Manual de Referencia de Lua 5.4

Por Roberto Ierusalimschy, Luiz Henrique de Figueiredo y Waldemar Celes

Traducido y Editado por:

Jesús Gabriel Rivera

En GitHub como `grChad`

Índice general

1

Introducción

Lua es un lenguaje de scripting poderoso, eficiente, liviano e integrable. Admite programación procedural, programación orientada a objetos, programación funcional, programación basada en datos y descripción de datos.

Lua combina una sintaxis procedural simple con potentes constructos de descripción de datos basados en arreglos asociativos y semánticas extensibles. Lua es de tipado dinámico, se ejecuta interpretando bytecode con una máquina virtual basada en registros y cuenta con una gestión automática de memoria con recolección de basura generacional, lo que lo hace ideal para configuración, scripting y prototipado rápido.

Lua se implementa como una biblioteca escrita en un `C` limpio, que es un subconjunto común de `C` estándar y `C++`. La distribución de Lua incluye un programa principal llamado "lua", que utiliza la biblioteca Lua para ofrecer un intérprete de Lua completo e independiente, ya sea para uso interactivo o por lotes. Lua está diseñado para ser utilizado tanto como un lenguaje de scripting poderoso, liviano e integrable para cualquier programa que lo necesite, como un lenguaje independiente poderoso, liviano y eficiente.

Como lenguaje de extensión, Lua no tiene noción de un programa "principal": trabaja integrado en un cliente hospedante, llamado programa de integración o simplemente el hospedante. (Frecuentemente, este hospedante es el programa independiente "lua".) El programa hospedante puede invocar funciones para ejecutar un fragmento de código Lua, puede escribir y leer variables Lua y puede registrar funciones en C para ser llamadas por código Lua. A través del uso de funciones en C, Lua puede ser ampliado para abordar una amplia gama

de dominios diferentes, creando así lenguajes de programación personalizados que comparten un marco sintáctico.

Lua es software libre y se proporciona sin garantías, como se establece en su licencia. La implementación descrita en este manual está disponible en el sitio web oficial de Lua, www.lua.org.

Al igual que cualquier otro manual de referencia, este documento puede resultar técnico en algunos lugares. Para obtener información sobre las decisiones detrás del diseño de Lua, consulte los documentos técnicos disponibles en el sitio web de Lua. Para obtener una introducción detallada a la programación en Lua, consulte el libro de Roberto, "Programming in Lua".

2

Conceptos Básicos

Esta sección describe los conceptos básicos del lenguaje.

2.1 Valores y Tipos

Lua es un lenguaje de tipado dinámico, lo que significa que las variables no tienen tipos, solo los valores los tienen. No existen definiciones de tipo en el lenguaje. Todos los valores en `Lua` son valores de primera clase, lo que significa que se pueden almacenar en variables, pasar como argumentos a otras funciones y devolver como resultados.

Hay ocho tipos básicos en Lua: nil, boolean, number, string, function, userdata, thread y table. El tipo nil tiene un único valor, nil, cuya principal propiedad es ser diferente de cualquier otro valor; a menudo representa la ausencia de un valor útil. El tipo boolean tiene dos valores, false y true. Tanto nil como false se consideran valores falsos; se les denomina colectivamente valores falsos. Cualquier otro valor se considera verdadero. A pesar de su nombre, false se utiliza frecuentemente como una alternativa a nil, con la diferencia clave de que false se comporta como un valor regular en una tabla, mientras que un nil en una tabla representa una clave ausente.

El tipo number representa tanto números enteros como números reales (punto flotante), utilizando dos subtipos: integer (entero) y float (flotante). `Lua` estándar utiliza enteros de 64 bits y flotantes de doble precisión (64 bits), pero también puedes compilar `Lua` para que utilice enteros de 32 bits y/o flotantes

de precisión simple (32 bits). La opción de 32 bits tanto para enteros como para flotantes es particularmente atractiva para máquinas pequeñas y sistemas integrados. (Consulta la macro LUA32BITS en el archivo luaconf.h).

A menos que se indique lo contrario, cualquier desbordamiento al manipular valores enteros realiza un ajuste, según las reglas habituales de la aritmética de complemento a dos. (En otras palabras, el resultado real es el único entero representable que es igual módulo 2n al resultado matemático, donde n es el número de bits del tipo entero).

Lua tiene reglas explícitas sobre cuándo se usa cada subtipo, pero también realiza conversiones automáticas entre ellos según sea necesario (consulte la sección 3.4.3). Por lo tanto, el programador puede optar por ignorar en su mayoría la diferencia entre enteros y flotantes, o asumir un control completo sobre la representación de cada número.

El tipo string representa secuencias inmutables de bytes. `Lua` es compatible con 8 bits: las cadenas pueden contener cualquier valor de 8 bits, incluidos ceros incrustados ('0'). `Lua` también es agnóstico con respecto a la codificación; no hace suposiciones sobre el contenido de una cadena. La longitud de cualquier cadena en `Lua` debe ajustarse a un entero de Lua.

Lua puede llamar (y manipular) funciones escritas en `Lua` y funciones escritas en `C` (consulte la sección 3.4.10). Ambas se representan mediante el tipo function.

El tipo userdata se proporciona para permitir que los datos arbitrarios de `C` se almacenen en variables de Lua. Un valor userdata representa un bloque de memoria sin procesar. Hay dos tipos de userdata: userdata completo, que es un objeto con un bloque de memoria administrado por Lua, y userdata ligero, que es simplemente un valor de puntero C. Los userdata no tienen operaciones predefinidas en Lua, excepto la asignación y la prueba de identidad. Mediante el uso de metatablas, el programador puede definir operaciones para valores de userdata completos (consulte la sección 2.4). Los valores userdata no se pueden crear ni modificar en Lua, solo a través de la API de C. Esto garantiza la integridad de los datos propiedad del programa hospedante y las bibliotecas de C.

El tipo thread representa hilos de ejecución independientes y se utiliza para implementar coroutines (consulte la sección 2.6). Los hilos de `Lua` no están relacionados con los hilos del sistema operativo. `Lua` admite coroutines en todos los sistemas, incluso aquellos que no admiten hilos nativamente.

El tipo table implementa arreglos asociativos, es decir, arreglos que pueden tener como índices no solo números, sino cualquier valor de Lua excepto nil y NaN. (Not a Number es un valor especial de punto flotante utilizado por la norma IEEE 754 para representar resultados numéricos indefinidos, como 0/0). Las tablas pueden ser heterogéneas; es decir, pueden contener valores de todos los tipos (excepto nil). Cualquier clave asociada al valor nil no se considera parte de la tabla. A su vez, cualquier clave que no sea parte de una tabla tiene un valor asociado nil.

Las tablas son el único mecanismo de estructuración de datos en Lua; se pueden usar para representar arreglos ordinarios, listas, tablas de símbolos, conjuntos, registros, gráficos, árboles, etc. Para representar registros, Lua utiliza el nombre del campo como índice. El lenguaje admite esta representación proporcionando a.name como azúcar sintáctica para a["name"]. Hay varias formas convenientes de crear tablas en Lua (consulte la sección 3.4.9).

Al igual que los índices, los valores de los campos de la tabla pueden ser de cualquier tipo. En particular, como las funciones son valores de primera clase, los campos de la tabla pueden contener funciones. Por lo tanto, las tablas también pueden llevar métodos (consulte la sección 3.4.11).

La indexación de tablas sigue la definición de igualdad cruda en el lenguaje. Las expresiones a[i] y a[j] denotan el mismo elemento de la tabla si y solo si i y j son iguales en bruto (es decir, iguales sin metamétodos). En particular, los flotantes con valores enteros son iguales a sus respectivos enteros (por ejemplo, $1.0 == 1$). Para evitar ambigüedades, cualquier flotante utilizado como clave que sea igual a un entero se convierte en ese entero. Por ejemplo, si escribes $a[2.0] = \text{true}$, la clave real insertada en la tabla será el entero 2.

Las tablas, funciones, hilos y valores de userdata (completos) son objetos: las variables no contienen realmente estos valores, solo referencias a ellos. La asignación, el paso de parámetros y la devolución de funciones siempre manipulan referencias a tales valores; estas operaciones no implican ningún tipo de copia.

La función de biblioteca type devuelve una cadena que describe el tipo de un valor dado (consulte type).

2.2 Entornos y el Entorno Global

Como se discutirá más adelante en las secciones 3.2 y 3.3.3, cualquier referencia a un nombre libre (es decir, un nombre no vinculado a ninguna declaración) var se traduce sintácticamente a `_ENV.var`. Además, cada fragmento se compila en el ámbito de una variable local externa llamada `_ENV` (consulte la sección 3.3.2), por lo que `_ENV` en sí nunca es un nombre libre en un fragmento.

A pesar de la existencia de esta variable externa `_ENV` y la traducción de los nombres libres, `_ENV` es un nombre completamente regular. En particular, puedes definir nuevas variables y parámetros con ese nombre. Cada referencia a un nombre libre utiliza el `_ENV` que es visible en ese punto del programa, siguiendo las reglas habituales de visibilidad de `Lua` (consulte la sección 3.5).

Cualquier tabla utilizada como valor de `_ENV` se llama entorno (environment).

Lua mantiene un entorno distinguido llamado entorno global. Este valor se guarda en un índice especial en el registro `C` (consulte la sección 4.3). En Lua, la variable global `_G` se inicializa con este mismo valor. (`_G` no se utiliza internamente, por lo que cambiar su valor solo afectará a tu propio código).

Cuando `Lua` carga un fragmento, el valor predeterminado para su variable `_ENV` es el entorno global (consulte load). Por lo tanto, de forma predeterminada, los nombres libres en el código `Lua` se refieren a entradas en el entorno global y, por lo tanto, también se denominan variables globales. Además, todas las bibliotecas estándar se cargan en el entorno global y algunas funciones operan en ese entorno. Puedes usar load (o loadfile) para cargar un fragmento con un entorno diferente. (En `C`, debes cargar el fragmento y luego cambiar el valor de su primer upvalue; consulta lua_setupvalue).

2.3 Manejo de Errores

Varias operaciones en `Lua` pueden generar un error. Un error interrumpe el flujo normal del programa, pero se puede continuar capturando el error.

El código `Lua` puede generar explícitamente un error llamando a la función error. (Esta función nunca retorna).

Para capturar errores en Lua, puedes hacer una llamada protegida utilizando pcall (o xpcall). La función pcall llama a una función dada en modo protegi-

do. Cualquier error que ocurra durante la ejecución de la función detiene su ejecución y el control vuelve inmediatamente a pcall, que devuelve un código de estado.

Como Lua es un lenguaje de extensión incrustado, el código Lua comienza a ejecutarse mediante una llamada desde el código C del programa hospedante. (Cuando usas Lua independiente, la aplicación lua es el programa hospedante). Por lo general, esta llamada está protegida; por lo tanto, cuando ocurre un error no protegido durante la compilación o ejecución de un fragmento Lua, el control vuelve al programa hospedante, que puede tomar medidas apropiadas, como imprimir un mensaje de error.

Cuando ocurre un error, se propaga un objeto de error con información sobre el error. Lua mismo solo genera errores cuyo objeto de error es una cadena, pero los programas pueden generar errores con cualquier valor como objeto de error. Depende del programa Lua o de su programa hospedante manejar dichos objetos de error. Por razones históricas, a menudo se llama mensaje de error a un objeto de error, aunque no tiene que ser una cadena.

Cuando usas xpcall (o `lua_pcall` en C), puedes proporcionar un manejador de mensajes que se llamará en caso de errores. Esta función se llama con el objeto de error original y devuelve un nuevo objeto de error. Se llama antes de desenrollar la pila de llamadas del error, por lo que puede recopilar más información sobre el error, por ejemplo, inspeccionando la pila y creando un seguimiento de la pila (stack traceback). Este manejador de mensajes también está protegido por la llamada protegida; por lo tanto, si ocurre un error dentro del manejador de mensajes, se volverá a llamar al manejador de mensajes. Si este ciclo continúa durante demasiado tiempo, Lua lo interrumpe y devuelve un mensaje apropiado. El manejador de mensajes solo se llama para errores de tiempo de ejecución regulares. No se llama para errores de asignación de memoria ni para errores durante la ejecución de finalizadores u otros manejadores de mensajes.

Lua también ofrece un sistema de advertencias (ver warn). A diferencia de los errores, las advertencias no interfieren en absoluto con la ejecución del programa. Por lo general, solo generan un mensaje al usuario, aunque este comportamiento se puede adaptar desde C (ver lua_setwarnf).

2.4 Metatablas y Metamétodos

Cada valor en Lua puede tener una metatabla. Esta metatabla es una tabla Lua ordinaria que define el comportamiento del valor original en ciertos eventos. Puedes cambiar varios aspectos del comportamiento de un valor estableciendo campos específicos en su metatabla. Por ejemplo, cuando un valor no numérico es el operando de una suma, Lua busca una función en el campo __add de la metatabla del valor. Si encuentra una, Lua llama a esta función para realizar la suma.

La clave para cada evento en una metatabla es una cadena con el nombre del evento precedido por dos guiones bajos; el valor correspondiente se llama metavalue. Para la mayoría de los eventos, el metavalue debe ser una función, que luego se llama metamétodo. En el ejemplo anterior, la clave es la cadena "__add" el metamétodo es la función que realiza la suma. A menos que se indique lo contrario, un metamétodo puede ser en realidad cualquier valor invocable, que puede ser una función o un valor con un metamétodo __call.

Puedes consultar la metatabla de cualquier valor utilizando la función getmetatable. Lua consulta metamétodos en las metatablas utilizando un acceso directo (ver rawget).

Puedes reemplazar la metatabla de las tablas utilizando la función setmetatable. No puedes cambiar la metatabla de otros tipos desde el código Lua, excepto utilizando la biblioteca debug (sección 6.10).

Las tablas y los userdata completos tienen metatablas individuales, aunque múltiples tablas y userdata pueden compartir sus metatablas. Los valores de todos los demás tipos comparten una única metatabla por tipo; es decir, hay una única metatabla para todos los números, una para todas las cadenas, etc. Por defecto, un valor no tiene metatabla, pero la biblioteca de cadenas establece una metatabla para el tipo de cadena (ver sección 6.4).

A continuación se presenta una lista detallada de las operaciones controladas por las metatablas. Cada evento se identifica por su clave correspondiente. Por convención, todas las claves de metatablas utilizadas por Lua están compuestas por dos guiones bajos seguidos de letras minúsculas del alfabeto latino.

- **__add:** La operación de suma (+). Si alguno de los operandos de una suma no es un número, Lua intentará llamar a un metamétodo. Co-

mienza revisando el primer operando (incluso si es un número); si ese operando no define un metamétodo para __add, entonces Lua revisará el segundo operando. Si Lua encuentra un metamétodo, llama al metamétodo con los dos operandos como argumentos, y el resultado de la llamada (ajustado a un solo valor) es el resultado de la operación. De lo contrario, si no se encuentra ningún metamétodo, Lua genera un error.

- **__sub:** La operación de resta (-) tiene un comportamiento similar a la operación de suma.
- **__mul:** La operación de multiplicación (*) tiene un comportamiento similar a la operación de suma.
- **__div:** La operación de division (/) tiene un comportamiento similar a la operación de suma.
- **__mod:** El operador modulo (%) tiene un comportamiento similar a la operación de suma.
- **__pow:** La operación de exponentiación (^) tiene un comportamiento similar a la operación de suma.
- **__unm:** La operación de negación (unario -) tiene un comportamiento similar a la operación de suma.
- **__idiv:** La operación de división entera (//) tiene un comportamiento similar a la operación de suma.
- **__band:** La operación de bitwise AND (&) tiene un comportamiento similar a la operación de suma, excepto que Lua intentará utilizar un metamétodo si alguno de los operandos no es un número entero o un número de punto flotante que pueda convertirse a un entero (consulte la sección 3.4.3).
- **__bor:** La operación de bitwise OR (|) tiene un comportamiento similar a la operación de bitwise AND.
- **__bxor:** La operación de bitwise exclusive OR (~) tiene un comportamiento similar a la operación de bitwise AND.
- **__bnot:** La operación de bitwise NOT (~) tiene un comportamiento similar a la operación de bitwise AND.
- **__shl:** La operación de desplazamiento a la izquierda («) tiene un comportamiento similar a la operación de bitwise AND.

- **__shr:** La operación de desplazamiento a la derecha (») tiene un comportamiento similar a la operación de bitwise AND.
- **__concat:** La operación de concatenación (..) tiene un comportamiento similar a la operación de suma. Si alguno de los operandos de la concatenación no es una cadena o un número (que siempre puede convertirse a una cadena).
- **__len:** La operación de longitud (#) tiene el siguiente comportamiento: si el objeto no es una cadena, Lua intentará utilizar su metamétodo. Si existe un metamétodo, Lua lo llama con el objeto como argumento y el resultado de la llamada (siempre ajustado a un solo valor) es el resultado de la operación. Si no hay metamétodo pero el objeto es una tabla, entonces Lua utiliza la operación de longitud de la tabla (ver sección 3.4.7). De lo contrario, Lua genera un error.
- **__eq:** La operación de igualdad (==) tiene un comportamiento similar a la operación de suma. Sin embargo, Lua intentará utilizar un metamétodo solo cuando los valores que se están comparando sean ambas tablas o ambos userdata completos, y no sean primitivamente iguales. El resultado de la llamada siempre se convierte en un valor booleano.
- **__lt:** La operación de menor que (<) tiene un comportamiento similar a la operación de suma. Sin embargo, Lua intentará utilizar un metamétodo solo cuando los valores que se están comparando no sean ambos números ni ambas cadenas. Además, el resultado de la llamada siempre se convierte en un valor booleano.
- **__le:** La operación de menor o igual que (<=) tiene un comportamiento similar a la operación de menor que (<).
- **__index:** La operación de acceso de índice `table[key]` ocurre cuando `table` no es una tabla o cuando `key` no está presente en `table`. El metavalue se busca en la metatabla de `table`.

  El metavalue para este evento puede ser una función, una tabla o cualquier valor con un metavalue `__index`. Si es una función, se llama con `table` y `key` como argumentos, y el resultado de la llamada (ajustado a un solo valor) es el resultado de la operación. De lo contrario, el resultado final es el resultado de indexar este metavalue con `key`. Esta indexación es regular, no raw, por lo tanto puede activar otro metavalue `__index`.

- **__newindex:** La asignación de índice `table[key] = value` ocurre cuando `table` no es una tabla o cuando `key` no está presente en `table`. El metavalue se busca en la metatabla de `table`.

  Al igual que con la indexación, el metavalue para este evento puede ser una función, una tabla o cualquier valor con un metavalue `__newindex`. Si es una función, se llama con `table`, `key` y `value` como argumentos. De lo contrario, `Lua` repite la asignación de índice sobre este metavalue con la misma key y value. Esta asignación es regular, no raw, por lo tanto puede activar otro metavalue `__newindex`.

  Cuando se invoca un metavalue `__newindex`, `Lua` no realiza la asignación primitiva. Si es necesario, el metamétodo en sí puede llamar a `rawset` para realizar la asignación.

- **__call:** La operación de llamada `func(args)` ocurre cuando `Lua` intenta llamar a un valor que no es una función (es decir, `func` no es una función). Se busca el metamétodo en `func`. Si está presente, se llama al metamétodo con `func` como primer argumento, seguido de los argumentos de la llamada original (`args`). Todos los resultados de la llamada son los resultados de la operación. Este es el único metamétodo que permite múltiples resultados.

Además de la lista anterior, el intérprete también respeta las siguientes claves en las metatablas: `__gc` (ver sección 2.5.3), `__close` (ver sección 3.3.8), `__mode` (ver sección 2.5.4) y `__name`. (La entrada `__name`, cuando contiene una cadena, puede ser utilizada por `tostring` y en mensajes de error).

Para los operadores unarios (negación, longitud y bitwise NOT), el metamétodo se calcula y se llama con un segundo operando ficticio igual al primero. Este operando adicional solo se utiliza para simplificar los aspectos internos de `Lua` (haciendo que estos operadores se comporten como una operación binaria) y puede ser eliminado en futuras versiones. Para la mayoría de los casos de uso, este operando adicional no es relevante.

Dado que las metatablas son tablas regulares, pueden contener campos arbitrarios, no solo los nombres de eventos mencionados anteriormente. Algunas funciones de la biblioteca estándar (por ejemplo, `tostring`) utilizan otros campos en las metatablas para sus propios propósitos.

Es una buena práctica agregar todos los metamétodos necesarios a una tabla

antes de establecerla como metatabla de algún objeto. En particular, el metamétodo `__gc` funciona solo cuando se sigue este orden (ver sección 2.5.3). También es una buena práctica establecer la metatabla de un objeto justo después de su creación.

2.5 Recolección de Basura

Lua realiza la gestión automática de memoria. Esto significa que no tienes que preocuparte por asignar memoria para nuevos objetos ni liberarla cuando los objetos ya no sean necesarios. `Lua` administra la memoria automáticamente ejecutando un *recolector de basura o (Garbage Collector)* para recoger todos los objetos muertos. Toda la memoria utilizada por `Lua` está sujeta a una gestión automática: cadenas, tablas, userdata, funciones, hilos, estructuras internas, etc.

Un objeto se considera muerto en cuanto el recolector puede estar seguro de que el objeto no será accedido nuevamente en la ejecución normal del programa. (.Ejecución normal.ªquí excluye los finalizadores, que pueden resucitar objetos muertos (ver sección 2.5.3), y también excluye las operaciones que utilizan la biblioteca de depuración). Es importante tener en cuenta que el momento en que el recolector puede estar seguro de que un objeto está muerto puede no coincidir con las expectativas del programador. La única garantía es que `Lua` no recogerá un objeto que aún pueda ser accedido en la ejecución normal del programa, y eventualmente recogerá un objeto que sea inaccesible desde Lua. (Aquí, *inaccesible desde Lua* significa que ni una variable ni otro objeto vivo hacen referencia al objeto). Debido a que `Lua` no tiene conocimiento sobre el código `C`, nunca recoge objetos accesibles a través del registro (ver sección 4.3), que incluye el entorno global (ver sección 2.2).

El recolector de basura (GC o Garbage Collector) en `Lua` puede trabajar en dos modos: incremental y generacional.

El modo GC predeterminado con los parámetros predeterminados es adecuado para la mayoría de los casos de uso. Sin embargo, los programas que desperdician una gran proporción de su tiempo asignando y liberando memoria pueden beneficiarse de otros ajustes. Ten en cuenta que el comportamiento del GC no es portátil tanto entre plataformas como entre diferentes versiones de Lua; por lo tanto, los ajustes óptimos tampoco son portátiles.

Puedes cambiar el modo y los parámetros del GC llamando a `lua_gc` en `C` o `collectgarbage` en Lua. También puedes usar estas funciones para controlar el recolector directamente (por ejemplo, para detenerlo y reiniciarlo).

2.5.1. Recolección de Basura Incremental

En el modo incremental, cada ciclo del recolector de basura realiza una colección de marcas y barrido en pequeños pasos entrelazados con la ejecución del programa. En este modo, el recolector utiliza tres números para controlar sus ciclos de recolección de basura: el *garbage-collector pause*, el *garbage-collector step multiplier* y el *garbage-collector step size*.

El `garbage-collector pause` controla cuánto tiempo espera el recolector antes de comenzar un nuevo ciclo. El recolector inicia un nuevo ciclo cuando el uso de memoria alcanza $n\%$ del uso después de la recolección anterior. Valores más grandes hacen que el recolector sea menos agresivo. Los valores iguales o inferiores a 100 significan que el recolector no esperará para comenzar un nuevo ciclo. Un valor de 200 significa que el recolector espera que el uso total de memoria se duplique antes de comenzar un nuevo ciclo. El valor predeterminado es 200; el valor máximo es 1000.

El `garbage-collector step multiplier` controla la velocidad del recolector en relación con la asignación de memoria, es decir, cuántos elementos marca o barre por cada kilobyte de memoria asignada. Valores más grandes hacen que el recolector sea más agresivo, pero también aumentan el tamaño de cada paso incremental. No se deben usar valores inferiores a 100, porque hacen que el recolector sea demasiado lento y pueden hacer que el recolector nunca termine un ciclo. El valor predeterminado es 100; el valor máximo es 1000.

El `garbage-collector step sizes` controla el tamaño de cada paso incremental, específicamente cuántos bytes asigna el intérprete antes de realizar un paso. Este parámetro es logarítmico: un valor de n significa que el intérprete asignará 2^n bytes entre pasos y realizará un trabajo equivalente durante el paso. Un valor grande (por ejemplo, 60) hace que el recolector sea un recolector de detención del mundo (no incremental). El valor predeterminado es 13, lo que significa pasos de aproximadamente 8 Kbytes.

2.5.2. Recolección de Basura Generacional

En el modo generacional, el recolector realiza colecciones menores frecuentes, que recorren solo los objetos creados recientemente. Si después de una colección menor el uso de memoria sigue por encima de un límite, el recolector realiza una colección mayor de detención del mundo, que recorre todos los objetos. El modo generacional utiliza dos parámetros: el *minor multiplier* y el *major multiplier*.

El `minor multiplier` controla la frecuencia de las colecciones menores. Para un multiplicador menor x, se realizará una nueva colección menor cuando la memoria crezca $x\,\%$ más grande que la memoria en uso después de la colección mayor anterior. Por ejemplo, para un multiplicador de 20, el recolector realizará una colección menor cuando el uso de memoria sea un 20 % más grande que el uso después de la colección mayor anterior. El valor predeterminado es 20; el valor máximo es 200.

El `major multiplier` controla la frecuencia de las colecciones mayores. Para un multiplicador mayor x, se realizará una nueva colección mayor cuando la memoria crezca $x\,\%$ más grande que la memoria en uso después de la colección mayor anterior. Por ejemplo, para un multiplicador de 100, el recolector realizará una colección mayor cuando el uso de memoria sea más grande que el doble del uso después de la colección anterior. El valor predeterminado es 100; el valor máximo es 1000.

2.5.3. Metamétodos de Recolección de Basura

Puedes establecer metamétodos del recolector de basura (finalizadores) para tablas y, utilizando la API de `C`, para full userdata (ver sección 2.4). Estos finalizadores son llamados cuando el recolector de basura de `Lua` detecta que la tabla o userdata correspondiente está marcada como inaccesible. Los finalizadores te permiten coordinar la recolección de basura de `Lua` con la gestión externa de recursos, como cerrar archivos, conexiones de red o bases de datos, o liberar tu propia memoria.

Para que un objeto (tabla o userdata) sea finalizado cuando se recolecta, debes marcarlo para finalización. Marcas un objeto para finalización cuando le asignas su metatabla y la metatabla tiene un metamétodo `__gc`. Ten en cuenta que si estableces una metatabla sin un campo `__gc` y más tarde creas ese campo

en la metatabla, el objeto no se marcará para finalización.

Cuando un objeto marcado se vuelve inaccesible, no es recolectado inmediatamente por el recolector de basura de Lua. En su lugar, Lua lo pone en una lista. Después de la recolección, Lua recorre esa lista. Para cada objeto en la lista, verifica el metamétodo `__gc` del objeto: Si está presente, Lua lo llama con el objeto como su único argumento.

Al final de cada ciclo de recolección de basura, los finalizadores son llamados en orden inverso a cómo los objetos fueron marcados para finalización, entre los objetos recolectados en ese ciclo; es decir, el primer finalizador en ser llamado es aquel asociado al objeto marcado en último lugar en el programa. La ejecución de cada finalizador puede ocurrir en cualquier momento durante la ejecución del código regular.

Dado que el objeto que está siendo recolectado todavía debe ser utilizado por el finalizador, ese objeto (y otros objetos accesibles solo a través de él) deben ser resucitados por Lua. Por lo general, esta resurrección es transitoria, y la memoria del objeto se libera en el siguiente ciclo de recolección de basura. Sin embargo, si el finalizador almacena el objeto en algún lugar global (por ejemplo, una variable global), entonces la resurrección es permanente. Además, si el finalizador marca un objeto en proceso de finalización para finalización nuevamente, su finalizador será llamado nuevamente en el siguiente ciclo donde el objeto esté inaccesible. En cualquier caso, la memoria del objeto se libera solo en un ciclo de recolección de basura donde el objeto esté inaccesible y no marcado para finalización.

Cuando cierras un estado de Lua (ver `lua_close`), Lua llama a los finalizadores de todos los objetos marcados para finalización, siguiendo el orden inverso en que fueron marcados. Si algún finalizador marca objetos para recolección durante esa fase, esas marcas no tienen efecto.

Los finalizadores no pueden pausar ni ejecutar el recolector de basura. Debido a que pueden ejecutarse en momentos impredecibles, es una buena práctica restringir cada finalizador al mínimo necesario para liberar adecuadamente su recurso asociado.

Cualquier error mientras se ejecuta un finalizador genera una advertencia; el error no se propaga.

2.5.4. Tablas Débiles

Una tabla débil (weak table) es una tabla cuyos elementos son referencias débiles. Una referencia débil es ignorada por el recolector de basura. En otras palabras, si las únicas referencias a un objeto son referencias débiles, entonces el recolector de basura recolectará ese objeto.

Una tabla débil puede tener claves débiles, valores débiles, o ambas. Una tabla con valores débiles permite la recolección de sus valores, pero evita la recolección de sus claves. Una tabla con claves débiles y valores débiles permite la recolección tanto de claves como de valores. En cualquier caso, si tanto la clave como el valor son recolectados, el par completo se elimina de la tabla. La debilidad de una tabla está controlada por el campo `__mode` de su metatabla. Este metavalue, si está presente, debe ser una de las siguientes cadenas: *"k"*, para una tabla con claves débiles; *"v"*, para una tabla con valores débiles; o *"kv"*, para una tabla con ambas claves y valores débiles.

Una tabla con claves débiles y valores fuertes también se llama tabla efímera (ephemeron). En una tabla efímera, se considera que un valor es alcanzable solo si su clave es alcanzable. En particular, si la única referencia a una clave proviene de su valor, el par se elimina.

Cualquier cambio en la debilidad de una tabla puede surtir efecto solo en el próximo ciclo de recolección. En particular, si cambias la debilidad a un modo más fuerte, `Lua` puede seguir recolectando algunos elementos de esa tabla antes de que el cambio surta efecto.

Solo los objetos que tienen una construcción explícita se eliminan de las tablas débiles. Los valores, como los números y las funciones `C` ligeras, no están sujetos a la recolección de basura y, por lo tanto, no se eliminan de las tablas débiles (a menos que se recolecten sus valores asociados). Aunque las cadenas están sujetas a la recolección de basura, no tienen una construcción explícita y su igualdad es por valor; se comportan más como valores que como objetos. Por lo tanto, no se eliminan de las tablas débiles.

Los objetos resucitados (es decir, objetos en proceso de finalización y objetos accesibles solo a través de objetos en proceso de finalización) tienen un comportamiento especial en las tablas débiles. Se eliminan de los valores débiles antes de ejecutar sus finalizadores, pero solo se eliminan de las claves débiles en la próxima recolección después de ejecutar sus finalizadores, cuando estos objetos son liberados efectivamente. Este comportamiento permite que

el finalizador acceda a propiedades asociadas con el objeto a través de tablas débiles.

Si una tabla débil se encuentra entre los objetos resucitados en un ciclo de recolección, es posible que no se limpie correctamente hasta el siguiente ciclo.

2.6 Coroutines

Lua admite coroutines, también llamadas *collaborative multithreading*. Una coroutine en Lua representa un hilo de ejecución independiente. Sin embargo, a diferencia de los hilos en sistemas de multiprocesamiento, una coroutine solo suspende su ejecución mediante una llamada explícita a la función yield.

Se crea una coroutine llamando a `coroutine.create`. Su único argumento es una función que es la función principal de la coroutine. La función create solo crea una nueva coroutine y devuelve un identificador para ella (un objeto de *type thread*); no inicia la coroutine.

Se ejecuta una coroutine llamando a `coroutine.resume`. Cuando se llama por primera vez a `coroutine.resume`, pasando como primer argumento un hilo devuelto por `coroutine.create`, la coroutine comienza su ejecución llamando a su función principal. Los argumentos adicionales pasados a `coroutine.resume` se pasan como argumentos a esa función. Después de que la coroutine comienza a ejecutarse, se ejecuta hasta que termina o hace un *yield*.

Una coroutine puede terminar su ejecución de dos maneras: de manera normal, cuando su función principal retorna (explícitamente o implícitamente, después de la última instrucción); y de manera anormal, si hay un error no protegido. En caso de terminación normal, `coroutine.resume` devuelve **true**, más cualquier valor devuelto por la función principal de la coroutine. En caso de errores, `coroutine.resume` devuelve **false** más el objeto de error. En este caso, la coroutine no desenrolla su pila, por lo que es posible inspeccionarla después del error con la API de depuración.

Una coroutine hace un yield llamando a `coroutine.yield`. Cuando una coroutine hace un yield, la llamada correspondiente a `coroutine.resume` devuelve inmediatamente, incluso si el yield ocurre dentro de llamadas de función anidadas (es decir, no en la función principal, sino en una función llamada directa o indirectamente por la función principal). En el caso de un

yield, `coroutine.resume` también devuelve **true**, más cualquier valor pasado a `coroutine.yield`. La próxima vez que se reanuda la misma coroutine, continúa su ejecución desde el punto donde hizo el yield, y la llamada a `coroutine.yield` devuelve cualquier argumento adicional pasado a `coroutine.resume`.

Al igual que `coroutine.create`, la función `coroutine.wrap` también crea una coroutine, pero en lugar de devolver la coroutine en sí misma, devuelve una función que, cuando se llama, reanuda la coroutine. Cualquier argumento pasado a esta función se pasa como argumentos adicionales a `coroutine.resume`. `coroutine.wrap` devuelve todos los valores devueltos por `coroutine.resume`, excepto el primero (el código de error booleano). A diferencia de `coroutine.resume`, la función creada por `coroutine.wrap` propaga cualquier error al llamador. En este caso, la función también cierra la coroutine (ver `coroutine.close`).

Como ejemplo de cómo funcionan las coroutines, considera el siguiente código:

```
function foo()
  print("foo", a)
  return coroutine.yield(2 * a)
end

co = coroutine.create(function(a, b)
  print("co-boby", a, b)
  local r = foo(a + 1)
  print("co-boby", r)
  local r, s = coroutine.yield(a + b, a - b)
  print("co-boby", r, s)
  return b, "end"
end)

print("main", coroutine.resume(co, 1, 10))
print("main", coroutine.resume(co, "r"))
print("main", coroutine.resume(co, "x", "y"))
print("main", coroutine.resume(co, "x", "y"))
```

Cuando lo ejecutas, produce la siguiente salida:

```
co-body 1       10
foo     2
main    true    4
co-body r
main    true    11      -9
co-body x       y
main    true    10      end
main    false   cannot resume dead coroutine
```

También puedes crear y manipular coroutines a través de la API de C: consulta las funciones `lua_newthread`, `lua_resume` y `lua_yield`.

3

El Lenguaje

Esta sección describe el léxico, la sintaxis y la semántica de Lua. En otras palabras, esta sección describe qué tokens son válidos, cómo pueden combinarse y qué significan sus combinaciones.

Los constructos del lenguaje se explicarán utilizando la notación BNF extendida habitual, en la que `{a}` significa 0 o más `a's`, y `[a]` significa un `a` opcional. Los no terminales se muestran como non-terminal, las palabras clave se muestran como **kword** y otros símbolos terminales se muestran como '='. La sintaxis completa de Lua se puede encontrar en la sección 9 al final de este manual.

3.1 Convenciones léxicas

Lua es un lenguaje de formato libre. Ignora los espacios y los comentarios entre elementos léxicos (tokens), excepto como delimitadores entre dos tokens. En el código fuente, Lua reconoce como espacios los caracteres de espacio en blanco ASCII estándar: espacio, avance de formulario, nueva línea, retorno de carro, tabulación horizontal y tabulación vertical.

Los nombres (también llamados identificadores) en Lua pueden ser cualquier cadena de letras latinas, dígitos arábigos y guiones bajos, que no comience con un dígito y no sea una palabra reservada. Los identificadores se utilizan para nombrar variables, campos de tablas y etiquetas.

Las siguientes palabras clave están reservadas y no pueden ser utilizadas como nombres:

```
and     break   do        else    elseif   end
false   for     function  goto    if       in
local   nil     not       or      repeat   return
then    true    until     while
```

Lua es un lenguaje sensible a mayúsculas y minúsculas: and es una palabra reservada, pero 'And' y 'AND' son dos nombres diferentes y válidos. Como convención, los programas deben evitar crear nombres que comiencen con un guión bajo seguido de una o más letras mayúsculas (como _VERSION).

Las siguientes cadenas de texto representan otros tokens:

```
+       -       *       /       %       ^       #
&       ~       |       <<      >>      //
==      ~=      <=      >=      <       >       =
(       )       {       }       [       ]       ::
;       :       ,       .       ..      ...
```

Una cadena de texto corta (literal) puede estar delimitada por comillas simples o comillas dobles y puede contener las siguientes secuencias de escape similares a las de C: '\a' (campana), '\b' (backspace), '\f' (avance de página), '\n' (nueva línea), '\r' (retorno de carro), '\t' (tabulación horizontal), '\v' (tabulación vertical), '\\' (backslash), '\"' (comilla doble) y '\'' (comilla simple). Si un carácter barra invertida se sigue de un salto de línea, esto resulta en una nueva línea en la cadena de texto. La secuencia de escape '\z' omite el siguiente conjunto de caracteres de espacio en blanco, incluyendo saltos de línea; esto es especialmente útil para dividir e indentar una cadena de texto larga en varias líneas sin agregar los saltos de línea y espacios en el contenido de la cadena. Una cadena de texto corta no puede contener saltos de línea no escapados ni secuencias de escape que no formen una secuencia de escape válida.

En una cadena de texto corta (literal), podemos especificar cualquier byte, incluso ceros incrustados, mediante su valor numérico. Esto se puede hacer utilizando la secuencia de escape \xXX, donde XX es una secuencia de exactamente dos dígitos hexadecimales, o con la secuencia de escape \ddd, donde ddd es una secuencia de hasta tres dígitos decimales. (Tenga en cuenta que si una secuencia de escape decimal debe ir seguida de un dígito, debe expresarse

con exactamente tres dígitos).

La codificación UTF-8 de un carácter Unicode se puede insertar en una cadena de texto corta (literal) con la secuencia de escape `\u{XXX}` (con las llaves obligatorias), donde `XXX` es una secuencia de uno o más dígitos hexadecimales que representan el punto de código del carácter. Este punto de código puede tener cualquier valor menor que 2^{31}. (`Lua` utiliza la especificación original de UTF-8 aquí, que no está restringida a puntos de código Unicode válidos).

Las cadenas de texto literales también se pueden definir utilizando un formato largo, que está encerrado por corchetes largos. Definimos un corchete largo de apertura de nivel n como un corchete cuadrado de apertura seguido de n signos de igual seguidos por otro corchete cuadrado de apertura. Así, un corchete largo de apertura de nivel 0 se escribe como `[[`, un corchete largo de apertura de nivel 1 se escribe como `[=[`, y así sucesivamente. Un corchete largo de cierre se define de manera similar; por ejemplo, un corchete largo de cierre de nivel 4 se escribe como `]====]`. Una cadena larga comienza con un corchete largo de apertura de cualquier nivel y termina en el primer corchete largo de cierre del mismo nivel. Puede contener cualquier texto excepto un corchete de cierre del mismo nivel. Las cadenas literales en esta forma entre corchetes pueden extenderse por varias líneas, no interpretan secuencias de escape y ignoran corchetes largos de cualquier otro nivel. Cualquier tipo de secuencia de fin de línea (retorno de carro, nueva línea, retorno de carro seguido de nueva línea o nueva línea seguida de retorno de carro) se convierte en una simple nueva línea. Cuando el corchete largo de apertura está inmediatamente seguido de una nueva línea, la nueva línea no se incluye en la cadena.

Como ejemplo, en un sistema que utiliza ASCII (en el cual '`a`' está codificado como 97, newline está codificada como 10 y '`1`' está codificado como 49), las cinco cadenas literales siguientes denotan la misma cadena:

```
a = 'alo\n123"'
a = "alo\n123\""
a = '\97lo\10\04923"'
a = [[alo
123"]]
a = [==[
alo
123"]==]
```

Cualquier byte en una cadena literal que no esté explícitamente afectado por las reglas anteriores se representa tal cual. Sin embargo, `Lua` abre archivos

para su análisis en modo de texto y las funciones del sistema para archivos pueden tener problemas con algunos caracteres de control. Por lo tanto, es más seguro representar datos binarios como una cadena literal con secuencias de escape explícitas para los caracteres que no son de texto.

Una constante numérica (o numeral) en Lua puede escribirse con una parte fraccional opcional y un exponente decimal opcional, marcado con una letra 'e' o 'E'. Lua también acepta constantes hexadecimales, que comienzan con 0x o 0X. Las constantes hexadecimales también pueden tener una parte fraccional opcional y un exponente binario opcional, marcado con una letra 'p' o 'P' y escrito en decimal. (Por ejemplo, 0x1.fp10 representa el número 1984, que es 0x1f dividido con 16 y multiplicado por 2^{10} —> `(0x1f/16)*2^10`).

Una constante numérica con un punto decimal o un exponente denota un número de tipo "float"(número de punto flotante); de lo contrario, si su valor cabe en un número entero o es una constante hexadecimal, denota un número entero; de lo contrario (es decir, si es un número entero decimal que desborda), denota un número de tipo "float". Los números hexadecimales sin punto decimal ni exponente siempre denotan un valor entero; si el valor desborda, se ajusta para que entre en un entero válido.

Ejemplos de constantes enteras válidas son:

```
3   345   0xff   0xBEBADA
```

Ejemplos de constantes flotantes válidas son:

```
3.0     3.1416    314.16e-2              0.31416E1   34e1
0x0.1E  0xA23p-4  0X1.921FB54442D18P+1
```

Un comentario comienza con dos guiones medios (--) en cualquier lugar fuera de una cadena. Si el texto inmediatamente después de -- no es un corchete largo de apertura, el comentario es un comentario corto, que se extiende hasta el final de la línea. De lo contrario, es un comentario largo, que se extiende hasta el corchete largo de cierre correspondiente.

3.2 Variables

Las variables son lugares donde se almacenan valores. En Lua, existen tres tipos de variables: variables globales, variables locales y campos de tablas.

Un solo nombre puede representar una variable global, una variable local (o un parámetro formal de una función, que es un tipo particular de variable local):

```
var ::= Name
```

El nombre denota identificadores (ver sección 3.1).

Cualquier nombre de variable se asume como global a menos que se declare explícitamente como local (ver sub-sección 3.3.7). Las variables locales tienen alcance léxico: las variables locales pueden ser accedidas libremente por las funciones definidas dentro de su alcance (ver sección 3.5).

Antes de la primera asignación a una variable, su valor es **nil**.

Los corchetes cuadrados se utilizan para indexar una tabla:

```
var ::= prefixexp '[' exp ']'
```

El significado de los accesos a los campos de una tabla puede ser cambiado mediante las metatablas (ver sección 2.4).

La sintaxis `var.Name` es simplemente azúcar sintáctico para `var["Nombre"]`:

```
var ::= prefixexp '.' Name
```

Un acceso a una variable global `x` es equivalente a `_ENV.x`. Debido a la forma en que los fragmentos de código son compilados, la variable `_ENV` en sí misma nunca es global (ver sección 2.2).

3.3 Sentencias

Lua admite un conjunto casi convencional de instrucciones, similar al de otros lenguajes convencionales. Este conjunto incluye bloques de código, asignaciones, estructuras de control, llamadas a funciones y declaraciones de variables.

3.3.1. Bloques

Un bloque es una lista de instrucciones que se ejecutan secuencialmente. En Lua, un bloque de código está compuesto por una serie de declaraciones o instrucciones que se ejecutan una tras otra en orden. Cada instrucción dentro

del bloque se ejecuta en secuencia, una vez que la instrucción anterior ha finalizado.

```
block ::= {stat}
```

En Lua, existen las "sentencias vacías"que permiten separar instrucciones con punto y coma, empezar un bloque con un punto y coma o escribir dos puntos y comas en secuencia.

```
stat ::= ';'
```

Por ejemplo, el siguiente código es perfectamente válido en Lua:

```
local x = 10;
```

Tanto las llamadas a funciones como las asignaciones pueden comenzar con un paréntesis abierto en Lua. Esta posibilidad lleva a una ambigüedad en la gramática de Lua. Considera el siguiente fragmento:

```
a = b + c
(print or io.write)('done')
```

La gramática podría interpretar este fragmento de dos formas:

```
a = b + c(print or io.write)('done')
a = b + c; (print or io.write)('done')
```

El analizador actual siempre interpreta construcciones de este tipo de la primera manera, considerando el paréntesis abierto como el inicio de los argumentos de una llamada. Para evitar esta ambigüedad, es una buena práctica preceder siempre con un punto y coma a las declaraciones que comienzan con un paréntesis:

```
;(print or io.write)('done')
```

Un bloque puede ser delimitado explícitamente para producir una sola declaración:

```
stat ::= do block end
```

Los bloques explícitos son útiles para controlar el alcance de las declaraciones de variables. También se utilizan a veces bloques explícitos para agregar una declaración de `return` en medio de otro bloque (ver sub-sección 3.3.4).

3.3.2. Fragmentos de Código(Chunks)

La unidad de compilación de Lua se llama '*chunk*'. Sintácticamente, un 'chunk' es simplemente un bloque:

```
chunk ::= block
```

En Lua, un 'chunk' se maneja como el cuerpo de una función anónima con un número variable de argumentos (ver sub-sección 3.4.11). Como tal, los 'chunks' pueden definir variables locales, recibir argumentos y devolver valores. Además, dicha función anónima se compila en el ámbito de una variable local externa llamada _ENV (ver sección 2.2). La función resultante siempre tiene _ENV como su única variable externa, incluso si no usa esa variable.

Correcto. En Lua, un 'chunk' puede ser almacenado en un archivo o en una cadena de texto dentro del programa principal (host program). Para ejecutar un 'chunk', Lua primero lo carga, precompilando el código del 'chunk' en instrucciones para una máquina virtual, y luego ejecuta el código compilado con un intérprete para la máquina virtual.

Los 'chunks' también pueden ser precompilados en forma binaria, lo que se conoce como 'binary chunks'. Para lograr esto, Lua proporciona un programa llamado luac que se utiliza para realizar la compilación previa del código fuente en forma binaria. Además, la función string.dump se puede utilizar para precompilar un 'chunk' en una cadena de bytes que representa su forma binaria.

Tanto los programas en forma de código fuente como los compilados en forma binaria son intercambiables en Lua. Esto significa que puedes cargar y ejecutar tanto 'chunks' en su forma de código fuente (texto) como en su forma precompilada (binaria) utilizando la función load. Lua automáticamente detectará el tipo de archivo y actuará en consecuencia, lo que proporciona flexibilidad en el manejo de los 'chunks' en diferentes formas.

3.3.3. Asignación

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec

vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.3.4. Estructuras de Control

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.3.5. Sentencia For

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu,

accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.3.6. Llamadas de Funciones como Sentencias

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.3.7. Declaraciones Locales

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.3.8. Variables con cierre automático

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

3.4 Expresiones

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.1. Operadores Aritméticos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.2. Operadores de Bits

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor

lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.3. Conversiones y Coerciones

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.4. Operadores Relacionales

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu

tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.5. Operadores Lógicos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.6. Concatenación

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.7. Operador de Longitud

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor

lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.8. Precedencia

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.9. Constructores de Tablas

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu

tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.10. Llamadas de Funciones

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.11. Definiciones de Funciones

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.4.12. Listas de expresiones, resultados múltiples y ajuste

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

3.5 Reglas de Visibilidad

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4

Interfaz de Programación de Aplicaciones (API)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.1 La Pila (Stack)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.1.1. Tamaño de la Pila

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.1.2. Índices Válidos y Aceptables

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.1.3. Punteros a Cadenas (Strings)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus

rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.2 Cierres en C (C Closures)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.3 Registro (Registry)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.4 Manejo de Errores en C

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.4.1. Códigos de Estado

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.5 Manejo de Pausas (Yields) en C

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec

vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.6 Funciones y Tipos de Datos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et

magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.7 Interfaz de Depuración (Debug)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

5

Biblioteca Auxiliar

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

5.1 Funciones y Tipos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6

Las Bibliotecas Estándar

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.1 Funciones Básicas

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.2 Manipulación de Coroutines

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.3 Módulos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.4 Manipulación de Cadenas

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec

vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.4.1. Patrones

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna.

Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.4.2. Cadenas de Formato para Empaquetar y Desempaquetar

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.5 Soporte UTF-8

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean

faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.6 Manipulación de Tablas

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.7 Funciones Matemáticas

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.8 Facilidades de Entrada y Salida

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.9 Facilidades del Sistema Operativo

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

6.10 La Biblioteca de Depuración

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec

vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

7

Lua Independiente (Standalone)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum.

Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

8

Incompatibilidades con la Versión Anterior

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

8.1 Incompatibilidades en el Lenguaje

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

8.2 Incompatibilidades en las Bibliotecas

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

8.3 Incompatibilidades en la API

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

9

La Sintaxis Completa de Lua

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum.

Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Lua y el logotipo de Lua son marcas comerciales de Lua.org.

Para obtener más información sobre Lua y acceder a la documentación original de Lua, visita: http://www.lua.org/