

# **Sistema de interpretación de ensayos de bombeo en acuíferos**

## **Glosario, apéndice y bibliografía**

### **Cliente**

Grupo de Hidrología Subterránea de Regional Norte

### **Tutores**

Carla Forni  
Pablo Gamazzo

### **Estudiantes**

Sebastián Daloia  
Mathias Chubrega  
Andrés Pías  
Álvaro Correa  
Jesús Guibert



## Índice:

<b>1</b>	<b>REFERENCIAS.....</b>	<b>5</b>
<b>2</b>	<b>APÉNDICE.....</b>	<b>9</b>
2.1	CASOS DE USO.....	9
2.1.1	Casos de Uso Críticos.....	9
2.1.1.1	Crear Proyecto.....	9
2.1.1.2	Definir Dominio.....	11
2.1.1.3	Ingresar caudal bombeado.....	14
2.1.1.4	Ingresar observaciones del ensayo.....	15
2.1.1.5	Crear visualización.....	16
2.1.1.6	Asociar Optimización.....	18
2.1.2	Casos de uso no críticos.....	19
2.1.2.1	Modificar Dominio.....	19
2.1.2.2	Importar caudal bombeado.....	19
2.1.2.3	Importar observaciones del ensayo.....	21
2.1.2.4	Exportar video.....	22
2.2	DIAGRAMAS DE SECUENCIA DEL SISTEMA.....	23
2.2.1	Diagramas no críticos.....	23
2.2.1.1	Modificar Dominio.....	23
2.2.1.2	Importar caudal bombeado.....	24
2.2.1.3	Importar Observaciones del ensayo.....	25
2.2.1.4	Exportar Video.....	26
2.3	EVALUACION DE LENGUAJES.....	27
2.3.1	Python.....	27
2.3.1.1	Sintaxis.....	27
2.3.1.2	Requerimientos.....	27
2.3.1.3	Soporte y Documentación.....	27
2.3.1.4	Características.....	28
2.3.1.5	Matplotlib.....	31
2.3.1.5.1	Componente Matplotlib-matlab.....	31
2.3.1.5.2	Características.....	31
2.3.1.5.3	Estructura Conceptual.....	32
2.3.1.5.4	Clases principales en Matplotlib API.....	33
2.3.1.5.5	Requerimientos.....	33
2.3.1.5.6	Soporte y Documentación.....	34
2.3.1.5.7	Ejemplo simple con Matplotlib.....	35
2.3.1.5.8	Ejemplo simple en 3D.....	36
2.3.1.5.9	Animación en matplotlib.....	36
2.3.1.5.10	PyQt4.....	37
2.3.1.5.11	Ventajas.....	37
2.3.1.5.12	Referencias.....	38
2.3.2	Java.....	39
2.3.2.1	Descripción general.....	39
2.3.2.2	Características.....	40
2.3.2.3	Librería Jzy3D.....	41
2.3.2.3.1	Funcionalidades disponibles.....	41
2.3.2.3.2	Características destacables.....	41
2.3.2.3.3	Funcionamiento general.....	42
2.3.2.3.4	Soporte y documentación.....	42
2.3.2.4	Librería Processing.....	43
2.3.2.4.1	Funcionamiento general.....	43
2.3.2.4.2	Documentación y soporte.....	43
2.3.2.5	Soporte y Documentación.....	44
2.3.2.6	Ejemplos de implementación, casos vistos.....	45
2.3.2.6.1	Jzy3D: Ejemplo 1.....	45
2.3.2.6.2	Jzy3D: Ejemplo 2.....	46
2.3.2.6.3	Jzy3D: Ejemplo 3.....	47
2.3.3	C/C++.....	48
2.3.3.1	Características.....	48
2.3.3.2	Librerías c/c++.....	49
2.3.3.2.1	iostream.h.....	49

2.3.3.2String.h.....	49
2.3.3.2stdio.h.....	50
2.3.3.2stdlib.h.....	51
2.3.3.3Ejemplo de procesamiento de archivos:.....	54
2.4VISIÓN INICIAL.....	55
2.5PROGRAMACIÓN ORIENTADA A OBJETOS (OO).....	58
<b>3GLOSARIO.....</b>	<b>60</b>

# 1 Referencias

---

- [1] Agile Models Distilled: Potential Artifacts for Agile Modeling  
<http://www.agilemodeling.com/artifacts/>
- [2] Agile Modeling. UML 2 Use Case Diagrams  
<http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>
- [3] Agile Modeling. UML 2 Sequence Diagrams  
<http://www.agilemodeling.com/artifacts/sequenceDiagram.htm>
- [4] Agile Modeling. UML 2 Component Diagrams  
<http://www.agilemodeling.com/artifacts/componentDiagram.htm>
- [5] Agile Modeling. UML 2 Deployment Diagrams  
<http://www.agilemodeling.com/artifacts/deploymentDiagram.htm>
- [6] Deployed Software: Artifacts – Ver página 18 de la presentación  
<http://es.scribd.com/doc/75364459/2/Deployed-Software-Artifacts>
- [7] Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives. N. Rozanski, E. Woods. Addison-Wesley, 2005
- [8] Software Architecture in practice, Second Edition. L. Bass, P. Clemens, R. Kazman. Addison-Wesley, 2003
- [9] An Introduction to Software Architecture. D. Garlan, M. Shaw. 1994.  
[http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro\\_softarch/intro\\_softarch.pdf](http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf)
- [10] Architectural Blueprints — The “4+1” View Model of Software Architecture. Kruchten, Philippe. 1995.  
<http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>
- [11] Software Architecture Links. Bredemeyer Consulting.  
<http://www.bredemeyer.com/links.htm>
- [12] Explicación del lenguaje B extraído de: [http://www.diclib.com/Lenguaje%20de%20programación%20B/show/es/es\\_wiki\\_10/24420](http://www.diclib.com/Lenguaje%20de%20programación%20B/show/es/es_wiki_10/24420)
- [13] Ejemplos de hola mundo para B, BCPL extraídos de:  
<http://www.helloworldexample.net>
- [14] Sistemas Operativos – Silberschatz Galvin
- [15] Introducción a la versión 2.7.2 de Python  
<http://docs.python.org/reference/introduction.html>

- [16] Cómo instalar Python en Windows XP  
<http://www.richarddooling.com/index.php/2006/03/14/python-on-xp-7-minutes-to-hello-world/>
- [17] Página oficial de Active State, en particular enlace de descarga del framework para Python <http://www.activestate.com/activepython/downloads>
- [18] Página de GNU PLOT herramienta gráfica 2D. 3D: <http://www.gnuplot.info/>
- [19] Página de matplotlib: <http://matplotlib.sourceforge.net/>
- [20] Página de Gnuplot: <http://www.gnuplot.info/>
- [21] GNU PLOT An interactive plotting program. Thomas Williams & Collin Kelley.
- [22] Página principal de Matplotlib: <http://matplotlib.sourceforge.net/index.html>
- [23] Guía de usuario de Matplotlib: <http://matplotlib.sourceforge.net/contents.html>
- [24] Enlace de descarga de Matplotlib:  
<http://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.0.1/>
- [25] Página principal de Qt: <http://qt.nokia.com/>
- [26] Página principal de PyQt:  
<http://www.riverbankcomputing.co.uk/software/pyqt/intro>
- [27] Página principal de NumPy: <http://numpy.scipy.org/>
- [28] Introducción a HTML 5  
<http://www.anexom.es/tecnologia/disenio-web/las-novedades-de-html5-i/>
- [29] Canvas Draw: <https://sites.google.com/site/canvasdraw/>
- [30] Ejemplo de un capítulo de un curso online del framework Django de Python donde se sincroniza el sonido con capturas de códigos, opciones o ventanas de Django: <http://django101.com/trial/>
- [31] Sincronización de contenidos del sistema solar en un video  
<http://webmademovies.org/popcorn-in-the-planetarium/>
- [32] El siguiente link es una herramienta que permite sincronizar contenido multimedia utilizando este framework: <http://butterapp.org/butter/>
- [33] Manejar formato avi: <http://msdn.microsoft.com/en-us/library/ms779636.aspx>
- [34] Api para el manejo de audio y video con HTML 5  
<http://www.storiesinflight.com/jsfft/visualizer/index.html>

- [35] Ciclo de vida. Concepto empleado en Ingeniería de Software, refiere a la construcción de un proyecto desde sus bases.  
[http://www.fing.edu.uy/tecnoinf/cursos/ingsoft/material/teorico/IS-03a-ModelosProceso\\_tecnologo.pdf](http://www.fing.edu.uy/tecnoinf/cursos/ingsoft/material/teorico/IS-03a-ModelosProceso_tecnologo.pdf)
- [36] Arquitectura del sistema (pág. 4 a 9) y Arquitectura en capas (pág 27 a 29)  
[http://www.fing.edu.uy/tecnoinf/cursos/ingsoft/material/teorico/IS-06a-ArquitecturaDeSoftware\\_tecnologo.pdf](http://www.fing.edu.uy/tecnoinf/cursos/ingsoft/material/teorico/IS-06a-ArquitecturaDeSoftware_tecnologo.pdf)
- [37] Página oficial de NotePad++:  
<http://notepad-plus-plus.org/download/v5.9.6.2.html>
- [38] Página oficial de gEdit:  
<http://projects.gnome.org/gedit/>
- [39] Descripción de los módulos de PyQt  
<http://www.riverbankcomputing.co.uk/static/Docs/PyQt4/html/modules.html>
- [40] Qt Modelo de Programación Modelo / Vista  
<http://developer.qt.nokia.com/doc/qt-4.8/model-view-programming.html>
- [41] Modelo de proceso  
[www.fing.edu.uy/tecnoinf/cursos/ingsoft/material/teorico/IS-03a-ModelosProceso\\_tecnologo.pdf](http://www.fing.edu.uy/tecnoinf/cursos/ingsoft/material/teorico/IS-03a-ModelosProceso_tecnologo.pdf)
- [42] Extract and Parse ODF Files with Python  
<http://www.linuxjournal.com/article/9347?page=0,0>
- [43] OASIS Open Document Format for Office Applications (OpenDocument) TC  
[http://www.oasis-open.org/committees/documents.php?wg\\_abbrev=office](http://www.oasis-open.org/committees/documents.php?wg_abbrev=office)
- [44] OpenOffice Página Oficial:  
<http://www.openoffice.org/>
- [45] Java Bytecodes  
[http://www.ibm.com/developerworks/ibm/library/it-haggar\\_bytecode/](http://www.ibm.com/developerworks/ibm/library/it-haggar_bytecode/)
- [46] Código Máquina:  
<http://www.wisegeek.com/what-is-machine-code.htm>
- [47] Java Virtual Machine, especificaciones:  
<http://java.sun.com/docs/books/jvms/>
- [48] Matlab – Introducción al análisis de datos en la biología computacional  
<http://www.mathworks.com/computational-biology/data-analysis.html>

- [49] Concepto Hidrológico: Optimización – Calibración – Documentos PDF  
[http://www.google.com.uy/url?sa=t&rct=j&q=calibraci%C3%B3n%20modelo%20hidrol%C3%B3gico&source=web&cd=12&ved=0CCsQFjABOAo&url=http%3A%2F%2Fwww.weap21.org%2Fdownloads%2FGuia\\_modelacion\\_WEAP\\_Espanol.pdf&ei=1wNNT5WoN4OCgAelz8m2Ag&usg=AFQjCNENV3EYNTeNAZGuRW0KpK-b6-SWww](http://www.google.com.uy/url?sa=t&rct=j&q=calibraci%C3%B3n%20modelo%20hidrol%C3%B3gico&source=web&cd=12&ved=0CCsQFjABOAo&url=http%3A%2F%2Fwww.weap21.org%2Fdownloads%2FGuia_modelacion_WEAP_Espanol.pdf&ei=1wNNT5WoN4OCgAelz8m2Ag&usg=AFQjCNENV3EYNTeNAZGuRW0KpK-b6-SWww)  
<http://www.mssanz.org.au/modsim05/papers/abbaspour.pdf>
- [50] Conceptos sobre Hidrología  
<http://www.h2ogeo.upc.es/Docencia/Paisajismo/AGUAS%20SUBTERR%C3%81NEAS.pdf>
- [51] Conceptos sobre Hidrología subterránea  
<http://pdf.rincondelvago.com/hidrologia-subterranea.html>
- [52] Sintaxis equivalentes entre Matlab y Numpy  
<http://mathesaurus.sourceforge.net/matlab-numpy.html>
- [53] Función gradient de Matlab  
<http://www.mathworks.com/help/techdoc/ref/gradient.html>
- [54] Página de github, enlace hacia sus características.  
<https://github.com/features/projects>
- [55] Integral exponencial, introducción conceptual.  
<http://astro.if.ufrgs.br/levato/ecutransporte4/exponenciales.htm>
- [56] Funciones de Bessel - Material educativo de comunicaciones 1, U.N.E.F.A.  
<http://www.slideshare.net/ejcomunicaciones/funcion-de-bessel-regla-de-carson-y-modulacion-exponencial-3073230>
- [57] Introducción al framework Graphics Vlew.  
<http://developer.qt.nokia.com/doc/qt-4.8/graphicsview.html>
- [58] Árbol Binario BSP – Explicación teórica y práctica.  
<http://www.faqs.org/faqs/graphics/bsptree-faq/>



## 2 Apéndice

### 2.1 Casos de Uso

#### 2.1.1 Casos de Uso Críticos

##### 2.1.1.1 Crear Proyecto

Caso de uso: Crear proyecto

Actores: Usuario

Descripción:

Se iniciará una nueva actividad. Se crearán las mínimas bases requeridas para trabajar.

Pre-condiciones:

No se ha iniciado ninguna actividad.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario dispara la acción de crear un nuevo proyecto.	
	2- Se despliega en pantalla una interfaz que pone a disposición ingresar las dimensiones del terreno a trabajar, seleccionar el método de solución a utilizar.
3- El usuario elige las dimensiones del dominio a representar.	
	4- El sistema recuerda estos datos y espera a que prosigan las acciones.
5- El usuario selecciona el método de solución a utilizar.	
	6- El sistema pone a disposición dinámicamente los parámetros correspondientes a ingresar.
6- El usuario ingresa los parámetros del método de solución seleccionado (por ejemplo S y T).	

	7- El sistema recuerda estos datos.
10- El usuario puede optar por agregar alguna condición externa al dominio. Selecciona condición y valor de la misma.	<i>Sólo si el método seleccionado es del tipo numérico.</i>
	11- El sistema recuerda estos datos.
12-El usuario acepta la creación de un nuevo proyecto.	
	13- El sistema verifica los datos; en caso de encontrar algún error no se crea el proyecto, de lo contrario el sistema carga una nueva interfaz y espera que prosigan las acciones.

#### Post-condiciones:

Se iniciará una nueva actividad. Estarán sentadas las bases para empezar a trabajar. Estarán establecidas las dimensiones del dominio y seleccionado el método de solución con sus respectivos parámetros. El sistema recuerda y sigue trabajando en base a ello.

### 2.1.1.2 Definir Dominio

Actores: Usuario

Descripción: El usuario crea un dominio identificando los elementos, en cuanto a posición y cantidad; su ubicación podrá hacerse mediante el ingreso de datos u operando gráficamente con el sistema. Los elementos son representaciones de pozos de bombeos u observación. Una vez definidos éstos, se les podrá asociar datos.

También se podrán definir barreras en el dominio, ingresando datos u operando gráficamente.

Pre-condiciones: Deberá haberse creado un proyecto, con la respectiva dimensión del plano. Los datos de observación o bombeo que se requieran deberán haberse cargado previamente.

Si el usuario eligió algún método solución que no incluya barreras en el dominio entonces aparecerá deshabilitado el ícono de la recta, junto a los eventos que este desencadena.

Post Condiciones: Se contará con un dominio bien definido, con los pozos y barreras necesarios y los datos de los mismos cargados.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario selecciona un ícono de la herramienta gráfica y lo arrastra a alguna parte del dominio; si el ícono es llevado fuera del dominio no se ejecuta ninguna operación.	
<i>Luego de esta respuesta el usuario puede continuar con la secuencia, o volver a la línea 1</i>	2- El sistema devuelve una representación del ícono seleccionado que se adherirá al mouse.
3-Soltando el botón izquierdo se posiciona el ícono.	<i>Si el ícono es un pozo pasar a línea 4</i> <i>Si el ícono es una recta pasar a línea 5</i>
	4- Se desadhiere el ícono del mouse, y se posiciona otro en el punto del dominio que el usuario seleccionó. Se guardan los datos cartesianos y se visualizan en el formulario de coordenadas.
<i>Una vez conocida la respuesta del sistema el usuario puede pasar a la línea 16, 6 o a la 10.</i>	5-Se desadhiere el ícono del mouse y se posiciona otro en el punto del dominio que el usuario seleccionó. En este caso una recta paralela al eje x. Se guardan los datos cartesianos. El sistema habilita el panel de coordenadas para que el usuario indique si la barrera es negativa o positiva.
6- Si el ícono es una recta entonces el	

usuario indica en el panel de datos de la recta si esta es positiva o negativa.	
	7 El sistema guarda este dato en el dominio,y determina orientacion de la recta.
8- Se posiciona el mouse en una de las semirrectas determinadas por el punto medio..	
	9- Se cambia el ícono del mouse, comúnmente una flecha, por un nuevo ícono que nemotécnicamente indique la posibilidad de mover la recta hacia uno de sus lados.
10- El usuario mantiene apretado el botón izquierdo del mouse, y lo mueve hacia o fuera del eje.	
	11- El sistema va cambiando la posición de la recta de acuerdo al nuevo par de puntos intersectados.
12- Se suelta el botón izquierdo	
<i>Se puede continuar con la secuencia, o volver a la línea 1, 6 o 7.</i>	13- Se guarda la posición actual de la recta
14- Se hace click derecho en alguno de los pozos, eligiéndose la opción asociar.	
	15- Se devuelve una ventana informativa que indique los datos cargados que actualmente tiene el pozo, junto con una caja de texto con los nombres de los datos que han sido previamente cargados.
16- El usuario opta por ingresar una observación aun no asociada.	
	17- El sistema devuelve una lista de datos aún no asociados a ningún pozo..
18.a- El usuario elije un conjunto de observaciones y acepta la asociación.	
	18.b- El sistema asocia dichas observaciones en el pozo, y las desasocia de la lista de observaciones aun sin asociar.
18.c- El usuario opta por ingresar un bombeo aun no asociado.	
	18.d- El sistema devuelve una lista de datos aun no asociados.
18.e- El usuario elije un conjunto de ensayos y acepta la asociación.	
	18.f- El sistema asocia dichos ensayos en el pozo y desasocia de la lista de bombeo aun sin asociar.
19- El usuario cierra la ventana emergente.	<i>Se puede volver a la línea 1, 6, 7 o 14</i>

20- Reposicionar ícono del dominio. El usuario mantiene apretado el botón izquierdo del mouse sobre el ícono y comienza a arrastrarlo.	
	21- El sistema anota las coordenadas actuales y va posicionando el elemento de acuerdo al movimiento del mouse.
22- El usuario suelta el ícono.	
	23- El sistema guarda la coordenada actual del elemento representado por el ícono. Se actualizan las coordenadas en el formulario correspondiente.

**Curso alterno:**

**Línea 1:** El usuario da coordenadas, x,y, para posicionar un punto.

**Línea 2:** El sistema posicionará en el mapa un punto de acuerdo a las coordenadas dadas.

**Línea 1:** El usuario da coordenadas, [x,y; x,y] para posicionar una recta.

**Línea 2:** El sistema posicionará en el mapa la recta de acuerdo a las coordenadas dadas.

Mientras el usuario no pase a la línea 3 o 5, los datos ingresados seguirán afectando al nuevo elemento.

**Línea 3:** Se acepta la ubicación y se continúa con la secuencia.

**Línea 4:** El sistema guarda los datos.

**Línea 5:** Se cancela la ubicación.

**Línea 6:** Se borran los datos de las coordenadas ingresadas y se elimina el ícono que se estaba representando.

**Línea 11:** Los puntos X, Y que conforman la recta no pueden ser iguales; por lo tanto si el usuario continúa moviendo el mouse hacia el eje, cuando la distancia entre éstos sea la mitad de la distancia que por defecto tiene la recta cuando se agrega gráficamente, entonces el sistema deja de modificar su tamaño, hasta que el usuario comience a mover el mouse de manera opuesta al eje.

**Línea 18.a:** El usuario decide cancelar la asociación.

**Línea 18.b:** El sistema hace un borrado lógico de las asociaciones generadas.

**Línea 18.e:** El usuario decide cancelar la asociación.

**Línea 18.f:** El sistema hace un borrado lógico de las asociaciones generadas.

### 2.1.1.3 Ingresar caudal bombeado

Actores: Usuario

Descripción: Este caso de uso permite ingresar los datos de los caudales bombeados por los pozos de bombeo a medida que transcurre el tiempo.

Pre-condiciones: Un proyecto fue creado.

Post-condiciones: Se creará un nuevo ensayo de bombeo en el sistema, almacenados los caudales bombeados.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario ingresa un valor en segundos del tiempo t en el cual se produce el bombeo	
2- El usuario ingresa el caudal bombeado para ese tiempo	
3- Presiona el botón Agregar para almacenar los datos	<i>El sistema verifica que los datos ingresados del bombeo son correctos y los almacena</i>
4- Se repiten los pasos del 2 al 3 para ingresar los datos de bombeo para otros tiempos	
5- El usuario presiona Aceptar para finalizar el registro de la información	<i>El sistema muestra un cuadro de diálogo</i>
6- El usuario ingresa un nombre para el ensayo	
7- Presiona Aceptar para confirmar el ingreso.	<i>El sistema retorna un id asociado al ensayo almacenado</i>

Curso alternativo de eventos:

Acción de actor	Respuesta del sistema
3-b-	<i>El sistema detecta que el tiempo ingresado es menor que el del bombeo anterior.</i>
3-c- Se vuelve al paso 1	
7-b- Presiona el botón Cancelar para desechar los datos	<i>El sistema cancela las acciones</i>

### 2.1.1.4 Ingresar observaciones del ensayo

Actores: Usuario

Descripción: Este caso de uso permite ingresar los niveles observados por los pozos de observación a medida que transcurre el tiempo.

Pre-condiciones: Un proyecto fue creado.

Post-condiciones: Se registrará un nuevo conjunto de observaciones en el sistema, almacenando las observaciones ingresadas.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario ingresa un valor en segundos del tiempo en el cual se produce la observación	
2- El usuario ingresa el caudal observado para ese tiempo	
3- Presiona el botón Guardar para almacenar los datos	<i>El sistema verifica que los datos ingresados de la observación son correctos y los almacena</i>
4- Se repiten los pasos del 2 al 3 para ingresar los datos de observaciones para otros tiempos	
5- El usuario presiona Aceptar para finalizar el registro de la información	<i>El sistema muestra un cuadro de diálogo</i>
6- El usuario ingresa un nombre para el conjunto de observaciones	
7- Presiona Aceptar para confirmar el ingreso	<i>El sistema retorna un id asociado al conjunto de observaciones almacenado</i>

Curso alternativo de eventos:

Acción de actor	Respuesta del sistema
3-b-	<i>El sistema detecta que el tiempo ingresado es menor que el del bombeo anterior.</i>
3-c- Se vuelve al paso 1	
7-b- Presiona el botón Cancelar para desechar los datos	<i>El sistema cancela las acciones</i>

### 2.1.1.5 Crear visualización

Actores: Usuario

Descripción: Este caso de uso provee al usuario una representación de los datos ingresados y procesados.

Pre-condiciones:

1. Es necesario que se haya creado un proyecto.
2. El dominio debe estar definido.
3. Debe haber por lo menos un pozo de bombeo con ensayos cargados.
4. Debe haber por lo menos un pozo de observación con observaciones cargadas.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario desea ver representaciones de los datos ingresados. Para ello, selecciona de un menú la opción para generar visualización.	
	2- El sistema le presenta al usuario un conjunto de gráficas y herramientas para interactuar con las mismas.
3- Si el usuario así lo desea, puede utilizar en cualquier momento los controles dispuestos para reproducir, pausar o exportar las gráficas generadas en formato de video.	
3.a- El usuario desea reproducir las representaciones animadas; para ello utiliza la funcionalidad reproducir presentada junto con otras herramientas.	
	3.b- El sistema entonces comienza a reproducir las gráficas que se pueden animar. La opción de reproducción es deshabilitada y presenta además la funcionalidad de pausar la animación.
3.b.i- El usuario puede pausar en cualquier momento la reproducción, pudiendo reanudar la misma en cualquier otro momento.	
	3.b.ii- El sistema detiene la reproducción de las gráficas y habilita nuevamente la opción de reproducción.
3.c- El usuario desea entonces guardar	



las gráficas y sus respectivas animaciones en un video; entonces utiliza la funcionalidad de exportar a video.	
	<i>3.d- El sistema entonces deriva al usuario a la funcionalidad de exportar video.</i>

Post-condiciones:

Se creará una ventana dividida en cuatro, donde cada una de las divisiones corresponderá a cada una de las representaciones. Además, se mostrarán junto con las vistas un conjunto de herramientas para interactuar con las mismas.

### 2.1.1.6 Asociar Optimización

Actores: Usuario

Descripción: Este caso de uso permite asociar a un pozo de observación un método de optimización.

Pre-condiciones: Al menos un pozo de observación creado en el dominio, junto con sus observaciones.

Post-condiciones: Se generan los datos de optimización dependiendo del método de seleccionado.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario selecciona de un menú la opción optimización.	<i>Inmediatamente se muestra una ventana donde se listarán los métodos de optimización disponibles para la solución del dominio.</i>
2- El usuario selecciona un algoritmo de optimización	El sistema asocia el pozo seleccionado con el método seleccionado; mostrará una pestaña con todas las asociaciones realizadas anteriormente.
3- El usuario ingresará los parámetros necesarios por los métodos de optimización.	
4- El usuario indicará que desea procesar las optimizaciones disponibles.	<i>El sistema realizará los cálculos correspondientes en todas las asociaciones disponibles. Y retornará las gráficas correspondientes.</i>

Curso alternativo de eventos:

Acción de actor	Respuesta del sistema
4-b- El usuario puede elegir qué asociaciones no procesar.	<i>El sistema realizará los cálculos correspondientes en aquellas asociaciones que no fueron quitadas.</i>

## 2.1.2 Casos de uso no críticos

### 2.1.2.1 Modificar Dominio

Actores: Usuario

Descripción: Trata sobre la modificación o eliminación de los datos definidos en el dominio.

Pre-condiciones: Ya se debieron haber ingresado datos al dominio.

Pos-Condicion: Se devuelve un Dominio más ajustado a la realidad.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1.a- Modificar. El usuario se posiciona sobre el elemento a modificar apretando una vez el botón izquierdo del mouse.	
<i>Si se trata de un pozo ir al paso 1.a.1 para modificar las observaciones, al paso 1.a.2 para modificar las coordenadas del pozo.</i> <i>Si se trata de una recta ir al paso 1.a.3 para modificar su posición en el dominio.</i>	El sistema devuelve las coordenadas .
1.a.1- Se procede a ejecutar la rutina descrita en las líneas desde 14 hasta 19 del caso de uso <b>Definir dominio</b> .	
1.a.2- Se procede a ejecutar la rutina descrita en las líneas desde 21 hasta 23 del caso de uso <b>Definir dominio</b> , o el curso alternativo para el punto de la línea número 1.	
<b>1.a.3-</b> Para modificar la recta se procede a ejecutar la rutina descrita en las líneas desde 6 hasta 12 del caso de uso <b>Definir dominio</b> o el curso alternativo para rectas de la línea número 1.	
2.- Eliminar símbolo del dominio. El usuario se posiciona sobre cualquier ícono y selecciona la opción de eliminar.	
	2.3- Se elimina el ícono del dominio; previamente se desasocian los datos y se eliminan las coordenadas que lo representaban.

### 2.1.2.2 Importar caudal bombeado

Actores: Usuario

Descripción: Esta funcionalidad permite que el usuario importe en un archivo los datos de los caudales bombeados en diferentes tiempos del ensayo. Esto permite que el sistema registre el ensayo, para posteriormente asociarlo con los pozos de bombeo.

Pre-condiciones: Un proyecto fue creado. El archivo suministrado por el usuario debe ser del tipo txt u ods (de Open Office Calc) y debe cumplir un formato especial.

Post-condiciones: Los caudales bombeados por los pozos de bombeo han sido ingresados.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario ingresa un nombre para el ensayo de bombeo	
2- El usuario selecciona un fichero para importar los datos	
3- Presiona el botón Importar.	<i>El sistema procesa el archivo chequeando primero que sea del tipo y formato adecuados. Determina que es correcto y muestra un cuadro de dialogo.</i>
4- El usuario ingresa un nombre para el ensayo	
5- El usuario confirma la operación presionando el botón Aceptar	<i>El sistema almacena los datos de bombeo y retorna un id identificadorio del ensayo de bombeo</i>

Curso alternativo de eventos:

Acción de actor	Respuesta del sistema
	<i>2-b- El sistema detecta un error en el tipo de archivo ingresado o en la sintaxis de su contenido (los tiempos están desordenados).</i>
2-c- Se vuelve al paso 1	
5-b- Presiona el botón Cancelar para desechar los datos	<i>El sistema cancela las acciones</i>

### 2.1.2.3 Importar observaciones del ensayo

Actores: Usuario

Descripción: Esta funcionalidad permite que el usuario importe en un archivo los datos de las observaciones para todos los tiempos del ensayo. Esto permite que el sistema registre el ensayo, para posteriormente asociarlo con los pozos de observaciones correspondientes.

Pre-condiciones: Un proyecto fue creado. El archivo suministrado por el usuario debe ser del tipo txt u ods (de Open Office Calc) y debe cumplir un formato especial.

Post-condiciones: Las observaciones del ensayo han sido ingresadas.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario ingresa un nombre para el conjunto de observaciones	
2- El usuario selecciona un fichero para importar los datos	
3- Presiona el botón Importar.	<i>El sistema procesa el archivo chequeando primero que sea del tipo y formato adecuados. Determina que es correcto y muestra un cuadro de diálogo.</i>
4- El usuario ingresa un nombre para el ensayo	
5- El usuario confirma la operación presionando el botón Aceptar	<i>El sistema almacena los datos de las observaciones y retorna un id identificatorio del conjunto de observaciones.</i>

Curso alternativo de eventos:

Acción de actor	Respuesta del sistema
	<i>2-b- El sistema detecta un error en el tipo de archivo ingresado o en la sintaxis de su contenido (los tiempos están desordenados).</i>
2-c- Se vuelve al paso 1	
5-b- Presiona el botón Cancelar para desechar los datos	<i>El sistema cancela las acciones</i>

### 2.1.2.4 Exportar video

Actores: usuario

Descripción: este caso de uso permitirá al usuario generar un video con las gráficas presentadas en la creación de visualizaciones.

Pre-condiciones:

1. Es necesario que se haya creado un proyecto.
2. El dominio debe estar definido.
3. Debe haber por lo menos un pozo de bombeo con ensayos cargados.
4. Debe haber por lo menos un pozo de observación con observaciones cargadas.

Flujo de eventos:

Acción de actor	Respuesta del sistema
1- El usuario desea generar un video con las gráficas generadas a partir de los datos ingresados. Para ello accede a la opción de generar video.	
	2- <i>El sistema entonces pide algunos datos al usuario como el lugar de destino del video.</i>
3- El usuario ingresa los datos correspondientes y confirma la creación del video.	
	4- <i>El sistema genera el video en el lugar de elección del usuario.</i>

Curso alternativo de eventos:

Acción de actor	Respuesta del sistema
5- El usuario desea cancelar la creación del video. Para ello utiliza la opción de cancelar que se le habilita al comenzar la generación del video.	
	6- <i>El sistema desecha todo el proceso llevado a cabo hasta el momento.</i>

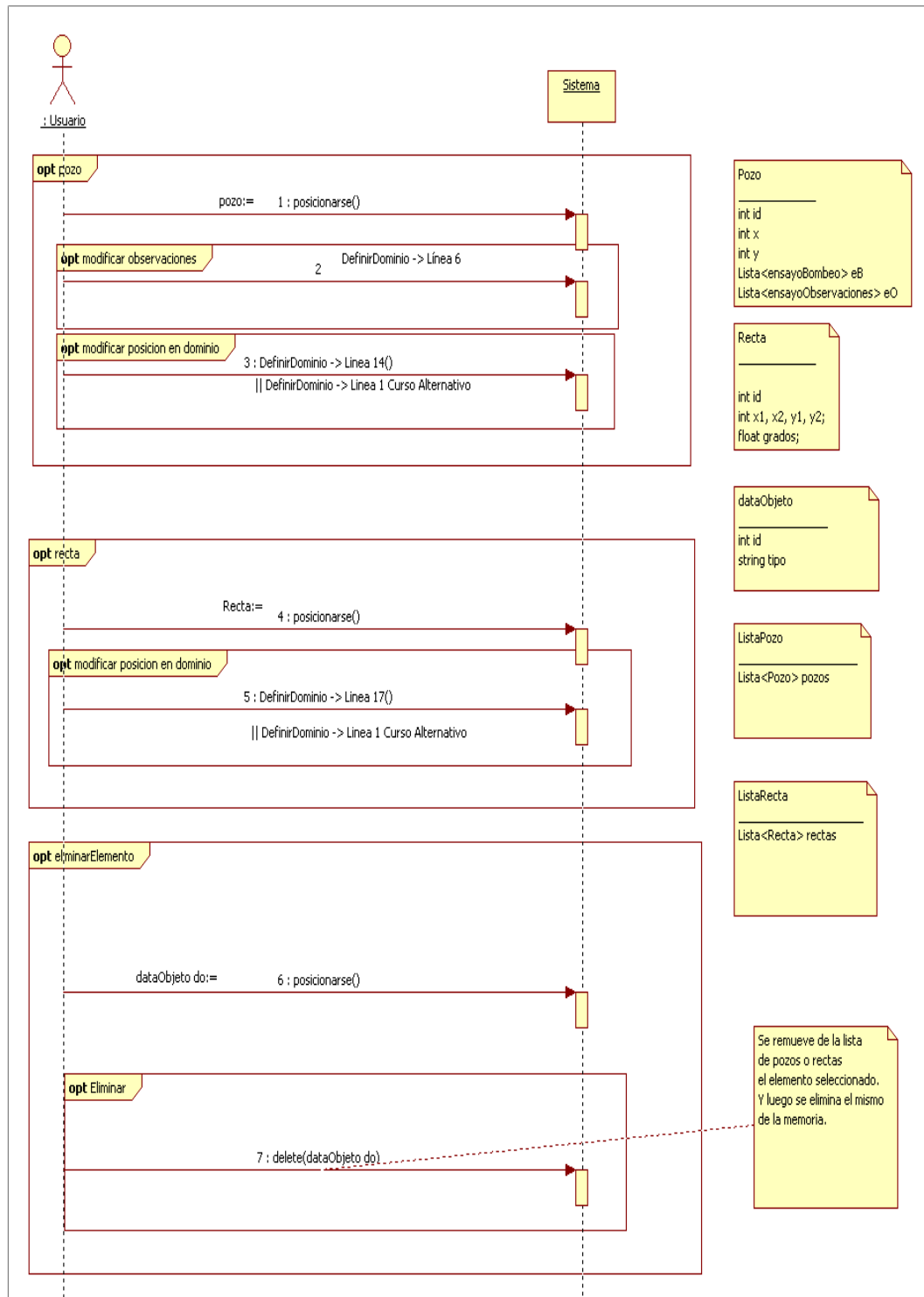
Post-condiciones:

1. Se creará un video.

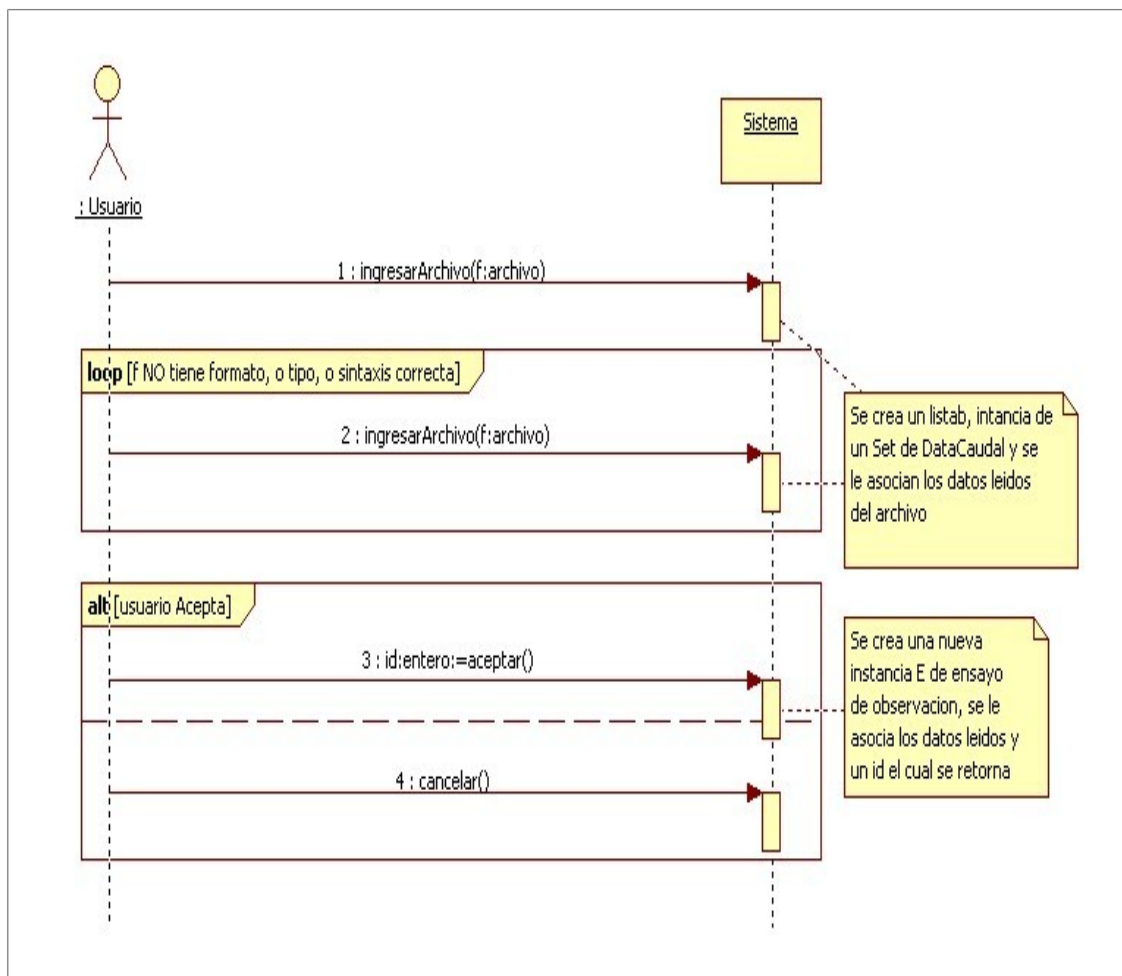
## 2.2 Diagramas de secuencia del sistema

### 2.2.1 Diagramas no críticos

#### 2.2.1.1 Modificar Dominio

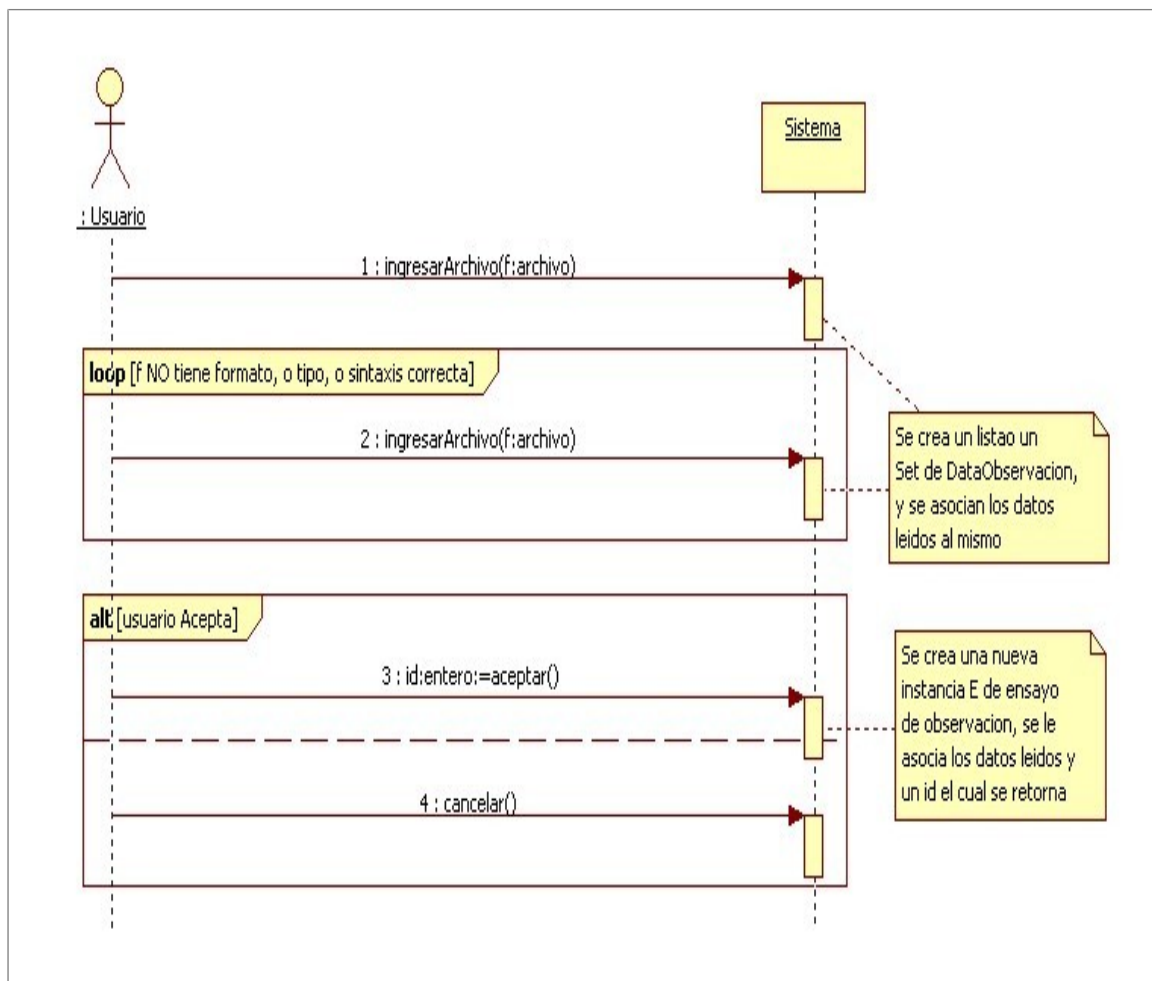


### 2.2.1.2 Importar caudal bombeado

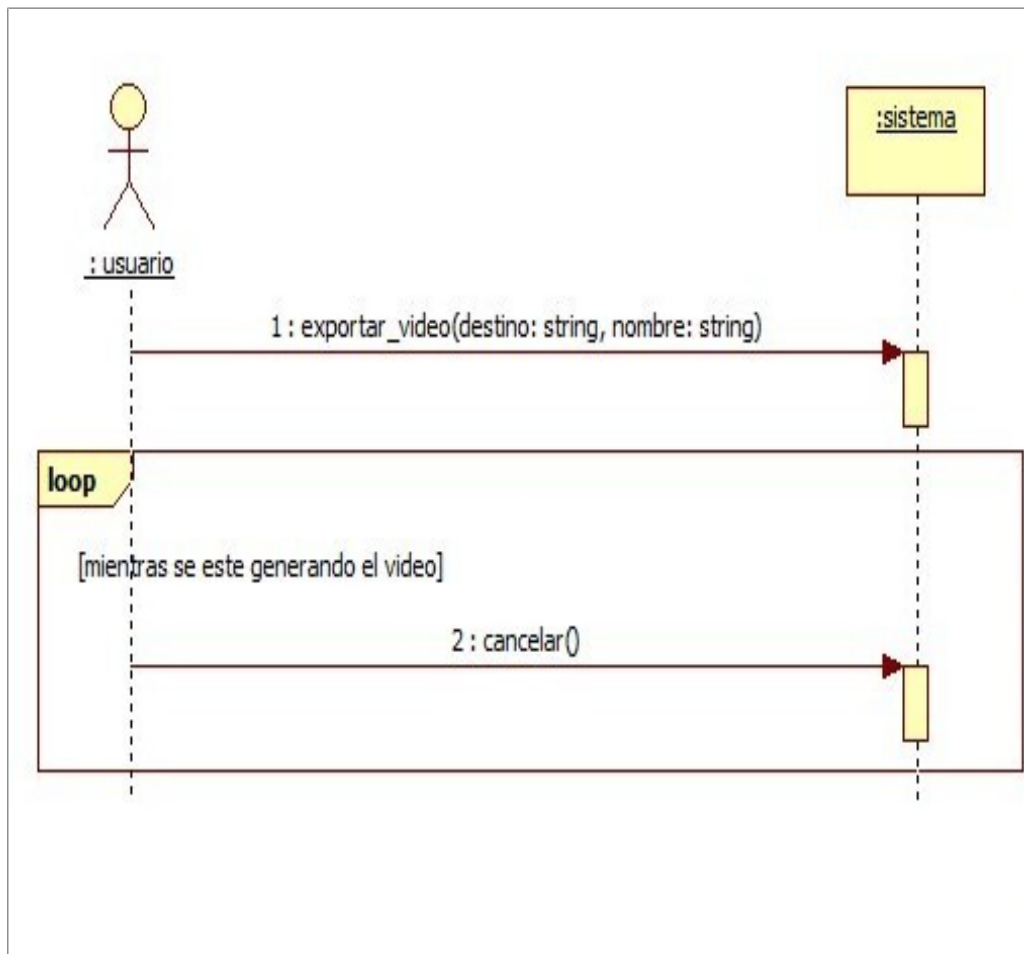




### 2.2.1.3 Importar Observaciones del ensayo



#### 2.2.1.4 Exportar Video



## **2.3 Evaluacion de lenguajes**

### **2.3.1 Python**

#### **2.3.1.1 Sintaxis**

Simple:

Python es simple y minimalista. Leer un buen lenguaje Python es casi como leer Inglés, aunque sea un Inglés muy estricto. Esta naturaleza de pseudo-código es uno de sus grandes fuertes ya que permite concentrarnos en la solución de un problema más que en el lenguaje.

Fácil de aprender:

Python es muy fácil de empezar a utilizar, al tener una extremadamente simple syntax.

#### **2.3.1.2 Requerimientos**

Intérprete Python

Python necesita un intérprete para correr sus programas. Este está disponible tanto en Windows como en Linux/Unix estando en este último instalado por defecto por la mayoría de sus más comunes distribuciones.

#### **2.3.1.3 Soporte y Documentación**

Respecto a lo que se refiere a documentación existe una vasta documentación referente a este lenguaje en una gran cantidad de sitios de Internet, así como también en los sitios oficiales de Python y también en bibliografía.

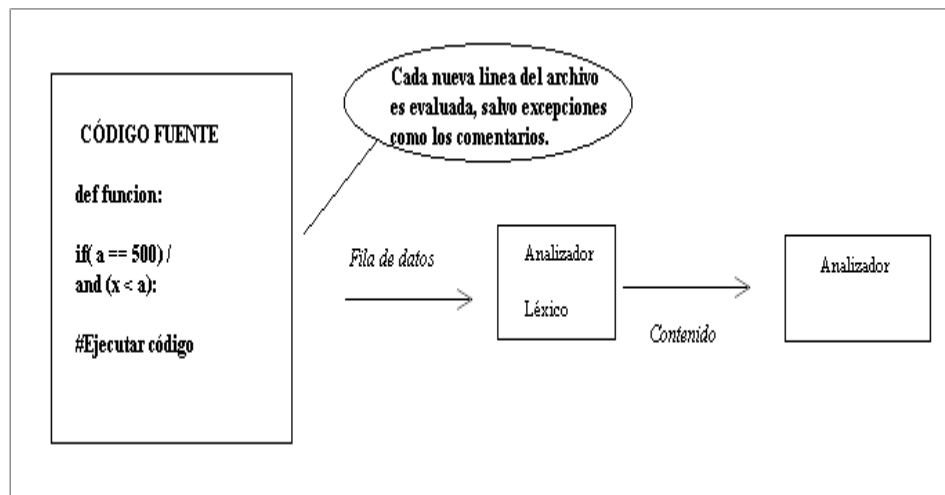
Así como sucede con otras tecnologías Open Source y software libre, existe una gran cantidad de foros, blogs y soporte sobre Python a la cual se puede recurrir y consultar, así como también tutoriales gratis y sitios de desarrollos dedicados a esta tecnología. Lo importante es que se puede encontrar esta información en idioma español.

### 2.3.1.4 Características

- Analizador léxico

*Phyton tiene dos elementos importantes para su funcionamiento: uno es el analizador léxico y el otro es el analizador de código.*

La función del analizador léxico es interpretar cada fila (línea) de código y determinar si es una línea lógica, física o un comentario. Si es una línea lógica enviará un evento al analizador con el código a ejecutar. En la figura 1 se muestra esquemáticamente la explicación.



Línea lógica, es a grosso modo, una sentencia a ejecutar o evaluar por el analizador. Ejemplo:

```

def funcion1:           #Línea lógica 1
    x = 100             #Línea lógica 2
  
```

Línea física: Es un conjunto de caracteres ASCII, u otro especificado explícitamente, cuya cota está determinada por cada nueva línea.

```

if a == 100):           #Línea física 1
    x = "Ejemplos em phyton" #Línea física 2
  
```

Relación entre línea física y lógica.

Una línea lógica puede estar compuesta por un conjunto de líneas físicas.

El caso más evidente es la concatenación o unión de una sentencia dividida en dos o más líneas.

Ejemplo:

```

Línea 1: If (a == 0) /
Línea 2: and (x > a):
  
```

En este caso la línea lógica, es decir la que el analizador va a interpretar, se compone de dos líneas físicas.

La declaración de enumeradores u arreglos también evidencia una clara diferencia entre línea lógica y física.

```
def x = [ "valor1", "valor2"  #La línea física se evidencia con cada nueva línea
        "valor3", "valor4"  #En tanto la línea lógica es la declaración del
        "valor5"]          #objeto que en este caso ocupa tres líneas de código
```

- Gratuito y Open Source

Python es un ejemplo de FLOSS (Free/Library and Open Source Software). Es decir, libre de distribuir copias del software, leer el código fuente, hacer cambios, usar pedazos del mismo en un nuevo programa y que sepas que puedes hacer estas cosas. FLOSS está basado en el concepto de comunidad que comparte el conocimiento. Esta es una de las razones por la que Python es tan bueno: ha sido creado y es constantemente mejorado por una comunidad que sólo quieren mejorarlo.

- Lenguaje de alto nivel.

Cuando se desarrolla en Python se evitan detalles de los lenguajes de bajo nivel como el manejo de memoria del programa.

- Portable

Dada su naturaleza Open Source, Python ha sido portado a muchas plataformas. Todo los programas Python pueden trabajar en cualquiera de estas, sin cambios en el código si se es lo suficiente cuidadoso como evitar servicios específicos de los sistemas.

Se puede usar Python en Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE e incluso PocketPC.

- Interpretado

Python, a diferencia de lenguajes como C++, no requiere compilación para generar un archivo binario. Tan sólo corre el programa directamente desde el código fuente. Internamente, Python convierte el código a una forma intermedia llamada bytecodes y luego se traduce a un lenguaje nativo de la computadora y esta la ejecuta. Todo esto, en verdad, hace el uso de Python más fácil al despreocuparnos por compilar, asegurando que las librerías están referenciadas y cargadas, etc. Esto también hace que los programas Python sean portables. Tan sólo copiando el programa Python hacia otra máquina éste funcionará.

- Orientado a Objetos

Python soporta *procedure-oriented programming* (programación orientada a procesos) así como también programación orientada a objeto. En lenguajes orientados a procesos (*procedure-oriented*), el programa se basa en funciones y procesos que no son otra cosa que pedazos de código reusables. En lenguajes *object-oriented*, el programa se basa en objetos que combinan datos y funcionalidades. Python tiene una potente pero simplista manera de manejar la orientación a objetos, especialmente comparado a grandes lenguajes como C++ o Java.

- Extensible

Si se necesita una pieza crítica de código extremadamente rápida o se tiene un algoritmo cuyo código no debe poder ser leído, se puede codificar esa pieza en C o C++ y luego usarla desde nuestro programa Python.

- Embebible

Se puede embeber Python en un programa C/C++ para darle características 'scripting' al mismo.

- Librerías extensibles

The Python Standard Library (conjunto de librerías estándares de Python) es gigantesca. Nos puede ayudar en tareas como trabajar con expresiones regulares, generación de documentación, pruebas unitarias, threading, base de datos, navegadores webs, CGI, ftp, email, XML, XML-RPC, HTML, WAV files, criptografía, GUI (graphical user interfaces), Tk y otras características dependientes de sistemas específicos. Todos esto estará disponible donde Python esté instalado. Esto es llamado la filosofía 'Batteries Included' de Python.

Además de las antes mencionadas, existen numerosas librerías de alta calidad como wxPython, Twisted, Python Imaging Library y muchas más.

### 2.3.1.5 Matplotlib

#### 2.3.1.5.1 Componente Matplotlib-matlab

Engloba a las funciones numéricas escritas en Python que permiten compatibilidad con los comandos escritos en MatLab.

En el siguiente enlace se encuentra la descripción de los comandos compatibles:

[http://matplotlib.sourceforge.net/api/mlab\\_api.html](http://matplotlib.sourceforge.net/api/mlab_api.html)

#### 2.3.1.5.2 Características

- Multiplataforma

Al igual que el lenguaje base, esta herramienta puede ser instalada en varios sistemas operativos. Linux tiene la ventaja de que la mayoría de sus distribuciones ya incluyen matplotlib y herramientas para potencializar su uso. Para el resto de los sistemas ya existen paquetes de instalación que incluyen todo lo necesario para ejecutarlo, por lo cual no es necesaria una configuración previa para comenzar a trabajar.

- Fácil de usar

Con unas pocas líneas de código el desarrollador podrá ver importantes resultados.

- Servidor de recursos

Las gráficas pueden ser generadas en un servidor y consumidas por uno o varios dispositivos clientes.

- Compatibilidad con otras Herramientas ya consolidadas

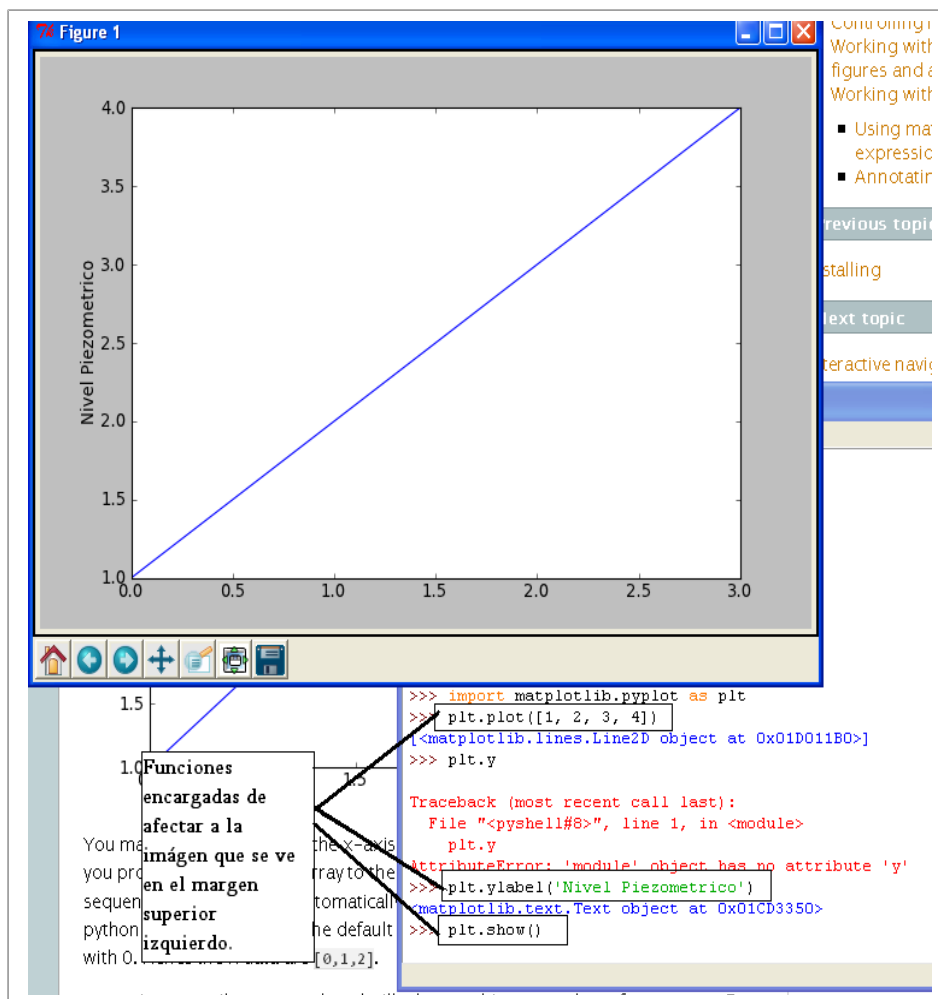
Como ya se resume en la descripción general, matplotlib-matlab es una librería que provee funciones idénticas a las existentes en MatLab, con lo cual se acelera y facilita la transcripción de código de este último a matplotlib.

### 2.3.1.5.3 Estructura Conceptual

Matplotlib se divide en tres partes, las cuales pasamos a enumerar:

**Pylab Interfaz:** Conjunto de funciones provistas por matplotlib.pylab que permiten generar gráficas con un código similar al de MatLab.

Cada función realiza un cambio a una figura determinada. Ejemplo:



**Matplotlib frontend o Matplotlib API:** Conjuntos de clases que hacen el trabajo pesado, es decir crear y manipular figuras.

**Backend:** Se encarga del formato de salida de la representación, debiendo manipular varias opciones de salida. Por ejemplo, una salida podría ser una ventana que represente el gráfico, podría estar embebida en alguna interfaz gráfica o bien podría ser un conjunto de imágenes PNG.

Por esto matplotlib es capaz de manejar varias salidas conforme a la necesidad del usuario.



#### 2.3.1.5.4 Clases principales en Matplotlib API

Así como conceptualmente esta herramienta puede ser dividida en tres partes, lo mismo sucede a nivel de clases. Existen tres tipos que cumplen funciones complementarias:

- `matplotlib.backend_bases.FigureCanvas`

Clase encargada de representar el área en donde se va a dibujar.

- `matplotlib.backend_bases.Renderer`

Es el objeto que sabe cómo se debe dibujar en el área.

- `matplotlib.artist.Artist`

Es el objeto que conoce cómo utilizar un objeto `Renderer` para dibujar sobre el `FigureCanvas`.

Las primeras dos clases son las encargadas de manejar los detalles de la comunicación con las herramientas de interfaz gráfica que el desarrollador esté utilizando. Mientras que `Artist` se encarga de la construcción de figuras, a partir del código de alto nivel provisto por el programador.

Las figuras creadas por `Artist` se dividen en dos grupos. Primitivos: son los objetos gráficos a ser representados dentro del `Canvas`, como líneas 2D, rectángulos o texto. Contenedores: Son los lugares donde se van a colocar los primitivos (ejemplo ejes, figuras o sub gráficas).

#### 2.3.1.5.5 Requerimientos

##### Python

Ante todo es esencial tener instalado Python en la máquina.

El enlace ofrecido para descargar los paquetes para los diferentes sistemas es:

<http://www.python.org/download/>

##### Numpy

Una vez instalado Python, en segundo orden se debe instalar esta librería, la cual proveerá estructuras de datos y funciones matemáticas para mejorar la performance en el código. Matplotlib requiere una versión de Numpy posterior a la 1.1.

##### Matplotlib

Instalar por último el paquete de matplotlib. En enlace de descarga es el siguiente: [sourceforge.net/projects/matplotlib/files/](https://sourceforge.net/projects/matplotlib/files/). Los desarrolladores que utilizan Debian o Ubuntu pueden instalarlo desde la línea de comandos:

**`sudo apt-get install python-matplotlib`**

#### 2.3.1.5.6 Soporte y Documentación

Esta herramienta posee su propia página de referencia:

<http://matplotlib.sourceforge.net/>

La documentación abunda en este sitio, el cual es manejado con una sintaxis fácil de comprender. Información disponible: Introducción, Guía de usuario, Tutoriales, Conceptos generales, Instalación, Ejemplos, FAQ's<sup>[1]</sup>

En cuanto a libros la página oficial recomienda Overview of Matplotlib for Python Developers, escrito por Sandro Tosi, disponible para su adquisición en Amazon<sup>[2]</sup>.

---

1 FAQ refiere a las preguntas hechas frecuentemente alrededor de un tema o sujeto determinado.

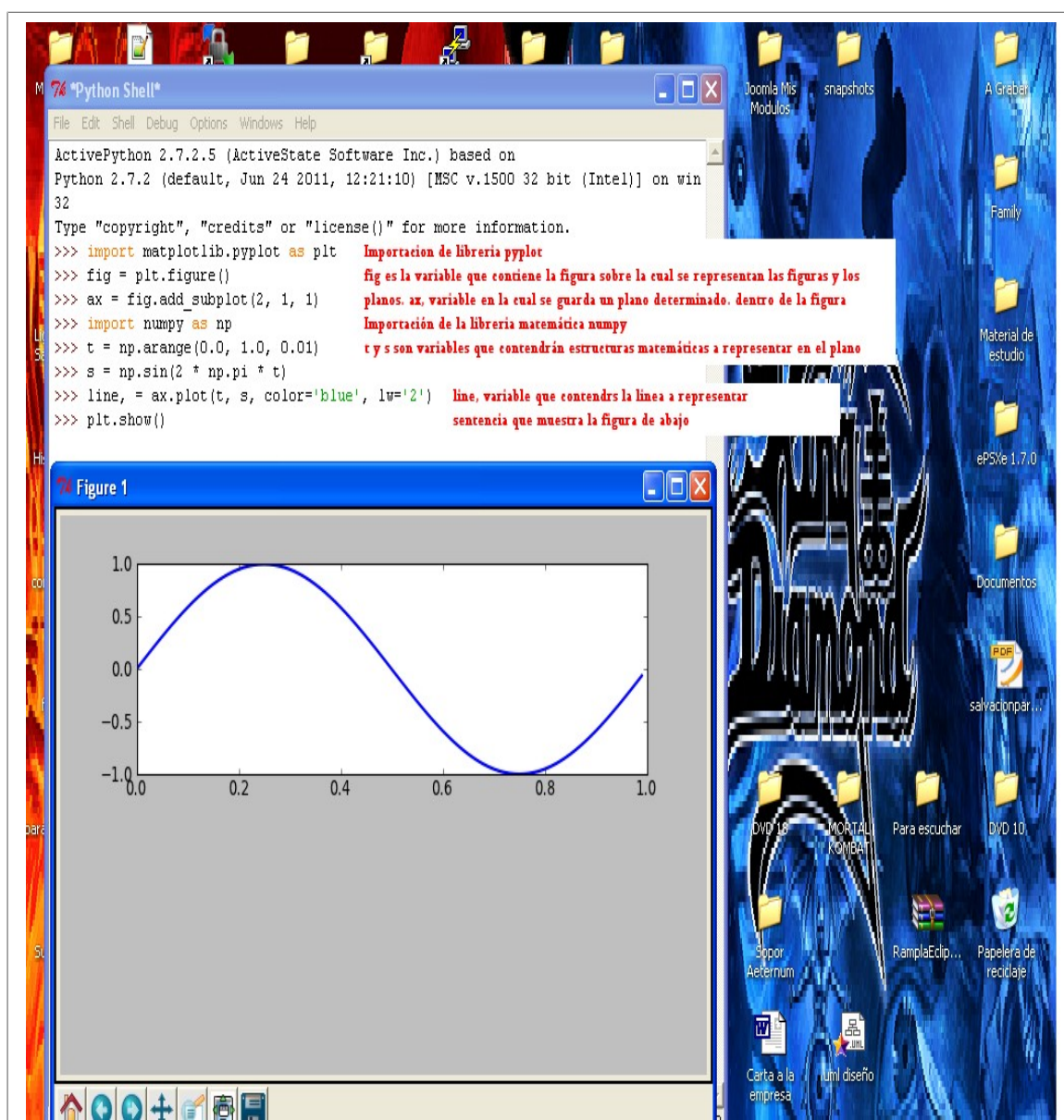
2 Sitio web dedicado a la compra y venta de productos. Es el sitio número uno para llevar a cabo negocios en el ámbito internacional.

### 2.3.1.5.7 Ejemplo simple con Matplotlib

El siguiente ejemplo demuestra un programa que contiene en ocho líneas de código el potencial de esta herramienta.

A través de dos estructuras matemáticas,  $t$  y  $s$ , es creada una representación gráfica, en este caso una línea de dos dimensiones, la cual es visualizada en el sub plano especificado,  $ax$ .

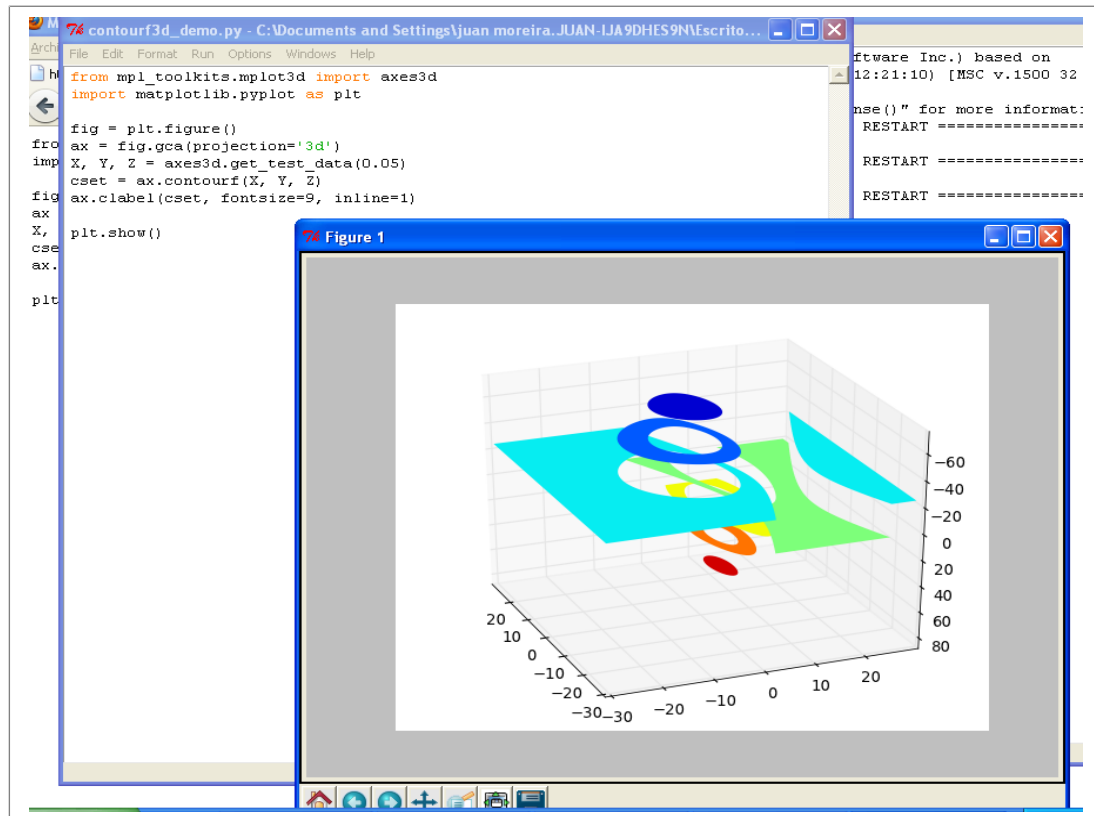
Cada elemento, en matplotlib, al ser tratado como una variable, puede ser modificado en tiempo de ejecución; por ejemplo modificar variables en  $t$  o  $s$ , o eliminar la línea del plano.



### 2.3.1.5.8 Ejemplo simple en 3D

El graficado en tercera dimensión requiere la utilización de la librería `mpl_toolkits.mplot3d` para dibujar los ejes en tercera dimensión.

Como se ve en la figura de ejemplo, el código necesario para representar datos en tercera dimensión no excede en gran medida al requerido para graficar en 2D.



(En la figura se representan los datos con contornos. También se pueden crear gráficas de dispersión, de barras, y superficie)

### 2.3.1.5.9 Animación en matplotlib

`matplotlib.animation` es la clase que envuelve la creación de animaciones. Los métodos de esta clase permiten la creación de animaciones de una manera más estructurada.

La limitante de este entorno de desarrollo es que es de reciente creación, por lo tanto carece de documentación y sus funcionalidades hasta el momento son tres, las cuales pasamos a describir:

`matplotlib.animation.FuncAnimation(fig, func, frames=None, init_func=None, fargs=None, save_count=None, **kwargs)`

La clase FuncAnimation crea una animación mediante la llamada a la función func.

```
matplotlib.animation.TimedAnimation(fig, interval=200, repeat_delay=None,  
repeat=True, event_source=None, *args, **kwargs)
```

Es una subclase que permite animaciones basadas en el tiempo, lo cual posibilita dibujar un nuevo frame cada cierto intervalo de tiempo, expresado en milisegundos.

```
save(filename, fps=5, codec='mpeg4', clear_temp=True, frame_prefix='_tmp')
```

Salva en una película las imágenes generadas en la animación.

#### 2.3.1.5.10 PyQt4

Es la herramienta para la interfaz de usuario elegida, a continuación la explicación del framework para gráficas, de utilidad para la elaboración del dominio.

##### 2.3.1.5.10.1 Graphics View Framework

Provee una superficie para administrar y manipular una amplia gama de items personalizados, en dos dimensiones, y una vista para visualizar los items, con soporte para las acciones de zoom, y rotación.

Este entorno de desarrollo, orientado a objetos, posee una arquitectura de propagación de eventos, con lo cual los items pueden manejar sus propios eventos como el click, doble click, entre otros.

Graphics View utiliza un árbol BSP<sup>[1]</sup> para proveer el descubrimiento de items sobre la vista de manera veloz, como resultado de esto se pueden visualizar superficies extremadamente largas en tiempo real, igualmente con millones de items existentes dentro de ellas<sup>[2]</sup>.

#### 2.3.1.5.11 Ventajas

MatPlotLib es una herramienta potente que permite con poco código elaborar gráficas en 2 y 3 dimensiones.

Se pueden generar animaciones y a partir de los frames de las mismas guardarlas en una película, con tan sólo llamar a una función.

Cualquier función matemática puede ser expresada en el plano.

Permite la integración de figuras de 2 dimensiones, como una línea, en gráfica de tres dimensiones.

---

1 Para un informe detallado sobre este tipo de árbol binario consultar la referencia 58.

2 Para un informe técnico sobre su arquitectura consultar la referencia 57.

En una misma figura es posible crear varias gráficas, en cualquiera de las dimensiones disponibles.

Permite una más fácil integración con MatLab, facilitando a través de matplotlib-mlab funciones semejantes a las existentes en este último.

#### *2.3.1.5.12 Referencias*

Overview of Matplotlib for Python Developers - Libro de Sandro Tosi

[http://www.packtpub.com/matplotlib-python-development/book?  
utm\\_source=matplotlib.sourceforge.net&utm\\_medium=link&utm\\_content=pod&utm\\_c  
ampaign=mdb\\_002124](http://www.packtpub.com/matplotlib-python-development/book?utm_source=matplotlib.sourceforge.net&utm_medium=link&utm_content=pod&utm_campaign=mdb_002124)

## **2.3.2 Java**

### **2.3.2.1 Descripción general**

Este lenguaje toma mucho de su sintaxis de C y C++, pero tiene un modelo de objetos más simples, ya que elimina herramientas de bajo nivel como el manejo de punteros de forma directa. También ha sido influido por lenguajes como Smalltalk y Objective-C.

Para cada sistema operativo y hardware específico existe una máquina virtual diferente lo que permite que una aplicación Java pueda correr en cualquier dispositivo, desde PC de escritorio a notebooks, smartphones, tablets, etc.

La implementación original del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías.

Entre finales de 2006 y principios del 2007, Sun Microsystems comenzó el relanzamiento de la plataforma y tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process.

En abril de 2009 Oracle adquirió Sun Microsystems, lo que pone en riesgo la libertad que tiene este software actualmente, junto a la continuidad de desarrollos existentes basados en este lenguaje y además pone en juego la posible migración de programadores de Java a otras plataformas. Oracle tiene bajo su control las versiones de Java posteriores a la 6.

### 2.3.2.2 Características

- Interpretado

Los programas Java no necesitan ser autosuficientes y no tienen por qué incluir todo el código máquina que se ejecuta en el ordenador. Son compilados creando *bytecodes* compactos y son estos los que la JVM lee e interpreta para ejecutar el programa. Cuando descargamos una *applet* Java de la red Internet lo que realmente descargamos es un archivo de *bytecodes*. De esta forma, el programa Java puede ser muy pequeño, ya que todo el código de máquina necesario para ejecutarlo está ya en el ordenador de destino y no tiene que descargarse.

- Orientado a objetos

Uno de los objetivos con los que se concibió este lenguaje es que fuera orientado a objetos y que tomara lo mejor de otros lenguajes POO como C++. Sin embargo, en Java todo es un objeto o reside en una clase; por ejemplo, para hacer un “hola mundo” se necesita declarar una clase con un método. Esta característica permite entre otras cosas la reutilización del código y facilita la detección de errores en la programación.

- Multiplataforma y portabilidad

Esta característica también forma parte de su filosofía tal como lo dice el axioma de Java, “write once, run anywhere”. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo (PC, Linux, Windows, MAC, notebooks, smartphones, etc) independientemente de su hardware o sistema operativo.

Para distribuir Java entre una gran variedad de ordenadores, Sun sólo tuvo que reescribir JVM para que funcionara en esos ordenadores. Dado que su programa está almacenado en un archivo de *bytecode*, se ejecutará en cualquier dispositivo en el que JVM esté instalado.

- Fácil de usar

Otro de los objetivos que persigue su filosofía es que tiene que ser fácil de usar y realmente para los desarrolladores resulta una tecnología fácil de asimilar.

- Trabajo en red

La versatilidad y eficiencia de la tecnología Java, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes. La plataforma incluye por defecto soporte para trabajo en red y se diseñó para ejecutar código en sistemas remotos de forma segura.



### 2.3.2.3 Librería Jzy3D

#### 2.3.2.3.1 Funcionalidades disponibles

- Gráficas de superficie, de barras, de dispersión, 2D, 3D ...
- Gráficas comunes a partir de varias primitivas (esferas, triángulos, polígonos, líneas, puntos)
- Barras de color y mapeo de color a objetos
- Funciones y gráficas de contorno
- Animaciones de estructuras de objetos (superficies, series de líneas, etc)
- Planos con textura
- Soporte experimental para métodos de máquina de vector y bezier teselación
- Interacción del mouse con los objetos (selección) y con las gráficas (rotación, zoom, escala)
- Interacción del teclado con las gráficas (rotación, zoom y escala)
- Se pueden poner rótulos a los ejes y poner referencias.
- Tooltips(mensajes sobre gráficas)
- Imágenes de fondo
- Renderizadores personalizables
- Fácil integración con cualquiera de las librerías GUI's : AWT, Swing, o Eclipse RCP
- Soporte para luz experimental.

#### 2.3.2.3.2 Características destacables

Este software utiliza la nueva licencia BSD, que es una licencia de software libre permisiva con tres cláusulas, menos restrictiva que la licencia GPL, ya que al contrario de ésta, permite el uso del código fuente en software no libre.

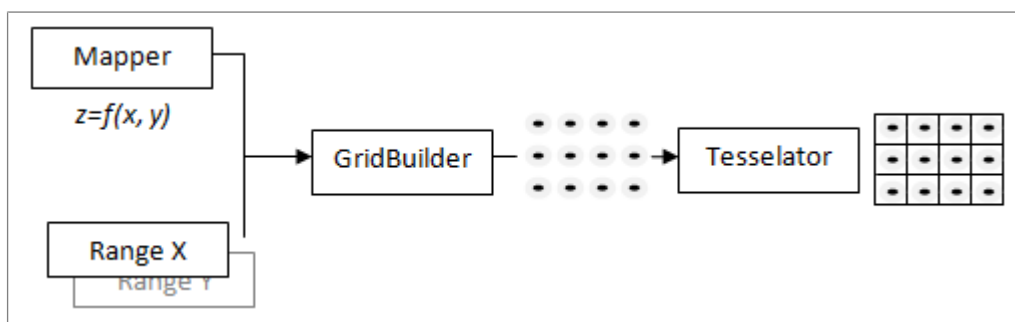
Esta librería requiere que instales la Java Open GL para cada arquitectura diferente; tienes que conseguir la librería jogl.dll para Windows e instalarla en el mismo directorio donde están instaladas la JDK (copiar el archivo dll). Los links para descargar la librería dll hoy por hoy están rotos.

Esta librería permite hacer animaciones utilizando los thread de Java que son como timers donde se puede controlar a medida que pasa el tiempo qué datos se pueden mostrar; no incluye una clase para poder exportar las animaciones generadas. Sí permite, como se mencionó más arriba, generar una secuencia de imágenes en formato png, a partir de las cuales se puede renderizar un video utilizando alguna otra librería Java.

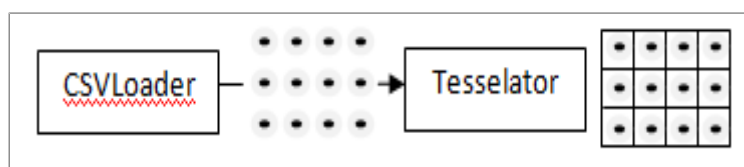
Otra característica importante es que es muy fácil incrustar las gráficas generadas en un formulario de la interfaz gráfica de una aplicación de escritorio.

### 2.3.2.3.3 Funcionamiento general

Como se muestra en la figura de abajo, una clase Mapper se encargará de describir la función que se quiere representar. Esta con un conjunto de rangos para escalar el plano, son pasados a una clase llamada GridBuilder, la cual se encarga de generar las coordenadas de entrada para un Tesselator. Tessellation <sup>[1]</sup> es el proceso de generar polígonos a partir de estas coordenadas, lo cual se puede hacer a través de varias estrategias.



Además es posible, gracias al OpenCSV, proveerle simples coordenadas de entrada desde un archivo CSV externo, donde los datos provistos tengan un formato acorde a los requerimientos del tessellator.



### 2.3.2.3.4 Soporte y documentación

La documentación oficial de esta librería está toda en idioma Inglés y se puede localizar en la siguiente url: <http://code.google.com/p/jzy3d/>. También hay disponibles variadas demostraciones hechas. La documentación específica para el programador acerca de las clases y métodos que utiliza esta librería se encuentra en dos javadocs: <http://jzy3d.free.fr/javadoc/0.8.1/> y <http://jzy3d.free.fr/javadoc/0.8.2/>.

Esta herramienta tiene un grupo de Google propio donde postear dudas o problemas técnicos surgidos al utilizar esta herramienta. Cualquier usuario con una cuenta de Google puede subscribirse al grupo ingresando a esta url <http://groups.google.com/group/jzy3d>. En la página también encontramos el mail de contacto del que realiza el mantenimiento del proyecto [martin.pernollet@gmail.com](mailto:martin.pernollet@gmail.com) para comunicarnos de forma directa en caso de necesitar soporte extra.

1 Tessellation se entiende por mosaico o patrón de varias figuras que completan el plano sin superponerse ni dejar espacios en blanco. Por extensión se llama mosaico a cualquier obra realizada con fracciones diversas.

### 2.3.2.4 Librería Processing

#### 2.3.2.4.1 Funcionamiento general

Para poder crear una animación con esta herramienta es necesario crear una clase en nuestro proyecto que herede de PApplet y que opcionalmente contenga un pseudo-atributo del tipo MovieMaker.

Esta clase será la encargada de renderizar el video y utiliza básicamente dos métodos: setup() y draw() para generar los videos. Dentro del método setup() se crea una nueva instancia de un objeto MovieMaker y se inicializan atributos. El método draw() es utilizado para ir agregando los frames del video al objeto MovieMaker creado. El método draw funciona básicamente como un bucle infinito que irá generando el video hasta que se llame al método finish() de la instancia de MovieMaker.

```
public class Embedded extends PApplet {  
    public void setup() {  
        // original setup code here ...  
        size(400, 400);  
        // prevent thread from starving everything else  
        noLoop();  
    }  
    public void draw() {  
        // drawing code goes here  
    }  
}
```

La clase PApplet (Processing Applet) hereda de la clase Applet de AWT por lo que dentro de la misma no es posible manejar ni utilizar controles de swing pero sí es posible utilizar controles de un Frame de AWT. Sin embargo es posible incrustar una PApplet dentro de un JFrame.

#### 2.3.2.4.2 Documentación y soporte

- Documentación específica de la librería:

<http://processing.org/reference/>

Libros: <http://processing.org/learning/books/>

La wiki donde está toda la documentación:  
[http://wiki.processing.org/w/Main\\_Page](http://wiki.processing.org/w/Main_Page)

- Ejemplos de aplicaciones desarrolladas utilizando processing:  
<http://www.openprocessing.org/>  
<http://www.creativeapplications.net/>
- También hay información en redes sociales:  
<http://vimeo.com/tag:processing.org>  
<http://www.flickr.com/groups/processing/>

### 2.3.2.5 Soporte y Documentación

- Sitio oficial de Java en español <http://java.com/es>
- La Biblia de Java 2 de Holzner, Steven.

Nº páginas: 960 Lengua: Español

Nº Edición: 1ª Fecha de edición: 2005-09-30

Plaza edición: MADRID Editorial: Anaya Multimedia

La biblia de Java 2 es un libro introductorio a la tecnología que muestra toda la historia del lenguaje de programación Java. Contiene no sólo toda la sintaxis de Java, desde la declaración de variables hasta temas de orientación a objetos avanzada, sino también el lenguaje Java en el mundo real mostrando ejemplos.

- Aprendiendo Java - Libro (edición libre) para estudiantes y profesores.

Es un libro orientado a estudiantes y profesores que quieran tener conocimientos en Java que comienza de un nivel muy básico explicando conceptos básicos de programación hasta introducirse en temas de programación orientada objetos propios de Java.

<http://www.compunauta.com/forums/linux/programacion/java/AprendiendoJava.pdf>

- The java programming language de Arnold y Gosling.

Esta es la versión en castellano de "The java programming language", de Arnold y Gosling. Es altamente recomendable para todo aquel que esté por iniciarse en el lenguaje, así como aquellos que ya están trabajando con el mismo. No es una de las últimas ediciones, -lo notarán aquellos que hayan leído el e-book en Inglés-, pero es realmente un recurso muy bueno.

<http://es.scribd.com/doc/6975024/El-Lenguaje-de-Programacion-Java-Libro-Espanol>

- Introducción a la Programación orientada a objetos con Java de Thomas

C. Wu

Idioma: Español

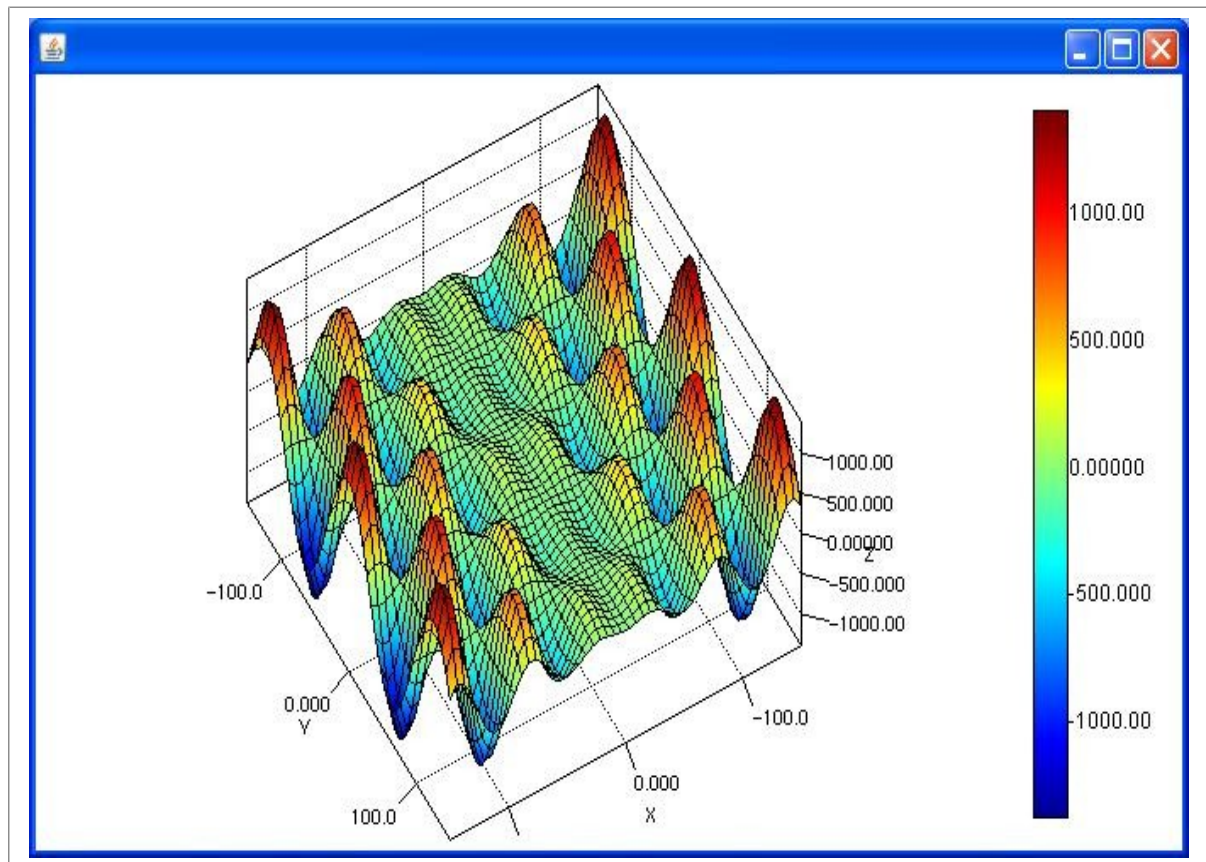
864 páginas

Fecha de publicación: 25 Mayo de 2001.

Editorial: McGraw-Hill

### 2.3.2.6 Ejemplos de implementación, casos vistos

#### 2.3.2.6.1 Jzy3D: Ejemplo 1

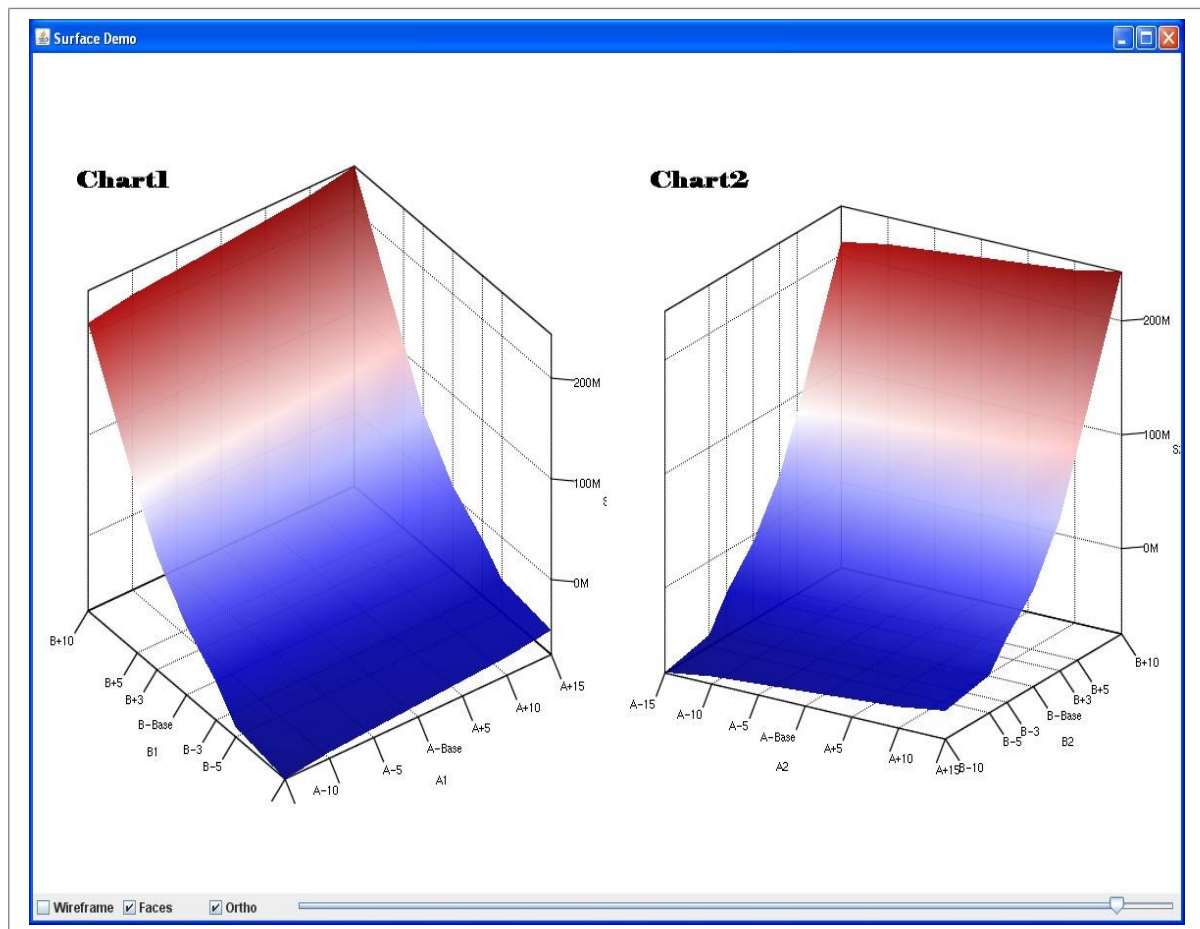


Ejemplo sencillo de gráfica 3d, útil para la parte de las simulaciones. Fue generado en Netbeans. En este ejemplo se definió una función matemática:

$$f(x,y)=10.\text{sen}(x/10) * \cos(y/20) * x$$

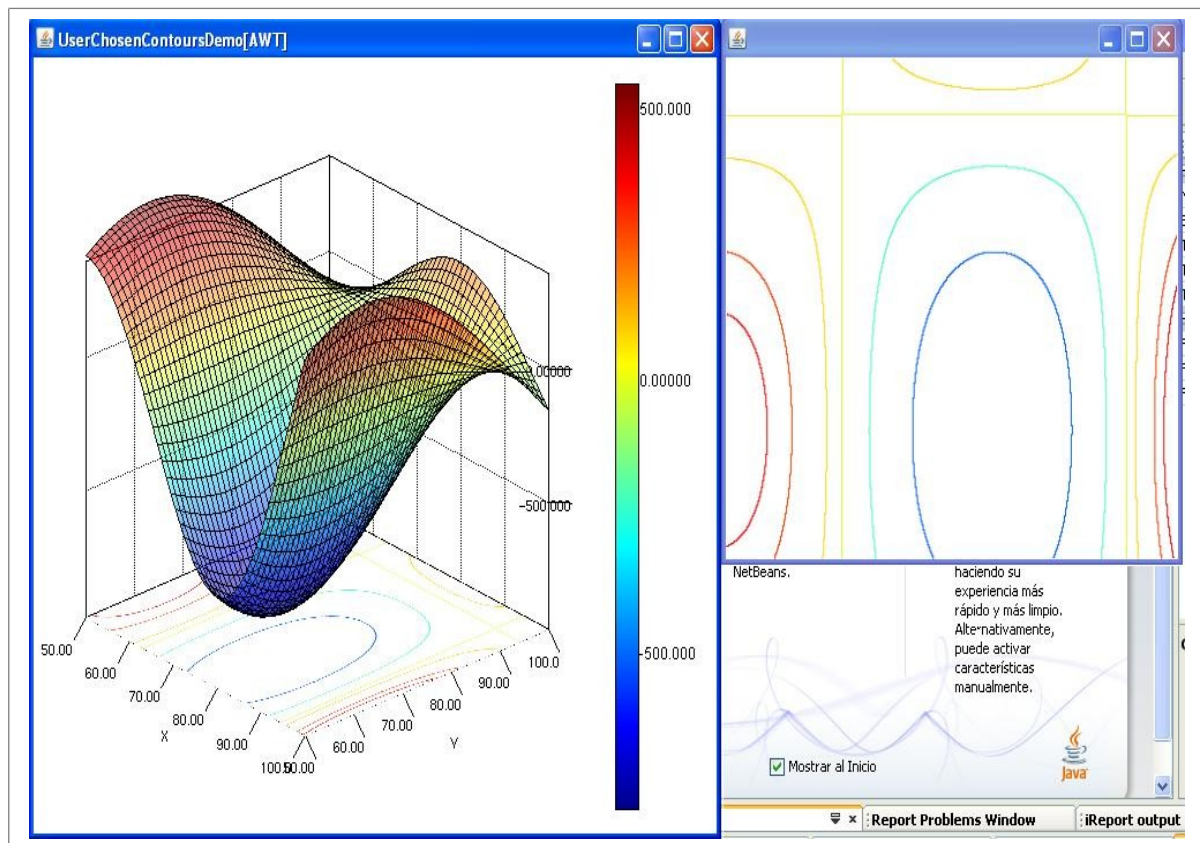
y un dominio para representarla. El ejemplo lleva unas 60 líneas, es un formulario de swing con la gráfica incrustada. No genera ninguna animación, ni nada de interactividad ya que es un ejemplo muy básico.

### 2.3.2.6.2 Jzy3D: Ejemplo 2



Este ejemplo tiene animación. Se puede cambiar el grado de transparencia al gráfico y rotarlo en cualquier dirección con el uso del mouse para poder ver desde varios puntos el comportamiento del gráfico. También se puede animar el gráfico, al hacer doble click sobre el mismo: esto lo que hace es que comience a rotar lentamente en una dirección. El ejemplo está formado por 320 líneas.

### 2.3.2.6.3 Jzy3D: Ejemplo 3



Este ejemplo se aproxima más a lo que tenemos que lograr en nuestro proyecto. Permite crear gráficas de contornos que servirían para modelar las curvas de nivel. En este se separó en dos paneles la vista de la gráfica en 3d y la de la gráfica en 2d, cosa que nos resultara útil en nuestro proyecto.

### 2.3.3 C/C++

#### 2.3.3.1 Características

- Propósito:

Es un lenguaje de propósito general, por lo tanto puede ser utilizado para fines diversos y versátiles; se puede desarrollar desde una simple calculadora, o código para censar la temperatura desde un micro controlador, hasta un sistema operativo.

- Portabilidad:

ANSI C es un estándar que asegura la funcionalidad del lenguaje en cualquier máquina y sistema operativo.

- Lenguaje estructurado:

El código es interpretado línea a línea, sin la posibilidad de cambiar la secuencia de ejecución. Esto es un problema para la realización de programas complejos.

- Lenguaje procedimental:

Otro concepto que se aplica en C son las llamadas a procedimientos: las funciones. Con esta característica, en una sola línea de código se puede invocar a una función que procese la petición y devuelva un resultado calculado a partir de un algoritmo de ejecución.

- Orientación a objetos:

**C++**, provee la posibilidad de crear objetos que representen un elemento extraído de la realidad; este elemento surge del análisis del problema, con atributos y métodos específicos.

- Escalable:

Dado a que sus especificaciones están estandarizadas, el lenguaje puede ser escalado, es decir, se puede personalizar para que realice funciones requeridas por el programador y que todavía no existan. La forma más común de escalar el lenguaje es creando librerías de propósito general.

- Librerías funcionales no estándares:

Existen librerías estándares, con funcionalidades de entrada / salida de datos, operaciones algorítmicas y para manejo de arreglos o cadenas de caracteres. No obstante, librerías más específicas, como las de graficado, deben ser descargadas desde Internet. Un punto a favor en cuanto a librerías, es que dada la cantidad de programas existentes y disponibles, de seguro habrá alguna que se ajuste a cada requerimiento específico que se necesite, evitando de esa forma reinventar la rueda.



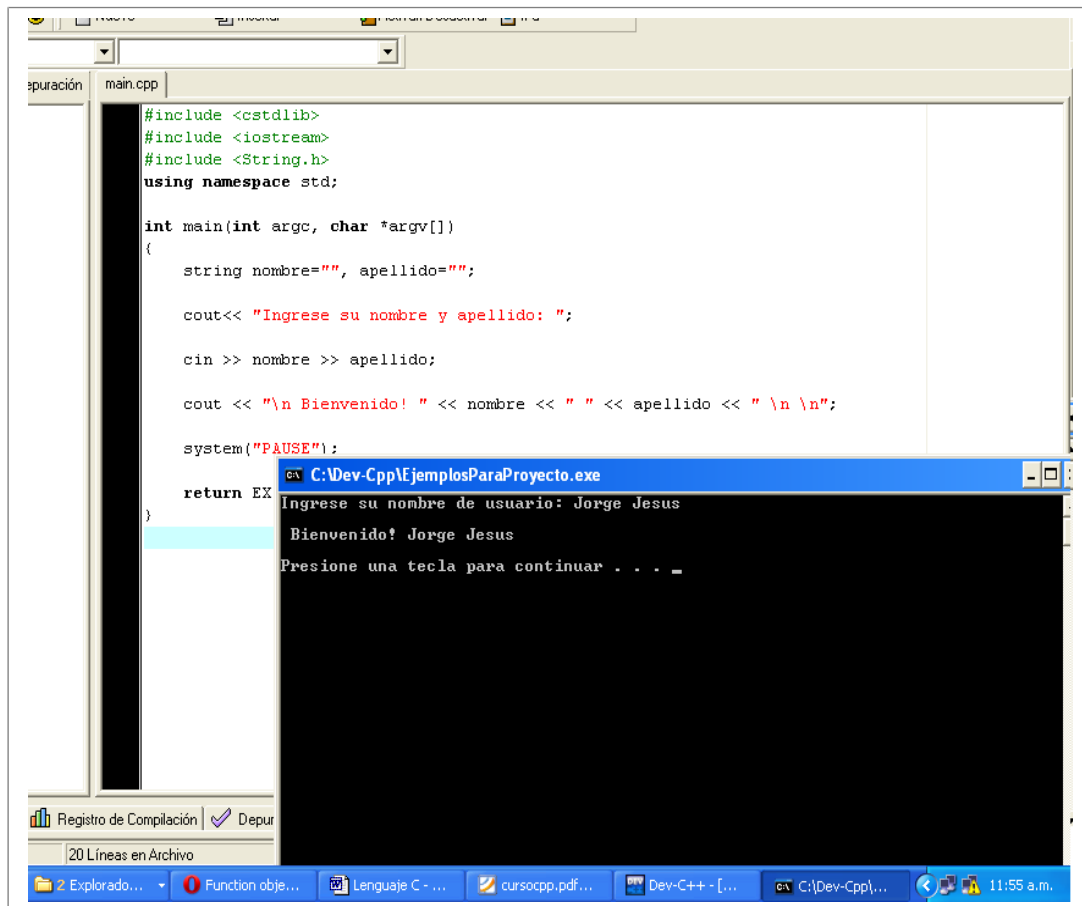
### 2.3.3.2 Librerías c/c++

#### 2.3.3.2.1 *iostream.h*

//Salida de datos a la pantalla del usuario.  
cout >> "Ingrese su nombre de usuario"

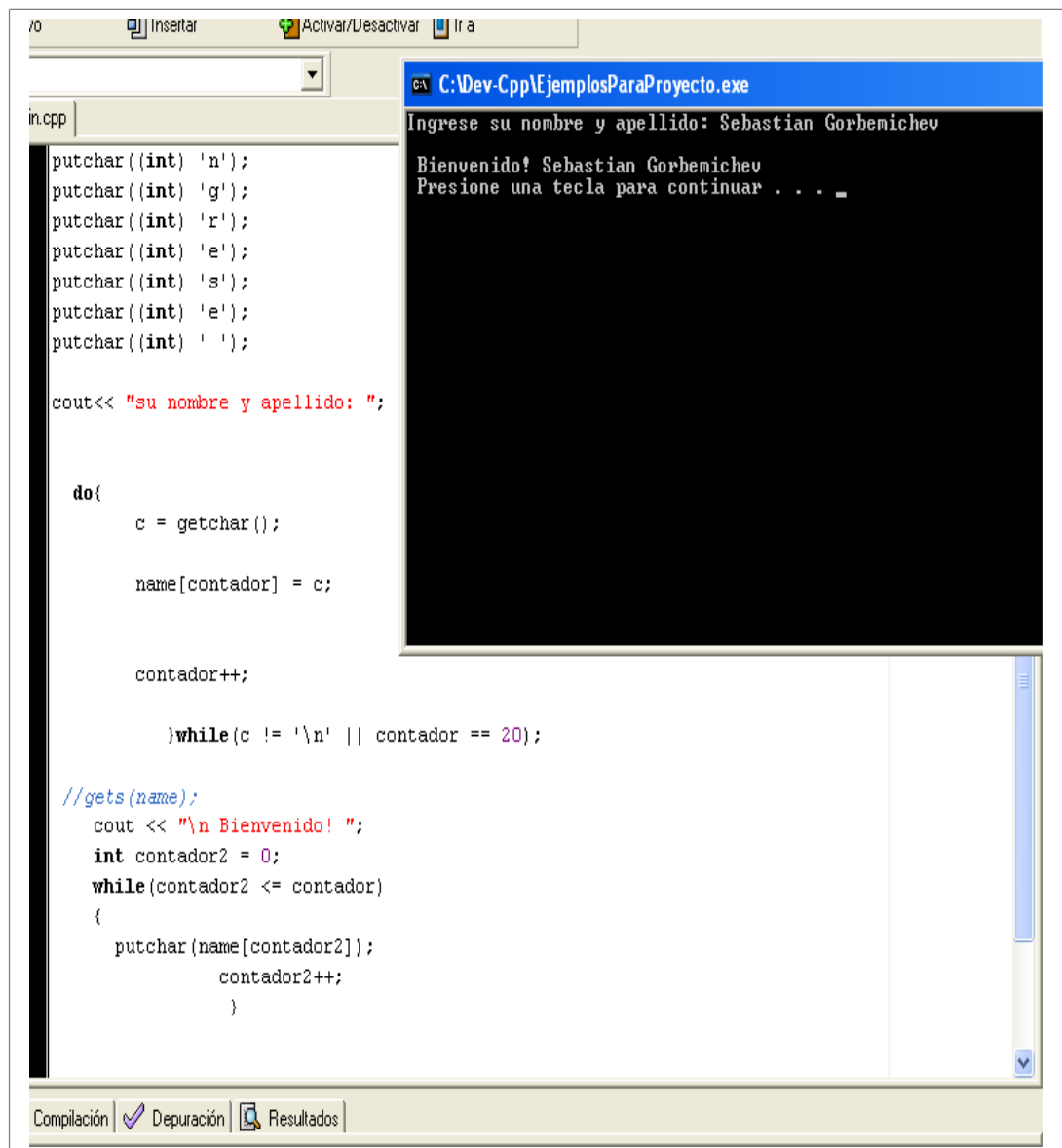
//Escritura de variables  
cin >> mensaje;

#### 2.3.3.2.2 *String.h*



(Ejemplo en DevC. Se observa la inclusión de las librerías estándares al principio del código y la posterior utilización de los literales string, cin, cout)

### 2.3.3.2.3 stdio.h



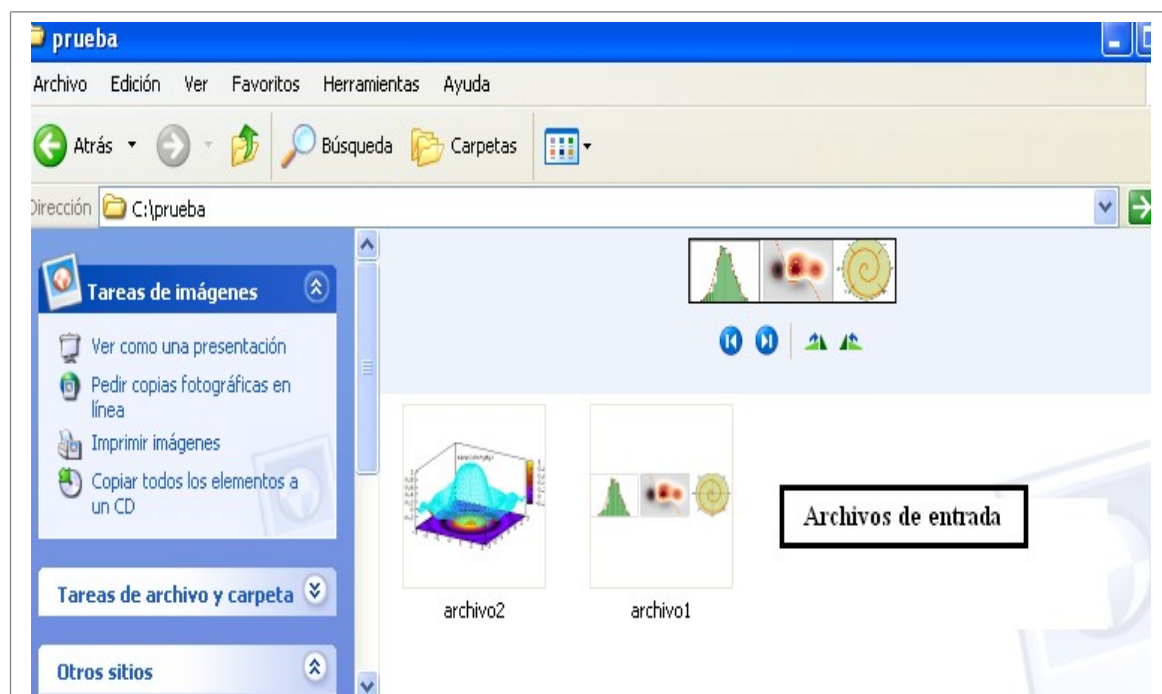
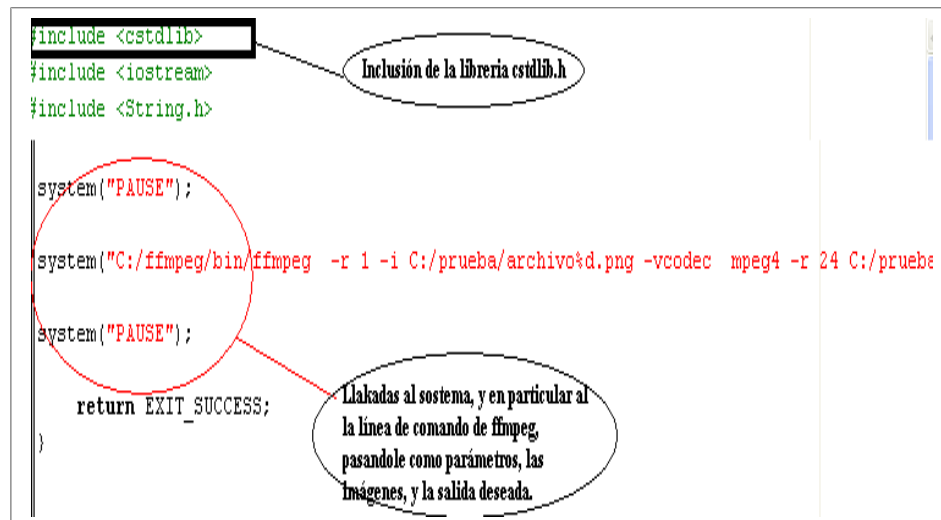
(Mismo resultado que la imagen anterior, utilizando stdio)

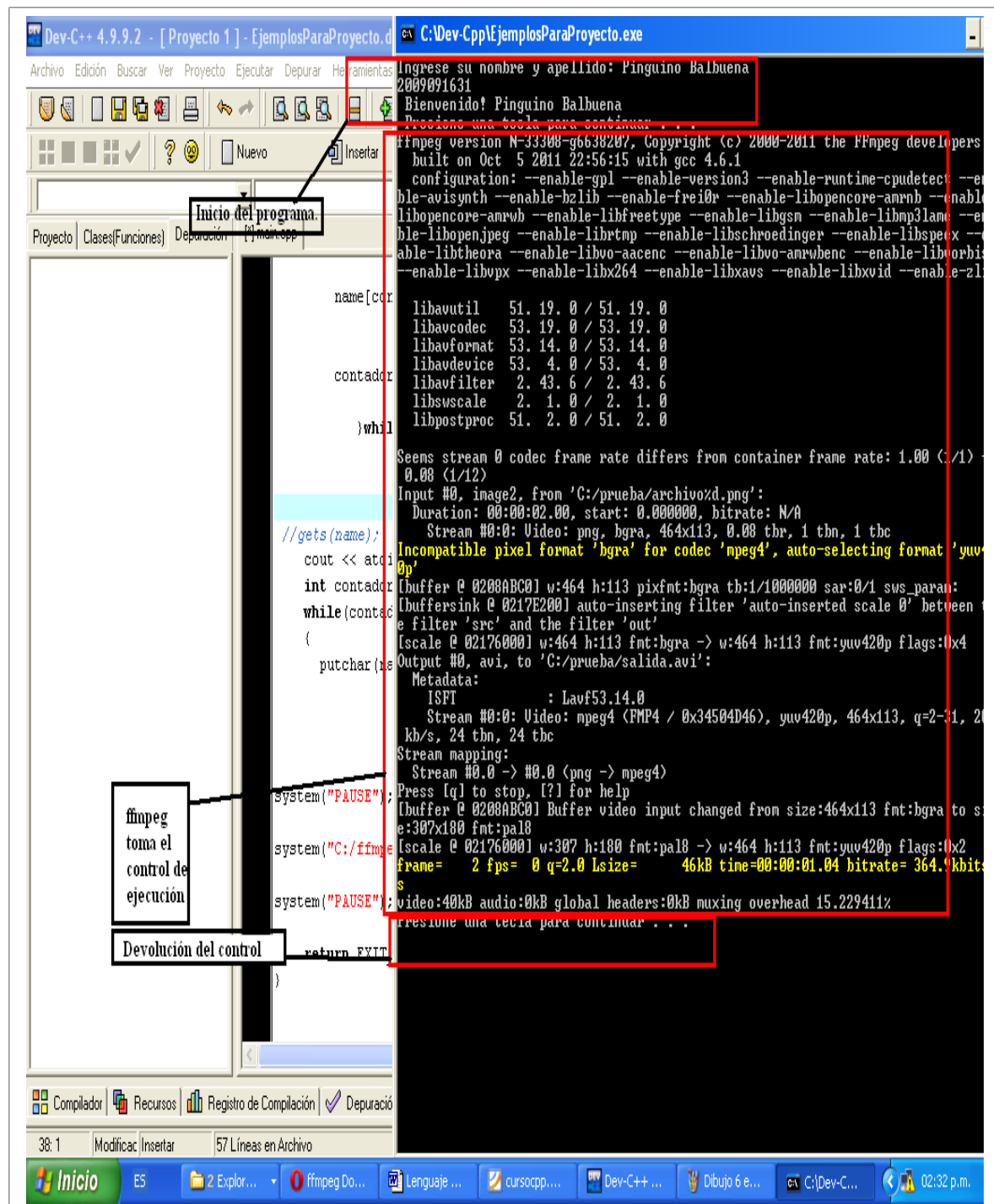
### 2.3.3.2.4 *stdlib.h*

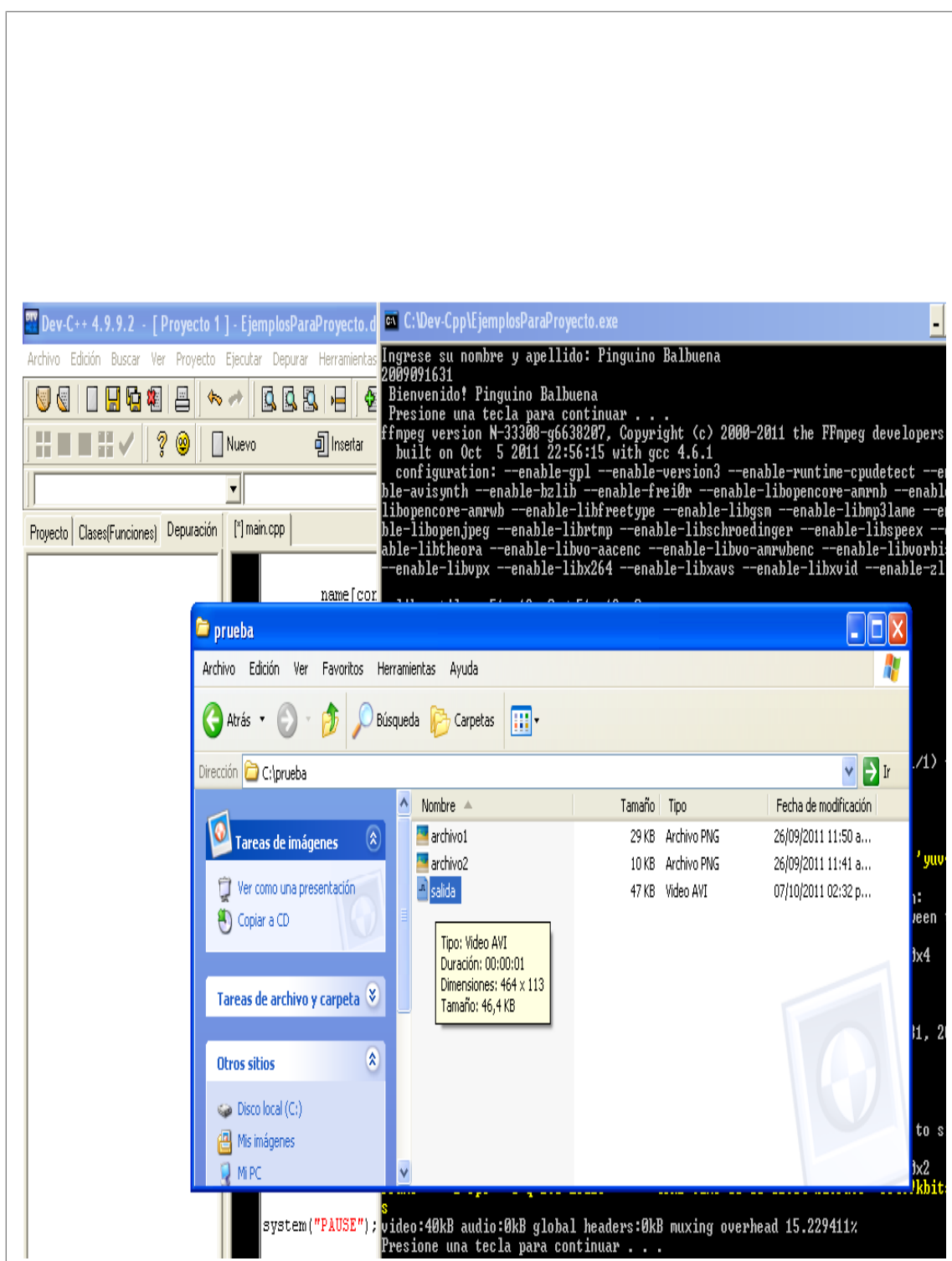
Ejemplo: Conversión de imágenes a archivo de video, en formato AVI.

Una de las exigencias del proyecto es la creación de videos a partir de los datos graficados. Estas pueden generar imágenes secuenciales, las cuales serían tomadas como datos de entrada, y procesarlas de tal manera que del conjunto de imágenes, se genere un video.

Las líneas de código necesarias para demostrar este ejemplo son insignificantes:







### 2.3.3.3 Ejemplo de procesamiento de archivos:

El ejemplo se realiza procesando datos de diferentes pozos de observación.

The screenshot shows a text editor window with a file named 'example.txt'. The data is organized into two sections: 'Columns 1 through 7' and 'Columns 8 through 14'. Each section contains 14 rows of data, with each row having 14 numerical values (10.0000). The editor shows line numbers from 1 to 28. The status bar at the bottom indicates 'length: 844688 lines: 14399'.

A través de la consola se procede a la apertura del archivo, enumerando en pantalla las funciones (val(x:x:y)) y la cantidad de columnas de la observación.

The screenshot shows a C++ IDE with a project named 'Proyecto 2'. The code in 'main.cpp' includes `<iostream>`, `<fstream>`, and `<string.h>`, and uses the `std` namespace. The `main` function opens 'example.txt' and processes it. The console output window on the right shows the execution results, including function numbers and column counts. A red box highlights the first two lines of the console output.

```

#include <iostream>
#include <fstream>
#include <string.h>
using namespace std;

int main(int argc, char *argv)
{
    string datos="";
    ifstream myfile;
    string line="hola", c="";
    myfile.open ("example.txt");

    if (myfile.is_open()) {
        int principio = myfile.tellg();
        myfile.seekg(0, ios::end);
        int final = myfile.tellg();
        myfile.seekg(0, ios::beg);

        cout << "Linea 7345 Funcion numero 103 val(:,:103)" << endl;
        cout << "Linea 7347 Datos: Columnas 1 Profundidad through 7" << endl;
        // ... (other console output lines) ...
    }
}

```

## 2.4 Visión inicial

### ¿Porque la aplicación debería ser web?

Para que pueda ser accedida desde diversos dispositivos <sup>[1]</sup> conectados a Internet. En este caso no sería necesario tener en cuenta la instalación de un programa, tampoco el sistema operativo <sup>[2]</sup> ni la arquitectura <sup>[3]</sup> del computador. Bastaría con tener un navegador web <sup>[4]</sup>.

Por ejemplo para html 5 <sup>[5]</sup>, desde un explorador como Firefox 3.6 en adelante, Internet Explorer 9.0 o Google Chrome se tendría soporte para canvas <sup>[6]</sup> de HTML5 que es una herramienta muy versátil para diseños gráficos.

El motivo para hacer una aplicación web es porque sería más fácil compartir la información generada por la aplicación y además porque puede ser accedida desde cualquier dispositivo sin necesidad de ser una PC. Por ejemplo, un Ingeniero Agrónomo podría consultar la información desde un celular con conexión a Internet en una ubicación remota en un campo. Por otra parte, el hecho de tener un servidor puede mejorar la performance de cálculos matemáticos en el caso de estar generando cálculos complejos con una máquina con pocas capacidades de procesamiento <sup>[7]</sup> y almacenamiento <sup>[8]</sup>.

En este caso el servidor <sup>[9]</sup> se encargaría de realizar toda la parte pesada de los cálculos. Además de esta forma, por ejemplo el profesor, luego de generar un video, podría persistir <sup>[10]</sup> toda la información para que los alumnos la pudieran ver desde las demás máquinas.

---

1Dispositivo, desde el punto de vista de la computación; el término puede variar dependiendo del contexto empleado. En este caso se interpreta por dispositivos, a celulares, tablets, iPhones, computadoras portátiles.

2 Véase Glosario, Sistema Operativo.

3 Véase Glosario, Arquitectura del computador.

4 Navegador Web es la aplicación que se encarga de presentar visualmente un sitio de Internet para que el usuario final realice acciones de navegación. Navegadores como Google Chrome y Firefox son ejemplos.

5 HTML 5 es una de las tecnologías investigadas; ver el siguiente artículo: **"HTML5"**

6 Ver artículo **"Canvas de HTML 5"**

7 Véase Glosario, Procesamiento

8 Véase Glosario, Almacenamiento

9 Véase Glosario, Servidor

10 Persistir es un término comúnmente utilizado en la jerga de programación y base de datos. Como su palabra lo sugiere, significa guardar cualquier tipo de datos en un medio que asegure su persistencia en el tiempo.

## HTML 5

Nuevo estándar que busca reemplazar los anteriores de HTML para hacer la web mucho más interactiva para generar mejores aplicaciones que logren el nivel de interactividad de las de escritorio.

La tecnología HTML5 está en pañales pero es una tecnología del futuro que no va a desaparecer sino que va a seguir evolucionando. El día de mañana habrá muchas más aplicaciones web en sustitución a las aplicaciones de escritorio; incluso la PC como dispositivo tenderá a desaparecer.

Como muestra de la capacidad de lo que se puede hacer entre html5 y javascript aquí se muestran unos ejemplos de juegos hechos con HTML 5

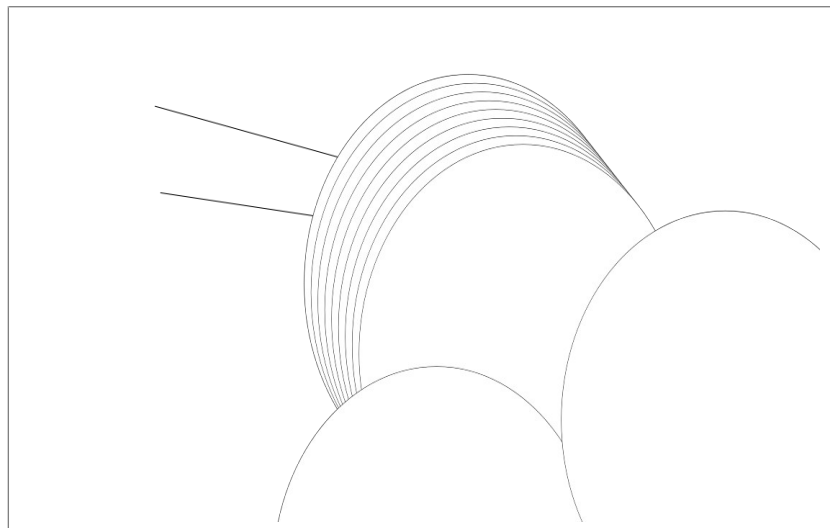
<http://www.isogenicengine.com/2011/09/04/isociety-limited-functionality-demo-released/>

<http://glacialflame.com/>

### Canvas de HTML 5

Etiqueta introducida por HTML 5 para generar gráficos en una página web y también manejo de animaciones. Esto se podría utilizar para generar las curvas de nivel, los puntos, las líneas sobre el plano. Hay una gran documentación sobre esta tecnología en la web.

La siguiente imagen fue generada con el editor gráfico Canvas Draw





## Manejo de animaciones con HTML5 y canvas

### POPCORN.JS

Permite sincronizar eventos <sup>[1]</sup> de varios elementos html y controlar la ejecución de un video. En este proyecto podría ser utilizada para mantener control sobre los videos: reproducir, pausar, rebobinar y detener animaciones.

Con estas herramientas se tiene la posibilidad de convertir un canvas en video, manejar formatos avi, entre otras posibilidades multimedia <sup>[2]</sup>. Incluso la tecnología orientada a la web ha incursionado en api's <sup>[3]</sup> para el manejo de audio y video con HTML5.

Para conocer mas sobre estas tecnologías y ver demostraciones, consulte las referencias 28 a 34 del capítulo "Referencias" de este documento.

---

1 Evento: en una página web se llama evento a una interacción del usuario en la página, ejemplo, clickear sobre un botón: también puede haber eventos que no involucren al usuario, como la actualización de la página, o parte de ésta, cada determinado tiempo.

2 Contenido Multimedia: refiere a la combinación de contenidos como videos, sonidos imágenes, textos, interactividad entre el usuario y la página.

3 Abreviación en Inglés de Interfaz de Programación de Aplicaciones. Son el conjunto de funcionalidades que brinda un programa para el usuario final o para el desarrollador de aplicaciones

## 2.5 Programación Orientada a Objetos (OO)

Se trata de uno de los paradigmas de la programación. La idea es crear un tipo de dato lo más parecido posible a la realidad que se intenta modelar.

Ejemplo: Una persona, en su vida real posee un nombre, un apellido, una edad conocida entre otra infinidad de detalles. Gracias a la programación OO se logra representar, *en código*, a la persona y los atributos más relevantes para el diseño del programa. Según esto una persona podría ser representada de la siguiente manera:

```
/* Ejemplo en Java*/
public class Persona{
//Declaración de los atributos de la persona
private string nombre;
private string apellido;
private date fechaNacimiento;
}
```

Características principales de la Programación Orientada a Objetos:

Polimorfismo, es la capacidad de asociar diferentes métodos a la misma operación.

Ejemplo:

```
Public class Persona{
private string nombre = "Mathias";

public void getNombre() //Operación getNombre()
    { //El código escrito entre llaves representa al método de la operación.
      return this.nombre;
    }
}

public class Mascota{

private string nombre = "Charlie";
public void getNombre() //Operación getNombre()
    { //El código escrito entre llaves representa al método de la operación.
      //Vemos que los métodos de ambos difieren.
    }
    string apodo;
    apodo = "Mi mascota se llama" + this.nombre;
    return apodo;
}
}
```

Herencia, refiere a la posibilidad que tiene un Objeto de heredar de otro Objeto.

Ejemplo: En la naturaleza, más precisamente en el reino animal, sabemos que un reptil es una especie vertebrada, por lo tanto un cocodrilo además de ser reptil es un vertebrado y hereda todas sus propiedades.

En la Programación Orientada a Objetos pasa algo similar: la clase Reptil tiene su conjunto de atributos, declarados de igual manera a la vista en el código del Polimorfismo, y la clase Vertebrado también tiene sus propiedades particulares. Pues bien, gracias a la herencia, la clase Reptil puede heredar también las propiedades de Vertebrado.

Ejemplo en Java:

```
public class Vertebrado{
private int cantidadDeVertebras;

//Método para colocar la cantidad de vértebras del animal
public virtual void setearCantidad()
{
    //Se deja el método vacío, para que sea implementado en una clase hijo
}
}

public class Reptil Inherits Vertebrado{
//Método para colocar la cantidad de vértebras del cocodrilo
public void setearCantidad()
{
this.cantidadDeVertebras = 70;
return;
}
}
```

### 3 Glosario

---

**Algoritmo:** En informática. Procedimiento lineal que dada las condiciones iniciales, las evalúa y determina un resultado.

**Almacenamiento:** es el concepto empleado en informática que refiere a la acción de guardar datos en un medio físico. El disco duro es un medio físico destinado a guardar los programas del sistema operativo y los archivos del usuario por lo cual constituye un medio de almacenamiento. La memoria principal del sistema es otro elemento que almacena temporalmente archivos o programas que en el momento está ejecutando el usuario. También reserva espacio para que la Unidad Central de Procesamiento (CPU) pueda leer o escribir una nueva instrucción.

Para la CPU el almacenamiento es una limitante, ya que al leer instrucciones tiene que acceder a la memoria principal u otra que haga de intermediaria entre éste y la memoria principal. Intuitivamente se deduce que cuanto más tiempo demore el CPU en acceder, menor será la cantidad de procesamiento del computador, aumentando el tiempo de espera del usuario final. La cantidad de memoria disponible también afecta la productividad del computador.

**ANSI:** Del Inglés, American Nacional Estandar Institute, es una institución dedicada a la estandarización en diversas áreas.

**API:** Abreviación en Inglés de: Interfaz de Programación de Aplicaciones. Se trata del conjunto de funcionalidades que brinda un programa para el usuario final o para el desarrollador de aplicaciones.

**Arquitectura del computador:** Permite estudiar cómo está diseñado, cuáles son los elementos que lo componen y cómo se comunica interna y externamente el computador en temas tales como:

Qué dispositivos periféricos (mouse, teclado, impresora) soporta, cuál es la forma de conexión de estos dispositivos a la computadora.

Análisis de la comunicación electrónica interna del computador.

Estudio de esta lógica de comunicación.

Son específicos de la arquitectura de la computación.

**B:** Es un lenguaje de programación diseñado por Dennis Ritchie y Ken Thompson en los Laboratorios Bell, principalmente usado para aplicaciones no numéricas y programación de sistemas. Apareció por primera vez en 1969. Su sintaxis es bastante similar a C.

Ejemplo de Hola Mundo en B:

```
main( ) {  
    extrn a, b, c;  
    putchar(a); putchar(b); putchar(c); putchar('!\n');  
}
```

a 'hell';  
b 'o, w';  
c 'orld';

**BCPL, Basic Combined Programming Language:** Desarrollado en la Universidad de Cambridge por Martin Richards. Es un lenguaje anterior a C del cual éste rescata características como la existencia de constantes, variables, caracteres, cadenas alfanuméricas, uso de funciones.

La diferencia más llamativa entre BCPL y C es que el primero es un lenguaje sin declaración de tipos. Al momento de crear una variable no es necesario indicar si es un valor decimal o un carácter; eso se resuelve al momento de la comparación o utilización de la variable.

Ejemplo de Hola Mundo! En BCPL

```
GET "LIBHDR"
```

```
LET START () BE  
$(  
WRITES ("Hello, world!*N")  
$)
```

Explicación del lenguaje extraído de:

[http://www.zator.com/Cpp/E0\\_lw2.htm](http://www.zator.com/Cpp/E0_lw2.htm)

Ejemplo de “hola mundo” extraído de:

<http://www.helloworldexample.net>

**Captación, hidrología:** Se refiere en la hidrología a la acción de recoger las aguas de acuíferos o manantiales.

**Cliente / Servidor:** La arquitectura cliente servidor consta de una o varias computadoras que ofrecen un servicio determinado, como por ejemplo acceso a archivos compartidos o simplemente a una página web; estas computadoras son llamados Servidores y se encargan de brindar los recursos que tengan disponibles a los dispositivos que lo soliciten, los cuales son conocidos como Clientes.

**Código Máquina:** Es el lenguaje básico de toda computadora actual. Consiste en una secuencia de bits, 0 y 1, ordenados dentro de un arreglo de caracteres de tamaño x. El microprocesador lee las sentencias de cada arreglo para ejecutar la siguiente acción del programa.

**CSV:** Comma-Separated Values, es un formato tabular estándar para escribir datos en archivos, el cual se basa en la existencia de filas con campos separados por el carácter coma ( , ); estos campos suelen contener sus datos dentro de comillas o comillas simples.

**Dispositivo:** Desde el punto de vista de la computación el término puede variar dependiendo del contexto empleado. En este caso se interpreta por dispositivos a celulares, tablets, iPhones, computadoras portátiles.

**Drag and Drop:** Término empleado en el diseño de interfaz de usuario para denominar la acción de arrastrado y posterior soltado de un elemento.

**DSL, Domain Solution Language:** Término que aplica a soluciones para un problema determinado en un lenguaje dado. La solución puede ser la creación de una librería o herramienta.

**Entorno:** Es donde la aplicación se desarrolla; entre los más importantes se encuentran el entorno orientado a web y el de escritorio. El primero está orientado a operar en un sistema distribuido en donde el intercambio de información se logra a través de una red de comunicaciones; el segundo está orientado a operar en un sistema monolítico, donde trabaje bajo una terminal sin necesidad de comunicarse con otros.

**Espesor, hidrología:** Densidad de un fluido.

**Evento:** En una página web se llama evento a una interacción del usuario en la página, por ejemplo clickear sobre un botón. También puede haber eventos que no involucren al usuario, como la actualización de la página, o parte de esta, cada determinado tiempo.

**Formación geológica, hidrología:** Capa de rocas con características comunes y que se diferencian de las adyacentes.

**Framework:** Es un término Inglés que en el ámbito informático significa 'Entorno de desarrollo'. Los frameworks son herramientas que alientan al programador a preocuparse más por la codificación funcional del programa y no tanto por configurar el programa, lenguaje y diferentes variables necesarias para el correcto funcionamiento en cada sistema operativo.

**GNU:** es un término que por extensión, es asociado al Software libre.

**Gradiente, hidrología:** variación de un elemento, siguiendo una dirección determinada.

**Hard – Code:** Término de la jerga informática, refiere a una mala práctica en el desarrollo del software consistente en introducir los datos directamente en el código fuente, no mediante obtención de parámetros u otro recurso externo.

**Herramientas colaborativas:** Tienen como principal característica facilitar el trabajo y organización de las personas a través de la red.

Entre lo más destacado está el hecho de poder crear un documento de tipo Office y compartirlo con determinados usuarios, cada uno con determinados privilegios, los cuales podrían, en potencia, estar editándolo y viéndolo al mismo tiempo.

**Integrated Development Environment:** Ambiente de Desarrollo Integrado; los IDE'S proveen herramientas que facilitan la programación.

**Interpolación bilineal:** En tratamiento digital de imágenes, procedimiento de interpolación basado en promediar los valores de los 4 píxeles vecinos.

**JNI:** Refiere a la interfaz nativa del lenguaje Java cuya función es poder integrar código escrito en Java, con otros códigos escritos en otros lenguajes.

**Lenguajes de programación:** Son los diferentes lenguajes en los cuales se codifica.

Ejemplo: Java, Perl, Delphi, Python, Ruby.

2.a El lenguaje interpretado refiere a que cada línea de código es analizada en el momento de su ejecución.

**Lentejón, hidrología:** En hidrología, pequeña masa de agua existente en estratos subterráneos.

**Librerías:** Conjunto de funcionalidades para un lenguaje de programación determinado. Las funcionalidades pueden ser matemáticas, de encriptado, conversión de formatos, etc. Cada lenguaje de programación tiene librerías por defecto, mientras que otras tienen que ser asociadas o instaladas.

**Máquina Virtual de Java:** Es una aplicación que se instala en la computadora; es multiplataforma y se encarga de interpretar el lenguaje compilado de Java que le llega en un formato binario particular. Una vez interpretada la instrucción procede a ejecutar lo que en la misma se indique. Es por esto que su existencia en la computadora es esencial, porque sin ella no habría ningún intérprete del lenguaje y por lo tanto no funcionaría el programa

**Multimedia, Contenido:** Refiere a la combinación de contenidos como videos, sonidos, imágenes, textos, interactividad entre el usuario y la página.

**Multiplataforma:** Capacidad de un sistema de software para operar igualmente en distintas plataformas.

**Navegador Web:** Es la aplicación que se encarga de presentar visualmente un sitio de Internet para que el usuario final realice acciones de navegación. Navegadores como Google Chrome y Firefox son ejemplos.

**Nivel topográfico, hidrología:** Desnivel existente entre puntos que se hallan a distintas alturas.

**ODF:** El formato Open Document Format fue diseñado para compartir información a través de diferentes aplicaciones de procesamiento de texto. Un archivo ODF puede tener diferentes extensiones, la cual determina el tipo de información que está almacenada. El tipo de archivo que interesa en nuestra aplicación es el ods (OpenDocument Spreadsheet).

**OpenOffice:** Conforman un conjunto de programas para procesar texto, hojas de cálculo, gráficas, presentaciones y bases de datos. Está disponible en varios lenguajes y es multiplataforma. Es código abierto y está disponible en su web oficial para ser descargado.

**OpenSource:** Es un movimiento existente desde principios del 1990, cuyos principios éticos se basan en la necesidad de mantener el código abierto y a disposición de la comunidad, para que ésta modifique o mejore los programas. También provee un marco jurídico, con existencia de licencias inclusive, para que los desarrolladores OpenSource tengan un amparo sobre sus programas.

**Portabilidad:** Facilidad de un programa de ser ejecutado en diferentes ambientes, con distintos sistemas operativos.

**Presión atmosférica, hidrología:** En su concepción general la atmósfera es la capa de aire que cubre la tierra, en tanto que la presión atmosférica es la presión ejercida por la atmósfera sobre todos los objetos que se encuentran inmersos en ella.

**Presión hidrostática, hidrología:** Es la presión de un fluido sobre las paredes, y el fondo que lo contiene y sobre la superficie de los objetos sumergidos.

**Programación:** Es el arte y la técnica de escribir en un lenguaje determinado, un programa funcional, que cumpla algún fin previamente estipulado.

**Procesamiento:** Es una función fundamental de cada computador. Una unidad central de procesamiento (CPU) es la encargada de ejecutar, coordinar y ordenar las múltiples tareas de éste. El CPU está ejecutando siempre una tarea a la vez, por esto existen diversas técnicas de ejecución de tareas, dependiendo de una limitante: la arquitectura del computador.

**Sistema Operativo:** Se trata de un conjunto de programas ejecutados en el PC, cuyos rasgos más importantes a tener en cuenta por el usuario final es que permiten gestionar el funcionamiento del computador y ejecutar programas de utilidad para el usuario, como el explorador, o alguna herramienta de escritorio, actuando como una interfaz entre el usuario y la arquitectura del computador.

**Servidor:** Es un computador cuya funcionalidad es brindar servicios a otros computadores que se lo soliciten.

Ejemplo: Un servidor web brinda acceso a una página de Internet o a otros computadores que se lo soliciten, mediante una dirección en el navegador.

**Tessellation:** Se entiende por mosaico o patrón de varias figuras que completan el plano sin superponerse ni dejar espacios en blanco. Por extensión se llama mosaico a cualquier obra realizada con fracciones diversas.

**Viscosidad, hidrología:** Resistencia que ofrece un fluido al movimiento relativo de sus moléculas.

**Viscoso, hidrología:** Glutinoso.



**Script:** Es un programa de procesamiento por lotes. Usualmente son simples, se almacenan en un archivo de texto plano y son interpretados por un interprete.