



LibreOffice WebAssembly (LOWA)

Introducción

Svante Schubert

`<svante.schubert.extern@allotropia.de>`

Congreso Latinoamericano, 2023-11-10

¿Quién habla?

- Svante Schubert - ex-StarDivision (empezó en 1999 en Hamburgo)
 - OASIS ODF TC: co-chair, co-editor
 - Co-mantenedor del ODF Toolkit
 - Filtro XSLT ODF2XHTML

allotropia



Un **alótropo** de un elemento químico es la existencia de múltiples formas en el mismo estado físico.

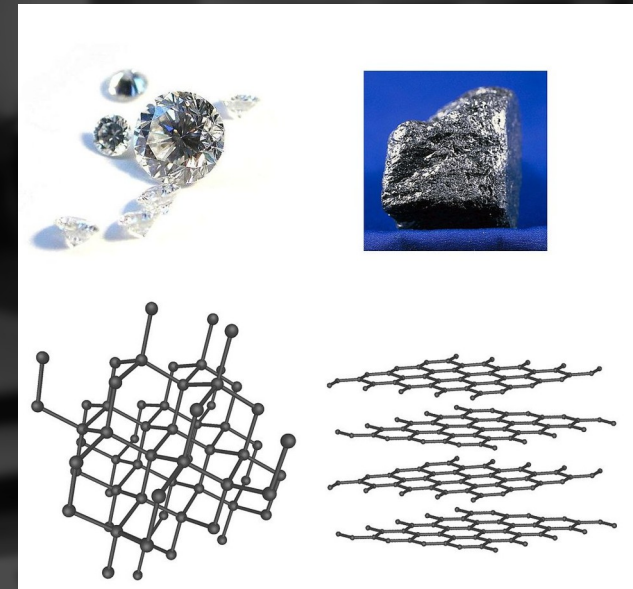
Diamante y grafito

son dos alótropos del **carbón**

(del griego antiguo

ἄλλος (**allos**) 'otro',

y τρόπος (**tropos**) 'modo, forma')



allotropia

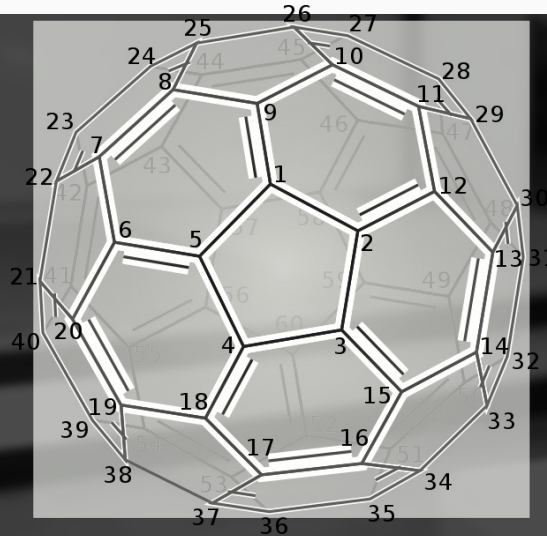


allotropia

“

And if you let FLOSS sit in just the right environment for long enough - you get a diamond (or the even more interesting buckyball, see picture).

”



<https://en.wikipedia.org/wiki/Buckminsterfullerene> (alias buckyball - fórmula

allotropia

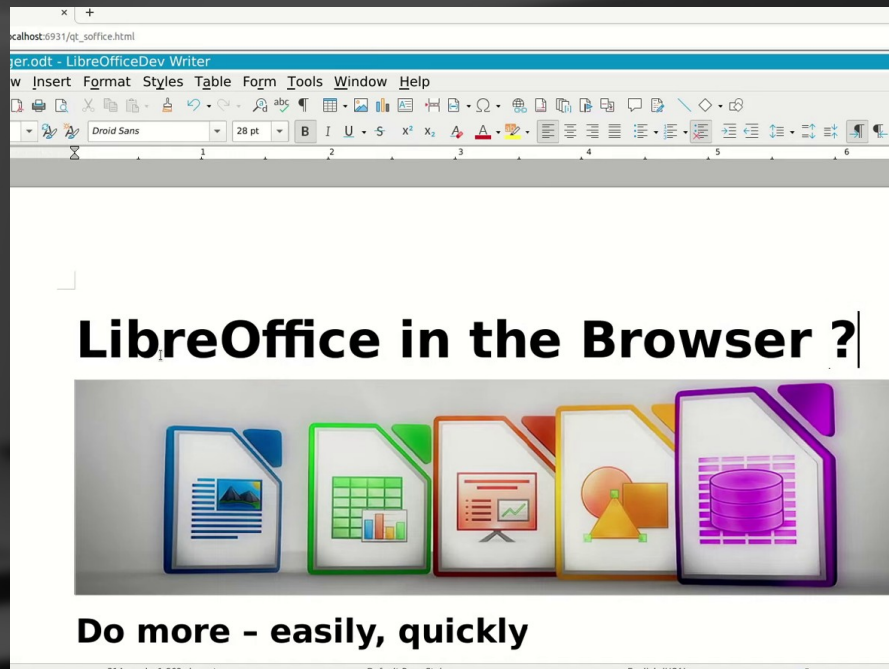


- Empresa alemana de Hamburgo por Thorsten Behrens (CEO de allotropia y Presidente del Consejo de TDF)
- Soporte profesional de LibreOffice para gobiernos y empresas
- Ser un gran partidario del ODF:
 - Michael Stahl (allotropía) ser coeditor en el OASIS ODF TC
 - Michael Stahl (allotropia) siendo co-mantenedor del ODF Toolkit (utilizado para el [validador ODF](#))

¿Cómo?



- LibreOffice WebAssembly (LOWA)
- Puerto nativo de LibreOffice, que se ejecuta del lado del cliente en el navegador
- Proyecto financiado por NLnet / Horizonte 2020 (CE), allotropia software GmbH (inventores de LOWA); utilizado por Collabora Productivity





¿Por qué WebAssembly?

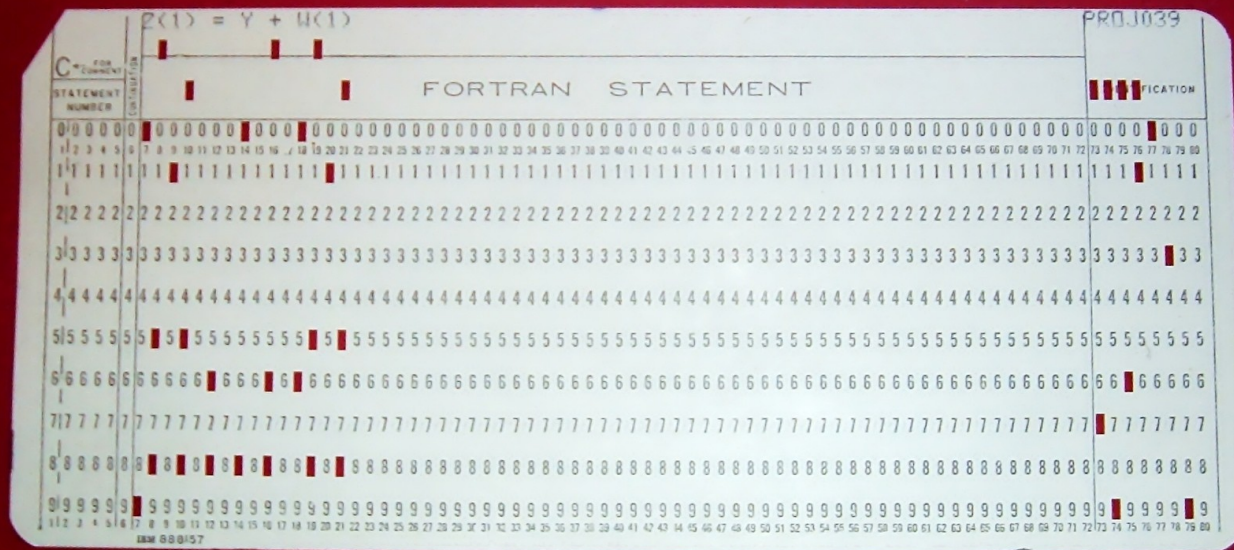
Programación

Tarjetas perforadas hasta ~1975



allotropia

- *Escribe una vez*
- 80 caracteres por tarjeta
- Programa = baraja de cartas



- ¿Cuántas tarjetas LO codifican?
~10 millones de líneas de código
(¡no tires ESA caja!)

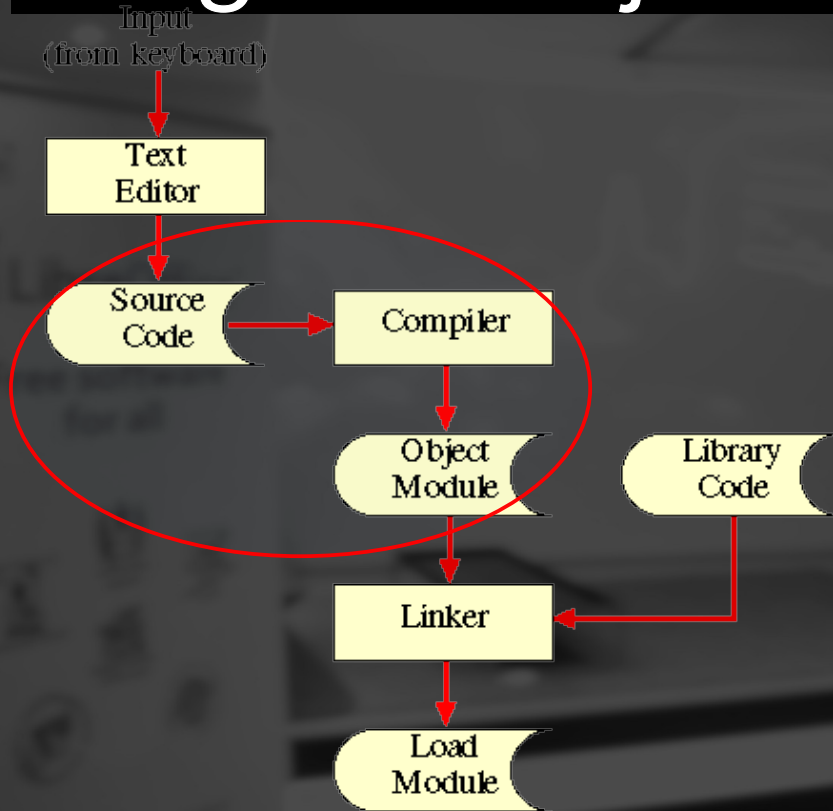
Del editor de texto a máquina



Programas informáticos en sus más diversas formas:

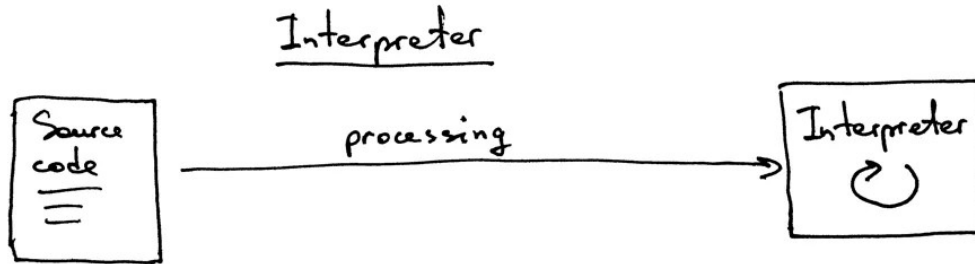
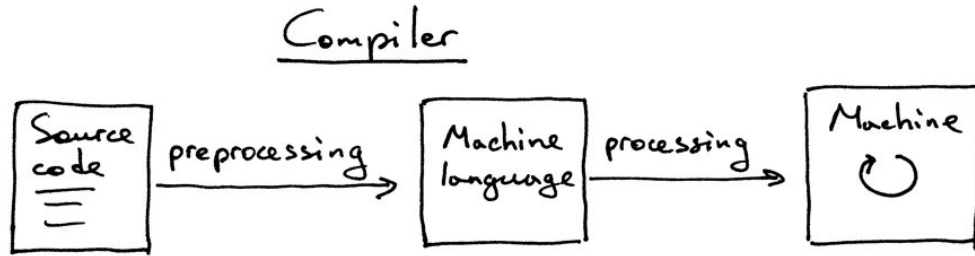
- 1) **Código fuente** del programador:
- 2) Archivo de texto **semántico de alto nivel** en un lenguaje de programación
- 3) **Código máquina** para CPU/GPU:
Archivo binario **semántico de bajo nivel** con órdenes a la máquina

Código fuente a Programa ejecutable



- 1) Semánticamente **de alto nivel**
Código fuente en archivos de texto de **Lenguaje de programación**
- 2) Compilador Ahead-Of-Time (AOT) **transforma** el **código fuente** en **código máquina** (módulo objeto) de un **Lenguaje Máquina**
- 3) Módulo Objeto es **Código máquina** para ejecutarse en **Arquitectura de ordenador** (por ejemplo, x64)
La compilación depende de **Sistema operativo** (por ejemplo, Linux)
debido a las diferentes

Compilador vs. intérprete



1) Compilador == Traductor en tiempo de compilación

PRO:

a) ¡Más rápido en RunTime!

2) Intérprete == Traductor en tiempo de ejecución

PRO:

a) El código funciona en múltiples plataformas.

b) El código aún se puede leer.

De <https://medium.com/@Sujithnath/angular-aot-vs-jit-comparision-ce1d96ede491>

Problema: Torre de Babel



- Miles de **lenguajes de programación**
 - Cada uno en diferentes versiones
- Cientos de **arquitecturas de ordenadores** cada una con un lenguaje de máquina propio debido a:

- Diferentes tamaños de registro de CPU: 32bit, 64bit,.
- Nuevas versiones de CPU con nuevos comandos

p. ej. SIMD Instrucción única Datos múltiples (utilizado [para analizar JavaScript con extrema rapidez](#))

https://en.wikipedia.org/wiki/Comparison_of_instruction_set_architectures (Arquitectura de ordenadores)

https://en.wikipedia.org/wiki/Comparison_of_programming_languages (Lenguajes de programación)

Problema: Torre de Babel



allotropia



1000 Programación
Lenguajes

*

100 Arquitectura de
ordenadores

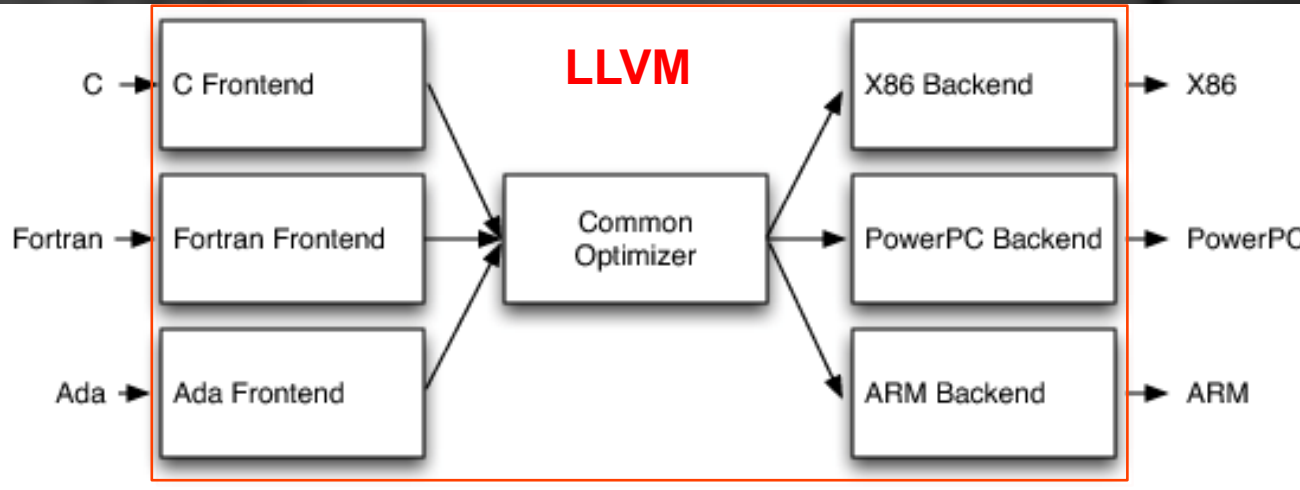
=>

100.000 compiladores (?)

https://en.wikipedia.org/wiki/Comparison_of_instruction_set_architectures (Arquitectura de ordenadores)

https://en.wikipedia.org/wiki/Comparison_of_programming_languages (Lenguajes de programación)

Solución: LLVM



1000 Lenguajes de
Programación

+

100 Arquitectura de
ordenadores

=>

1000 frontales

y

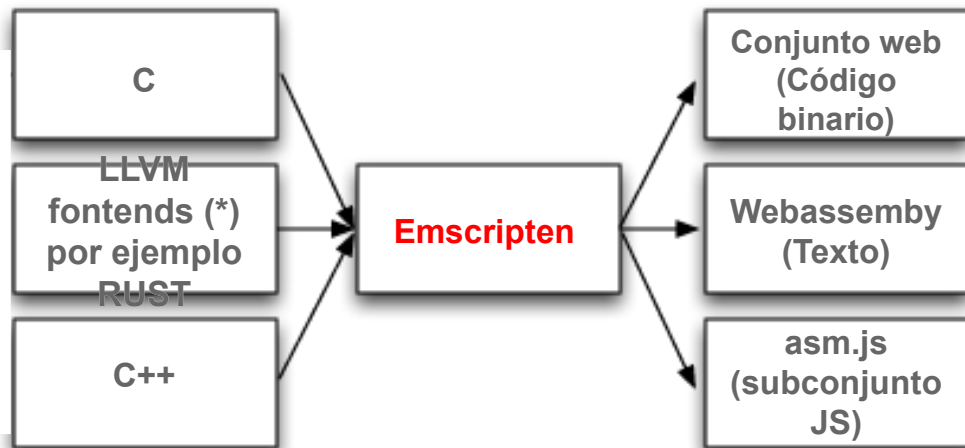
100 backends

<https://aosabook.org/en/v1/llvm.html> - Diagrama de arquitectura de Chris Lattner (coinventor de LLVM)

<https://en.wikipedia.org/wiki/LLVM> - Máquina virtual de bajo nivel

Solución: Emscripten

(compilador basado en LLVM/Clang)



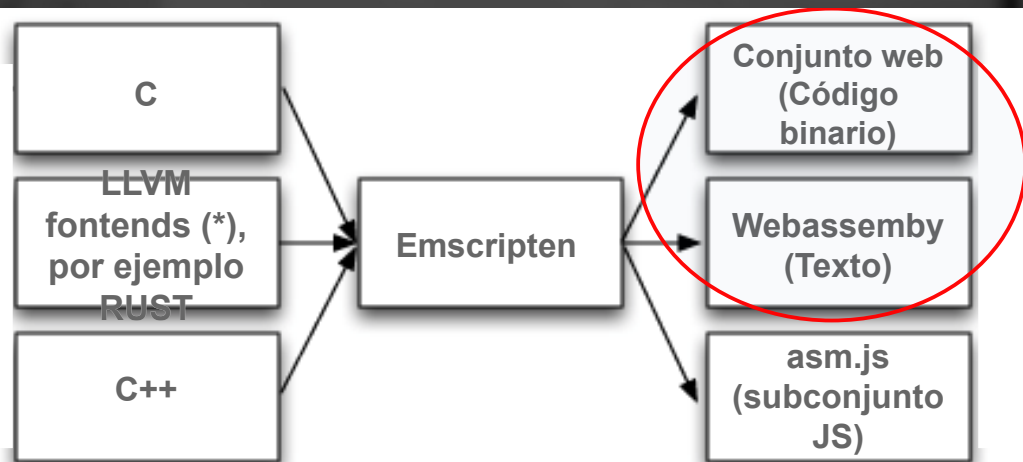
Crea Webassembly/WASM

- 1) En formato binario
- 2) En formato texto
- 3) O asm.js (subconjunto JS) fue en su día uno de los orígenes de WASM

(*) Todos los lenguajes de programación soportados por LLVM pueden ser compilados a WebAssembly
(no directamente por WebAssembly)

<https://en.wikipedia.org/wiki/Emscripten> - Emscripten

Solución: Webassembly



- Diseñado/Dirigido por W3C
- De asm.js (por Mozilla)
- De NaCl (por Google)
- (Cliente nativo)
- Con el apoyo de Microsoft y Apple
- **Incluido en los navegadores**

(*) Todos los lenguajes de programación soportados por LLVM pueden ser compilados a **WebAssembly**
(no directamente por WebAssembly)

<https://en.wikipedia.org/wiki/Emscripten> - Emscripten

Compilador:

AOT, JIT e Intérprete



	Editing Time		Compile Time		Runtime			
AOT compiler		Source		Source --> Machine Code		executing	"Machine Code"	
JIT compiler		Source		Source --> Intermediate Code		interpreting	"Intermediate Code"	
Interpreter		Source				interpreting	"Source Code"	

Ahead of Time (AOT) - compilar una vez por adelantado

Just in Time (JIT) - compila cada vez que se ejecuta

NOTA:

¡Mixform existe donde "interpretar" es también "compilar" HotSpots!

Hoja de ruta de WebAssembly

Intérprete integrado



	Your browser	Chrome	Firefox	Safari	Wasmtime	Wasmer	Node.js	Deno	wasm2c
Standardized features									
JS BigInt to Wasm i64 integration	✓	85	78	14.1 ^[a]	N/A	N/A	15.0	1.1.2	N/A
Bulk memory operations	✓	75	79	15	0.20	1.0	12.5	0.4	1.0.30
Extended constant expressions	✓	114	112	✗	✗	✗	🚩 ^[k]	🚩 ^[r]	🚩 ^[y]
Garbage collection	✓	119	120	✗	✗	✗	🚩 ^[l]	✗	✗
Multiple memories	✗	120	🚩 ^[d]	✗	🚩 ^[g]	✗	✗	✗	🚩 ^[aa]
Multi-value	✓	85	78	✓	0.17	1.0	15.0	1.3.2	1.0.24
Mutable globals	✓	74	61	12	✓	0.7	12.0	0.1	1.0.1
Reference types	✓	96	79	15	0.20	2.0	17.2	1.16	1.0.31
Relaxed SIMD	✓	114	🚩 ^[c]	✗	🚩 ^[h]	✗	🚩 ^[o]	🚩 ^[u]	✗
Non-trapping float-to-int conversions	✓	75	64	15	✓	✓	12.5	0.4	1.0.24
Sign-extension operations	✓	74	62	14.1 ^[a]	✓	✓	12.0	0.1	1.0.24
Fixed-width SIMD	✓	91	89	16.4	0.33	2.0	16.4	1.9	1.0.33
Tail calls	✓	112	🚩 ^[c]	✗	🚩 ^[i]	✗	🚩 ^[p]	🚩 ^[v]	🚩 ^[ab]
Threads and atomics	✓	74	79	14.1 ^[a]	🚩 ^[i]	✗	16.4	1.9	✗
In-progress proposals									
Exception handling	✓	95	100	15.2	✗	✗	17.0	1.16	🚩 ^[x]
JS Promise Integration	✗	🚩 ^[a]	✗	✗	N/A	N/A	🚩 ^[m]	🚩 ^[s]	N/A
Memory64	✗	🚩 ^[b]	🚩 ^[c]	✗	🚩 ^[f]	✗	🚩 ^[n]	🚩 ^[t]	🚩 ^[z]
Type reflection	✗	🚩 ^[b]	🚩 ^[c]	✗	✗	2.0	🚩 ^[q]	🚩 ^[w]	✗

- Número verde = número de versión desde que la función está activada
- Bandera = requiere una bandera de compilación
- (disponible en la versión Nightly no beta ni release)
- Cruz Roja = no apoyada
- Su navegador = mi navegador:
- Chrome Ver. 119.0.6045.124

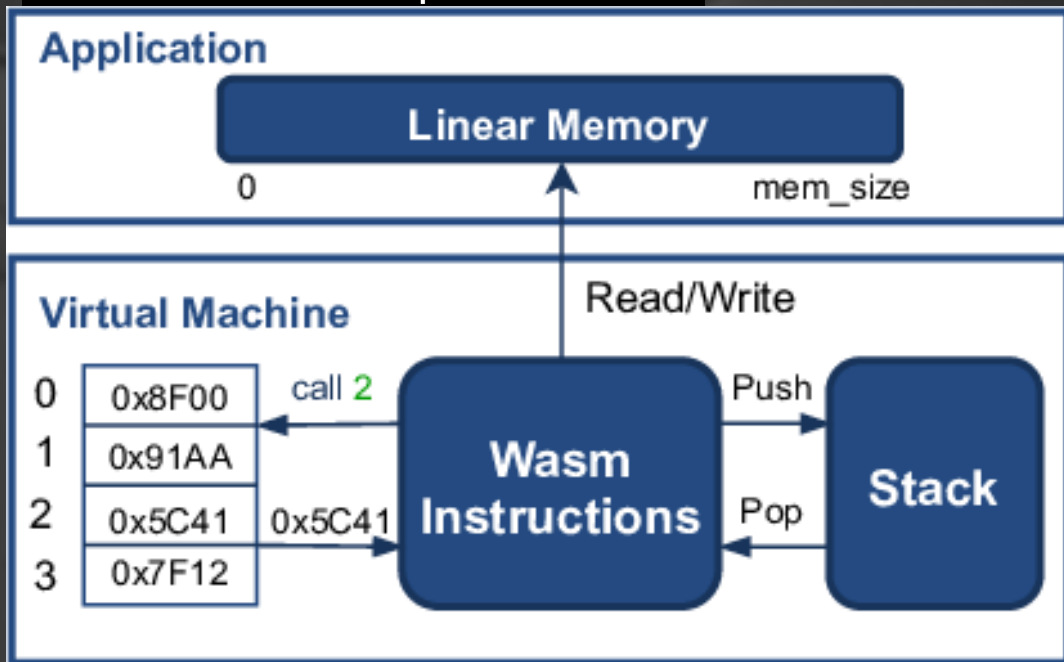
<https://webassembly.org/roadmap/> - Hoja de ruta de WebAssembly

WebAssembly

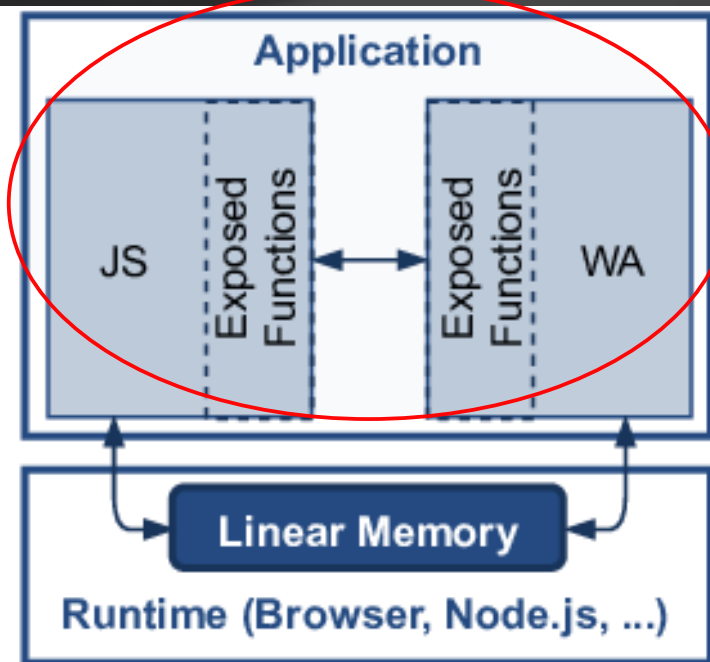
Diseño e interoperabilidad



allotropia



a) WebAssembly memory model



b) Design of typical application using WebAssembly

El código de ensamblaje web funciona con el código JavaScript existente

De [Wasmati: Un eficiente escáner estático de vulnerabilidades para WebAssembly](#)

LOWA

<https://lab.allotropia.de/wasm>



allotropia

soffice x

https://lab.allotropia.de/wasm/


example_larger.odt - LibreOfficeDev Writer 7.4 [d9b2f82b09d89ea918e6d1555d860426fd153c71]

File Edit View Insert Format Styles Table Form Tools Window Help

Title Droid Sans 28 pt B I U S x² x₂ A A

1 2 3 4 5 6 7

What is LibreOffice?



Do more - easily, quickly

LibreOffice is a powerful office suite; its clean interface and powerful tools let you unleash your creativity and grow your productivity. LibreOffice embeds several applications that make it the most powerful Free & Open Source Office suite on the market: Writer, the word processor, Calc, the spreadsheet application, Impress, the presentation engine, Draw, our drawing and flowcharting application, Base, our database and database frontend, and Math for editing mathematics.

Finally, documents that look good

Your documents will look professional and clean, regardless of their purpose: a letter, a master thesis, a brochure, financial reports, marketing presentations, technical drawings and diagrams.

Use documents of all kinds

LibreOffice is compatible with many document formats such as Microsoft® Word, Excel, PowerPoint and Publisher. But LibreOffice goes further by enabling you to use a modern open standard, the OpenDocument Format (ODF).

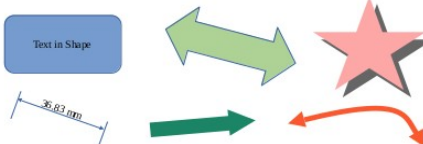
Free as in Freedom, now and forever

LibreOffice is Free and Open Source Software. Its development is open to new talent and new ideas. Our software is tested and used daily by a large and devoted user community; you, too, can [get involved](#) and influence its future development.

Table

A	B	C	D	E	F
I					
II					
III					
IV					

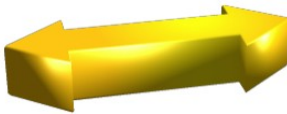
DrawShapes



Textframe

He heard quiet steps behind him. That didn't bode well. Who could be following him this late at night and in this deadbeat part of town? And at this particular moment, just after he pulled off the big time and was making off with the greenbacks. Was there another crook who'd had the same idea, and was now watching him and waiting for a chance to grab the fruit of his labor? Or did the steps behind him mean that one of many law officers in town was on to him and just waiting to pounce and snap those cuffs on his wrists?

3DShape



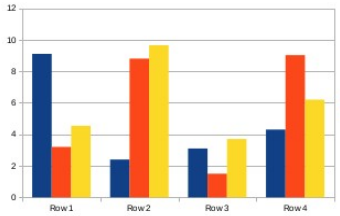
Textbox

The quick brown Fox Jumps over the lazy Dog

Fontwork

NOTE

Chart



Row	Column 1	Column 2	Column 3
Row1	9	3	4
Row2	2	9	10
Row3	3	1	4
Row4	4	9	6

Properties

Style

Title

Character

Droid Sans 28 pt B I U S A A

Paragraph

Spacing: Indent: 0.17" 0.00" 0.08" 0.00" 0.00"

LOWA

tiempo de carga: @hotel-Wifi)



allotropia

Name	Status	Type	Initiator	Size	Time
wasm	301	document / Redir...	Other	544 B	950 ms
wasm/	200	document	<u>wasm</u>	(disk cache)	3 ms
qtlogo.svg	200	svg+xml	<u>wasm/:26</u>	9.9 kB	171 ms
qtloader.js	200	script	<u>wasm/:69</u>	6.3 kB	494 ms
soffice.js	200	fetch	<u>qtloader.js:249</u>	149 kB	6.58 s
soffice.wasm	200	wasm	<u>qtloader.js:249</u>	53.3 MB	29.63 s
soffice.worker.js	200	javascript	Other	4.1 kB	188 ms
soffice.worker.js	200	javascript	Other	0 B	189 ms
soffice.worker.js	200	javascript	Other	0 B	189 ms
soffice.worker.js	200	javascript	Other	0 B	190 ms
blob:https://lab.allotropia.de/cc442c1f-209a-464c-8a71-f04f76a626c7	200	text/javascript	<u>soffice.worker.js:105</u>	0 B	15 ms
blob:https://lab.allotropia.de/27e8d7a5-49f8-448c-804e-b0b8b034c8e9	200	text/javascript	<u>soffice.worker.js:105</u>	0 B	28 ms
blob:https://lab.allotropia.de/a440406f-8f21-4a14-8918-8ae19b6bfe70	200	text/javascript	<u>soffice.worker.js:105</u>	0 B	29 ms
blob:https://lab.allotropia.de/b65d33d5-5169-4960-8300-bdc163b55def	200	text/javascript	<u>soffice.worker.js:105</u>	0 B	21 ms
soffice.data.js.metadata	200	xhr	<u>VM297:319</u>	21.1 kB	546 ms
soffice.data	200	xhr	<u>VM297:111</u>	36.5 MB	8.61 s

16 requests | 90.0 MB transferred | 312 MB resources | Finish: 43.56 s | DOMContentLoaded: 901 ms | Load: 2.72 s

Cargando - https://lab.allotropia.de/wasm

con Chrome Ver. 119.0.6045.124 - 43,5 seg (ver arriba) con FireFox 119 - 7min 16 seg

Ventajas de LOWA



allotropia

- Sin duplicación de la aplicación (1 código base)
- La implantación de Office es tan fácil como cargar JavaScript (más tarde, Browser Cache)
- LOWA funciona en todas las plataformas WASM (navegadores)
- Reutilización de bibliotecas web (por ejemplo, cifrado)
- La necesidad de modularización de la BAJA aumentará la escalabilidad (por ejemplo, para convertidores autónomos)

Evolución de la Colaboración en la oficina



allotropia

1. Intercambio de papel

- **Un usuario** puede ver/editar el documento
- Sin ordenador (sin digitalización)

2. Intercambio de documentos digitales

- **Un usuario** puede ver/editar el documento
- Ninguna ventaja sobre los archivos de texto

Evolución de la Colaboración en la oficina



allotropia

3. Intercambio de documentos de oficina (seguimiento de cambios)

- **Un usuario** puede ver/editar el documento
- Los cambios se pueden ver y aceptar/rechazar fácilmente
- Pero no se realiza un seguimiento de la aceptación/rechazo de cambios.
- En ODF XML y OOXML sólo se guarda el "estado anterior" (no se pueden rastrear cambios superpuestos)
- Los cambios no están definidos para el seguimiento de cambios (por ejemplo, insertar columna)

Evolución de la Colaboración en la oficina



4. Compartir el mismo programa y el mismo ordenador

- **Varios usuarios** pueden ver/editar el documento al mismo tiempo **en línea**

5. Compartir copias de un mismo documento

- **Varios usuarios** pueden ver/editar el documento al mismo tiempo **sin conexión.**
- ODF Cambios intercambiados y fusionados (como GIT)

Escalera a Oficina Colaboración



- Necesitamos una **API OASIS ODF**, similar a la API DOM del W3C pero a nivel semántico - `insertColumn(..)`
 1. Qué paquetes ODF XML a "Características ODF", por ejemplo, **Tabla**
 2. Qué cambios espera el usuario, por ejemplo `insertColumn(..)`
- LO requiere refactorización (como para WASM y ODF API)
- Norma OASIS ODF
 - Sólo XML no conoce características como una tabla
 - Requiere digitalización: generar software a partir de especificaciones
 - Reducir la carga para los nuevos ejecutores

Desafíos LOWA

Primeros retos de WASM



- La madurez de la plataforma (emscripten/WASM/navegadores) no está clara.
- Tamaño del problema ...
- Multitarea y peculiaridades de la gestión de eventos de la interfaz gráfica de usuario
- El tamaño máximo del montón es de 4 GB, debido a los punteros de 32 bits de WASM32.
- Entornos de desarrollo y depuración aún muy "frescos".

Desafíos LOWA



allotropia

- Tamaño del binario WASM resultante

Actualmente: empaquetado = 35M, optimizado = 150M, depuración = 200M + ~1G de información DWARF separada.

- Tamaño de la imagen del sistema de archivos

~ 100M con todas las fuentes LO, puede almacenarse localmente y dividirse si es necesario.

→ menos descargas en las actualizaciones

→ puede utilizar webfonts

- Integrabilidad y programabilidad

- quieres usar esto desde tu framework JS

- API masivamente potente ("UNO API"), cómo vincular esa

Ideas básicas

Construcción e interfaz gráfica



allotropia

- LibreOffice es un proyecto de autotools y GNU make
 - cíñase a eso, evite otros sistemas de construcción paralelos
 - y ya es bastante portable, la compilación cruzada está soportada fuera de la caja
- LibreOffice tiene su propia abstracción GUI
 - con plugins para Gtk, Qt/KF5, Win32 y OSX
 - con Qt5 soportando WASM de forma nativa, fuimos con eso

Ideas básicas

Construcción e interfaz gráfica



allotropia

- LibreOffice es básicamente C++ (en general c++17)
 - usamos emscripten como compilador de plataforma (actualmente en la versión 2.0.31)
- No queremos utilizar ninguna función experimental de WASM
 - sin roscado (¡pero ahora con todas las funciones!)
 - sin enlace dinámico (por desgracia, será necesario modificar el sistema de compilación)
 - sin excepciones WASM nativas
- Queríamos centrarnos en Writer inicialmente (y ahorrar tamaño al no construir/enviar el resto)

Qué podemos esperar

(y nuestra visión de LOWA)



- **No** sustituye a LibreOffice para escritorio/móvil ni a Collabora Online.
- En lugar de atender necesidades insatisfechas
- ¿Quieres jugar tú mismo?

<https://lab.allotropia.de/wasm>

NOTA: en este momento no es la última versión desplegada - ¡crea la tuya propia! :-)

Situación actual de la LOWA



- <https://wiki.documentfoundation.org/Development/WASM>
 - se reduce a: usar master, leer static/README.wasm.md
- Rama con funciones adicionales
[:https://git.libreoffice.org/core/+refs/heads/feature/wasm](https://git.libreoffice.org/core/+refs/heads/feature/wasm)
- Maestro ahora puede construir y ejecutar:
 - Una oficina WASM.(html|wasm)*
 - Calc y/o Writer, casi siempre trabajando*.
 - Un conversor de PDF sin cabeza
 - Un plugin WASM offline para Collabora Online



allotropia



Collabora
Online

Preguntas y respuestas

