradas	Valores numéricos ingresados a través de la Terminal	
Resultados	Edición de cualquier dato almacenado en el stack	
Actores:	Usuario , Terminal	
Precondiciones:	1.Ingreso de datos numéricos a la Terminal	
Flujo Normal	Actor	Sistema
	1.Incersion de datos	2.Almacenamiento de los datos ingresados en un stack 3.Edicion de cualquier dato almacenado
Flujo Alternativo:	1.Usuario no ingresa datos a la terminal 2.Sistema no almacene los datos en el stack	
Pos condiciones:	Sistema registra los datos que se le ingresaron y posteriormente los almacene en un stack y luego permita la edición de los datos contenidos	

Caso de uso 7: Reinicio de la Terminal.

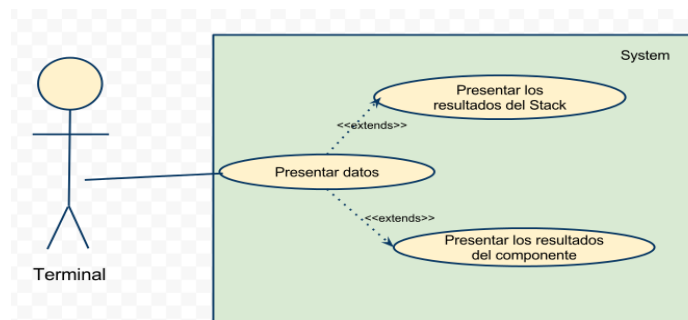


Gráfica 14: Reinicio de la Terminal.

Fuente: Los autores.

Nombre	Requerimiento 7: Reinicio de la Terminal	
Descripción	La Terminal posibilitará el reinicio del Stack a un estado inicial a través de una instrucción dada por el usuario.	
Entradas	Valores numéricos ingresados al Stack. Instrucción que permite el reinicio	
Resultados	Stack vuelve a su estado inicial donde no contiene ningún dato.	
Actores:	Usuario , Terminal	
Precondiciones:	1. Ingreso de datos numéricos a la Terminal.	
Flujo Normal	Actor	Sistema
	1. Ingreso de datos numéricos a la Terminal 2. Ingreso de la instrucción de reinicio	3.Sistema vuelve a su estado inicial
Flujo Alternativo:	1. Usuario no ingresa el correspondiente comando de reinicio	
Pos condiciones:	Sistema registra los datos que se le ingresaron, luego procesa la instrucción de reinicio y vuelve a su estado inicial.	

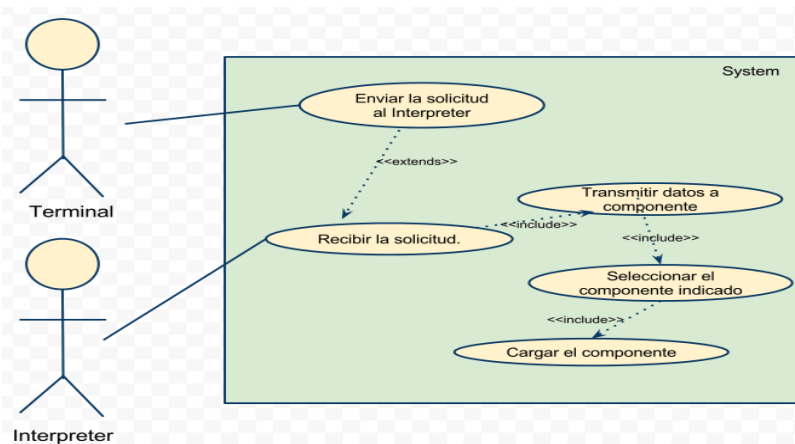
Caso de uso 8: Presentación de resultados.



Gráfica 15: Presentación de resultados.

Nombre	Requerimiento 8: Presentación de resultados	
Descripción	La Terminal presentará los distintos resultados de las operaciones solicitadas por el usuario.	
Entradas		
Valores numéricos ingresados a la pila		
Instrucción que permite seleccionar la operación o script correspondiente.		
Resultados		
Terminal presentara los resultados de las operaciones básicas o el script ejecutado desde un componente diferente.		
Actores:		
Terminal		
Precondiciones:		
1.Ingreso de datos numéricos a la Terminal		
2. Ingreso de la instrucción que permitirá hacer la operación seleccionada		
Flujo Normal		
Actor	Sistema	
1. Ingreso de datos numéricos a la Terminal	3. Sistema presenta el resultado de la operación seleccionada	
2. Ingreso de la instrucción que permitirá hacer la operación		
Flujo Alternativo:		
1. Usuario no ingresa datos a la Terminal.		
2. Usuario no ingresa el correspondiente comando de selección de la operación.		
Pos condiciones:		
Sistema presenta en pantalla los distintos resultados de las operaciones que el usuario seleccionó.		

Caso de uso 9: Trasmisión de solicitudes

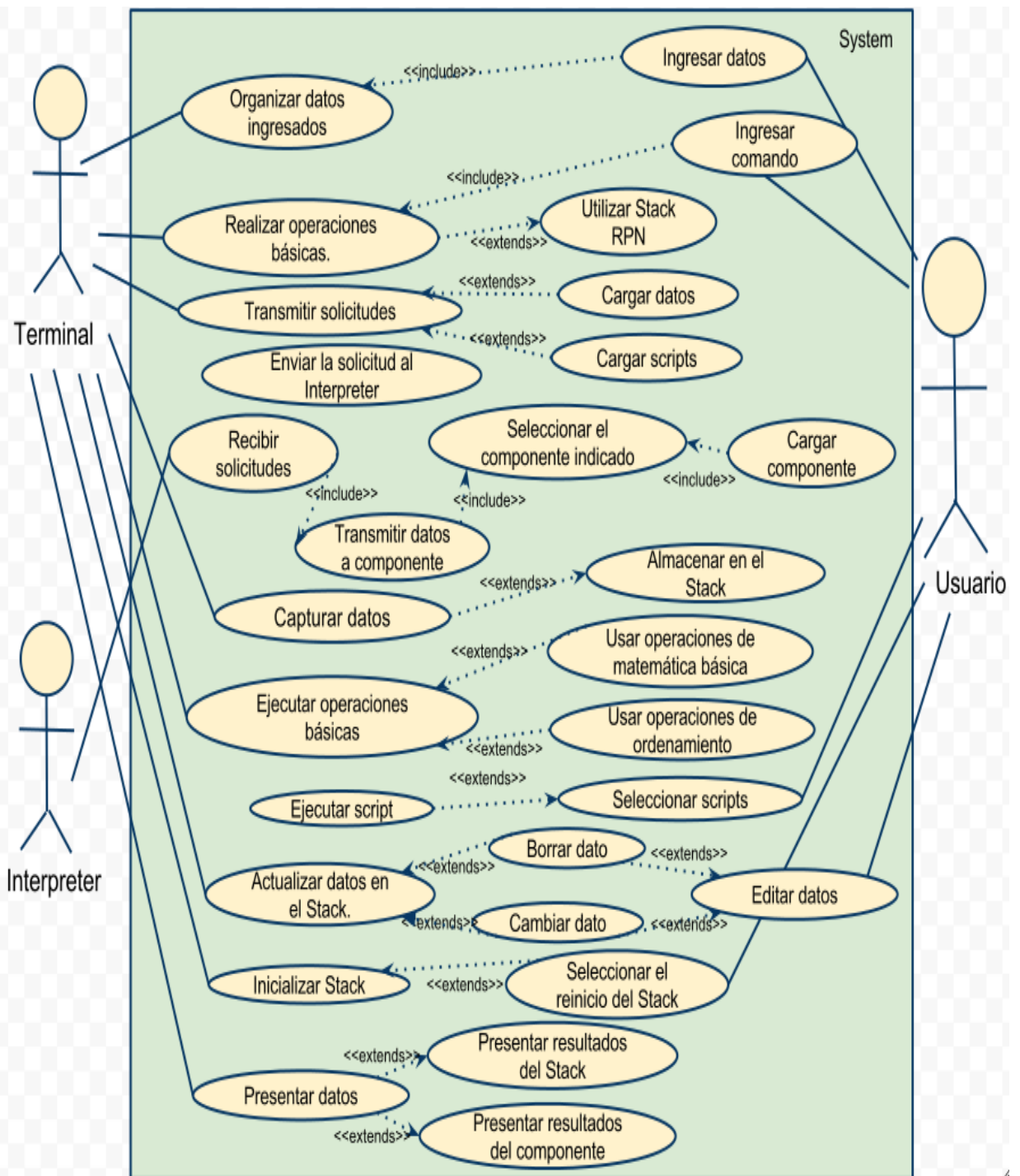


Gráfica 16: Trasmisión de solicitudes.

Fuente: Los autores.

Nombre	Requerimiento 9: Transmisión de Solicitudes.	
Descripción	Interpreter envía solicitudes para la comunicación entre la Terminal y diferentes módulos del proyecto actuales o posteriores, de acuerdo a las solicitudes hechas por el usuario.	
Entradas		
	Comandos u órdenes dadas por el usuario.	
Resultados		
	Presenta el módulo solicitado a través de la Terminal.	
Actores:		
	Terminal, Interpreter.	
Precondiciones:		
	1. Usuario ingresa la orden para solicitar el módulo requerido.	
Flujo Normal		
	Actor	Sistema
	1. Usuario envía la orden a través de la Terminal	2. Interpreter recibe la orden enviada. 3. Transmisión de orden al servidor. 4. Selecciona el módulo solicitado.
	5. Terminal carga el módulo que fue solicitado	
Flujo Alternativo:		
	1. Usuario ingrese un comando no válido	
Pos condiciones:		
	Interpreter recibe la solicitud transmitida y se carga el módulo correspondiente	

Diagrama de caso de uso general del sistema:



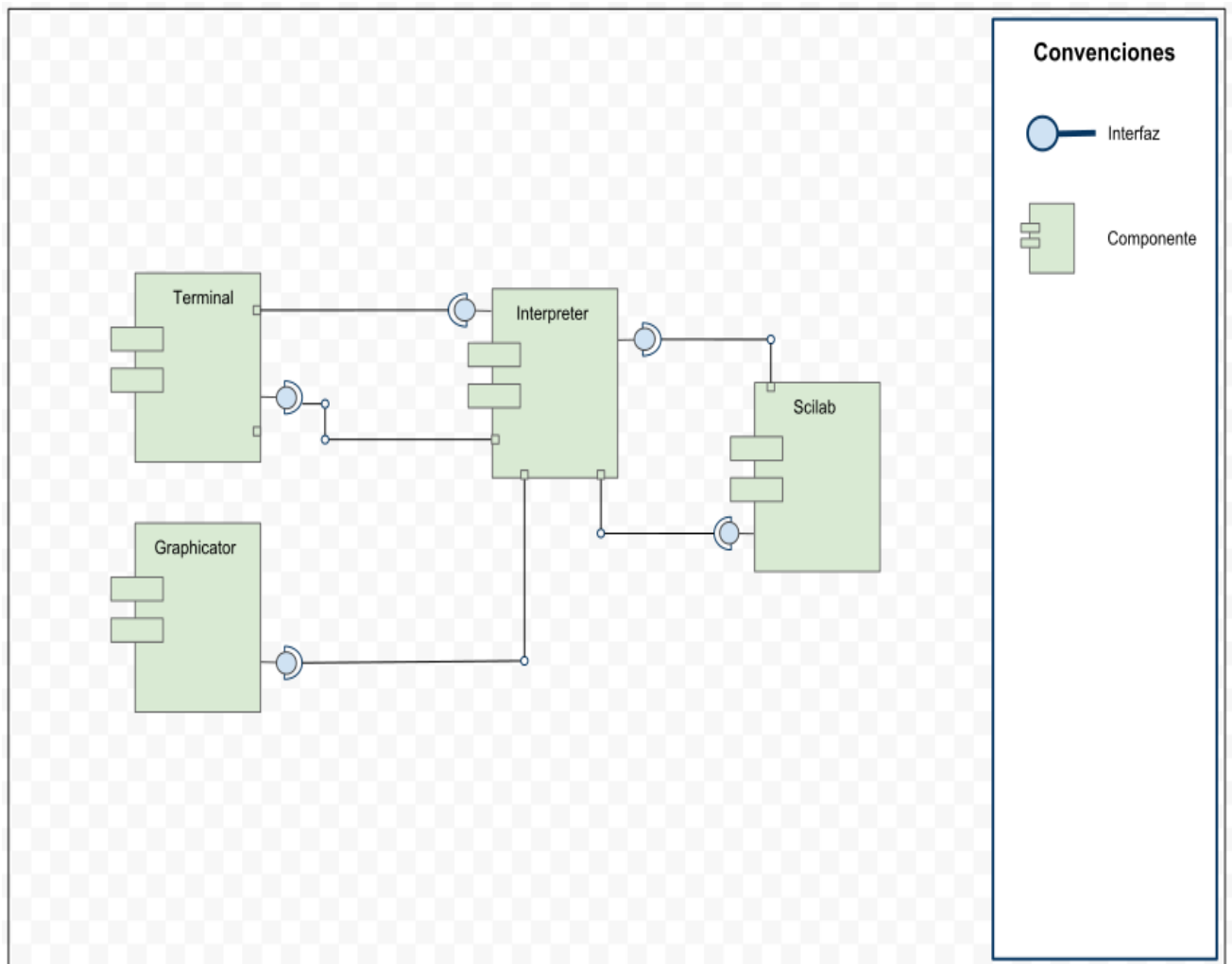
Gráfica 17: Diagrama e caso de uso general

Fuente: Los autores.

- Diagrama de componentes

En el siguiente diagrama se visualizará los componentes que hacen parte del proyecto, sus medios de comunicación por medio de interfaces

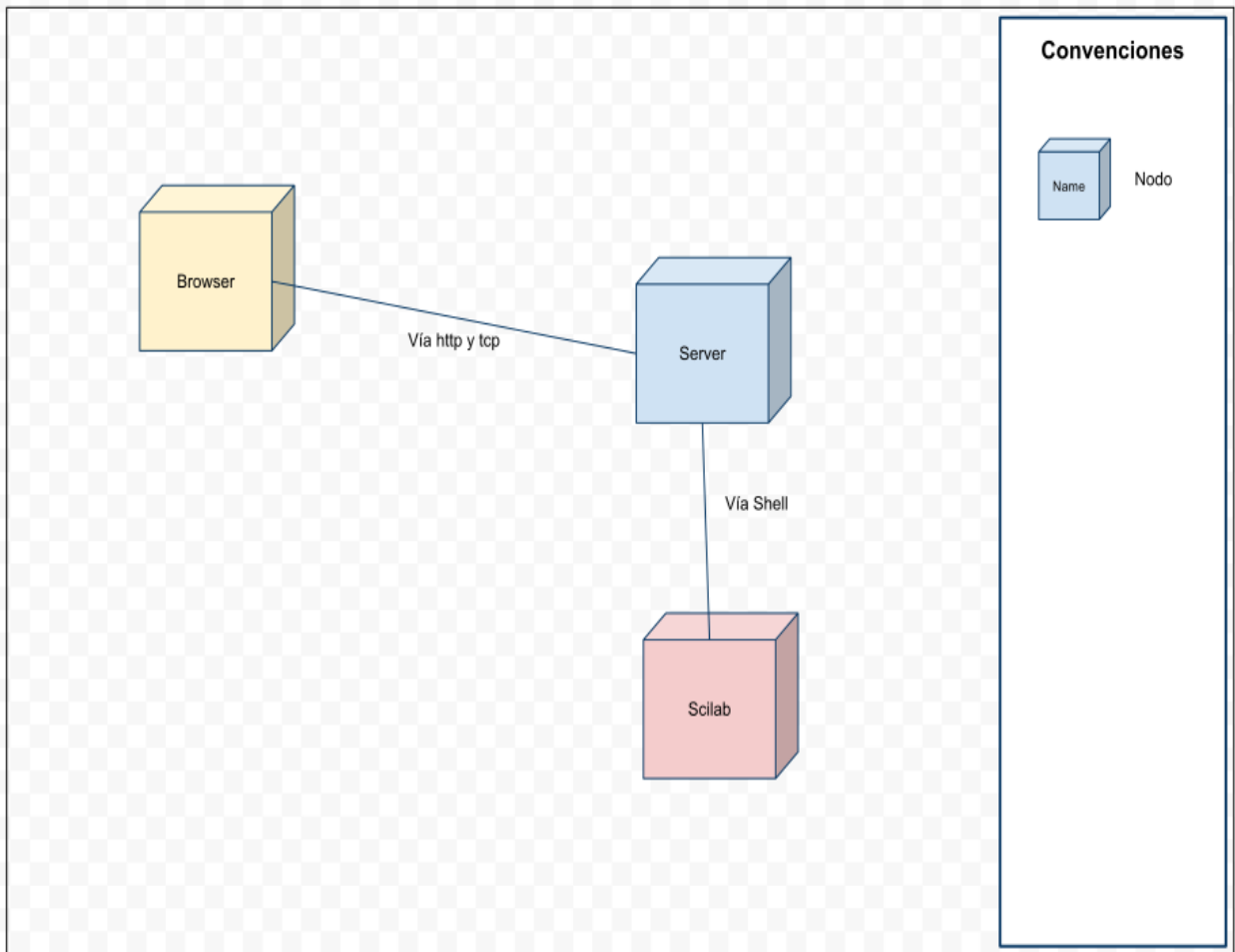
representadas por medio de web services y la distribución general de Engineer+.



Gráfica 18: Diagrama de componentes
Fuente: Los autores.

Diagrama de Despliegue

- Se realiza una apreciación del hardware utilizado en las implementaciones del sistema y las relaciones entre sus componentes.



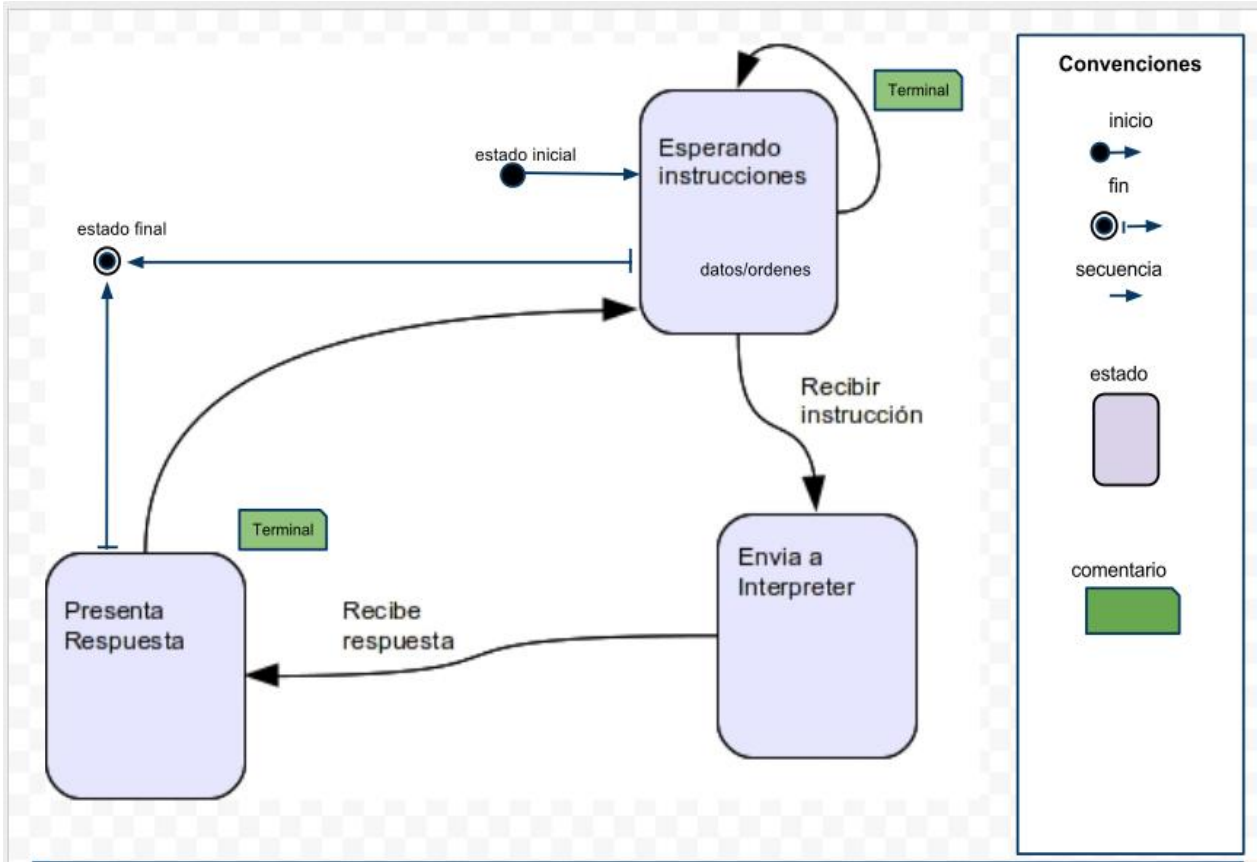
Gráfica 19 : Diagrama de despliegue
Fuente: Los autores.

3.3.2 Diagramas Dinámicos

Se usan para expresar y modelar el comportamiento del sistema a lo largo del tiempo, con el que describe las relaciones temporales entre objetos. Muestran las interacciones entre objetos ocurridas en un escenario del sistema, en esta aplicación se determinaron los siguientes:

- Diagrama de máquina de estado.

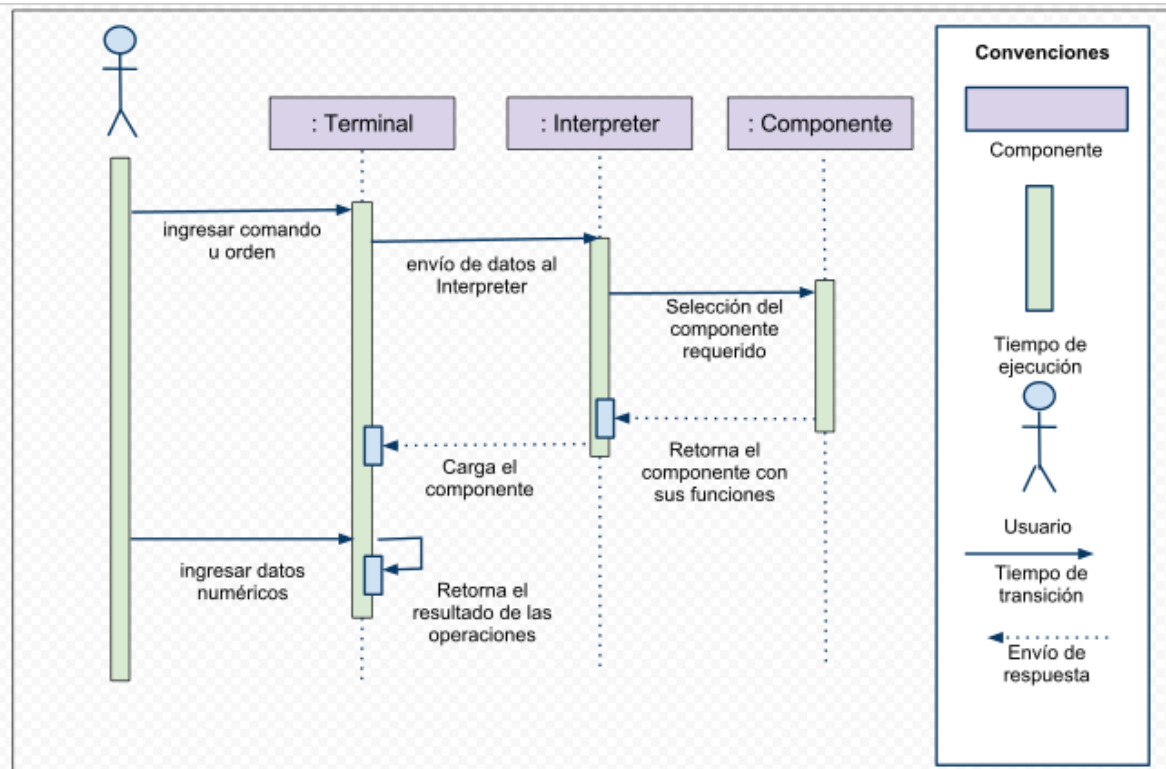
En el siguiente diagrama se hará una apreciación de los flujos que componen el sistema ya que se evidencia cada una de las rutas o caminos en los que la aplicación se verá envuelta, además de todas las solicitudes que transmite el usuario.



Gráfica 20: Diagrama de máquina de estado
Fuente: Los autores.

- Diagrama de secuencia

En el siguiente gráfico se observan las diferentes relaciones de todos los usuarios que interactúan con el sistema, desde la fase inicial hasta las peticiones que realizan dependiendo de las instrucciones que se hayan ingresado.



Gráfica 21: Diagrama de secuencia.
Fuente: Los autores.

4. DESARROLLO

En el desarrollo de la aplicación se tienen en cuenta dos aspectos fundamentales del proyecto, las especificaciones técnicas de software y de hardware ya que con estas se puede medir la capacidad de procesamiento y el impacto que el software tendrá en las diferentes ramas donde se utilizará.

4.1 Especificaciones Técnicas.

En esta etapa se definen las diversas características con las que el software interactuará además las normas, exigencias y procedimientos a ser empleados en el uso del aplicativo definiendo su capacidad de procesamiento y rendimiento en los diferentes equipos de computo.

4.1.1. Software

Para tener un correcto uso de la aplicación es necesario contar con:

- Una a Conexión a Internet (Banda Ancha superior a 1 mega).
- Cualquier navegador web: Mozilla Firefox desde su versión 3.0 o Google Chrome desde su versión 2.0 preferiblemente.
- Sistemas operativos: Windows, Linux, Mac
- Servidor xampp APACHE (en caso de no conocer la herramienta ver el anexo).
- Servidor Hamachi (ver anexo para conocer su instalación y demás procedimientos).

4.1.2. Hardware

Las especificaciones técnicas del equipo deben contar con lo siguiente:

Requerimientos Oficiales:

Requerimientos Mínimos:

- Procesador: Intel Pentium 4 3200 MHz (o equivalente)
- RAM: 2048 MB

- Tarjeta Gráfica: NVIDIA GeForce 8600 256 MB o ATI Radeon X1950 256 MB.
- Disco Duro: 500 MB de espacio libre para la instalación
- Teclado, Mouse.

Requerimientos Recomendados:

- Procesador: Intel Core 2 Duo 2400 MHz (o equivalente).
- RAM: 4096 MB.
- Tarjeta Gráfica: NVIDIA GeForce GTS 250 512 MB o ATI Radeon 4850 512 MB.
- Disco Duro: 500 MB de espacio libre para la instalación.
- Teclado, Mouse.

5. GLOSARIO

Apache: Es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 [1] y la noción de sitio virtual.

Componente: Es un paquete de clases donde interactúan unas con otras.

JavaScript: Es un lenguaje de programación el cual es interpretado por el navegador web y es una extensión del lenguaje HTML

MApache: Es un paquete de software implementado en PHP que expresa un estilo de programación que reduce el tiempo de desarrollo de una aplicación con acceso a bases de datos.

HTTP: El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW)

PHP: PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más Conocido y utilizado en la actualidad.

XAMPP: Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

Stack: Pila de almacenamiento de datos con un comportamiento LIFO (Last Input First Output)

Módulo: es una parte de un programa de ordenador los módulos suelen estar organizados jerárquicamente en niveles, de forma que hay un módulo superior que realiza las llamadas oportunas a los módulos del nivel inferior.

Tópico: Grupo de clases que en conjunto solucionan requerimientos de un tema específico.

Terminal: Representa la interfaz con la que el usuario va a interactuar

Intérprete: Es un modulo que interactúa con los diferentes componentes del proyecto Engineer+

RPN: notación polaca inversa es un método algebraico alternativo de introducción de datos.

RUP: Proceso Unificado Rational; es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos

Servidor web: Es una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

Web Services: Permiten la comunicación entre aplicaciones o componentes de aplicaciones de forma estándar a través de protocolos comunes (como http) y de manera independiente al lenguaje de programación, plataforma de implantación.

6. CONCLUSIONES

Se alcanzaron los objetivos propuestos por lo tanto se concluye:

- Se controla el acceso de los usuarios, identificándolos como usuario administrador y usuario, de esta forma los administradores, poseen más opciones al momento de registrar un nuevo componente, porque es necesario tener control, sobre las personas que van a aportar o a realizar cambios sobre Engineer+.
- La utilización del Stack permite el manejo de grandes cantidades de datos, y el uso de la notación polaca inversa, facilita el proceso de solución de cualquier tipo cálculo, debido a que evita la utilización de símbolos y su distribución es óptima tanto al almacenar los datos como al realizar operaciones.
- La comunicación entre componentes, a través de Interpreter, se realiza de forma óptima, presentando los resultados esperados en la interfaz debido a que las funciones de un componente externo se cargan en Terminal, facilitando el uso e interacción con el usuario.
- El uso de una línea de comandos por parte del usuario facilita el manejo de instrucciones y la utilización correspondiente de los datos almacenados en el Stack, incluso al momento de presentar los resultados, porque la recepción de comandos evita el proceso complejo de llenar de formularios e incluso es imposible tener formularios para recibir miles de datos.
- Se logró adaptar operaciones básicas al proyecto, para probar su funcionalidad, como la conexión con SCILAB y los scripts adaptados posteriormente.
- Se logró el buen funcionamiento del componente Terminal e Interpreter realizando todas las tareas propuestas, comprobando la presentación y captura de datos, la comunicación entre módulos y tiempos de respuesta.

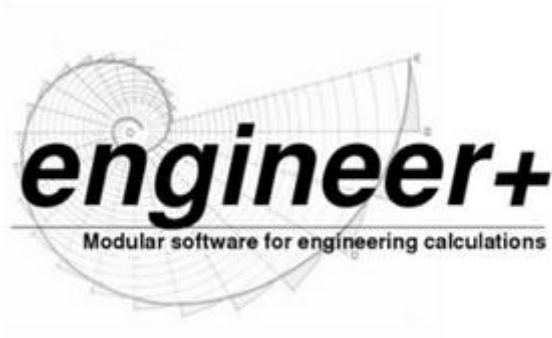
7. BIBLIOGRAFÍA

- About the Apache HTTP Server Project(2005), proyecto Apache. Disponible en URL: http://httpd.apache.org/ABOUT_APACHE.html
- Sitio oficial del proyecto Apache(1999), Disponible en URL: <http://www.apache.org/>
- Sitio oficial del servidor local Xampp (2010), Documentación y descarga del servidor Disponible en URL: <http://www.apachefriends.org/es/xampp.html>
- Definición componente (2005), Diccionario de la lengua española. Disponible en URL: <http://www.wordreference.com/definicion/componente>
- JavaScript, Información General, (2005) Disponible en URL: http://www.webdevelopersnotes.com/basics/languages_on_the_internet.php3
- Tutorial Javascript (2010). Explicación y ejemplos del lenguaje. Disponible en URL: <http://www.w3schools.com/js/default.asp>
- Ejemplos de JavaScript (2005), Una interesante y variada recopilación de pequeños ejemplos en JavaScript. Disponible en URL: <http://javascript.mis-algoritmos.com/>
- Documentación general JavaScript Fundación Mozilla (2005), Disponible en URL: <https://developer.mozilla.org/es/Javascript>
- Sitio oficial del Arca-csl (2010), Disponible en URL: <http://sites.google.com/site/arcacslsite/>
- Documentación general del proyecto Mapache. Disponible en URL: <http://www.facebook.com/topic.php?uid=23783828200&topic=4887>
- PHP, definición y sus características. (2009) Existen tanto PHPEclipse como PDT for Eclipse. Disponible en URL: <http://www.phpclipse.com/> y <http://www.eclipse.org/pdt>

- Site oficial de http (2003), Documentación e información general del protocolo. Disponible en URL: <http://www.w3.org/Protocols/>
- MVC (2005), Información general del modelo vista controlador. Disponible en URL: <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
- Gráficos e información MVC (2010), Descripción por medio de gráficos. Disponible en URL : <http://blog.micayael.com/2010/08/06/framework-mvc/>
- Ejemplo de MVC (2004), Variedad de ejemplos sobre el modelo vista controlador. Disponible en URL: <http://ciclope.fi.upm.es/doc/mvc/>
- Información general Diagrama de clases (2008) Disponible en URL: <http://www.dcc.uchile.cl/~psalinas/uml/modelo.html>
- Tutorial y ejemplos UML (2008), referenciación sobre Diagrama de componentes. Disponible en URL: http://www.programacion.com/articulo/introduccion_a_uml_181/6
- Presentación y documentación (2010), Diagramas de despliegue. Disponible en URL: <http://www.slideshare.net/arcangelsombra/diagramas-de-despliegue-uml-1475353>
- Ejemplos maquinas de estado (2008), documentación general sobre los diagramas de maquinas de estado. Disponible en URL: http://www.sparxsystems.com.ar/resources/tutorial/uml2_statediagram.html
- Información general Diagrama de Casos de uso (2009), Disponible en URL: <http://www2.uah.es/jcaceres/uploaded/capsulas/DiagramaCasosDeUso.pdf> y <http://www.clikear.com/manuales/uml/diagramascasouso.aspx>
- Presentación diagramas de secuencia (2007), Disponible en URL: http://www.chuidiang.com/ood/metodologia/diagrama_secuencia.php

- Tutorial UML (2005), Extenso tutorial UML (lenguaje unificado de modelado). Disponible en URL: <http://www.clikear.com/manuales/uml/index.aspx>
- Sitio oficial UML (2011), Disponible en URL: <http://www.uml.org/>
- Curso online UML (2010), Descripción general de UML. Disponible en URL: <http://www.lawebdelprogramador.com/cursos/UML/index1.html>
- Sitio web oficial de PHP. (2010) Disponible en URL: <http://www.php.net/>
- Minutes PHP Developers Meeting. (2009) Manual de Referencia PHP. Disponible en URL: <http://www.php.net/~derick/meeting-notes.html>
- RPN, conceptos generales. (2010) RPN: Introducción a la notación polaca inversa. Disponible en URL: <http://h41111.www4.hp.com/calculators/es/es/articles/rpn.html>
- An Analysis of a Logical Machine Using Parenthesis-Free Notation. (2005) Disponible en URL: <http://www.jstor.org/pss/2001990>
- Modelo UP. (2008). Disponible en URL: <http://es.wordpress.com/tag/rup/>

8. MANUALES



8.1 Manual Técnico

**DESARROLLO DE CLASES Y COMPONENTES
REUTILIZABLES PARA EL PROYECTO ENGINEER+
DEDICADOS A LA TERMINAL RPN**

V 1.0

Tabla de contenido

INTRODUCCIÓN	3
1.1 Definiciones previas.....	3
2. COMPONENTES.....	8
2.1 Terminal.	8
2.2 Interpreter.....	9
2.3 Modo de ejecución.	9

INTRODUCCIÓN

Engineer+ es un proyecto de software que busca la creación de una aplicación orientada a la web que permita el desarrollo de cálculos ingenieriles y que además permita la integración de métodos numéricos de quienes así lo requieran, de forma simple y rápida a través del lenguaje de programación SCILAB, el cual es muy usado en el ámbito de la ingeniería y las matemáticas.

El presente manual de usuario, se encarga de especificar el comportamiento de los componentes Terminal e Interpreter del proyecto Engineer+, su patrón arquitectural, el modelo de desarrollo, el lenguaje utilizado el comportamiento del sistema y las clases que lo componen.

Se especifica la interacción del software con el usuario y entre componentes, requerimientos del sistema a nivel hardware/software, además de la especificación de los términos utilizados, con el objetivo de guiar y mostrar todo lo trabajado en este software a las personas interesadas.

1.1 Definiciones previas

Terminal la GUI del proyecto.

Interpreter que permite el llamado de scripts .sce hechos en SCILAB de acuerdo a los comandos dados por el usuario.

Componente: aquellos recursos desarrollados para un fin concreto y que puede formar solo o junto con otros, un entorno funcional requerido por cualquier proceso predefinido.

Módulo: son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo.

SCILAB: software matemático, con un lenguaje de programación de alto nivel, para cálculo científico, interactivo de libre uso.

Stack: es una estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar datos.

MVC: es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Localhost: nombre reservado de un equipo router o dispositivo.

Engineer+, está desarrollado con el uso de librerías MApache, el cual es un proyecto de la comunidad de software libre Arca-csl de Uniminuto, que permite la creación de interfaces gráficas en html siguiendo el paradigma de programación orientada a objetos.

Engineer+ está dividido en módulos, los cuales se trabajan a partir de componentes, en el cual en el presente proyecto, se trabajan los prototipos de dos de ellos Terminal e Interpreter. La primera es la interfaz gráfica del sistema, el cual se encarga de presentar los datos y de interactuar con los usuarios. Interpreter es el encargado de comunicar a la interfaz con los componentes integrantes del proyecto, y debe cumplir con la responsabilidad, de transmitir datos, y adaptar nuevos componentes a través de la evolución del aplicativo, en la siguiente gráfica podemos observar la distribución modular y los componentes iniciales del proyecto:

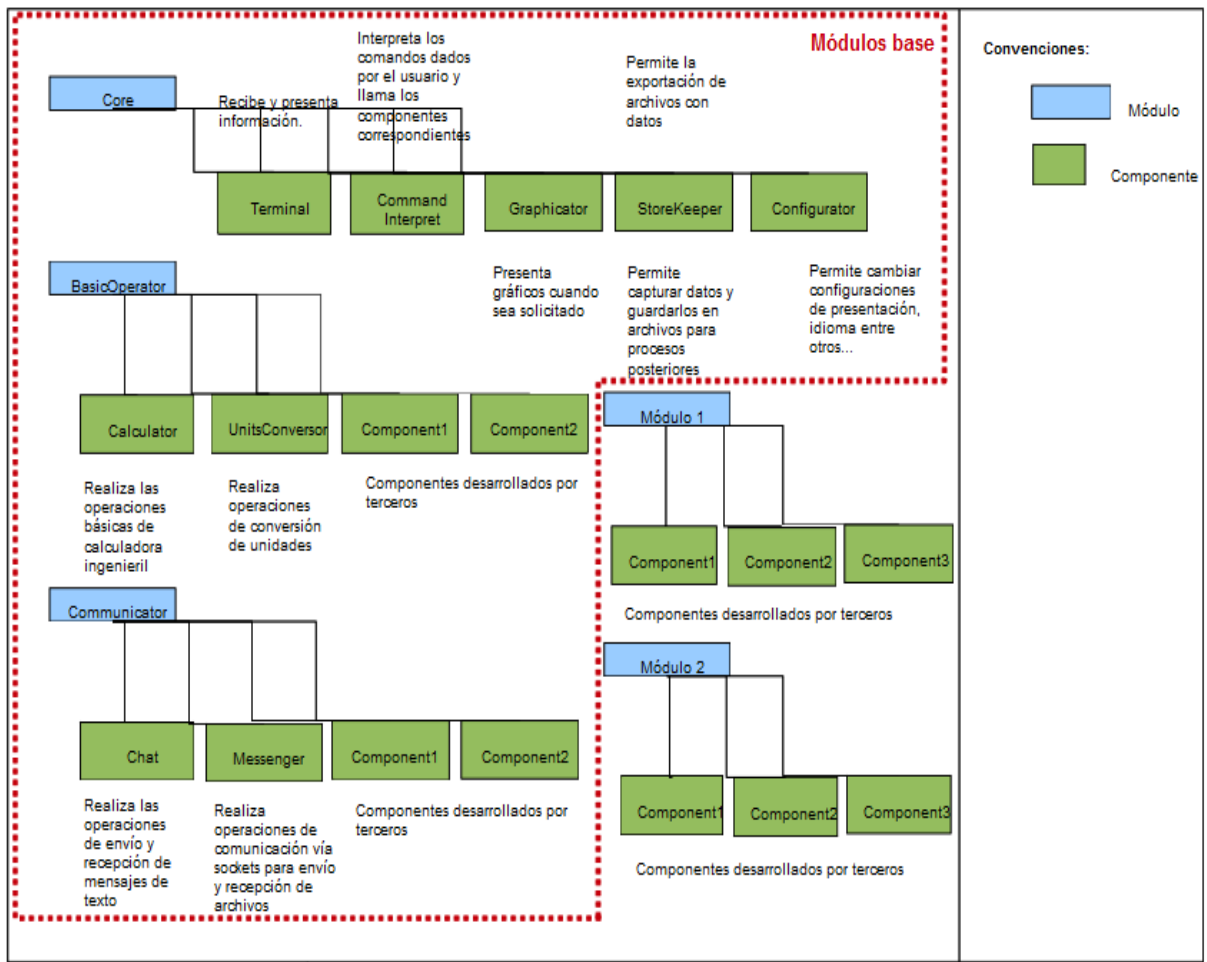


Gráfico N° 1: Organización modular.

Patrón arquitectural.

Este esquema permite ver la aplicación en función de la organización de sus archivos, se sigue este estándar con el fin de estructurar, y facilitar la adaptación de scripts a la aplicación, esto siguiendo el patrón modelo-vista-controlador, el cual permite separar estos tres aspectos, de igual forma se dividen las carpetas de configuración las cuales son: Config – Themes – Lenguajes. En la carpeta Config, se encuentran los datos sobre el desarrollo del software, es decir sus integrantes y las páginas web en las cuales se encuentra más información sobre el proyecto www.engineerplus.tk y www.arca-csl.or, En Themes se almacena las imágenes y propiedades (css) de un Tema del software. Finalmente en Lenguaje, es la carpeta destinada a almacenar la configuración, por si el software, quisiera ser presentado en distintos idiomas. Estos paquetes, pueden ser reutilizados, por diferentes proyectos, lo cual permite minimizar el trabajo.

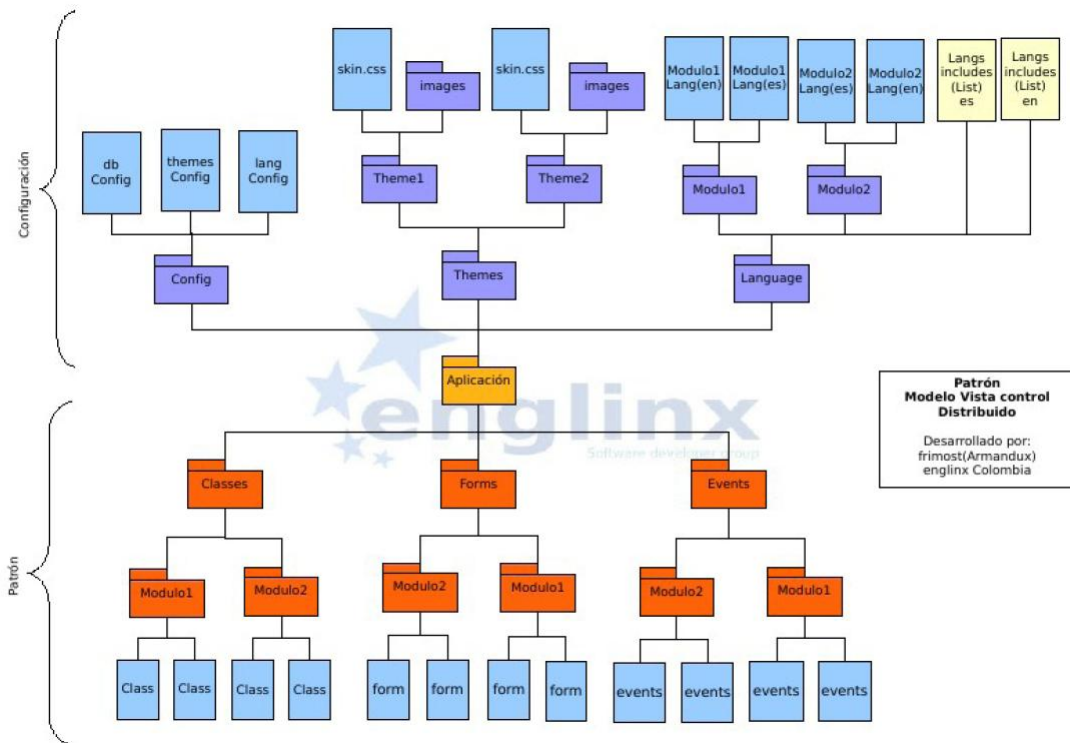


Gráfico N° 2: Esquema de Engineer+

El esquema presenta el comportamiento del sistema desde la perspectiva de servicios, se muestra la funcionalidad respectiva de cada componente y la interacción entre ellos, realizada para presentar los resultados al usuario.

En primera instancia se observa a Terminal, quien interactúa con el usuario, el cual puede recibir datos u órdenes, en el primer caso, Terminal almacena los datos ingresados, en segundo caso puede tener dos opciones. La primera es que la orden sea ejecutada desde ella misma, y la segunda cuando el usuario solicite las funciones de un componente externo, por lo cual Terminal, se comunica con Interpreter, quien devuelve al usuario una lista de componentes registrados en Engineer+, para que éste seleccione el componente sobre el cual desea trabajar, y al seleccionarlo, por medio de web services, se carga en Terminal las funciones del componente externo.

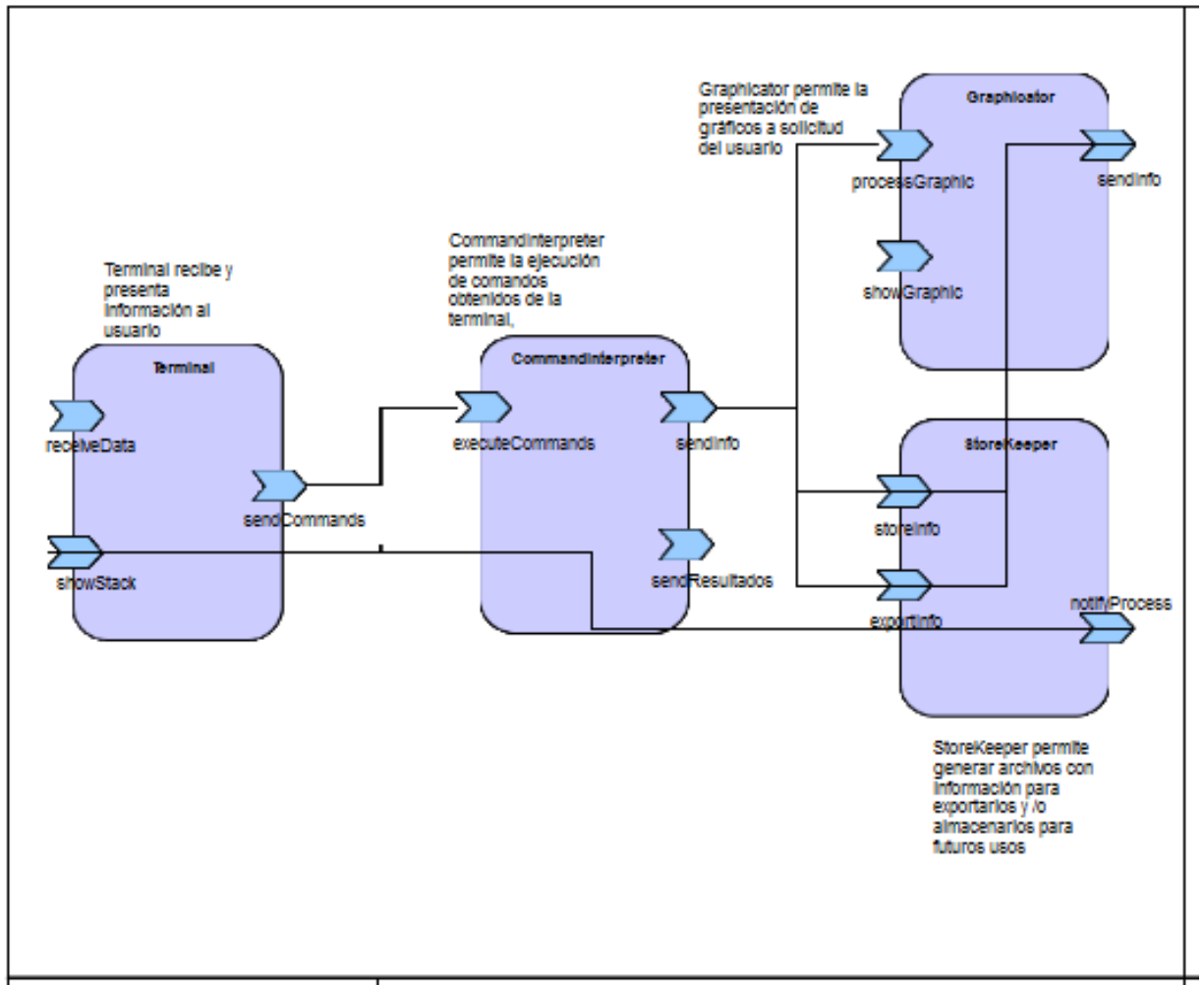


Gráfico N° 3: Diseño de componentes

Componentes del sistema:

El sistema está basado en diferentes componentes como lo vimos en el anterior gráfico, de los cuales se explicarán los dos presentes del proyecto Terminal e Interpreter, los cuales se especificarán a continuación.

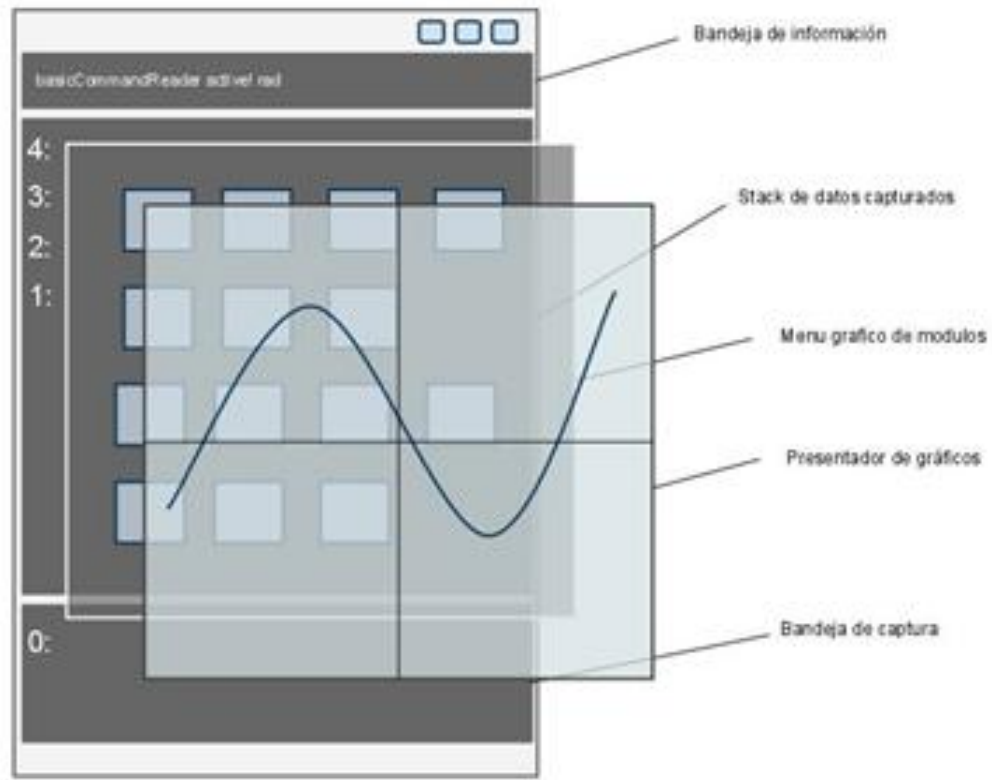
2. COMPONENTES

2.1 Terminal.

Terminal, es la interfaz gráfica del proyecto, ésta debe admitir la lectura de archivos y la carga de componentes enviados a través de Interpreter, tiene múltiples funcionalidades, como:

- Admite cambiar su presentación.
- Tiene comportamiento RPN (Notación Polaca Inversa) que acelera la captura de los datos,
- Se maneja a través de comandos
- Ofrece ayuda para conocer sus opciones y/o comandos.
- Tiene un control respecto a los envíos de datos realizados hacia el servidor, debido a que los evita.
- Su código se encuentra bien optimizado gracias a MApache quien provee librerías que facilitan el trabajo.

En el siguiente esquema se visualizará la distribución del componente Terminal:



2.2 Interpreter.

Componente encargado de comunicar la Terminal, con los componentes incorporados al sistema, actuales y posteriores, y optimiza el comportamiento del aplicativo. En el siguiente esquema se puede observar su diseño:

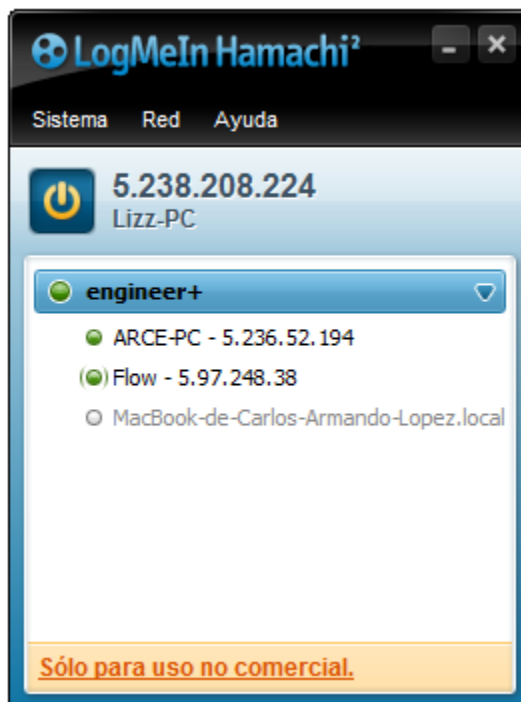
2.3 Modo de ejecución.

Actualmente, para los usuarios que deseen conectarse a la aplicación, como requisito previo es tener Hamachi, la cual es una aplicación gratuita configuradora de redes privadas virtuales, capaz de establecer vínculos, directos entre

ordenadores es decir que establece una conexión a través de internet simulando una red de área local.

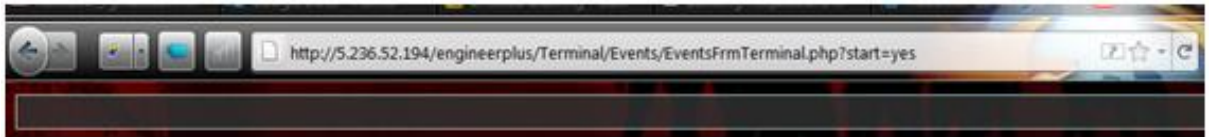
Para descargar el software, se debe realizar las siguientes indicaciones:

1. Ingresar al link `HYPERLINK`
`"https://secure.logmein.com/ES/products/hamachi2/" \t "_blank"`
`https://secure.logmein.com/ES/products/hamachi2/ .`
2. Seleccione la opción "probar gratis" y luego elija la opción "sin gestión".
3. Al instalarlo, Hamachi preguntará por la red a la que se quiere añadir.
4. Id Red: Engineer+.
5. Contraseña: angelica.
6. Espere un momento hasta que aparezca en la aplicación el nombre de la red con un punto verde indicando que hay acceso a esa red.



Cuando esté conectado a la red, puede ingresar a la aplicación usando la IP: 5.236.52.194.

Es decir, puede desde un navegador ir a la url: **HYPERLINK**
"http://5.236.52.194/Engineerplus/Terminal"
http://5.236.52.194/Engineerplus/Terminal



Allí aparece la aplicación con su configuración predeterminada como se muestra a continuación:



Para modificar, agregar y/o optimizar un componente de Engineer+, se debe tener instalado el servidor independiente Xampp, el cual nos permite presentar el

proyecto en el localhost del equipo, si desea instalar debe realizar los siguientes pasos:

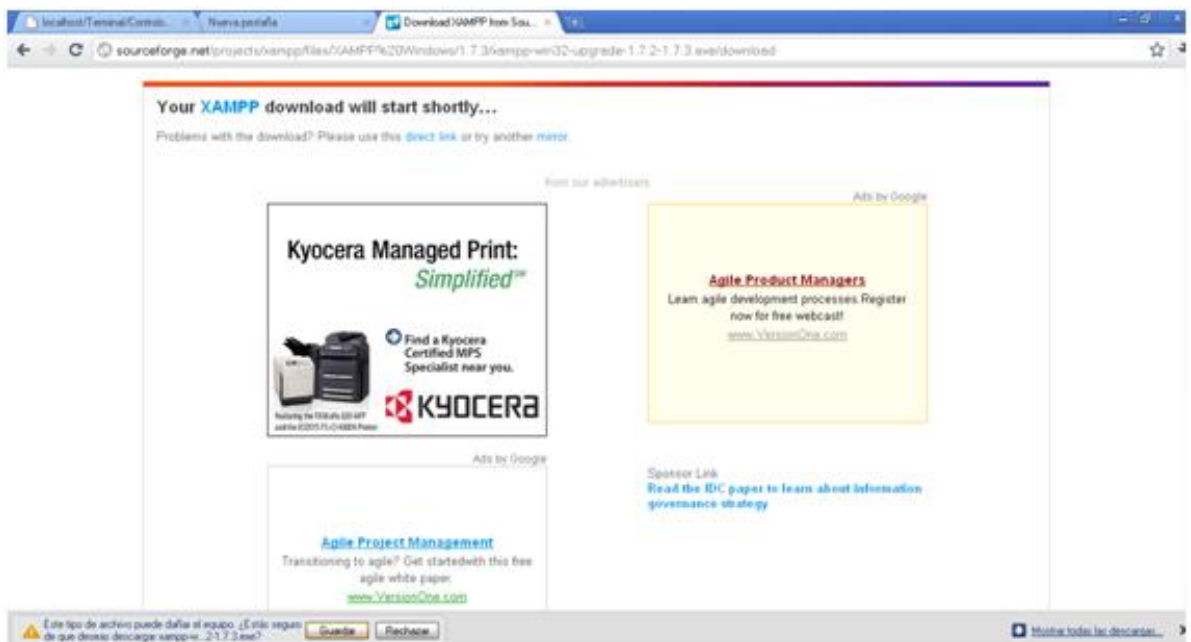
1. Se ingresa a la url: <http://www.apachefriends.org/es/xampp.html>



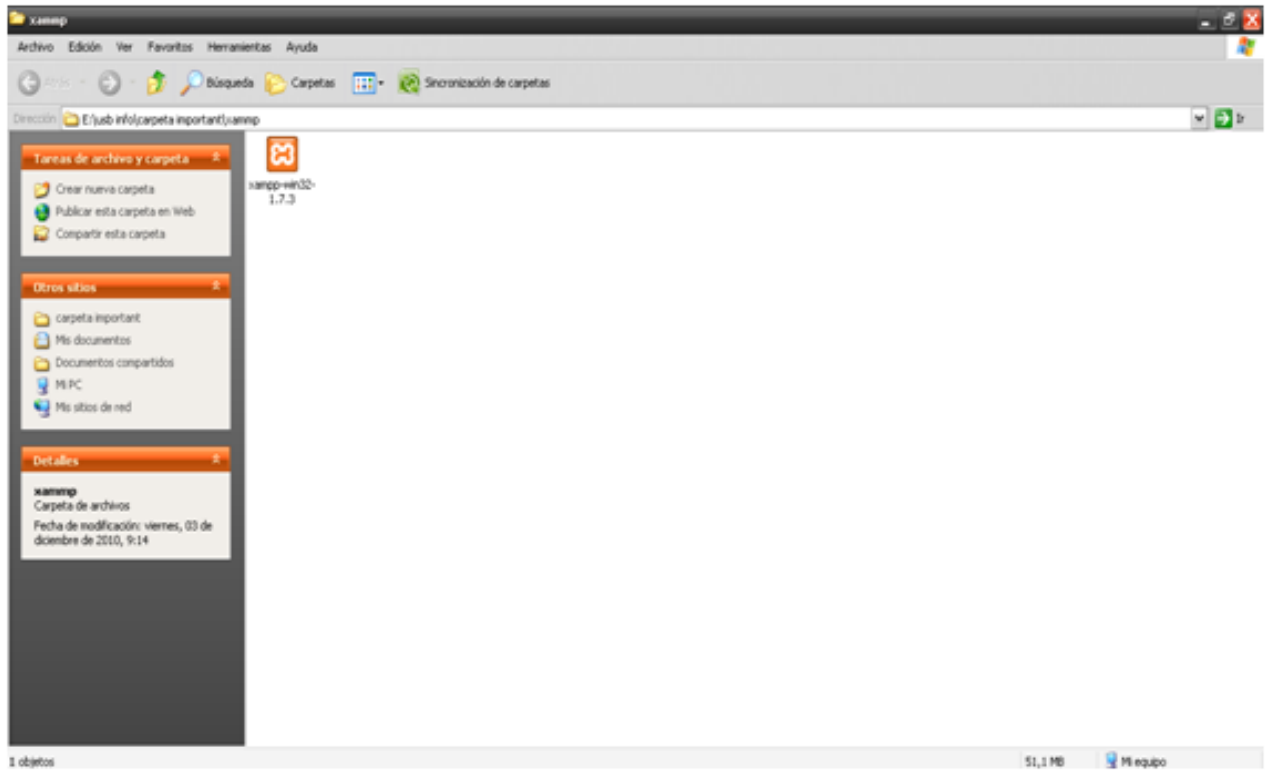
2. Luego se escoge el sistema operativo.



3. Se selecciona el instalador y se descarga.



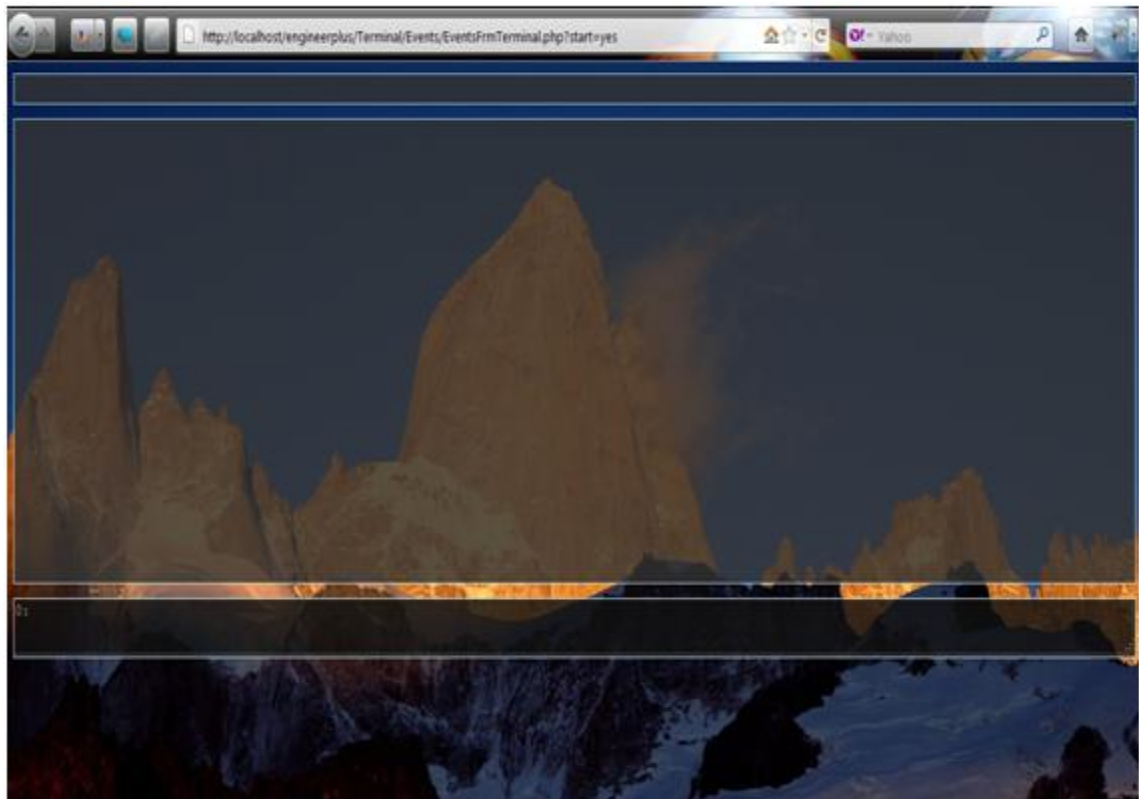
4. Después de descargar el instalador se debe tener un archivo de esta forma:



5. Luego se procede a instalarlo.



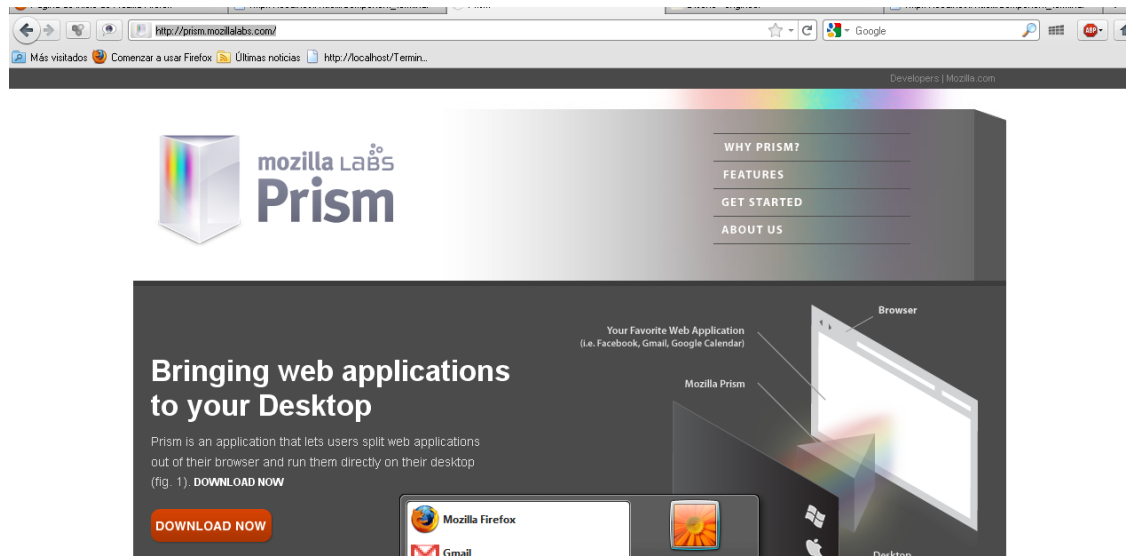
Luego de tener xampp, se debe llevar el código a la ruta -> c:/xampp/htdocs, posteriormente se prueba el aplicativo en la url: [HYPERLINK "http://localhost/Engineerplus/terminal"](http://localhost/Engineerplus/terminal) , en su equipo se visualizará lo siguiente:



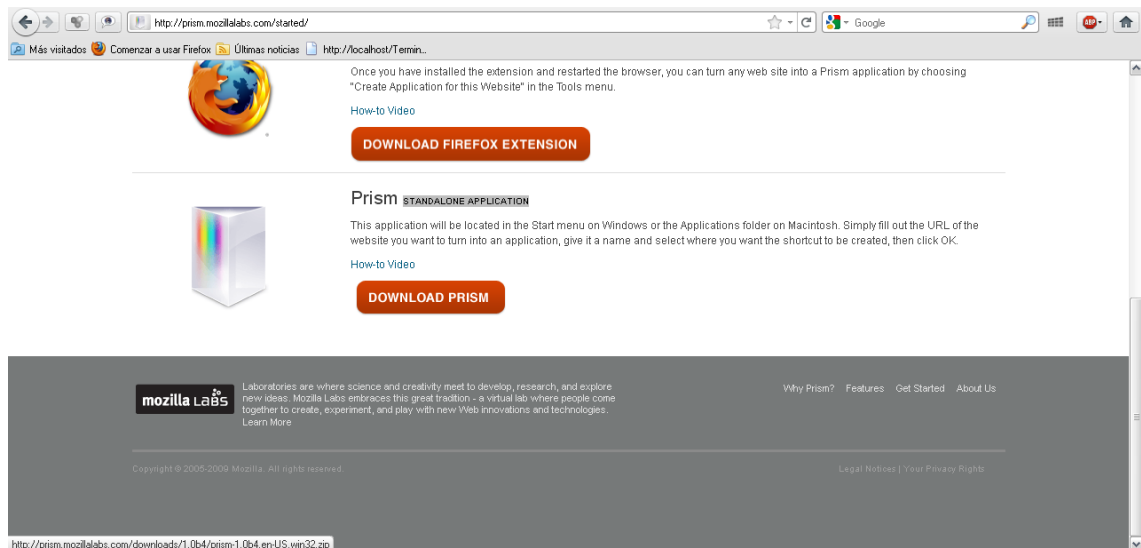
Ejecución del programa.

Se realiza la utilización de prisma de la Fundación Mozilla, el cual permite convertir una aplicación web a una aplicación stand-alone o de escritorio. A continuación se describe el proceso para generar la conversión de la aplicación.

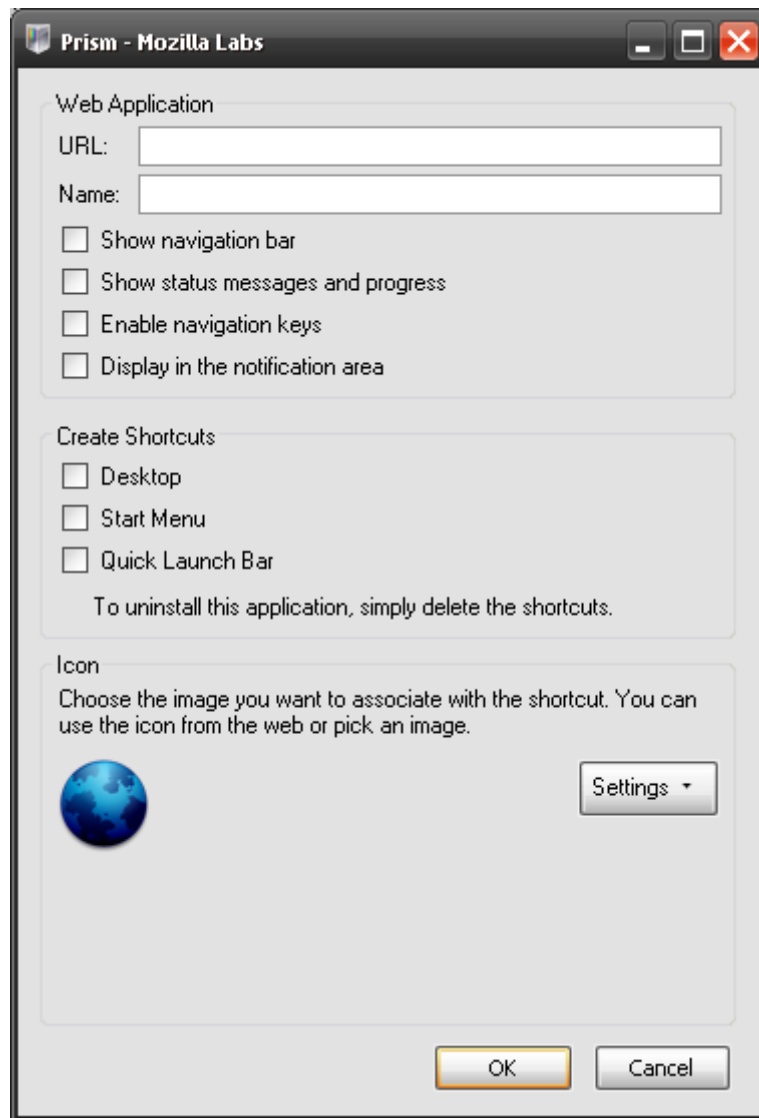
1. Ingresar a la URL: <http://prism.mozillalabs.com/>



2. Descargar el software en la opción Standalone application



3. Una vez descargado, ejecutar el archivo prism.exe, y se observará lo siguiente:



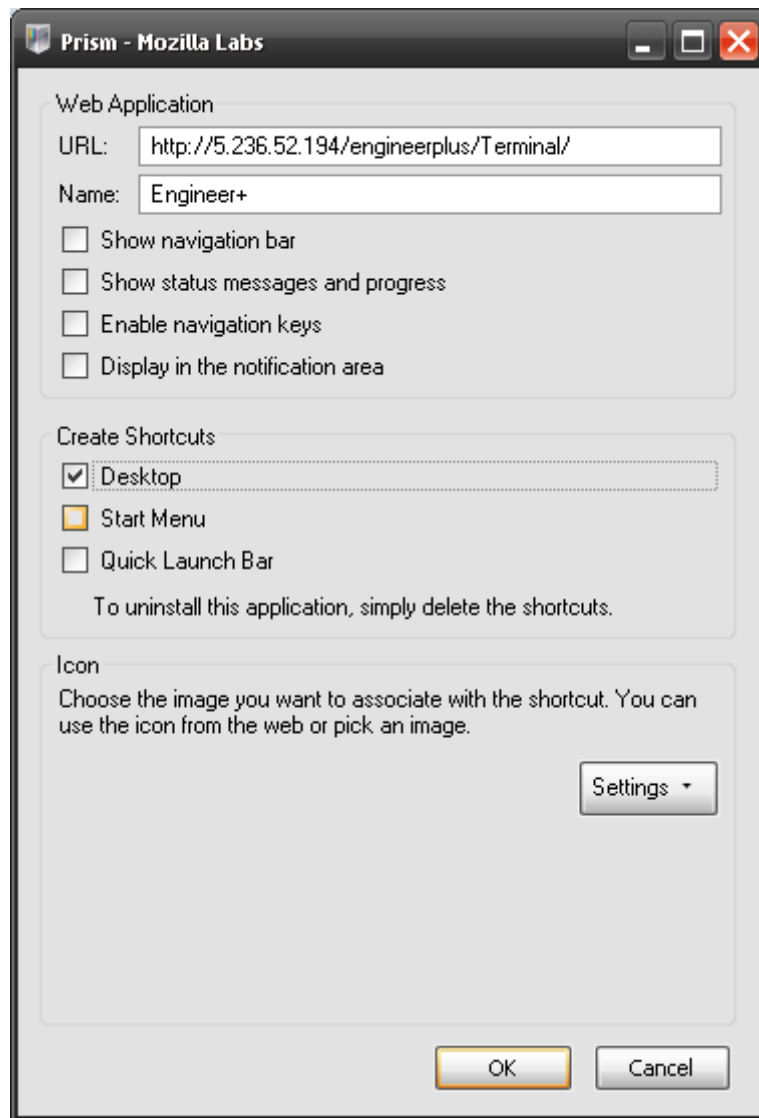
4. Llenar los campos solicitados con la siguiente información.

URL: <http://5.236.52.194/engineerplus/Terminal/>

Name: Engineer+

Seleccionar la opción Desktop el cual generará un icono en el escritorio.

Dar click sobre el botón OK y esperar un momento, mientras se genera la aplicación.

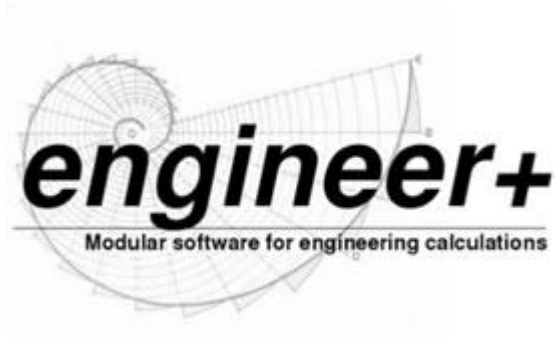


5. Luego se visualiza en el escritorio el icono de la aplicación. Dar doble click y se visualizará la aplicación de la siguiente manera:



Aplicación ejecutada desde prisma.

8.2. Manual de Usuario



Julio 2011

Manual Del Usuario

Terminal RPN

V. 1.0

**Corporación Universitaria Minuto de Dios
Bogotá D.C. – Colombia**

Tabla de Contenido

INTRODUCCIÓN:.....	3
1. Especificación Del Componente Terminal Engineer+	4
2. Ingreso A Terminal	6
3. Logueo e Inicio De Sesión Del Componente Session	7
4. Funciones Intermedias Del Componente Terminal	11
5. Funciones Avanzadas Del Componente Terminal	19
6. Presentación de Información para el usuario	37
7. Tabla De Comandos	40

INTRODUCCIÓN:

Este manual tiene como objetivo ser una guía en el uso del componente Terminal RPN, Interpreter y Session del proyecto Engineer+. En su contenido se podrá observar los aspectos más esenciales para poder utilizar de una manera muy fácil dicha aplicación.

Terminal es un software desarrollado con el fin de comunicar el sistema con los usuarios de manera gráfica, encargada de almacenar y presentar datos, realizar procesos matemáticos básicos y mostrar la interacción sostenida con el usuario.

Además aquellos datos que serán enviados por Terminal y recibidos por otro componente del sistema cuyo nombre es Interpreter el cual procesarán las solicitudes y será el encargado de transmitir las órdenes entre los demás componentes.

Session es el encargado de permitir el login de los distintos usuarios validando el rol y los permisos que se le concederán para posteriormente utilizar las funciones del aplicativo.

Los prototipos de los componentes mencionados permiten el comportamiento principal de Engineer+, ya que ellos, representan la interfaz gráfica del sistema y la comunicación entre componentes, lo cual permite la adición de scripts o funcionalidades posteriores.

A continuación se realizará una explicación más profunda del funcionamiento del prototipo, observando la interacción de sus componentes claves.

1. Especificación Del Componente Terminal Engineer+

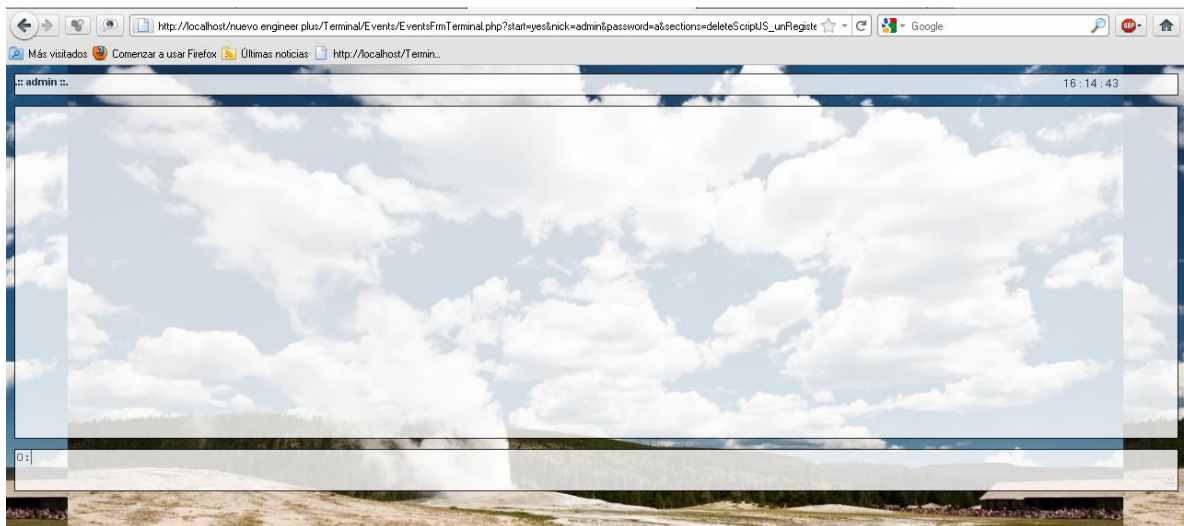
Descripción Proceso: Especificación del componente Terminal

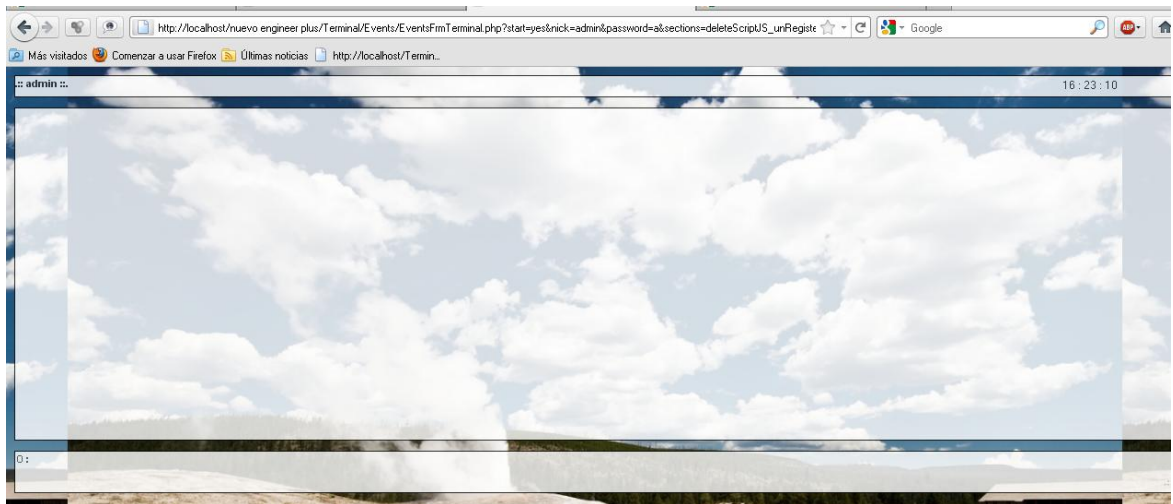
Definición: Terminal es el componente encargado de la captura de datos para el posterior procesamiento de las operaciones.

Una pequeña descripción del componente es el siguiente:

Terminal es un componente muy importante del proyecto Engineer+ ya que es el prototipo inicial y es el encargado de recibir las solicitudes dadas por el usuario, algunas de estas permiten realizar tareas como: seleccionar un componente, enviar datos, entre otras. Además existe otro componente que es la parte lógica pues permite la comunicación entre componentes y es el encargado de recibir las solicitudes por parte de Terminal, permitiendo el procesamiento de la información.

- Específicamente la Terminal consta de tres partes fundamentales, la primera área se encuentra ubicada en la parte inferior y su función es la recepción de datos por parte del usuario, la segunda área se encuentra ubicada en la parte intermedia y es la encargada de mostrar los datos almacenados junto con los resultados de las operaciones y la tercera área que se encuentra ubicada en la parte superior la cual muestra la hora y el componente en el cual el usuario se encuentra trabajando.

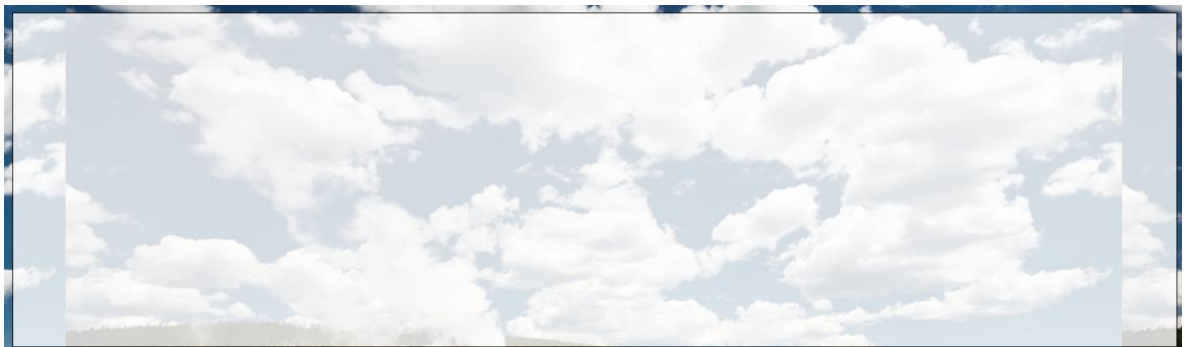




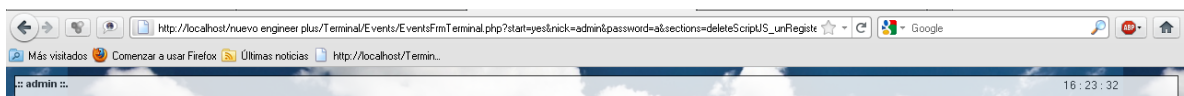
- En la interfaz de la Terminal el usuario encontrará un área de texto en la parte inferior para ingresar datos y comandos.



- De igual forma el usuario encontrará en la parte intermedia un área de texto, donde se mostrarán los datos ingresados y el resultado. En esta área no se muestran los operadores.



- Además en la parte superior se encuentra el área de notificaciones de la Terminal, donde se mostrará la hora e información del componente en el cual se encuentra trabajando el usuario además del rol con el cual el usuario ingreso al componente.



2. Ingreso A Terminal.

Terminal RPN

Engineer+

Requerimientos:

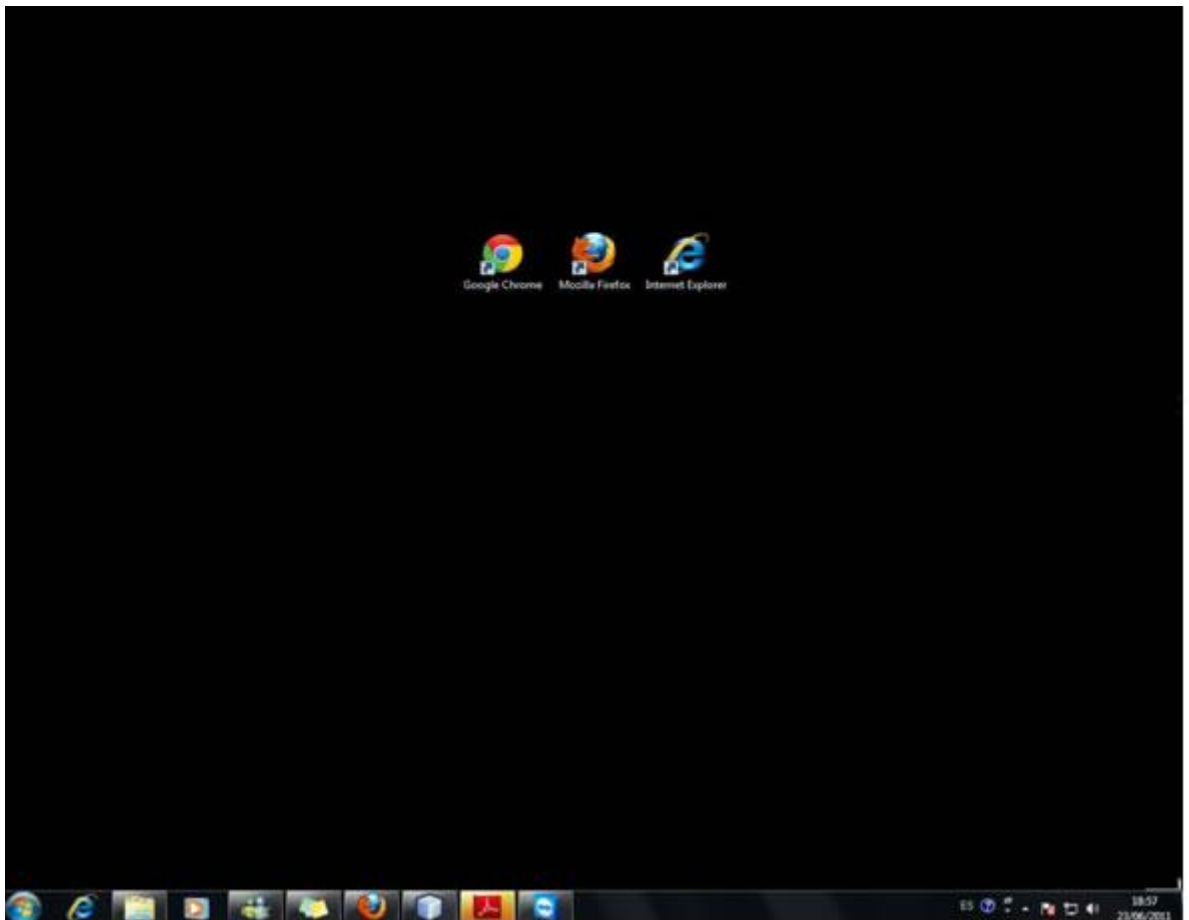
Para utilizar la Terminal RPN es necesario contar con los siguientes requerimientos:

- Conexión a Internet (Banda Ancha)
- Cualquier navegador web: Mozilla Firefox desde su versión 3.0 o Google Chrome desde su versión 2.0 preferiblemente o Internet Explorer desde su versión 8.
- Sistemas operativo: Windows, Linux, Mac

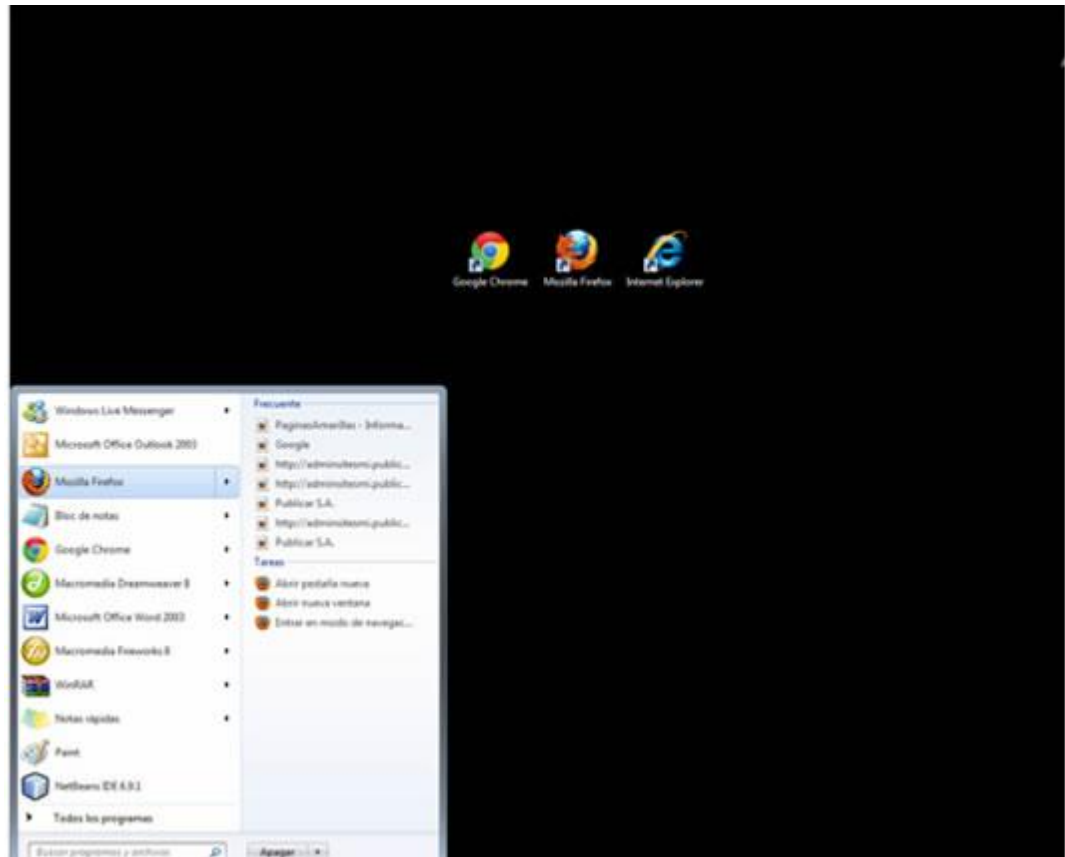
3. Logueo e Inicio De Sesión Del Componente Terminal Engineer+

Para ingresar a la Terminal RPN debemos seguir con los siguientes puntos:

1- Ingresar a un navegador web, para esto diríjase al escritorio y encontrará un acceso directo del navegador (Mozilla Firefox, Google Chrome, Internet Explorer).



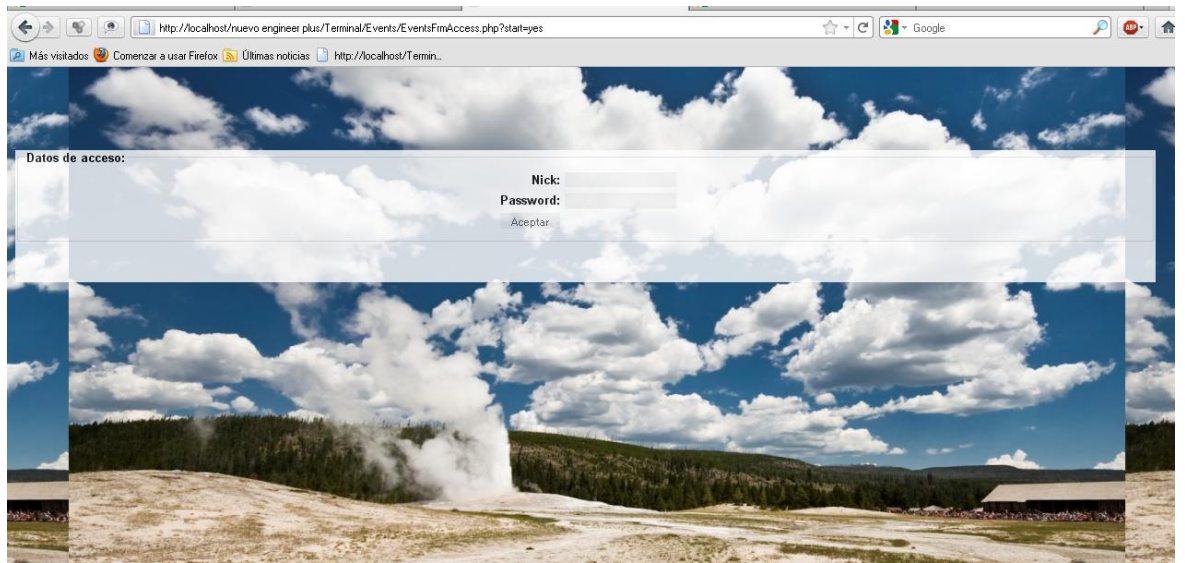
En el caso de no contar con el acceso directo diríjase a inicio, clickee en todos los programas y en la sección de navegadores escoja el de preferencia. (Para el ejemplo se usara el navegador web Mozilla Firefox).



2- Al ingresar al navegador, se debe dirigir a la barra de direcciones e ingresar la siguiente dirección web <http://5.236.52.194/Engineerplus/Terminal> y presionar la tecla Enter.



En un determinado momento visualizará una interfaz como lo muestra la siguiente imagen.



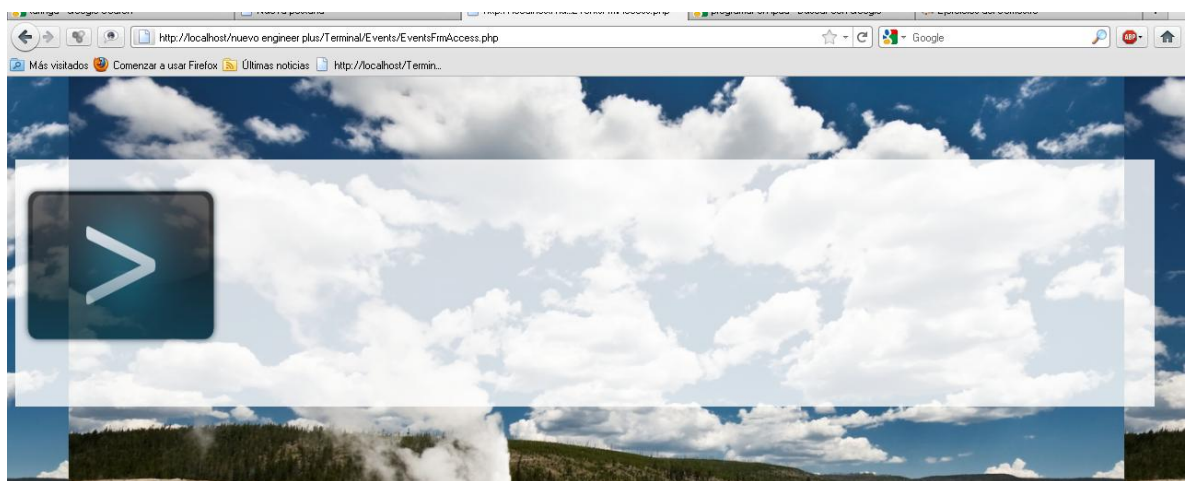
Donde se observara el formulario inicial de ingreso a la aplicación. Inicialmente observará el componente Session donde deberá digitar un usuario y una contraseña válidos, para el ejemplo se usará:

Nick: admin (administrador del sistema)

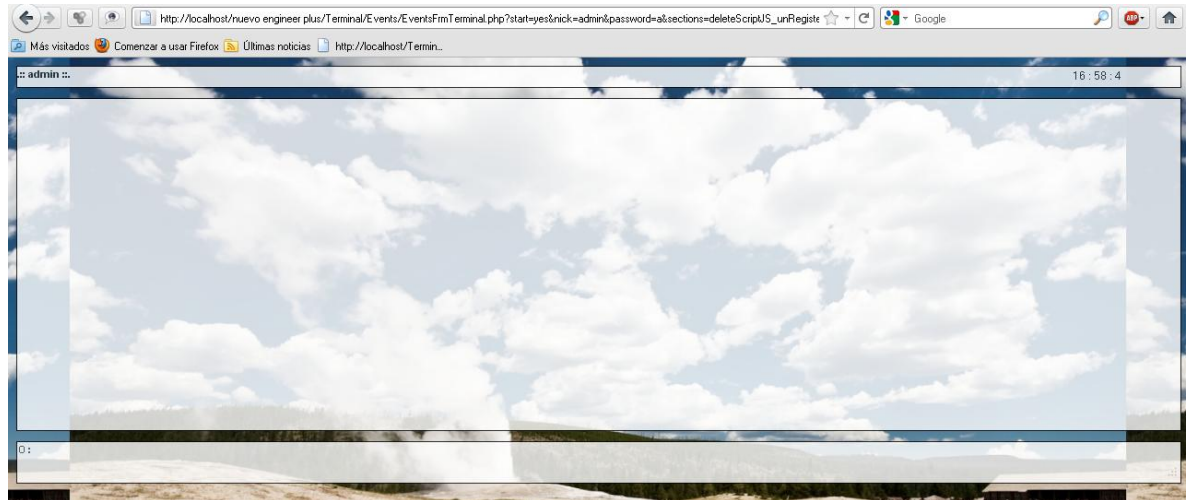
Password: Engineer

Nota: Inicialmente el prototipo del software no permitirá el registro de los usuarios.

3. Después de haberse logueado el usuario administrador visualizará la siguiente imagen.



Deberá dar click sobre el icono para acceder a la aplicación. En un determinado momento visualizará lo siguiente.



Con esta visualización el usuario ya tendrá acceso al aplicativo y obtendrá todas las funcionalidades y servicios que le puede prestar.

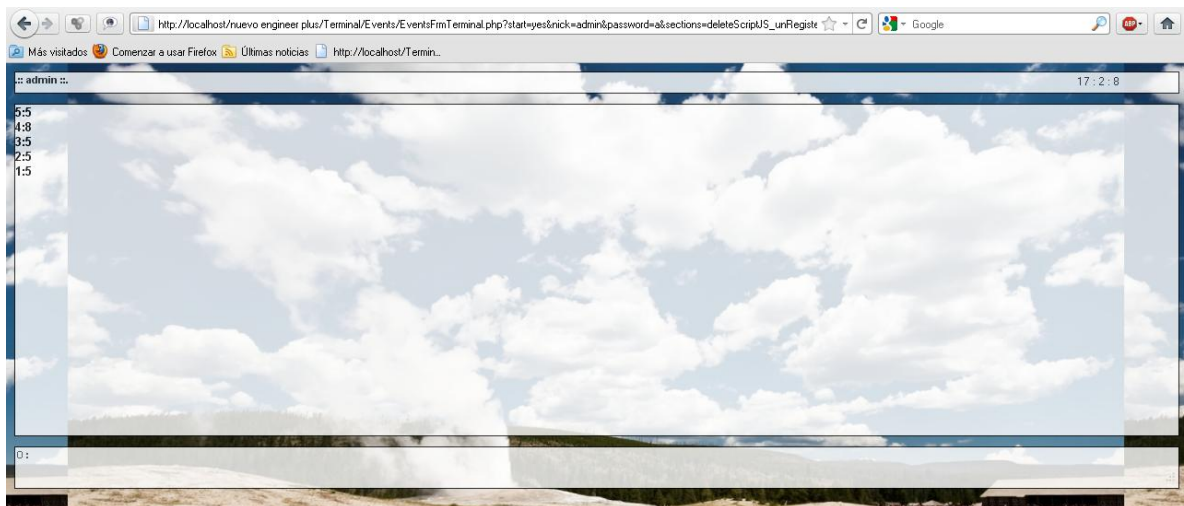
4. Funciones Intermedias Del Componente Terminal Engineer+

En esta sección se centra en las funciones que la aplicación puede prestar a los determinados usuarios, examinando paso a paso los servicios que esta puede prestar.

1. Como se había mencionado anteriormente Terminal consta de tres partes, en el siguiente ejemplo se observará la inserción de datos, lo primero que se debe hacer es colocar diferentes números en el área inferior y presionar la tecla Enter.



En un momento se visualizará lo siguiente:



Se observara que la bandeja intermedia de Terminal se empezará a llenar de los números ingresados con un comportamiento RPN y además en forma de un Stack el cual almacenará los números para el posterior procesamiento.

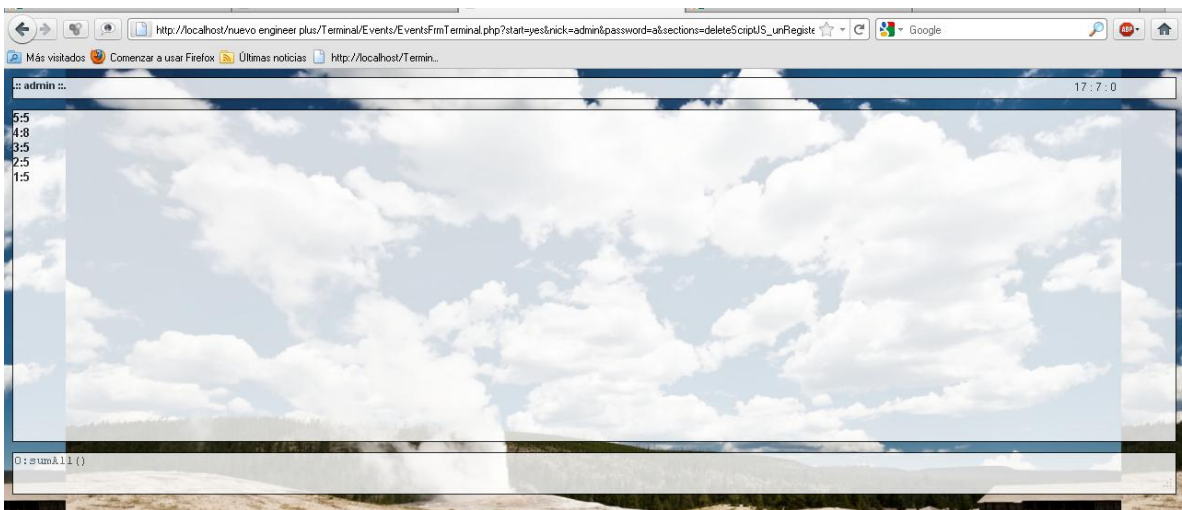
2. Realizar cálculos matemáticos: se deben ingresar los datos en el orden que se desee, repitiendo el paso anterior.



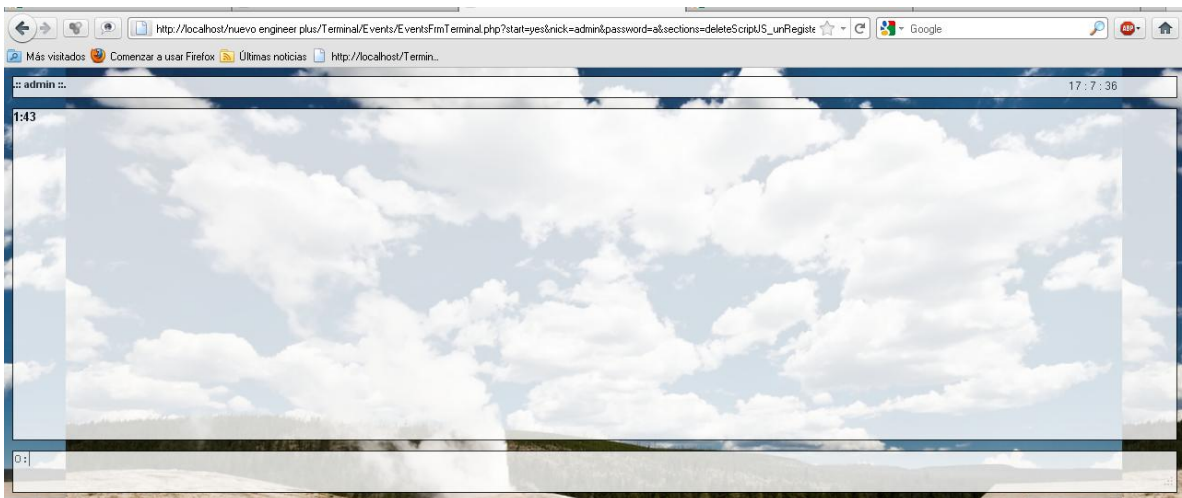
Luego insertar el comando respectivo ([Comando de ayuda para el usuario enlace](#)) de acuerdo a la operación que se desea realizar y presionar Enter, en el siguiente ejemplo se visualiza la inserción y posterior ejecución del comando sumAll (), el cual permite la suma de los datos contenidos.



Se visualiza la inserción de datos y el ingreso del comando.



Luego de insertar el comando se observa la ejecución de la suma de los números contenidos.

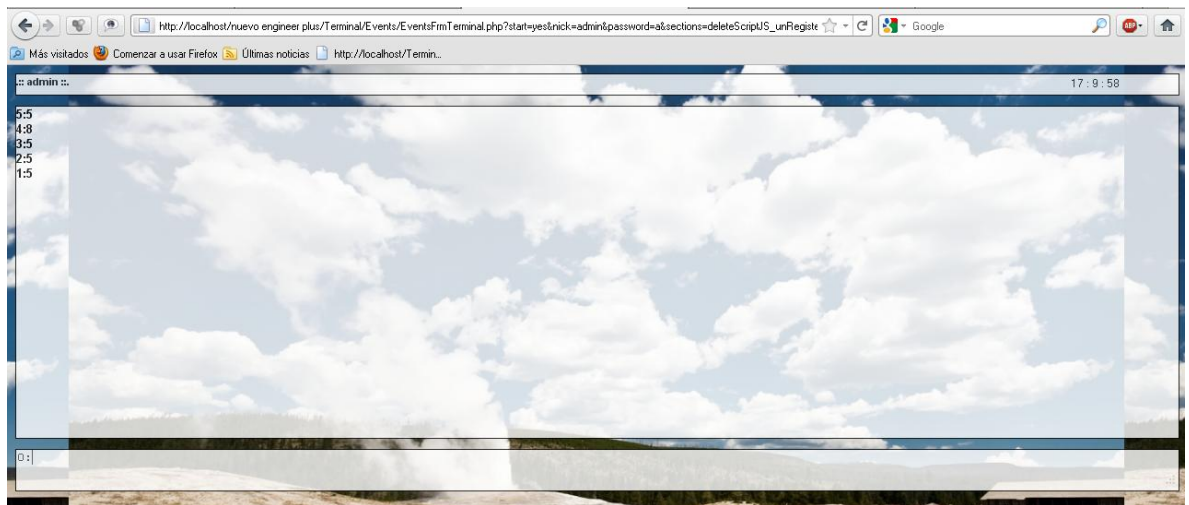


Observamos cómo fueron procesados los datos por el componente.

3. En el siguiente ejemplo se observará la ejecución de una división. Lo primero que se debe hacer es la inserción de los datos.



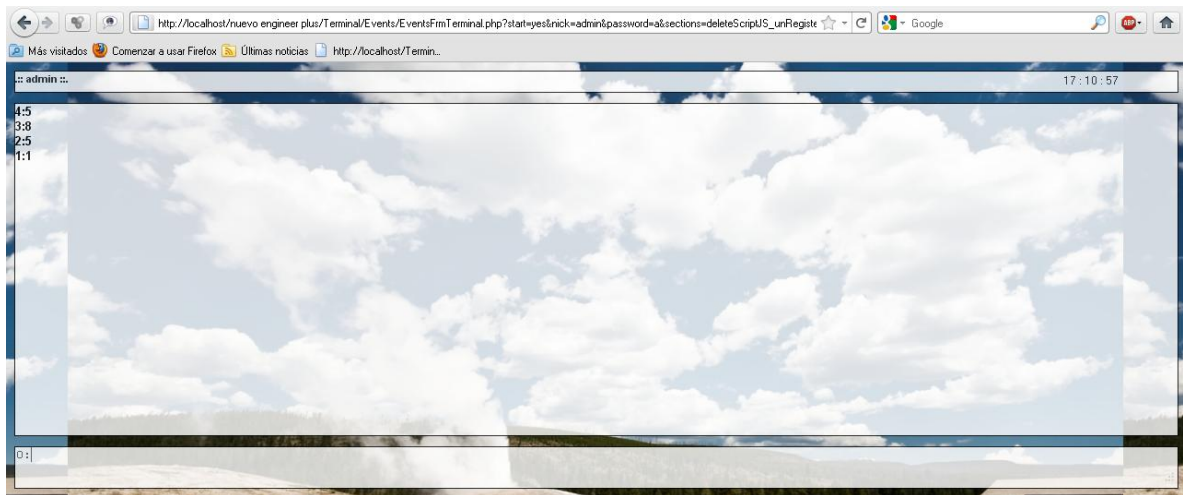
Luego se visualizará los datos contenidos en la pila de datos, en el orden que se ingresaron



Luego se insertara el comando `div()`, el cual dividirá los dos últimos números ingresados.



Por último se visualiza la respuesta de la operación que se ejecutó.

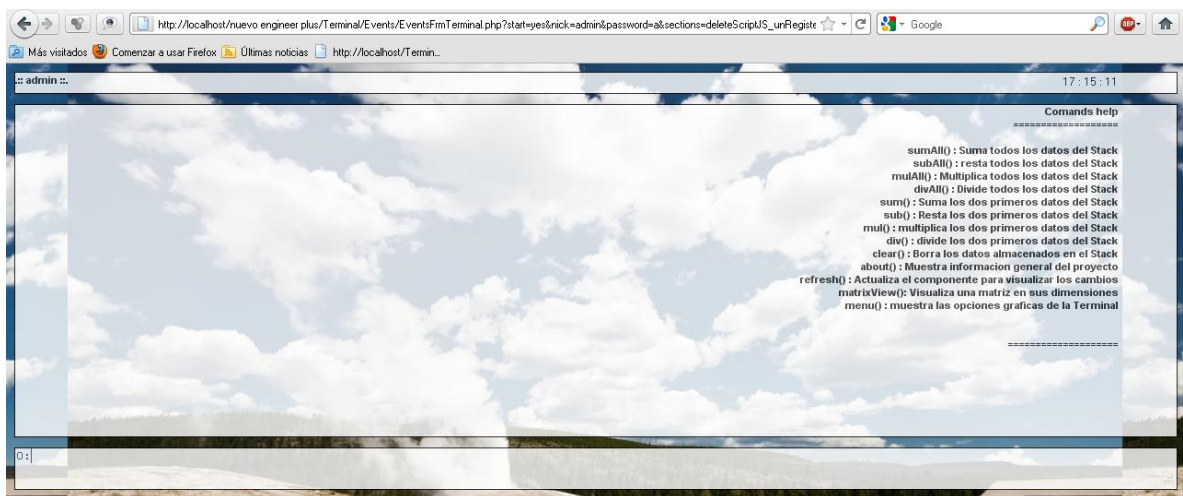


4. Comando de ayuda: El usuario contará con un comando de ayuda para conocer todos los comandos que hacen parte del componente Terminal donde se muestra la sintaxis y una breve descripción de la función del respectivo comando.

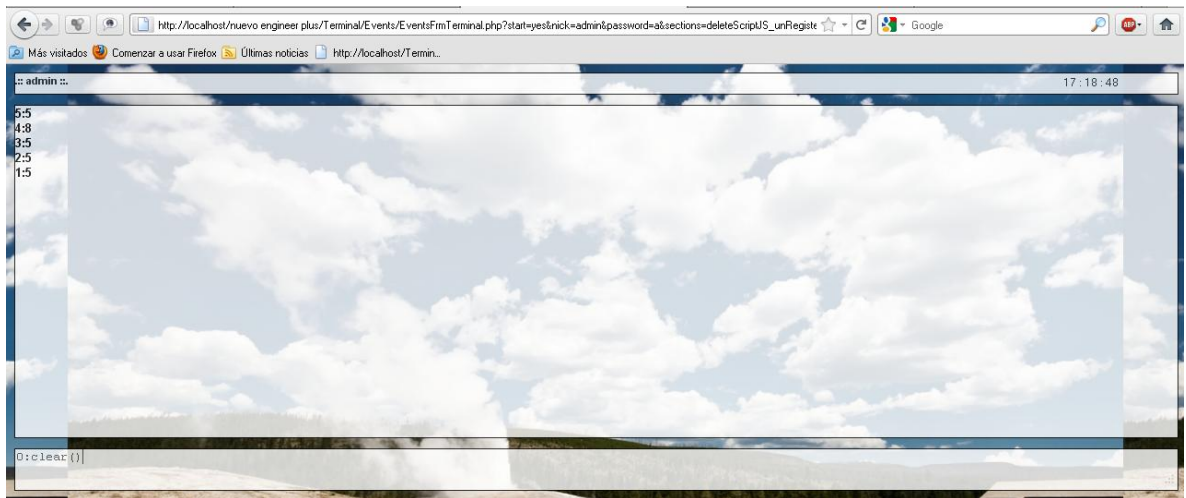
Para esto debe escribir el parte inferior `help()` y presionar la teclas Enter.



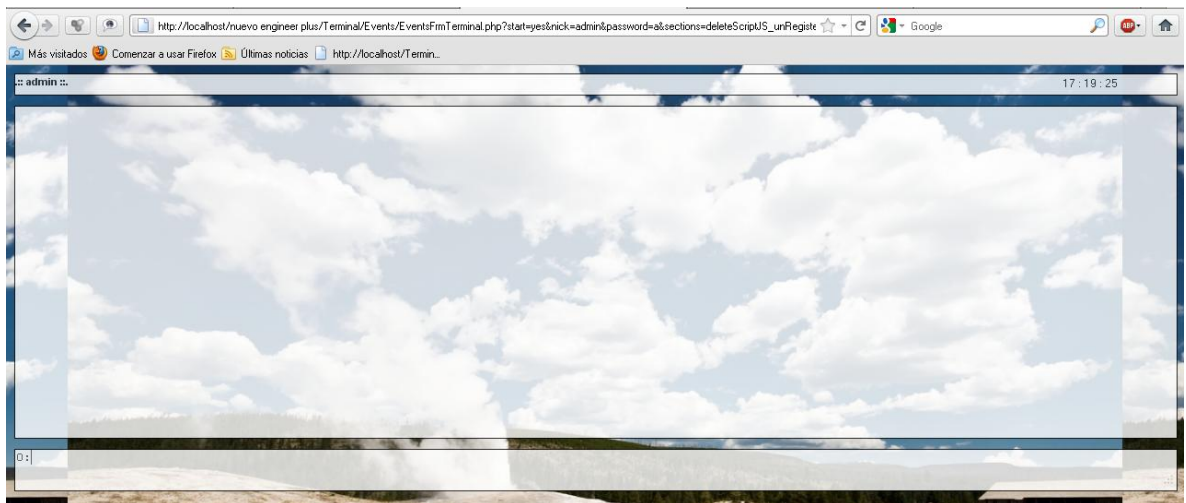
En la parte intermedia observará los comandos y sus funciones.



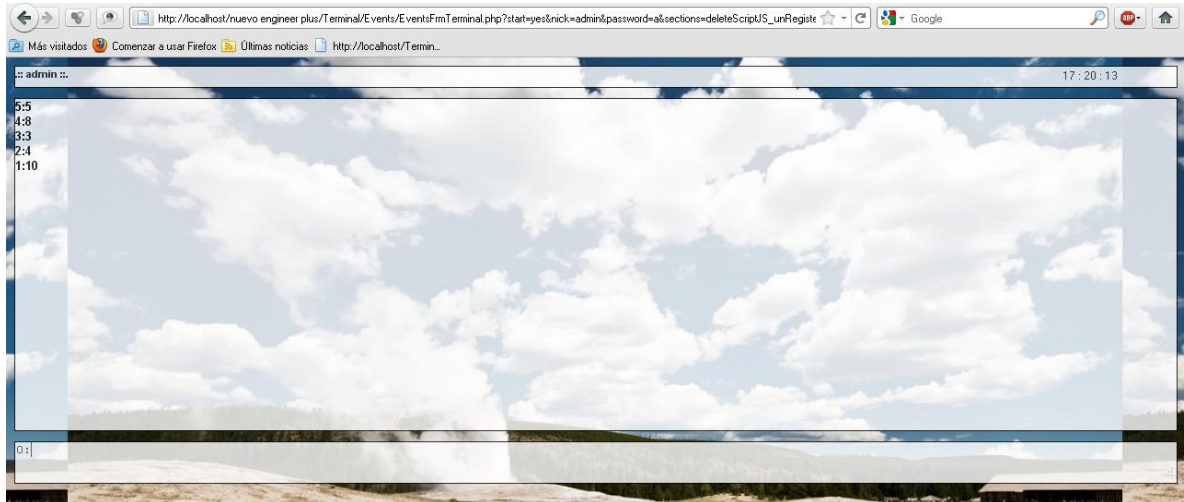
4. Si desea borrar los datos contenidos en el área de resultados, ingrese el comando `clear()` y presione Enter.



En un instante observará que los datos han sido borrados y se podrá insertar información nuevamente.



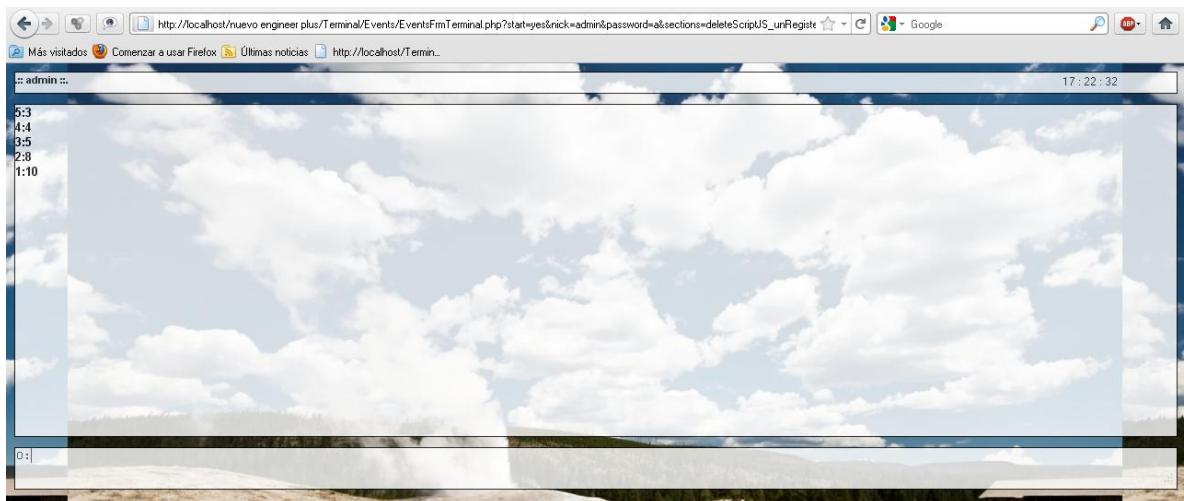
5. Para ordenar los datos contenidos en la Terminal: primero se deben ingresar los datos que se requieran.



Luego se debe ingresar el comando `sort()` y presionar la tecla Enter.



Después se visualizará la ordenación de los datos contenidos:



Cálculos con Matrices: Para el ingreso de matrices a la Terminal se debe seguir la siguiente estructura:

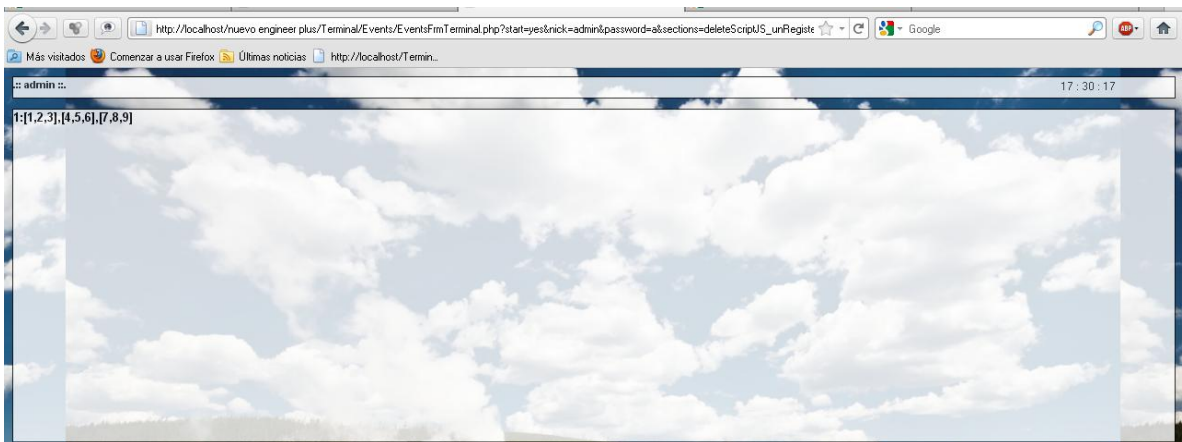
1- Ingresar los números dentro la matriz como se muestra a continuación:

[1,2,3],[4,5,6],[7,8,9]

En el ejemplo será un matriz de 3*3 donde se contendrán los números del 1 al 9, se debe proceder la inserción de la matriz de la forma como se encuentra en la parte superior, colocando la anotación en la parte inferior y luego presionar la tecla Enter así como se muestra en la siguiente imagen:



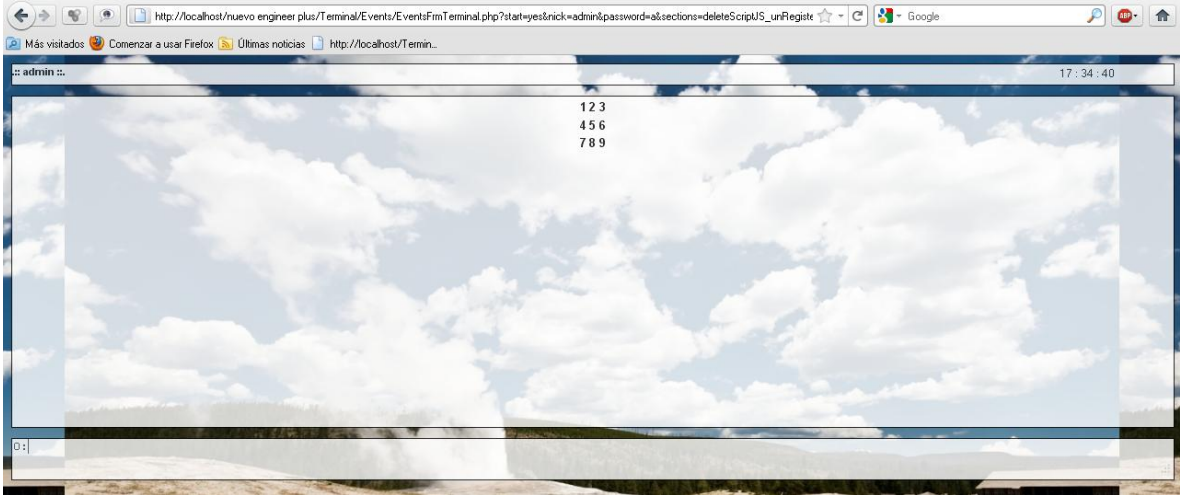
Se visualizará lo siguiente.



Luego se insertara el comando `matrixView()`, el cual permite que la matriz se observe con sus determinadas dimensiones, para esto se insertará el comando en la parte inferior y presionar la tecla Enter.



Se visualizará la matriz en su determinada forma con la cual se procederán al cálculo de operaciones matriciales.



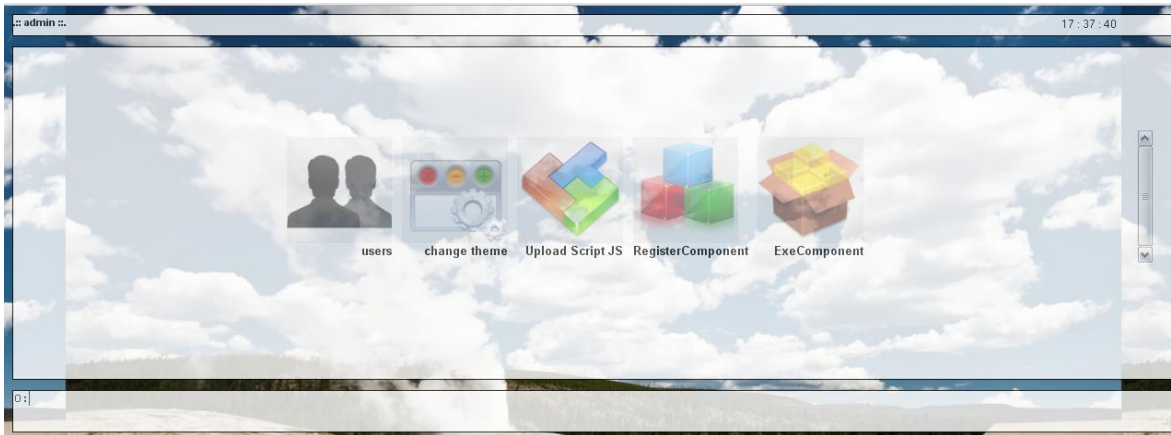
5. Funciones Avanzadas Del Componente Terminal

La Terminal cuenta con una serie de funcionalidades que son muy útiles, ya que permite la interacción con el usuario a continuación observaremos detalladamente unas de ellas.

Ingresa el comando `menu()` en el área de inserción de comandos



Después de procesado se desplegará una serie de iconos en el área intermedia de la Terminal



1. El primer icono `users` es donde se controla la administración de los usuarios, pues permite que se añadan los determinados usuarios del sistema, en el ejemplo el usuario `admin` puede acceder a esta área ya que tiene los permisos. A continuación debe dar click sobre el icono y se procederá al registro de un nuevo usuario sin derechos de administrador.

Para esto debe dar click sobre el icono.



En la parte superior se encuentra el formulario de registro donde se debe llenar indicando que el rol y los datos.

id	Nombre	Apellido	Nick	Telefono	Direccion	Email	Rol
11111111	englinx	frimost	admin	22222222	asasas	englinx@gmail.com	Administrador
33334	admdassin	adadmin	admin	222	sdsd	eee@sas.com	usuario

En la parte intermedia se muestra los usuarios que están registrados, además de sus datos y el rol con el que pertenecen al sistema.

Cuando el usuario se haya registrado correctamente obtendrá el siguiente mensaje.

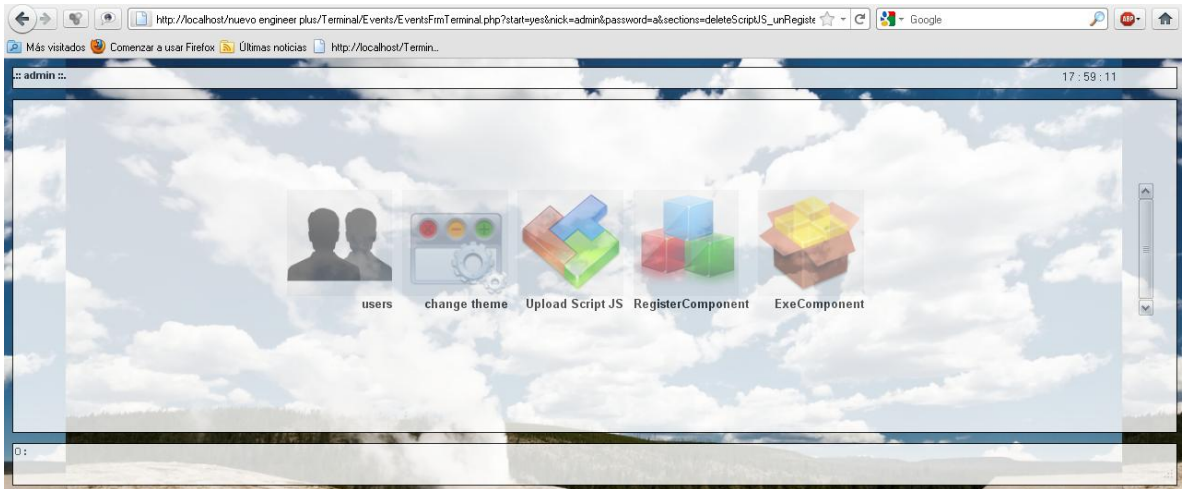
A la derecha se encuentran unos iconos que permiten la edición, borrado y obtener información sobre el usuario.



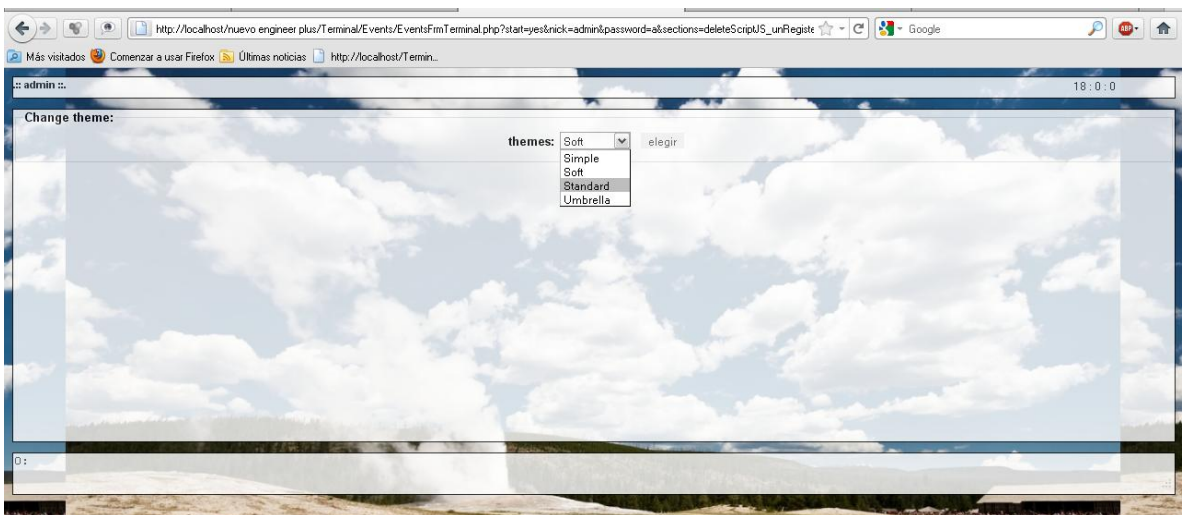
Después de ingresar los datos se presionará click en el botón registrar y automáticamente el usuario ya podrá ingresar a la aplicación con su Nick y password que haya elegido.

En determinado caso cuando el usuario no es administrador observará que al acceder al menú no tendrá esa opción, puesto que no tiene los suficientes permisos.

2. El siguiente icono change theme tendrá la funcionalidad de permitir cambiar el tema o presentación de la Terminal de acuerdo al gusto del usuario, si lo desea clickee sobre este y obtendrá lo siguiente:



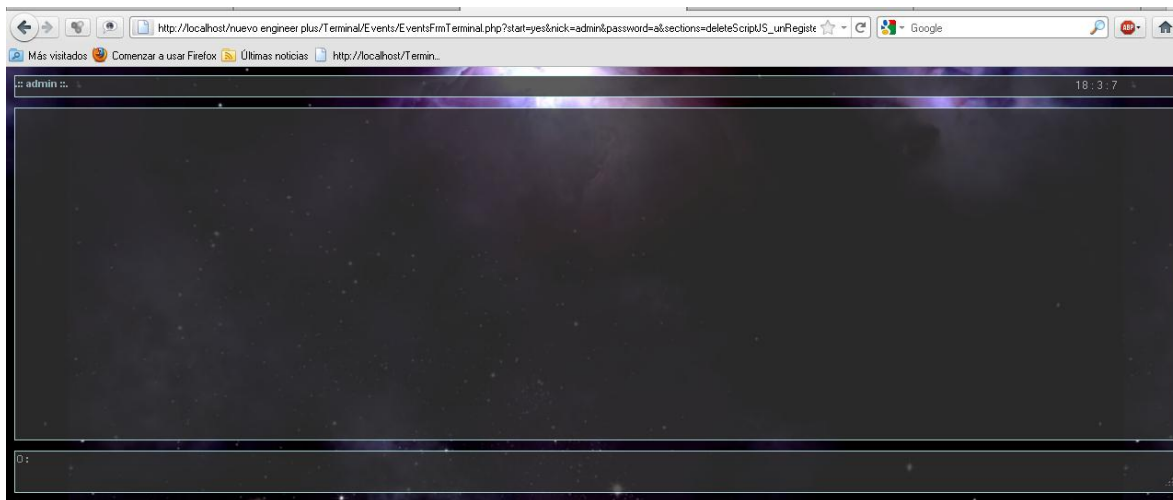
En esta vista podrá elegir un tema, dando click en la barra desplegable allí escogerá la presentación y luego dará un click sobre elegir.



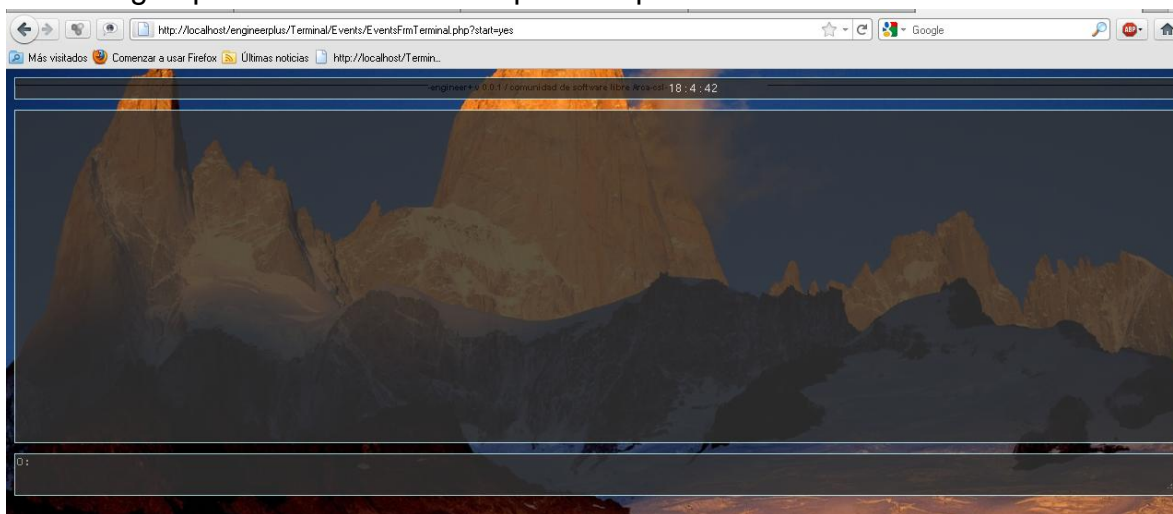
Luego en la bandeja inferior deberá ingresar el comando refresh(), el cual permite guardar los cambios almacenados y luego debe presionar la tecla Enter



En un momento la presentación de la Terminal cambiará de acuerdo a la opción que se halla seleccionado.

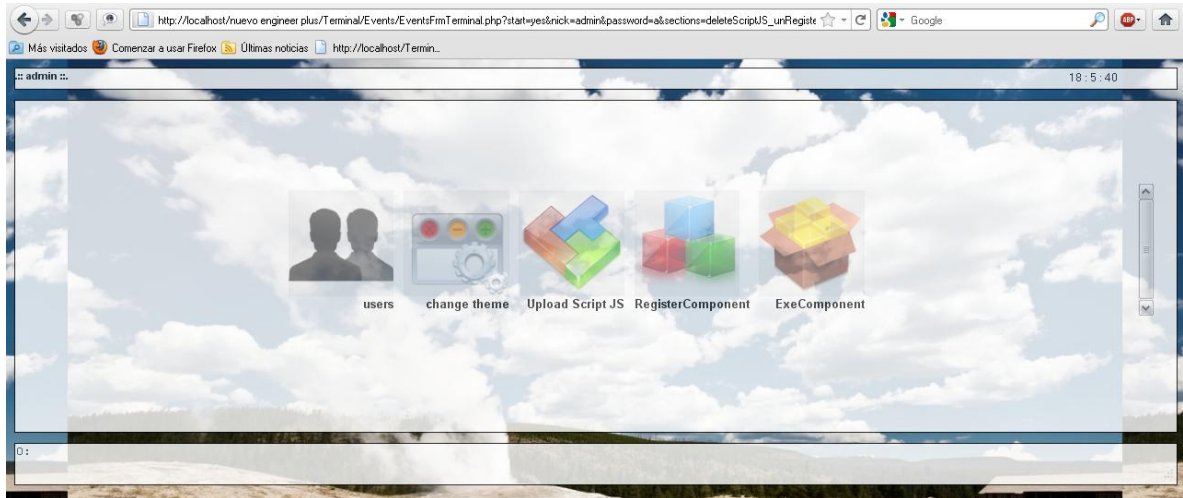


Esta imagen pertenece a otro tema previsto para la selección:

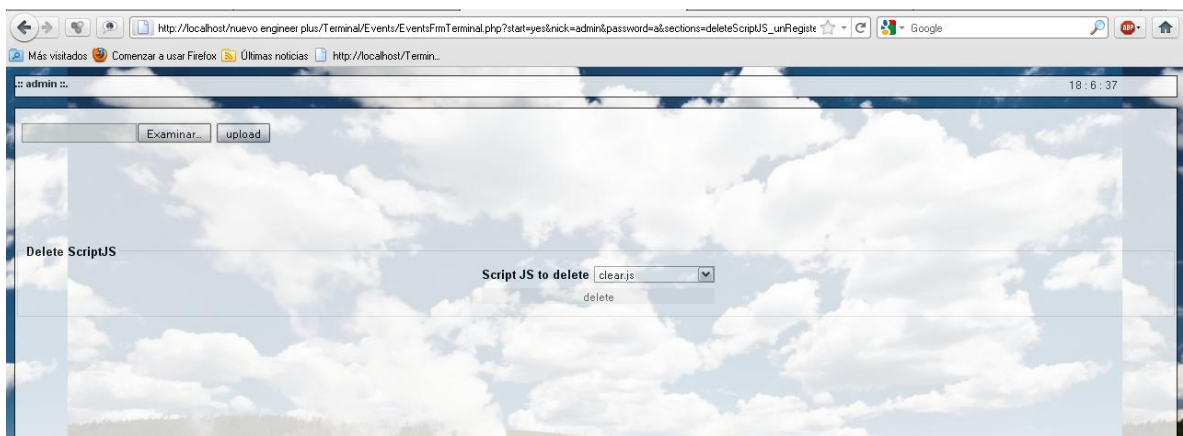


El usuario podrá escoger el tema más adecuado.

3. El tercer icono tiene la función de permitir cargar archivos de extensión JS (javascript file), para el posterior procesamiento en la Terminal y así cualquier persona puede hacer un script y comprobar el funcionamiento.

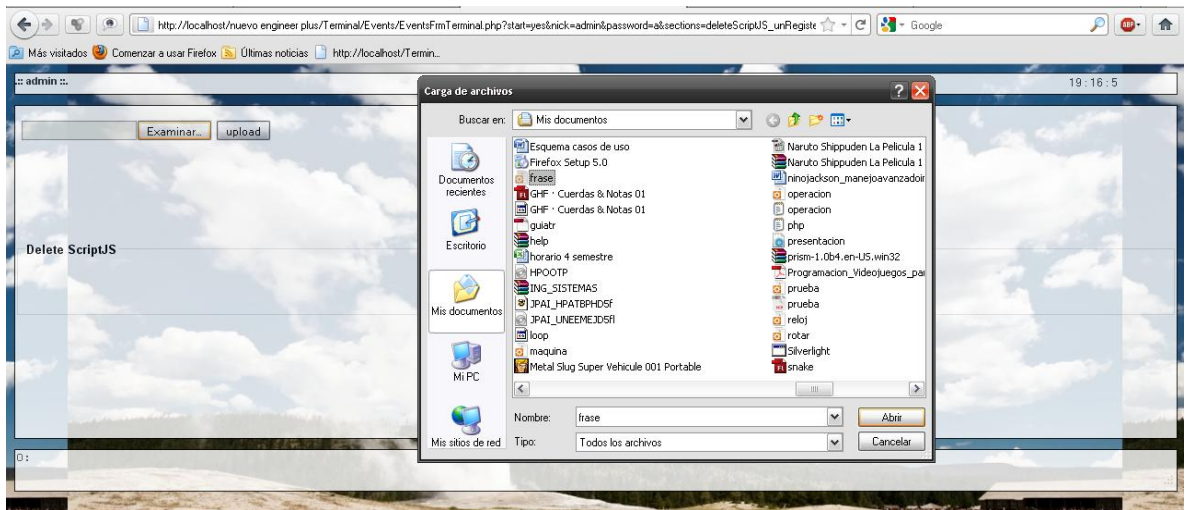


Clickee sobre el icono y obtendrá lo siguiente.

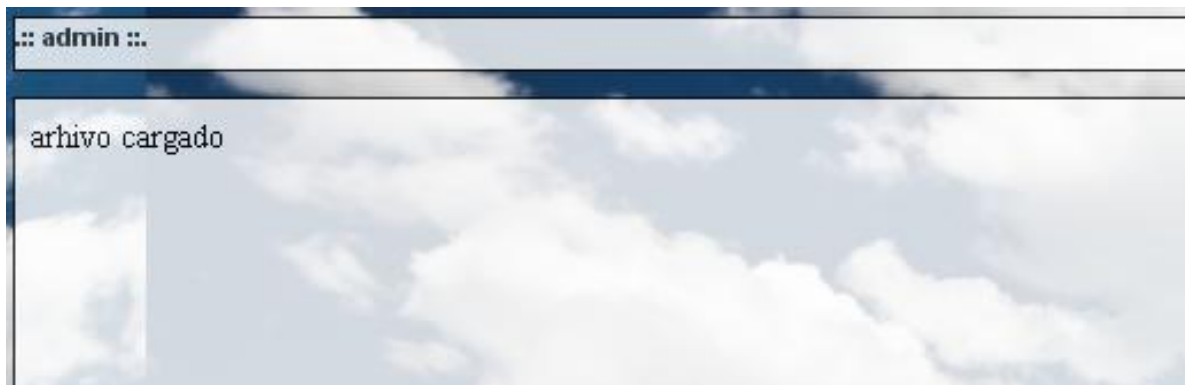
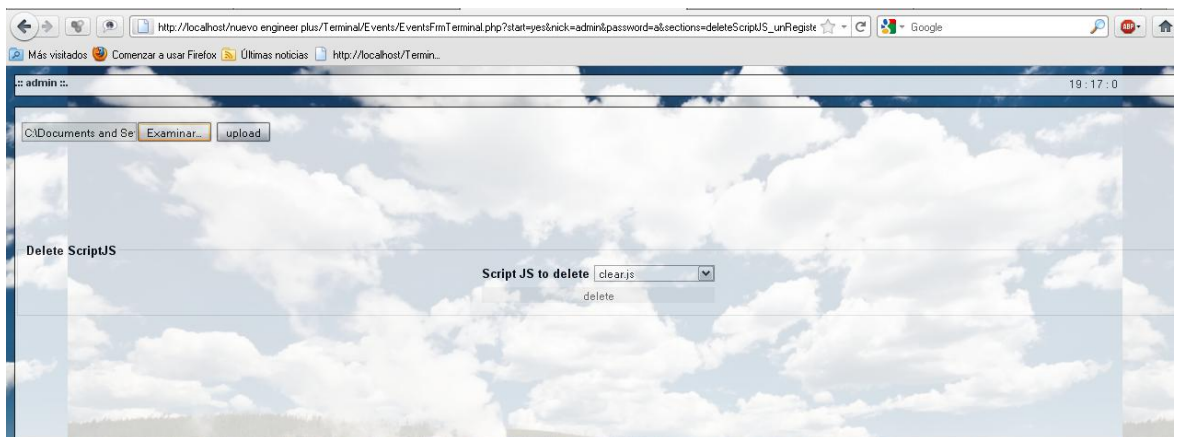


En la parte intermedia aparecerá una serie de menú, en el área de superior se cargaran los scripts.

Luego de un click sobre Examinar y escoja el archivo .js en la ubicación de su computadora.



Luego de click sobre abrir y diríjase a upload y de un click sobre este.



Cuando haya cargado correctamente el archivo obtendrá un mensaje que dice "archivo cargado" y así estará listo para el procesamiento en la Terminal.

Luego se insertara el comando refresh() para que los cambios sean guardados.



- En el ejemplo anterior se incluyo un archivo .js que contiene la siguiente instrucción:

```
//frase()
function frase() {

    var tipoLetra = 'verdana'
    var tamanoLetra = '5'
    var colorLetra = 'red'

    //array de frases
    array_frases = new Array(
        "Un amigo es quien rie nuestra risa y llora nuestras lágrimas",
        "La mejor forma de conseguir un amigo es serlo",
        "Los amigos se convierten con frecuencia en ladrones de nuestro tiempo",
        "Un amigo es una persona con la que se puede pensar en voz alta"
    )

//obtiene numero aleatorio
var numero = Math.random()
numero = numero * (array_frases.length - 1)
numero = Math.round(numero)

//muestra la frase
document.getElementById("divContent").innerHTML =('<font face="' + tipoLetra + "'
size="' + tamanoLetra + "' color="' + colorLetra + "'>' + array_frases[numero] +
'</font>');

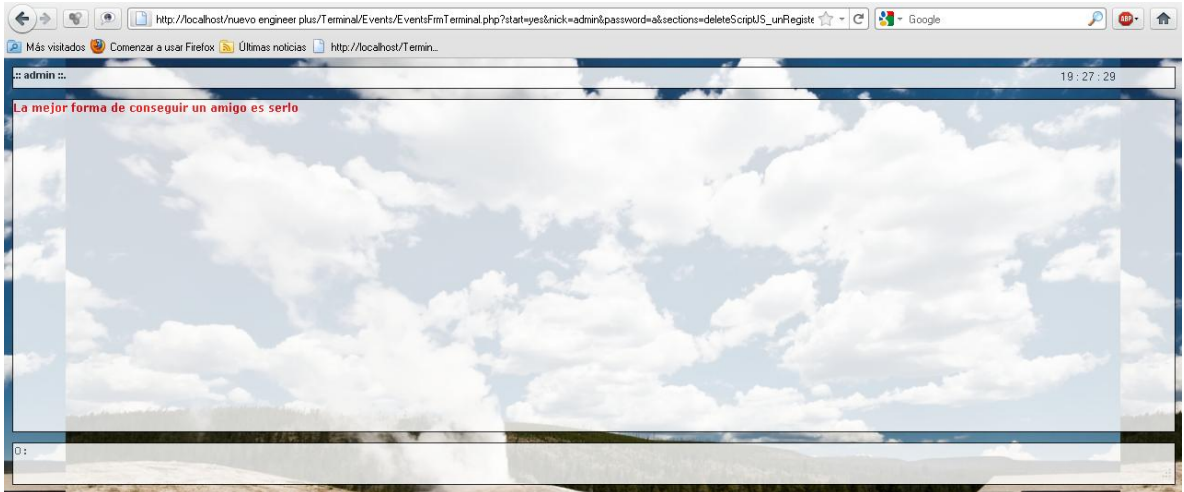
}
```

- Una vez cargado el archivo se procederá a la ejecución del script, la manera en que se procesará es escribiendo en el área de comandos el nombre del script mas dos paréntesis uno que abre y otro que cierra () y

luego presionar la tecla Enter. El comando con el cual se procesa es la anotación con la que aparece en la parte superior el script (//frase()).

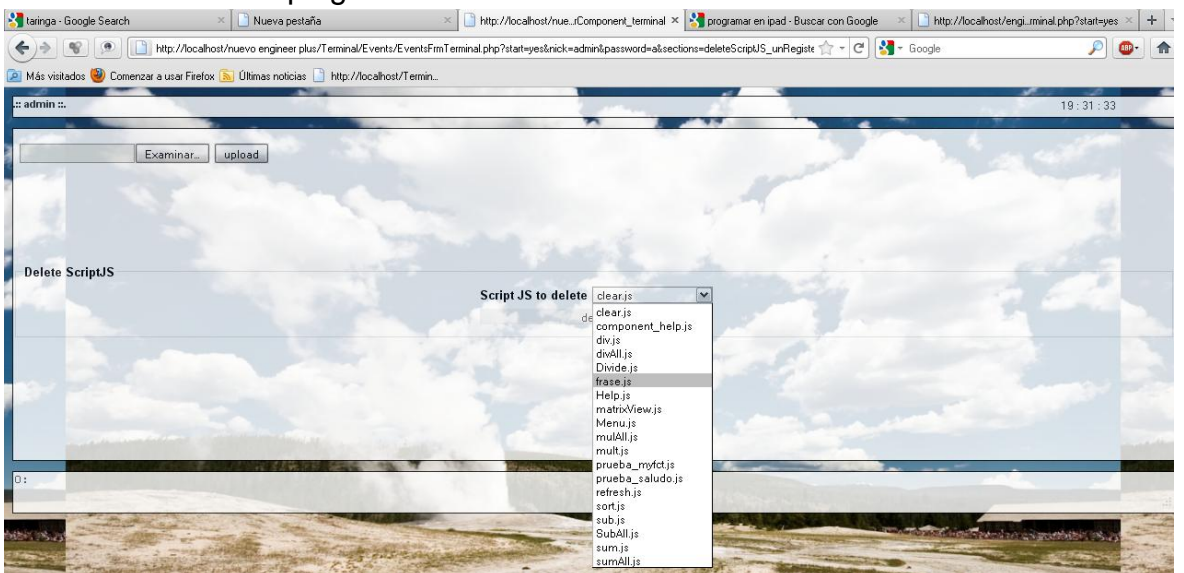


Luego se visualizará la información que se contenía dentro del script

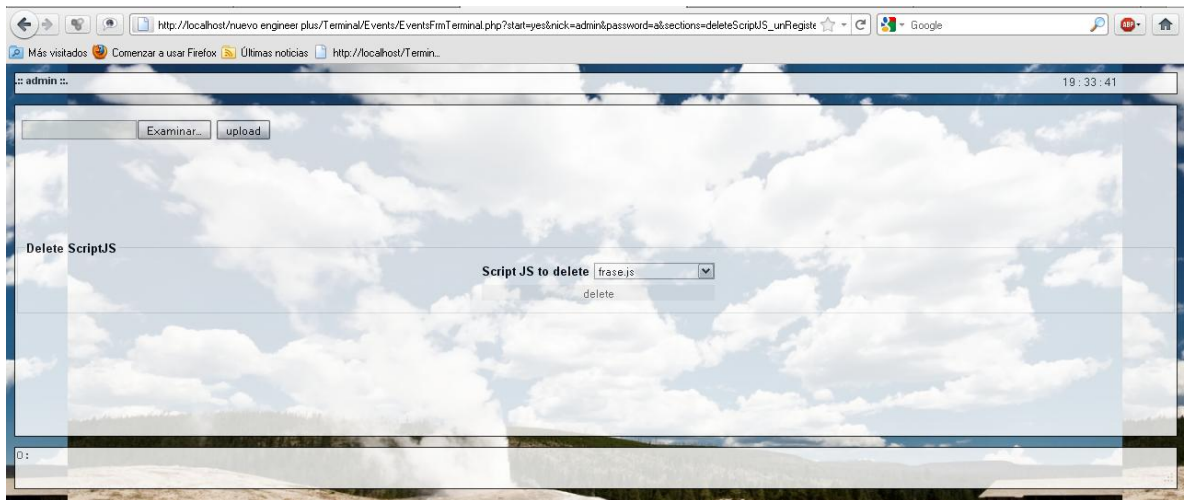


El script muestra frases aleatorias cada vez que se inserta el comando.

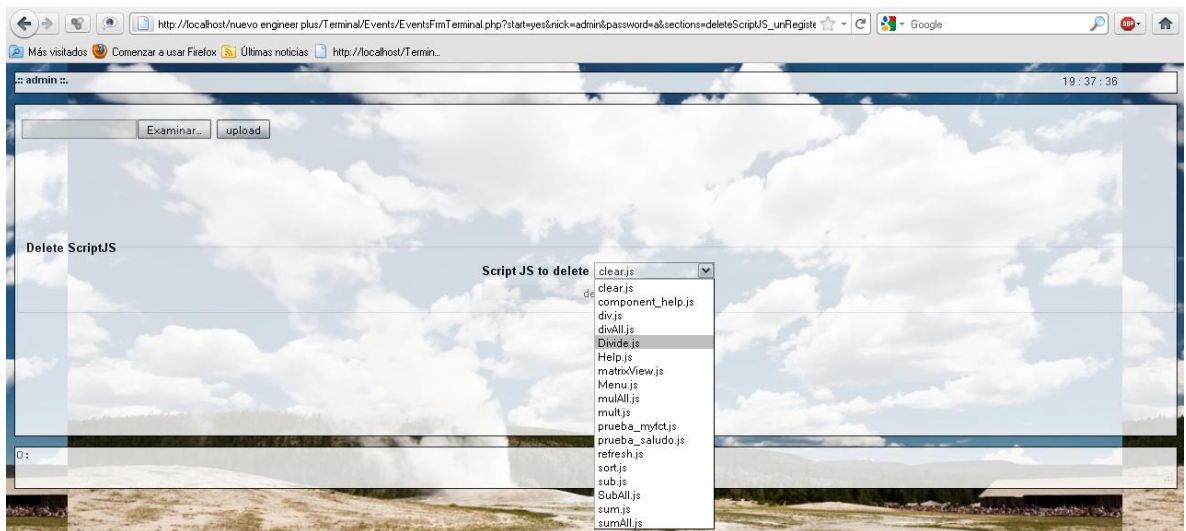
-En la parte intermedia se encuentra una sección que permite el borrado de los scripts que se han cargado a la Terminal, debe dar un click sobre la lista y a continuación se desplegará.



Luego debe seleccionar el script que desea borra y dar un click sobre delete, para el ejemplo se uso frase.



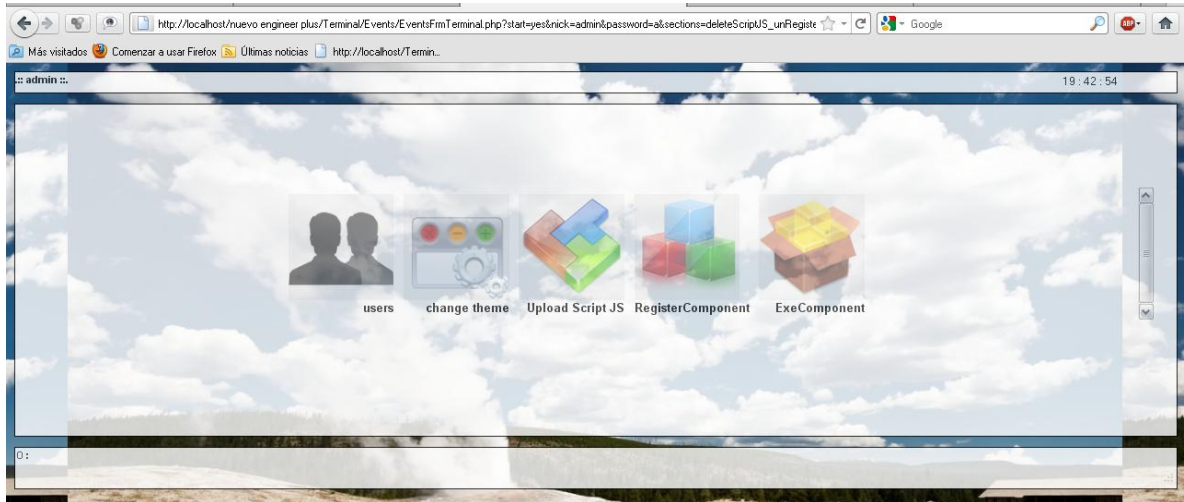
A continuación se observa que el script fue borrado.



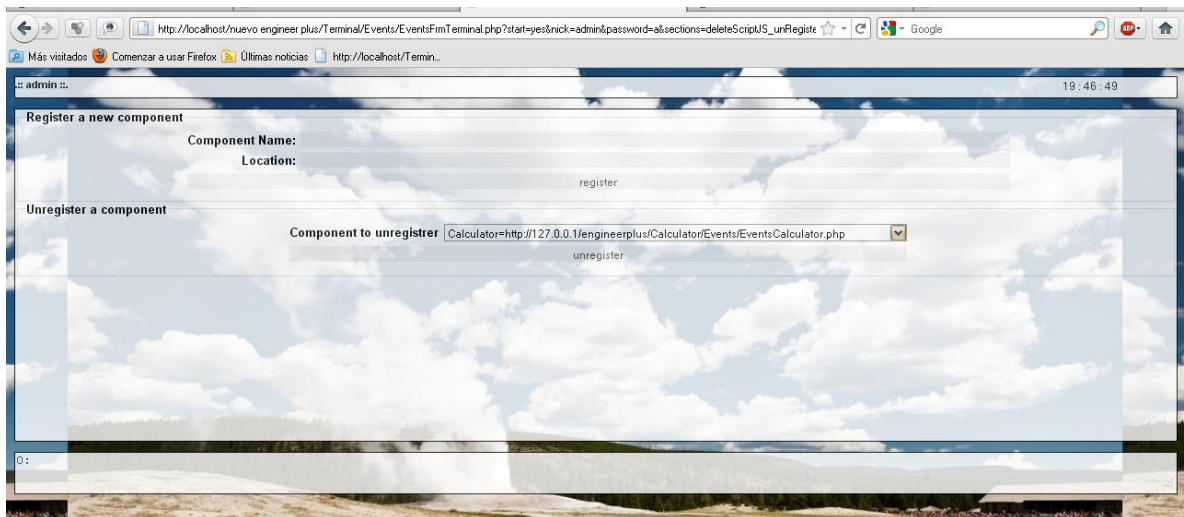
En determinado caso si un usuario no es administrador no tendrá acceso a la opción de borrado de scripts ya que no tiene los permisos necesarios además observará lo siguiente.



4. El cuarto icono tiene la función de permitir registrar y des-registrar un componente, el cual se añadirá posteriormente a Engineer+, para ingresar debe dar un click sobre el icono.



Luego visualizará lo siguiente.

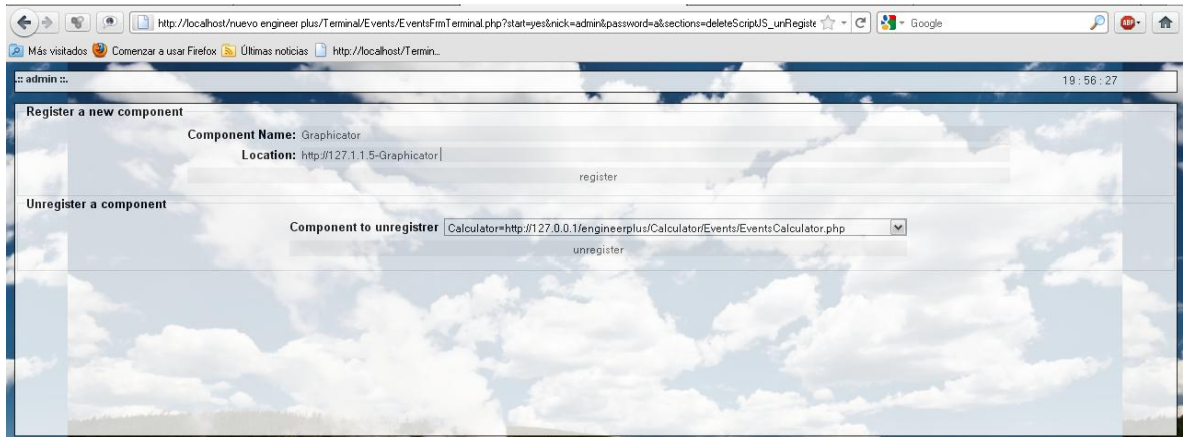


En el área superior se encuentra el formulario donde se registran los nuevos componentes, indicando su nombre y lo localización por medio de una dirección ip y en el área inferior se encuentra la sección donde se des-registran los componentes de engineer+.

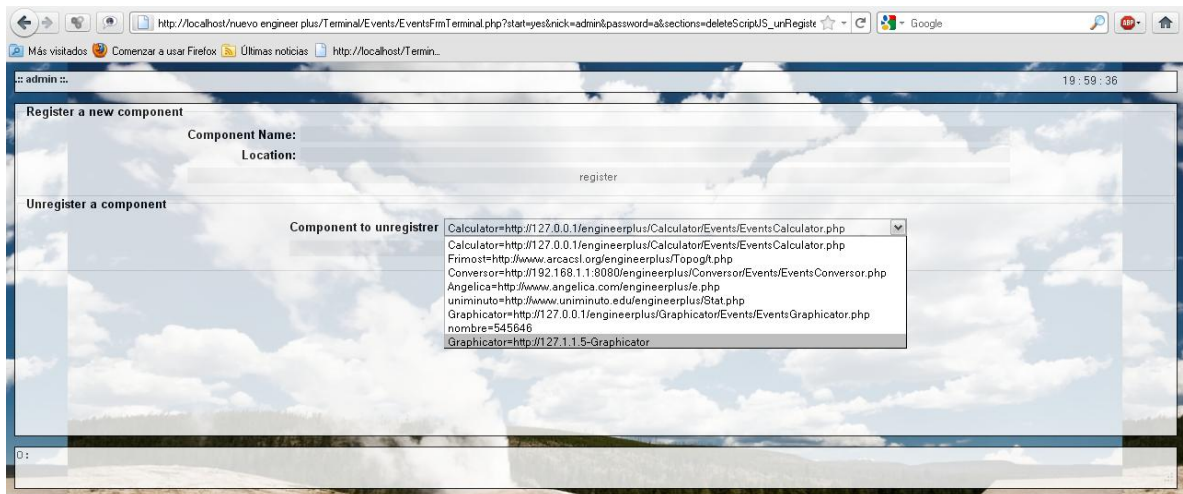
Para registrar un componente se debe indicar el nombre y la dirección ip en donde está ubicado, para el ejemplo se usaran lo siguientes datos, con esta información se llenara el formulario requerido.

Component Name: Graphicator

Location: <http://127.1.1.5-Graphicator>

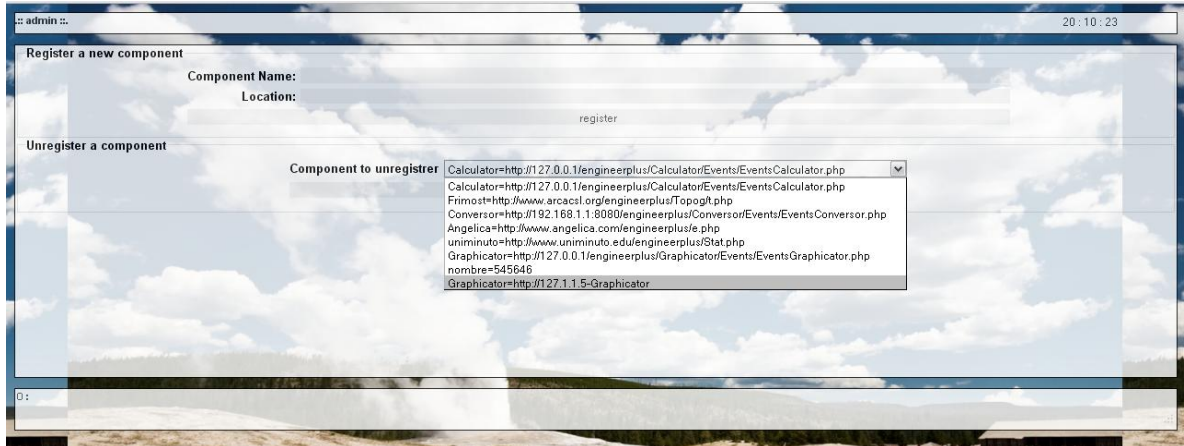


Luego se dará un click sobre register.

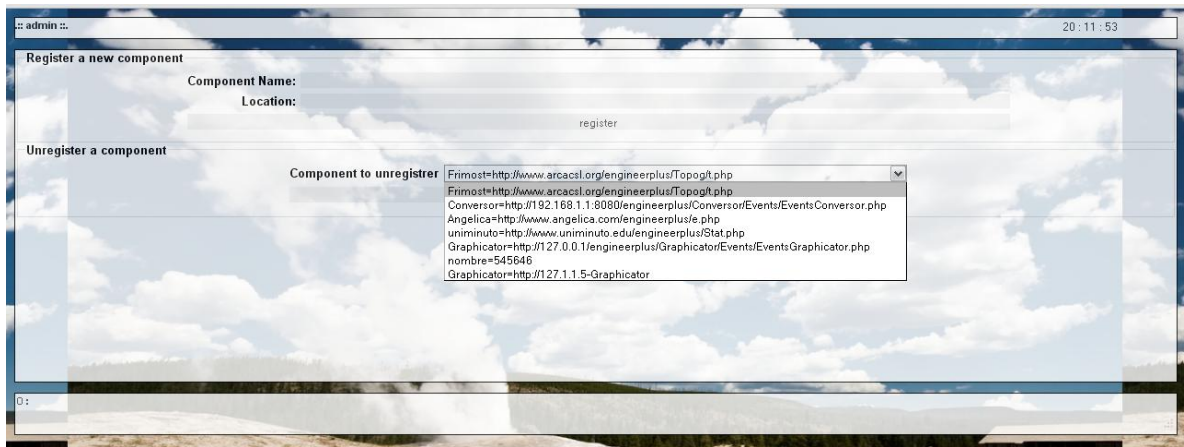


Luego en el área inferior se observa el componente registrado, con los datos con los cuales se lleno el formulario.

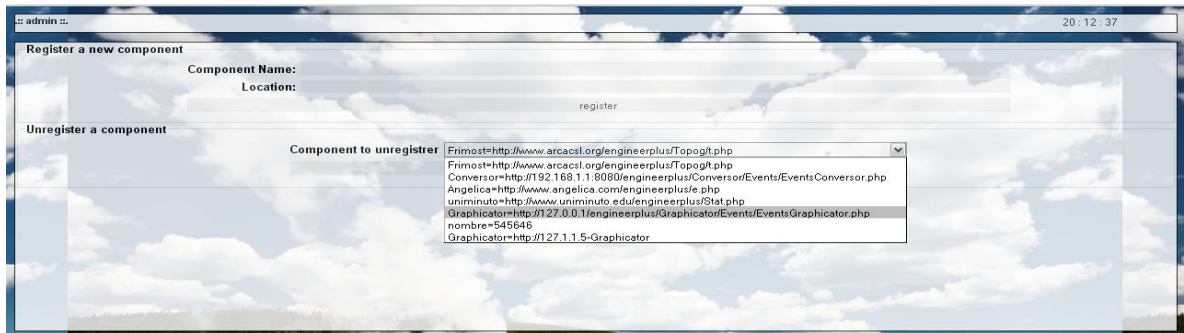
-Si en algún determinado momento se desea borrar el componente que se ha registrado se debe dirigir a la sección de unregister componet y escoger el componente que desea des-registrar.



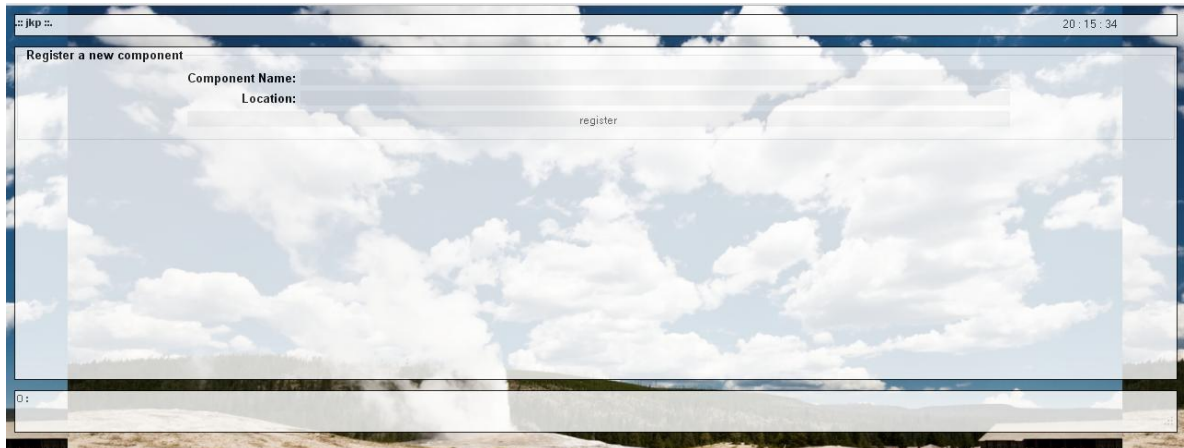
Después de seleccionarlo debe dar click en el botón unregister.



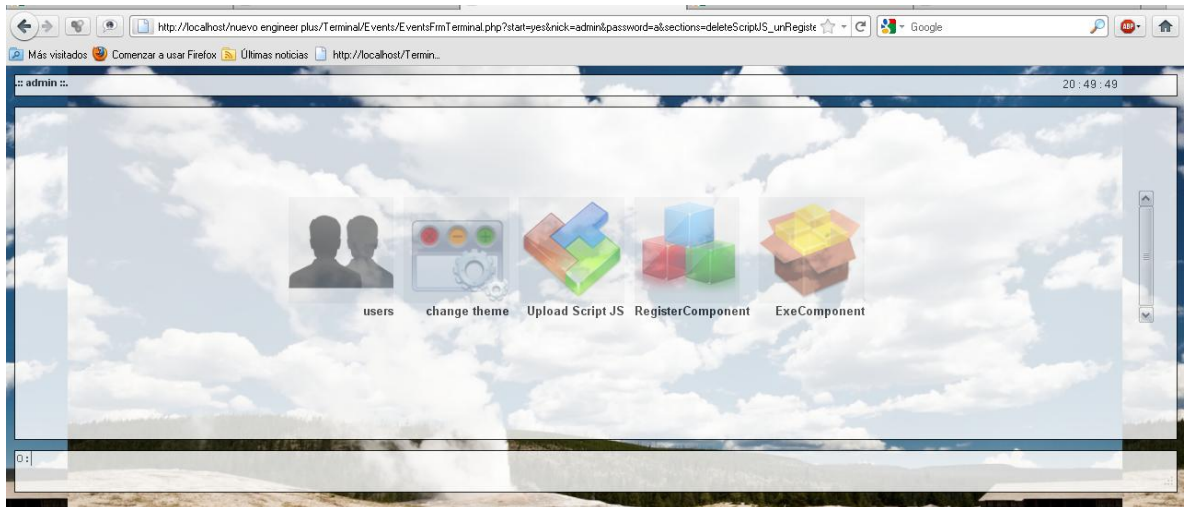
Luego visualizará que el componente ya no se encuentra en la lista.

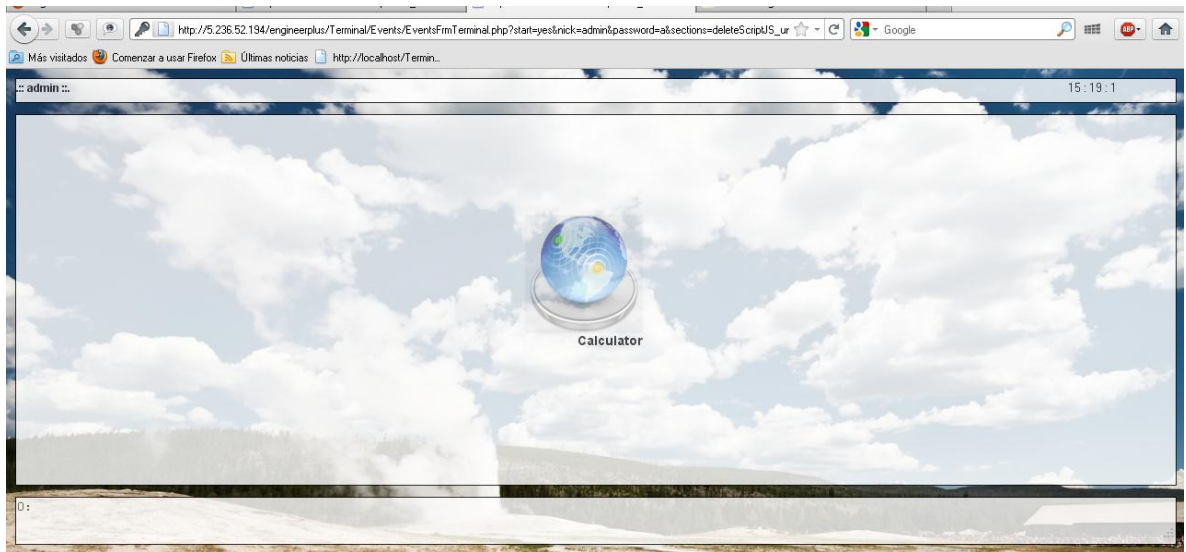


En determinado caso que el usuario no se administrador la opción de des-registrar componente no se encontrará habilitada ya que no cuenta con los permisos y se observara la siguiente imagen.



5. El quinto icono permite cargar los determinados componentes que han sido registrados donde cada usuario podrá acceder a las funciones del componente, debe dar click sobre el icono y visualizará lo siguiente.

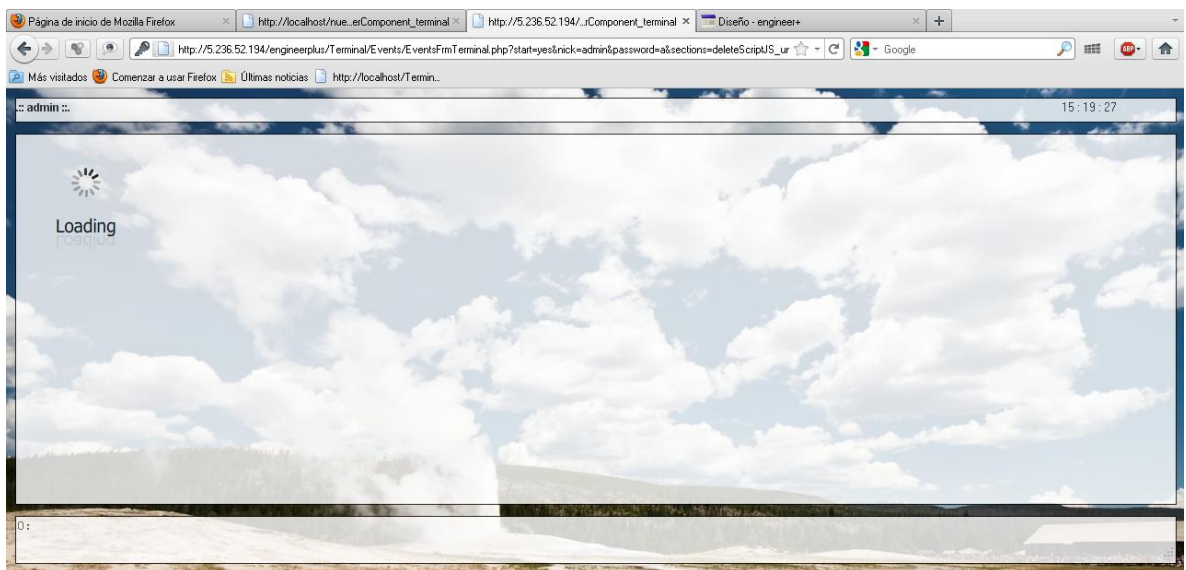




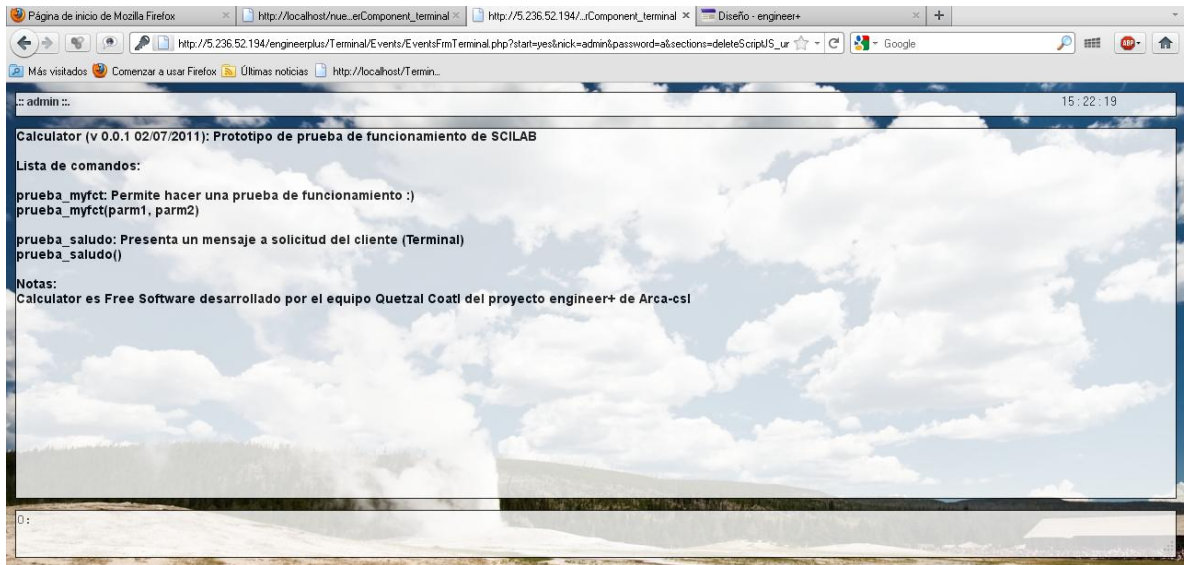
Se observarán todos los componentes registrados.

Para acceder a las funciones de un determinado componente se debe dar click sobre el que se desea trabajar y así el usuario ya tendrá las funciones del componente que ha cargado. (Para el ejemplo se usara el componente Calculator).

Cuando presione sobre el icono, tendrá que esperar un momento.



Cuando termine de cargar el componente obtendrá la siguiente visualización:



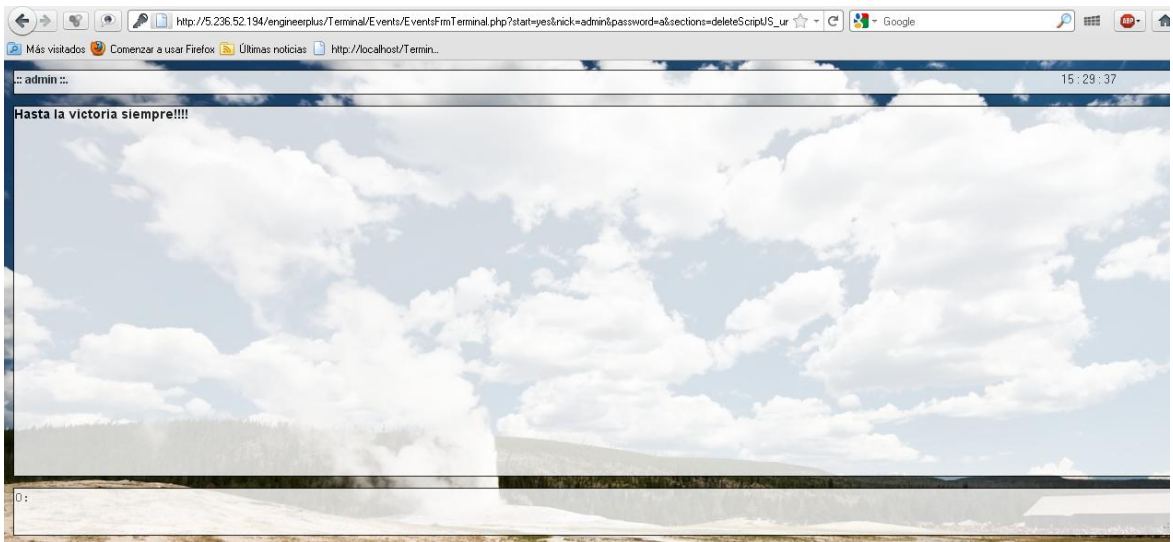
En la imagen se observa la carga completada del componente Calculator, con un mensaje de bienvenida y las funciones con las que el usuario tendrá acceso al uso de la aplicación (Para el ejemplo se observaran las funciones prueba_saludo y prueba_myfct(param1, param2), las cuales interactúan entre sí usando los 3 componentes del sistema).

-Ingrese en la bandeja inferior el comando prueba_saludo() y presione la tecla Enter.

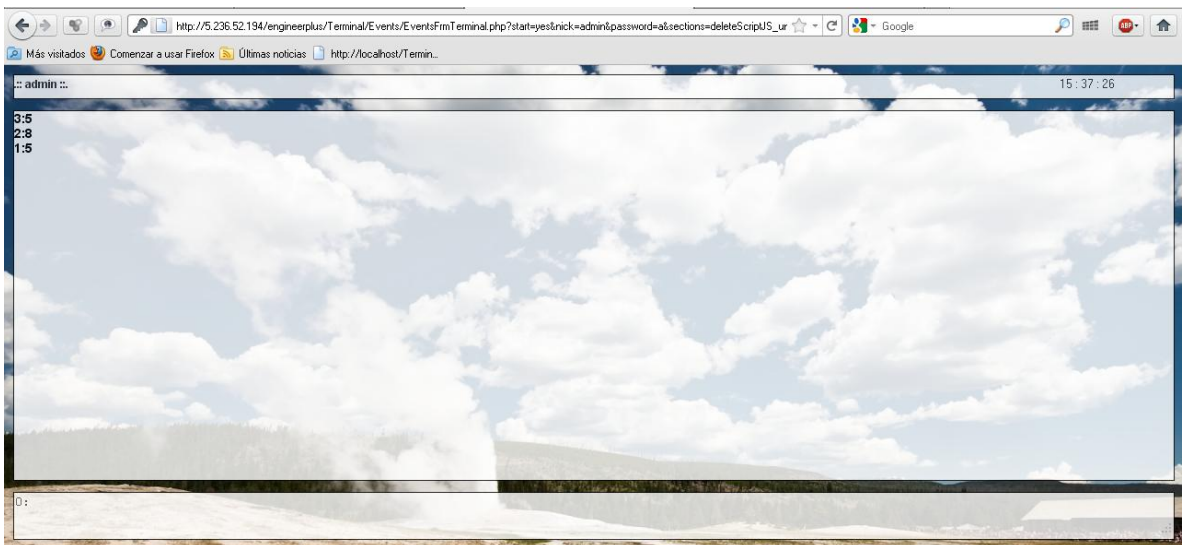


En determinado momento obtendrá la siguiente visualización, un mensaje de prueba que proviene del componente Calculator. El mensaje pertenece a una función SCI (scilab) la cual se carga a través del componente calculator.

```
function prueba_saludo()  
  printf("Hasta la victoria siempre!!!!");  
endfunction
```



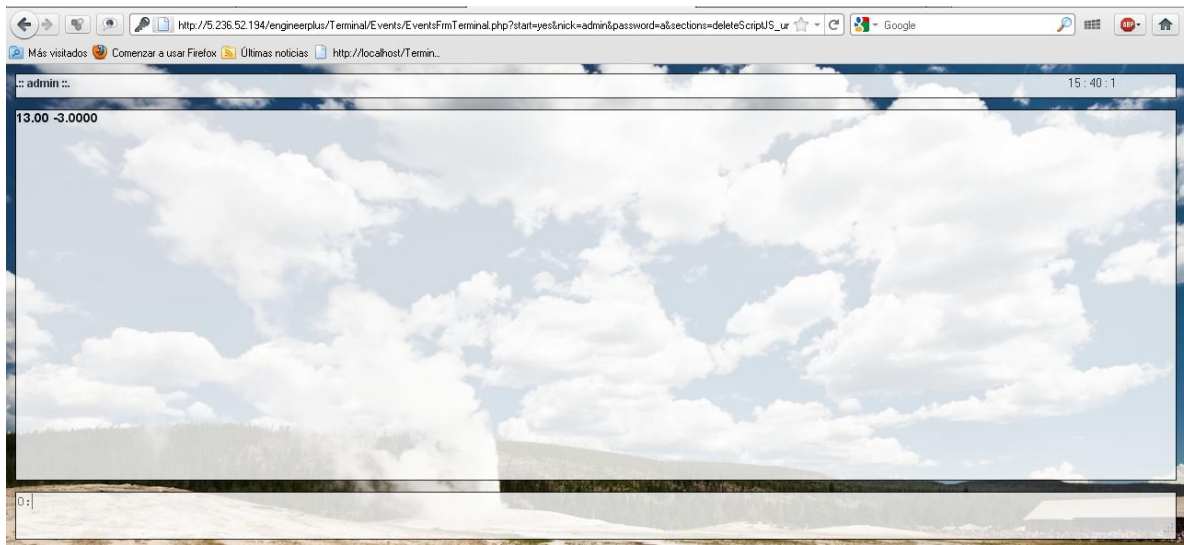
-Para la ejecución del siguiente se requiere que se ingresen algunos números a la pila, por esto debe ingresar cualquier numero y enviarlo a través de la tecla Enter.



Cuando haya ingresado los números debe ingresar el comando prueba_myfct(parm1, parm2), el cual ejecuta una suma y resta de los dos últimos dígitos contenidos en la pila y presionar la tecla Enter.



En un determinado momento se visualizará la suma y resta de los números contenidos en la pila.



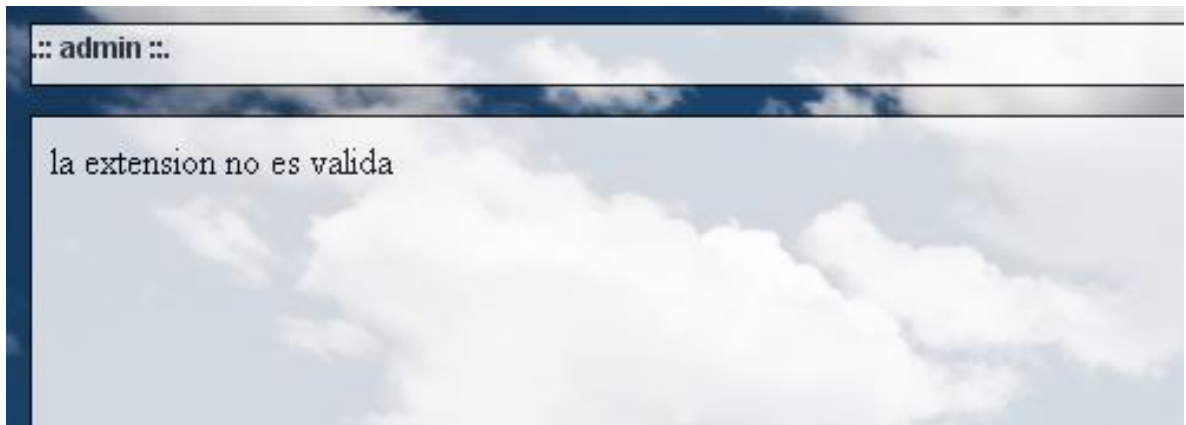
La función que se ejecuto es la siguiente:

```
function prueba_myfct(a, b)
    x=a+b;
    y=a-b;
    printf("%0.2f %0.4f",x,y);
endfunction
```

EXCEPCIONES EN LA CARGA DE ARCHIVOS

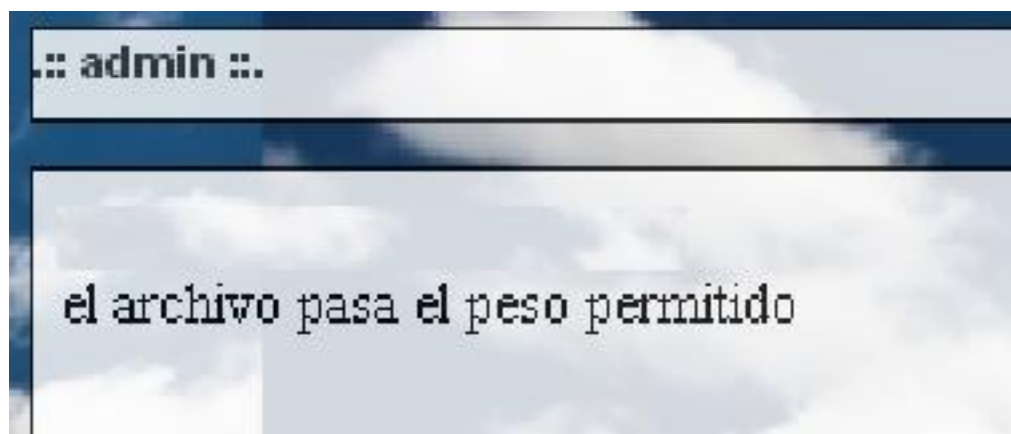
Existen unas reglas para el procesamiento de la carga de archivos ya que no se permitirá cargar archivos con extensiones diferentes a JS, ni tampoco se permitirá sobrepasar un peso permitido del tamaño del archivo.

1. Carga de archivo con extensión diferente: Si en determinado momento se escoge un archivo con una extensión diferente obtendrá lo siguiente:



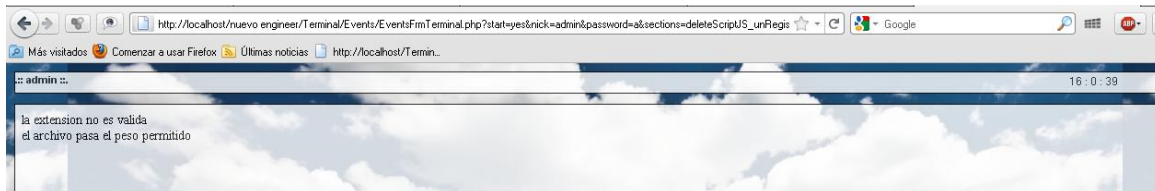
Una alerta que le informe “la extensión no es válida”.

2. Sobrepasar el peso permitido: Cuando se intente sobrepasar el peso permitido del archivo obtendrá lo siguiente:



Una alerta que le informe “el archivo pasa el peso permitido”.

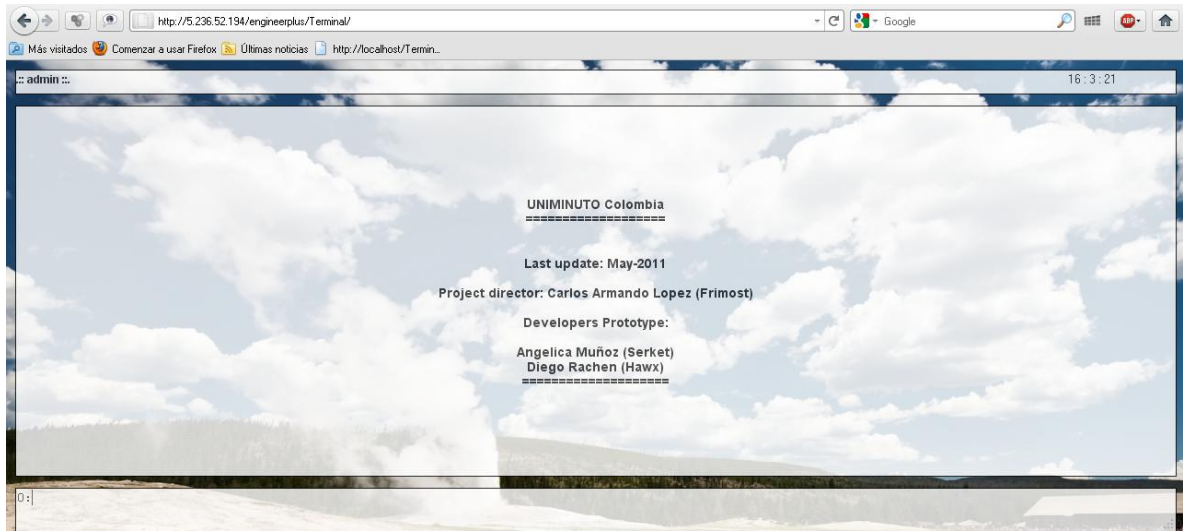
Nota: en el determinado caso que incumplan las dos excepciones obtendrá lo siguiente:



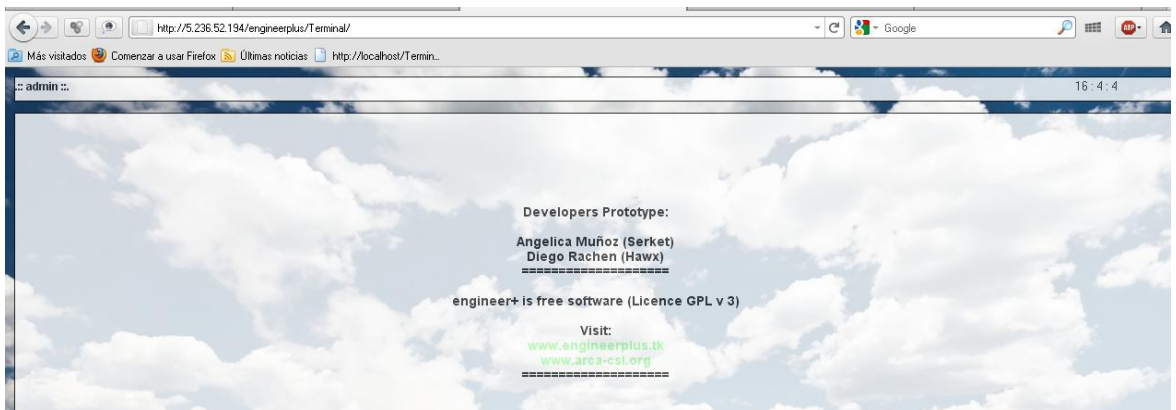
Una alerta que le informe sobre las anomalías encontradas.

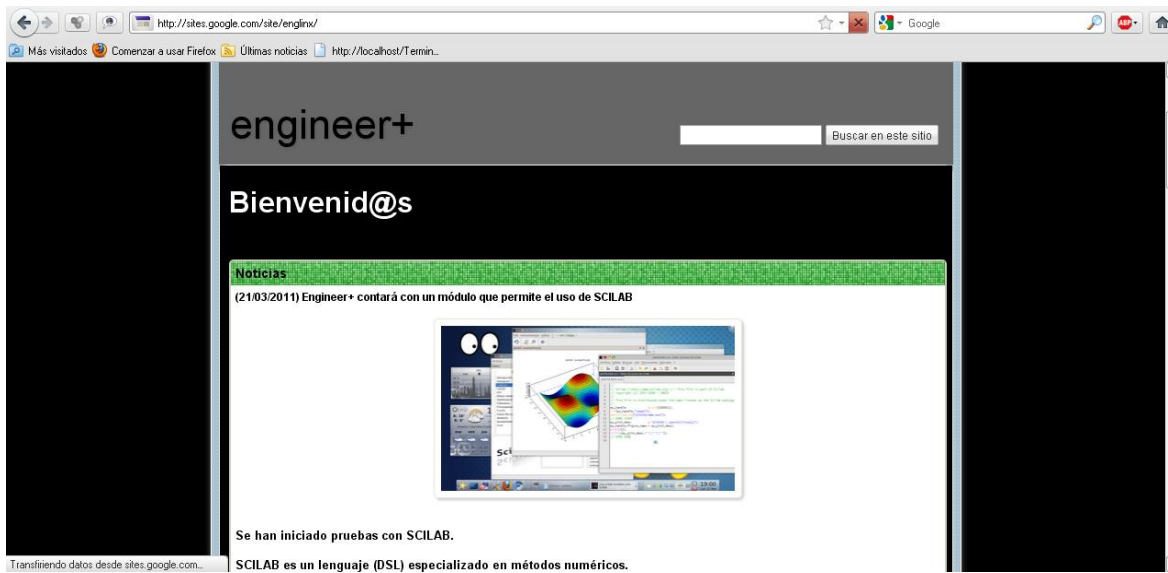
6. Presentación de Información para el usuario

1. Si desea conocer información sobre el proyecto: Ingrese el comando about () y presione la tecla Enter



Se observará la información general sobre el proyecto, sus desarrolladores y algunos otros datos como las páginas oficiales donde se encuentra la información sobre el proyecto y la comunidad de software libre a donde pertenece el grupo Engineer+, además podrá visitarlas haciendo click sobre la url del site.





Site oficial del grupo engineer+.



Página oficial del Arca-csl.

7. Tabla De Comandos

A continuación se mostraran los comandos en el uso de la Terminal

Comando	Función
about ()	Comando que devuelve la información sobre el proyecto
clear ()	Borra los datos contenidos para permitir la inserción de nueva información
div()	Divide los dos últimos datos ingresados
divAll()	Divide todos los datos ingresados
mult()	Multiplica los dos últimos datos ingresados
multiplyAll()	Multiplica todos los datos ingresados
sort()	Ordena los datos ingresados
sub()	Resta los dos últimos datos ingresados
subAll()	Resta todos los datos ingresados
sum()	Suma los dos últimos datos ingresados
sumAll()	Suma todos los datos ingresados
menu()	Muestra el menú de opciones avanzadas de la Terminal
matrixView()	Visualiza una matriz en sus dimensiones
refresh()	Actualiza el componente para visualizar los cambios