velneo

Fernando Félix Gutiérrez Blanco
Velneo Development Team Manager
fgutierrez@velneo.com
X/Twitter @fgutierrez_
https://velneo.com


WASM

Giving Low Code to the Web with WebAssembly: Velneo's Success Story

Agenda

1. Velneo PaaS
2. The Challenge
3. Why WebAssembly
4. Integrating WebAssembly with Qt
5. Technical Insights
6. Benefits
7. Challenges
8. Future

Top 10 of the best companies to work in Spain

Best company in the ICT sector to work in Spain

International presence

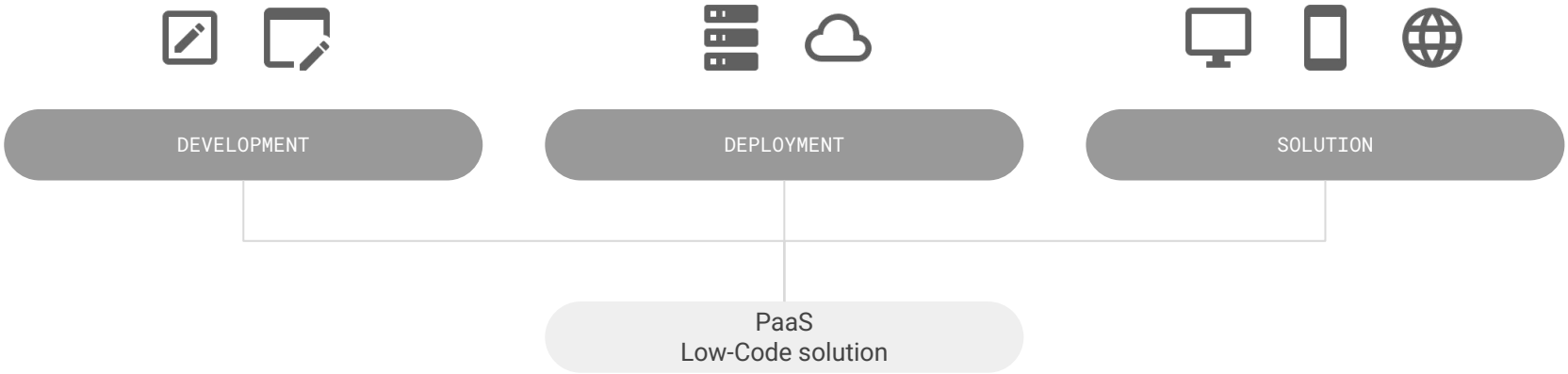
Velneo’s customers are present in more than 30 countries

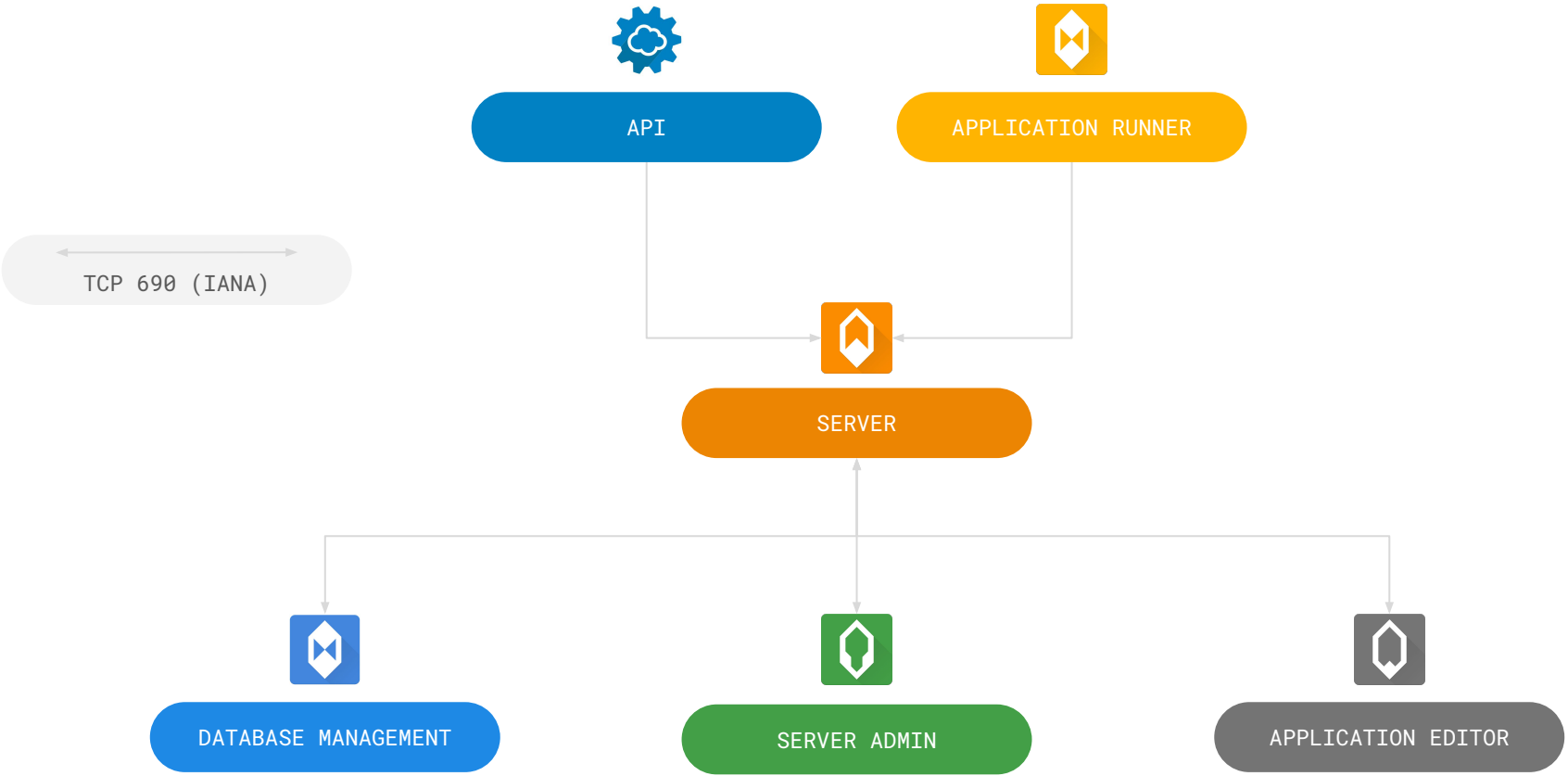

/* - - - - - - - - - - - - - - - - - - -
Create the application that your company needs
Build any business management application with Low-Code, you have the power to build your application alone or together with us, all in one place.
- - - - - - - - - - - - - - - - - - - */

Low-Code software category leaders. Gartner Report 2023↗

What is Velneo for?

- Enterprise Resource Planning (ERP)
- Customer Relationship Management (CRM)
- Inventory and Warehouse Management
- Manufacturing and Production Systems
- Accounting and Financial Management
- Human Resources (HR) and Payroll Solutions
- Project and Resource Management
- Supply Chain and Logistics Solutions


DEVELOPMENT
DEPLOYMENT
SOLUTION
PaaS
Low-Code solution


API
APPLICATION RUNNER
TCP 690 (IANA)
SERVER
DATABASE MANAGEMENT
SERVER ADMIN
APPLICATION EDITOR

Project explorer (⌘5)

Solution vERP_2 | Loaded projects

100% | All

vERP_2_app 35.2

vERP_2_dat 35.2

Project explorer (⌘5) | Project (⌘1)

Properties (⌘2)

A3 (MOV_G Table) — Trigger

Description	Value
Properties	
Identifier	A3
Name	
Styles	
Comments	
Type	New record: after a new record entry

Start

Create

New solution...

SQL Importer

Learn

Features of version

Documentation

Courses and webinars

Forum

Last solutions | My servers

Open solution...

Search solutions

Solution	Server		
vERP_2	vERP (Desarrollo)	vatps://c3.velneo.com:8940	Remove
vMarketing	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove
Bolsa de empleo	vERP (Desarrollo)	vatps://c3.velneo.com:8940	Remove
Tablas grandes Delagro	Localhost	vatps://127.0.0.1:6900	Remove
djson	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove
vEstandar	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove
vCMI_TEST	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove
vContactos Movil	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove
vDesignSystem	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove
vRegistroLicencias	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove
QMLTextRendering	VisualMS (Desarrollo)	vatps://c5.velneo.com:23030	Remove

Inspectors (⌘4)

Identifier	Name
vERP_2_dat 35.2	
Attached files	
Complex indexes	
Constants	
Consume web services	
Functions	
Image	
Processes	
Queries	
Schemes	
Static tables	
Tables	
Variables	

Subobjects (⌘3)

Identifier	Name
A1	
A3	
B1	
B3	
M1	
M3	

MULTIPLATAFORM

vGest

apps1.velneo.net

Cuadro de mando

Gestión

Maestros

Desconectar

Presupuesto de venta: 483

Serie: VTA

Fecha: 3/7/24

Cliente: Software Reunido, S.A.

Nº: 483

Estado: Pendiente

Tipo de retención:

Cantidad	Foto	Artículo	Precio	% dto.	Importe
8,00		Apple Watch SE	279,00	0,00	2.232,00
3,00		Huawei Matebook X Pro 2023	1.599,90	0,00	4.799,70
4,00		Dell XPS 17	2.849,01	0,00	11.396,04

Base imponible	Impuestos	Base retención	% retención	Retenciones	Total
18.427,74	3.869,83	18.427,74	0,00	0,00	22.297,57

Eliminar | Duplicar | Imprimir | Cancelar | Aceptar

The Challenge

Desktop

- Our platform is built on over 20 years of C++ development, which provided us with performance and stability.
- Separate builds for each operating system
    - Windows, macOS, Linux, Android, iOS
- Time-consuming and resource-intensive due to maintaining multiple builds.
    - Qt framework helps to set them low coding once but…
- Inconsistencies in user experience across platforms.

Web

- Build once, deploy everywhere
- Bringing our desktop-centric platform to the web without losing performance and stability
- Ensuring seamless integration and user experience across devices

Why WebAssembly

Previously

- NPAPI, Native Client, Pepper, PPAPI, etc. (Discontinued)
- Emscripten for JavaScript (Low performance)

Main features

- Unified Codebase: Compile once, run anywhere via the web, eliminating the need for separate builds.
- High Performance: Near-native execution speed, ensuring efficient performance across all devices.
- Broader Reach: Access applications on any device with a modern browser, enhancing accessibility.

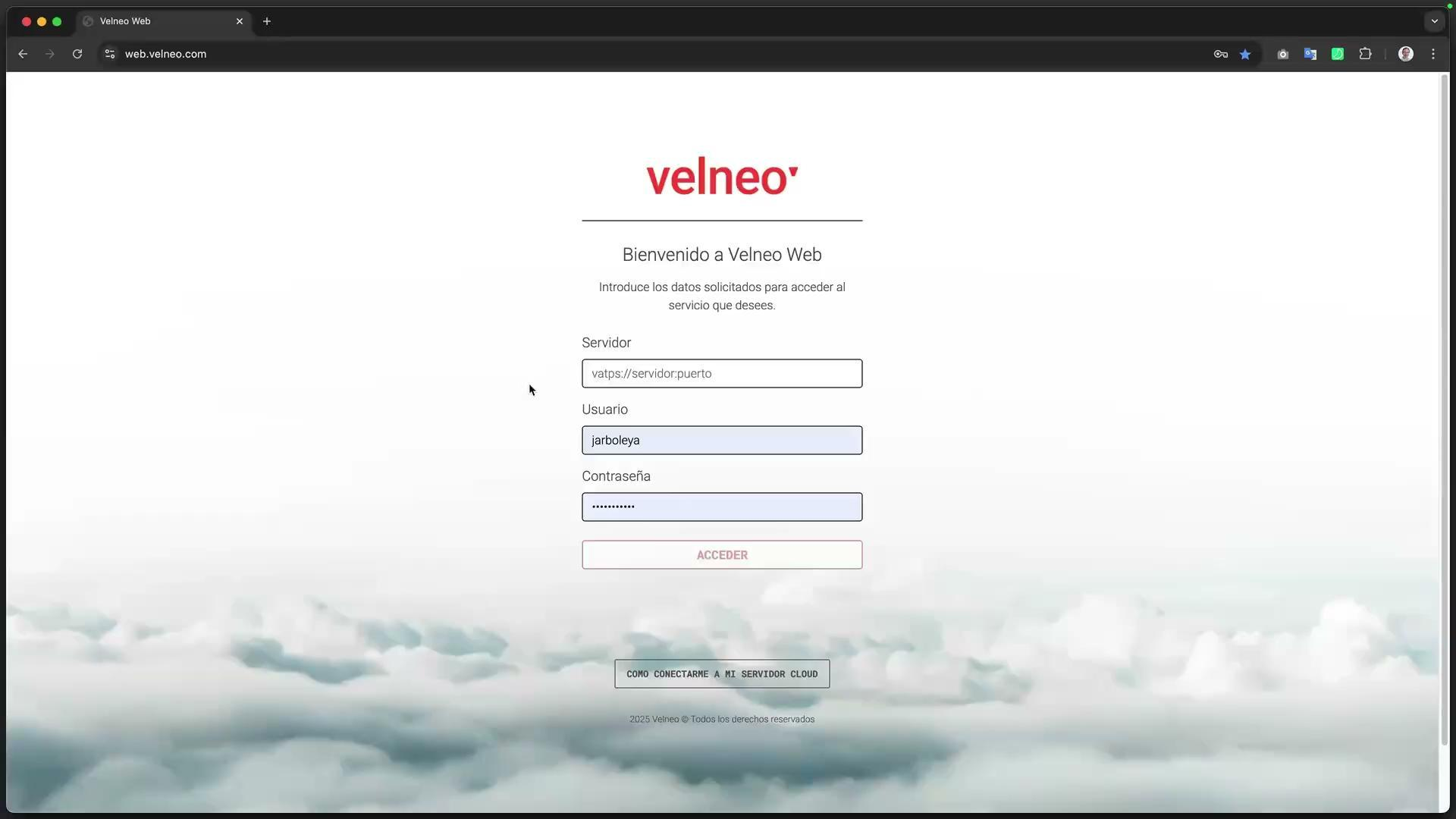
Integrating WebAssembly

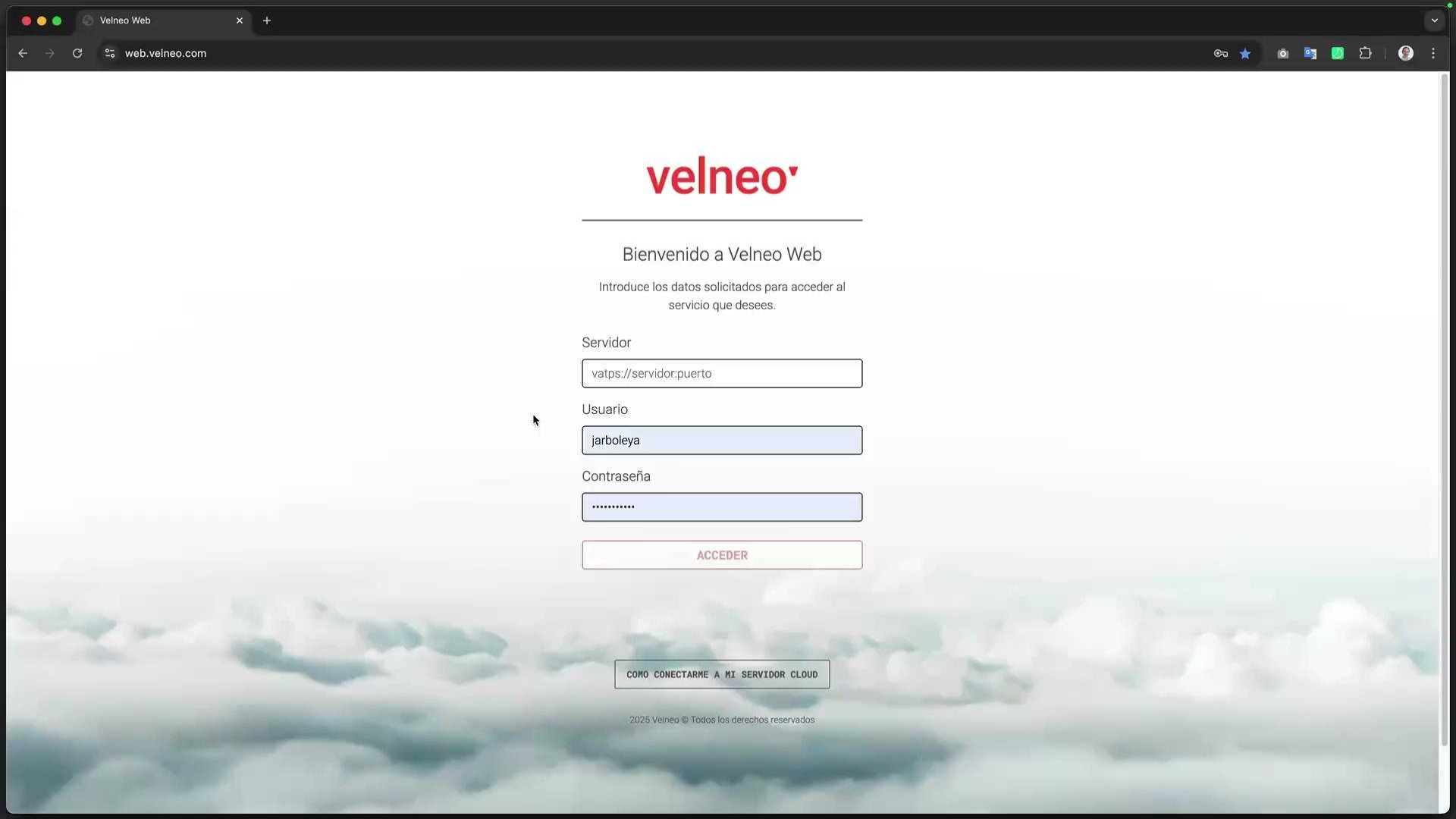
Key Steps in the Integration

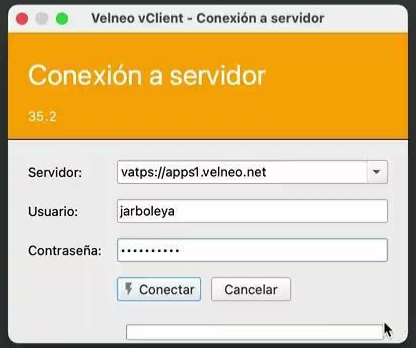
- Emscripten Compilation from C++
- Qt framework for WebAssembly helps with pre-built binaries and binary compatibility.
- Compilation flags and Emscripten optimizations to strike a balance between download size and runtime speed, and for better performance during execution.

Benefits

- Cross-System Compatibility
  - Velneo’s business logic can operate across different platforms including Web without changes in the applications
  - Velneo web components can also interact like the desktop components


Velneo Web
web.velneo.com
velneo
Bienvenido a Velneo Web
Introduce los datos solicitados para acceder al servicio que desees.
Servidor
vatps://servidor:puerto
Usuario
jarboleya
Contraseña
ACCEDER
COMO CONECTARME A MI SERVIDOR CLOUD
2025 Velneo © Todos los derechos reservados


Velneo Web
web.velneo.com
velneo
Bienvenido a Velneo Web
Introduce los datos solicitados para acceder al servicio que desees.
Servidor
vatps://servidor:puerto
Usuario
jarboleya
Contraseña
ACCEDER
COMO CONECTARME A MI SERVIDOR CLOUD
2025 Velneo © Todos los derechos reservados


Velneo vClient - Conexión a servidor
Conexión a servidor
35:2
Servidor:
vatps://apps1.velneo.net
Usuario:
jarboleya
Contraseña:
Conectar
Cancelar

Technical Insights: Core Architecture Overview

WebAssembly & Qt Interplay

- C++ compiles via Emscripten into a `.wasm` module
- Qt framework helps and create html and js files to load the module

Browser Environment

- Loaded with javascript (`qtloader.js`), the .wasm module runs in the browser's execution environment.
- Qt framework handle UI rendering in the canvas, event management, user interaction, and cross-platform APIs.
- Communication with the Velneo server uses TCP protocol 690 (IANA) under standard WebSockets thanks to Emscripten.

Technical Insights: Deploy

AppRunner vs Cloudfront

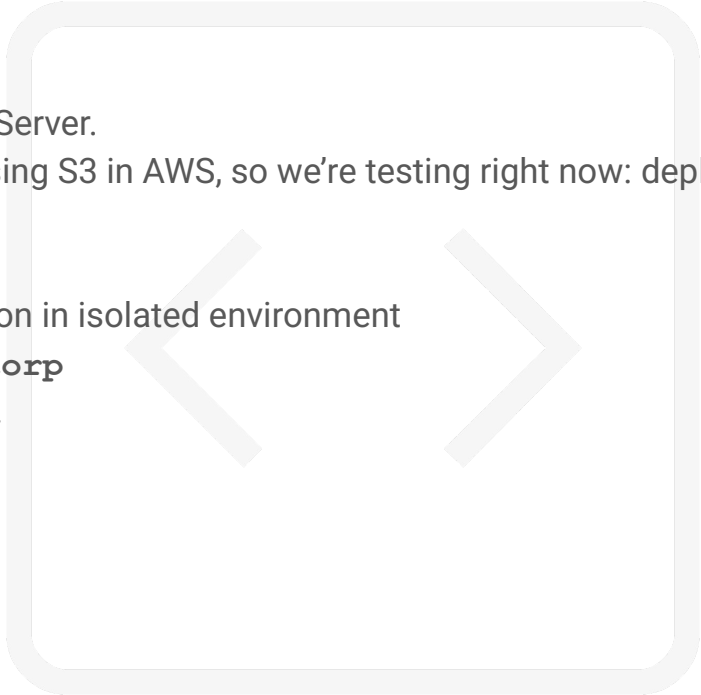
- We used AppRunner to deploy thanks to the Apache Server.
- Lately, required headers can be used in Cloudfront using S3 in AWS, so we're testing right now: deploy is just drop the files in a bucket.

Multi-thread requires SharedArrayBuffers flag and execution in isolated environment

- `Cross-Origin-Embedder-Policy: require-corp`
- `Cross-Origin-Opener-Policy: same-origin`

Third party Social Login

- Google, Microsoft Entry, etc.



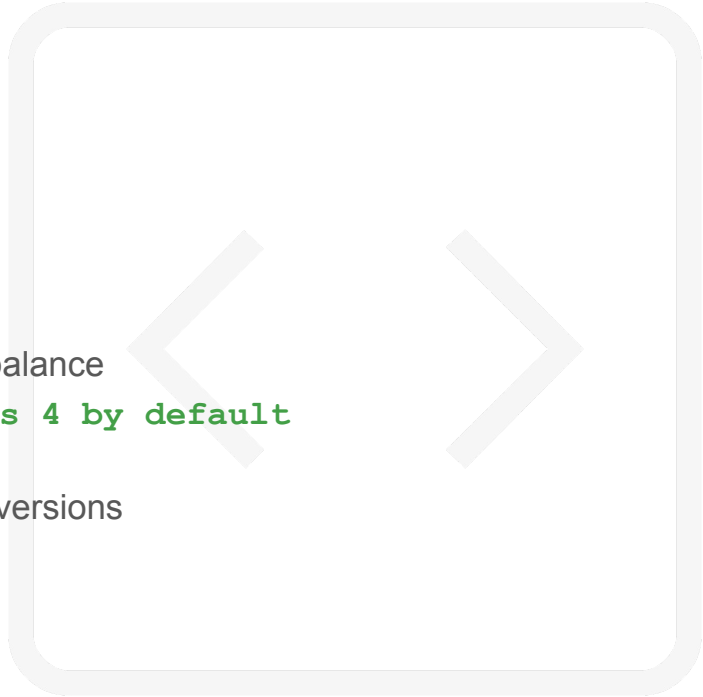
Technical Insights: Optimizing Performance

Caching & Compression

- Leveraging browser caching
- Brotli compression

Threading & Concurrency

- SharedArrayBuffer flag required
- Predefined concurrency: Performance and memory balance
  - `QT_WASM_PTHREAD_POOL_SIZE=9 // Qt sets 4 by default`
  - >9 Hangs and other bad behaviour
  - To be retested after new Emscripten and browser versions



Technical Insights: Memory Management Considerations

WASM Memory Allocation 32 bits

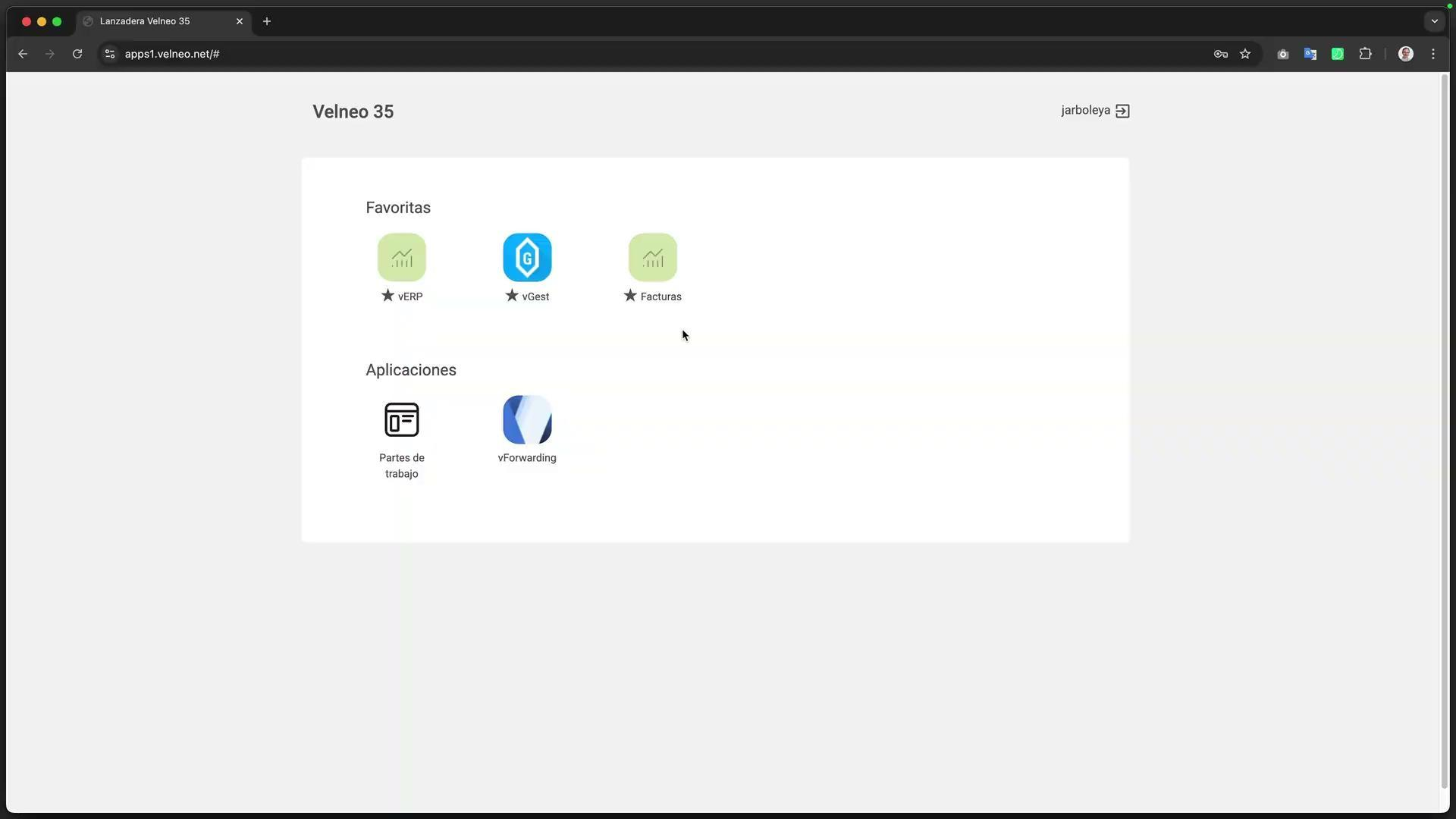
- We tuned memory settings to balance performance and memory footprint.
- We started our product with 32 bits so far ago so we knew the limitations and were ready for this.
- Qt framework helped interoperability with 64 bits.

Dynamic memory allocation

- In JavaScript is expensive due to garbage collection, fixed allocation is generally recommended.
- Emscripten optimizations and careful memory management have minimized its impact in our case.
    - `INITIAL_MEMORY=1522365440` `// 1.4GB`
    - `ALLOW_MEMORY_GROWTH=true` `// Default by Qt)`
    - `MAXIMUM_MEMORY` `// 2GB by default`

Resource Cleanup

- Proper cleanup of resources helps maintain consistent performance during prolonged sessions (especially when dealing with large datasets and cache)


Lanzadera Velneo 35
apps1.velneo.net/#
Velneo 35
jarboleya
Favoritas
★ vERP
★ vGest
★ Facturas
Aplicaciones
Partes de trabajo
vForwarding

Technical Insights: Asynchrony

Transitioning to Asynchrony: Reengineering Velneo's C++ Codebase

- Velneo Interface completely connected to database

Asyncify

- `-sASYNCIFY -Os`
- However, we eventually dropped that approach because it slowed down the performance and made debugging significantly more challenging.

Rewrite nearly our entire codebase to operate asynchronously exploring event-driven alternatives

- -e.g. Avoid Qt exec calls in dialogs, avoid tcp synchronous communications
- 21.545.720 millions of code lines

Technical Insights: Adaptations to the browser environment

Disk Access in the Browser

- Instead of direct disk access, we use browser APIs within browser security limits
    - LocalStorage (10MB limit): Caché (compressed) & downloads & applications logic
    - File System Access API helped by Qt framework API: User downloads/uploads from desktop

PDF Generation for Printing:

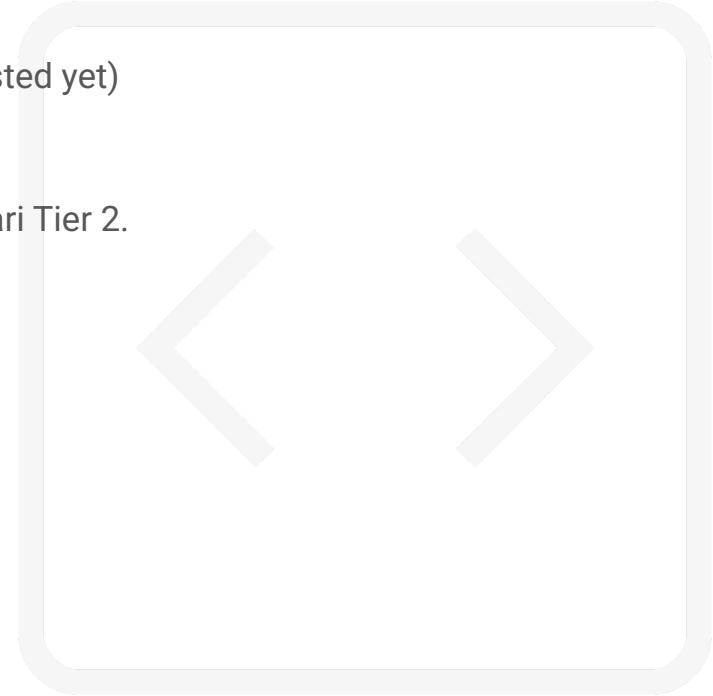
- Since direct printing isn’t available in browsers, we switched to generating PDFs.

System Settings Access

- Browsers restrict direct access to system settings for security reasons.
- We adapted by creating a virtual settings interface helped by Qt framework, which simulates system setting operations like in operating systems

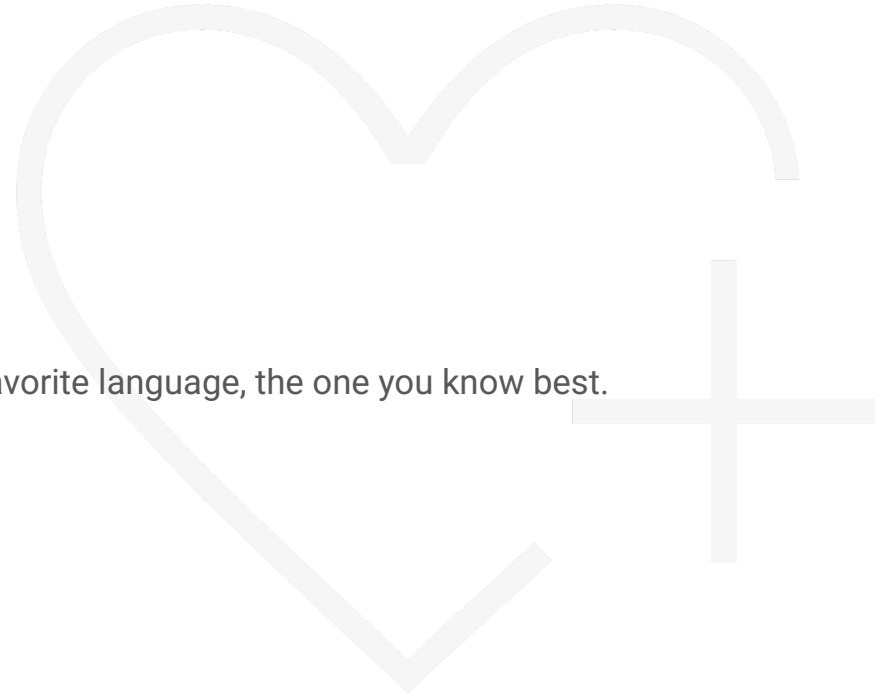
Technical Insights: Debugging

- Lately Qt helps to test and debug (but we haven’t tested yet)
- Console
- `DISABLE_EXCEPTION_CATCHING=0`
- Selecting a browser: Chrome Tier 1, Firefox and Safari Tier 2.



Benefits

- Performance & Speed & Rich interface
- Simplified Development
- Simplified Deployment
- Webbrowser Security
- Enhanced User Experience
- Code in C++.
    - The Success of WASM: Code in your favorite language, the one you know best.



Lanzadera Velneo 35

apps1.velneo.net/#

Velneo 35

jarboleya

Favoritas

★ vERP

★ vGest

★ Facturas

Aplicaciones

Partes de trabajo

vForwarding

Challenges

Browsers and operating systems

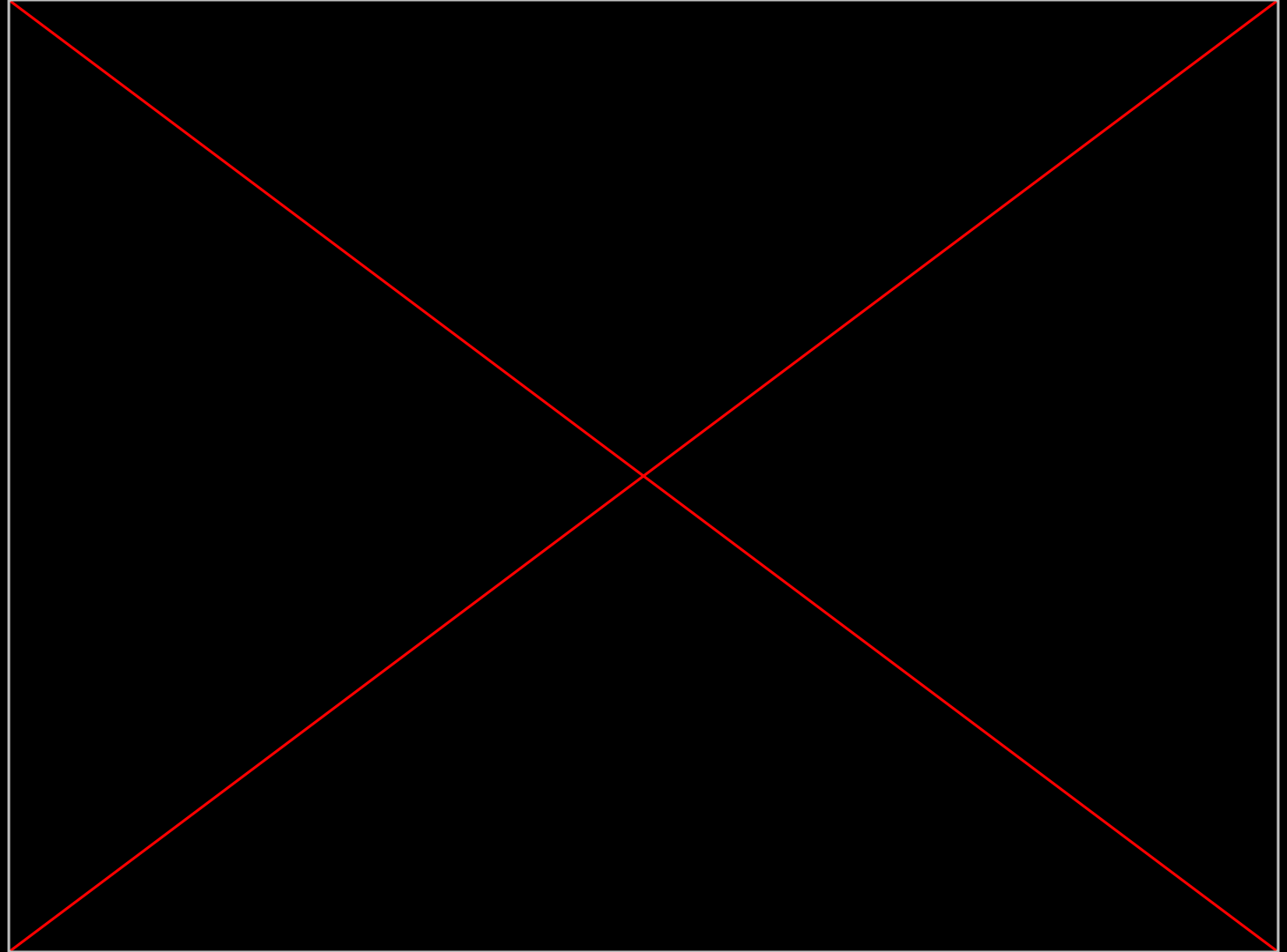
- Testing and debugging Complexity

Resource Management

- Threads and memory

Conversion to Asynchronous code

- Full covering (e.g. QML & QJSEngine scripts)



Future

- Mobile
- JSPI vs Asyncify vs Refactor code
- Direct Sockets API vs WebSockets
- 64 bits
- More help from Qt for WebAssembly
- Stand only apps with server inside

Questions?

Fernando Félix Gutiérrez Blanco
Velneo Development Team Manager
fgutierrez@velneo.com
X/Twitter @fgutierrez_
https://velneo.com

velneo