[](https://imasters.com.br/perfil/thomaspeham)

[**THOMAS PEHAM**](https://imasters.com.br/perfil/thomaspeham)

13

É um gerente de marketing na startup austríaca Usersnap. É especializado em inbound &amp; content marketing. Em seu tempo livre, escreve sobre tendências de marketing e de tecnologia em seu blog. Ele pode ser encontrado no Twitter: @tompeham.

[LEIA MAIS](https://imasters.com.br/perfil/thomaspeham)

18 ABR, 2016

[**Plugins do WordPress essenciais para desenvolvedores e designers – Parte 02**](https://imasters.com.br/back-end/plugins-do-wordpress-essenciais-para-desenvolvedores-e-designers-parte-02)

15 MAR, 2016

[**Por que ninguém fala sobre testes Agile…**](https://imasters.com.br/agile/por-que-ninguem-fala-sobre-testes-agile)

21 JAN, 2016

[**5 dicas essenciais para melhorar mockups de website**](https://imasters.com.br/design-ux/5-dicas-essenciais-para-melhorar-mockups-de-website)

Você já ouviu alguma notícia sobre WebAssembly? Google, Microsoft, Mozilla e outros estão se unindo para lançar um novo formato binário para a web.

Vamos direcionar um olhar mais atento para o que é WebAssembly e por que seu aspecto e desenvolvimento futuro devem importar para todos.

Ele vai mudar a web. Pelo menos alguma extensão dela.

[](https://static.imasters.com.br/wp-content/uploads/2015/11/webassembly-1.jpg)

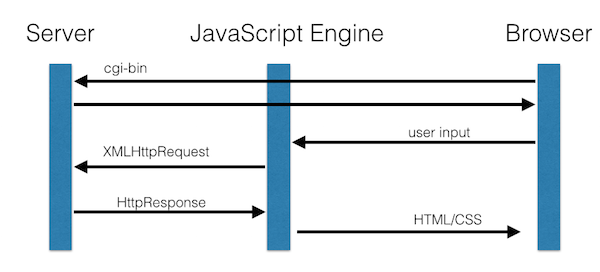
**JavaScript – o idioma da web**

A web é baseada em vários padrões, o que é absolutamente bom. E isso faz com que a vida de construtores e criadores de código seja mais fácil. Imagine que não existisse quaisquer regra, linguagem, estrutura ou conceito sob os quais construiríamos sites e aplicações.

O padrão web que detém diversos conjuntos de informações tem um nome bem conhecido. É chamado de JavaScript.

O JavaScript é basicamente a linguagem de programação padrão para plataformas web. E como cada vez mais os softwares se tornam softwares web, o JavaScript ganhou uma imensa popularidade.

Tem havido algumas tentativas ao longo dos anos para contornar as limitações do JavaScript, o que ele definitivamente tem. Arquivos JavaScript são arquivos de texto simples. Esses arquivos são baixados do servidor e, em seguida, analisados e compilados pelo motor JavaScript no navegador.

[](https://static.imasters.com.br/wp-content/uploads/2015/11/webassembly-2.png)

Os navegadores atualmente usam JavaScript para interpretar código e habilitar funcionalidades, como conteúdo dinâmico. Melhorias de desempenho com frameworks JavaScript também foram feitas. No entanto, os sistemas baseados em bytecode são ainda mais rápidos e mais eficientes.

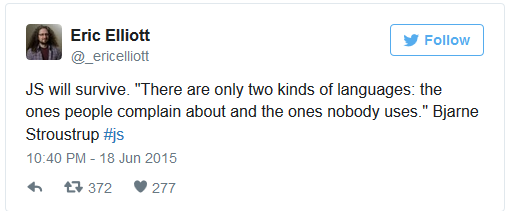
**Então, o que diabos é WebAssembly?**

WebAssembly é uma espécie de novo código de byte. WebAssembly – ou abreviado: “wasm” – é uma nova sintaxe para código binário seguro de baixo nível. Ele é definido como “um formato portátil para execução, carregamento, verificação e exibição de modelos de código”, projetado para servir como um alvo de compilação para a web.

Isso definitivamente significa melhorias de desempenho no navegador. E nos dá acesso a um conjunto de blocos de construção de baixo nível, como uma gama de tipos e operações.

Não me interpretem mal. Ele não é anunciado e concebido como uma alternativa ao JavaScript.

E como Bjarne Stroustup declarou:

[](https://static.imasters.com.br/wp-content/uploads/2015/11/webassembly-3.png)*(Tradução: JS vai sobreviver. “Há apenas dois tipos de linguagem: aquelas sobre as quais as pessoas reclamam e aquelas que ninguém usa”.)*

Ou como [Eric Elliott afirmou:](https://medium.com/javascript-scene/what-is-webassembly-the-dawn-of-a-new-era-61256ec5a8f6) “Pode ser útil não pensar em WebAssembly como uma linguagem de fato. Ele é mais como uma máquina”.

**De asm.js para WebAssembly?**

O [asm.js](http://asmjs.org/) é ótimo para código na compilação escrito – por exemplo – em C ou C ++. O asm.js é um subconjunto do JavaScript, que pode ser usado em plataformas de baixo nível, e é uma linguagem de destino eficiente para compiladores.

A ideia do asm.js é codar o JavaScript de uma forma que os motores de compilação produzam código de máquina mais eficiente. Se você compilar C ++ para asm.js, poderá alcançar grandes melhorias de velocidade em navegadores web.

[](https://static.imasters.com.br/wp-content/uploads/2015/11/webassembly-4.png)

O WebAssembly tem algumas vantagens sobre o asm.js que envolvem principalmente problemas de desempenho. De acordo com o FAQ do WebAssembly: “Em dispositivos móveis, grandes códigos compilados podem facilmente levar 20-40s apenas para serem analisados […] As primeiras experiências mostram que o WebAssembly pode ser carregado mais do que 20 vezes mais rapidamente, porque o trabalho de análise é mínimo”.

**O que isso tem de tão incrível, afinal?**

Então, provavelmente você está perguntando: por que toda essa febre sobre a notícia? O WebAssembly é definitivamente uma melhoria para o JavaScript. Mas não apenas isso. É uma grande melhoria para o navegador também.

Os navegadores vão entender o formato binário. Então você vai ser capaz de compilar pacotes binários que comprimem muito mais do que um conteúdo JavaScript.

Isso poderia trazer a “performance parecida com aplicativos” para todos os aplicativos de conteúdos da web. Parece ótimo, né? Em vez de ter que analisar o código completo, porém, que muitas vezes pode demorar um pouco (especialmente no celular), o WebAssembly pode ser decodificado significativamente mais rápido.

Veja esta conversa por Brendan Eich:

**Quem já está nessa?**

Google, Microsoft e Mozilla são apenas alguns nomes. Engenheiros líderes lançaram uma Comunidade WebAssembly com a missão de “promover a colaboração cross-browser em estágio inicial em formato portátil do novo, […] formato para compilação na web”.

No entanto, é necessário lembrar que a obra “acaba de começar” no WebAssembly. Então este é um estágio muito cedo do projeto e vai certamente demorar algum tempo até que o WebAssembly se torne um padrão web bem conhecido.

**Por que isso afetará todos os desenvolvedores web?**

Porque ele vai permitir que os desenvolvedores façam todo tipo de coisas novas que não teriam sido possíveis ou tão fáceis com JavaScript.

A comunidade [W3C WebAssembly](https://www.w3.org/community/webassembly/) tem poucos casos de uso para o WebAssembly e como ele vai facilitar a vida dos desenvolvedores web. Aqui estão alguns:

* A melhor forma de execução das linguagens de programação é cross-compilada para a web.
* Para o desenvolvedor das ferramentas, dentro do navegador.
* Cliente mais rápido para aplicações empresariais (por exemplo, bancos de dados).

O WebAssembly pode ser usado de várias maneiras. Por exemplo, pode ser utilizado em código existente JavaScript/HTML que pode ser incorporado. Por outro lado, o WebAssembly pode ser usado como o framework principal, enquanto a IU (interface de usuário) ainda é feita em JavaScript / HTML.

**Menos código, melhor desempenho, menos bugs?**

De acordo com o grupo WebAssembly, o simples fato de utilizar WebAssembly significa menos código-fonte. Em comparação com asm.js, isso significa uma redução de cerca de 25% no tamanho do código – embora ele seja apenas um protótipo, e os resultados variam.

Então, provavelmente você está querendo saber como manter-se atualizado sobre WebAssembly? Eu recomendo os seguintes recursos úteis para WebAssembly:

* [Página do GitHub WebAssembly](https://github.com/WebAssembly)
* [Documentação de projeto WebAssembly](https://github.com/WebAssembly/design/blob/master/BinaryEncoding.md)
* [Comunidade W3C WebAssembly](https://www.w3.org/community/webassembly/)
* [O que é WebAssembly por Eric Elliott](https://medium.com/javascript-scene/what-is-webassembly-the-dawn-of-a-new-era-61256ec5a8f6)

\*\*\*

[Homepage](https://medium.com/)

[](https://medium.com/javascript-scene?source=logo-lo_dnt_E7GpRM1saYlz---c0aeac5284ad)

Follow

[Sign in](https://medium.com/m/signin?redirect=https%3A%2F%2Fmedium.com%2Fjavascript-scene%2Fwhat-is-webassembly-the-dawn-of-a-new-era-61256ec5a8f6&source=--------------------------nav_reg&operation=login)[Get started](https://medium.com/m/signin?redirect=https%3A%2F%2Fmedium.com%2Fjavascript-scene%2Fwhat-is-webassembly-the-dawn-of-a-new-era-61256ec5a8f6&source=--------------------------nav_reg&operation=register)

* [HOME](https://medium.com/javascript-scene)
* [LEARN JS](https://medium.com/javascript-scene/tagged/learnjs)
* [VIDEO LESSONS](https://ericelliottjs.com/product/lifetime-access-pass/)

[[](https://medium.com/@_ericelliott?source=post_header_lockup)](https://medium.com/@_ericelliott?source=post_header_lockup)

[Eric Elliott](https://medium.com/@_ericelliott?source=post_header_lockup)Follow

Make some magic. #JavaScript

Jun 18, 2015

# What is WebAssembly?

## The Dawn of a New Era

The future of the web platform looks brighter than ever.

*Yesterday,****[Brendan Eich dropped a bomb](https://brendaneich.com/2015/06/from-asm-js-to-webassembly/" \t "_blank)****on the web development community: The web platform is getting a new low-level binary compile format that will do a better job at being a compiler target than JavaScript.*

Google, Microsoft, Mozilla, and a few other people have been secretly toiling away in a new [W3C WebAssembly Community Group](https://www.w3.org/community/webassembly/" \t "_blank), and what they have been working on is no small thing.

For more depth, see the follow-up article, [“Why We Need WebAssembly: An Interview with Brendan Eich”](https://medium.com/javascript-scene/why-we-need-webassembly-an-interview-with-brendan-eich-7fb2a60b0723" \t "_blank).

### ****WebAssembly is:****

* **An improvement to JavaScript:** Implement your performance critical stuff in wasm and import it like a standard JavaScript module.
* **A new language:** WebAssembly code defines an AST (Abstract Syntax Tree) represented in a **binary format**. You can **author and debug in a text format** so it’s readable.
* **A browser improvement:** **Browsers will understand the binary format**, which means we’ll be able to compile binary bundles that compress smaller than the text JavaScript we use today. Smaller payloads mean faster delivery. Depending on **compile-time optimization opportunities**, WebAssembly bundles may run faster than JavaScript, too!
* **A Compile Target:** A way for other languages to get first-class binary support across the entire web platform stack.

### What does this mean for JavaScript?

Before we answer this question, let’s back up a moment. Let me take you back in time… back… back…, back before React, before Angular, before Backbone, before jQuery…

Ah, here we are.

*The web is a scattering of hypertext documents on bulletin board systems, not yet interlinked. The first web server is being hacked together on a NeXT workstation at CERN…*



NeXT Computer used by [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee" \t "_blank" \o "Tim Berners-Lee) at [CERN](https://en.wikipedia.org/wiki/CERN" \t "_blank" \o "CERN)

The year is 1991 and my hair isn’t gray yet. I’m hacking together my ten thousandth text adventure game (something like that, I’m not counting).

I made a peculiar language choice for this one. I was sick of working in BASIC and Pascal. I wanted to use C, but I was still saving up for my first Borland Turbo C++ **boxed set** (they literally came in boxes packed with manuals and install disks). I didn’t even have Turbo Assembler, yet.

I was writing in assembly language and “compiling” to an executable with the DOS `debug` command line tool. If that sounds crazy, believe me, it was. I’m betting even those of you who used DOS probably didn’t realize you could use debug to assemble instructions, as well as disassemble (reverse engineer) existing code.

Sound cool? **I hated it.**I couldn’t wait to get my hands on Borland Turbo C++ so I could write code like a human being. Eventually, I got it as a gift. Score!

But **one of my favorite things** about Borland Turbo C++ was that **it came bundled with Borland Turbo Assembler**. What?! Why would you want to write code in assembly language when you have a great high-level object-oriented language like C++ in your toolbelt?

**Sometimes you want to get to the bare metal,** or as close to it as you can get without melting your brain. Did I mention that I also wrote a lot of machine language prior to getting my hands on C++?

*I lost my mind.*



Losing My Mind — Mark Auer (CC BY-NC 2.0)

It’s much harder to get real work done when you’re writing directly in assembly language. So **why do we need this WebAssembly thing?**

**We need WebAssembly** because as flexible as JavaScript is, it’s still too hard to express many of the things we may want to in JavaScript, and the features we’d need to make it easy might add complexity to a language that already [confuses many users](https://medium.com/javascript-scene/javascript-training-sucks-284b53666245" \t "_blank).

**WebAssembly gives us access to a set of low level building blocks** that we can use to construct just about anything you can imagine.

How is this different from JavaScript? The key words are **low-level**. It defines primitives including [a range of types](https://github.com/WebAssembly/design/blob/master/AstSemantics.md" \l "local-and-memory-types" \t "_blank) and operations on those types, literal forms for them, control-flow, calls, a heap, etc…

These are very simple primitives. Nothing fancy. **No complicated object system** (prototypal or otherwise). **No built-in automatic garbage collector**following you around and stopping you periodically while it cleans up your scraps.

### What Exactly is WebAssembly?

WebAssembly defines an [Abstract Syntax Tree](https://github.com/WebAssembly/design/blob/master/AstSemantics.md" \l "abstract-syntax-tree-semantics" \t "_blank) (AST) that gets stored in a **[binary format](https://github.com/WebAssembly/design/blob/master/BinaryEncoding.md" \t "_blank)**. Binary is great because it means we can create [smaller app bundles](https://github.com/WebAssembly/design/blob/master/BinaryEncoding.md" \l "why-a-binary-encoding-instead-of-a-text-only-representation" \t "_blank). You’re probably wondering how we’ll debug a binary language format.

Luckily, while we’re stepping through the debugger that will inevitably appear in browsers, the AST will be represented in a (moderately) friendly [**text format**](https://github.com/WebAssembly/design/blob/master/TextFormat.md). I’d love to show you examples, but they’re a bit scarce right now. It will probably be a bit less readable than handwritten JavaScript, but **about as readable as compiled asm.js**. Maybe more readable. We’ll see.

### What will WebAssembly be used for?

Among other things, it will be easy to express things like threads and **SIMD —**a fancy word that means you can line up multiple chunks of data next to each other and invoke a single instruction to operate on all of them at the same time. It stands for **S**ingle **I**nstruction, **M**ultiple **D**ata.

That means fat, **parallel processing pipelines** for your realtime video stream effects processor. If you have your finger on the pulse, you’ve probably already heard of doing this in JS, but I’ve always found it a bit awkward to try to do low level stuff like this using JavaScript’s existing type system.

This is one of those cases where you’ll probably want to forget about the object system, the garbage collector, and all the fancy dynamic stuff. Just line up some raw bits in little rows and crunch through them as fast as possible.

#### Show me the Apps

Games, VR and augmented reality are the obvious examples. Most of the current WebAssembly demos use Unity or Unreal Engine, both of which already support compiling to asm.js. Right now, music production apps like **[Ableton Live](https://www.ableton.com/" \t "_blank)** and video production apps like [**Adobe Premier Pro**](http://www.adobe.com/products/premiere.html) are still a bit awkward to port to the web. Not impossible, mind you, just awkward. There are still lots of problems to solve, like better timing guarantees for data intensive realtime apps.

We’ll also see apps that stream big channels of data through networks of processing functions like a **[guitar effects pedalboard](http://dashersw.github.io/pedalboard.js/demo/" \t "_blank)**. These are traditionally things that most people don’t think of when they think of JavaScript. A lot of people would probably think it’s crazy to even try.

But **JavaScript is really a great language** to build **most of the code** required by any app you can dream up.

***WebAssembly fills in the gaps*** *that would be awkward to fill  
with JavaScript.*

Those gaps are no secret. Even JavaScript’s big fans would admit that there are times when **we’re stretching the G-forces JavaScript can handle**without losing its lunch. Up until yesterday, I thought that the plan would be to fill those gaps by adding complexity to the JavaScript language itself. Brendan Eich talked about that in a Fluent conference keynote that I applauded:

But what some of us have really been missing is the ability to **write most of the code in an amazing high-level language** and still be able to **drop down to a specialized, bare metal assembly language** once in a while when we really need a boost.

*WebAssembly is JavaScript’s****nitrous boost!***

I’m sure there are 200 other articles on WebAssembly being published today all over the web. I’ll bet most of them will focus on the other side of WebAssembly, which I’m equally excited about…

### WebAssembly Brings Language Diversity to the Web Platform

We don’t really need WebAssembly to bring other languages to the web platform. **We already have the best [AAA game](https://www.unrealengine.com/what-is-unreal-engine-4" \t "_blank) [engines](https://unity3d.com/unity/multiplatform" \t "_blank)** running smoothly at stunning quality by **compiling to JavaScript**.

*In case you’ve heard  
JavaScript is slow…*it isn’t.

WebAssembly adds things that most JS developers would agree **we don’t need in JavaScript.** We may still get those things, but not because we need them for JavaScript code. We’ll get them to support **compiling from other languages** which use them.

WebAssembly gives us an **alternative compile target —**One specifically designed for that purpose.

It will be easier now to port code that relies heavily on features like shared memory threads. I’d bet that the process of writing a compiler for WebAssembly will be somewhat less complicated than writing a compiler for JavaScript, simply because there’s a better mapping from the source language features to the target AST.

As great as it is to hear that all the old languages we know and love will run on the web platform now, WebAssembly means one more very important thing:

WebAssembly is an open invitation *to developers building*

***future programming languages.***

The future of the web platform has never looked brighter. You’d better pull out your shades.

### Update / FAQ

#### What’s wasm?

Short for **W**eb**As**se**m**bly.

#### Why not use the JVM?

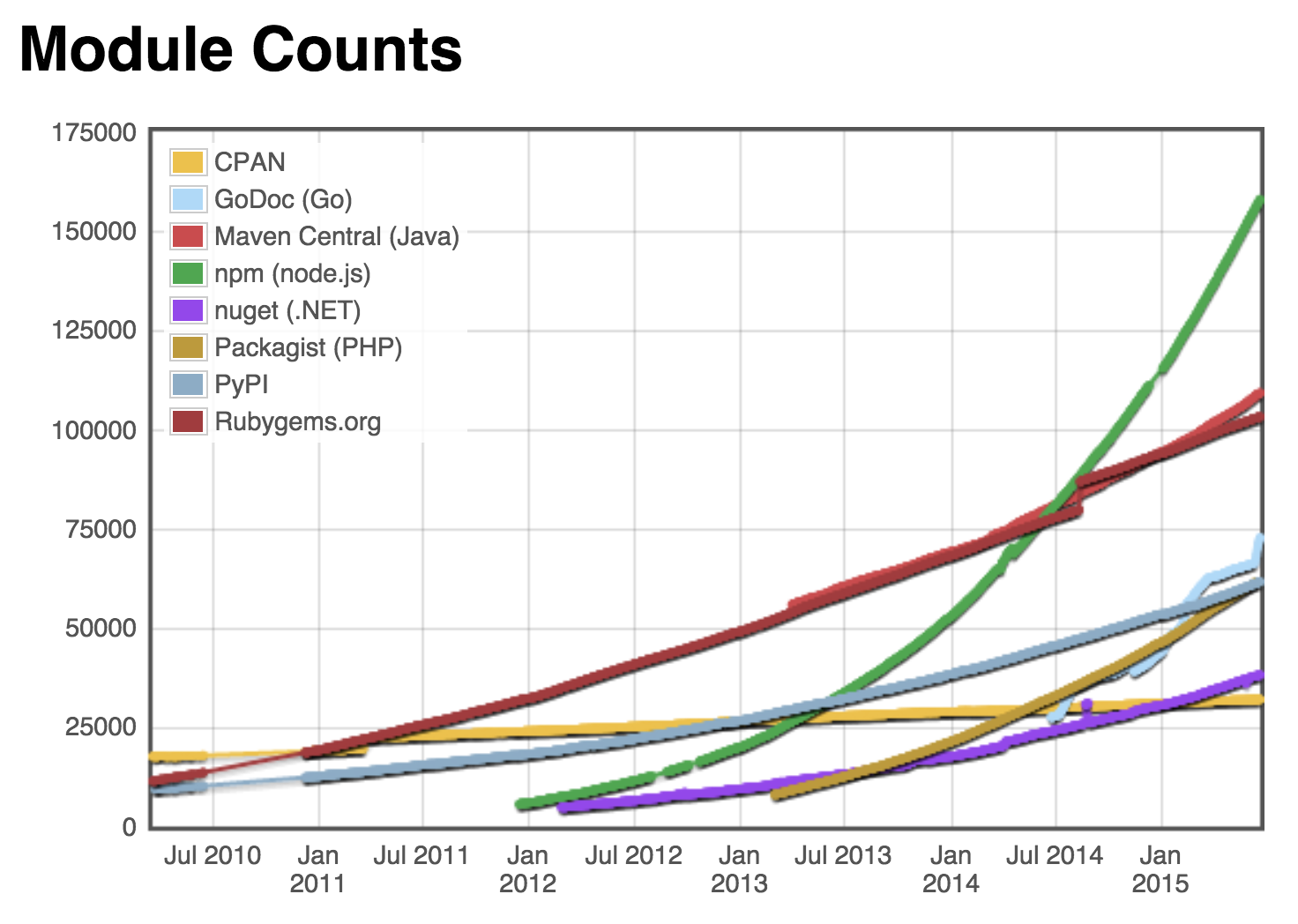
We’ve tried to put the JVM in browsers before via plugins. It didn’t work out so well. JavaScript already has a VM. Another VM means a second set of API hooks to give the VM access to the DOM, networking, sensors, input devices, etc… That’s not free. How will VM processes like garbage collection get along competing for the same resources? That may be harder than it sounds.

Initially, WebAssembly will run on an asm.js polyfill, meaning it can take advantage of the **existing JavaScript VM foundations.** The design will evolve from that base, so there is a smoother browser integration path than alternative VMs could provide.

#### Q: Doesn’t this mean other languages will take over? Won’t this cause fragmentation?

A: **JavaScript has always had strong competition** on the server side and in embedded programming for small hardware and robots, but in spite of many existing, viable alternatives with established package ecosystems and lots of trained developers, **Node is still taking over startup and enterprise web servers at a torrential pace.**

JavaScript also has significant network lock-in with the ecosystem and developer community. I love posting the modulecounts graph because it gets more impressive every time:



That green line there is **npm,** the standard package repository for JavaScript. It comes bundled with Node.

JavaScript is also becoming a popular choice for game programming, robotics, and IoT devices, all of which have very strong competition from C, C++, and Java. This has little to do with JavaScript’s dominant position as a web browser language. Those developers have choices, and they’re choosing JavaScript because **they like it**.

#### What are the Text Formats Like?

The most popular is a lisp-like S-expression form. For a taste, check out the [WebAssembly Playground](http://ast.run/" \t "_blank).

#### Where Can I Learn More?

See the follow-up article, [“Why We Need WebAssembly: An Interview with Brendan Eich”](https://medium.com/javascript-scene/why-we-need-webassembly-an-interview-with-brendan-eich-7fb2a60b0723" \t "_blank).

2015 LLVM Developers Meeting talk by JF Bastien and Dan Gohman

#### How can I stay in the loop?

* [W3C WebAssembly Community Group](https://www.w3.org/community/webassembly/)
* [Mailing list](http://lists.w3.org/Archives/Public/public-webassembly/)
* IRC: irc://irc.w3.org:6667/#webassembly
* [GitHub](https://github.com/webassembly)
* [Who’s involved?](https://www.w3.org/community/webassembly/participants)

[*Learn JavaScript with  
Eric Elliott*](https://ericelliottjs.com/)

**Eric Elliott** is the author of [“Programming JavaScript Applications”](http://pjabook.com/" \t "_blank) (O’Reilly), and [“Learn JavaScript Universal App Development with Node, ES6, & React”](https://leanpub.com/learn-javascript-react-nodejs-es6/" \t "_blank). He has contributed to software experiences for **Adobe Systems**, **Zumba Fitness**,**The Wall Street Journal**, **ESPN**, **BBC**, and top recording artists including **Usher**, **Frank Ocean**, **Metallica**, and many more.

He spends most of his time in the San Francisco Bay Area with the most beautiful woman in the world.

* [Webassembly](https://medium.com/tag/webassembly?source=post)
* [JavaScript](https://medium.com/tag/javascript?source=post)
* [Html5](https://medium.com/tag/html5?source=post)
* [Tech](https://medium.com/tag/tech?source=post)
* [Technology](https://medium.com/tag/technology?source=post)

Like what you read? Give Eric Elliott a round of applause.

From a quick cheer to a standing ovation, clap to show how much you enjoyed this story.

2.7K

27

 Follow

[[](https://medium.com/@_ericelliott?source=footer_card)](https://medium.com/@_ericelliott?source=footer_card" \o "Go to the profile of Eric Elliott)

### [Eric Elliott](https://medium.com/@_ericelliott" \o "Go to the profile of Eric Elliott)

Medium member since Feb 2018

Make some magic. #JavaScript

 Follow

[](https://medium.com/javascript-scene?source=footer_card)

### [JavaScript Scene](https://medium.com/javascript-scene?source=footer_card)

To submit, DM your proposal to [@JS\_Cheerleader](https://medium.com/@JS_Cheerleader" \o "Medium profile for @JS_Cheerleader) on Twitter

[More from JavaScript Scene](https://medium.com/javascript-scene/10-interview-questions-every-javascript-developer-should-know-6fa6bdf5ad95?source=placement_card_footer_grid---------0-41)

[10 Interview Questions Every JavaScript Developer Should Know](https://medium.com/javascript-scene/10-interview-questions-every-javascript-developer-should-know-6fa6bdf5ad95?source=placement_card_footer_grid---------0-41)

[[Go to the profile of Eric Elliott](https://medium.com/@_ericelliott)](https://medium.com/@_ericelliott)

[Eric Elliott](https://medium.com/@_ericelliott?source=placement_card_footer_grid---------0-41)

22K

[Also tagged JavaScript](https://medium.com/tech-tajawal/modern-backend-developer-in-2018-6b3f7b5f8b9?source=placement_card_footer_grid---------1-43)

[Modern Backend Developer in 2018](https://medium.com/tech-tajawal/modern-backend-developer-in-2018-6b3f7b5f8b9?source=placement_card_footer_grid---------1-43)

[Go to the profile of Adnan Ahmed](https://medium.com/@idnan)

[Adnan Ahmed](https://medium.com/@idnan?source=placement_card_footer_grid---------1-43)

19.3K

[Also tagged Tech](https://medium.freecodecamp.org/we-fired-our-top-talent-best-decision-we-ever-made-4c0a99728fde?source=placement_card_footer_grid---------2-43)

[We fired our top talent. Best decision we ever made.](https://medium.freecodecamp.org/we-fired-our-top-talent-best-decision-we-ever-made-4c0a99728fde?source=placement_card_footer_grid---------2-43)

[Go to the profile of Jonathan Solórzano-Hamilton](https://medium.freecodecamp.org/@peachpie)

[Jonathan Solórzano-Hamilton](https://medium.freecodecamp.org/@peachpie?source=placement_card_footer_grid---------2-43)

90K

**Responses**

Write a response…

Conversation with [Eric Elliott](https://medium.com/@_ericelliott).

[Go to the profile of hon](https://medium.com/@qhc004)

[hon](https://medium.com/@qhc004?source=responses---------0----------------)

[Dec 3, 2016](https://medium.com/@qhc004/great-article-115ab6488d9?source=responses---------0----------------)

[Great article. I wonder can webassembly run without browser? Is there a way to run it as a native app, in the future?](https://medium.com/@qhc004/great-article-115ab6488d9?source=responses---------0----------------)

4

[1 response](https://medium.com/@qhc004/great-article-115ab6488d9?source=responses---------0----------------#--responses)

[[Go to the profile of Eric Elliott](https://medium.com/@_ericelliott)](https://medium.com/@_ericelliott)

[Eric Elliott](https://medium.com/@_ericelliott?source=responses---------0----------------)

[Dec 3, 2016](https://medium.com/@_ericelliott/depends-on-what-you-mean-by-without-a-browser-86e532dba7d3?source=responses---------0----------------)

[Depends on what you mean by “without a browser”. If you mean look and feel like a native app, the answer is yes, eventually. See “Native Apps are Doomed”](https://medium.com/@_ericelliott/depends-on-what-you-mean-by-without-a-browser-86e532dba7d3?source=responses---------0----------------)

3

[2 responses](https://medium.com/@_ericelliott/depends-on-what-you-mean-by-without-a-browser-86e532dba7d3?source=responses---------0----------------#--responses)

[Go to the profile of hon](https://medium.com/@qhc004)

[hon](https://medium.com/@qhc004?source=responses---------0----------------)

[Dec 4, 2016](https://medium.com/@qhc004/i-meant-run-directly-on-os-as-a-true-native-app-rather-than-on-browser-environment-cordova-on-25b24940f120?source=responses---------0----------------)

[I meant: run directly on OS (as a true native app) rather than on browser environment(cordova on mobile or electron on desktop).](https://medium.com/@qhc004/i-meant-run-directly-on-os-as-a-true-native-app-rather-than-on-browser-environment-cordova-on-25b24940f120?source=responses---------0----------------)

[2 responses](https://medium.com/@qhc004/i-meant-run-directly-on-os-as-a-true-native-app-rather-than-on-browser-environment-cordova-on-25b24940f120?source=responses---------0----------------#--responses)

[[Go to the profile of Eric Elliott](https://medium.com/@_ericelliott)](https://medium.com/@_ericelliott)

[Eric Elliott](https://medium.com/@_ericelliott?source=responses---------0----------------)

[Dec 4, 2016](https://medium.com/@_ericelliott/new-versions-of-android-google-will-treat-pwas-just-like-native-apps-they-will-show-up-in-the-all-bdbf02df36ae?source=responses---------0----------------)

[New versions of Android Google will treat PWAs just like native apps: they will show up in the all apps view. Notifications will be properly attributed to them, etc…](https://medium.com/@_ericelliott/new-versions-of-android-google-will-treat-pwas-just-like-native-apps-they-will-show-up-in-the-all-bdbf02df36ae?source=responses---------0----------------)

[Microsoft has pledged to index PWAs automatically in their app store. Google and Apple have not yet followed suit, but may in the future. Amazon’s app store already indexes web apps.](https://medium.com/@_ericelliott/new-versions-of-android-google-will-treat-pwas-just-like-native-apps-they-will-show-up-in-the-all-bdbf02df36ae?source=responses---------0----------------)

6

Show all responses

* 2.7K

[](https://medium.com/javascript-scene)

Never miss a story from**JavaScript Scene**, when you sign up for Medium. [Learn more](https://medium.com/@Medium/personalize-your-medium-experience-with-users-publications-tags-26a41ab1ee0c" \l ".hx4zuv3mg)

# Quem é Kamala Khan, a heroína muçulmana da Marvel que pode virar filme

## Conheça a jovem que assumiu o posto de Miss Marvel nos quadrinhos e que está nos planos do estúdio

Por **Da Redação**

access\_time4 jul 2018, 21h39 - Publicado em 4 jun 2018, 17h49

[chat\_bubble\_outline](https://super.abril.com.br/mundo-estranho/quem-e-kamala-khan-a-heroina-muculmana-da-marvel-que-pode-virar-filme/#respond)more\_horiz



 (Divulgação/Reprodução)

Capitã Marvel, primeiro filme do Universo Cinematográfico da Marvel estrelado por uma mulher, chegará aos cinemas no começo de 2019. E, se depender do presidente do estúdio, Kevin Feige, não demorará até que outra heroína dos quadrinhos apareça nas telas. Em entrevista à [**BBC**](https://twitter.com/StaarksHeart/status/995108155976757248), ele diz que há a possibilidade de inserir a jovem Kamala Khan (ou Miss Marvel) no UCM após o longa do ano que vem.

“Nós temos planos para isso, assim que apresentarmos a Capitã Marvel para o mundo”, disse Feige. Sem revelar mais detalhes, a declaração bastou para aquecer o coração dos fãs da personagem.

**Afinal, quem é Kamala Khan?**

Criada em 2014, Kamala é uma adolescente de origem paquistanesa e que mora com a família em Nova Jersey, nos EUA. Nerd, ela escreve fan fictions dos Vingadores, frequenta (a contragosto) a mesquita do bairro e, por seguir as tradições de sua religião, se sente diferente em relação ao resto do colégio, o que não raro a deixa chateada.

Em **“**[**Miss Marvel, Nada Normal**](https://loja.panini.com.br/panini/produto/marvel-ms-marvel-nada-normal.aspx)**“**, sua HQ de estreia, vemos como ela conseguiu os seus poderes. Voltando de uma festa, Kamala se depara com uma estranha nuvem que cobre toda a cidade. Ao entrar em contato com ela, a garota começa a ter alucinações com a sua ídolo: Carol Danvers, a Miss Marvel original e que, hoje em dia, assumiu o nome de Capitã.

Após um desmaio, ela percebe que o seu corpo está diferente. Dentre as suas novas (e poderosas) habilidades, estão o fator de cura acelerado e a capacidade de esticar, aumentar, diminuir e alterar a sua forma física da maneira que ela quiser. Tempos depois, descobre-se que Kamala possui genes inumanos, que foram ativados com a tal nuvem que, na verdade, era o resultado de uma bomba terrígena, lançada por Raio Negro, líder dos Inumanos.



 (Divulgação/Reprodução)

**A origem da personagem**

Com uma pequena ajuda de um amigo, ela improvisa um uniforme, adota o antigo nome de sua heroína favorita e começa a combater o crime. Das características da sua roupa ao significado de “Kamala”, todo esse novo universo busca referências na cultura muçulmana. As responsáveis por isso? A quadrinista G. Willow Wilson e a editora Sana Amanat, que desenvolveram a ideia de criar uma heroína que representasse o Islã na Marvel.

Wilson é muçulmana e Amanat possui ascendência paquistanesa, assim como a nova Miss Marvel. A editora se inspirou em suas experiências da adolescência em Nova Jersey para criar o pano de fundo da história, cheio de detalhes sobre uma cultura com a qual não estamos acostumados. O dia a dia de Kamala com seus rígidos pais, as tradições religiosas do seu bairro e até as restrições alimentares na cantina do colégio. Está tudo ali.



 (Divulgação/Reprodução)

**O que esperar de um filme da heroína**

A criação da dupla rendeu bons frutos. Em 2015, a HQ venceu o Prêmio Hugo de Melhor Graphic Story, dado às melhores histórias de fantasia ou ficção científica publicadas nesse formato. A personagem ganhou rapidamente o carinho do público, muito por conta de sua personalidade e de toda a euforia que ela demonstra por andar lado a lado com os seus heróis favoritos.

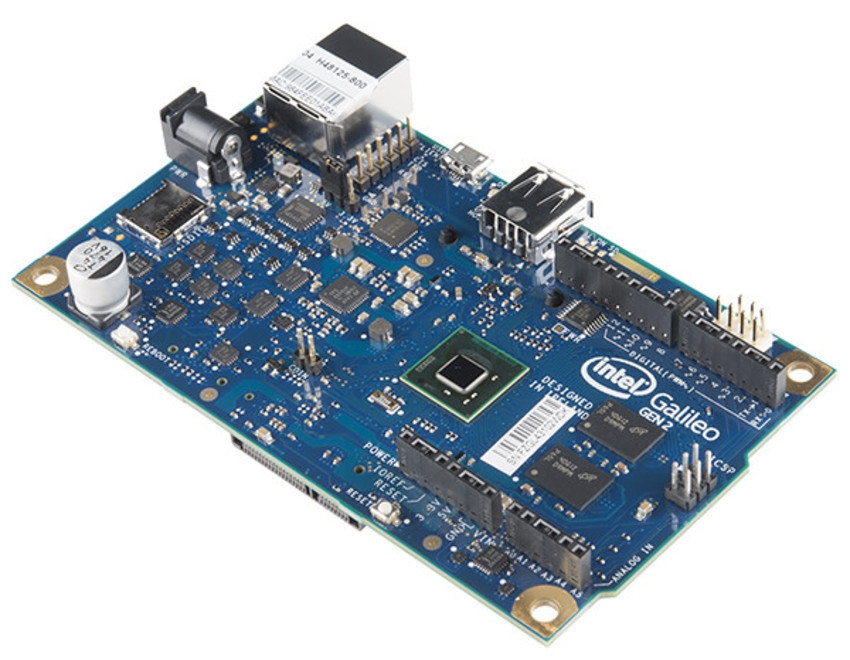
Um filme da nova Miss Marvel pode trazer tanta representatividade quanto Pantera Negra trouxe ao tratar de temas como racismo e cultura africana. É o que diz a jornalista Coco Khan em um artigo para o jornal britânico **[The Guardian](https://www.theguardian.com/commentisfree/2018/may/17/ms-marvel-female-muslim-superhero-kamala-khan" \t "_blank).**Para ela, uma aventura com uma super-heroína muçulmana poderia ajudar a combater o preconceito contra a religião e a diminuir os estereótipos de personagens islâmicos em Hollywood, raramente vistos em papéis principais.



# Primeiros passos com Intel Galileo Gen 2[17](https://www.filipeflop.com/blog/primeiros-passos-intel-galileo-gen-2/#comments)

[Embarcados](https://www.filipeflop.com/blog/category/embarcados/)  [23 de março de 2015](https://www.filipeflop.com/blog/primeiros-passos-intel-galileo-gen-2/) [Adilson Thomsen](https://www.filipeflop.com/blog/author/adilsonth/)

A placa Intel Galileo Gen 2 é um projeto que une a simplicidade do Arduino com a reconhecida performance e qualidade dos processadores Intel. O resultado disso é uma placa compatível com shields e sensores usados no Arduino, mas que também pode ser utilizada com os sistemas operacionais mais comuns, criando um ambiente de integração entre esses dois mundos.



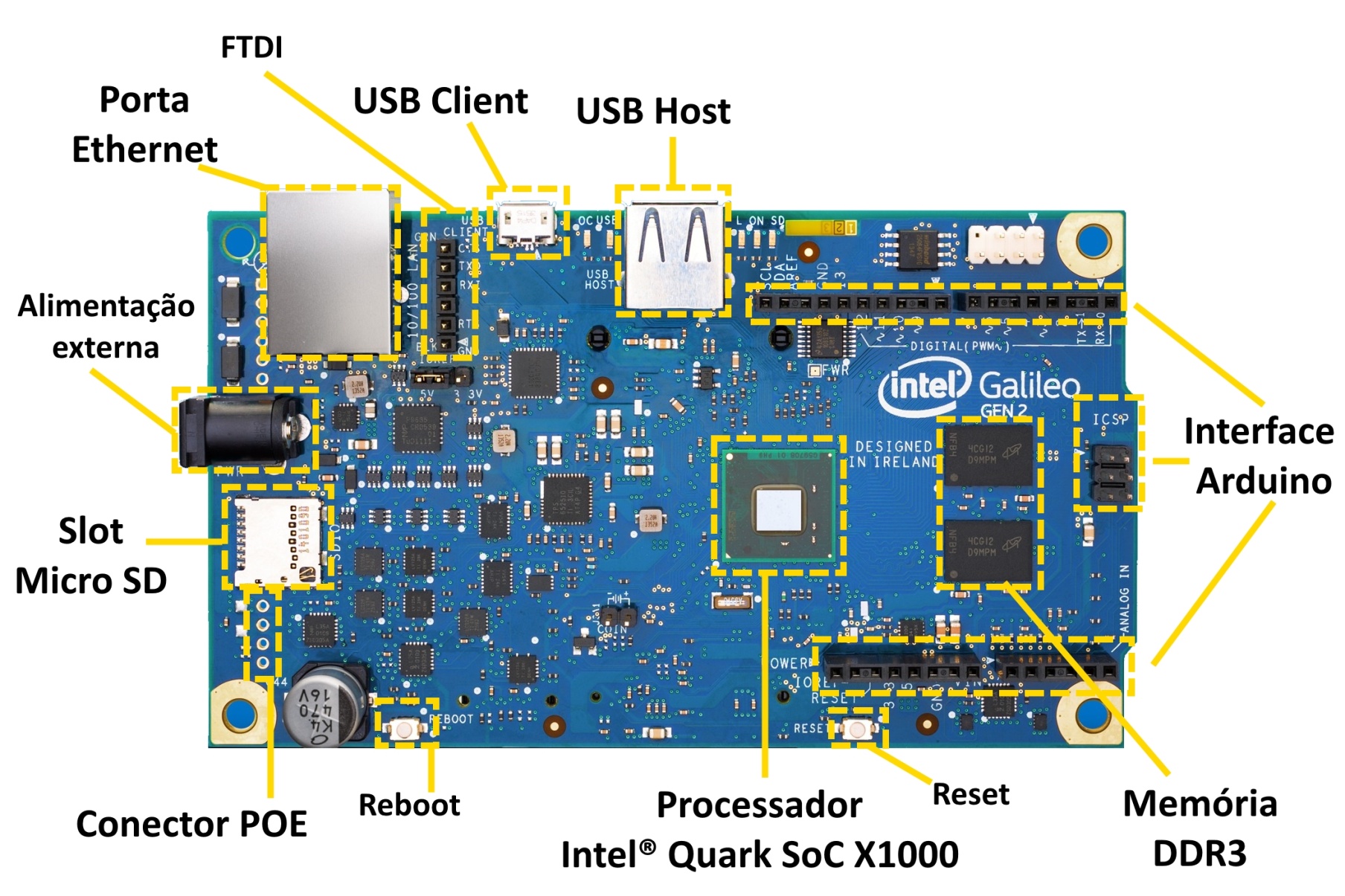
Assim o usuário pode, com a programação adequada, instalar uma distribuição Linux , Windows ou até mesmo o Mac OS X e fazer com que o sistema operacional utilize as funções do Arduino para se comunicar com o mundo externo.

Apesar de todo o poder da Intel Galileo Gen 2, ela é indicada tanto para iniciantes como para usuários avançados e aqui vamos mostrar um pequeno tutorial de como dar os primeiros passos com a Intel Galileo Gen 2.

## ****Características da placa Intel Galileo Gen 2****

A **Intel Galileo Gen 2** é uma placa compatível com uma grande variedade de shiels e módulos utilizados com [Arduino Uno R3](https://www.filipeflop.com/produto/placa-uno-r3-cabo-usb-para-arduino/" \t "_blank). No lado direito da placa, temos a disposição dos conectores com as saídas digitais, analógicas, tensão, etc, iguais às do Arduino Uno. É ali que podemos encaixar shields, utilizar sensores e outros componentes compatíveis.

Temos também uma porta Ethernet, portas USB Client e USB Host, slot para cartão microSD, conector de força e conector FTDI. O botão **Reboot** reinicia a placa inteira e o botão **Reset** reinicia apenas o programa do Arduino que estiver rodando no momento.



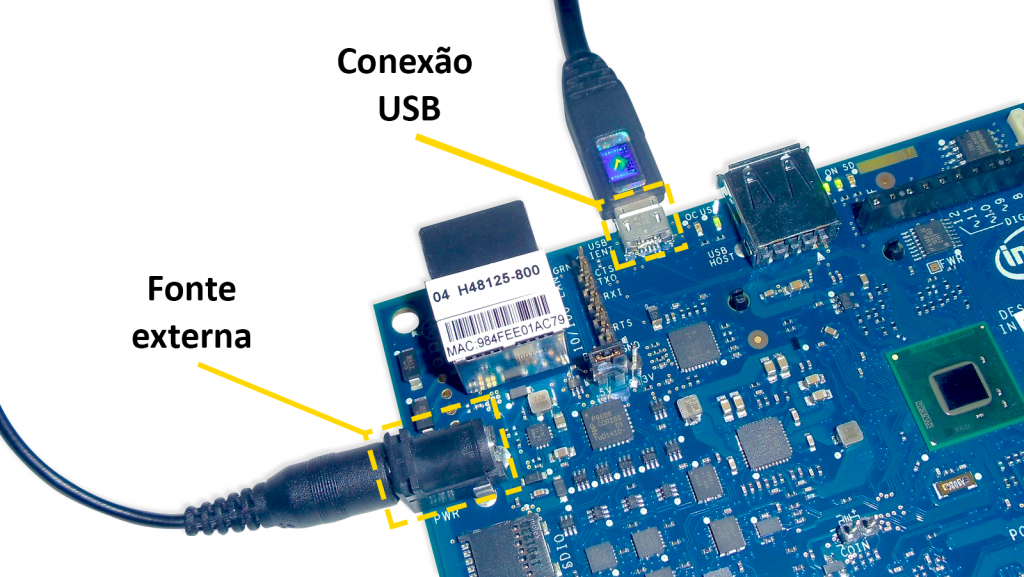
A placa também pode ser alimentada por meio do cabo de rede (Ethernet), utilizando a tecnologia **POE** (**P**ower **O**ver **E**thernet), mas esse recurso requer um hardware adicional.

Principais especificações técnicas:

* Processador Intel® Quark SoC X1000 de 400 Mhz (32 bits – núcleo único)
* 512 KB de SRAM
* Memória DRAM de 256 MB, DDR3
* Bootloader : 8 MB
* Porta Ethernet 10/100
* EEPROM : 8 KB
* SD Card : até 32 GB
* Compatível com USB 2.0
* Alimentação : 7 à 15 V

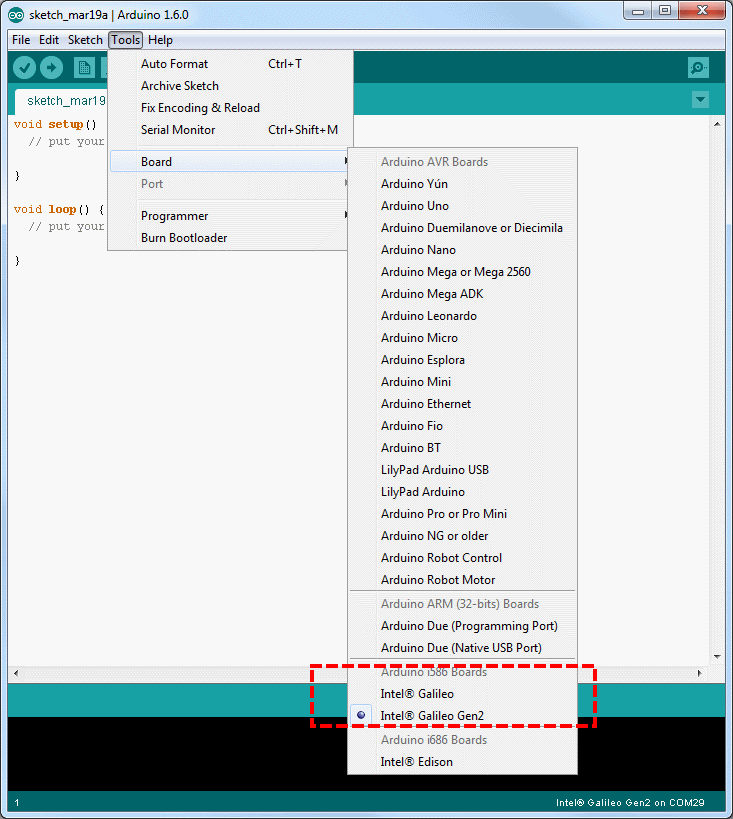
## ****Alimentação e conexão da placa Galileo****

A **Intel Galileo Gen 2** vem com uma fonte de alimentação que fornece 12V e 1,5 A. A conexão ao computador é feita por meio de um cabo USB com conector mini-usb , que deve ser ligado ao conector USB-Client da Galileo.

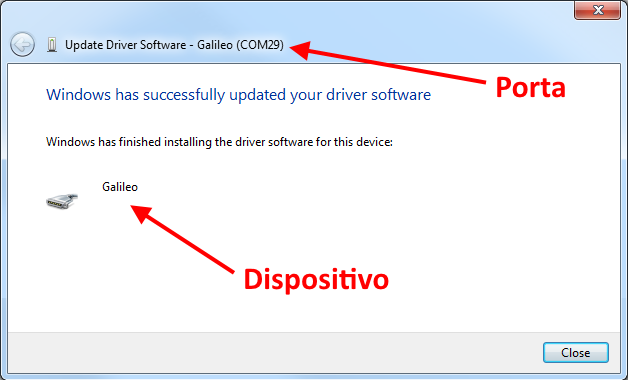
[](https://uploads.filipeflop.com/2015/03/Galileo_gen2.png)

Para enviar os programas para a Galileo, usamos uma IDE modificada, que pode ser baixada [nesse link](http://www.intel.com/support/galileo/sb/CS-035101.htm). Na página de downloads, escolha a versão do sistema operacional do seu computador abaixo de “**Arduino Software 1.6.0 – Intel 1.0.4″.**

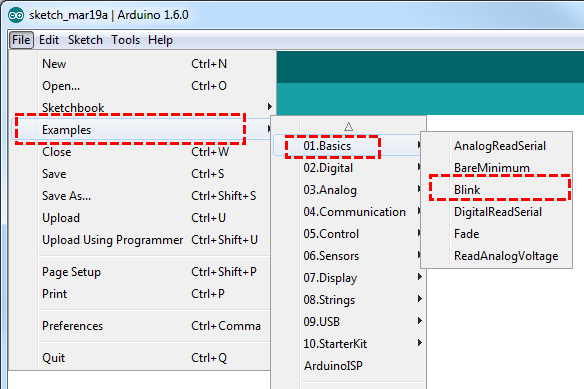
Após o download, você terá o mesmo ambiente de programação que já usa para as placas Arduino, mas agora com a possibilidade de escolher também a placa Intel Galileo:



Nessa pasta que contém a IDE modificada você também vai encontrar os drivers (em <caminho>arduino-1.6.0+Inteldrivers) para instalação da placa no Windows, que será identificada no painel de controle e no gerenciador de dispositivos como **Galileo**, e será relacionada à uma porta serial (COM), conforme destacado abaixo:



Finalizada a instalação dos drivers, faça um teste básico de funcionamento carregando o exemplo BLINK, no menu **Arquivo => Exemplos => Basics => Blink**



Esse é um dos programas mais comuns do Arduino e faz com que o led ligado à porta 13 do Arduino comece a piscar em intervalos de 1 segundo. Na Intel Galileo Gen 2, esse led fica ao lado do conector USB Host.

[](https://uploads.filipeflop.com/2015/03/Galileo_gen2_LED.png)

No próximo artigo falaremos um pouco mais sobre a utilização da Intel Galileo Gen 2.

 Gostou? Deixe seu comentário logo abaixo. Em caso de dúvidas, caso queira trocar uma ideia, ou até mesmo dividir seu projeto, acesse nosso [Fórum](https://www.filipeflop.com/forum/)!

[**Intel apresenta o Galileo, a sua versão do Arduino com o minúsculo chip Quark**](https://tecnoblog.net/141926/intel-galileo/)

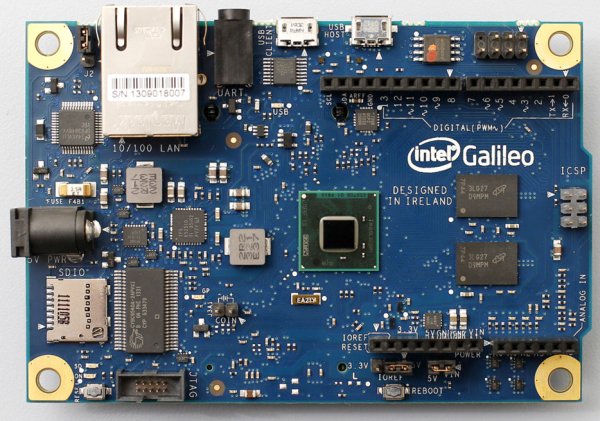
https://secure.gravatar.com/avatar/97ff7e9d810a219879764225596c38c9?s=30&d=mm&r=gPor [Emerson Alecrim](https://tecnoblog.net/author/ejas/)  
5 anos atrás

[NEWS](https://tecnoblog.net/categoria/news)

Já conhece a nova extensão do **Tecnoblog**? [**Baixe Agora**](https://tecnoblog.net/141926/intel-galileo/)

A [**Intel**](https://tecnoblog.net/sobre/intel/) não resistiu à onda de projetos baseados em Arduino e fechou uma parceria para tentar emplacar seu nome no segmento. O resultado foi apresentado recentemente no evento Maker Faire 2013: as placas Intel Galileo.

Em sua essência, estas são placas de desenvolvimento compatíveis com Arduino, com a diferença de serem baseadas na arquitetura da Intel. Neste sentido, chama atenção o fato de o processador que compõe o dispositivo ser um [Quark X1000](https://tecnoblog.net/139964/quark-menor-chip-soc-intel/) single-core de 400 MHz, cache L1 de 16 KB e 512 KB de SRAM. Lembra daquele chip apresentado pela Intel na IDF 2013 tido como o seu menor SoC até agora? É ele.



Medindo 10,6 cm por 7,1 cmm, este é o Intel Galileo

A Galileo não seria uma placa programável se não houvesse múltiplas interfaces de entrada e saída. Entre elas há portas e slots PCI Express, Ethernet 10/100, serial RS-232, USB 2.0, EHCI/OHCI USB e JTAG (para depuração de código, testes e afins). Há ainda um chip flash NOR programável de 8 MB, 256 MB de RAM e, opcionalmente, leitor de microSD.

PUBLICIDADE

[inRead invented by Teads](http://inread-experience.teads.tv/)

Sendo compatível com Arduino, será possível utilizar softwares, shields (“placas de expansão”), acessórios e afins destas placas nos dispositivos Galileo. A Intel garante também suporte a desenvolvimento por meio de [**Windows**](https://tecnoblog.net/sobre/windows/), OS X e [**Linux**](https://tecnoblog.net/sobre/linux/).

As placas Galileo serão comercializadas a partir do próximo mês de novembro, com preços ainda não revelados. A Intel promete ainda distribuir 50 mil unidades do dispositivo a mil universidades do mundo todo nos próximos 18 meses. Como placas do tipo são amplamente usadas em projetos educacionais, esta é uma bela tentativa de popularizar a novidade.

# Kiki ou bouba?': o fenômeno por trás da pergunta a que todos respondemos da mesma forma

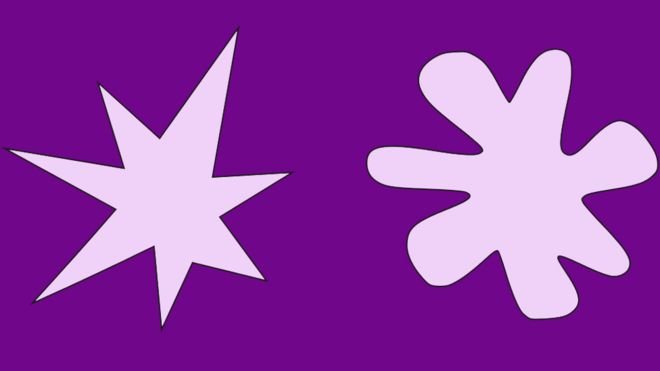
* 23 abril 2017
* [Compartilhe este post com Facebook](https://www.bbc.com/portuguese/geral-39685606)

* [Compartilhe este post com Messenger](https://www.bbc.com/portuguese/geral-39685606)

* [Compartilhe este post com Twitter](https://www.bbc.com/portuguese/geral-39685606)

* [Compartilhe este post com Email](mailto:?subject=Compartilhado%20pela%20BBC%20Brasil&body=https%3A%2F%2Fwww.bbc.com%2Fportuguese%2Fgeral-39685606)

* [Compartilhar](https://www.bbc.com/portuguese/geral-39685606#share-tools)

Image captionA sinestesia ajuda a explicar por que a maioria das pessoas nomeia estas figuras do mesmo jeito

**Olhe as figuras acima e diga: qual se chama "bouba" e qual se chama "kiki"?**

Assim como quase todas as pessoas que ouvem essa pergunta, você provavelmente parece saber instintivamente a resposta: a pontiaguda é "kiki" e a curvilínea, "bouba".

Não se trata de um truque nem uma mera curiosidade, mas de um fenômeno que parece se manifestar em diferentes culturas, mesmo aquelas sem escrita, e entre pessoas de todas as idades.

Tudo começou em Tenerife, uma ilha da Espanha, quando o psicólogo alemão Wolfgang Kohler começou a fazer experimentos apresentando às pessoas duas formas, uma com pontas e outra com curvas, e pedindo que elas as nomeassem como "takete" ou "baluba".

Direito de imagemISTOCKImage captionFenômeno foi testado por psicólogo alemão em ilha espanhola

A maioria disse que a figura pontiaguda era "takete" e a curvilínea, "baluba". O resultado desse trabalho foi publicado em 1929.

Não chamou muita atenção nem foi levado a sério pela comunidade científica, permanecendo como uma mera curiosidade até 2001, quando dois pesquisadores americanos publicaram pesquisas sobre algo conhecido como sinestesia.

Vilayanur S. Ramachandran e Edward Hubbard, da Universidade da Califórnia, nos Estados Unidos, se dedicaram a estudar um fenômeno que faz as pessoas misturarem experiências sensoriais, inclusive sons e imagens.

Acredita-se que ele se manifeste em uma a cada 20 pessoas, mas Ramachandran suspeitava que as conexões sensoriais cruzadas eram uma característica do cérebro humano.

E o melhor meio de testar a hipótese era o experimento feito quase oito décadas antes pelo psicólogo alemão. O teste de Kohler mostrava como, por meio dos sons e formas apresentados por ele, as pessoas vinculavam sentidos diferentes: visão e audição.

Usando figuras similares, mas alterando ligeiramente seus nomes, os dois cientistas as apresentaram a um grupo que incluia estudantes americanos e falantes do idioma tamil: um índice assombroso de 95% dos participantes identificaram o desenho com pontas como "kiki" e o curvo como "bouba".

Direito de imagemALAMYImage captionDiálogo em 'Alice Através do Espelho' explora a natureza das palavras

*— Meu nome é Alice, mas...*

*— Que nome mais estúpido! - Humpty Dumpty a interrompeu impacientemente. —O que significa?*

*— Por acaso, um nome precisa significar alguma coisa? — perguntou Alice timidamente.*

*— Claro que sim! — disse Humpty Dumpty com um risinho — Meu nome significa exatamente a forma que tenho, uma forma, com certeza, muito bonita. Você por sua vez, com um nome assim, poderia ter qualquer forma.*

\* Diálogo de *Alice Através do Espelho e O Que Ela Encontrou Por Lá* (1871), de Lewis Carroll.

Desde que foi confirmado o resultado de Kohler, várias áreas científicas têm explorado o efeito bouba/kiki e realizado mais estudos para entender suas implicações.

Será esse o elo perdido da linguagem?, se perguntam linguistas, destacando que talvez haja uma ligação entre os nomes de objetos e os sons e as formas que fazemos com os lábios para pronunciar uma palavra. Talvez tenha sido assim que nasceram os idiomas?

Ainda que, no início do século 20, o linguista suíço Ferdinand de Saussure tenha apresentado um sólido argumento que convenceu a maioria de seus pares da arbitrariedade da linguagem, o efeito bouba/kiki fez renascer a ideia de que o som de uma palavra pode estar vinculado a algum tipo de significado inerente.

Outros cientistas notaram que o efeito bouba/kiki não se manifesta apenas com a visão e a audição, mas também com outros sentidos, tais como paladar, olfato e tato.

Pense, por exemplo: entre uma água com gás e uma sem, qual é "bouba" e qual é "kiki"? E entre o chocolate ao leite e o amargo?

Direito de imagemISTOCKImage captionQual é 'kiki' e qual é 'bouba'?

A ciência indica que nossos sentidos estão interconectados. Nossos cérebros formam víncuclos entre formas, sons e sabores,

Estudos como o de Merle Fairhust e seus colegas na Universidade de Londres, no Reino Unido, revelaram que a comida nos parece mais doce quando servida em um prato redondo em comparação com um quadrado ou triangular.

Outros experimentos mostraram que os sabores picantes - como o do gengibre e do wasabi - são percebidos de forma mais intensa se servidos em recipientes "kiki" (ásperos e pontiagudos) enquanto a comida parece ficar mais cremosa em recipientes "bouba" (suaves e redondos).

## Um mundo por descobrir

Como explicar que possamos usar sons e formas sem significado para cruzar as fronteiras dos sentidos? Ainda não sabemos exatamente, mas estamos cientes de que isso ocorre.

"Temos todo um mundo por descobrir: se você escreve 'vinho' com uma letra pontiaguda e, depois, com uma arredondada e toma o mesmo vinho, você sente gostos distintos", disse à BBC o chef Heston Blumental, que tem testado esse universo.

"Se você bebe um gole de vinho levantando a taça com a mão direita, o sabor é diferente de quando você faz isso com a esquerda: com uma delas, será mais adstringente. Se coloco uma música, isso também muda o sabor."

Quem já teve a chance de desfrutar de um banquete de Blumenthal pôde viver essa aventura sensorial. Poucos exploram nossas percepções como ele.

O chef usa o efeito bouba/kiki para intensificar sabores e supreender, como quando apresentou pratos que se parecem com doces, mas têm gosto salgado.

Direito de imagemISTOCKImage captionO chef Blumenthal faz brincadeiras com nossos sentidos

No entanto, não é preciso pagar muito dinheiro para descobrir esse mundo. Na verdade, fazemos isso o tempo todo. Falamos de cheiros nos referindo a ela como sons - o de menta seria de alta frequência enquanto o almíscar seria de baixa frequência. Ou quando dizemos que cores são ácidas, frias ou quentes.

"O que realmente é excitante é que isso revela que nossos sentidos estão sempre ativos e interagindo entre si sutilmente, e essas conexões moldam nossa percepção do mundo e de nós mesmos", destaca o filósofo Barry Smith.

Agora, antes que você vá embora, podemos te fazer mais uma pergunta? Se tivesse de classificar limões e bananas em uma escala de velocidade, qual seria mais rápido?

A maioria das pessoas diria que os limões são velozes, e as bananas, lentas. Curioso, não?

Direito de imagemISTOCKImage captionQual é mais veloz: a banana