



Modelo conceptual de datos

Fco. Javier Bohórquez Ogalla

Índice

1. Visión general	10
2. Intérprete	12
3. Nodos ejecutables	13
4. Tipos de datos	14
5. Sentencias de control de flujo	15
5.1. Sentencia	15
5.2. Sentencia vacía	16
5.3. include	16
5.4. exit	16
5.5. goto	17
5.6. if	17
5.7. switch	17
5.8. while	18
5.9. do...while	18
5.10. for	18
5.11. foreach	19
5.12. iloop	19
5.13. continue	19
5.14. break	20

5.15. try	20
5.16. throw	20
5.17. with	20
6. Definiciones	21
6.1. Variables	21
6.2. Funciones	22
6.3. Clases	22
6.4. Objetos	23
6.5. Listas	23
6.6. Pares clave/valor	23
6.7. Etiquetas	24
6.8. Definiciones globales	24
6.9. Generadores	24
7. Asignaciones	25
7.1. Asignación	25
7.2. Asignación de referencia	25
8. Operadores aritméticos	26
8.1. Suma	26
8.2. Diferencia	26
8.3. Producto	27
8.4. División	27
8.5. Potencia	27

8.6.	Módulo	28
8.7.	Tamaño	28
8.8.	Incremento y asignación	28
8.9.	Asignación e incremento	29
8.10.	Decremento y asignación	29
8.11.	Asignación y decremento	30
8.12.	Suma y asignación	30
8.13.	Diferencia y asignación	30
8.14.	Producto y asignación	30
8.15.	División y asignación	30
8.16.	Potencia y asignación	31
8.17.	Módulo y asignación	31
9.	Operadores lógicos	32
9.1.	Or	32
9.2.	And	32
9.3.	Negación	33
9.4.	Igual que	33
9.5.	Distinto que	33
9.6.	Menor que	34
9.7.	Menor o igual que	34
9.8.	Mayor que	34
9.9.	Mayor o igual que	35
9.10.	Idéntico a	35

9.11. No idéntico a	35
9.12. Es nulo	36
9.13. Vacío	36
10.Operadores sobre cadenas	37
10.1. Concatenación	37
10.2. explode	37
10.3. implode	38
10.4. sprintf	38
10.5. Buscar subcadena	38
10.6. Buscar y remplazar	39
10.7. Remplazar subcadena	39
10.8. Convertir a mayúsculas	39
10.9. Convertir a minúsculas	40
10.10Concatenación y asignación	40
11.Operadores sobre array	41
11.1. Dividir en fragmentos	41
11.2. Reducir mediante función	41
11.3. Obtener último	42
11.4. Obtener primero	42
11.5. Insertar en posición	42
11.6. Eliminar posición	42
11.7. Insertar al inicio	42

11.8. Insertar al final	42
11.9. Buscar	42
11.10 Ordenar	42
11.11 in_array	43
11.12 Rango	43
11.13 Eliminar e insertar a partir de posición (splice)	43
12. Operadores sobre expresiones regulares	44
12.1. Crear expresión regular	44
12.2. match	44
12.3. search	45
13. Conversión de tipos	46
13.1. Conversión a lógico	46
13.2. Conversión a entero	46
13.3. Conversión a flotante	46
13.4. Conversión a cadena	46
14. Operadores de acceso	47
14.1. Acceso a clave	47
14.2. Acceso a función	47
14.3. Acceso a variable de entorno	48
15. Operadores condicionales	49
15.1. Ternario	49

15.2. Fusión de nulos	49
16.Operadores de entrada/salida	50
16.1. Salida estándar	50
16.2. Entrada estándar	50
17.Operadores informativos	51
17.1. Tipo de	51
17.2. Tamaño de	51
17.3. Información sobre	52
18.Processos	53
18.1. Crear proceso	53
18.2. Esperar finalización de proceso	53
18.3. Obtener identificador de proceso	54
18.4. Obtener identificador de proceso padre	54
18.5. Ejecutar función como proceso	54
18.6. Salir de proceso	55
18.7. Señal a proceso	55
18.8. Manejador de señales	55
18.9. Evaluar cadena	56
18.10Ejecutar comando del sistema	56
19.Ficheros	57
19.1. Obtener un flujo a fichero	57

19.2. Escribir en flujo a fichero	57
19.3. Leer de flujo a fichero	58
19.4. Cambiar posición en flujo a fichero	58
19.5. Obtener posición en flujo a fichero	58
19.6. Cerrar flujo a fichero	59
19.7. Leer fichero	59
19.8. Escribir en fichero	59
19.9. Escribir al final de fichero	60
20.Fechas	61
20.1. Fecha y hora con formato	61
20.2. Tiempo Unix	61
20.3. sleep	62
21.Errores	63
21.1. Error	63
21.2. Manejador de errores	63
21.3. Manejador de excepciones no capturadas	63
22.Extensiones	64
22.1. Biblioteca GNU de internacionalización (gettext)	64
22.1.1. gettext	65
22.1.2. dgettext	65
22.1.3. bindtextdomain	65
22.1.4. setlocale	66

22.1.5. textdomain	66
22.2. Operaciones sobre un SGBD Mysql	66
22.2.1. Abrir conexión	67
22.2.2. Consulta	67
22.2.3. Cerrar conexión	67
22.2.4. Insertar datos	68
22.2.5. Seleccionar datos	68
22.2.6. Actualizar datos	69
22.2.7. Eliminar datos	69

1. Visión general

El presente documento representa un análisis de los datos que construyen el sistema OMI y cómo estos se relacionan. El documento describe el modelo conceptual de datos del sistema mediante diagramas de clases. Las clases son organizadas en paquetes para facilitar la modularidad del sistema y su entendimiento.

El diagrama general de paquetes describe los paquetes que componen el sistema según el carácter funcional de las entidades que contienen. Un paquete podrá contener clases u otros paquetes.

El paquete “interpreter” describe las entidades que procesan el contenido fuente según el léxico y la gramática del lenguaje OMI. El objetivo es generar el árbol de nodos ejecutables correspondiente al programa. Al procesarse este árbol en profundidad se aplicará la semántica que que encierran las líneas de código del contenido fuente, produciéndose de esta forma la ejecución del programa.

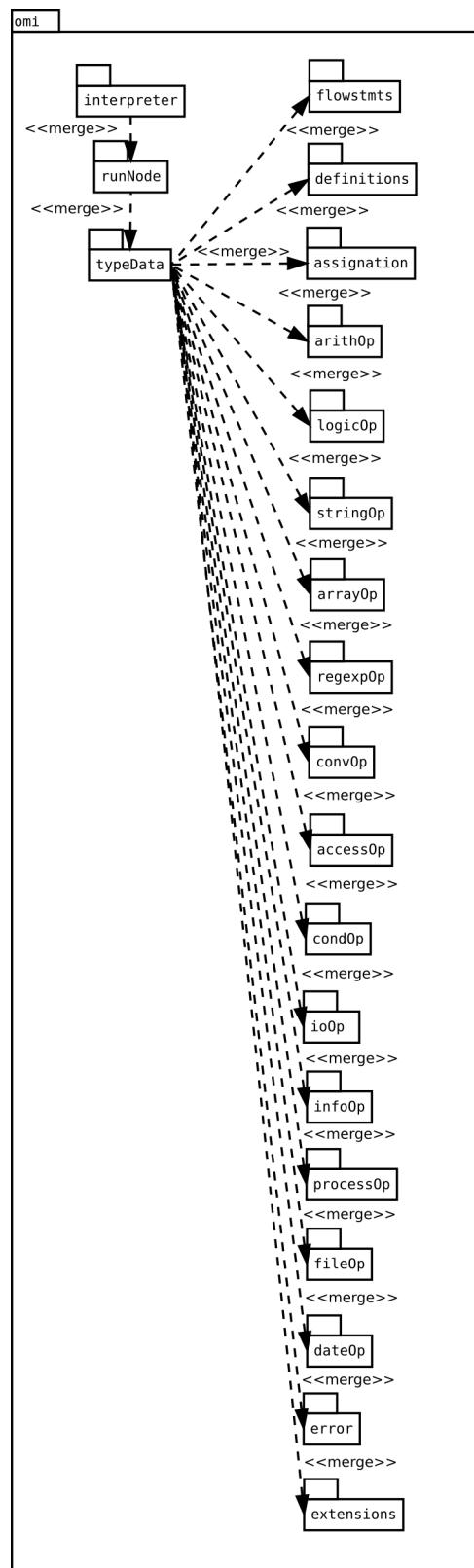
El paquete “runNode” describe el nodo ejecutable y aquellos tipos de nodos derivados de este, que son abstractos y que serán extendidos por tipos más específicos.

El paquete “typeData” describe los nodos correspondientes a los tipos de datos básicos que pueden ser manipulados por el sistema.

El paquete “error” describe el sistema de errores y los nodos ejecutables que permiten su control.

El paquete “extensions” describe el sistema de extensiones del interprete, el cual permite extender la funcionalidades del lenguaje de una forma dinámica. Además contiene dos el modelado de dos extensiones concretas.

Los paquetes restantes categorizan y agrupan nodos ejecutables según la funcionalidad que encierran y el tipo de dato sobre el que operan.



2. Intérprete

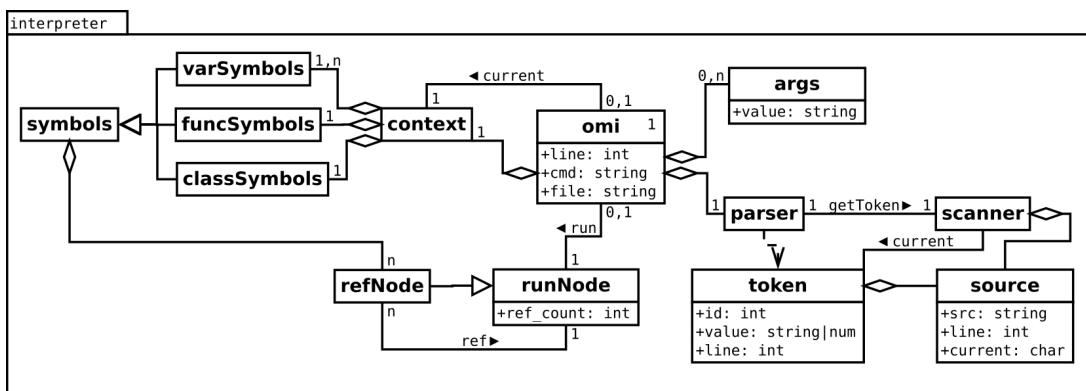
El sistema OMI se corresponde con un interprete que opera sobre un contenido fuente escrito en el lenguaje con el mismo nombre. El interprete se compone de un analizador sintáctico que encierra la gramática del lenguaje, esta es descrita a partir de una serie de tokens.

El analizador sintáctico se vale de un analizador léxico que validará y obtendrá los tokens (bajo petición) desde el código fuente. El analizador léxico debe controlar el fichero que contiene el código fuente, así como la línea y posición que se encuentra procesando en el mismo.

Los tokens obtenidos se definen por un identificador y la línea del código fuente en la que se generó, además pueden tener asociado un valor que puede ser numérico o cadena. Serán utilizados por el analizador sintáctico para determinar las reglas gramaticales que se deben aplicar y construir el árbol sintáctico correspondiente. Este árbol está formado por nodos denominados ejecutables, dado que al ser procesados en profundidad se llevará a cabo la ejecución del programa. Los nodos ejecutables dan significado semántico a cada una de las sentencias que componen el contenido fuente.

El interprete se compone además de un contexto denominado principal, que será sobre el que opere de forma predeterminada. Un contexto está formado por una serie de tablas de símbolos que serán manipuladas por ciertos nodos ejecutables cuando sean procesados. Estas tablas guardan referencias a nodos ejecutables correspondientes a símbolos variables, funciones y clases de objetos que son definidos en el código fuente. Existen determinados nodos que al ser ejecutados pueden cambiar el contexto en uso.

El interprete es ejecutado con una serie de argumentos que alteran su funcionamiento.



3. Nodos ejecutables

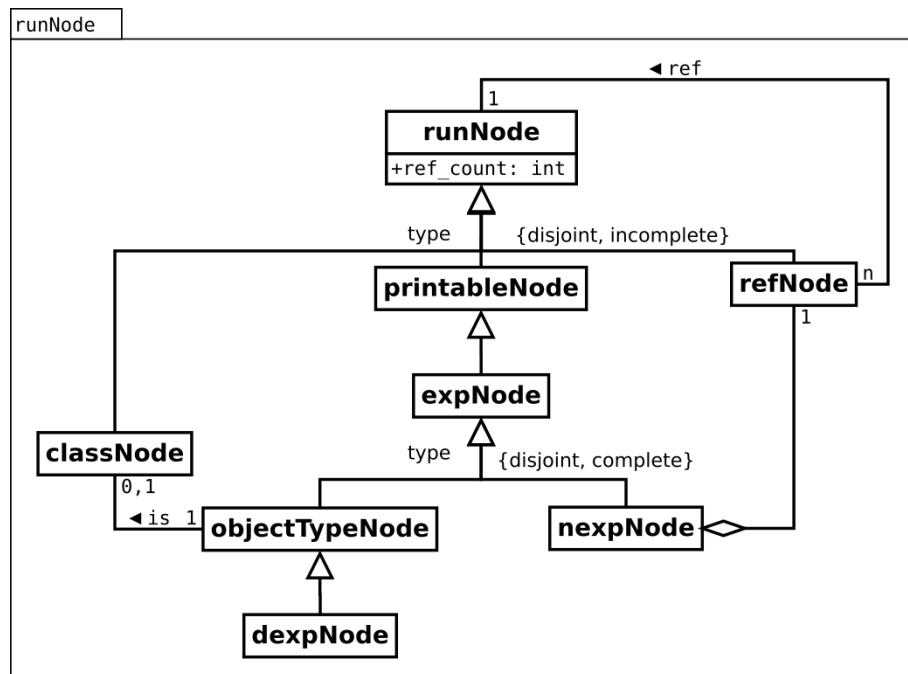
Se definen un nodo ejecutable para cada aspecto o funcionalidad que contemple el lenguaje. Cada sentencia se corresponde con un nodo ejecutable, que a su vez puede estar compuestos de otros nodos. Cada nodo ejecutable guarda el número de nodos que lo referencian para que se pueda hacer un uso óptimo del mismo.

Las expresiones son nodos ejecutables que tomarán un valor tras ser procesados. Generalmente forman parte de otros nodos correspondientes a sentencias u otras expresiones. El valor que toman pueden ser de un tipo determinado y conocido (numérico, lógico, etc), o de tipo indeterminado o no conocido hasta que el nodo es procesado.

Las expresiones de tipo determinado son extendidas por cada tipo de dato soportado por el lenguaje. Además pueden ser consideradas tipos de objetos y estar así asociadas a una clase. De esta forma toda expresión puede disponer de métodos y atributos según el tipo de dato que guarde.

Las expresiones de tipo indeterminado se componen de una referencia al nodo que guarda el valor tras la ejecución, este podrá ser una expresión de tipo determinado.

Las expresiones son nodos imprimibles lo que significa que tienen una representación gráfica asociada que puede ser volcada en la salida estándar.

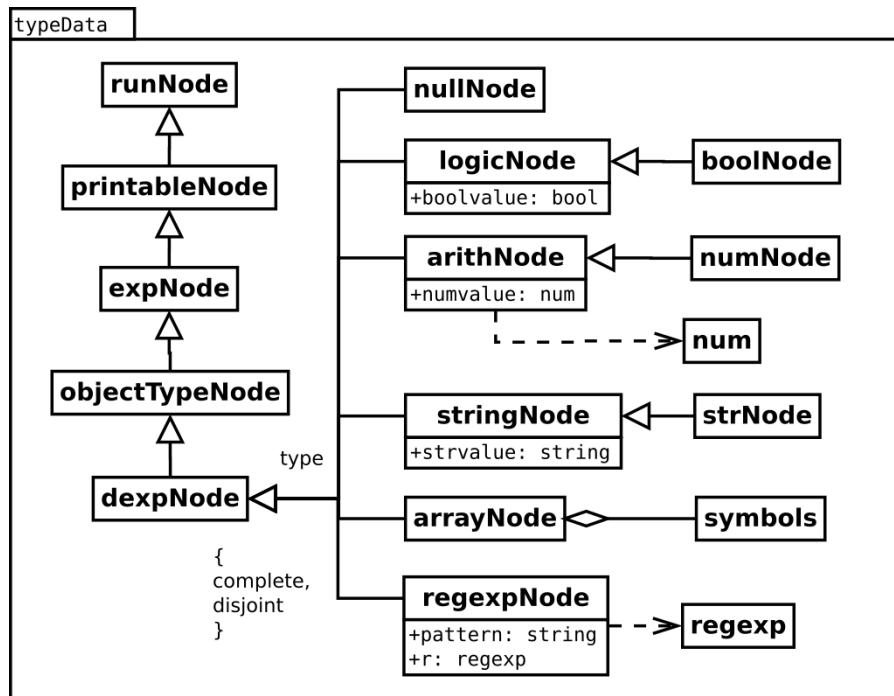


4. Tipos de datos

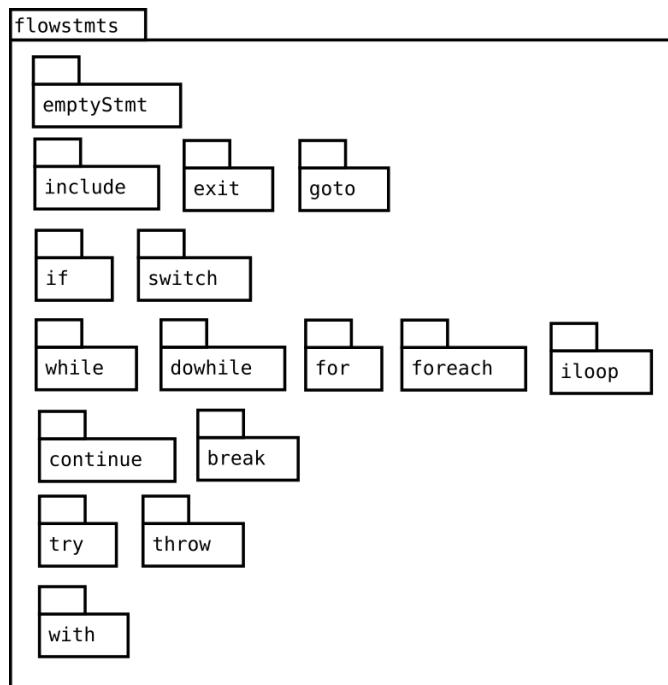
Este paquete contiene los nodos que representan expresiones con tipos de datos definidos. Se describe cada tipo de dato como un nodo con un valor asociado (en algunos casos el tipo puede comprender un único valor).

Muchos nodos son especializaciones de tipos de datos, correspondiéndose con expresiones que guardan un valor del tipo de dato al que extienden. Así por ejemplo los nodos de operaciones aritméticas generalmente extenderán al nodo del mismo tipo de dato.

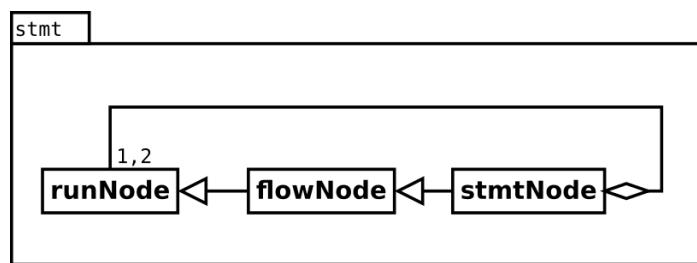
Algunos nodos de tipos datos son concretados por nodos que representan un valor constante de dicho tipo de dato.



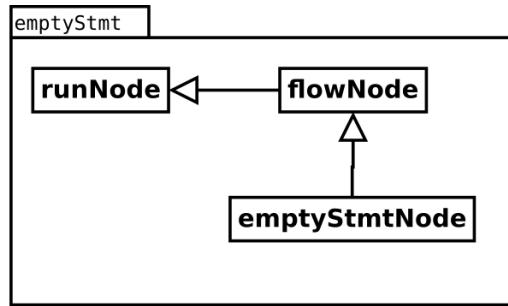
5. Sentencias de control de flujo



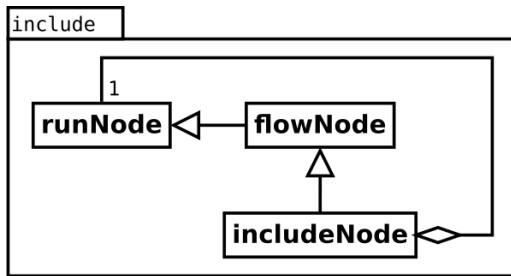
5.1. Sentencia



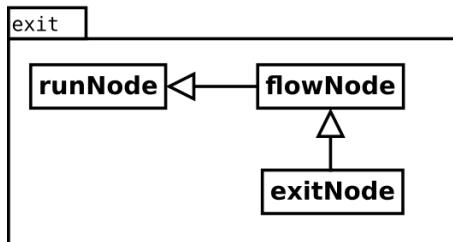
5.2. Sentencia vacía



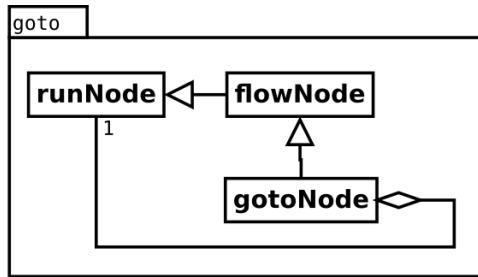
5.3. include



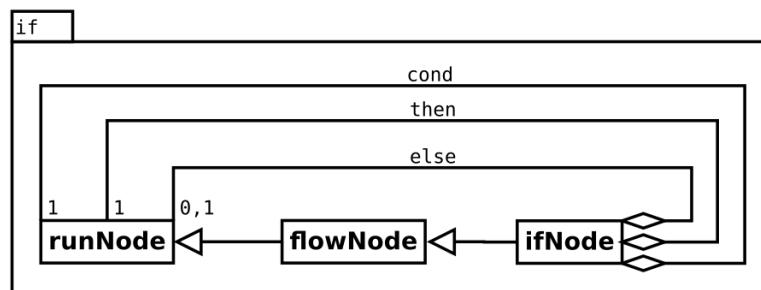
5.4. exit



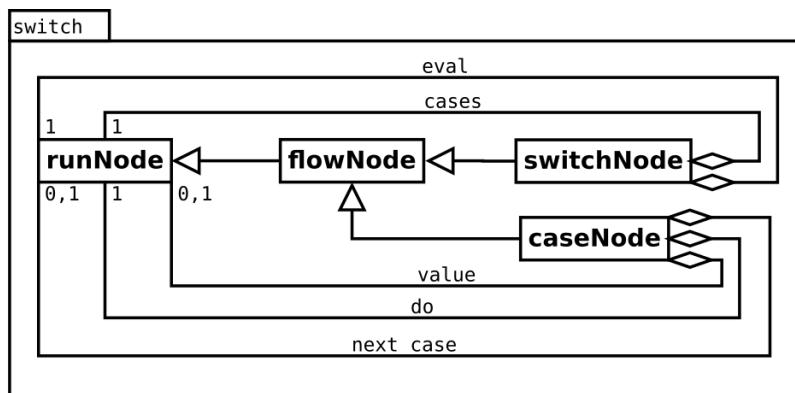
5.5. goto



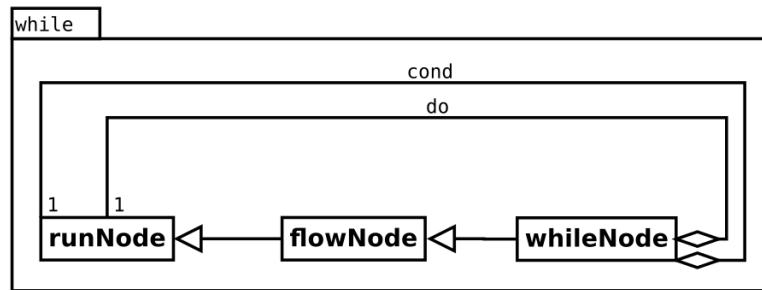
5.6. if



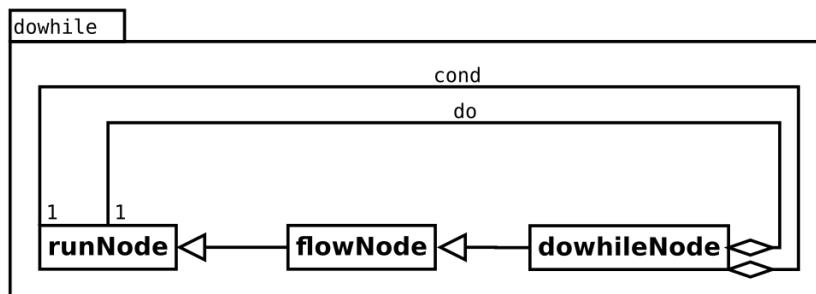
5.7. switch



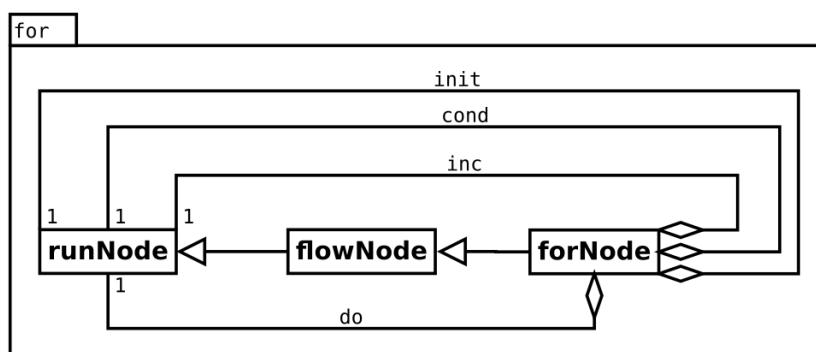
5.8. while



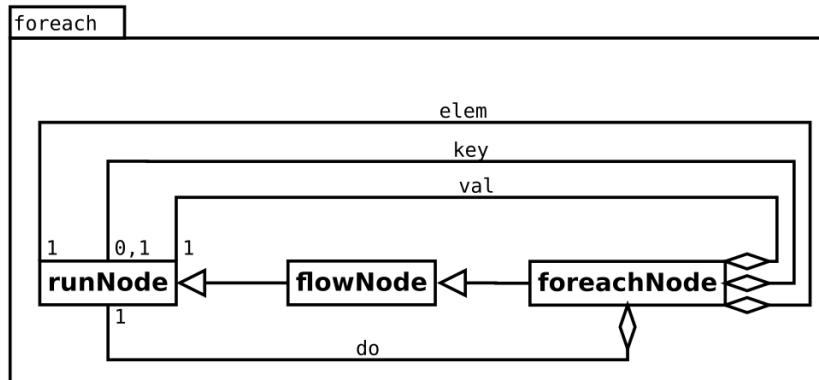
5.9. do...while



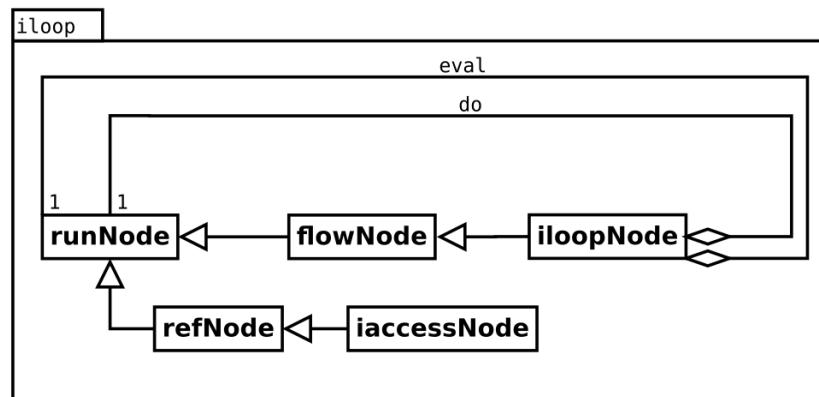
5.10. for



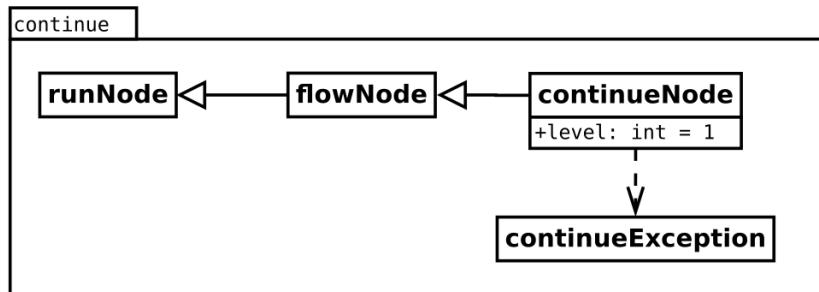
5.11. foreach



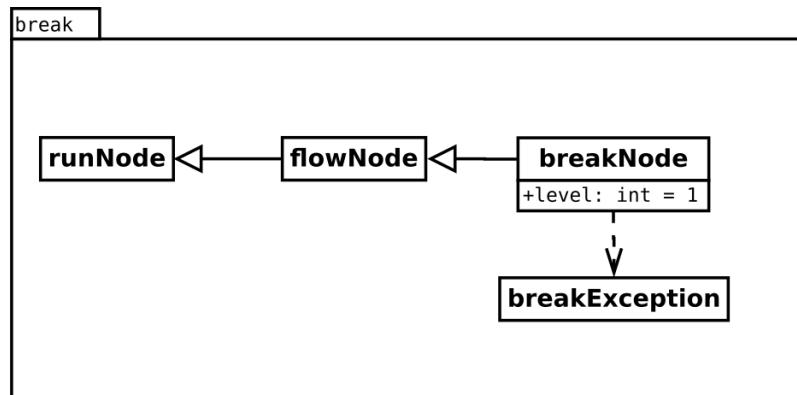
5.12. iloop



5.13. continue



5.14. break



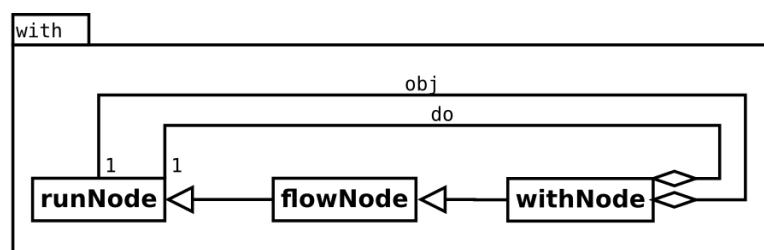
5.15. try

Diagrama aún por realizar.

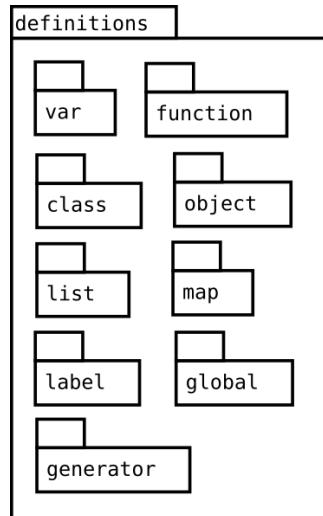
5.16. throw

Diagrama aún por realizar.

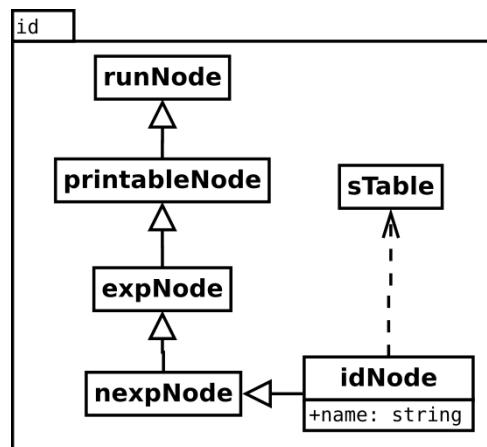
5.17. with



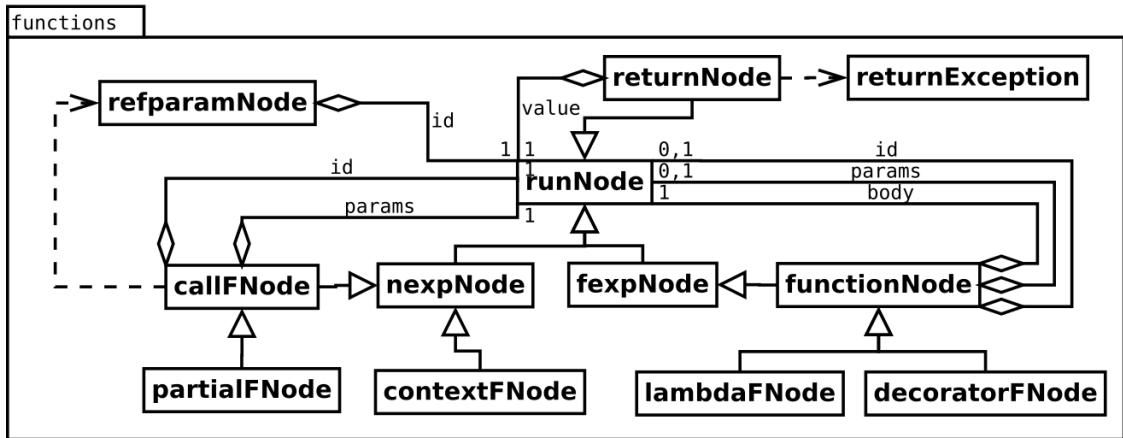
6. Definiciones



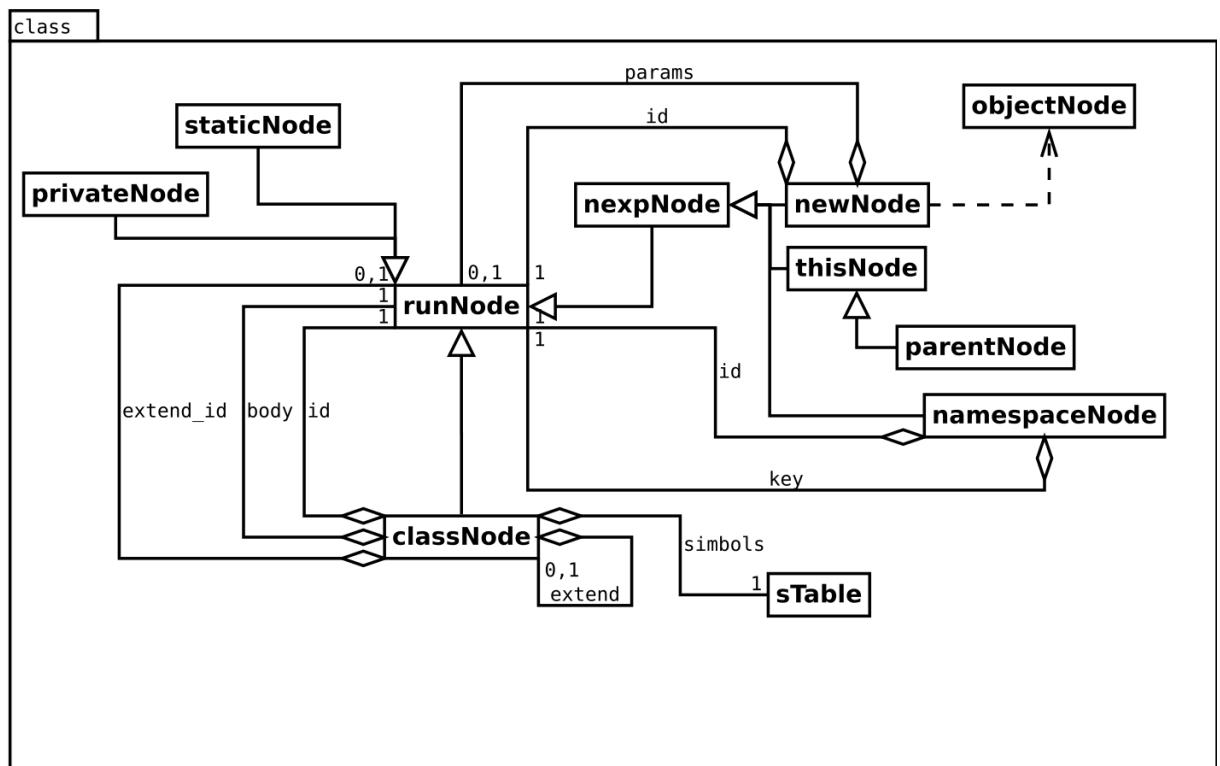
6.1. Variables



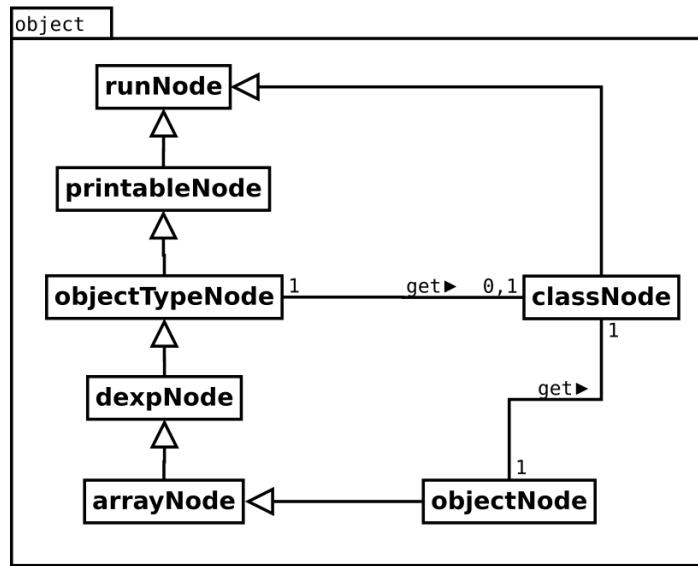
6.2. Funciones



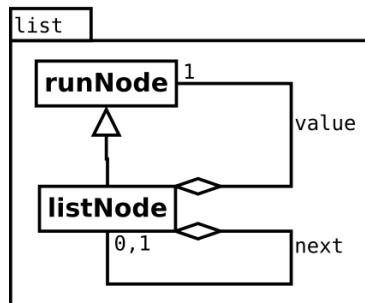
6.3. Clases



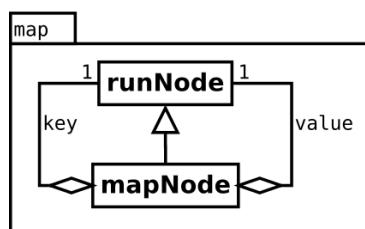
6.4. Objetos



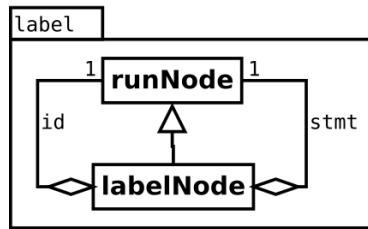
6.5. Listas



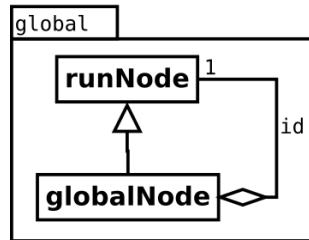
6.6. Pares clave/valor



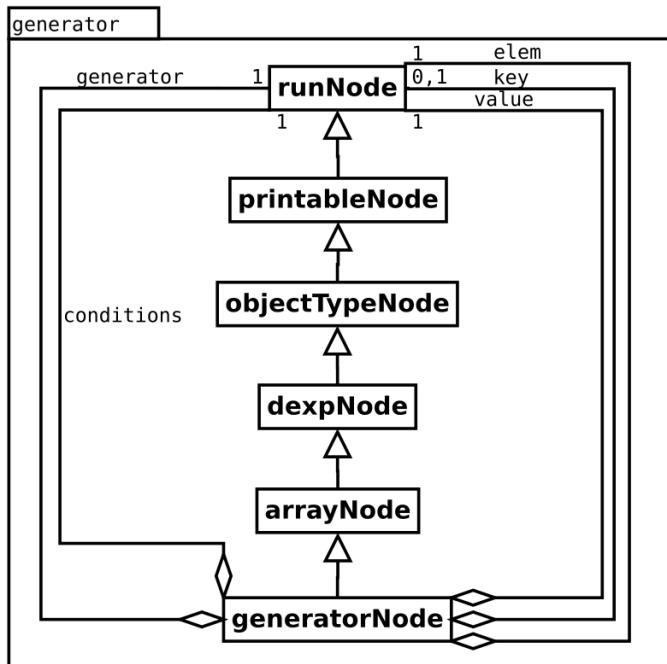
6.7. Etiquetas



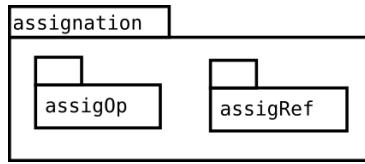
6.8. Definiciones globales



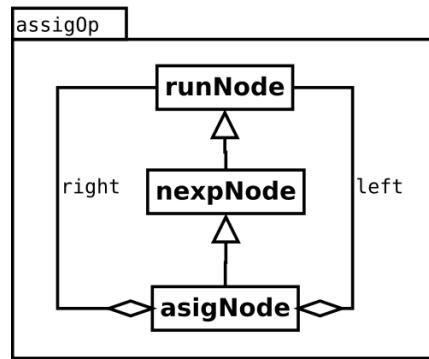
6.9. Generadores



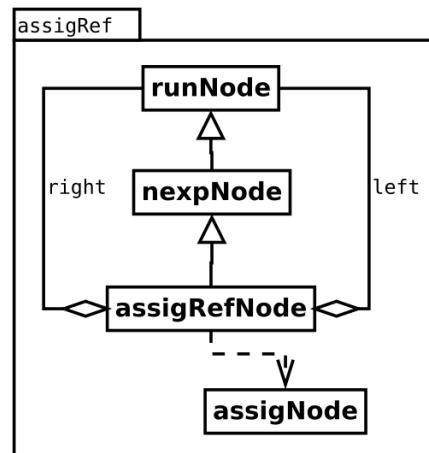
7. Asignaciones



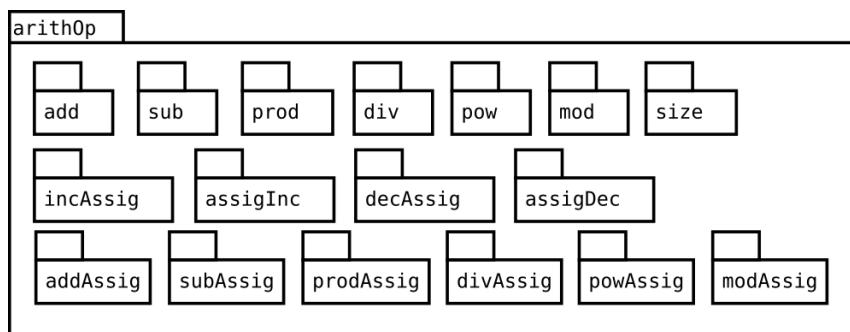
7.1. Asignación



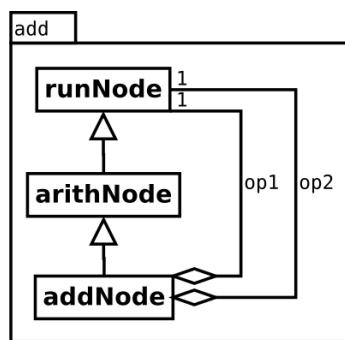
7.2. Asignación de referencia



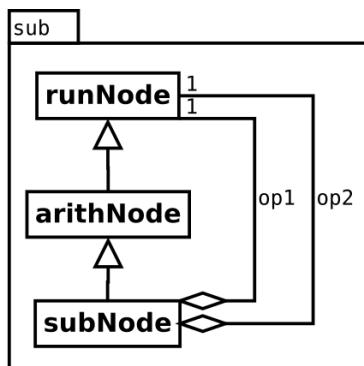
8. Operadores aritméticos



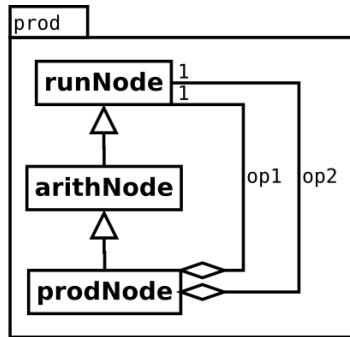
8.1. Suma



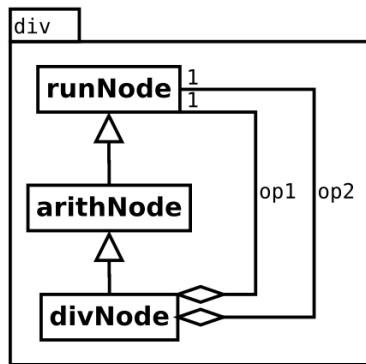
8.2. Diferencia



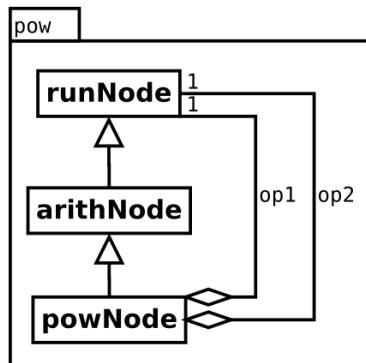
8.3. Producto



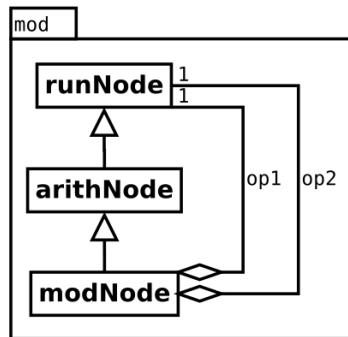
8.4. División



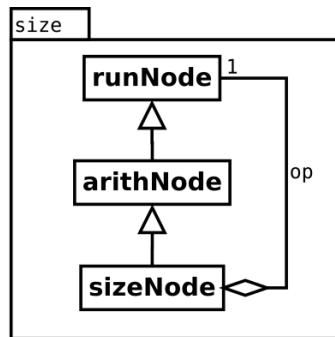
8.5. Potencia



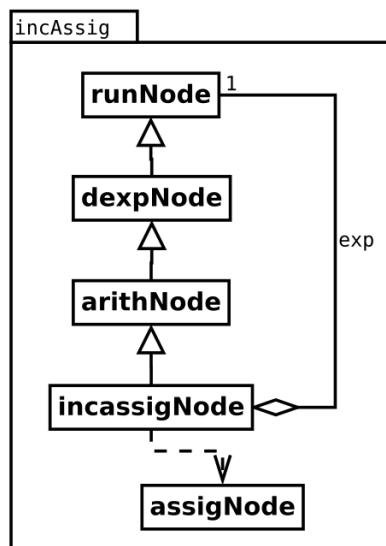
8.6. Módulo



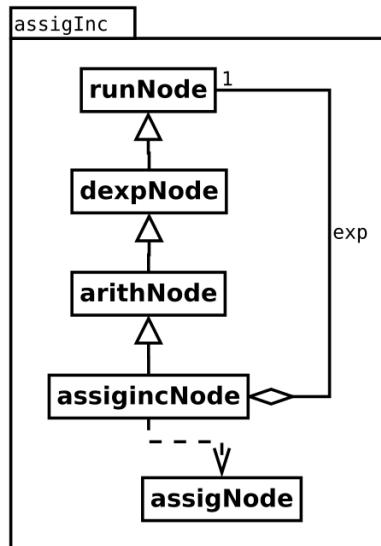
8.7. Tamaño



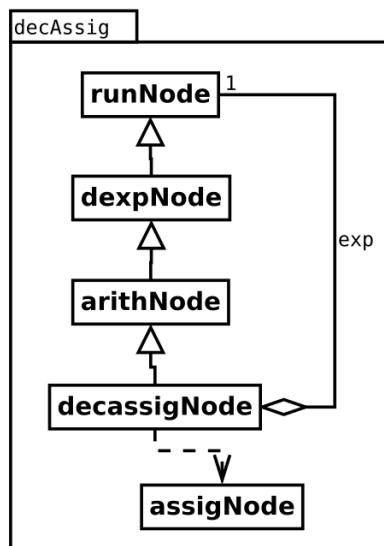
8.8. Incremento y asignación



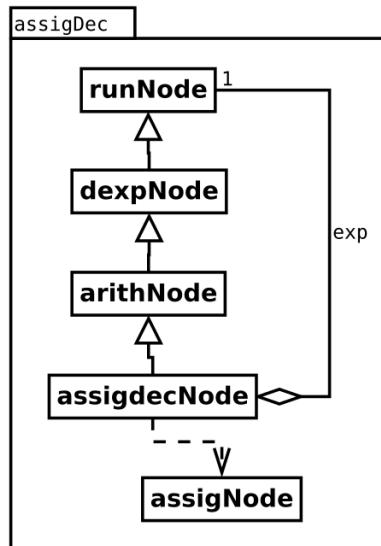
8.9. Asignación e incremento



8.10. Decremento y asignación



8.11. Asignación y decremento



8.12. Suma y asignación

Diagrama aún por realizar.

8.13. Diferencia y asignación

Diagrama aún por realizar.

8.14. Producto y asignación

Diagrama aún por realizar.

8.15. División y asignación

Diagrama aún por realizar.

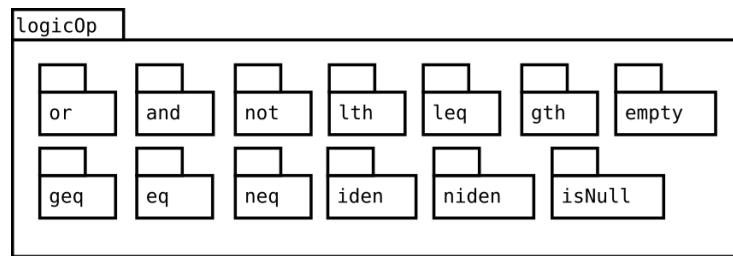
8.16. Potencia y asignación

Diagrama aún por realizar.

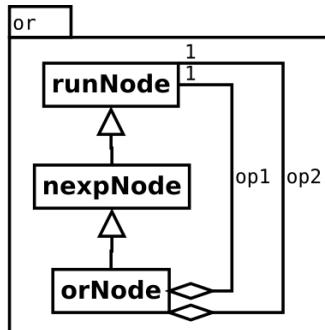
8.17. Módulo y asignación

Diagrama aún por realizar.

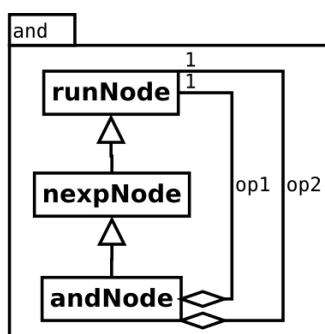
9. Operadores lógicos



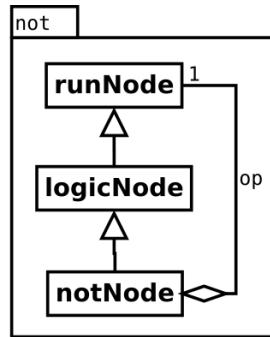
9.1. Or



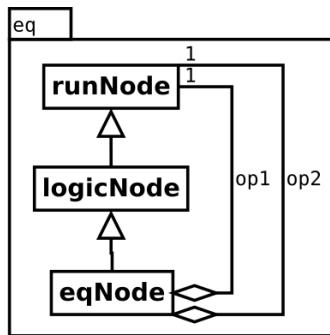
9.2. And



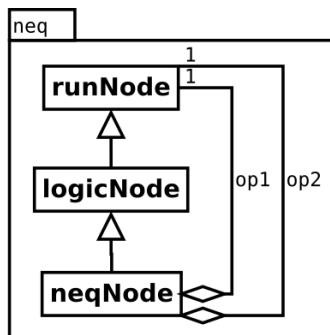
9.3. Negación



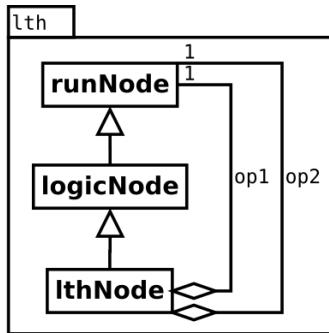
9.4. Igual que



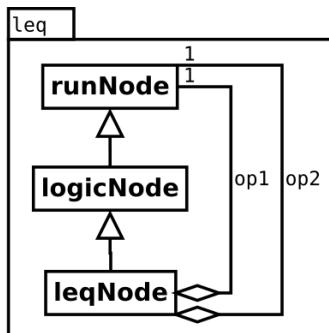
9.5. Distinto que



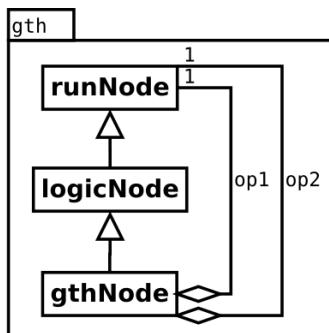
9.6. Menor que



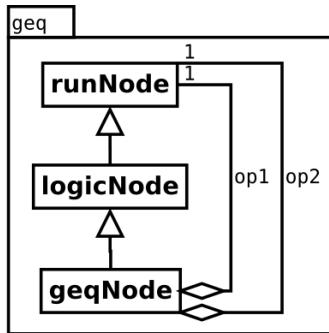
9.7. Menor o igual que



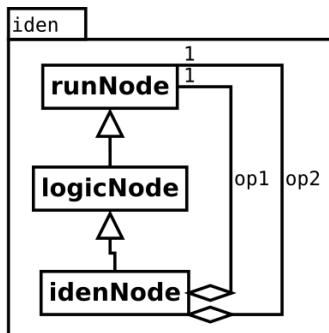
9.8. Mayor que



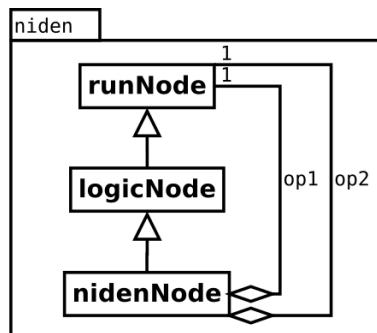
9.9. Mayor o igual que



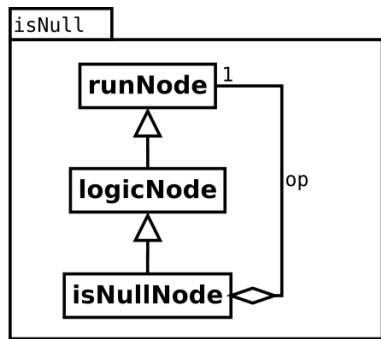
9.10. Idéntico a



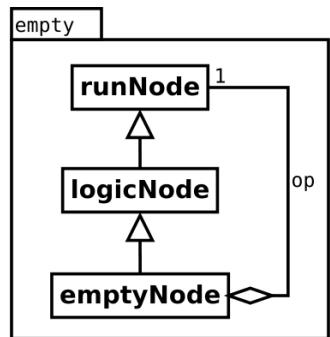
9.11. No idéntico a



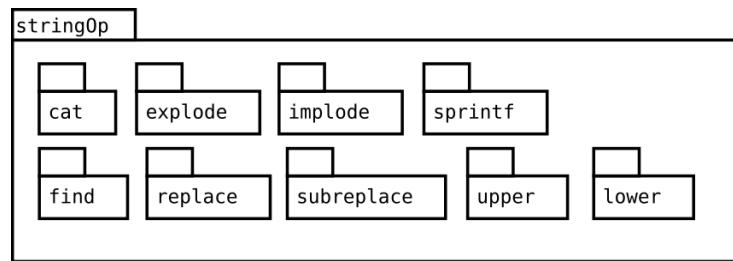
9.12. Es nulo



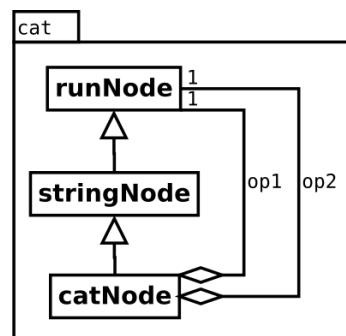
9.13. Vacío



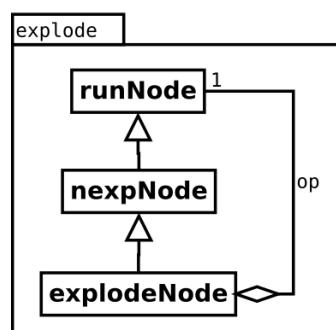
10. Operadores sobre cadenas



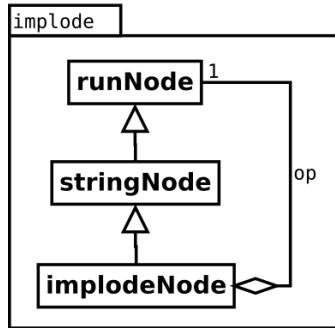
10.1. Concatenación



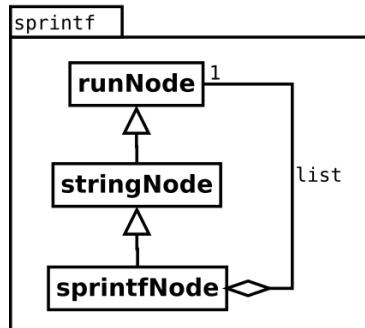
10.2. explode



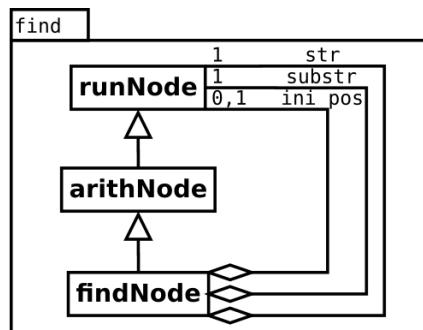
10.3. implode



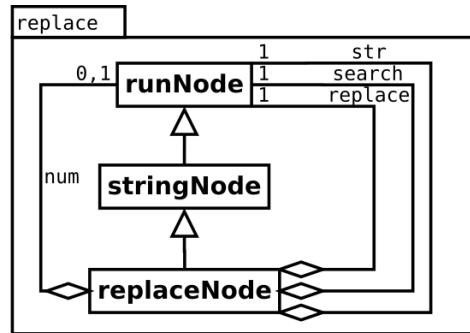
10.4. sprintf



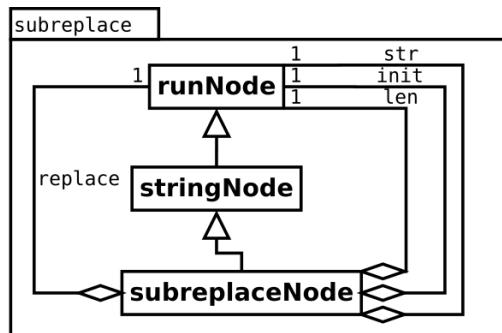
10.5. Buscar subcadena



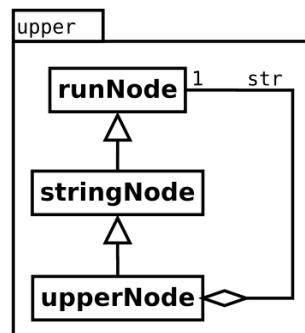
10.6. Buscar y remplazar



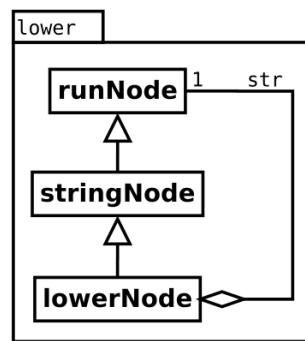
10.7. Remplazar subcadena



10.8. Convertir a mayúsculas



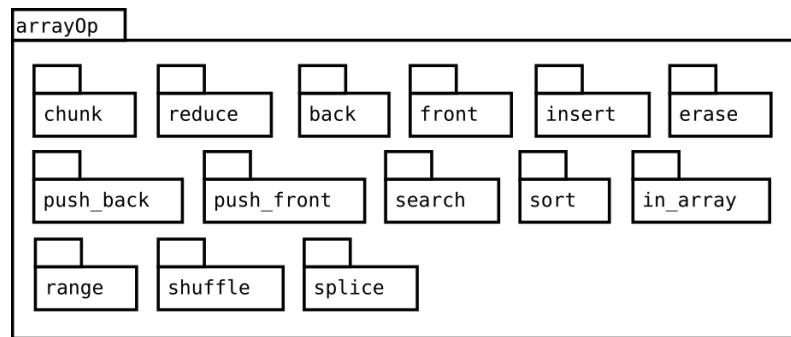
10.9. Convertir a minúsculas



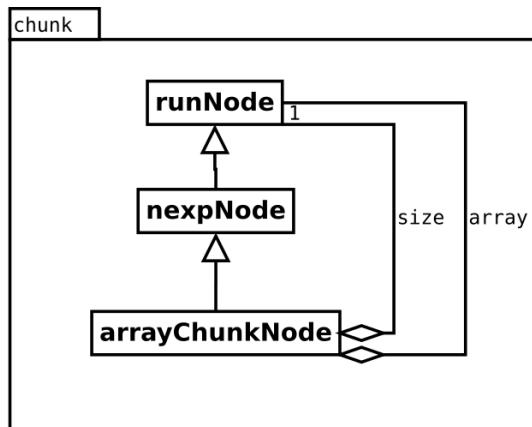
10.10. Concatenación y asignación

Diagrama aún por realizar.

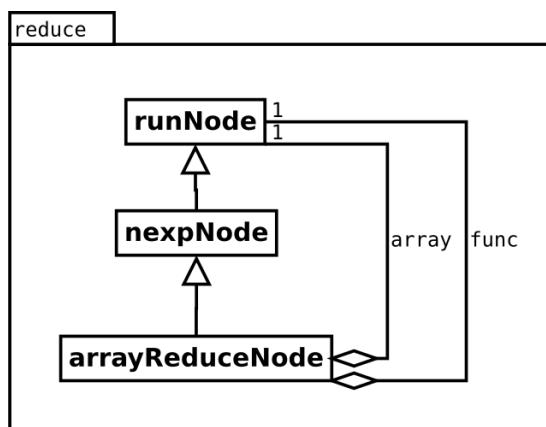
11. Operadores sobre array



11.1. Dividir en fragmentos



11.2. Reducir mediante función



11.3. Obtener último

Diagrama aún por realizar.

11.4. Obtener primero

Diagrama aún por realizar.

11.5. Insertar en posición

Diagrama aún por realizar.

11.6. Eliminar posición

Diagrama aún por realizar.

11.7. Insertar al inicio

Diagrama aún por realizar.

11.8. Insertar al final

Diagrama aún por realizar.

11.9. Buscar

Diagrama aún por realizar.

11.10. Ordenar

Diagrama aún por realizar.

11.11. in_array

Diagrama aún por realizar.

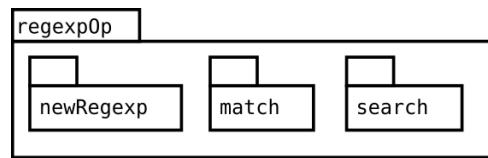
11.12. Rango

Diagrama aún por realizar.

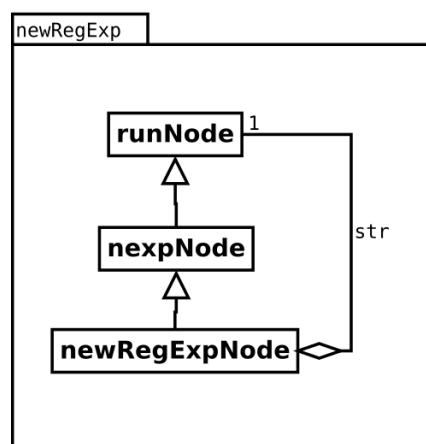
11.13. Eliminar e insertar a partir de posición (splice)

Diagrama aún por realizar.

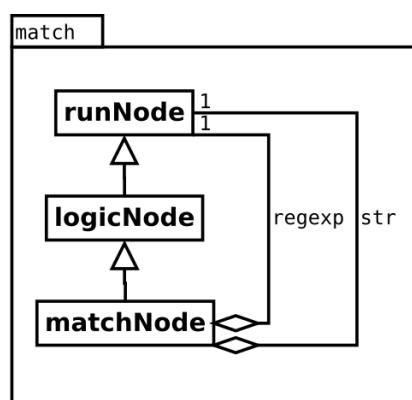
12. Operadores sobre expresiones regulares



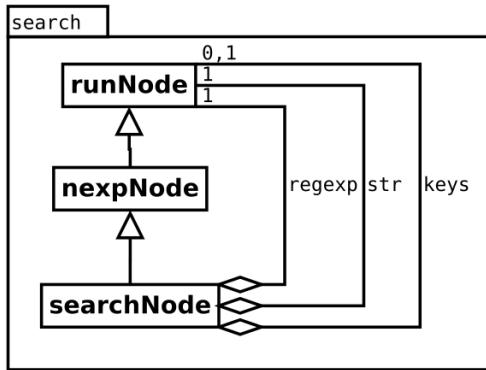
12.1. Crear expresión regular



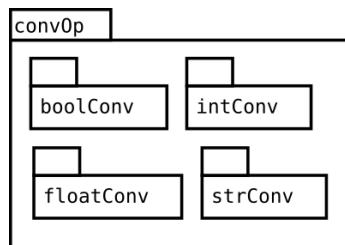
12.2. match



12.3. search



13. Conversión de tipos



13.1. Conversión a lógico

Diagrama aún por realizar.

13.2. Conversión a entero

Diagrama aún por realizar.

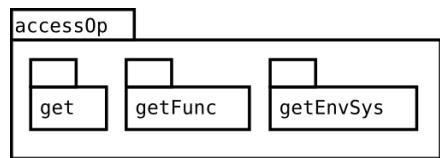
13.3. Conversión a flotante

Diagrama aún por realizar.

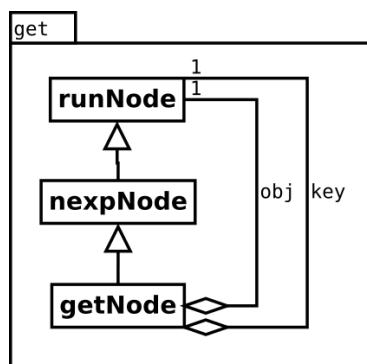
13.4. Conversión a cadena

Diagrama aún por realizar.

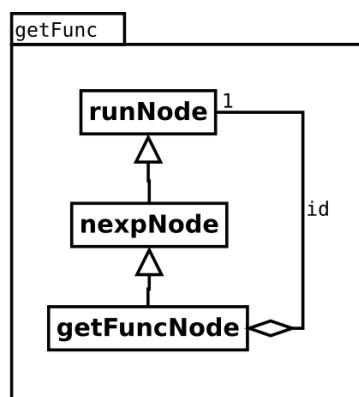
14. Operadores de acceso



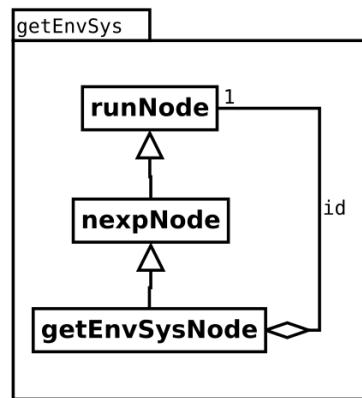
14.1. Acceso a clave



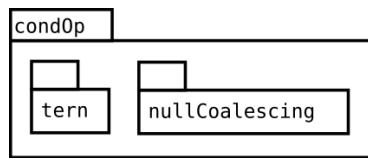
14.2. Acceso a función



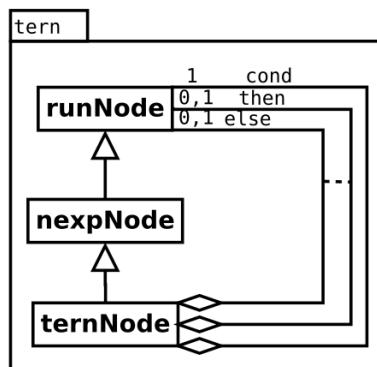
14.3. Acceso a variable de entorno



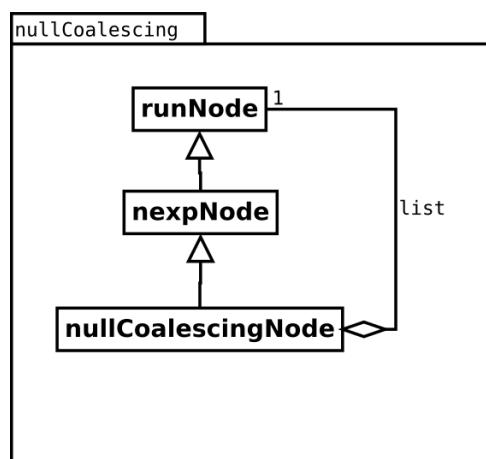
15. Operadores condicionales



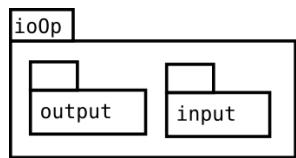
15.1. Ternario



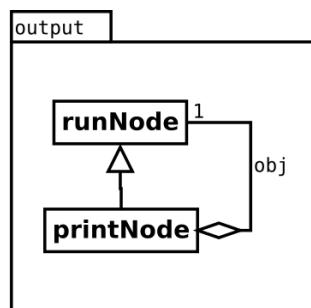
15.2. Fusión de nulos



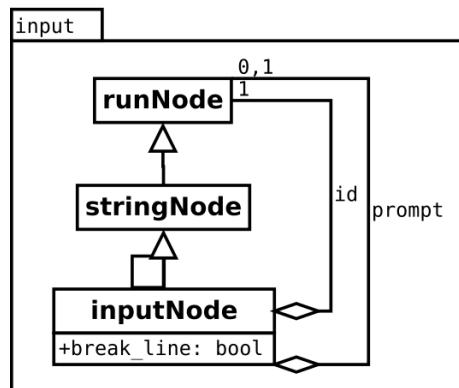
16. Operadores de entrada/salida



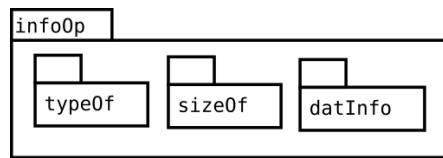
16.1. Salida estándar



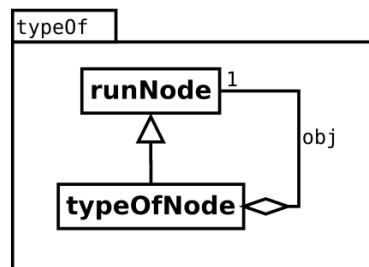
16.2. Entrada estándar



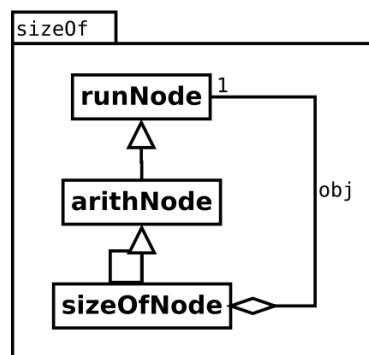
17. Operadores informativos



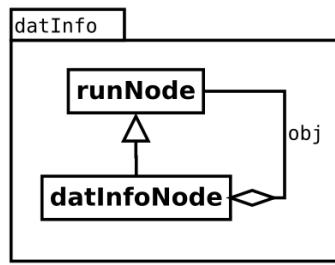
17.1. Tipo de



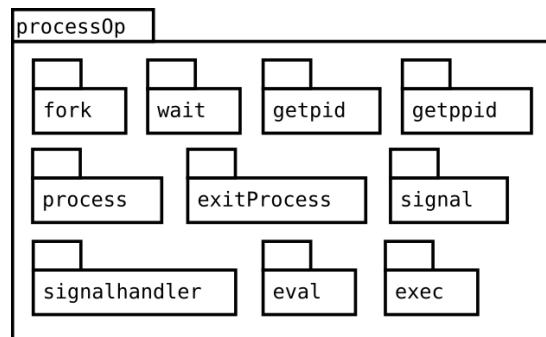
17.2. Tamaño de



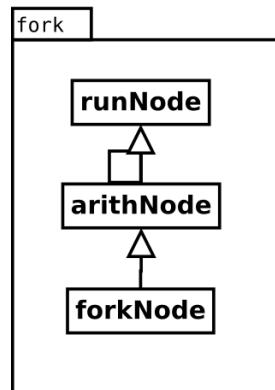
17.3. Información sobre



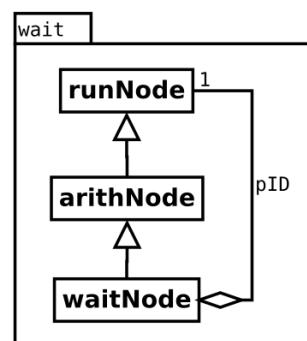
18. Procesos



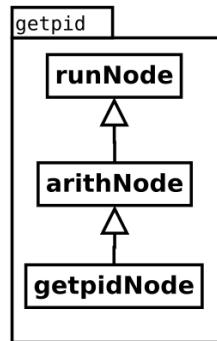
18.1. Crear proceso



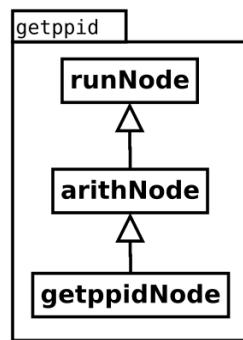
18.2. Esperar finalización de proceso



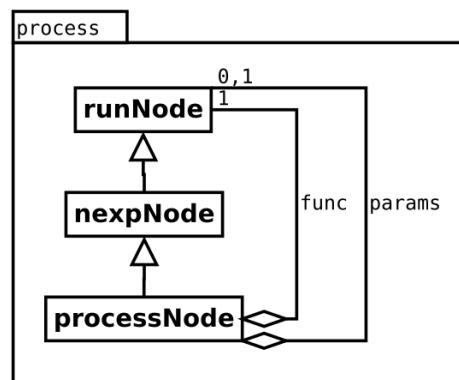
18.3. Obtener identificador de proceso



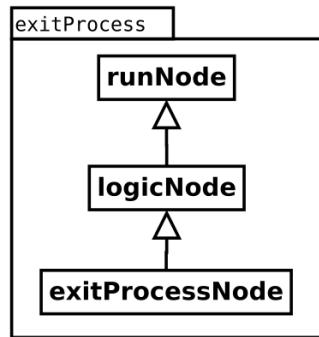
18.4. Obtener identificador de proceso padre



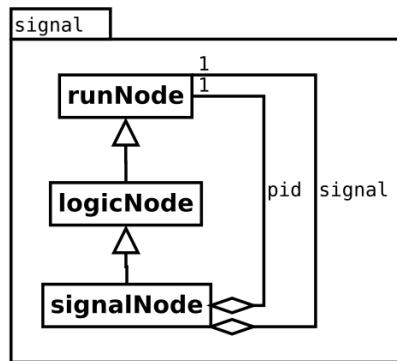
18.5. Ejecutar función como proceso



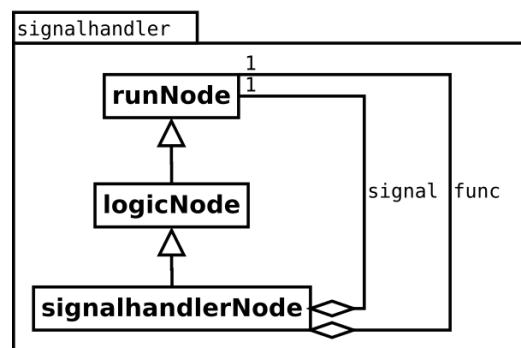
18.6. Salir de proceso



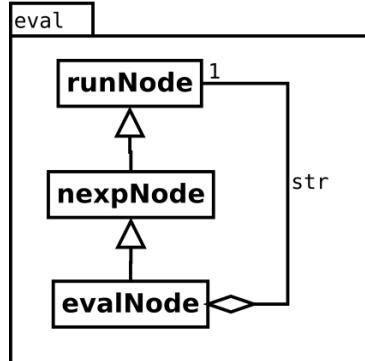
18.7. Señal a proceso



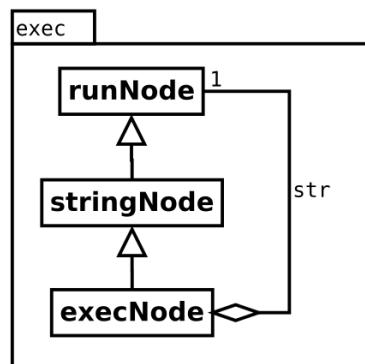
18.8. Manejador de señales



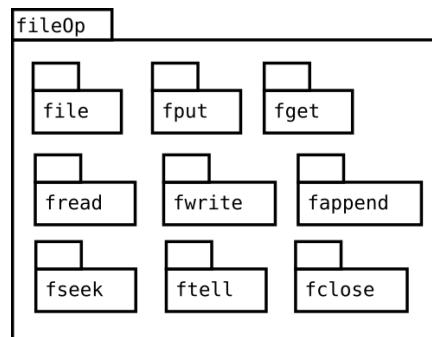
18.9. Evaluar cadena



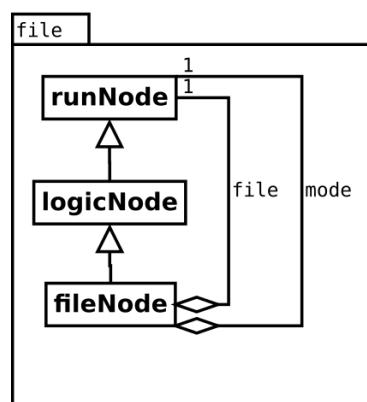
18.10. Ejecutar comando del sistema



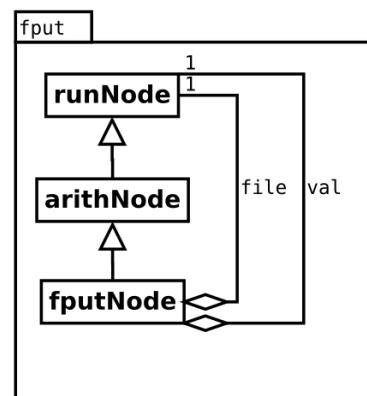
19. Ficheros



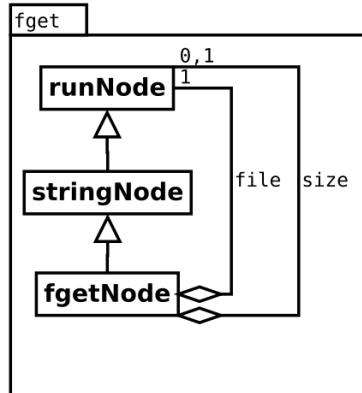
19.1. Obtener un flujo a fichero



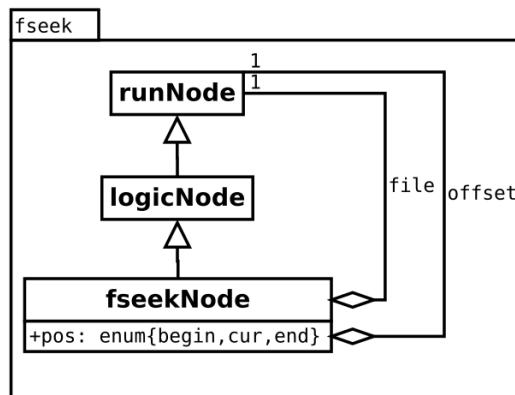
19.2. Escribir en flujo a fichero



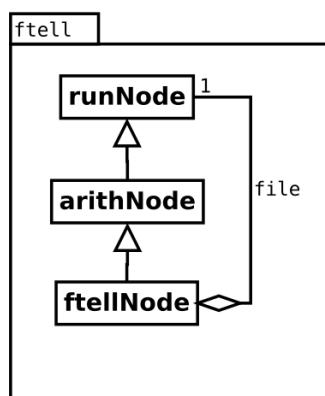
19.3. Leer de flujo a fichero



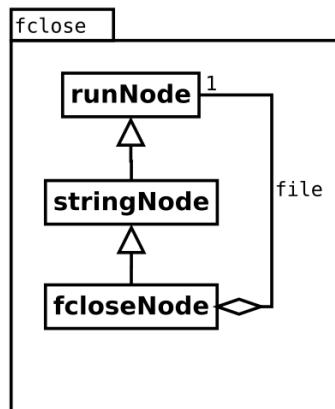
19.4. Cambiar posición en flujo a fichero



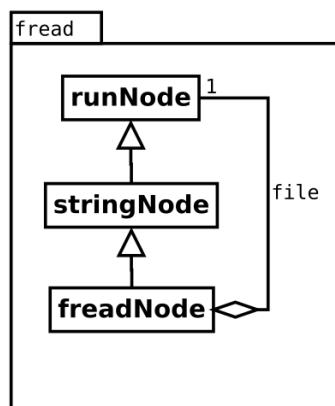
19.5. Obtener posición en flujo a fichero



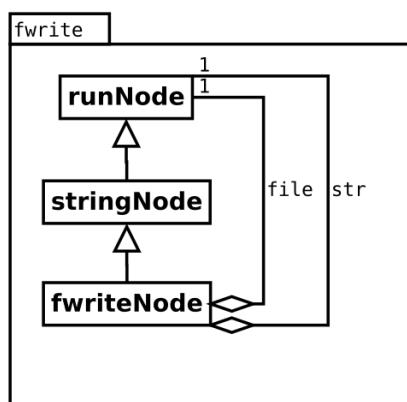
19.6. Cerrar flujo a fichero



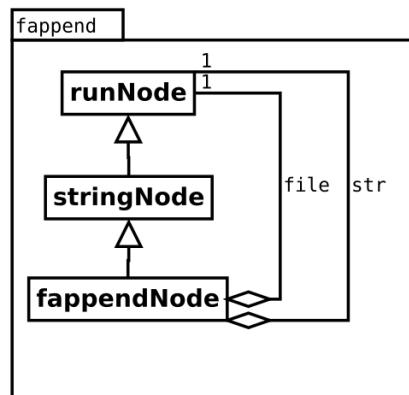
19.7. Leer fichero



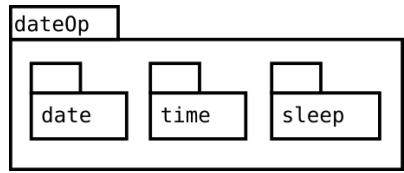
19.8. Escribir en fichero



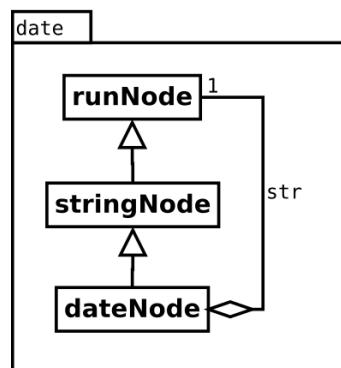
19.9. Escribir al final de fichero



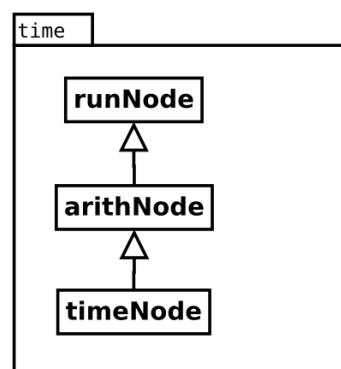
20. Fechas



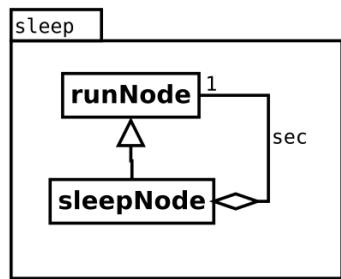
20.1. Fecha y hora con formato



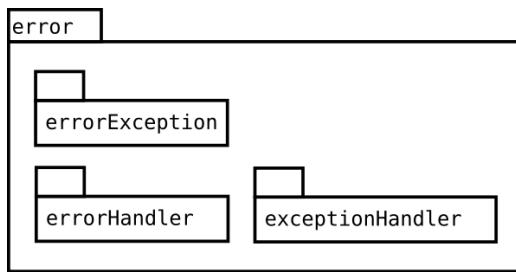
20.2. Tiempo Unix



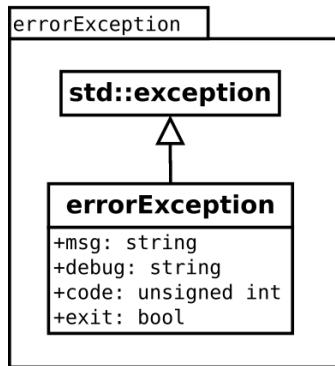
20.3. sleep



21. Errores



21.1. Error



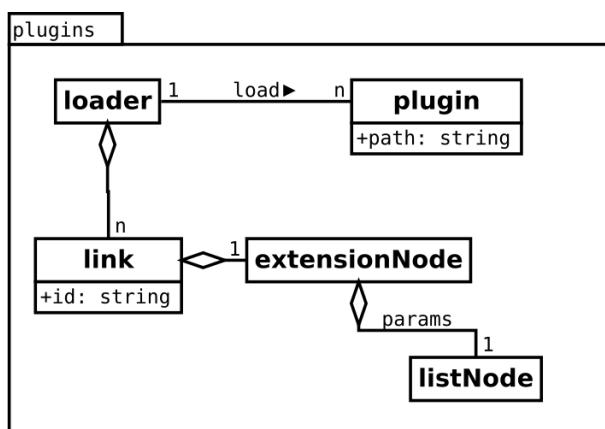
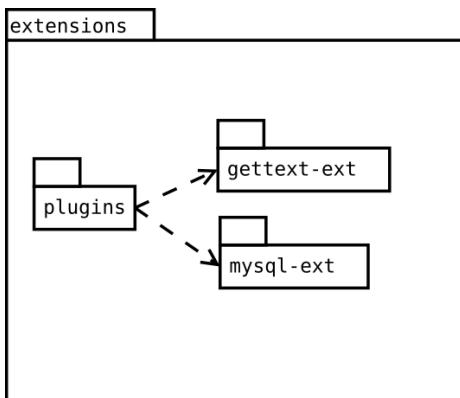
21.2. Manejador de errores

Diagrama aún por realizar

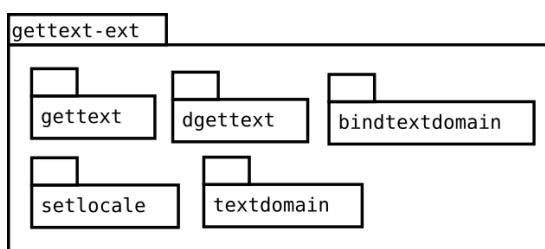
21.3. Manejador de excepciones no capturadas

Diagrama aún por realizar

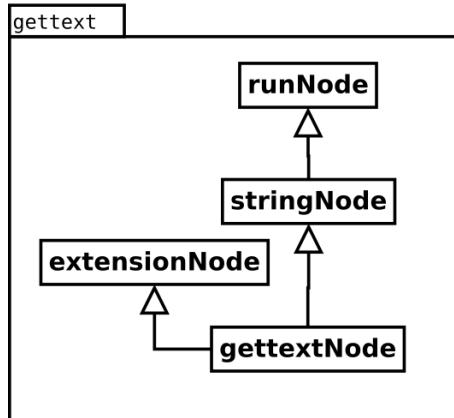
22. Extensiones



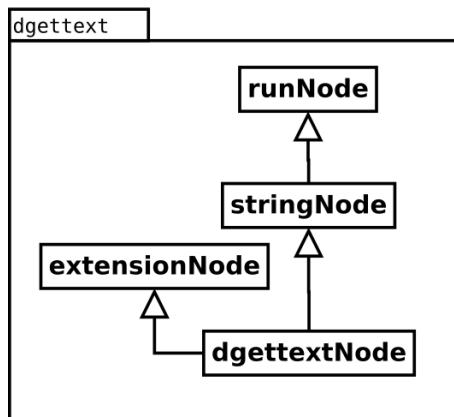
22.1. Biblioteca GNU de internacionalización (gettext)



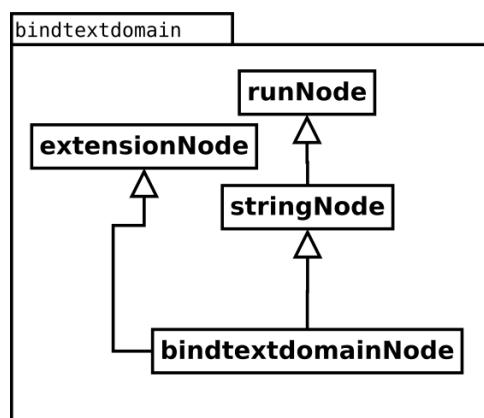
22.1.1. gettext



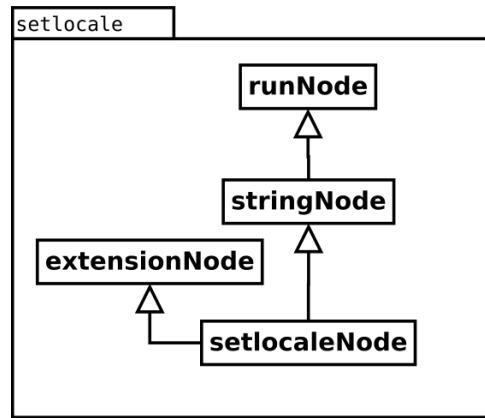
22.1.2. dgettext



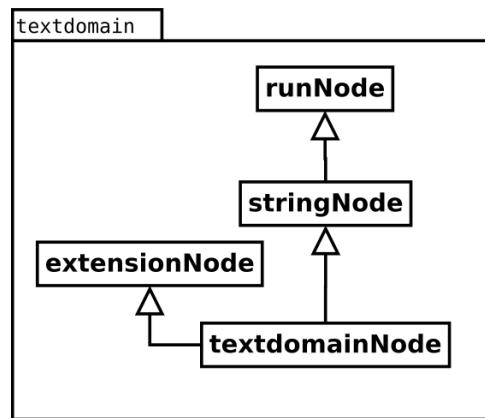
22.1.3. bindtextdomain



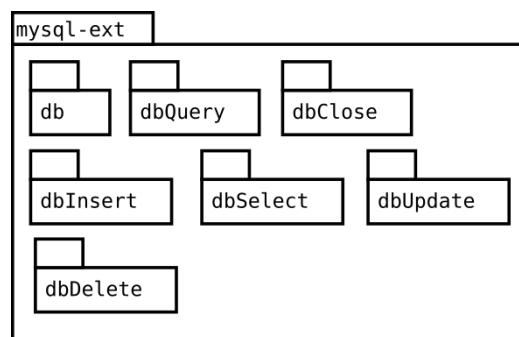
22.1.4. setlocale



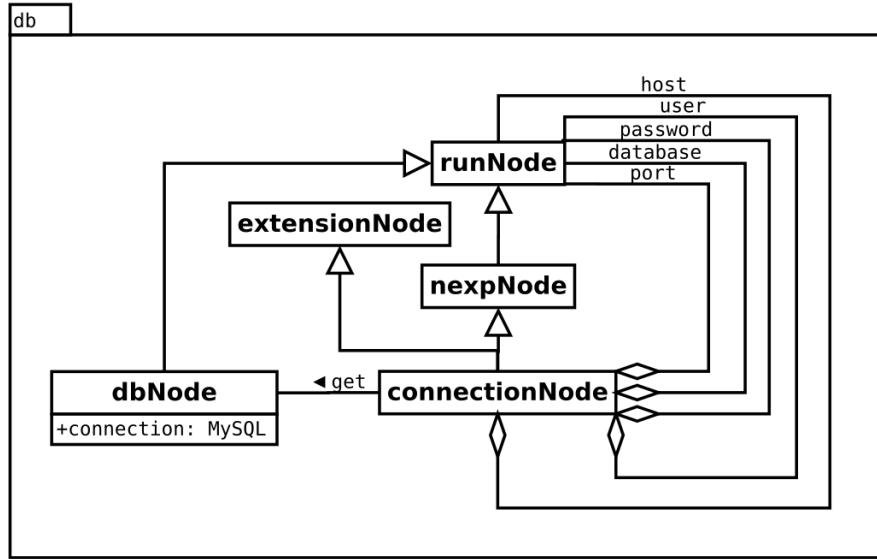
22.1.5. textdomain



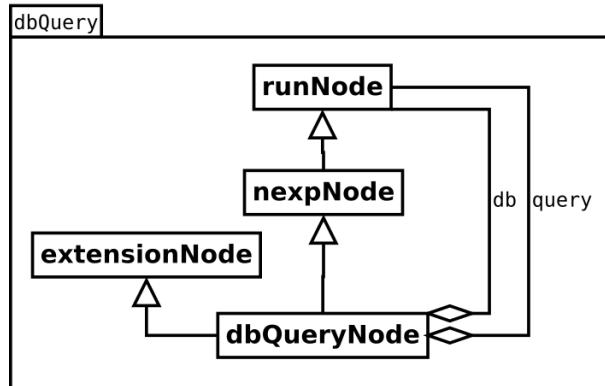
22.2. Operaciones sobre un SGBD Mysql



22.2.1. Abrir conexión



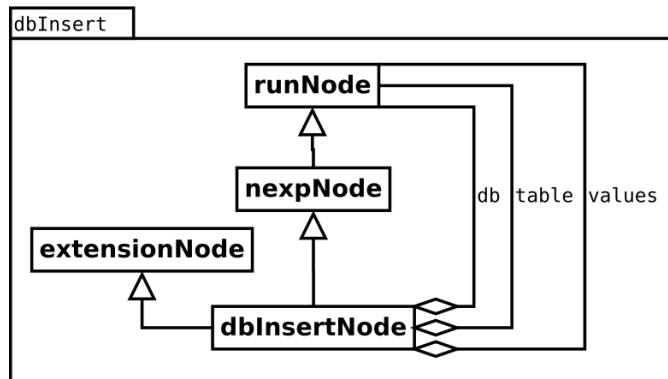
22.2.2. Consulta



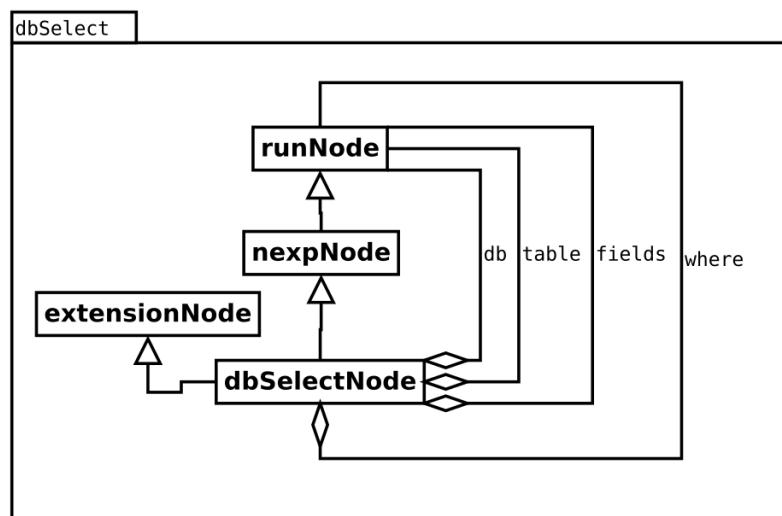
22.2.3. Cerrar conexión

Diagrama aún por realizar.

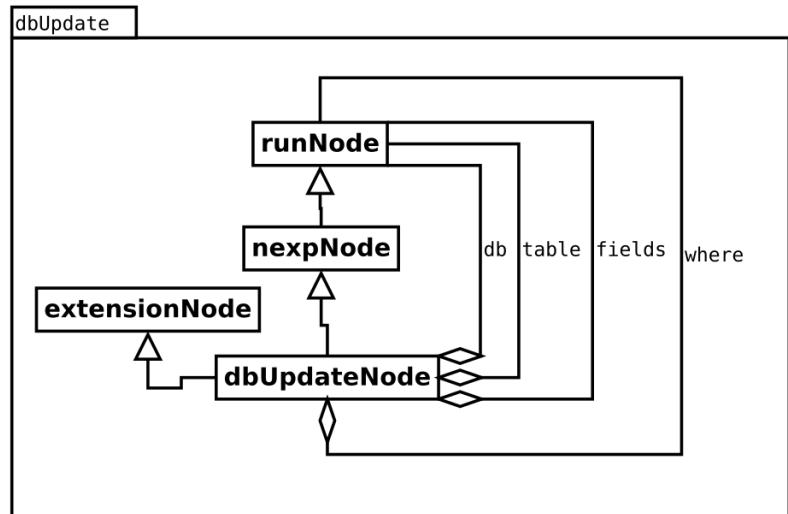
22.2.4. Insertar datos



22.2.5. Seleccionar datos



22.2.6. Actualizar datos



22.2.7. Eliminar datos

