

Sistema de interpretación de ensayos de bombeo en acuíferos

Estado del Arte

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:

1INTRODUCCIÓN.....	5
2CONCEPTOS GENERALES SOBRE LOS ACUÍFEROS.....	6
2.1AGUA SUBTERRÁNEA.....	6
2.2ACUÍFEROS	8
2.2.1Cuantificación de los recursos de un acuífero.....	9
2.2.2Ley de Darcy.....	10
2.3NIVEL PIEZOMÉTRICO	11
2.3.1Superficie piezométrica	12
2.4TIPOS DE ACUÍFEROS.....	13
2.5ENSAYOS DE BOMBEO.....	15
2.6ANÁLISIS DE LOS ENSAYOS DE BOMBEO.....	17
2.7METODO THEIS.....	18
2.8ALGORITMOS DE OPTIMIZACIÓN.....	22
3HERRAMIENTAS CIENTÍFICAS EXISTENTES.....	23
3.1MATLAB.....	23
3.1.1Descripción general.....	23
3.1.2Características.....	23
3.1.3Herramientas.....	24
3.1.4Exportación e Importación de archivos externos.....	24
3.1.5Desarrollo de la interfaz de usuario.....	24
3.1.6Gráficas.....	24
3.1.7Visualización en 3D.....	24
3.1.8Funciones matemáticas y análisis de datos.....	25
3.1.9Programación Orientada a objetos.....	25
3.1.10Ventajas y desventajas.....	25
3.1.10.1Ventajas.....	25
3.1.10.2Desventajas.....	25
4INVESTIGACIÓN DE TECNOLOGÍAS.....	26
4.1OBJETIVO.....	26
4.2PYTHON.....	26
4.2.1 Descripción.....	26
4.2.2Características [].....	26
4.2.3Herramientas: Librerías destacadas, herramientas.....	27
4.2.3.1NumPy.....	27
4.2.3.2Framework Qt – PyQt.....	27
4.2.3.3Matplotlib [].....	27
4.3JAVA.....	28
4.3.1Descripción General.....	28
4.3.2Características.....	28
4.3.3Herramientas: Librerías, Frameworks, Ámbito de desarrollo.....	29
4.3.3.1JRE.....	29
4.3.3.2OpenGL.....	29
4.3.3.3Java OpenGL.....	29
4.3.3.4Jzy3D.....	30
4.3.3.5Processing.....	30
4.3.4Soporte y Documentación.....	31
4.3.5Ejemplo de implementación combinando Jzy3D con Proccesing.....	32
4.3.6Ventajas y desventajas.....	34
4.3.6.1Ventajas.....	34
4.3.6.2Desventajas.....	34
4.3.7Referencias.....	34
4.4C/C++.....	35
4.4.1Descripción General.....	35
4.4.2Características.....	35

4.4.3Herramientas.....	36
4.4.3.1Compilador.....	36
4.4.3.2IDE BloodShed Dev C++.....	36
4.4.3.3IDE Code:Blocks.....	36
4.4.3.4Librerías estándares.....	38
4.4.3.4.1iostream.h.....	38
4.4.3.4.2String.h.....	38
4.4.3.4.3stdio.h.....	38
4.4.3.4.4stdlib.h.....	38
4.4.3.5Librería Matemática.....	38
4.4.3.5.1Math.h.....	38
4.4.3.6Librerías de flujo de datos.....	39
4.4.3.6.1ifstream, ofstream, fstream.....	39
4.4.3.7Interfaz de Usuario.....	40
4.4.3.7.1GTK+.....	40
4.4.3.7.2QT.....	40
4.4.4Ventajas y Desventajas.....	41
4.4.4.1Ventajas.....	41
4.4.4.2Desventajas.....	41
4.4.5Referencias.....	42
5CODIFICADORES DE VIDEO.....	43
5.1FFMPEG.....	43
5.1.1Descripción general.....	43
5.1.2Características.....	43
5.1.3Bibliografía.....	44
5.2MENCODER.....	44
5.2.1Descripción General	44
5.2.2Características.....	44
5.2.3Bibliografía.....	44
5.3LIBRERÍAS LIBAVCODEC Y LIBAVFORMAT.....	45
5.3.1Descripción General.....	45
5.3.2Características.....	45
5.3.3Bibliografía.....	45

1 Introducción

En el marco de la investigación del proyecto, es fundamental determinar que avances existen en el tema específico que se va a tratar, esto permitirá establecer un punto de partida desde la investigación y también más adelante visualizar con claridad el aporte realizado por el proyecto en contraste con aplicaciones y herramientas ya existentes.

En este documento en primera instancia se va a tratar sobre la introducción a los conceptos generales sobre los acuíferos, luego se analizará la herramienta científica existente sobre la cual hasta el momento se procesan los cálculos y se genera el graficado de los datos. A continuación se realiza un análisis de diferentes tecnologías para más tarde determinar, mediante una comparativa, cual de ellas es la más óptima de acuerdo a los requisitos.

2 Conceptos generales sobre los acuíferos

2.1 Agua subterránea

El agua subterránea representa una fracción importante de la masa de agua presente en cada uno de los continentes. Las masas de aguas subterráneas se forman a través de gran parte del agua de lluvia que se infiltra en el terreno alcanzando las capas profundas.

A una gran profundidad por debajo de la superficie de la tierra existen formaciones geológicas impermeables que por su constitución forman una barrera para el agua impidiendo que esta fluya, de forma que encima de ellas exista suficiente agua, la cual se acumulará saturando todos los pozos existentes en las mencionadas barreras. Esta acumulación dará lugar a lo que se conoce como *zona saturada* la cual será alojada en los acuíferos. Por encima de esta zona, nos encontramos con la *zona no saturada* la cual contiene a la *zona reticular* donde las raíces de las plantas toman el agua necesaria para su ciclo vegetativo. Se denomina nivel freático a la línea de separación entre la zona saturada y la zona superior no saturada.

El volumen del agua subterránea, si bien es menor al de los mayores glaciares, es de gran importancia en comparación con el resto de las masas de aguas, como por ejemplo la retenida en lagos o el agua circulante por ríos. Las masas más extensas pueden alcanzar millones de km². El agua del subsuelo es un recurso importante, y de este se abastece gran parte de la población mundial, pero de difícil gestión, por su sensibilidad a la contaminación y a la sobre explotación.

Los acuíferos son un tipo de formación geológica, aunque existen otros tipos. Las unidades geológicas pueden ser clasificadas de acuerdo a su relación con las aguas subterráneas en los siguientes cuatro tipos:

- **Acuífero:** es una formación geológica capaz de almacenar agua y transportarla, por lo que se dice que tiene permeabilidad alta. Sobre esta formación se tratará a lo largo del presente capítulo.
- **Acuitardo:** es la formación geológica que contiene agua pero apenas la transmite, lo que significa que tiene permeabilidad baja, por lo cual no es apto para el emplazamiento de captaciones^[1] de agua subterráneas. Está constituido por arcillas y arenas.
- **Acuicludo:** se define como la unidad geológica que contiene agua en su interior, incluso hasta su saturación, pero no la transmite, por lo cual no es apto para su explotación y se dice que tiene permeabilidad muy baja. Un ejemplo de esto son los depósitos de arcillas.
- **Acuífugo:** este tipo de formación no almacena agua en su interior dejando que esta fluya, por lo que tampoco es apto para su explotación hidrológica. Pueden estar localizados en un macizo granítico que no esté fragmentado.

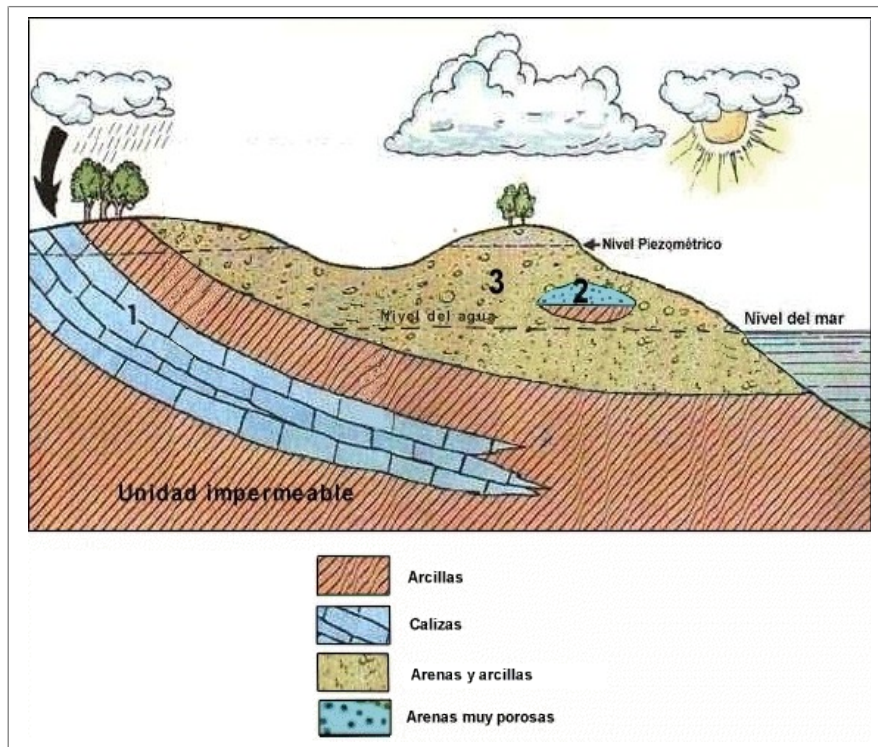
1 Consultar glosario, definición de captación.

La anterior clasificación se ilustra en la siguiente tabla.

	Captación	Transmite	Almacena
Acuífero	SI	SI	SI
Acuitardo	-	Poco	SI
Acuícludo	NO	NO	SI
Acuífugo	NO	NO	NO

2.2 Acuíferos

Un acuífero es aquel estrato o formación geológica^[1] permeable que permite la circulación y el almacenamiento del agua subterránea por sus poros o grietas. Dentro de estas formaciones es posible encontrar materiales muy variados como gravas de río, limo, calizas muy agrietadas, areniscas porosas poco cementadas, arenas de playa, algunas formaciones volcánicas, depósitos de dunas e incluso ciertos tipos de arcilla.



Como háse mencionado en el capítulo anterior, un acuífero se divide en varios niveles, dentro de ellos los más señalados para la consideración son:

Nivel freático: En el caso de un acuífero libre^[2], se denomina nivel freático o tabla de agua al nivel real del agua subterránea.

Nivel piezómetro: Nivel ideal que alcanzaría el agua a presión atmosférica^[3].

El agua de un acuífero está contenida en sus poros, entonces cuando ésta es extraída se produce una disminución de su volumen, fenómeno que se conoce como subsidencia, el cual es responsable del hundimiento lento de áreas de la superficie de la tierra. Los fenómenos relacionados con los acuíferos que más le interesan a la Hidrología Subterránea son la recarga y la descarga de los mismos.

La *recarga* natural se produce cuando el agua de lluvia se infiltra en el terreno y pasa a los acuíferos. Esta es la fuente más habitual de recarga y formación de acuíferos pero también la más variable. También es habitual la recarga desde el río

1 Consultar Glosario, definición de Formación geológica.

2 Definición en el punto 1.4.

3 Consultar Glosario, definición de Presión atmosférica..

hacia el acuífero. El acuífero se recargará mientras la capacidad de almacenamiento de éste no sea sobrepasada.

Una vez que el acuífero no pueda contener más agua, el sobrante saldrá a la superficie de la tierra, y fluirá hacia otra área como un río, lago, mar, otro acuífero o formará un manantial. Los manantiales se localizan en una discontinuidad del estrato impermeable, ladera, cauce de río, falla o diaclasa y recibirán nombres característicos según su ubicación: manantiales de ladera, de falla, de diaclasa, etc.

De la misma forma que la recarga de un acuífero se produce desde el río, también es habitual que suceda el proceso inverso en sentido contrario, o sea la *descarga* desde el acuífero hacia el río. Estos fenómenos se ven muy influidos por los niveles de agua en el río y en el acuífero y serán los que determinarán el sentido de flujo.

2.2.1 Cuantificación de los recursos de un acuífero

El volumen de agua contenido en un acuífero se mide en función del área y de la porosidad media del mismo, lo cual puede expresarse mediante la siguiente ecuación:

$$V=A*b*m$$

donde V es el volumen total,
A es el área del acuífero,
b es el espesor^[1] medio del acuífero,
y m es la porosidad media.

En algunos casos como el de los acuíferos no confinados^[2], sólo una parte del total de agua contenida es extraída, quedando el resto atrapado dentro de éste. Por esto se define la propiedad del *caudal específico útil* del acuífero como el volumen extraído por unidad de superficie para un metro de descenso del nivel freático. Este valor depende de la porosidad de la roca y también de las características de retención del agua y aunque puede ser muy variable, se encontrará dentro del rango de 0,05 y 0,40.

1 Consultar glosario.

2 Definición de tipos de acuífero en el punto 1,4.

2.2.2 Ley de Darcy

Darcy en 1856 desarrolló su propia ley para cuantificar el caudal de agua filtrado a través de arenas de la siguiente manera:

$$Q = K * A * h / L$$

donde,

Q es el caudal filtrado medido en metros cúbicos por segundo,

A es el área de la sección,

h es el incremento de presión entre los puntos extremos de una capa filtrante,

L es la distancia entre dichos puntos y

K es el coeficiente de permeabilidad.

Siendo $i = \frac{h}{L}$ el gradiente^[1] hidráulico, se tiene que $Q = A * K * i$.

El valor de K depende de la viscosidad del fluido^[2], de la gravedad g y de la densidad del fluido. En la siguiente se muestra una tabla de valores medidos de K para diferentes tipos de rocas.

Roca	K (cm/s)
Arenisca	10^{-4}
Marga	10^{-7}
Arena	10^{-2}
Grava	10^{-1}
Limo	10^{-5}
Arcilla	10^{-8}
Caliza	10^{-4}
Esquisto	10^{-4}

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

2 Viscosidad: Resistencia que ofrece un fluido al movimiento de sus moléculas.

2.3 Nivel piezométrico

El nivel piezométrico es la variable de estado en temas de hidráulica, así como lo es la temperatura para temas de calor. Es la energía por unidad de peso del agua y se mide en metros respecto al nivel de referencia (mar). En palabras simples, en el caso de un ensayo de bombeo, sería el nivel de agua en los pozos.

Este nivel (h) queda determinado la siguiente ecuación de Bernoulli:

$$h = z + \frac{p}{\gamma} + \frac{v^2}{2g}$$

donde,

Z es el nivel de referencia,

P la presión de succión ejercida por las partículas del suelo,

γ el peso específico del agua (que se puede considerar constante) y

V la velocidad del agua subterránea.

V puede ir desde $1 \frac{mm}{siglo}$ a $10 \frac{m}{dia}$ por lo cual es demostrable que el cociente

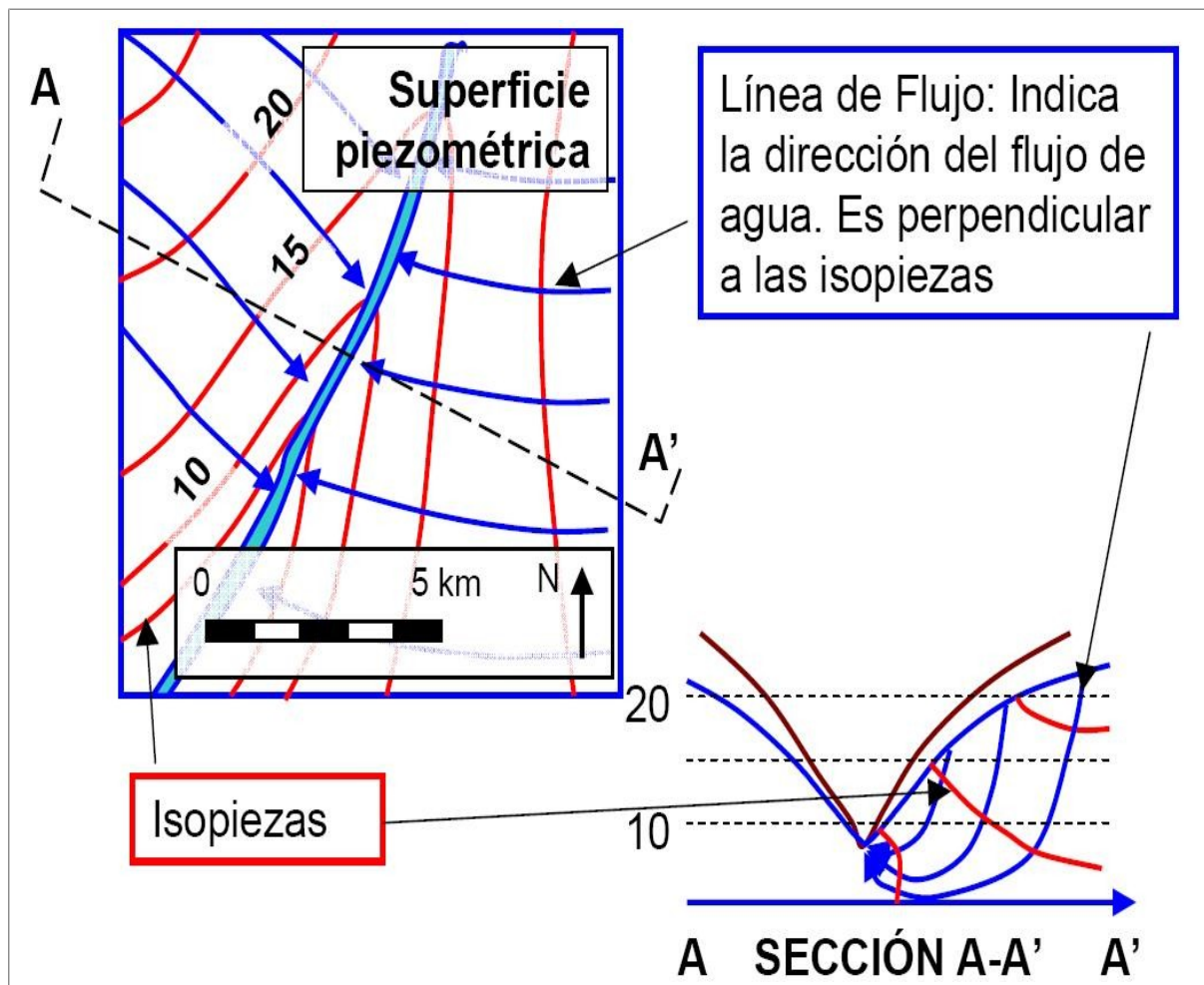
$\frac{v^2}{2g}$ es muy pequeño y por lo tanto despreciable en esta expresión, de lo que se obtiene la siguiente ecuación:

$$h = z + \frac{p}{\gamma}$$

2.3.1 Superficie piezométrica

Es la representación de la distribución espacial de los niveles. En acuíferos libres coincide con la freática.

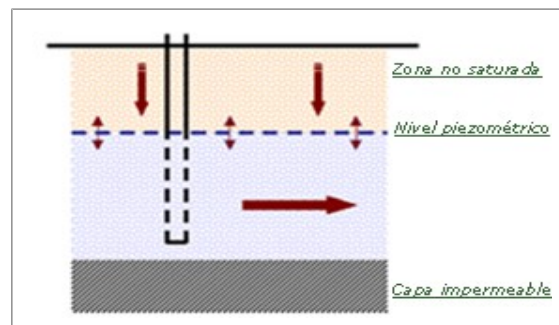
Se le llaman Isopiezas a las curvas de igual nivel piezométrico. En nuestro proyecto serán las curvas de nivel que deberemos mostrar en las gráficas 2D que reflejarán los niveles de descenso en todo el dominio.



2.4 Tipos de acuíferos

Los acuíferos pueden recibir diferentes clasificaciones de acuerdo al factor que se tenga en cuenta. Una de sus clasificaciones se basa en los materiales litológicos que los constituyan como: detríticos, fisurados, volcánicos, etc. Otra caracterización es la que se muestra a continuación, para la cual se tiene en cuenta el factor de la presión hidrostática^[1] del agua encerrada en los mismos.

Acuíferos libres: son aquellos que contienen una superficie libre del agua encerrada en ellos la cual se encuentra a presión atmosférica. El nivel freático en este caso está en la superficie del agua la cual puede estar en contacto directo con el aire, pero por encima de ésta no habrá ningún material impermeable que impida la circulación del agua. En el momento que se realicen perforaciones en el terreno que atraviese un acuífero de este tipo, el agua alcanzará el mismo nivel que posee dentro de esta unidad geológica, por lo cual el nivel freático coincidirá con el nivel piezométrico.

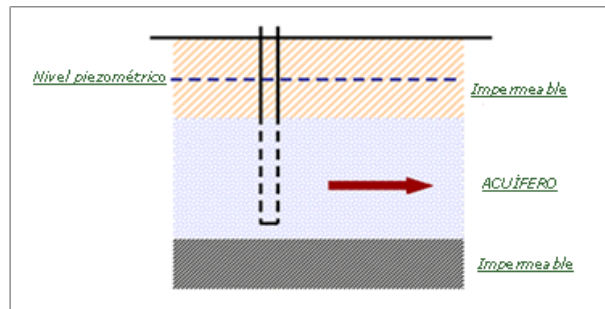


Acuíferos confinados: el agua contenida dentro de un acuífero de este tipo está sometida a una presión superior a la atmosférica y ocupa en su totalidad los huecos de esta unidad geológica, saturándola por completo. A diferencia de los acuíferos libres, estos están sellados por materiales impermeables que impiden que el nivel del agua ascienda hasta equipararse su presión con la atmosférica.

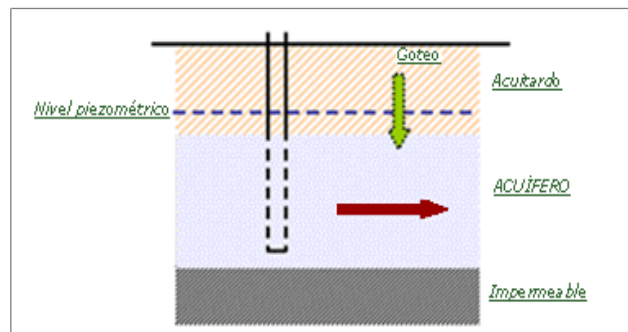
Es por esto que al hacer perforaciones de pozos en el terreno que atraviesen el techo del acuífero, se producirá que el nivel del agua ascenderá a una velocidad muy rápida hasta estabilizarse en el nivel piezométrico. Podrán darse pozos surgentes si el nivel del agua queda por encima del nivel topográfico^[2] y pozos artesianos si el nivel se estabiliza por debajo de la cota del terreno. De esta manera, si se perfora una sucesión de pozos atravesando un acuífero de este tipo y luego se unen los niveles observados en cada pozo, se obtendrá una superficie piezométrica que no coincidirá con el nivel freático que tenía el acuífero en estado natural.

1 Es la presión de un fluido sobre las paredes, y el fondo que lo contiene y sobre la superficie de los objetos sumergidos.

2 Desnivel existente entre puntos que se hallan a distintas alturas.



Acuíferos semiconfinados: Un acuífero de este tipo es un caso especial de los confinados, ya que está sellado por materiales semi-impermeables, que constituyen un acuitardo, es decir, que permiten un flujo lento de agua hacia el acuífero principal.



Acuíferos colgados:

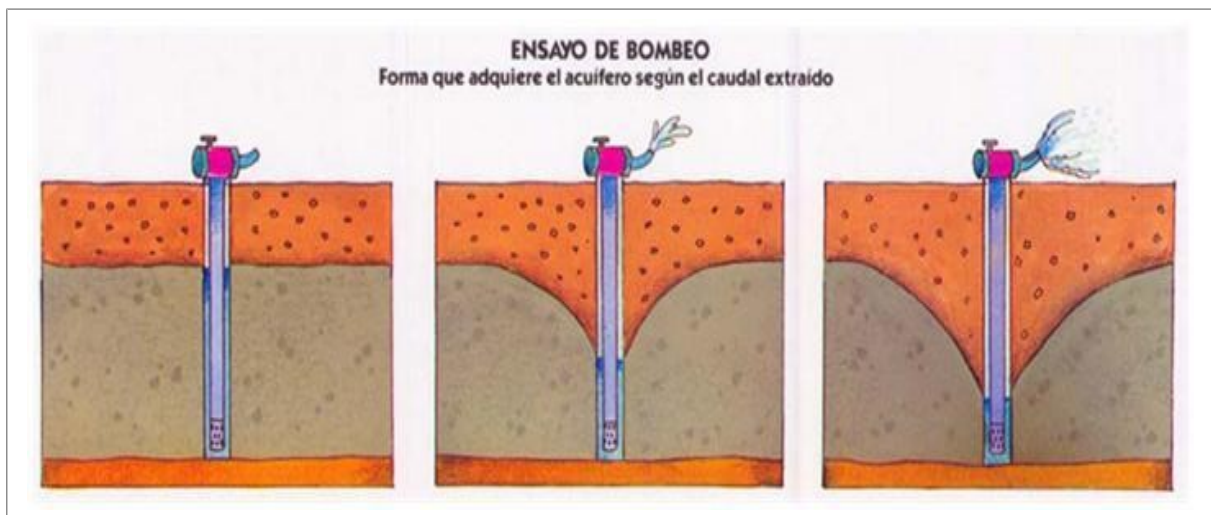
Son acuíferos que ocupan pocos centímetros de espesor que se forman de manera ocasional dentro de una capa con impermeabilidad relativa sobre el nivel freático de un acuífero. El agua que se infiltra y queda contenida dentro de esta capa forma un lentejón^[1] de una extensión reducida sobre la zona saturada más próxima. No determinan un recurso fiable, ya que en algunas ocasiones la formación geológica puede ser perforada completamente y el pozo construido facilitaría el drenaje del agua contenida en el lentejón hacia la zona saturada.

1 Lentejón: Pequeña masa de agua.

2.5 Ensayos de bombeo

Un ensayo de bombeo se realiza para evaluar un acuífero y verificar de forma práctica sus características, estimulándolo por medio de bombeo, y observando los descensos de nivel en pozos de observación. Un ensayo de bombeo es una herramienta común utilizada en hidrogeología para caracterizar un sistema de acuíferos.

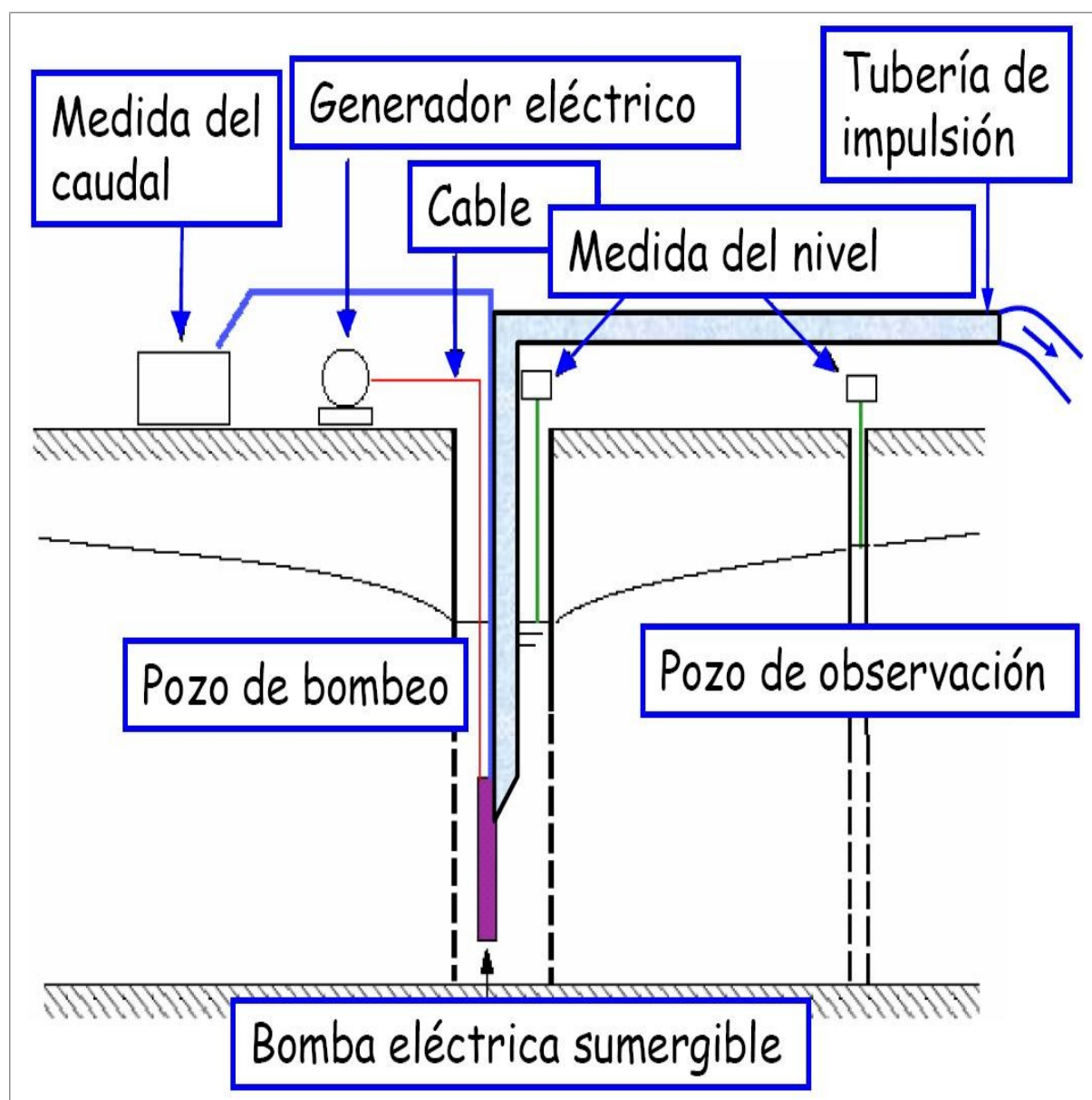
Un ensayo de bombeo consiste en extraer durante cierto tiempo, dos días por ejemplo, un caudal del acuífero a una tasa constante, a través del pozo, mientras se miden cuidadosamente los niveles de agua en los pozos de observación. Cuando el agua es bombeada desde el pozo de bombeo, la presión en el acuífero disminuye, provocando el descenso del nivel de agua en los pozos de observación. El descenso disminuye radialmente desde el pozo de bombeo, y aumenta a medida que el bombeo continúa. Se controla como varía el nivel del agua en el pozo a lo largo de este tiempo.



Las características del acuífero evaluadas más comúnmente a través de ensayos de bombeo son:

- Conductividad hidráulica
- Almacenamiento Específico
- Transmisividad

En la imagen a continuación se ilustran los elementos que participan en un ensayo de bombeo. Al bombear en un pozo desciende el nivel piezométrico. Básicamente la Transmisividad y el Coeficiente de almacenamiento se infieren a partir de la velocidad con la que se producen estos descensos.



2.6 Análisis de los ensayos de bombeo

Existen dos formas de interpretar los resultados de un ensayo de bombeo: utilizando un modelo analítico o uno numérico.

Modelos analíticos

- Son generalmente los más utilizados.
- El más fundamental de estos es el de Theis.
- Consisten en hacer coincidir los datos observados en el mundo real, asumiendo que los parámetros del modelo ideal son aplicables al acuífero del mundo real.

Métodos numéricos

- Son usados en casos más complejos.
- No aseguran mejores resultados que los analíticos, pero son menos restrictivos en cuanto a las hipótesis necesarias para su aplicación.

Casi todos los métodos analíticos de solución para ensayos de bombeo, están basadas en la solución de Theis, la cual está desarrollada considerando una serie de hipótesis que la hacen aplicable solo a casos simples. Existen otros métodos más flexibles, con hipótesis menos restrictivas que las consideradas por Theis, pero los mismos son más complejos de implementar. Los métodos de solución más comúnmente utilizados son:

- Solución transiente de Theis
- Solución de Cooper-Jacob
- Hantush-Jacob
- Neuman (para acuíferos libres)
- Boulton (para acuíferos anisotrópicos)

2.7 Metodo Theis

El método de Theis posee la siguiente ecuación

$$s = \frac{Q}{4\pi T} W(u)$$

donde, $u = \frac{r^2 S}{4Tt}$ $W(u) = -\gamma - \ln(u) - \sum_{n=1}^{\infty} \frac{(-1)^n u^n}{n \cdot n!}$

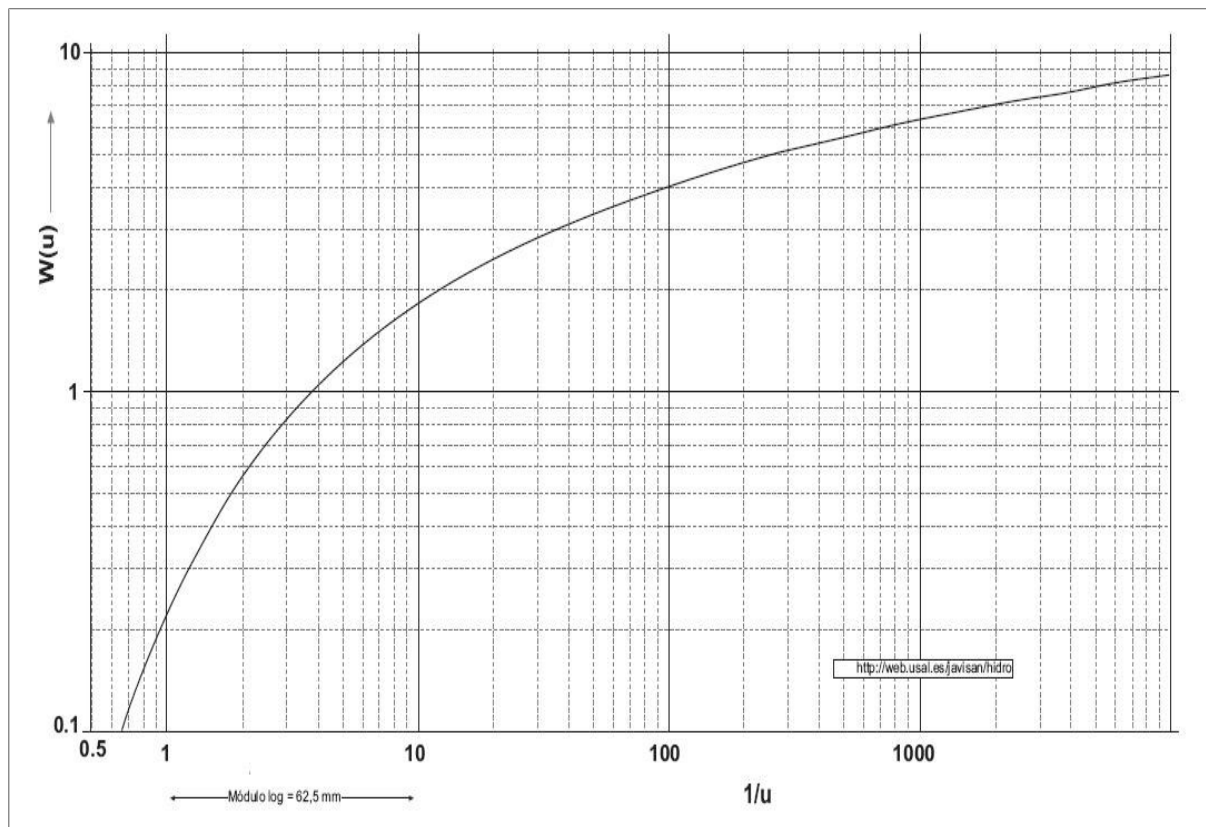
T es la Transmisividad,

S Coeficiente de almacenamiento,

r distancia radial,

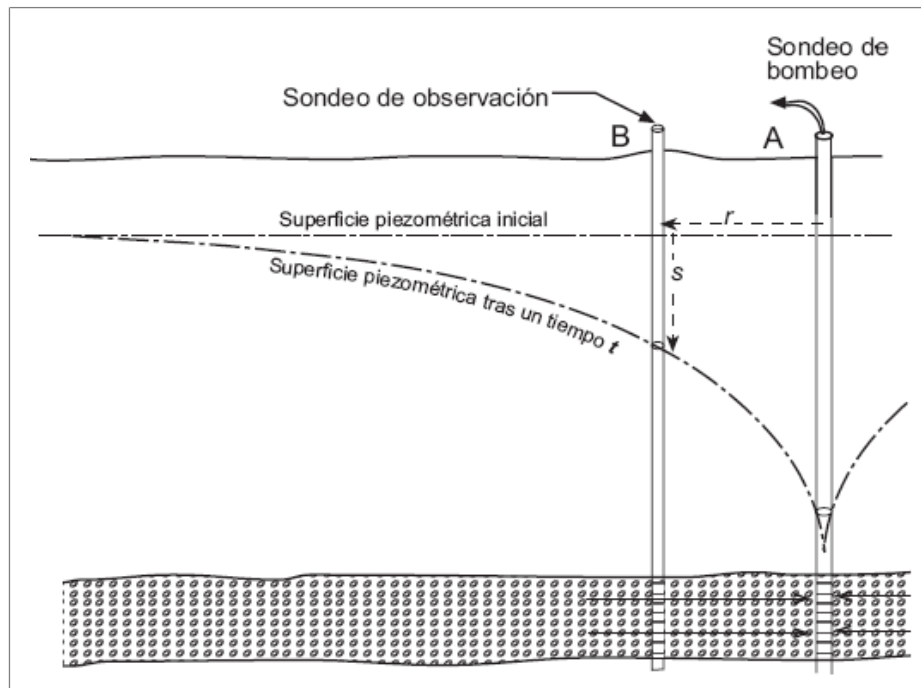
$\gamma = 0.577215664901532860$ es la constante de Euler.

El siguiente es el gráfico patrón de Theis:



A continuación se muestra una aplicación práctica del método de Theis en una interpretación de ensayos de bombeo, donde los cálculos se realizan de manera manual. Se parte de observaciones de campo para determinar los parámetros hidráulicos del acuífero.

Supongamos que se realiza un ensayo de bombeo en un acuífero confinado para medir sus parámetros hidráulicos. Supongamos también que el medio físico se ajusta al modelo teórico que se representa en la siguiente figura.



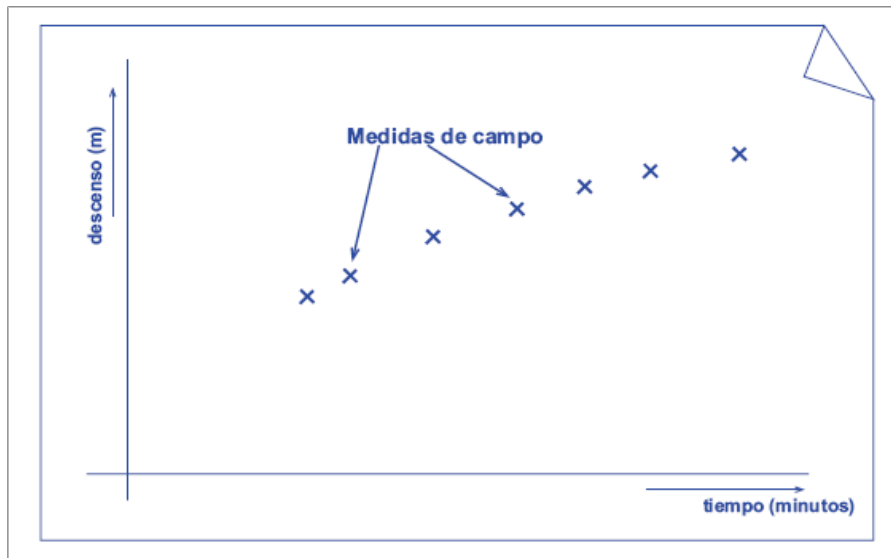
Son necesarios dos pozos abiertos en el mismo acuífero. En (A) se bombea un caudal constante, en el otro (B) se observan las variaciones del nivel para una serie de tiempos.

Supongamos que luego del ensayo se obtuvieron los siguientes datos. El caudal (Q) constante bombeado fue: 20 litros/seg. La distancia radial (r) entre el pozo de bombeo (A) y el de observación (B) es 150 metros.

Además se registraron los siguientes niveles de descenso para varios tiempos.

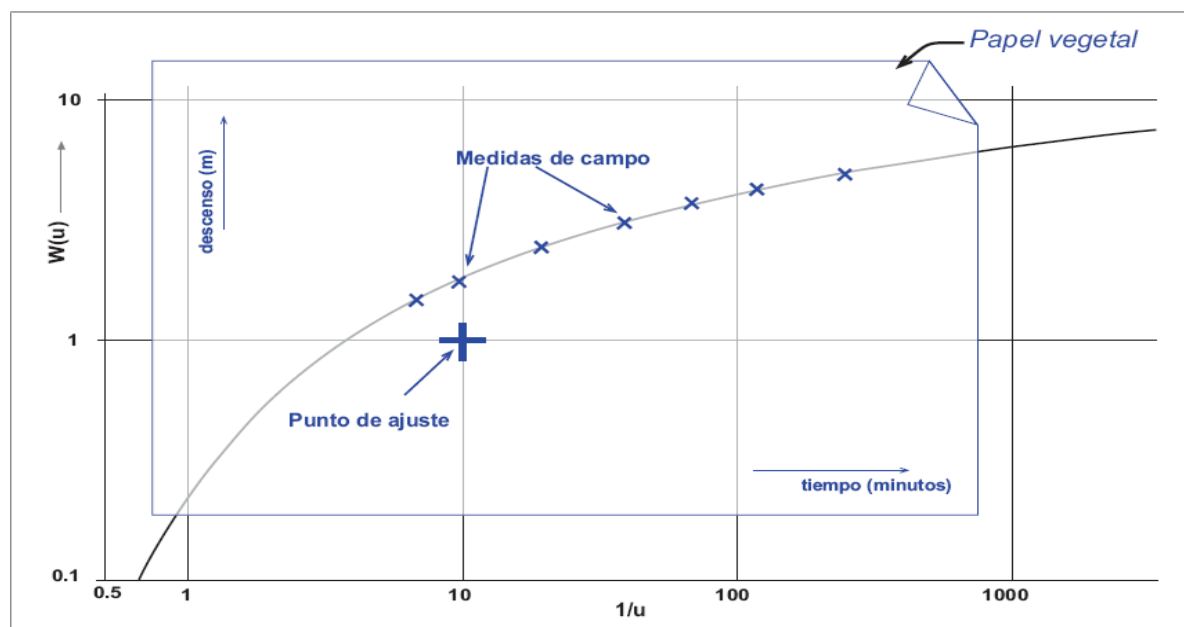
t (minutos)	s (metros)
7	1,80
10	2,15
20	3,00
40	3,80
70	4,60
120	5,25
250	6,05

A continuación se procede a representar las mediciones sobre un papel doble logarítmico, donde el tiempo va en el eje horizontal y los descensos en el eje vertical. Luego se colocan los datos sobre un papel. Estos datos también se representan sobre un papel vegetal representando los ejes, como se ve en la siguiente imagen.



Luego se superpone el papel vegetal sobre la gráfica patrón de Theis buscando la coincidencia de los puntos de las medidas de campo sobre la línea del gráfico patrón. El papel vegetal puede ser libremente desplazado sobre la gráfica patrón sin rotarlo, para buscar alinear los puntos de las mediciones con la curva del gráfico.

Lograda la superposición, se calca sobre el papel vegetal un punto cualquiera del gráfico patrón, que no tiene que pertenecer a la curva necesariamente. Este será un punto de ajuste. En el ejemplo de la figura se ha elegido el punto de coordenadas: $W(u)=1$ $1/u=10$, pero podría ser cualquier otro. Las coordenadas del punto de ajuste se deben anotar sobre el papel vegetal.



Luego se vuelve a superponer el papel vegetal sobre el papel logarítmico en el que se habían dibujado inicialmente los datos de campo, en la misma posición que cuando fueron calcados estos puntos. Se toman las coordenadas del punto de ajuste ahora en el gráfico de campo, que son las siguientes: tiempo=11,5 minutos y descenso=1,3 metros.

Por lo tanto, el punto de ajuste tiene dobles coordenadas en el gráfico de campo: tiempo, descenso y en el gráfico patrón: $1/u$, $W(u)$

Para el cálculo de la **Transmisividad**, utilizamos la ordenada del punto de ajuste en ambos gráficos, las cuales son $W(u)=1$ y descenso=1,3 metros. Sustituimos esta pareja de valores en la ecuación de Theis:

$$s = \frac{Q}{4\pi T} W(u) \quad \frac{20 * 86,4}{4\pi T} * 1$$

Como Q esta en litros/segundos, se multiplica por 86,4 para pasar a $m^3 / \text{día}$

Despejando T se obtiene: $T = 106 \text{ m}^2 / \text{día}$

Para el cálculo del **Coficiente de Almacenamiento**, ahora se utiliza la abscisa del punto en ambos gráficos: $1/u=0,1$, y tiempo=11,5 minutos. Utilizando estos valores, se deben reemplazar todas las variables de la ecuación de u y a continuación despejar la variable S.

$$u = \frac{r^2 S}{4 T t} \quad 0,1 = \frac{150^2 S}{4 * 106 * \frac{11,5}{1440}}$$

Se divide por 1440 para convertir minutos en días

Despejando obtenemos: $S = 1,5 * 10^{-5}$

2.8 Algoritmos de Optimización

Los valores de los parámetros de un modelo son raramente conocidos en su totalidad y en ocasiones cuando estos tienen poco sentido físico son desconocidos por completo. Es necesario entonces proceder a su optimización. En términos generales se trata de elegir la combinación para la cual la correlación entre los valores de las variables de salida generadas por el modelo y las medidas en el sistema real son coincidentes ^[1].

Los conceptos necesarios al momento de elaborar una optimización son:

Valores de decisión: son los valores óptimos que queremos determinar.

Funciones objetivo: establecen formas de valoración de los resultados.

Restricciones: Limitan los valores de las variables de decisión para que sean encontrados valores viables.

Los algoritmos se dividen en:

- Monobjetivo: una única función.

Consiste en encontrar los valores de las variables de decisión que producen el máximo o mínimo de una función objetivo.

- Multiobjetivo: más de una función.

En general, no existe una única solución, dividiéndose en dos grupos:

Soluciones dominadas: considerando todas las funciones objetivo del problema; serán peores que las otras soluciones.

Soluciones no dominadas: que al ser comparadas con respecto a todas las restantes, serán mejores en unas o más funciones objetivas y peores en otra u otras.

1 Véase la referencia 49 del capítulo "Referencias" del documento "Glosario, apéndice y bibliografía" para un análisis más detallado

3 Herramientas científicas existentes

Para el procesamiento matemático y graficado de los modelos de datos a visualizar, a lo largo de varios años se han venido elaborando herramientas científicas que cumplen con tal propósito. A continuación se brinda una investigación sobre la herramienta hasta el momento utilizada en la ilustración de los ensayos de bombeo, observaciones, y diferentes métodos de solución y optimización.

3.1 Matlab

3.1.1 Descripción general

Sus principales características son que se trata de un lenguaje de programación técnico, y provee un ambiente de programación para el desarrollo de algoritmos, visualización y análisis de datos, y computaciones numéricas ^[1].

El rango de utilización va desde el procesamiento de imágenes, graficado de pruebas y verificaciones, modelo y análisis financiero, hasta la realización de predicciones de biología computacional ^[2].

También facilita la documentación del trabajo, y el compartimiento del mismo con otros usuarios en la red.

3.1.2 Características

- Lenguaje técnico de alto nivel
- Funciones matemáticas para resolver álgebra lineal, estadísticas, optimización e integración numérica.
- Funciones para visualizar datos en dos y tres dimensiones.
- Herramientas para construir interfaces de usuarios personalizadas.
- Funciones para integrar algoritmos basados en MATLAB con aplicaciones o lenguajes externos, como C, C++, Java, Fortran.

1 Computación numérica: Es una rama de las matemáticas que resuelve el problema de usar aproximaciones numéricas. Abarca el diseño de métodos que dan una aproximada, pero precisa solución numérica.

2 Véase la referencia 48 del capítulo "Referencias" del documento "Glosario, apéndice y bibliografía" para una análisis más detallado

3.1.3 Herramientas

Matlab posee una amplia gama de herramientas y librerías con las cuales trabajar. A continuación se compendian las propiedades más destacadas de las librerías incluidas por Matlab, que pudiesen ser útiles para llevar a cabo el proyecto.

3.1.4 Exportación e Importación de archivos externos.

Posee funciones para abrir archivos y transferir datos desde y hacia Matlab.
Soporte para planillas Excel y archivos Lotus1-2-3.
Lectura y escritura a nivel de bytes.
Obtención de información de archivos y directorios.
Lectura y escritura de audio y video. Grabación y reproducción.
Compresión y obtención de información desde archivos comprimidos.

3.1.5 Desarrollo de la interfaz de usuario

La presentación es un aspecto que también es abarcado por esta herramienta, tanto es así que existen un sinnúmero de funciones y librerías que le permiten al desarrollador avanzar en la creación de interfaces gráficas (GUI's).

A continuación se enumeran opciones para la GUI.

Cajas de diálogo para mostrar errores o interactuar con el usuario.
Interfaz para desarrollar y desplegar la GUI.
Funciones que manejan eventos como mover objetos, arrastrar textos, capturar el paso del mouse por encima de los mismos.
Funciones para ubicar a los diferentes objetos creados.

3.1.6 Gráficas

Matlab abarca tareas gráficas como:

Creación de gráficos simples como gráficos lineales, log y semi logs.
Creación de anotaciones para escribir títulos, leyendas en cada eje o símbolos matemáticos.
Creación de gráficos especializados: gráfica de barras, pastel, gráficos de contorno.
Impresión y exportación de figuras hacia formatos estándares.
Creación de objetos gráficos.

3.1.7 Visualización en 3D

Permite representación de matrices, visualizar funciones de dos variables, especificar el color en el mapa. Además se pueden programar los controles sobre la gráfica, ya sea escribir una función para cuando se hace un zoom sobre alguna parte específica, para rotar la imagen o establecer límites de visualización en torno a los ejes.

3.1.8 Funciones matemáticas y análisis de datos

Como lenguaje técnico provee una rica gama de operaciones numéricas, y de análisis.

Creación y operaciones básicas sobre arreglos y matrices.

Creación y operaciones especializadas sobre arreglos y matrices.

Aplicación de álgebra lineal; desempeña tareas como el análisis de matrices, factorización, empleo de logaritmos y exponenciales.

Aplicación de matemática elemental: trigonometría, valores complejos, matemática discreta.

Operaciones sobre polinomios.

Interpolación y geometría computacional.

Algoritmos de reordenamiento, operaciones sobre árboles.

Análisis de los datos mediante estadística descriptiva.

Filtrado y circonvolución de datos.

Análisis de derivadas e integrales.

3.1.9 Programación Orientada a objetos

Es una característica nueva del lenguaje. Provee todas las funcionalidades para desarrollar bajo este paradigma de la programación.^[1]

3.1.10 Ventajas y desventajas

3.1.10.1 Ventajas

Es una herramienta cuya existencia data de 1984, por lo tanto es consolidada en el mercado.

Posee una comunidad de usuarios numerosa e importante.

La empresa propietaria brinda capacitación online y cursos a distancia.

La documentación técnica, de referencia y para el usuario final disponible en la página oficial es clara y bien explicada, inclusive existen demos para descargar.

Es multiplataforma, pudiendo operar en Windows, Linux y MAC OS.

Sus funciones pueden ser referenciadas desde varios lenguajes de programación.

3.1.10.2 Desventajas

Tiene licencia propietaria, es decir, se es poco permisivo sobre el código del programa, no se lo puede modificar, y su uso está restringido a las condiciones de la licencia.

Requiere de una memoria mínima de 1024 Megas.

Una instalación típica requiere más de 4 gigas de espacio en disco.

¹ Véase el apéndice 2.5 del documento "Glosario, apéndice y bibliografía"

4 Investigación de tecnologías

4.1 Objetivo

Encontrar el lenguaje de programación que aporte soluciones a los requerimientos, que en lo posible abarque todas las funcionalidades del proyecto, sin necesidad de estar dependiendo de componentes de terceros.

4.2 Python

4.2.1 Descripción

Python es un lenguaje de programación interpretado. Python nos permite trabajar de manera más rápida e integrar nuestro sistema más efectivamente. Cuenta con una sintaxis simple y minimalista muy fácil de aprender. Podemos aprender a usar Python y ver inmediatamente las ganancias en productividad y tiempo de mantenimiento. ^[1]

Python es de uso gratuito, incluso bajo uso comercial, por su licencia Open Source. Es esta licencia la que permite que un sinnúmero de programadores apoyen Python creando librerías extensibles que brindan una enorme cantidad de facilidades para los demás programadores.

Python se encuentra hoy disponible en los sistemas operativos Windows, MacOS, y Linux ^[2] y cuenta además con una vasta documentación ^[3].

4.2.2 Características ^[4]

- Gratuito y Open Source
- Lenguaje Interpretado.
- Lenguaje de alto nivel.
- Lenguaje Orientado a Objetos
- Portable o multiplataforma.
- Vasta colección de Librerías extensibles.

1 Para una información más detallada acerca de la sintaxis, ir al apéndice 2.3.1.1 del documento "Glosario, apéndice y bibliografía".

2 Para una información más detallada acerca de los requerimientos para correr Python, ir al apéndice 2.3.1.2 del documento "Glosario, apéndice y bibliografía".

3 Para una información más detallada acerca de la documentación, ir al apéndice 2.3.1.3 del documento "Glosario, apéndice y bibliografía".

4 Para una información más detallada acerca de cada característica, ir al apéndice 2.3.1.4 del documento "Glosario, apéndice y bibliografía".

4.2.3 Herramientas: Librerías destacadas, herramientas

4.2.3.1 NumPy

Es el paquete de librerías fundamental para computar cálculos científicos. Contiene entre otros: Arreglos N-Dimensionales, Funciones matemáticas, Herramientas para integrar C/C++ y código Fortran, Utilidades para álgebra lineal.

4.2.3.2 Framework Qt – PyQt

Qt es una aplicación multiplataforma, que permite un entorno de desarrollo para generar interfases de usuario. El lenguaje que utiliza es C++.

PyQt se sirve de las propiedades de Qt, esto es la posibilidad de desarrollar interfases para el usuario en múltiples plataformas, -Linux, Mac, Windows-, así como de las propiedades de PyQt o sea ligar Qt, cuyo código es C++, con el lenguaje Python.

4.2.3.3 Matplotlib ^[1]

Es una herramienta de graficado en dos dimensiones para Python. También incluye extensiones para graficar o representar datos en tercera dimensión. Dado que el proyecto surgió por una inconformidad hacia MatLab, este no sólo cubre sus funcionalidades si no que también intenta dar una mejor alternativa.

Con su lema *“Hacer fácil las cosas fáciles, y las cosas difíciles: posibles”*, la idea es lograr representaciones con unas pocas líneas de código.

El graficado comprende gráficas de barras, de dispersión, de error, histogramas, diagramas de espectro, entre otras opciones. También se puede personalizar el plano, con funciones que modifican el aspecto de los ejes, puntos o barras.

Numpy y matplotlib-mlab ^[2] son las librerías matemáticas de Python utilizadas.

1 Para una información mas detallada acerca de Matplotlib, ir al apéndice 2.3.1.5 del documento “Glosario, apéndice y bibliografía”.

2 Para una información mas detallada acerca del componente Matplotlib-mlab, ir al apéndice 2.3.1.5.1 del documento “Glosario, apéndice y bibliografía”.

4.3 Java

4.3.1 Descripción General

Java es actualmente el lenguaje de programación orientado a objetos más popular, desarrollado por Sun Microsystems a principios de los 90. Es multiplataforma, portable y no requiere licencias privativas para desarrollo ya que es software libre.

Es un lenguaje interpretado, ya que por lo general las aplicaciones Java son compiladas en bytecodes ^[1]. Luego, en tiempo de ejecución el código es traducido a código máquina ^[2] a través de la máquina virtual de java JVM ^[3].

Con Java se han desarrollado potentes y eficientes aplicaciones tanto a nivel personal como empresarial que están presentes en cualquier tipo de dispositivo digital: equipos, teléfonos móviles, portátiles, tarjetas inteligentes, sintonizadores, impresoras, cámaras web, juegos, sistemas de navegación para automóviles, terminales de lotería, dispositivos médicos, cajeros, centro de datos, super equipos científicos, etc. Por esto, hoy en día existen vastas cantidades de librerías y APIs ^[4] desarrolladas con este lenguaje.

4.3.2 Características

- Interpretado
- Orientado a objetos
- Multiplataforma y portabilidad
- Fácil de usar
- Trabajo en red

1 Véase la referencia 45 del capítulo "Referencias" del documento "Glosario, apéndice y bibliografía" para una análisis más detallado

2 Véase la definición de código máquina en el capítulo "Glosario" o la referencia 46 del capítulo "Referencias" del documento "Glosario, apéndice y bibliografía" para una análisis más detallado

3 Véase la referencia 47 del capítulo "Referencias" del documento "Glosario, apéndice y bibliografía" para una análisis más detallado

4 Véase la definición de API en el capítulo "Glosario" del documento "Glosario, apéndice y bibliografía" para una análisis más detallado

4.3.3 Herramientas: Librerías, Frameworks, Ámbito de desarrollo

4.3.3.1 JRE

En el caso del desarrollo de una aplicación de escritorio, cada PC de la organización que vaya a instalar este software, necesitará tener instalado JRE (Java SE Runtime Environment). Existe una versión de JRE para cada sistema operativo, y es el entorno de ejecución que permite que una aplicación Java se ejecute en un computador con determinadas características de hardware y Sistema Operativo.

4.3.3.2 OpenGL

Provee un ambiente de desarrollo de gráficas interactivas 2D y 3D. Parte de su carácter es el hecho de ser multiplataforma. Esta herramienta, al igual que su par GNPLOT tiene varios años de existencia e implementación en el ámbito de la programación, siendo el año de su primera versión 1992, y asegurando ser una herramienta capaz de utilizarse para el diseño asistido por computadora, C.A.D ^[1].

OpenGL, hasta el momento, es un estándar multiplataforma independiente de cualquier empresa.

Lenguajes de programación soportados:

C, C++, Python, Fortran, Perl y Java..

4.3.3.3 Java OpenGL

Es la librería ^[2] de Java que permite acceder desde este lenguaje a OpenGL.

Es actualmente desarrollada por el grupo Game Technology Group de Sun Microsystems.

A través del JNI ^[3], Java OpenGL (JOGL) puede comunicarse con OpenGL.

JOGL involucra un conjunto de clases para la representación gráfica:

GLDrawable
GLCanvas
GLJpanel
GLCapabilities
GLDrawableFactory

También incluye una clase para responder a los eventos del usuario, ya sea clickear, o mover el mouse, o apretar una combinación de teclas:

GLEventListener

1 CAD Sigla anglosajona que significa Diseño Asistido por Computadora. Es una forma de englobar al conjunto de herramientas informáticas al servicio ingenieros y arquitectos, con un fin profesional.

2 Véase la definición de Librerías en el capítulo "Glosario" del documento "Glosario, apéndice y bibliografía"

3 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.

4.3.3.4 Jzy3D

Java Easy 3D es una librería para representar datos científicos en 2 y 3D. Esta API ha sido escrita con el fin de ocultar la complejidad de trabajar directamente con OpenGL, como en el ordenamiento de polígonos 3D y la gestión de transparencias. Jzy3d es la respuesta a la necesidad de una herramienta simple, similar a plt3d en Matlab, que permita al usuario concentrarse en los datos en lugar de programar la interfaz gráfica.

El usuario puede definir una función $z=f(x,y)$ para vincularlas a mapas de colores, así como interpretar simples figuras 3D predefinidas o generar gráficos a partir de archivos csv ^[1]. Diferentes estrategias pueden ser aplicadas para generar las superficies y gráficos desplegados. Además permite captura de imágenes, en archivos con formato png.

El sitio web de Jzy3d es: <http://code.google.com/p/jzy3d/>.

Esta librería cuenta con una documentación oficial en inglés. Además a disposición de los programadores hay varias demostraciones, dos sitios con documentación y grupo de Google. Para acceder a los enlaces hacia la documentación de esta librería y conocer su funcionamiento, ir al anexo Java.

4.3.3.5 Processing

Processing es un lenguaje y ambiente de programación Open Source ^[2] para personas que quieren crear imágenes, animaciones e interacciones. Inicialmente desarrollado para servir como un software para hacer dibujos y para enseñar los fundamentos de la programación con un contexto visual, Processing ha evolucionado en una herramienta para generar completos trabajos profesionales. Este software es usado por estudiantes, artistas, diseñadores e investigadores para el aprendizaje, prototipación y producción.

Características:

- Libre para bajar y Open Source.
- Permite crear programas interactivos usando salidas 2D, 3D o PDF.
- Tiene integración con OpenGL para aceleración 3D.
- Se puede correr en las plataformas: GNU/Linux, Mac OS X y Windows.
- Tiene más de cien librerías que permiten extender el software para generar videos con sonido y muchas cosas más.
- Cuenta con excelente documentación; además existen varios libros disponibles.

1 Véase la definición de CSV en el capítulo "Glosario" del documento "Glosario, apéndice y bibliografía"

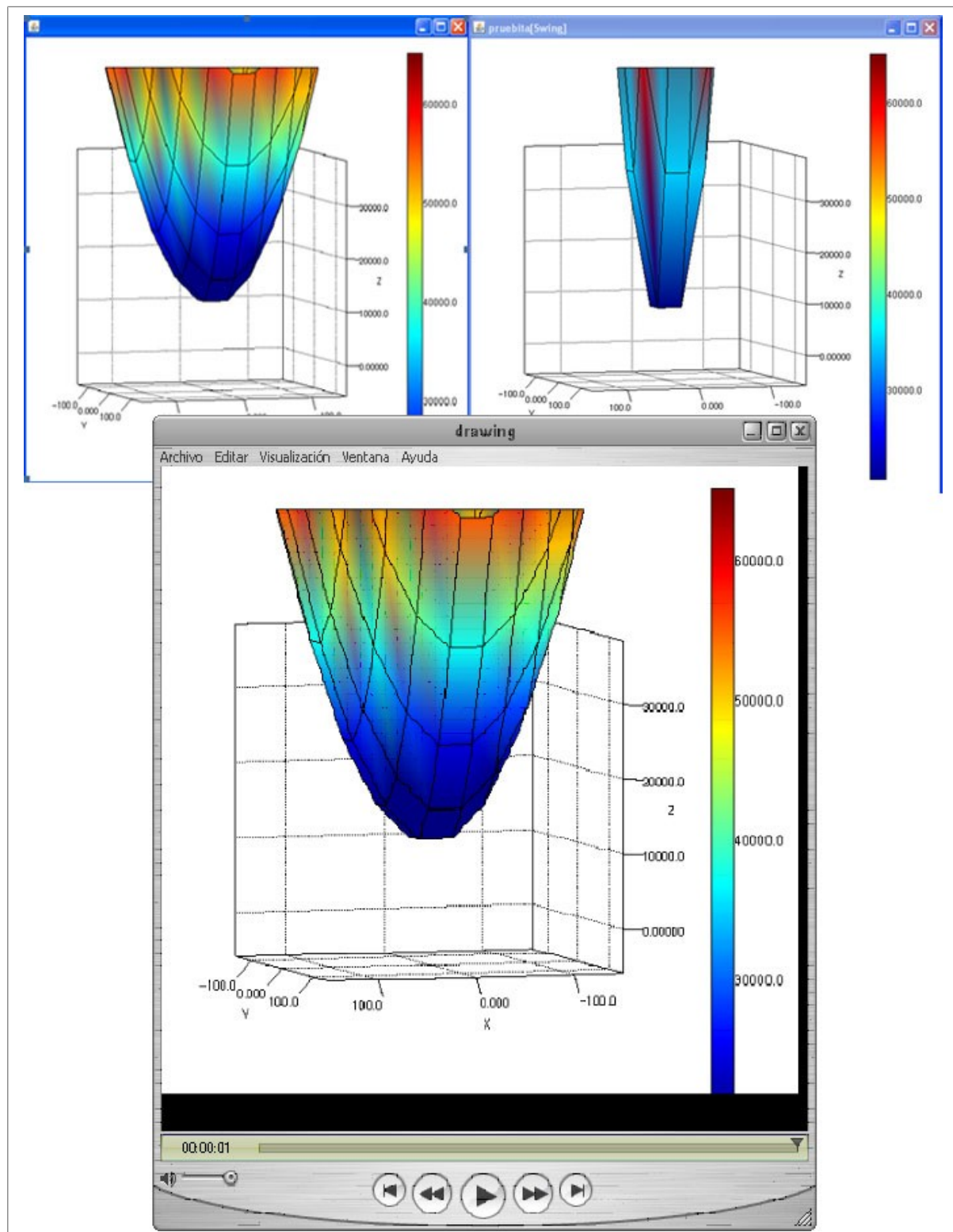
2 Véase la definición de Open Source en el capítulo "Glosario" del documento "Glosario, apéndice y bibliografía"

4.3.4 Soporte y Documentación

Con respecto a la documentación, existe una gran cantidad de material en sitios de Internet, así como también en las páginas oficiales de Java y en bibliografías.

Así como sucede con otras tecnologías Open Source y software libre existe una gran cantidad de foros, blogs, soporte sobre Java, 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.

4.3.5 Ejemplo de implementación combinando Jzy3D con Proccesing



Se generó un ejemplo combinando Processing con Jzy3d. Al ejecutar esta aplicación de prueba se generó una animación por medio de Jzy3d utilizando sus threads ^[1] para hacer animaciones. La función de la gráfica va cambiando a medida que transcurre el tiempo para simular como se vería representado el registro de un episodio similar a un ensayo de bombeo según lo que nos pudo mostrar en la primera reunión el profesor Gamazo.

La función es $f(x,y)=\text{Math.abs}(p*\text{Math.pow}(x,2) + p*\text{Math.pow}(y,2) + 20000/p)$; donde p va cambiando con el tiempo, cada 100 milisegundos. Con cada incremento del tiempo que registre un tiempo, se incrementa la variable p hasta llegar a 20 modificando la forma de la gráfica. Luego este valor empieza a descender hasta retornar a la función original. Con cada incremento del tiempo, se toma captura de la pantalla para luego generar el video a partir de una secuencia de imágenes.

Al alcanzar las 40 capturas, a los 12 segundos se detiene la animación y se llama a la clase Papplet encargada de renderizar la animación. La misma se incrusta en otra ventana para mostrar el video generado. Mientras transcurren los 12 segundos, se puede cambiar la escala o la posición desde donde se quiere ver la gráfica y eso se va a reflejar en el video generado. Al terminar, se va guardar el video generado en formato .mov en un fichero que como se muestra en la captura anterior se puede ver en cualquier reproductor que soporte el formato del fichero.

En este ejemplo las imágenes generadas no se almacenan en disco, sino que permanecen en memoria hasta que se renderice el video.

1 Término referente a la tecnología de componer un proceso (dentro de un sistema operativo) en varios hilos, donde cada uno de ellos se encarga de una tarea totalmente distinta y específica, ejecutándose todos al mismo tiempo, creando un entorno de multitarea, incrementando enormemente el tiempo de respuesta del procesador (CPU)

4.3.6 Ventajas y desventajas

4.3.6.1 Ventajas

- Tenemos experiencia y buen conocimiento del lenguaje.
- Madurez: Java ha sido probado, mejorado y ampliado.
- Orientado a objetos, lo que facilita la modularización de la aplicación, el trabajo en equipo y la localización más rápida de fallas.
- Portabilidad: Java permite desarrollar software en una plataforma (sistema operativo) y ejecutarlo en prácticamente cualquier otra plataforma.
- Versatilidad, eficiencia
- Existe una gran comunidad de desarrolladores que utilizan y dan soporte a esta tecnología, ya que según la página oficial son más de 6,5 millones de desarrolladores. Es la mas activa del mundo.

4.3.6.2 Desventajas

- La continuidad de Java en manos de Oracle puede significar un giro inesperado en el futuro de la herramienta que conduzca a que para el desarrollo y distribución de software bajo esta tecnología sean necesarias licencias privativas.

4.3.7 Referencias

[Vista general sobre OpenGL]

<http://www.opengl.org/about/>

[Explicación sobre JOGL]

<http://www.docstoc.com/docs/23958233/JAVA-OPEN-GL>

[Página oficial de Jzy3D]

<http://code.google.com/p/jzy3d/>

[Apéndice JOGL]

<http://www.cs.umd.edu/~meesh/kmconroy/JOGLTutorial/>

4.4 C/C++

4.4.1 Descripción General

Lenguaje C:

Es definido como un lenguaje de programación de propósito general, y como un lenguaje potente para la programación de Sistemas ^[1].

En 1983 fue estandarizado por ANSI ^[2], conformándose un Comité para dicha empresa, teniendo como objetivo la portabilidad ^[3] y operabilidad en diferentes máquinas.

Lenguaje C++:

Nace del lenguaje anteriormente descrito, siendo una evolución de C. C++ trae nuevos conceptos de programación que se suman a los ya existentes para C, como por ejemplo la programación orientada a objetos ^[4].

C++ agrega nuevas características, y permite un mayor nivel de abstracción, por ejemplo ya no se necesita ser tan cuidadoso con la administración de la memoria del computador; el programador puede entonces terminar centrando su preocupación, un poco más, en la resolución del problema real, simplificando así la programación.

4.4.2 Características

- Portabilidad.
- Lenguaje estructurado..
- Lenguaje procedimental.
- Orientación a objetos.
- Escalable.
- Librerías funcionales no estándares.

1 A finales de los '80 del siglo pasado apareció como una evolución de los lenguajes BCPL y B (véase definiciones en el capítulo "Glosario" del documento "Glosario, apéndice y bibliografía") haciéndose en aquella época, un lenguaje popular

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

3 Portabilidad, fundamento de la Ingeniería de Software, que describe la aptitud de un software de funcionar en diferentes máquinas y sistemas operativos.

4 Véase el apéndice 2.5 del documento "Glosario, apéndice y bibliografía"

4.4.3 Herramientas

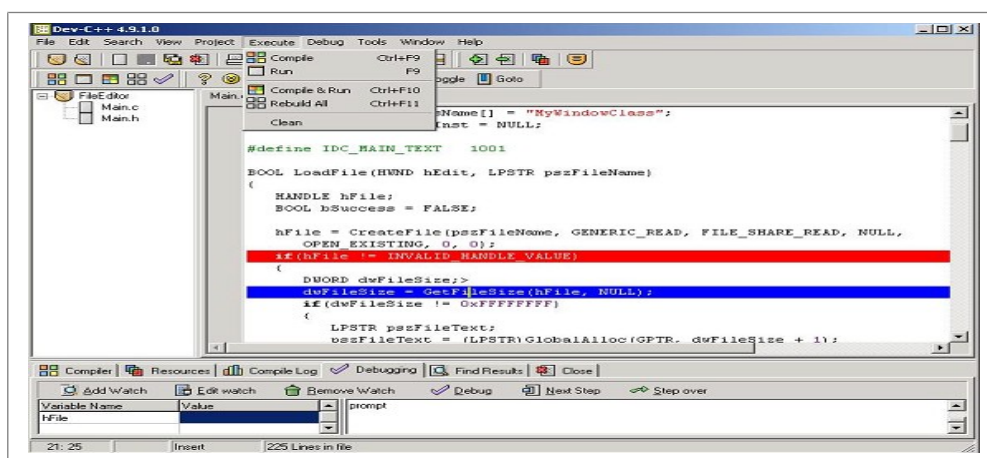
4.4.3.1 Compilador

Fundamental para interpretar el código de cualquier lenguaje. Es la herramienta que permite volver ejecutable el código que uno ha programado.

Existen varios IDE'S ^[1] tanto libres como privativos.

4.4.3.2 IDE BloodShed Dev C++

Provee un entorno de desarrollo; el tipo de compilador que utiliza son los basados en la Colección de Compiladores GNU ^[2]. Tanto su descarga como su instalación son gratuitas.



(La figura muestra un proyecto en C, utilizando el IDE DevC++)

Instaladores:

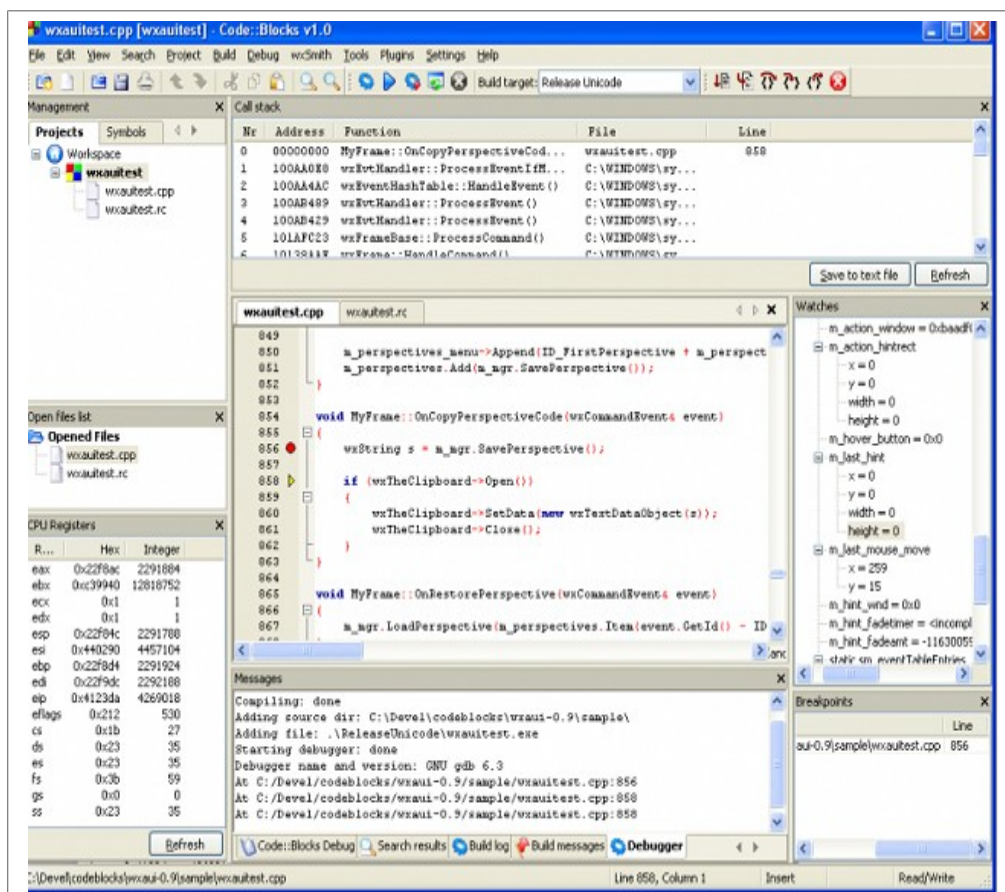
<http://www.bloodshed.net/download.html>

4.4.3.3 IDE Code:Blocks

Entorno de desarrollo libre que provee funcionalidades importantes, como compilación y detección de errores en el código. Al igual que DevC++, se basa en compiladores GNU.

1 Integrated Development Environment, ambiente de desarrollo integrado, los IDE'S proveen herramientas que facilitan la programación.

2 GNU, por extensión se utiliza como un término asociado al Software libre.



(Entorno de desarrollo en Code:Blocks, los diversos IDE's proveen funcionalidades similares)

Instaladores:

<http://www.codeblocks.org/downloads>

Las características notables de los IDE's son: organización gráfica del proyecto, barra de herramientas para diversas acciones y posibilidad de compartir un proyecto de desarrollo en red, a través de la Web.

4.4.3.4 Librerías estándares

Las librerías estándares son herramientas simplificadoras de la codificación del lenguaje. Las más utilizadas son:

4.4.3.4.1 iostream.h

Utilizada para el flujo de entrada y salida de datos.

El literal cin, nos permite enviar datos, y cout, posibilita extraerlos ^[1].

4.4.3.4.2 String.h

Utilizada para almacenar una cadena de caracteres, desde el inicio de la palabra hasta el primer espacio en blanco que encuentre. También provee métodos de comparación, y operación sobre cadena de caracteres ^[1].

4.4.3.4.3 stdio.h

Incluye las funciones de entrada y salida, tanto del teclado y la pantalla, como también entrada y salida de ficheros. En suma cumple las mismas funciones que iostream; no obstante la ventaja para el manejo de entrada y salida reside en que al ser un estándar ANSI C, el código puede ser interpretado por C y C++ ^[1].

4.4.3.4.4 stdlib.h

Librería que permite utilizar el método system, de utilidad para ejecutar comandos del sistema; también nos permite ejecutar un programa externo desde consola ^[1].

4.4.3.5 Librería Matemática

4.4.3.5.1 Math.h

Provee funciones trigonométricas, hiperbólicas, exponenciales, logarítmicas, de redondeo y de valor absoluto.

1 Véase el apéndice 2.3.3 del documento "Glosario, apéndice y bibliografía" por mas ejemplos.

4.4.3.6 Librerías de flujo de datos

El mouse, teclado y ficheros son un subconjunto de los diferentes elementos con los cuales interactúa el sistema operativo. C y C++ también pueden responder a entradas de datos por teclado, al movimiento del mouse o escribir en un fichero, produciéndose un intercambio de datos de entrada y salida entre los elementos y C; al este enviar y recibir se le conoce como *flujo de datos*.

El lenguaje no incorpora la comunicación con los diferentes dispositivos, sino que le deriva este trabajo a librerías externas, hechas para cada sistema operativo en particular. Es decir, el método que se invoca para abrir un fichero va a ser sintácticamente igual en Windows y en Linux, lo único que cambia es la implementación de ese método, lo cual es transparente para el desarrollador.

4.4.3.6.1 *ifstream, ofstream, fstream*

Dentro de las clases útiles para el manejo de ficheros, se encuentran estas tres; *ifstream* es específica para leer datos de un archivo. *ofstream* tiene aptitud para escribir datos en un archivo, mientras que *fstream* es una librería que permite tanto escritura como lectura sobre archivos.

Son funcionales tanto para el procesamiento de texto o imágenes u otro tipo de archivos. Nótese que utilizando simples y conocidas clases se puede procesar cualquier tipo de fichero ^[1].

1 Véase el apéndice 2.3.3.2 del documento “Glosario, apéndice y bibliografía” para analizar un ejemplo de procesamiento de archivo con estas clases

4.4.3.7 Interfaz de Usuario

4.4.3.7.1 GTK+

Gimp Toolkit está conformado por un conjunto de librerías, escritas en C, para desarrollar interfaces gráficas. Fue creado dentro del ámbito de los sistemas GNU, pero actualmente es usado como una API multiplataforma, pudiéndose utilizar en MacOS y Windows; para este último se requieren unas librerías que pueden ser descargadas del siguiente enlace:

<http://sourceforge.net/projects/gladewin32/files/gtk%2B-win32-devel/2.12.9/>

GTK+ puede ser desarrollado desde código o también utilizando una interfaz gráfica, intuitiva y amigable como el GUI Glade, una herramienta pensada para el rápido desarrollo de la aplicación, presentando diferentes opciones gráficas al desarrollador.

GTK+, dentro de sus librerías incluye a Cairo, utilizada para hacer gráficos en dos dimensiones, Pango la cual permite la internacionalización del programa, y ATK, que permite brindar otro tipo de interacción a personas con discapacidades físicas. Si bien no incorpora gráficos 3d, GTK+ provee una extensión para utilizar OpenGL (Véase Artículo sobre OpenGL)

4.4.3.7.2 QT

Framework codificado en C++, para hacer interfaces de usuario. Es multiplataforma, y es software libre.

Provee una programación orientada a objetos, interfaz gráfica con elementos variados, desde dialogos para interacción con el usuario o presentación de mensajes de advertencia o errores, hasta botones, etiquetas y contenedores de objetos, útiles para el ordenamiento de los mismos en la aplicación en la que esten corriendo.

Implementa el patrón MVC¹ en varios de los frameworks que a él pertenecen.

Su documentación oficial es clara, basta en descripción y ejemplos y provee una navegación con enlaces relacionados al tema que se esté leyendo.

¹ Véase punto 4.3.3 del documento Informe.

4.4.4 Ventajas y Desventajas

4.4.4.1 Ventajas

Se puede programar en cualquier computador, no demanda demasiados requerimientos para funcionar.

Es un lenguaje multipropósito y por lo tanto permite programar cualquier aplicación.

Si bien es un lenguaje en el que se precisa mucha codificación, también en pocas líneas se pueden resolver los problemas presentados.

Requiere que se sea consciente de todos los aspectos y detalles del programa, por ende el conocimiento del programador es mayor.

4.4.4.2 Desventajas

El programador debe de codificar la mayoría de las funciones, desde generar una ventana, y los eventos a los que responderá el programa, hasta detalles como cerrar adecuadamente la conexión con un fichero. Lenguajes como Java ayudan al programador a enfocarse más en el problema a resolver, y no delegarle tantos detalles de diseño y arquitectura.

Las líneas de código y funciones tienden a ser extensas, y con fines muy diversos en el programa; los desarrolladores deben generar políticas que gobiernen la presentación del código y comentar cada función en virtud de la entendibilidad.

No tiene tantos tipos de datos como otros lenguajes compilados, con lo cual aumenta el número de líneas de código en el programa.

4.4.5 Referencias

[Libro sobre el lenguaje C]

Programación en C Metodología Algoritmos y Estructura de datos Editorial.McGraw-Hill

[Curso en pdf, del sitio <http://www.conclase.com>]

Curso de C++ Versión 2003 – Salvador Pozo Coronado

[Sitio con Tutoriales y Código fuente para C/C++]

<http://www.cprogramming.com/>

[Sitio descriptor de librerías de referencia para C++]

<http://en.cppreference.com/w/cpp>

[Comunidad de C++]

<http://www.cplusplus.com/>

[Tutorial GTK+]

<http://zetcode.com/tutorials/gtktutorial/introduction/>

<http://developer.gnome.org/gtk-tutorial/2.90/>

5 Codificadores de video

Para este proyecto, se nos ha pedido que exportemos una secuencia de imágenes a video. Para ello es necesaria una herramienta conocida como codificador de video, que básicamente se encarga de convertir un grupo de imágenes en un formato de video y encapsularlo en un formato contenedor o viceversa.

Nuestra investigación nos ha arrojado las siguientes herramientas para esta tarea:

5.1 FFMPEG

5.1.1 Descripción general

FFMPEG es una aplicación multi plataforma que permite grabar, reproducir y transmitir video y audio. Utiliza libavcodec (del proyecto FFMPEG), la librería líder en audio y video.

Está liberado bajo la licencia GNU/GPL versión 2.

5.1.2 Características

- Se puede tomar y codificar video/audio en tiempo real desde una tarjeta de TV.
- Los fotogramas de entrada pueden ser editados como texto y / o superposición de imágenes.
- Reúne múltiples archivos de entrada en un único archivo de salida.
- Soporta entrada de fotogramas.
- Soporta salida en fotogramas (RGB, MJPEG, h.264, YUV).
- Soporta captura de pantalla.
- Soporta captura de audio.
- Soporta filtros de video, como volteo, negativo, escala, etc.
- Soporta pre visualización durante conversión.
- Se puede configurar la prioridad de los hilos de ejecución.
- Operaciones para pausar, resumir y detener están disponibles.
- Evento de conversión en progreso.
- Evento de registro de mensajes de monitorización gracias al componente FFLogger.

5.1.3 Bibliografía

Cuenta con una documentación oficial bastante nutrida y una comunidad bastante grande.

Sitio web oficial: <http://ffmpeg.org/>.

Nota: Recientemente ha ocurrido una serie de altercados entre los desarrolladores de esta aplicación, por lo que el proyecto terminó perdiendo desarrolladores y ganando un competidor más: ahora hay un proyecto llamado libav que se basa en esta iniciativa pero que es incompatible con el mismo.

5.2 Mencoder

5.2.1 Descripción General

Es una aplicación parte del paquete del reproductor MPlayer que permite la codificación de audio y video en varios formatos.

Es multi plataforma y está liberado bajo la licencia GNU/GPL versión 2.

5.2.2 Características

- Codificación de la gran variedad de formatos y decodificadores de MPlayer.
- Codificación de todos los codecs de libavcodec (FFMPEG).
- Codificación de video V4L compatible con sintonizadores de TV.
- Codificación/multiplexado hacia archivos AVI intercalados con índice apropiado.
- Crea archivos de fuentes de audio externas.
- Codificación de uno, dos o tres pasos.
- Audio MP3 con tasa de bits variable.
- Audio captado por modulación por impulsos codificados.
- Corrección de fps mediante el comando -ofps (útil cuando se está codificando 30000/1001 fps VOB a 24000/1001 fps AVI).
- Poderoso sistema de filtrado (cortar, expandir, dar vuelta, pos-procesar, rotar, escalar, conversión RGB/YUV).
- Puede codificar DVD/VOBsub y texto de subtítulos en el archivo resultante.
- Puede pasar subtítulos de DVD a formato VOBsub.

5.2.3 Bibliografía

Cuenta con una documentación oficial bastante nutrida y una comunidad bastante grande.

Sitio web oficial: <http://mplayerhq.hu>.

5.3 Librerías *libavcodec* y *libavformat*

5.3.1 Descripción General

Estas librerías ofrecen funciones y procedimientos para la manipulación de archivos de audio y video.

Ambas librerías son utilizadas por las aplicaciones antes mencionadas (Mencoder y FFMPEG) y muchas otras aplicaciones más.

5.3.2 Características

- Son multi plataforma, lo cual permite su uso en distintos sistemas operativos.
- Cuenta con proyectos derivados para otros lenguajes como pyFFMPEG (python), jFFMPEG (java) y ffruby (ruby).
- Son unas de las librerías más utilizadas para el desarrollo de aplicaciones que requieren la manipulación de audio y video.
- Escasa documentación.
- La poca documentación que hay es bastante pobre.
- Ambas librerías están siendo desarrolladas por equipos distintos de programadores pero mantienen el mismo nombre, lo que puede causar aún más confusión a la hora de su uso. Además son incompatibles.

5.3.3 Bibliografía

Desgraciadamente, no tiene una buena documentación y es muy escasa.

Información oficial en las páginas <http://www.ffmpeg.org/doxygen/trunk/index.html> y <http://libav.org/doxygen/master/index.html>.