

Artículo publicado en el Repositorio Institucional del IMTA

<i>Título</i>	Expresión de características de accesorios en trenes de descarga de pozos de bombeo.
<i>Autor / Adscripción</i>	Julio Sergio Santana Edmundo Pedroza González Instituto Mexicano de Tecnología del Agua
<i>Publicación</i>	Ingeniería Hidráulica en México, 19(4): 65-84
<i>Fecha de publicación</i>	2004
<i>Resumen</i>	Se discute el problema del manejo de la información para la descripción adecuada de las plantas de bombeo. Se introduce un lenguaje gráfico que permite expresar la configuración de trenes de descarga de pozos de bombeo. Este lenguaje permite describir los accesorios instalados en el tren, su disposición y características particulares. El lenguaje, desarrollado en el contexto de un sistema de información para la administración de/ agua, se describe formalmente mediante sus reglas sintácticas y semánticas. Finalmente, se muestra la incorporación del lenguaje al sistema de información mediante una interfaz apropiada con el usuario y se concluye que una herramienta de esta naturaleza es útil para el levantamiento en el campo de grandes volúmenes de información de las instalaciones de pozos de bombeo.
<i>Identificador</i>	<a href="http://hdl.handle.net/123456789/1278">http://hdl.handle.net/123456789/1278</a>

# Expresión de características de accesorios en trenes de descarga de pozos de bombeo

Julio Sergio Santana  
Edmundo Pedroza-González

Instituto Mexicano de Tecnología del Agua

*Se discute el problema del manejo de la información para la descripción adecuada de las plantas de bombeo. Como un paso encaminado a la solución de este problema, se introduce un lenguaje gráfico que permite expresar la configuración de trenes de descarga de pozos de bombeo. Este lenguaje permite describir los accesorios instalados en el tren, su disposición y características particulares. El lenguaje, desarrollado en el contexto de un sistema de información para la administración del agua, se describe formalmente mediante sus reglas sintácticas y semánticas. Finalmente, se muestra la incorporación del lenguaje al sistema de información mediante una interfaz apropiada con el usuario y se concluye que una herramienta de esta naturaleza es útil para el levantamiento en el campo de grandes volúmenes de información de las instalaciones de pozos de bombeo.*

**Palabras clave:** sistemas de información, administración del agua, lenguajes gráficos, trenes de descarga, plantas de bombeo.

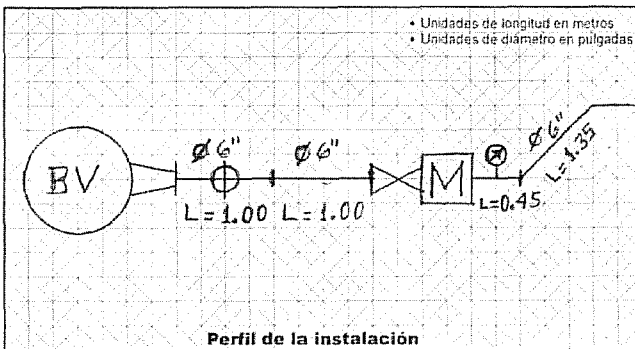
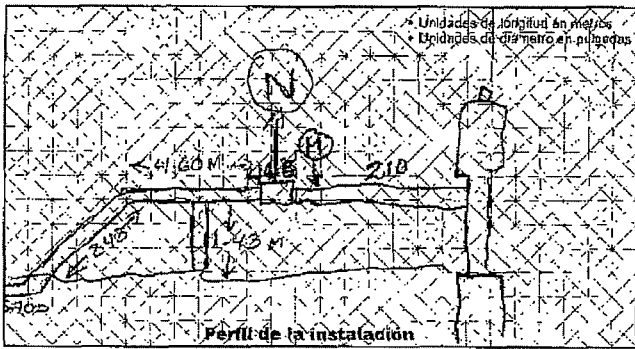
## Introducción

En 2001, la Comisión Nacional del Agua (CNA) encargó al Instituto Mexicano de Tecnología del Agua (IMTA) un proyecto para la mejora de la medición en pozos agrícolas. El desarrollo de este proyecto involucró el levantamiento de la información de alrededor de cuatro mil pozos en diversos distritos de riego en el país (Pedroza G. *et al.*, 2001; Pedroza G. *et al.*, 2002). Con este propósito se diseñaron formatos específicos y se recopiló la información en una base de datos (Santana y Pedroza, 2002). Durante el desarrollo de este proyecto, la información correspondiente al tren de descarga de cada pozo se colectó mediante un croquis ayudado por alguna información textual somera acerca del equipo instalado. Para el objetivo del proyecto, que era constatar el estado de operación del medidor instalado en cada pozo y generar una recomendación para la mejora de la medición con la posible inclusión de un nuevo medidor de acuerdo con criterios semejantes a los dados por Enríquez y Pedroza (Enríquez y Pedroza, 2001), esto era suficiente.

Sin embargo, como una base para registrar la operatividad de todo el tren de descarga, estos croquis resultaron insuficientes. Como se puede apreciar en la ilustración 1, la calidad de los croquis levantados en campo no resultó uniforme, ya que dependía del entendimiento y habilidad de cada operador encargado de su elaboración, y esto difícilmente permitiría sistematizar la totalidad de la información de los trenes de descarga; tómesese en cuenta el gran volumen de croquis que tendrían que ser analizados uno por uno.

Debido a que una propuesta para el mejoramiento integral de la administración de pozos agrícolas y urbanos requeriría de manera obligada registrar la información precisa del tren de descarga de cada uno de ellos, surgió la necesidad de contar con una herramienta capaz de expresar y reconocer esta información de manera sistemática. El presente artículo describe de manera breve el desarrollo y funcionamiento de la primera etapa de un sistema para la expresión de equipos y sus características en trenes de descarga de pozos de bombeo, que ha sido creado con ese propósito.

Ilustración 1. Dos croquis del tren de descarga.



### Información de trenes de descarga de pozos

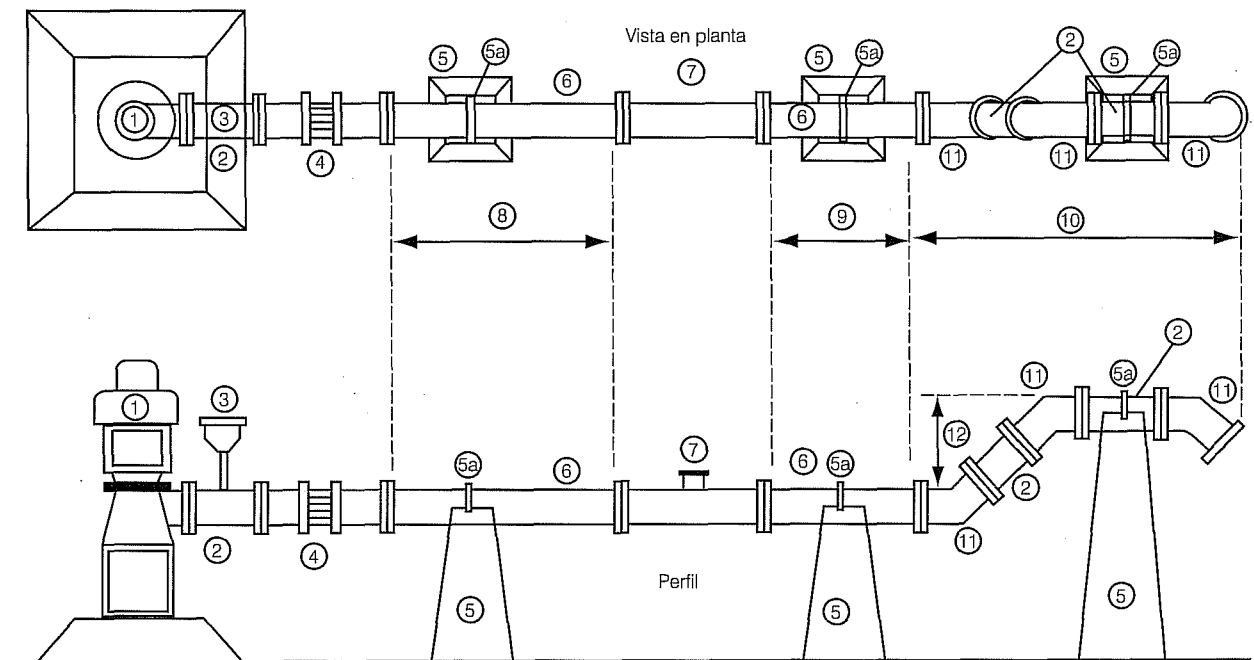
Los datos que se requieren para describir de manera unívoca cada tren de descarga son de una naturaleza diversa: numéricos, de texto, fechas, gráficos y fotográficos. En la ilustración 2 se puede apreciar un tren de descarga típico.

Cada uno de los elementos de la ilustración 2, con sus características particulares, debe ser registrado como parte de la información relativa al tren de descarga del pozo de bombeo. Debe notarse que tanto el equipo instalado en el tren de descarga como su disposición en el mismo es muy variable de acuerdo con las condiciones de cada pozo en particular.

### Estructura de la información

Para poder capturar la información de las condiciones de cada pozo, así como del arreglo de su tren de descarga, se propone una estructura como la que se muestra en la ilustración 3. Esta estructura se exhibe como un diagrama simplificado de entidad-relación (Chen *et al.*, 1998; Chen, 1996). En este diagrama las relaciones de una tabla a otra se señalan como una polilínea que une los campos de dos tablas que se relacionan. Un registro de una tabla

Ilustración 2. Un tren de descarga típico.



- |   |   |   |
|---|---|---|
| 1. Equipo de bombeo.                          | 5. Atraques.  | 9. Tramo recto aguas abajo del medidor. |
| 2. Carretes de Fo Fo.                         | 6. Carretes de tubo soldable (cédula 40) bridado en los extremos. | 10. Cuello de ganso.                    |
| 3. Válvula para admisión y expulsión de aire. | 7. Medidor de propela en carrete.                                 | 11. Codo de 45° de Fo Fo.               |
| 4. Junta mecánica (tipo Gibault o Dresser).   | 8. Tramo recto aguas arriba del medidor.                          |   |



- c) **Concesionario.** Nombre de la persona u organización a la que está concesionado el pozo.
- d) Campos de ubicación del pozo. Estos campos permiten ubicar geográficamente el pozo y son los siguientes: **NombPredio, IdEdo, IdMpio, Longitud, Latitud, Altitud y Acceso.**
- e) Campos de información general del pozo. Estos campos permiten identificar algunas características generales del pozo: **UsoPozo, AreaServ, Cultivos, LongDesc, TipoSalida, TSistPres, TuboLleno, ProfPozo.**
- f) **GastoConces.** Es el gasto concesionado del pozo.
- g) **GastoEst.** Es el gasto estimado.
- h) **DiaDescREP.** Diámetro de la descarga, según obra en el registro nacional de pozos.
- i) **NFotoTrenD.** Nombre del archivo que contiene la fotografía del perfil del tren de descarga del pozo.
- j) **NFotoMed.** Nombre del archivo que contiene la fotografía del medidor instalado en el pozo.
- k) Información del agua. Estos campos registran la calidad del agua que se extrae del pozo. Estos campos son los siguientes: **SolidSedim, pH, Temp, ObsCampo** (notas relevantes de la calidad del agua).
- l) **Recopilador.** Este campo es el identificador numérico de la persona que se encargó de recavar la información en el campo.
- m) **Fecha y Hora.** Es la fecha y la hora en la que se colectó la información en el campo.
2. **Equipo.** Mediante esta tabla se registra toda la información correspondiente al equipo instalado en el tren de descarga del pozo. Sus campos son:
- a) **IdEq.** El identificador numérico de un equipo instalado en el tren de descarga. Este campo también servirá como parte de la llave de la tabla.
- b) **Folio.** Es un campo para establecer la relación con la tabla principal FichaPozo.
- c) **NomEq.** El nombre del tipo de equipo instalado.
3. **Característica.** Cada equipo instalado en el tren de descarga tiene una serie de características de acuerdo con el tipo de equipo de que se trate. Los campos de esta tabla son:
- a) **IdCar.** Identificador de la característica. Por ejemplo, si el tipo de equipo del que se trata es una bomba, una de sus características podría ser la potencia de la bomba, y el identificador de esta característica pudiera ser **Potencia.**
- b) **IdEq.** Este campo es el identificador del equipo cuyas características se establecen mediante esta tabla. El campo, junto con el campo **Folio**, se usa para relacionar esta tabla con la tabla **Equipo.**
- c) **Folio.** Campo para relacionar con las tablas **Equipo y FichaPozo.**
- d) **TipoCar.** El tipo de dato para interpretar la descripción de la característica (campo **DescCar**), el cual puede ser Numérico o Texto. Por ejemplo, la potencia de una bomba podría estar dada como un dato numérico.
- e) **DescCar.** Mediante este campo se describe la característica cuyo nombre está dado en el campo **IdCar.** Por ejemplo, las características de una bomba pudieran ser:
- Tipo:** tipo de bomba, con tipo de dato TTxt.  
**Diam:** diámetro de la bomba, con tipo TNum.  
**Pres:** presión de trabajo de la bomba, con tipo TNum.  
**Pot:** potencia de la bomba, con tipo TNum.  
**Costo:** costo de la bomba, con tipo TNum.
- De acuerdo con esto, si **IdCar** es **Pot**, el dato registrado en este campo pudiera ser, por ejemplo, 75, indicando que se trata de una bomba con una potencia de 75 HP.
4. **DixEq.** Lo registrado en esta tabla permite cifrar y descifrar la información contenida en las tablas Equipo y Característica. Sus campos son:
- a) **IdCar.** Identificador de la característica que se describe aquí. La tabla Característica tomará de estos identificadores según el equipo de que se trate.
- b) **NomEq.** Nombre genérico del equipo que se describe.

c) **TipoCar.** Tipo de dato asociado con la característica, el cual puede ser numérico (TNum) o textual (TTxt).

- Estados.** Esta tabla auxiliar contiene los nombres oficiales de los estados del país y una clave numérica que funciona como índice de la tabla.
- Municipios.** Esta tabla auxiliar contiene los nombres oficiales de los municipios del país y una clave numérica que funciona como índice de la tabla.

Para tener una mejor idea de cómo se mantiene la información en estas tablas, en los cuadros 1 y 2 se muestran ejemplos de las tablas **Equipo** y **Característica**, que representan la parte más elaborada de la base de datos.

En el cuadro 1 se presenta la tabla del equipo instalado en el pozo que corresponde al número de folio 81. El equipo instalado en el tren de descarga consta de ocho elementos a saber: una bomba, un tramo recto de tubo, un medidor, otro tramo de tubo, un codo a 45°, un carrete, otro codo a 45°, y una extremidad. Al principio de esta sección de la tabla se muestra un elemento cuyo **IdEq** es 0, y cuyo **NomEq** es GLang. Este elemento no denota ninguna pieza particular de equipo, sino que proporciona el espacio para la introducción de una expresión del lenguaje GLang que describe el arreglo físico del equipo instalado. Este lenguaje, GLang, se describe a mayor detalle más adelante.

El cuadro 2 muestra la tabla de características del equipo. Para ayudar a su comprensión se ha incluido el campo **NomEq**, que no existe en la tabla real. Observemos los renglones 2, 3 y 4 del cuadro, esto es, aquellos renglones que tienen el texto Bmb en su columna **NomEq** (que son los mismos que tiene un valor 1 en la columna **IdEq**). Todos estos renglones o registros son los que se

**Cuadro 1. Ejemplo de la tabla Equipo.**

Folio	IdEq	NomEq
...	...	...
81	0	GLang
81	1	Bmb
81	2	Tubo
81	3	Med
81	4	Tubo
81	5	Codo 45
81	6	Carrete
81	7	Codo 45
81	8	Extrem
...	...	...
122	3	Vlv

**Cuadro 2. Ejemplo de la tabla Característica.**

Folio	IdEq	(NomEq)	IdCar	TipoCar	DescCar
81	0	GLang	Expr	TTxt	0: Bmb/1,Tubo/2,Med/3, Tubo/4,Codo45/5,Carrete/6, Codo45b/7,Extrem/8 .
81	1	Bmb	Tipo	TTxt	Vertical
81	1	Bmb	Diam	TNum	10
81	1	Bmb	Pot	TNum	60
...	...	...	...	...	...
81	3	Med	Tipo	TTxt	Propela
81	3	Med	Marca	TTxt	Micrometer
81	3	Med	Diam	TNum	10

usan para describir las características de la bomba, que se ha mencionado como el primer elemento en el tren de descarga descrito en la tabla del cuadro 1. En este caso se trata de una bomba vertical, con un diámetro de 10 pulgadas y una potencia de 60 HP. El primer renglón de la tabla (cuadro 2) registra la expresión del lenguaje GLang: "0: Bmb/1, Tubo/2, Med/3, Tubo/4, Codo45/5, Carrete/6, Codo45/7, Extrem/8". Por el momento baste decir que esta expresión denota una disposición del equipo como la secuencia simple que empieza en la bomba Bmb/1, y termina en la extremidad Extrem/8. Todo el equipo está conectado aquí de manera natural, uno enseguida del otro. Los detalles del lenguaje GLang para expresar estos arreglos se darán en la sección titulada "Glang: el lenguaje de objetos gráficos".

### GLang: el lenguaje de objetos gráficos

La parte medular de este trabajo la constituye el lenguaje GLang, diseñado y desarrollado para expresar gráficamente la disposición del equipamiento en trenes de descarga de pozos de bombeo.

Un lenguaje, que es el dispositivo para la expresión de algo, se puede considerar como un conjunto enumerable de símbolos y un conjunto de reglas para construir expresiones bien formadas del lenguaje; a este conjunto de reglas se le llama sintaxis del lenguaje. De manera simple, el significado de las expresiones del lenguaje surge cuando sus símbolos se hacen corresponder con objetos o individuos de algún mundo físico o abstracto; a la evaluación de este significado se le llama semántica del lenguaje. En los siguientes párrafos se describe la sintaxis y la semántica del lenguaje GLang.

#### Sintaxis del lenguaje

La definición sintáctica de un lenguaje establece las reglas para la construcción de sus expresiones bien formadas.

Para expresar la sintaxis del lenguaje *GLang* utilizaremos un formalismo de la familia de las gramáticas de producciones, y más específicamente de las gramáticas libres de contexto (véase Aho y Ullman, 1972; Aho y Ullman, 1979; Hopcroft y Ullman, 1979; Wulf *et al.*, 1981). Este formalismo, semejante al formalismo de reglas *Backus-Naur Form* (BNF) (véase Naur, 1960; Naur, 1963; Tracy y Bouthoorn, 1997), o al popular formalismo gramatical de *Prolog Definite Clause Grammar* (DCG) (véase Bratko, 1990), tiene las siguientes particularidades:

Las reglas gramaticales se construyen con cuatro tipos de elementos:

1. Delimitadores de la regla. Los delimitadores de la regla son únicamente la flecha ( $\rightarrow$ ) y el punto (.).
2. Número de orden. Es un número entero único al principio de cada regla.
3. Símbolos terminales. Dentro de una regla, estos símbolos se caracterizan por su nombre escrito en itálicas o entre apóstrofos. A estos símbolos también se les denomina categorías terminales.
4. Símbolos no terminales. Dentro de una regla, estos símbolos están dados por su nombre en caracteres romanos normales. A estos símbolos también se les denomina categorías sintácticas.

Cada regla tiene la siguiente estructura:

$n \text{ LIR} \rightarrow \text{LDR} .$

donde  $n$  es un número entero que le da orden a la regla; LIR, el lado izquierdo de la regla, constituido por una sola categoría o símbolo no terminal; la flecha ( $\rightarrow$ ), un indicador que sólo sirve para delimitar los dos lados de la regla; LDR, el lado derecho de la regla, constituido por una secuencia de símbolos terminales y no terminales; el punto (.), un indicador del final de la regla.

Cada regla define la forma de construir un símbolo no terminal, el que aparece en el lado izquierdo de la regla. Si este símbolo aparece como el lado izquierdo de más de una regla, esto significa que el símbolo se puede construir de tantas maneras como reglas haya con el símbolo como lado izquierdo.

Los símbolos terminales son textualmente el carácter o caracteres entre apóstrofos, o, en el caso de nombres, en itálicas, la expansión convencional denotada por el nombre. Por ejemplo, el símbolo terminal *Number* denota todos los símbolos correspondientes a los números

enteros positivos; *GNumber* denota los símbolos correspondientes a los números reales; *ANSeq* denota los símbolos correspondientes a las secuencias alfanuméricas.

A continuación se dan las reglas gramaticales del lenguaje *GLang*:

Reglas gramaticales del lenguaje *GLang*

- 1 *SLSeq*  $\rightarrow$  *SLine SLSeq* .
- 2 *SLSeq*  $\rightarrow$  *SLine* .
- 3 *SLine*  $\rightarrow$  *IdLine SimSeq* ' ' .
- 4 *IdLine*  $\rightarrow$  *Number* ' : ' .
- 5 *SimSeq*  $\rightarrow$  *Sblock* ' ; ' *SimSeq* .
- 6 *SimSeq*  $\rightarrow$  *Sblock* .
- 7 *Sblock*  $\rightarrow$  *SimId Args* .
- 8 *Sblock*  $\rightarrow$  *SimId* .
- 9 *Args*  $\rightarrow$  ' ( ' *SimId* ' ) ' .
- 10 *Args*  $\rightarrow$  ' ( ' *NSeq* ' ) ' .
- 11 *NSeq*  $\rightarrow$  *Number* ' ; ' *NSeq* .
- 12 *NSeq*  $\rightarrow$  *Number* .
- 13 *SimId*  $\rightarrow$  *SName* ' / ' *Number* .
- 14 *SName*  $\rightarrow$  *ANSeq* ' { ' *ArgsList* ' } ' .
- 15 *SName*  $\rightarrow$  *ANSeq* .
- 16 *ArgsList*  $\rightarrow$  *Arg* ' ; ' *ArgsList* .
- 17 *ArgsList*  $\rightarrow$  *Arg* .
- 18 *Arg*  $\rightarrow$  *GNumber* .
- 19 *Arg*  $\rightarrow$  *Number* .
- 20 *Arg*  $\rightarrow$  *SName* .

Como un ejemplo de la interpretación de este lenguaje se propone la siguiente expresión válida del lenguaje:

0: Bmb/1, Tubo/2, Med/3, Tubo/4, Codo45/5, Carrete/6, Codo45b/7, Extrem/8.

Como podrá notarse, esta expresión es la misma que aparece en el primer renglón de la tabla que aparece en el cuadro 2. La ilustración 4 muestra la interpretación gráfica de esta expresión, y en la ilustración 5 se muestra una parte del resultado del análisis de la estructura de la frase dada en esa expresión; la parte que corresponde al segmento: "0: Bmb/1, Tubo/2...". Reglas gramaticales como las mostradas siempre desarrollan la estructura de una frase como un árbol. Cada regla desenvuelve un nivel del árbol en el que su cabeza o raíz está dada por el lado izquierdo de la regla, y sus ramas, por cada uno de los elementos en el lado derecho de la regla. Por ejemplo, en la ilustración 5, el primero y segundo nivel del árbol (de arriba a abajo) están definidos por la regla gramatical número 2; esto es, una ramificación de una sola rama: *SLSeq* como cabeza y *SLine* como única rama.

Ilustración 4. Interpretación gráfica de una expresión del lenguaje GLang.

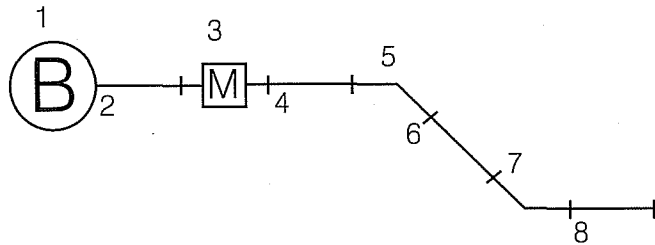
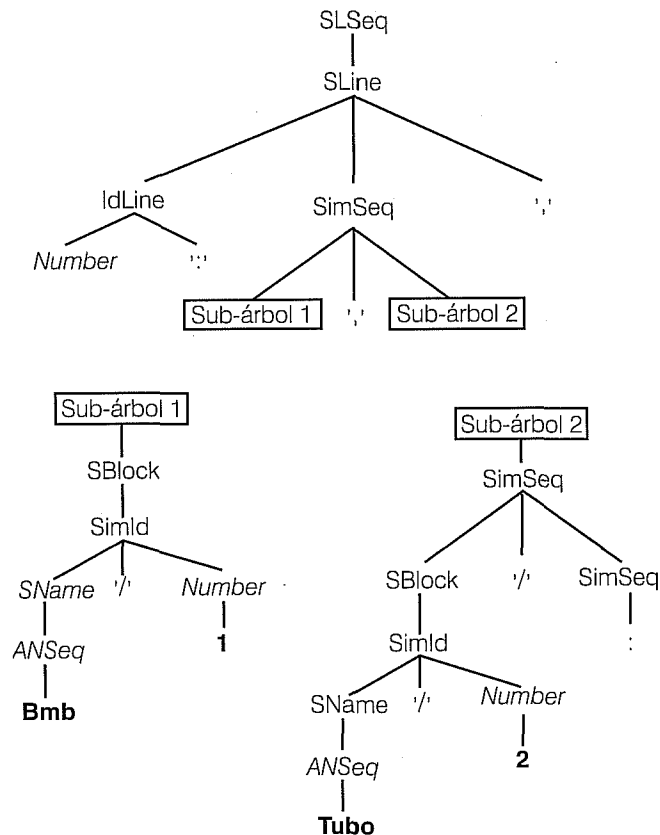


Ilustración 5. Fragmento de un árbol de análisis de la estructura de una frase del lenguaje GLang.



Los niveles 2 y 3 corresponden, de manera similar, a la regla número 3, donde la cabeza es SLine, y con tres ramas: IdLine, SimSeq, y ':'.

Para realizar el análisis de la estructura de frases de este lenguaje se desarrolló un parser tipo *chart* de arriba hacia abajo (*top-down*) siguiendo las estrategias descritas

por Gazdar y Mellish (1989) y por Allen (1995); los principios de esta técnica fueron originalmente desarrollados por Kay (1973; 1980/1986).

Para facilitar la definición de la semántica del lenguaje, en la sección "Semántica del lenguaje" se introduce una notación que permite describir árboles sintácticos como el de la ilustración 5.

#### Convención notacional 1

- El árbol vacío se representa con  $[\ ]$ .
- Un árbol con raíz o cabeza  $a$ , y ramas  $\beta_1, \beta_2, \dots, \beta_n$ , se representa con  $\alpha[\ \beta_1, \beta_2, \dots, \beta_n \ ]$ .
- Un árbol con raíz o cabeza  $\alpha$ , y cuyas ramas no se necesita o no se quiere mencionarlas, se representa con  $\alpha[\ \dots \ ]$ .
- La linealización de un árbol,  $T$ , se denota por  $\ell(T)$  y consiste de la concatenación de todas las hojas (extremos del árbol) encontradas durante un recorrido en orden del primero más profundo (*depth-first order*) del árbol.
- El árbol de estructura de frase dado por el análisis sintáctico de la frase  $p$  de acuerdo con la categoría  $Cat$ , se denota por  $\Lambda(Cat)(p)$ .

Por ejemplo, el subárbol 2 de la ilustración 5 se representaría con:

subárbol 2 =  $[\text{SBlock}[\text{SimId}[\text{SName}[\text{ANseq}[\ ]], \text{'/'}, \text{Number}[\mathbf{2}]], \text{'/'}, \text{SimSeq}[\dots]]]$ , y la linealización de este árbol sería:

$$\ell(\text{subárbol 2}) = \text{"Tubo/2, \dots"}$$

El subárbol 1 de la ilustración 5 resulta del análisis sintáctico de la frase "Bmb / 1" de acuerdo con la categoría sintáctica SBlock. Simbólicamente, esto lo expresamos como:

$$\Lambda(\text{SBlock})(\text{Bmb} / 1) = \text{SBlock} [\text{SimId} [\text{SName} [\text{ANSeq} [\mathbf{Bmb}]], \text{'/'}, \text{Number} [\mathbf{1}]]]$$

Nótese que las operaciones  $\ell$  y  $\Lambda$  son inversas; esto es,  $\ell(\Lambda(Cat)(p)) = p$ , para cualquier categoría  $Cat$ .

#### Semántica del lenguaje

Usualmente para una descripción formal de la semántica de cualquier lenguaje se emplean tratamientos como los de Montague (Dowty *et al.* 1981; Santana, 1999), o bien, en el área de los lenguajes de computación, la semántica denotacional (Gordon, 1979;



Tennent, 1991). La descripción simplificada que se brinda en los siguientes párrafos está inspirada en esos dos enfoques.

$$\text{cond}(d_1, d_2) t = \begin{cases} d_1, & \text{si } t = \text{true} \\ d_2, & \text{si } t = \text{false} \end{cases}$$

### Notación y definiciones

En este apartado introduciremos algunas definiciones y notación que se estarán utilizando en la descripción semántica que se da más adelante.

### Secuencias

#### Convención notacional 2

Sea  $D$  un dominio cualquiera. La notación  $D^*$  se usa para indicar el dominio de todas las secuencias finitas de elementos de  $D$ , esto es:

$$D^* = \{ () \} \cup D \cup [D \times D] \cup [D \times D \times D] \cup \dots$$

#### Definición 1

Si  $d \in D^*$  y  $d = (d_0, d_1, \dots, d_i, \dots, d_n)$ , entonces definimos la función  $\varepsilon_i : D^* \rightarrow D$  de la manera siguiente:

$$\varepsilon_i(d) = d_i.$$

Con  $d \in D$ , también para expresar esta función se adoptará la siguiente notación:

$$d[i] = d_i.$$

### Asignaciones

#### Convención notacional 3

Sea  $g : U \rightarrow D$  una función que asigna a cada elemento del dominio  $U$  un elemento del dominio  $D$ . La notación  $g^{u/d}$  se usa para indicar un valor de asignación exactamente como el definido por  $g$ , excepto que asigna el elemento  $d \in D$  al elemento  $u \in U$ . De manera semejante, sea  $g_0 = g^{u_0/d_0, u_1/d_1, \dots, u_n/d_n}$ , esto indica un valor de asignación exactamente como el definido por  $g$ , excepto que:

$$g_0(u_0) = d_0, g_0(u_1) = d_1, \dots, g_0(u_n) = d_n$$

### Condicional

#### Definición 2

Sea  $\text{cond} : [D \times D] \rightarrow T \rightarrow D$ , donde  $D$  es un dominio arbitrario, y  $T = \{\text{false}, \text{true}\}$ , la función definida por:

Para expresar esta función también se utilizará la siguiente notación:  $t \rightarrow d_1, d_2$ , que significa  $\text{cond}(d_1, d_2) t$ .

### Operador de secuenciación

#### Definición 3

Sean:

$f : [Tx(D_1 \cup \{\text{indef}\})] \rightarrow [Tx(D_2 \cup \{\text{indef}\})]$  y  $g : [Tx(D_2 \cup \{\text{indef}\})] \rightarrow [Tx(D_3 \cup \{\text{indef}\})]$  dos funciones, donde  $D_1, D_2$ , y  $D_3$  son dominios arbitrarios, y  $T = \{\text{false}, \text{true}\}$ . Definimos el operador  $\circ$  entre  $f$  y  $g$ , de la siguiente manera:  $f \circ g : [Tx(D_1 \cup \{\text{indef}\})] \rightarrow [Tx(D_3 \cup \{\text{indef}\})]$  es una función que está dada por  $f \circ g = \lambda(t, x) \bullet \varepsilon_0(f(t, x)) \rightarrow g(f(t, x))$ , ( $\text{false}, \text{indef}$ ).

Para definir funciones estamos utilizando el cálculo lambda estándar, desarrollado originalmente por Alonzo Church (Church, 1940). Una breve introducción al cálculo lambda se puede encontrar en Gordon (1979) o en Dowty *et al.* (1981). Note que todas las funciones involucradas aquí incluyen la adición del conjunto  $\{\text{indef}\}$ . Esto se hace para poder asignar en el resultado de la función un valor indefinido cuando no se requiere o no se puede calcular el valor.

### Concatenación

En algunas ocasiones es necesario dejar los símbolos gramaticales tal cual vienen para pasar su interpretación a algún otro componente del sistema. Con este propósito se define aquí la operación de concatenación de símbolos gramaticales.

#### Definición 4

Sean  $\alpha$  y  $\beta$  dos símbolos gramaticales cualquiera, entonces  $\alpha \parallel \beta$  es otro símbolo gramatical formado a partir de la concatenación de  $\alpha$  y  $\beta$ , esto es:  $\alpha \parallel \beta = \alpha\beta$ .

### Estado de interpretación

Los croquis que describen las configuraciones de los trenes de descarga de pozos de bombeo viven o se representan en un espacio geométrico bidimensional. Para ir colocando los distintos objetos gráficos en este espacio introducimos aquí el concepto de estado de interpretación, que es un conjunto de variables o pará-

metros que describen la disposición de las cosas en un momento de la interpretación.

Antes de proseguir con la descripción del estado de interpretación es importante señalar que la inserción de objetos tales como una Tee o una Cruz en el tren de descarga introducen ramificaciones secundarias de éste. A esas ramificaciones las llamaremos líneas de inserción. Un ejemplo de esto se puede apreciar en la ilustración 6.

Las variables que constituyen el estado de interpretación son las siguientes:

**PDot[]**  $\in (\mathfrak{R}^2)^*$ . Es un arreglo de tantos puntos como líneas de inserción haya. Cada uno de estos puntos indica el sitio de inserción del siguiente objeto gráfico en la línea correspondiente.

**Angle[]**  $\in \mathfrak{R}^*$ . Es un arreglo de tantos números reales (ángulos) como líneas de inserción haya. Cada uno de estos ángulos indica la orientación a la que se debe insertar el siguiente objeto gráfico en la línea correspondiente.

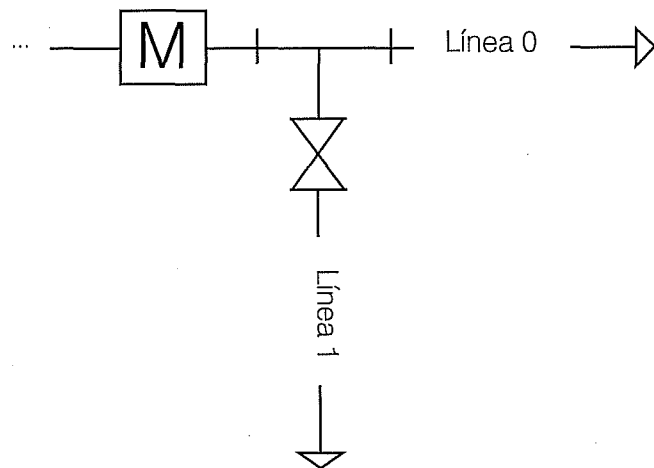
**IndLine**  $\in \mathfrak{X}$ . Un entero usado para identificar la línea de inserción en la que se están insertando los objetos gráficos en un momento de interpretación dado.

**nLabel**  $\in \mathfrak{X}$ . Un número entero que funge como etiqueta del símbolo que se está insertando en un momento de interpretación dado.

**IDef**  $\in \mathfrak{X}$ . Es un número entero que servirá para identificar elementos en los arreglos **PDot[]** y **Angle[]**.

**OnInserted**  $\in T$ . Una variable booleana que indica si el objeto gráfico que se está insertando es parte de la línea de inserción (*false*) o va insertado como parte del objeto gráfico que se insertó anteriormente (*true*).

Ilustración 6. Líneas de inserción en el croquis del tren de descarga.



**LastSymb**  $\in \mathfrak{S}$ . Identificador del último símbolo u objeto gráfico insertado.

**Sketch**  $\in \mathfrak{R}^2$ . Es el croquis propiamente dicho. De manera simple, es un subconjunto del espacio euclidiano bidimensional que contiene la descripción gráfica de los símbolos, abstraída del espacio físico para brindar una idea del arreglo del equipo en el tren de descarga.

Llamemos al conjunto de los identificadores de estas variables **IdEd**, que entonces está definido como:

**IdEd** = { **PDot[]**, **Angle[]**, **IndLine**, **nLabel**, **IDef**, **OnInserted**, **LastSymb**, **Sketch** }.

Entonces, de manera formal, el espacio de los estados de interpretación estará dado como:

$$\text{IntEdo} = (\mathfrak{R}^2)^* \times \mathfrak{R}^* \times \mathfrak{X} \times \mathfrak{X} \times \mathfrak{X} \times T \times \mathfrak{S} \times P(\mathfrak{R}^2) + \{\text{indef}\}$$

donde  $\mathfrak{R}$  es el conjunto de los números reales;  $\mathfrak{X}$ , el conjunto de los números enteros positivos;  $T$ , el conjunto de los valores booleanos:  $T = \{\text{false}, \text{true}\}$ , y  $\mathfrak{S}$  es el conjunto de los identificadores de los símbolos u objetos gráficos:  $\mathfrak{S} = \{\text{Med}, \text{CruzB}, \text{Codo45}, \text{Codo45b}, \text{Ampl}, \text{Atrq}, \text{Bmb}, \text{Carrete}, \text{Enterr}, \text{JMec}, \dots\}$ . La notación  $A^*$  se refiere a todas las secuencias finitas de elementos de cualquier conjunto  $A$ , y  $P(A)$  se refiere al conjunto potencia de cualquier conjunto  $A$ ; esto es, el conjunto de todos sus subconjuntos. Se ha agregado aquí un elemento *indef* al conjunto de estados posibles de interpretación para manejar un posible estado indefinido.

Para los propósitos de lo que describiremos más adelante, definamos el conjunto **EltEd** de la siguiente forma:

$$\text{EltEd} = (\mathfrak{R}^2)^* \times \mathfrak{R}^* + \mathfrak{X} + T + \mathfrak{S} + P(\mathfrak{R}^2)$$

donde el símbolo  $+$  representa la unión disjunta de conjuntos.

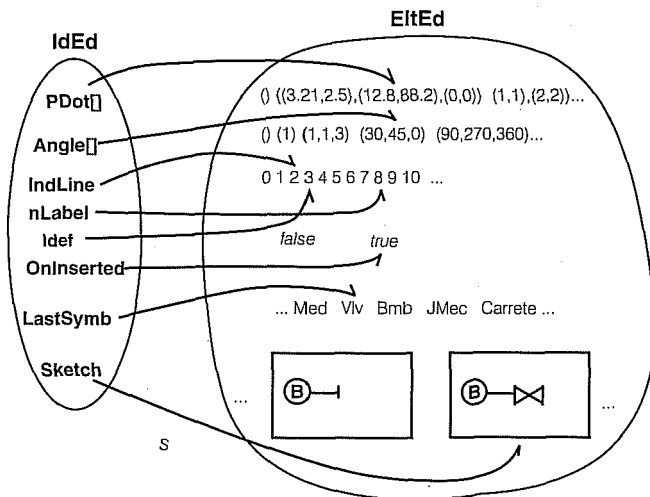
Un estado  $s \in \text{IntEdo}$ , puede ser visto como un punto en el espacio **IntEdo** o alternativamente como una función de asignación,  $s: \text{IdEd} \rightarrow \text{EltEd}$ , que asigna a cada uno de los elementos de **IdEd** un único elemento de **EltEd**. Un ejemplo de este tipo de funciones se puede ver en la ilustración 7.

Para el caso de los identificadores que se asocian con secuencias, por ejemplo **PDot[]** y **Angle[]**, de acuerdo con la definición 1, se introduce aquí también la siguiente convención notacional:

Convención notacional 4

Si **X[]** o **X** es un identificador que una función de estado

Ilustración 7. Un estado de interpretación  $s$ , como función.



de interpretación,  $s$ , pone en correspondencia con una secuencia de elementos  $d = (d_0, d_1, \dots, d_n, \dots, d_n)$ , entonces  $\mathbf{X}[i]$  es el identificador que la misma función  $s$ , pone en correspondencia con el  $i$ -ésimo elemento de la secuencia  $d_i$ .

Entonces, de acuerdo con la definición 1,  $s(\mathbf{X}[i]) = \varepsilon_i(s(\mathbf{X})) = s(\mathbf{X})[i]$ .

Sea la función  $s$  la que se encuentra definida en la ilustración 7, entonces:

- $s(\text{OnInserted}) = \text{true}$ , y
- $s(\text{LastSymb}) = \text{Vlv}$ , pero,
- $s_{\text{LastSymb/JMec}}(\text{LastSymb}) = \text{JMec}$ , aunque
- $s_{\text{LastSymb/JMec}}(\text{OnInserted}) = \text{true}$ .

Aparte de esto, existe un área a la que llamaremos **Mem** y que contiene una descripción gráfica de cada uno de los símbolos u objetos gráficos disponibles para ser insertados como parte de **Sketch: Mem**  $\subseteq P(\mathbb{R}^2)$ .

A estas descripciones se puede tener acceso mediante la función:

$$\text{mem}: \mathcal{S} \rightarrow \text{Mem}$$

De una manera práctica, la descripción de estos objetos gráficos se da mediante un catálogo de símbolos como el que se muestra en el listado 1. La descripción gráfica de las instrucciones dadas en este archivo se muestra en la ilustración 8 para cada uno de los objetos gráficos.

El catálogo de símbolos (listado 1) registra, entre otras cosas, la descripción gráfica de cada uno de los símbolos con significado dentro del sistema. En el fragmento mostrado se exhibe la descripción de tres símbolos: el

medidor, la cruz bridada, y el codo a 45°. La descripción de cada símbolo empieza con el nombre del símbolo y termina con un punto aislado (.) en un renglón. Después de esto vienen los puntos y ángulos significativos del objeto gráfico que pueden representarse como vectores de dirección en  $\mathbb{R}^2$ , con el formato general: 'D(X,Y), $\alpha$ ', donde X, Y son las coordenadas del punto y el ángulo,  $\alpha$ , asociado con el punto, se puede omitir, en cuyo caso toma un valor cero (0). El primero de estos puntos sólo se usa para indicar el espacio en el que se definirá el símbolo; en todos los casos mostrados, éste es un espacio de 10 x 10 unidades. La ilustración 8 muestra las gráficas correspondientes a los tres símbolos, cuya descripción se tiene en el fragmento de archivo del listado 1. En esta ilustración, los puntos de interés se muestran secuencialmente con las letras *a, b, c, ...*, también se muestran los ángulos de interés. El punto marcado con la letra *t* es el punto de inserción de etiquetas. Para tener acceso a esos puntos y ángulos de interés definiremos dos funciones en el listado 1.

#### Definición 5

**Dot:**  $[\text{Mem} \times \mathcal{K}] \rightarrow \mathbb{R}^2$  es una función que toma un símbolo u objeto gráfico, **sg**, y un número entero,  $i$ , y entrega el  $i$ -ésimo punto de interés del objeto gráfico **sg**. Esto es, **Dot(sg, i)** es el  $i$ -ésimo punto de interés del objeto **sg**. **Ang:**  $[\text{Mem} \times \mathcal{K}] \rightarrow \mathbb{R}$  es una función que toma un identificador de un símbolo u objeto gráfico, **sg**, y un número entero,  $i$ , y entrega el  $i$ -ésimo ángulo de interés del objeto gráfico **sg**. Esto es, **Ang(sg, i)** es el  $i$ -ésimo ángulo de interés del objeto **sg**.

Después del último punto de interés, el punto de inserción de etiquetas ( $t$ ), se da la descripción gráfica del símbolo propiamente dicha. Ésta, por ejemplo, puede contener una o más de las siguientes instrucciones:

1. Una secuencia de líneas discontinuas: 'L{(X<sub>0</sub>,Y<sub>0</sub>), (X<sub>1</sub>,Y<sub>1</sub>), ..., (X<sub>n</sub>,Y<sub>n</sub>)}' , donde cada par de puntos representa un segmento de línea recta.
2. Una polilínea, formada por líneas una a continuación de otra: 'p{(X<sub>0</sub>,Y<sub>0</sub>), (X<sub>1</sub>,Y<sub>1</sub>), ..., (X<sub>n</sub>,Y<sub>n</sub>)}'.
3. Un polígono cerrado: 'P{(X<sub>0</sub>,Y<sub>0</sub>), (X<sub>1</sub>,Y<sub>1</sub>), ..., (X<sub>n</sub>,Y<sub>n</sub>)}'.
4. La inserción de un carácter: 'I{C, (X<sub>0</sub>,Y<sub>0</sub>), (X<sub>1</sub>,Y<sub>1</sub>)}' , donde C es el carácter a insertar, (X<sub>0</sub>,Y<sub>0</sub>) y (X<sub>1</sub>,Y<sub>1</sub>) son el punto de inserción y el tamaño del carácter, respectivamente.

De esa forma quedan pues descritos los símbolos gráficos en el área denominada **Mem**, y entonces, por ejemplo, la expresión:  $\text{mem}(\text{Med})$  es la descripción gráfica

Listado 1. Fragmento de un catálogo de símbolos.

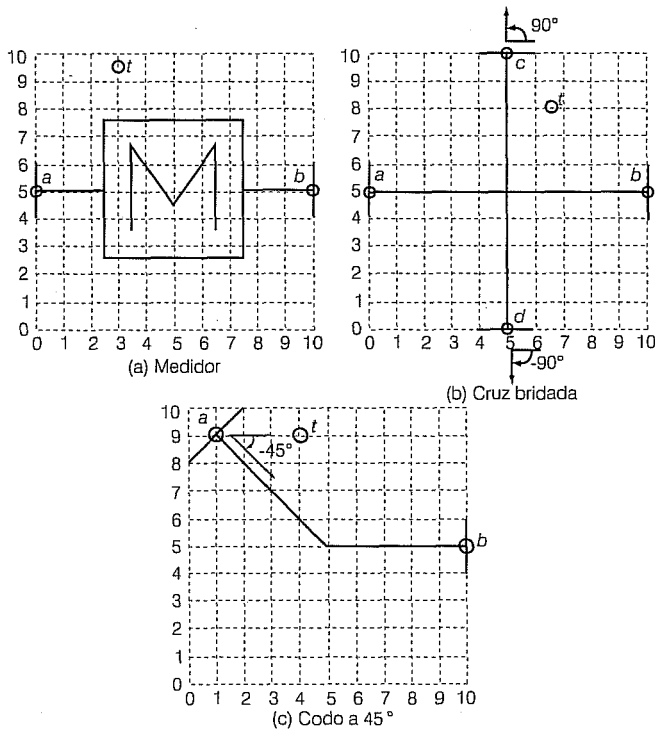
Med				
	D(10,10)			
	D(0,5)			
	D(10,5)			
	t (3,9 .5)			
	L {(0, 4), (0, 6), (0, 5), (2.5, 5)}			
	P {(2. 5,2.5),(7. 5,2.5),(7. 5,7.5), (2.5 , 7.5) }			
	w{2}			
	f {M,(3.5,3. 5),(3,3)}			
	w{1}			
	L {(7. 5,5), (10,5) ,(10, 4),(10,6)}			
##	Medidor			
##	Tipo	Tipo	Tx	20 Propela 'Placa de Orificio' Ultrasónico
##	Marca	Marca	Tx	20
##	Tiempo	'Tiempo colocación'	N	10
##	Func	'¿Funciona?'	Tx	5 No Sí
##	Falla	'Tipo Falla'	Tx	30
##	Diam	'Diámetro'	N	10
##	Pres	'Presión de Trabajo'	N	10
##	Costo	Costo	N	10
CruzB				
	D(10,10)			
	D(0, 5)			
	D(10, 5)			
	D(5, 10),90			
	D(5, 0),-90			
	t (6.5, 8)			
	L {(0, 4), (0, 6), (0, 5), (10, 5), (10, 4), (10, 6), (5,10), (5, 0), (4, 0), (6,0), (4,10), (6, 10) }			
##	Cruz Bridada			
##	Mat	Material	Tx	20
##	Diam 1	'Diámetro 1'	N	10
##	Diam 2	'Diámetro 2'	N	10
##	Pres	'Presión de Trabajo'	N	10
##	Costo	Costo	N	10
Codo45b				
	D(10, 10)			
	D(1, 9),- 45			
	D(10, 5)			
	t (4, 9)			
	L {(0, 8), (2, 10), (10, 4), (10 6) }			
	p {(1, 9), (5, 5), (10, 5) }			
##	Codo a 45 grados			
##	Mat	Material	Tx	20
##	Diam	Diámetro	N	10
##	Pres	'Presión de Trabajo'	N	10
##	Costo	Costo	N	10

del símbolo cuyo identificador es Med, y que corresponde, en este caso, con lo mostrado en la ilustración 8(a). Es decir:

$$\text{mem}(\text{Med}) = \boxed{\text{M}}$$

El cuadro 3 muestra una tabla con una parte de lo que sería esta función: los valores del argumento en la columna etiquetada con "Identificador", y el resultado correspondiente a cada valor del argumento, en la columna etiquetada con "Símbolo".

Ilustración 8. Definición de algunos objetos gráficos.



Cuadro 3. Algunos símbolos gráficos.

Identificador	Símbolo	Significado
Ampl		Ampliación
Atrq		Atraque
Bmb		Bomba
Carrete		Carrete
Codo45		Codo a 45°
Enterr		Continúa conducción enterrada
JMec		Junta mecánica
Med		Medidor

Como se ha visto en los párrafos anteriores, estos símbolos u objetos gráficos están descritos en un espacio virtual normalizado, usado únicamente para su descripción. Para introducir alguno de estos objetos al croquis (**Sketch**) de un estado de interpretación se requiere de transformaciones geométricas que lo coloquen en el sitio, con el tamaño, y la orientación adecuados. Para un estado de interpretación dado, esta transformación involucra prácticamente todas sus variables: **PDot[]**, **Angle[]**, **IndLine**, etcétera. Para no ahondar en muchos detalles, aquí solamente simbolizaremos con  $\tau$  todas las transformaciones necesarias, así:  $\tau(s, id)$ , donde  $s \in \text{IntEdo}$ , e  $id \in \mathcal{S}$  produce el símbolo correspondiente a  $id$ , pero en la disposición requerida en un momento dado por el estado  $s$ .

Reglas semánticas

De igual manera que la sintaxis de un lenguaje se define mediante reglas, la semántica se define mediante reglas que corresponden una a una con las reglas sintácticas. En los siguientes párrafos se dan estas reglas.

Cada una de las reglas sintácticas dadas en la sección "Sintaxis del lenguaje" define la manera de estructurar una expresión del lenguaje. Llamemos al valor semántico de cada una de estas expresiones su denotación. Entonces, hay una función,  $\Delta$ , que asocia cada expresión del lenguaje con su denotación de la siguiente manera:

$$\Delta: \text{Exp} \rightarrow [[\text{Val} \times \text{IntEdo}] \rightarrow [\text{Val} \times \text{IntEdo}]]$$

donde **Exp** es el conjunto de todas las expresiones del lenguaje que pueden ser generados con las reglas gramaticales; **IntEdo**, el conjunto de posibles estados de interpretación, tal como se definió en la sección "Estado de interpretación"; **Val** es el conjunto de valores posibles definido así:  $\text{Val} = \mathcal{T} \cup \mathcal{R} \cup \mathcal{N}$ . Entonces  $\Delta$  es una función semántica que asocia cada expresión del lenguaje con otra función, que asocia cada pareja de un valor y un estado de interpretación posible con otra pareja de un valor y un estado de interpretación posible, esto es,  $\Delta[e](v, s) = (v_o, s_o)$  es la denotación (o valor semántico) de la expresión e con un valor de entrada v en el estado de interpretación s, que resultará, de manera general, en un valor de salida y un nuevo estado de interpretación  $(v_o, s_o)$ . Conviene aquí definir una función que permite de manera simple generar un estado a partir de otro estado dado:

Definición 6

La función de cambio de estado,  $\varphi: (\text{IdED} \times \text{ElEd})^* \rightarrow [[\text{Val} \times \text{IntEdo}] \rightarrow [\text{Val} \times \text{IntEdo}]]$ , se define como:

$$\varphi = \lambda((i_0, a_0), (i_1, a_1), \dots, (i_n, a_n)) \bullet [\lambda(v, s) \bullet (v \neq \text{false}) \rightarrow (v, s^{i/a_0 i_1 / a_1 \dots i_n a_n}), (\text{false}, \text{indef})].$$

Entonces, por ejemplo, si se aplica la función  $\varphi(i, a)(v, s)$ , se obtiene:  $\varphi(i, a)(v, s) = (v, s^{i/a})$  si  $v \neq \text{false}$ . Esto es, dado un estado inicial,  $s$ , el identificador  $i \in \text{IdEd}$ , y el valor  $a \in \text{EltEd}$ , la función  $\varphi$  genera un nuevo estado,  $s_0$ , igual en todo a  $s$ , excepto que ahora el identificador  $i$  tiene un valor  $a$ , o sea  $s_0(i) = a$ .

La definición de  $\Delta$  se hace mediante reglas semánticas, empezando con una regla que permitirá interpretar los elementos léxicos, que son a los que nos hemos referido como símbolos terminales en los párrafos anteriores. Estas reglas son:

- **Regla 0 (elementos léxicos):** Si un subárbol  $\gamma = \alpha[\beta]$  es introducido por una categoría léxica  $\alpha$ , por ejemplo *ANSeq*, *GNumber*, o *Number*, y  $\beta$  es una constante léxica (esto es una constante del lenguaje GLang de la categoría  $\alpha$ ), entonces  $\Delta[\gamma](v, s) = \Delta[\beta](v, s)$ . La tabla del cuadro 4 muestra el significado de algunas de las constantes léxicas y su significado. Debe notarse que el significado está dado como un par que incluye un valor y el mismo estado de entrada; esto es, para todos estos casos, el estado de interpretación no se modifica.
- **Regla 1:** (derivada de  $1 \text{ SLSeq} \rightarrow \text{SLine SLSeq}$ .) Si  $\alpha$  es *SLine* [...],  $\beta$  es *SLSeq* [...], y  $\gamma$  es *SLSeq* [ $\alpha$ ,  $\beta$ ] (esto es,  $\gamma$  es el árbol *SLSeq* [*SLine* [...], *SLSeq* [...]]), entonces:

$$\Delta[\gamma](v, s) = (\Delta[\alpha] \circ \Delta[\beta])(v, s).$$

O bien, simplificando,  $\Delta[\gamma] = (\Delta[\alpha] \circ \Delta[\beta])$ .

Ejemplo

- Del análisis de la expresión "0:Bmb/1, TeeB/2(1) . 1: Vlv/3 ." surge el árbol  $\gamma = \text{SLSeq} [\Delta(\text{SLine})("0:\text{Bmb}/1, \text{TeeB}/2(1)."), \Delta(\text{SLSeq})("1:\text{Vlv}/3.")]$ . Su denotación está dada por:  $\Delta[\gamma](v, s) = (\Delta[\Delta(\text{SLine})("0:\text{Bmb}/1, \text{TeeB}/2(1).)] \circ \Delta[\Delta(\text{SLSeq})("1:\text{Vlv}/3.")])(v, s)$ .
- **Regla 2:** (derivada de  $2 \text{ SLSeq} \rightarrow \text{SLine}$ .) Si  $\alpha$  es *SLine* [...], y  $\gamma$  es *SLSeq* [ $\alpha$ ], entonces  $\Delta[\gamma](v, s) = (\Delta[\alpha])(v, s)$ , o lo que es lo mismo,  $\Delta[\gamma] = \Delta[\alpha]$ .

Ejemplo

Del análisis de la expresión "1: Vlv/3" surge el árbol  $\gamma = \text{SLSeq} [\Delta(\text{SLSeq})("1:\text{Vlv}/3.")]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = \Delta[\Delta(\text{SLSeq})("1:\text{Vlv}/3.")](v, s)$ .

- **Regla 3:** (derivada de  $3 \text{ SLine} \rightarrow \text{IdLine SimSeq '.'$ .) Si  $\alpha$  es *IdLine* [...],  $\beta$  es *SimSeq* [...], y  $\gamma$  es *SLine* [ $\alpha$ ,  $\beta$ , '.'], entonces  $\Delta[\gamma](v, s) = (\Delta[\alpha] \circ \Delta[\beta])(v, s)$ , o lo que es lo mismo  $\Delta[\gamma] = (\Delta[\alpha] \circ \Delta[\beta])$ .

Ejemplo

Del análisis de la expresión "0:Bmb/1, TeeB/2(1) ." surge el árbol  $\gamma = \text{SLine} [\Delta(\text{IdLine})("0:"), \Delta(\text{SimSeq})("Bmb/1, TeeB/2(1)"), '[:']]$ . Su denotación está dada por:  $\Delta[\gamma](v, s) = (\Delta[\Delta(\text{IdLine})("0:")] \circ \Delta[\Delta(\text{SimSeq})("Bmb/1, TeeB/2(1)"))](v, s)$ .

Estrictamente, cada símbolo de puntuación ('.', ':', '::', etcétera), introduce un árbol de una sola rama:  $\mathbf{s}[\mathbf{s}]$ , donde  $\mathbf{s}$  es cualquier símbolo de puntuación. Para abreviar, ese árbol lo representaremos aquí solamente

Cuadro 4. El significado de algunas constantes léxicas.

Categoría ( $\alpha$ )	Descripción	Ejemplos de constantes ( $\beta$ )	Denotación o significado
<i>GNumber</i>	Los números reales	3.1416 -18.25 0000.032	(3.1416, s) (-18.25, s) (0.032, s)
<i>Number</i>	Los números enteros positivos	0 25 88	(0, s) (25, s) (88, s)
<i>ANSeq</i>	Secuencias alfanuméricas	Codo45 CruzB Otro5Cualquiera	(Codo45, s) (CruzB, s) (Otro5Cualquiera, s)

con **s**. Por ejemplo, el subárbol '!', que se ha usado en este ejemplo, lo podremos representar simplemente con '!'.

- **Regla 4:** (derivada de 4 IdLine  $\rightarrow$  Number '!') Si  $\alpha$  es Number [...], y  $\gamma$  es IdLine [ $\alpha$ , '!'], entonces  $\Delta[\gamma](v, s) = (v_0, s_0)$ , donde  $v_0 = true$ , y  $s_0 = s^{IndLine/\epsilon_0(\Delta[\alpha](v, s))}$ . Debe notarse aquí que si  $v = false$ , entonces  $(v_0, s_0) = (false, indef)$  cualquiera que sea el valor de  $\alpha$ . La aplicación de esta regla significa que hay un cambio del estado de  $s$  al estado  $s_0$ , donde a la variable **IndLine** se le asigna el valor  $\Delta[\gamma](v, s)$ , que es la denotación de  $\alpha$  en el estado  $s$ .

Ejemplo

Del análisis de la expresión "0:" surge el árbol  $\gamma = IdLine [Number[0], '!']$ . Su denotación está dada por:  $\Delta[\gamma](v, s) = (true, s^{IndLine/\epsilon_0(\Delta[Number][0](v, s))}) = (true, s^{IndLine/0})$ . Esto es, en el estado de interpretación resultante, aplicando además la regla 0 referente a elementos léxicos, la variable **IndLine** tendrá un valor cero (0):  $s_0(IndLine) = 0$ .

- **Regla 5:** (derivada de 5 SimSeq  $\rightarrow$  Sblock '!', SimSeq) Si  $\alpha$  es Sblock [...],  $\beta$  es SimSeq [...], y  $\gamma$  es SimSeq [ $\alpha$ , '!',  $\beta$ ], entonces  $\Delta[\gamma](v, s) = (\Delta[\alpha] \circ \Delta[\beta])(v, s)$ , o lo que es lo mismo  $\Delta[\gamma] = (\Delta[\alpha] \circ \Delta[\beta])$ .

Ejemplo

Del análisis de la expresión "0:Bmb/1, TeeB/2(1)" surge el árbol  $\gamma = SimSeq[\Lambda(Sblock)("Bmb/1"), '!', \Lambda(SimSeq)("TeeB/2(1)")]$ . Su denotación está dada por:  $\Delta[\gamma](v, s) = (\Delta[\Lambda(Sblock)("Bmb/1")] \circ \Delta[\Lambda(SimSeq)("TeeB/2(1)"])(v, s)$ .

- **Regla 6:** (derivada de 6 SimSeq  $\rightarrow$  Sblock) Si  $\alpha$  es Sblock [...], y  $\gamma$  es SimSeq [ $\alpha$ ], entonces  $\Delta[\gamma](v, s) = (\Delta[\alpha])(v, s)$ , o lo que es lo mismo,  $\Delta[\gamma] = \Delta[\alpha]$ .

Ejemplo

Del análisis de la expresión "TeeB/2(1)" surge el árbol  $\gamma = SimSeq[\Lambda(Sblock)("TeeB/2(1)")]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = \Delta[\Lambda(Sblock)("TeeB/2(1)"])(v, s)$ .

- **Regla 7:** (derivada de 7 Sblock  $\rightarrow$  SimId Args.) Si  $\alpha$  es SimId [...],  $\beta$  es Args [...], y  $\gamma$  es Sblock [ $\alpha$ ,  $\beta$ ], entonces  $\Delta[\gamma](v, s) = (\Delta[\alpha] \circ \Delta[\beta])(v, s)$ , o lo que es lo mismo  $\Delta[\gamma] = (\Delta[\alpha] \circ \Delta[\beta])$ .

Ejemplo

Del análisis de la expresión "TeeB/2(1)" surge el árbol  $\gamma = Sblock[\Lambda(SimId)("TeeB/2"), \Lambda(Args)("1")]$ . Su denotación está dada por:  $\Delta[\gamma](v, s) = (\Delta[\Lambda(SimId)("TeeB/2")] \circ \Delta[\Lambda(Args)("1")])(v, s)$ .

- **Regla 8:** (derivada de 8 Sblock  $\rightarrow$  SimId.) Si  $\alpha$  es SimId [...], y  $\gamma$  es Sblock [ $\alpha$ ], entonces  $\Delta[\gamma](v, s) = (\Delta[\alpha])(v, s)$ , o lo que es lo mismo,  $\Delta[\gamma] = \Delta[\alpha]$ .

Ejemplo

Del análisis de la expresión "Bmb/1" surge el árbol  $\gamma = Sblock[\Lambda(SimId)("Bmb/1")]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = \Delta[\Lambda(SimId)("Bmb/1"])(v, s)$ .

- **Regla 9:** (derivada de 9 Args  $\rightarrow$  ('SimId')'.) Si  $\alpha$  es SimId [...], y  $\gamma$  es Args ['(',  $\alpha$ , ')'], entonces  $\Delta[\gamma](v, s) = (\varphi(\mathbf{OnInserted}, true) \circ \Delta[\alpha] \circ \varphi(\mathbf{OnInserted}, false))(v, s)$ , o lo que es lo mismo  $\Delta[\gamma] = \varphi(\mathbf{OnInserted}, true) \circ \Delta[\alpha] \circ \varphi(\mathbf{OnInserted}, false)$  (véase la definición 6 acerca de la función de cambio de estado  $\varphi$ ). La aplicación de esta regla indica que la introducción del símbolo indicado por el identificador  $\alpha$  se hace, no en la línea principal de inserción, sino sobre el símbolo insertado anteriormente. Esta es la razón por la cual el identificador **OnInserted** cambia primero su valor a *true* y después de interpretar  $\alpha$ , cambia a *false*.

Ejemplo

Del análisis de la expresión "(Atrq/7)" surge el árbol  $\gamma = Args['(', \Lambda(SimId)("Atrq/7"), ')']$ . Su denotación está dada por  $\Delta[\gamma](v, s) = (\varphi(\mathbf{OnInserted}, true) \circ \Delta[\Lambda(SimId)("Atrq/7")] \circ \varphi(\mathbf{OnInserted}, false))(v, s)$ . Esto es, dado que el ataque al que se refiere el símbolo "Atrq/7" no será insertado como una parte de la línea principal del flujo, sino que más bien va como un accesorio adjunto a algún otro elemento principal del tren de descarga, se cambia la variable de estado **OnInserted** temporalmente a *true*, mientras se interpreta "Atrq/7", y luego se cambia nuevamente su valor a *false*.

- **Regla 10:** (derivada de 10 Args  $\rightarrow$  ('NSeq')'.) Si  $\alpha$  es NSeq [...], y  $\gamma$  es Args ['(',  $\alpha$ , ')'], entonces  $\Delta[\gamma](v, s) = (\varphi(\mathbf{IDef}, 2) \circ \Delta[\alpha])(v, s)$ , o lo que es lo mismo,  $\Delta[\gamma] = \varphi(\mathbf{IDef}, 2) \circ \Delta[\alpha]$ .

Ejemplo

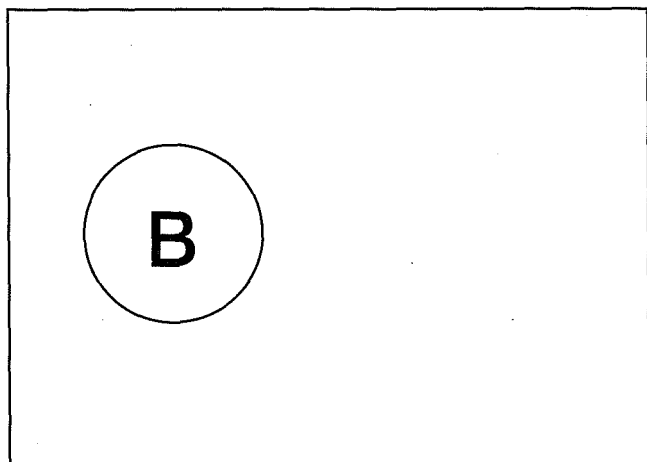
Del análisis de la expresión "(1)" surge el árbol  $\gamma = Args['(', \Lambda(NSeq)("1"), ')']$ . Su denotación está dada por  $\Delta[\gamma](v, s) = (\varphi(\mathbf{IDef}, 2) \circ \Delta[\Lambda(NSeq)("1")])(v, s)$ .

- **Regla 11:** (derivada de 11 NSeq  $\rightarrow$  Number '!', NSeq) Si  $\alpha$  es Number [...],  $\beta$  es NSeq [...], y  $\gamma$  es NSeq [ $\alpha$ , '!',  $\beta$ ], entonces  $\Delta[\gamma](v, s) = (\varphi(\mathbf{PDot}[\epsilon_0(\Delta[\alpha](v, s))], \mathbf{Dot}(\tau(s, \mathbf{LastSymb}), s(\mathbf{IDef}))), (\mathbf{Angle}[\epsilon_0(\Delta[\alpha](v, s))])$ .

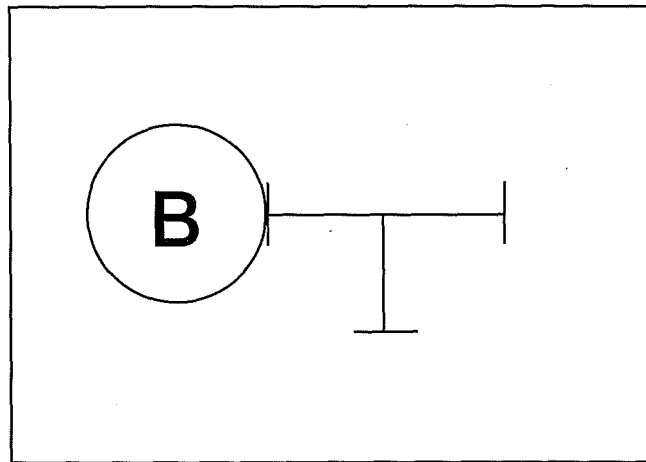




Ilustración 9. La interpretación de la regla semántica 13.



(a) Estado de interpretación inicial



(b) Estado de interpretación después de aplicar la regla 13

$\beta]$ , entonces  $\Delta[\gamma](v, s) ((\epsilon_0(\Delta[\alpha](v, s)) \parallel ' \parallel \epsilon_0(\Delta[\beta](v, s))), s)$ .

Ejemplo

Del análisis de la expresión "Vlv,7" surge el árbol  $\gamma = \text{ArgsList} [\Delta(\text{Arg})("Vlv"), ', \Delta(\text{ArgsList})("7")]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = ("Vlv,7", s)$ .

- **Regla 17:** (derivada de 17  $\text{ArgsList} \rightarrow \text{Arg}$ .) Si  $\alpha$  es  $\text{Arg}[\dots]$ , y  $\gamma$  es  $\text{ArgsList}[\alpha]$ , entonces  $\Delta[\gamma](v, s) = (\epsilon_0(\Delta[\alpha](v, s)), s)$ .

Ejemplo

Del análisis de la expresión "7" surge el árbol  $\gamma = \text{ArgsList} [\text{Arg} [\text{Number} [7]]]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = (7, s)$ .

- **Regla 18:** (derivada de 18  $\text{Arg} \rightarrow \text{GNumber}$ .) Si  $\alpha$  es  $\text{GNumber}[\dots]$ , y  $\gamma$  es  $\text{Arg}[\alpha]$ , entonces  $\Delta[\gamma](v, s) = (\alpha, s)$ .

Ejemplo

Del análisis de la expresión "7.35" surge el árbol  $\gamma = \text{Arg} [\text{GNumber} [7.35]]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = (7.35, s)$ .

- **Regla 19:** (derivada de 19  $\text{Arg} \rightarrow \text{Number}$ .) Si  $\alpha$  es  $\text{Number}[\dots]$ , y  $\gamma$  es  $\text{Arg}[\alpha]$ , entonces  $\Delta[\gamma](v, s) = (\alpha, s)$ .

Ejemplo

Del análisis de la expresión "7" surge el árbol  $\gamma = \text{Arg} [\text{Number} [7]]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = (7, s)$ .

- **Regla 20:** (derivada de 20  $\text{Arg} \rightarrow \text{SName}$ .) Si  $\alpha$  es  $\text{SName}[\dots]$ , y  $\gamma$  es  $\text{Arg}[\alpha]$ , entonces  $\Delta[\gamma](v, s) = \Delta[\alpha](v, s)$ .

Ejemplo

Del análisis de la expresión "Vlv" surge el árbol  $\gamma = \text{Arg} [\text{SName} [\text{ANSeq} [\text{Vlv}]]]$ . Su denotación está dada por  $\Delta[\gamma](v, s) = (\text{Vlv}, s)$ .

**Interfaz al usuario**

El sistema cuenta con una interfaz que le permite introducir la información de un pozo a distintos niveles:

*Información física del pozo*

Como se puede ver en la tabla **FichaPozo** de ilustración 3, se requiere de una gran cantidad de información acerca de las características físicas del pozo. El sistema cuenta con una interfaz gráfica que le permite introducir estos datos de una manera ágil y sencilla para cada uno de los pozos que se estén registrando. Esta interfaz se muestra en la ilustración 10. El equipamiento del pozo se maneja en el siguiente nivel; por lo tanto, al oprimir el botón etiquetado

"Introducir" en la sección de equipamiento, se desplegará la pantalla que se muestra en la ilustración 11, para introducir el equipamiento mediante una expresión del lenguaje GLang, presentado en este documento.

*Información del equipo existente en el tren de descarga*

Para facilitarle al usuario común la introducción de expresiones del lenguaje GLang, el sistema le presenta la pantalla que se muestra en la ilustración 11. Este desplegado le permite seleccionar los símbolos y la forma en que se agregarán al espacio geométrico, mediante el ComboBox y los botones que se muestran en la esquina inferior derecha de la ilustración 11.

Al ir seleccionando los símbolos, automáticamente el sistema va generando la expresión del lenguaje GLang, tal como se muestra en la caja de texto de la parte central inferior de la ilustración 11. Cuando la expresión está lista se puede oprimir el botón con la etiqueta "Grafica", la

cual es inmediatamente traducida a su representación geométrica, tal como se presenta en el espacio geométrico de la parte superior de la ilustración.

*Información particular de cada uno de los equipos instalados en el tren de descarga*

Finalmente, cada vez que el usuario selecciona algún símbolo de la pantalla mostrada en la ilustración 11 para incluirlo en su tren de descarga, el sistema le responderá con un desplegado para la captura de la información correspondiente al equipo asociado con el símbolo. La ilustración 12 muestra un desplegado de este tipo para un símbolo que representa una bomba.

**Conclusiones**

A lo largo de este artículo se ha mostrado un método para uniformizar la descripción del equipamiento instalado

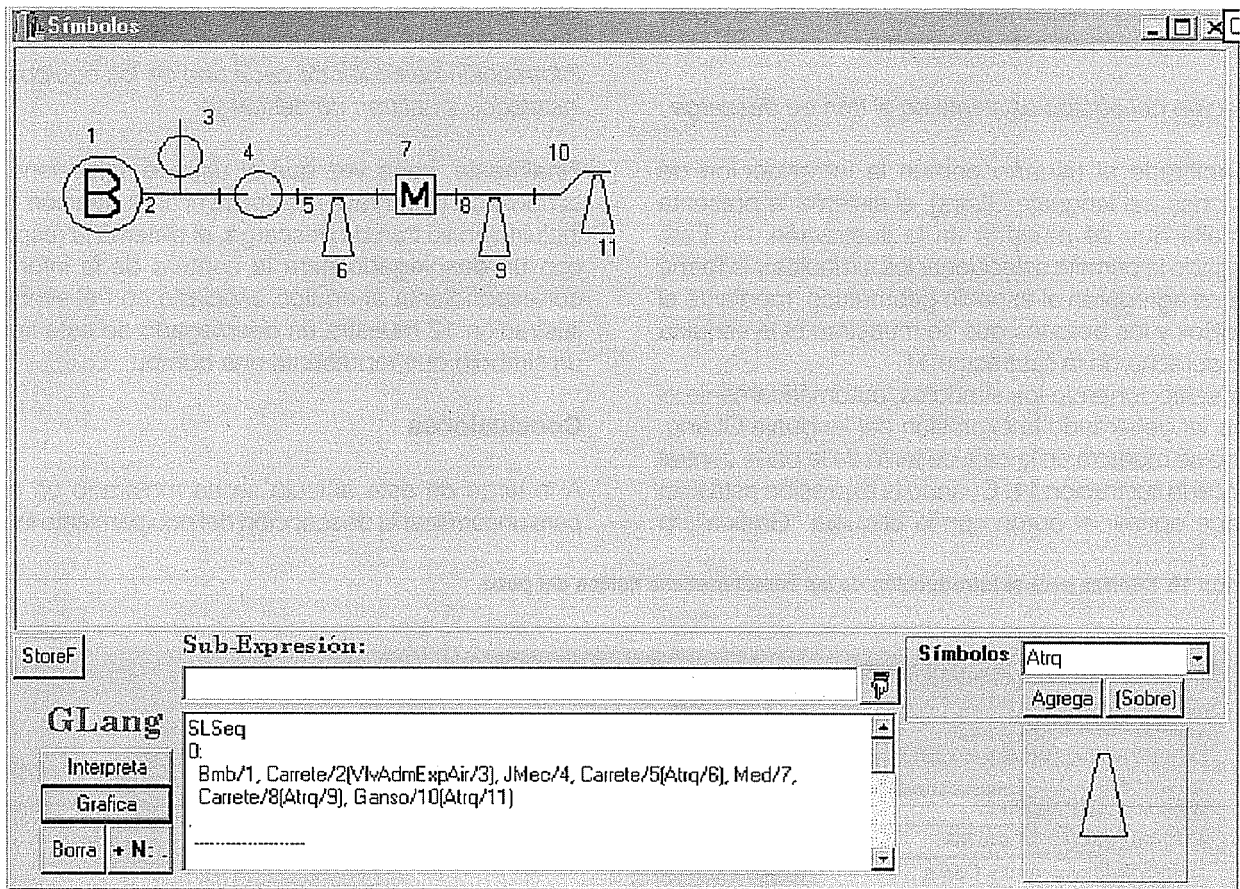
**Ilustración 10. Interfaz para la introducción de las características físicas del pozo.**

The screenshot shows a software window titled "Capturador" with a toolbar at the top. The interface is divided into several sections:

- A. Ubicación e identificación del pozo:** Includes fields for "folio" (148-089-063), "Expediente REPDA" (SN-27), "Concesionario" (Emmanuel Pérez), "Nombre del predio" (Santo Niño), "Municipio" (Buenaventura, Colima), "Núm. Medidor CFE", "Longitud" (106 51 25), "Latitud" (30 08 32), "Altitud" (230), and "Acceso" (Tomar la carretera 65, salir a la izquierda en el KM 42 y recorrer por la terracería 5 KM).
- B. Uso del pozo:** Includes "Uso del pozo" (radio buttons for Agícola, Mixto, Pecuario), "Área servida" (125), and "Cultivo(s)" (Tomate, Alfalfa).
- C. Características del pozo:** Includes "Longitud de la descarga" (Larga), "Tipo de salida" (radio buttons for Sistema presurizado, Tanque), "¿Salida a tubo lleno?" (radio buttons for No, Sí), "Profundidad del Pozo" (35), "Gasto concesionado" (60), "Gasto estimado" (55), "Diámetro de la descarga", and "De acuerdo con REPDA" (8).
- D. Equipamiento:** Features a prominent "Introducir" button.
- E. Características del agua:** Includes "Sólidos sedimentables" (0), "pH" (7.6), "Temperatura" (20), and "Observación de campo" (Agua limpia).
- F. Gráficos:** Includes fields for "Nombre arch. tren de descarga" (089-063-B.png) and "Nombre arch. medidor" (089-063-M.png).

At the bottom, there is a "Recopiló" section with "Nombre" (Fernández Tarango Héctor M), "Fecha" (10/07/01), "Hora" (14:23), and an "Agregar" button.

Ilustración 11. Interfaz gráfica para la introducción de expresiones de GLang.



en los trenes de descarga de pozos de bombeo, con un énfasis especial en su descripción gráfica. El núcleo de este método es un lenguaje, denominado GLang, que consiste prácticamente en ir mencionando de manera secuencial, uno a uno, los equipos en el tren de descarga.

La definición del lenguaje GLang para describir trenes de descarga se ha hecho de manera formal mediante el enunciado de sus reglas sintácticas y sus reglas semánticas.

Una herramienta de esta naturaleza permitirá recopilar la información de un gran volumen de pozos de bombeo de una manera ágil y libre de ambigüedades; pero no sólo eso, sino que permitirá, además, definir de manera más sencilla procedimientos automáticos para la emisión de recomendaciones de mejora para los diversos trenes de descarga, cuya información se levante mediante esta herramienta.

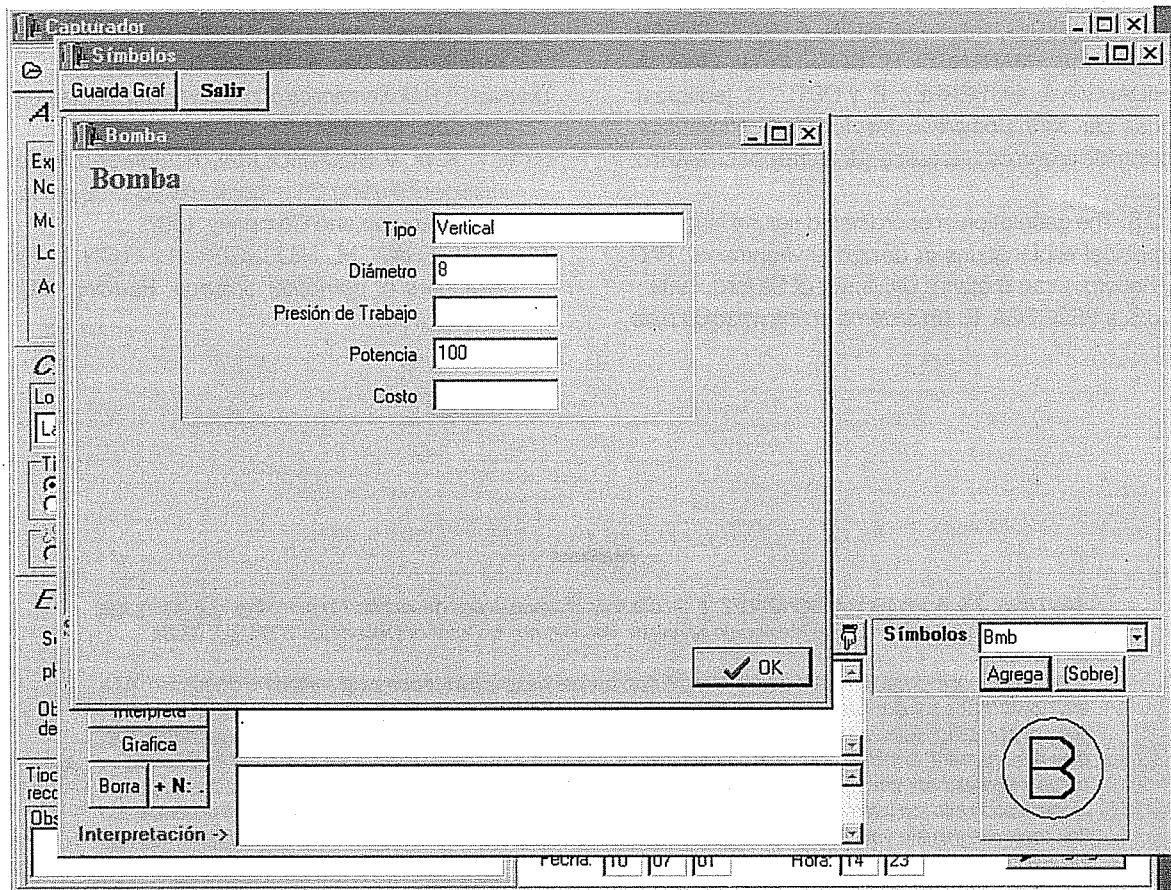
Cabe notar que por la forma de diseño de este lenguaje, es muy sencillo adaptarlo a otro tipo de aplicaciones.

Recibido: 31/03/2003  
Aprobado: 02/10/2003

## Referencias

- AHO, A. y ULLMAN, J. *The theory of Parsing, translation, and compiling*. Englewood Cliffs, EU: Prentice-Hall, 1972.
- AHO, A. y ULLMAN, J. *Principles of compiler design*, Reading, EU: Addison-Wesley, 1977.
- ALLEN, J. *Natural language understanding*. Segunda edición. Redwood City, EU: The Benjamin/Cummings Publishing Company, Inc., 1995.
- BRATKO, I. *Prolog programming for artificial intelligence*. Segunda edición. International Computing Science Series. Reading, EU: Addison Wesley Publishing Company, 1990.
- CHEN, P.P. The entity-relationship model: toward a unified view of data. *ACM transactions on database systems*. Vol. 1, núm. 1, 1976, pp. 9-36.
- CHEN, P., THALHEIM, B. y WONG, L. Future directions of

Ilustración 12. Interfaz para la introducción de las características de un equipo.



conceptual modeling. Chen, P., Akoka, J., Kangassalo, A. y Thalheim, B. (editors). *Conceptual modeling: current issues and future directions. Lecturing notes in computer sciences*. Núm. 1565. Springer-Verlag, Heidelberg, 1998, pp. 294-308.

CHURCH, A. A formulation of a simple theory of types. *Journal of symbolic logic*. Vol. 5, 1940, pp. 56-68.

DOWTY, D.R., WALL, R. E. y PETERS, S. *Introduction to Montague semantics*. Vol. 11 of *Studies in Linguistics and Philosophy*. Dordrecht: D. Reidel Publishing Company, P.O. Box 17, 3300 AA, Holland, 1981.

ENRÍQUEZ, S. y PEDROZA, E. *Medidores para pozos: manual técnico*. Jiutepec, México: Comisión Nacional del Agua/ Instituto Mexicano de Tecnología del Agua, 2001.

GAZDAR, G. y MELLISH, C. *Natural language processing in PROLOG*. Wokingham, Inglaterra: Addison-Wesley, 1989.

GORDON, M.J.C. *The denotational description of programming languages*. New York: Springer-Verlag, 1979.

HOPCROFT, J.E. y ULLMAN, J.D. *Introduction to automata theory,*

*languages and computation*. Reading, EU: Addison-Wesley, 1979.

KAY, M. The mind system. Rustin, R. (editor). *Natural language processing*. New York: Algorithmics Press, 1973, pp. 155-188.

KAY, M. Algorithm schemata and data structures in syntactic processing. Grosz, B.J., Jones, K. S. y Webber, B.L. (editors). *Readings in natural language processing*. Los Altos, EU: Morgan Kaufmann, 1980/1986, pp. 35-70.

NAUR, P. Revised report on the algorithmic language ALGOL 60. *Communications of the ACM*. Vol. 3, núm. 5, 1960, pp. 299-314.

NAUR, P. Revised report on the algorithmic language Algol 60', *Communications of the ACM*. Vol.6, núm. 1, 1963, pp. 1-17.

PEDROZA, E., ARROYO, V.M., OCHOA, L.H., RUIZ, J.A. y SANTANA, J.S. *Propuesta técnica para mejorar la medición en pozos agrícolas*. Technical report. Jiutepec, México: Instituto Mexicano de Tecnología del Agua, 2001.

- PEDROZA, E., ARROYO, V.M., OCHOA, L.H., RUIZ, J.A. y SANTANA, J.S. Mejoramiento de la medición en pozos agrícolas. *Ramírez et al.*, 2002 pp. 439-444.
- RAMÍREZ, A.I., APARICIO, F.J., ARREGUÍN, F., PAZ-SOLDÁN, G., DE LEÓN, B., MARENGO, H., SOTELO, G., GARCÍA, N.H., HIDALGO, J., GUTIÉRREZ, E. y ORTIZ, G. (editores). *Memorias del XVII Congreso Nacional de Hidráulica*. Asociación Mexicana de Hidráulica, Monterrey, N.L., México, 2002.
- SANTANA, J.S. The generation of coordinated natural language and graphical explanations in design environments. PhD thesis. Salford, Gran Bretaña: University of Salford, 1999.
- SANTANA, J.S. y PEDROZA, E. *Un sistema de información para el mejoramiento de la medición en pozos agrícolas en* *Ramírez et al.*, 2002, pp. 481-486.
- SPROAT, R. Lexical analysis, en Dale, R., Moisl, H. y Somers, H. (editors). *Handbook of natural language processing*. New York: Marcel Dekker, Inc., 2000, pp. 37-57.
- TENNENT, R.D. *Semantics of programming languages*. New York: Prentice-Hall, 1991.
- TRACY, K.W. y BOUTHORN, P. *Object-oriented artificial intelligence using C++ computer science press*. New York: W.H. Freeman and Company, 1997.
- WULF, W., SHAW, M., HILFINGER, P. y FLON, L. *Fundamental structures of computer science*. Reading, EU: Addison-Wesley, 1981.

#### Abstract

SANTANA, J.S. & PEDROZA-GONZÁLEZ, E. *Accessory characteristics description in pumping well discharge lines*. *Hydraulic engineering in Mexico (in Spanish)*. Vol. XIX, no. 4, October-December, 2004, pp. 65-84.

*The problem of information management for the appropriate description of pumping stations is discussed. As a step towards the solution of this problem, a graphical language aimed at expressing the configuration of discharge lines of pumping wells is introduced, which allows to describe the accessories installed on a discharge line, their arrangement, and their particular characteristics. The language, developed in the context of an information system for water administration, is formally described by its syntactic and semantic rules. Finally, the language incorporation into the information system by means of an adequate user interface is shown, and it is concluded that tools like this are very useful for the in-field gathering of great volumes of information about pumping well installations.*

**Keywords:** *information systems, water administration, graphical languages, discharge lines, pumping stations.*

#### Dirección institucional de los autores:

Dr. Julio Sergio Santana  
M. en I. Edmundo Pedroza-González

Instituto Mexicano de Tecnología del Agua,  
Coordinación de Tecnología de Riego y Drenaje,  
Paseo Cuauhnáhuac 8532,  
colonia Progreso,  
62550 Jiutepec, Morelos, México.  
ssantana@tlaloc.imta.mx  
epedroza@tlaloc.imta.mx