



Formato de planos

FP-02



Nombre del caso:	Necesidad	ID:	01
Autor	Carlos Hernandez	Fecha Creación:	01/04/17

Se necesita un programa intuitivo que le ayude a facilitar el análisis y la presentación de datos.





Nombre del caso:	Idea	ID:	02
Autor	Carlos Hernandez	Fecha Creación:	01/04/17

Por la necesidad de no ser tan tedioso el generar reportes a mano, y no efectuar graficas basándose en esos reportes, asumiendo que no haya errores humanos, además de ser demasiado tedioso es poco eficiente, y atrasa la producción en una empresa.

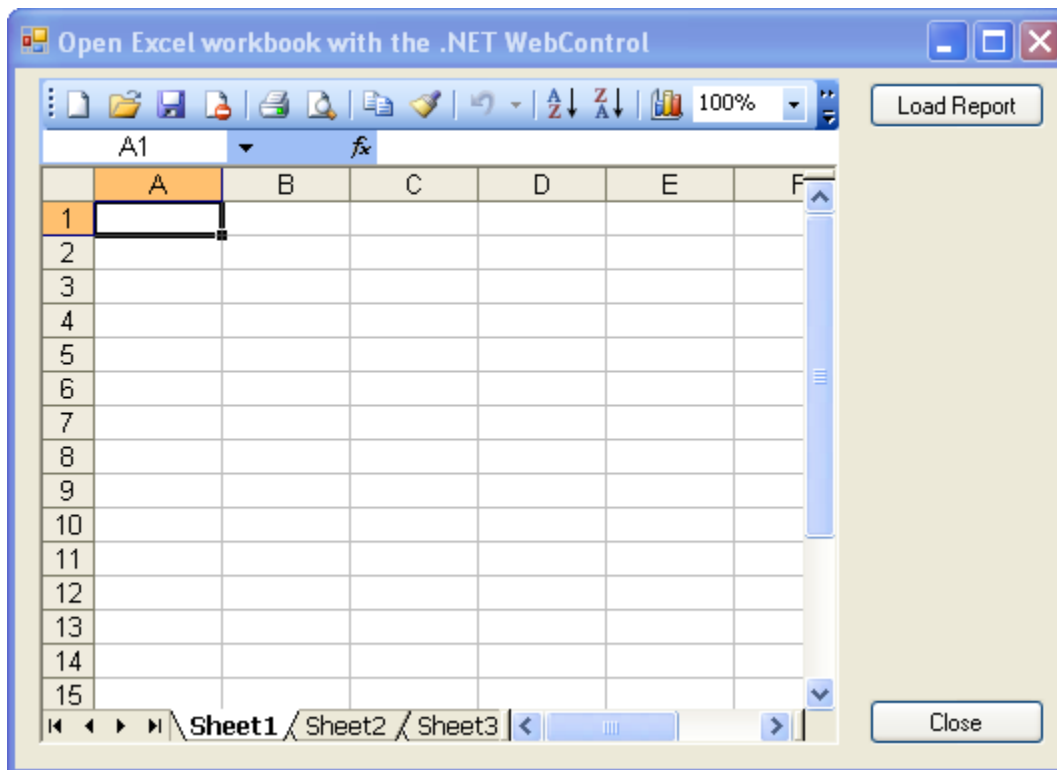
Con la experiencia de los trabadores, es recomendable tener una aplicación que les permita hacer reportes y graficas con mayor facilidad.





Nombre del caso:	Desarrollo de la Idea	ID:	03
Autor	Carlos Hernandez	Fecha Creación:	01/04/17

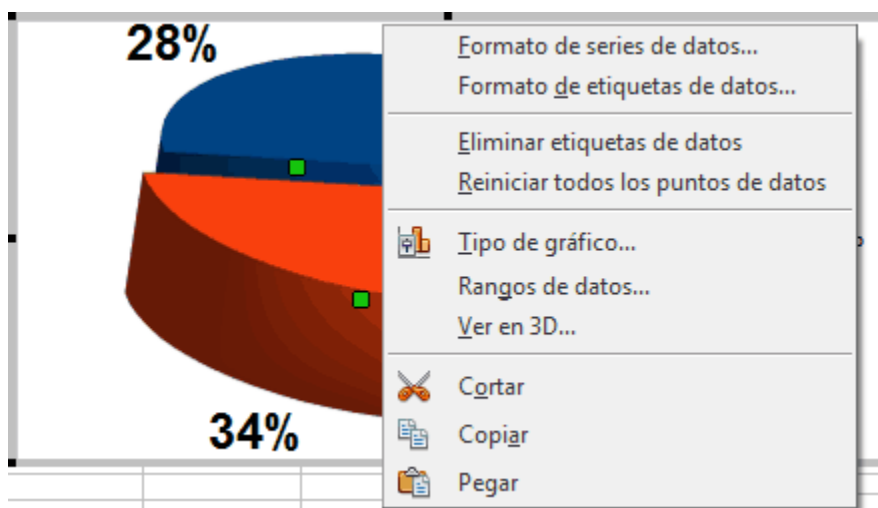
La aplicación OpenExcel permitirá el ingreso de datos para posteriormente poder analizarlos y hacer una grafica de ellos.





Nombre del caso:	Métricas(Core de la Aplicación)	ID:	04
Autor	Carlos Hernandez	Fecha Creación:	01/04/17

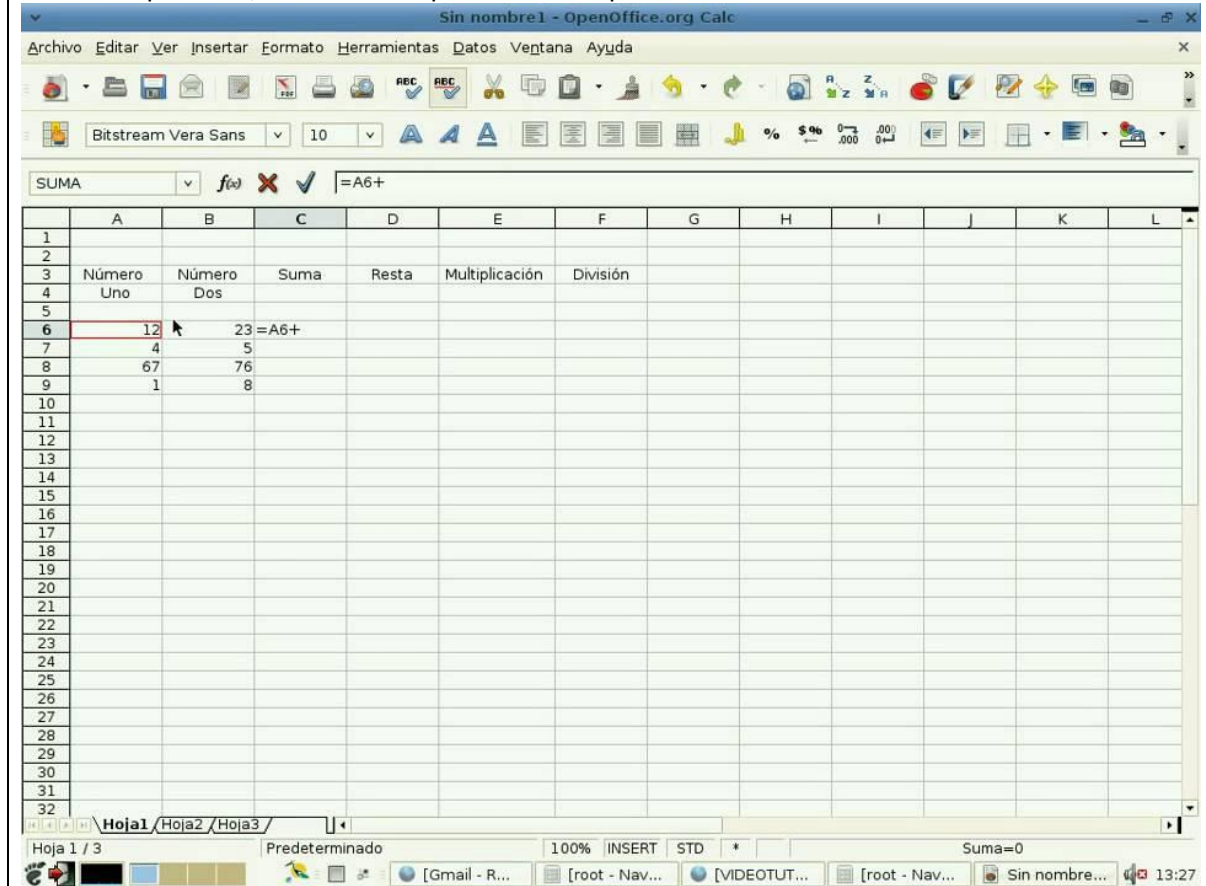
La interfaz siguiente, es del apartado de gráficas, donde posteriormente de haber ingresado los datos a analizar se generan diferentes tipos de graficas a gusto del operador.





Nombre del caso:	Métricas(Core de la Aplicación)	ID:	05
Autor	Carlos Hernandez	Fecha Creación:	01/04/17

La interfaz siguiente, es el apartado de operaciones matemáticas, donde se podrá escoger si es por celdas la operación, o solo serán operaciones simples aritméticas.





Nombre del caso:	Métricas(Core de la Aplicación)	ID:	06
Autor	Carlos Hernandez	Fecha Creación:	01/04/17
<ul style="list-style-type: none"> Las métricas a utilizar para obtener el log de valores: <ul style="list-style-type: none"> Característica 1: valor anterior Característica 2: valor actual Cada vez que se inserta o se borra un valor en una celda, se debe agregar un nodo con los valores que cambiaron. Estos cambios simularan el deshacer (Ctrl + z) y rehacer (Ctrl + y). Se recorrerá la lista desde el primer cambio, hasta el último para poder recuperar la información. Razón de existir: <ul style="list-style-type: none"> Esta aplicación al generar reportes, siempre se verá afectada por los errores humanos, y con el afán, de no perder los cambios realizados, o por el mal ingreso de un dato, se puede alterar un reporte, la lista doblemente enlazada permitirá el poder deshacer y rehacer los datos ingresados. 			



Nombre del caso:	Listado de Funcionalidades	ID:	06
Autor	Carlos Hernandez	Fecha Creación:	01/04/17
<ul style="list-style-type: none">• Módulos extras que permitirán una mejor experiencia con la aplicación:<ul style="list-style-type: none">○ La aplicación podrá editar el estilo del texto en cualquier celda.○ Apartado de operaciones matemáticas con un rango de celdas.○ Generar graficas sobre los reportes ya generados.			



ANEXOS

1. DESCRIPCION DEL PROBLEMA

Se necesita un programa intuitivo que le ayude a facilitar el análisis y la presentación de datos.

Los reportes generados a manos, para un posterior análisis de los datos ingresados y generar una gráfica, este proceso es demasiado tedioso y lento.

2. DEFINICION DE LOS REQUIRIMIENTOS DEL SISTEMA

- El sistema dejara ingresar datos como formulas en las celdas seleccionadas.
- El sistema generara reportes de los datos ingresados y generara un log de los cambios realizados.
- Generará graficas de los reportes.
- El sistema ejecutara operaciones matemáticas.
- El sistema será adaptable a cualquier necesidad del usuario.
- El sistema se desarrollará en un entorno Java EE.
- El sistema deberá pedir la autorización del usuario para poder ejecutar cambios.
- El sistema podrá almacenar archivos con extensión csv con datos previamente guardados.
- El sistema automáticamente realizara guardados automáticos.



3. PLANTEO INICIAL DE LA SOLUCION

Por la necesidad de no ser tan tedioso el poder hacer reportes con sus respectivas graficas a mano además de tardado es poco eficiente.

Con la experiencia de los trabadores, es recomendable tener una aplicación que les permita hacer reporte y que automáticamente pueda generar graficas del mismo

Las ventajas que poseerá este proyecto, es que con la tecnología que tenemos al alcance no podemos limitarnos a una sola computadora, el proyecto puede expandirse a cualquier dispositivo que el usuario desee.

4. CICLO DE VIDA DEL SOFTWARE

➤ Análisis:

Se necesitan cierta cantidad de datos para alimentar el sistema estos pueden ser 2.

- Datos de Entrada: números y formulas
- Datos de Salida: analizar y graficar datos

➤ Diseño:

En esta fase se procede a diseñar algoritmos que satisfagan las necesidades ya descritas, cabe destacar que un algoritmo es un conjunto finito de pasos para resolver un problema, este puede ser escrito en pseudocódigo y para su mejor comprensión graficado en un diagrama de flujo.

El sistema será trabajado en una forma modular, es decir que cada proceso será individual, es decir que, si algún proceso por algún motivo no logra terminarse de ejecutar o no se ejecuta correctamente, las demás partes del sistema seguirán funcionando.



➤ **Implementación:**

La fase donde el sistema se empieza a testear con productos “falsos” para verificar su correcto funcionamiento, esto lo hará el tester del software y aun no se procede a instalarlo en la empresa.

➤ **Prueba e Integración:**

En esta fase el software está listo para ser puesto a prueba por la empresa, con datos reales, ejecutando las operaciones en tiempo real, posteriormente con un visto bueno, se procede a instalar el software y ponerlo en marcha.

➤ **Depuración:**

En esta fase suelen salir a luz los pequeños bugs que no fueron detectados en las fases anteriores, o los cambios ya sean estéticos o backend que desee la empresa, estos errores son corregidos antes de proceder a la completa implementación del software.

➤ **Mantenimiento:**

Esta fase es una recursiva ya que no se ejecutará solo una vez, es recomendado al menos cada 6 meses realizarle un mantenimiento o actualización al sistema, ya que puede suceder que algunos algoritmos estén obsoletos.



DICCIONARIO DE CLASES

Clase: Principal

Esta es la clase principal, encargada de todo el funcionamiento del sistema.

Atributos:

Metodos:

Menu	Posee el menú de acciones
------	---------------------------

Clase: Operaciones

Esta es la clase de Operaciones, encargada de procesos matematicos

Atributos:

Metodos:

sumar	Suma el rango de celdas seleccionadas
restar	Resta el rango de celdas seleccionadas
Multiplicar	Multiplica el rango de celdas seleccionadas
Dividir	Divide el rango de celdas seleccionadas

Clase: Estilos

Esta es la clase de Estilos, encargada de editar los estilos de las celdas, asi como su texto

Atributos:

Metodos:

negrita	Negrita el texto
cursiva	Cursiva el texto
subrayado	subrayar el texto

Clase: GuardarDatos

Esta es la clase de GuardarDatos, encargada de guardar y recuperar los datos ingresados

Atributos:

Metodos:

Guardar	Guarda todos los datos en un archivo csv
Recuperar	Recupera todos los datos en un archivo csv



GLOSARIO DE TERMINOS

Términos:
Algoritmo
Método que describe cómo se resuelve un problema en término de las acciones que se ejecutan y especifica el orden en que se ejecutan estas acciones. Los algoritmos ayudan al programador a planificar un programa antes de su escritura en un lenguaje de programación.
Ámbito de clase:
Las variables privadas definidas fuera de los métodos internos a la clase tienen ámbito de clase. Son accesibles desde todos los métodos del interior de la clase, con independencia del orden en que están definidas. Los métodos privados también tiene ámbito de clase.
Análisis:
Proceso de identificación, modelado y descripción de lo que hace un sistema y de cómo trabaja
Aplicación:
Programa autónomo Java tal como cualquier programa escrito utilizando un lenguaje de alto nivel. Las aplicaciones se pueden ejecutar desde cualquier computadora con un intérprete Java.
Diccionario de Clase:
Colección organizada de clases que proporciona un conjunto de componentes y abstracciones reutilizables
Boolean:
Tipos primitivos de datos en Java. El tipo boolean puede tomar sólo dos valores: true (verdadero) y false (falso).
Clase:
Colección encapsulada de datos y operaciones que actúan sobre los datos. El concepto de clase es fundamental en programación orientada a objetos. Una clase consta de métodos y datos. Los métodos de una clase definen el conjunto de operaciones permitidas sobre los datos de una clase (sus atributos). Una clase puede tener muchas instancias de la clase u objetos.
Compilación:
Proceso de traducción de un lenguaje de programación. Normalmente este proceso implica la traducción de un lenguaje de programación de alto nivel a lenguaje de programación de bajo nivel, o el formato binario de un conjunto de instrucciones específicas. La traducción se realiza con un programa denominado compilador. Un compilador java traduce los programas en bytecodes.
Compilador:
Programa de software que realiza un proceso de compilación (traducción del lenguaje fuente a lenguaje máquina) de un programa escrito en un lenguaje de programación de alto nivel. En el caso de Java, es un programa que traduce el código fuente Java en bytecode. El compilador de J2SDK se denomina javac.
Constructor:
Método especial utilizado para inicializar el estado de un nuevo objeto. El constructor permite crear objetos utilizando el operador new. El constructor tiene exactamente el mismo nombre que la clase que lo contiene. Los constructores se pueden sobrecargar con



el objetivo de facilitar la construcción de objetos con diferentes tipos de valores iniciales.
Depuración:
Proceso de encontrar, fijar y eliminar errores en un programa. Para estas tareas se suele utilizar una herramienta de programación conocida como depurador.
Diseño:
Actividad de definir como se debe estructurar e implementar un programa.
Función:
Construcción matemática a la que se pueden aplicar valores y que devuelve un resultado.
Implementación:
La actividad de escribir, compilar, probar y depurar el código de un programa.
Interfaz:
Una interfaz se trata como una clase especial de Java. Cada interface se compila en un archivo independiente de bytecode, tal como una clase ordinaria. No se puede crear una instancia de la interfaz. La estructura de una interfaz Java es similar al de una clase abstracta en la que se puede tener datos y métodos. Los datos, sin embargo, deben ser constantes y los métodos pueden tener sólo declaraciones sin implementación. En Java existe sólo herencia simple y una clase puede heredar de una superclase. Esta restricción se puede superar por el uso de una interfaz.
Método:
Una colección de sentencias que se agrupan juntos para ejecutar una operación.
Objeto:
Una instancia de una clase específica. En general, se puede construir cualquier número de objetos a partir de una clase.
Programa:
Un conjunto de instrucciones (o sentencias) que describen alguna aplicación o actividad ejecutada en una computadora.
UML:
Lenguaje unificado de modelado que proporciona notación estándar visual para documentar el análisis y diseño de sistemas orientados a objetos.