Componentização e Reúso de Software

ComponenteVerso

André Santanchè
Laboratory of Information Systems – LIS
Instituto de Computação – UNICAMP

Agosto de 2019

Picture by Doug Wheller [http://www.flickr.com/photos/doug88888/5817711538/]

Componentes

“Aquilo que entra na composição de alguma coisa.” (Aurélio, 2004)

“que ou o que compõe ou ajuda na composição de algo” (Houaiss, 2006)

Origens

Engenharia de Software

- The NATO Software Engineering Conferences

  (http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF)

  - “Crise de Software”
  - Nascimento da Engenharia de Software
  - McIlroy “Mass Produced Software Components” (Mcilroy, 1968)

Mass Produced Software Components

“Sem dúvida nós produzimos software usando técnicas ultrapassadas. Sem dúvida nós ficamos com o lado curto do palitinho em confrontos com as pessoas de hardware porque eles são os industriais e nós somos os lavradores.” (Mcilroy, 1968)

Tradução do original feita pelo autor: “We undoubtedly produce software by backward techniques. We undoubtedly get the short end of the stick in confrontations with hardware people because they are the industrialists and we are the crofters.” (Mcilroy, 1968)

Mass Produced Software Components

- Analogia com técnicas industriais relevantes:
  - Subassemblies
  - partes intercambiáveis – modularidade
  - máquinas (machine tools) – compiladores e montadores

Componente em UML

- "[...] sistemas de software de tamanho e complexidade arbitrários."[1]
- Componente[2]:
  - unidade modular
  - com interfaces bem definidas
  - substituível dentro do ambiente

[1] "[...] software systems of arbitrary size and complexity" (Cook, 2015)

[2] "[...] Component as a modular unit with well-defined Interfaces that is replaceable within its environment." (Cook, 2015)

Componentes x Composição

“components are for composition” (Szyperski, 2002)

Composição

“Composition enables prefabricated 'things' to be reused by rearranging them in ever-new compositities”. (Szyperski, 2002)

O que é um componente?

"Today, few terms in the software industry are less precise than component software." (Olsen, 2006)

O que é um componente?

“[...] software components are executable units of independent production, acquisition, and deployment that can be composed into a functioning system.” (Szyperski, 2002)

“The characteristic properties of a component are that it:

is a unit of independent deployment;

is a unit of third-party composition;

has no (externally) observable state.” (Szyperski, 2002)

O que é um componente?

“A Component is:

an opaque implementation of functionality

subject to third-party composition

conformant with a component model”

(Bachmann, 2000)

O que é um componente?

“A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to composition by third parties.” (Workshop on Component-Oriented Programming, ECOOP, 1996)

O que é um componente?
Encapsulamento

“A component is simply a **data capsule**. Thus **information hiding** becomes the core construction principle underlying components.”
Wolfgang Pree & Gustav Pomberger in (Broy, 1998)

O que é um componente?
Sub-componentes

"A component may **consist of other components**." Anton Deimel in (Broy, 1998)

O que é um componente?

Interfaces

“A software component is a unit of composition with contractually specified **interfaces** and explicit context dependencies only.”

(Workshop on Component-Oriented Programming, ECOOP, 1996)

“5. A component uses precisely-defined **interfaces** to communicate with other components.”

Anton Deimel in (Broy, 1998)

O que é um componente?
Interfaces

"A component is a system-independent binary entity which implements **one or more interfaces**. An interface is a collection of signatures of services belonging logically together."

Kai Koskimies in (Broy, 1998)

O que é um componente?
Interfaces

“The component (module) **interface** is described either

textually by means of an **interface description language (IDL)** or

visually / interactively using appropriate tools.”

Wolfgang Pree & Gustav Pomberger in (Broy, 1998)

O que é um componente?
Características Comuns

- Publica sua funcionalidade através de uma interface
  - interface guia relacionamento componente x ambiente
- Entidade concebida para ser composta
  - do latim *componens*, derivado de *componere*, que quer dizer “colocar junto”.
- Componentes podem ser aninhados em outros componentes
  - componentes e sub-componentes

O que é um componente?

Características Desejáveis

- Contém código binário que implementa a funcionalidade declarada na interface
- Serviços acessíveis exclusivamente pela interface (black-box)
- Pacote padrão para distribuição

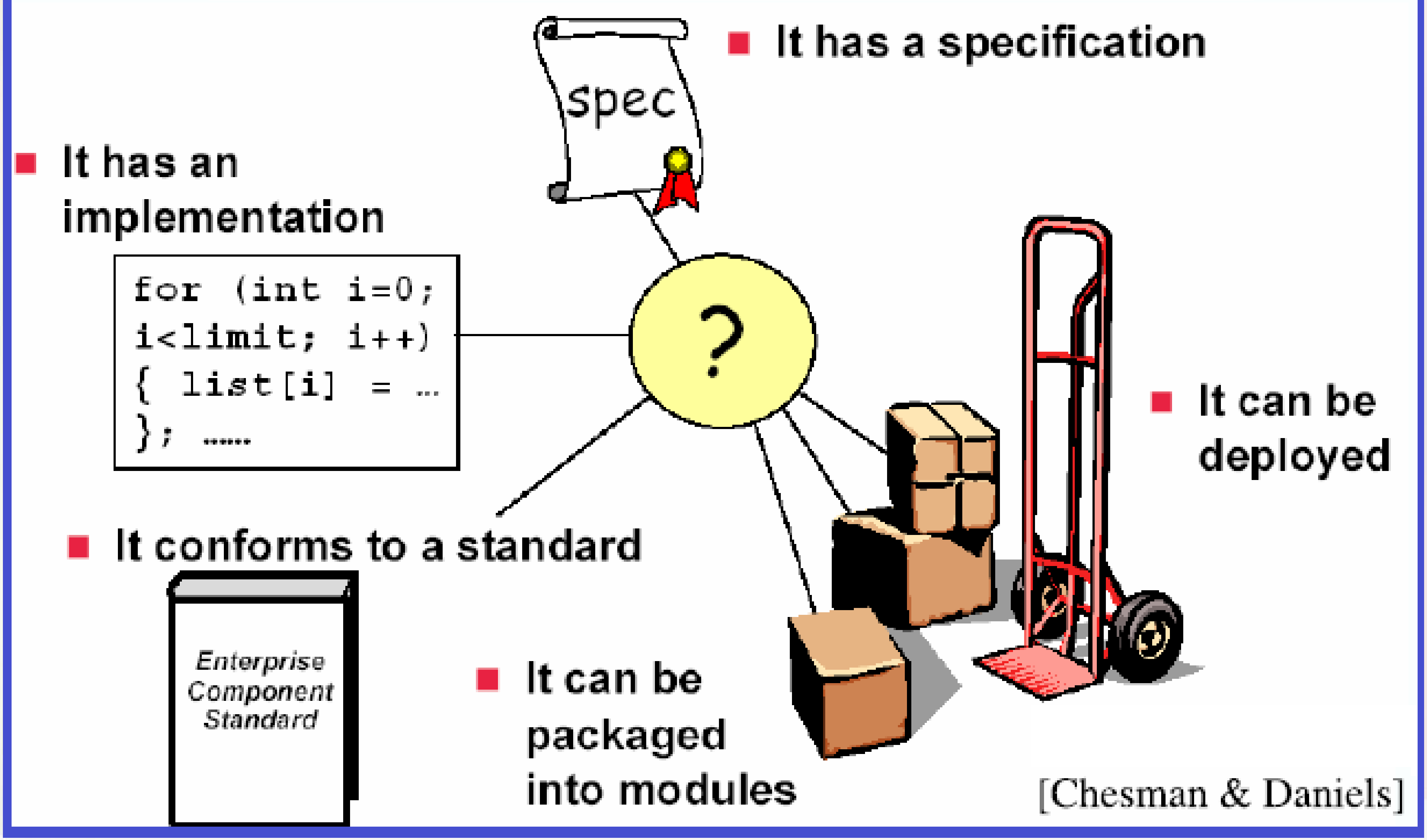
O que é um componente?

Características de Design

- Intercambiável dentro do ambiente
- Alta Coesão e Baixo Acoplamento

O que é um componente?

(Cheesman & Daniels, 2000)



- It has a specification

- It has an implementation

```
for (int i=0;
i<limit; i++)
{ list[i] = ...
}; ......
```

- It conforms to a standard

- It can be deployed

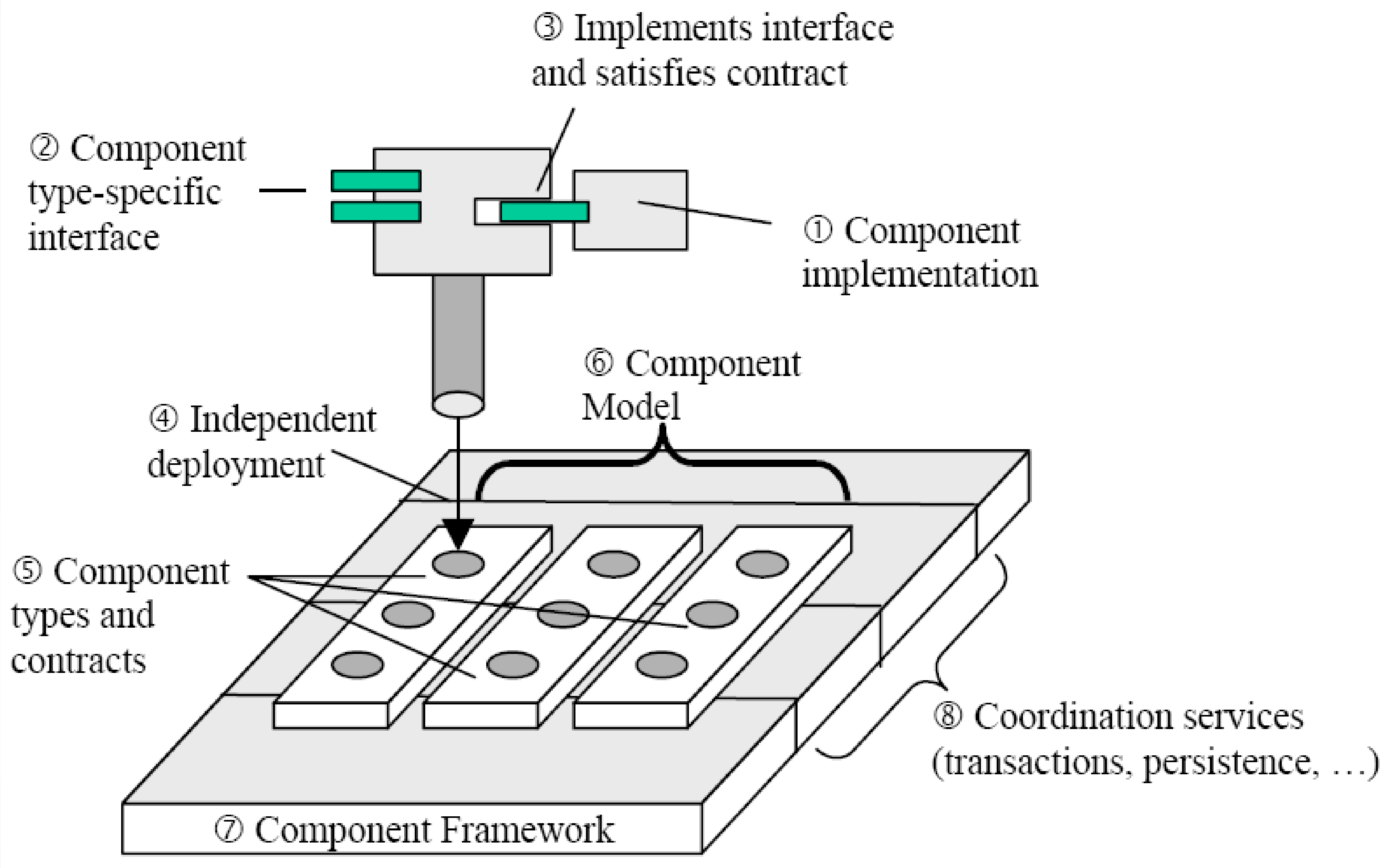
- It can be packaged into modules

[Chesman & Daniels]

Analisando Modelos de Componentes

- Especificação / Padrão
- Implementação
- Empacotamento / Distribuição

The Component-Based Design Pattern


③ Implements interface and satisfies contract
② Component type-specific interface
① Component implementation
⑥ Component Model
④ Independent deployment
⑤ Component types and contracts
⑧ Coordination services (transactions, persistence, …)
⑦ Component Framework

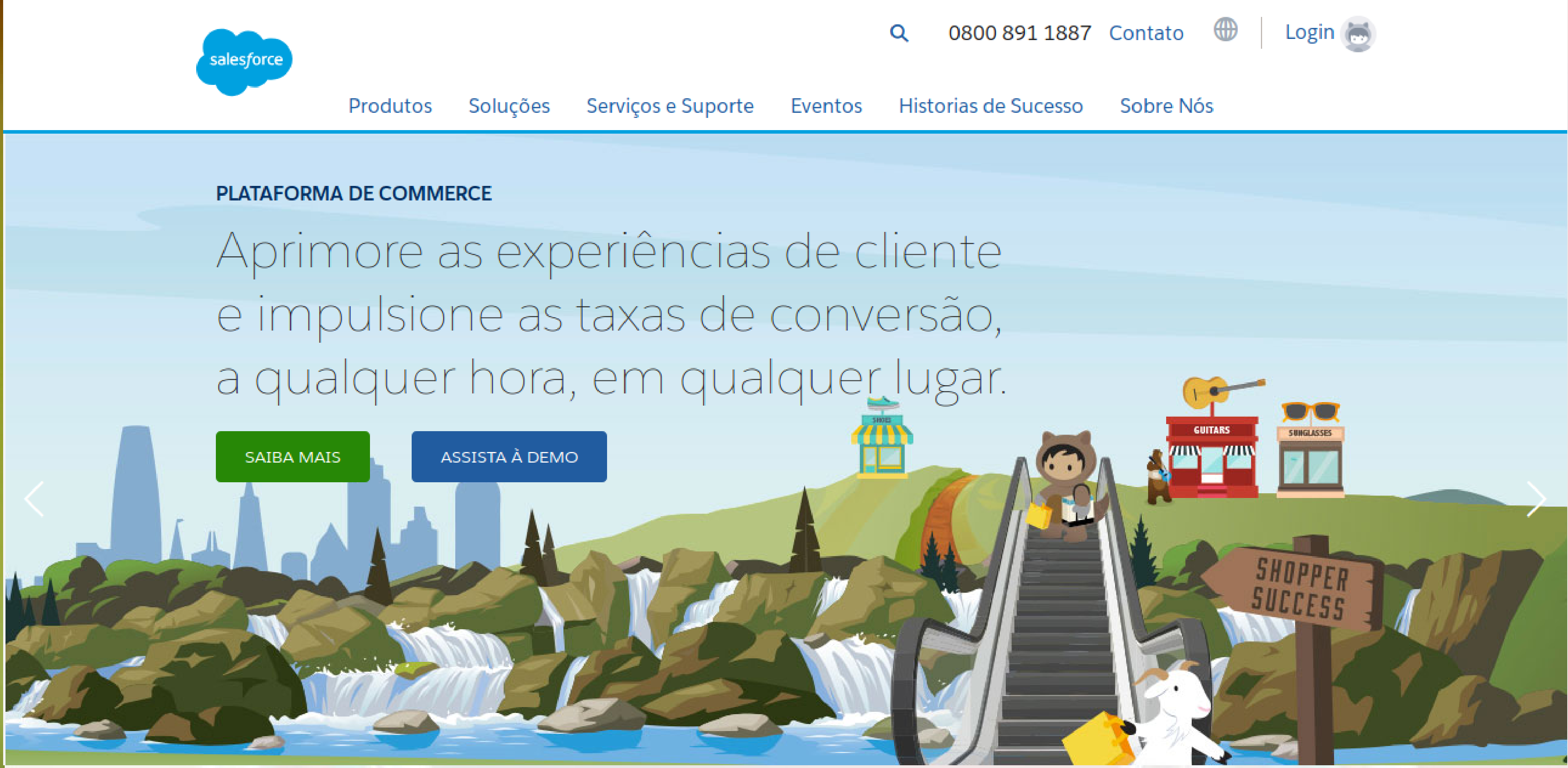
(Bachmann, 2000)

ComponenteVerso

Cenário 1
Salesforce

Salesforce

https://www.salesforce.com



PLATAFORMA DE COMMERCE

Aprimore as experiências de cliente e impulsione as taxas de conversão, a qualquer hora, em qualquer lugar.

Salesforce

- Plataforma de CRM na nuvem
- Customer Relationship Management (CRM) - Gestão de Relacionamento com o Cliente

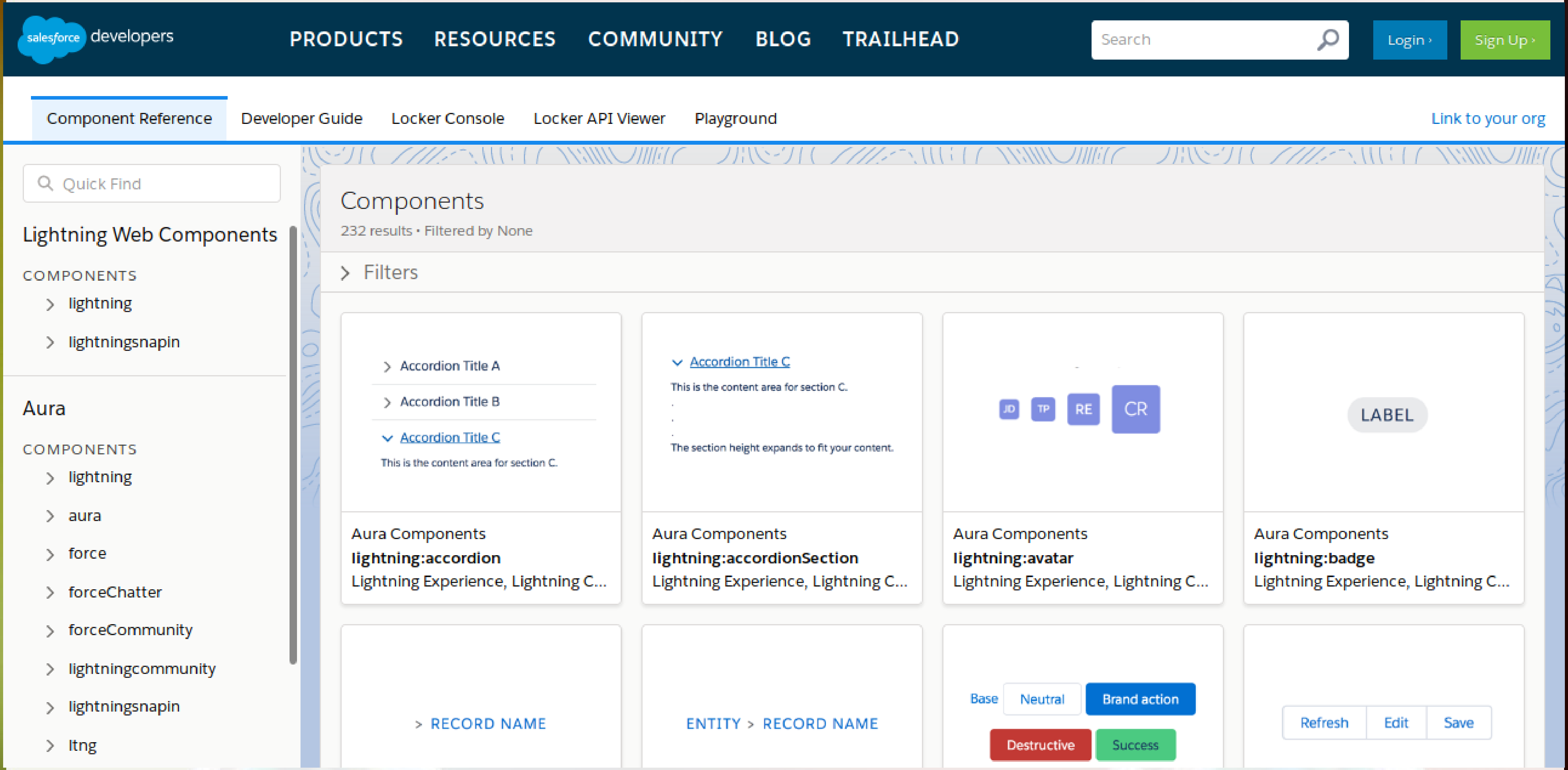
Lightning Components

https://lightningdesignsystem.com

- “Components are the building blocks of Salesforce applications, enabling designers and developers with ready-to-go interface elements available in HTML and CSS code, or in Sketch Templates.”

Lighting Component Reference

https://developer.salesforce.com/docs/component-library/overview/components



Lightning Web Components

COMPONENTS
- lightning
- lightningsnapin

Aura

COMPONENTS
- lightning
- aura
- force
- forceChatter
- forceCommunity
- lightningcommunity
- lightningsnapin
- ltng

Components

232 results • Filtered by None

› Filters

Aura Components
lightning:accordion
Lightning Experience, Lightning C...

Aura Components
lightning:accordionSection
Lightning Experience, Lightning C...

Aura Components
lightning:avatar
Lightning Experience, Lightning C...

Aura Components
lightning:badge
Lightning Experience, Lightning C...

Lighting
Modelo

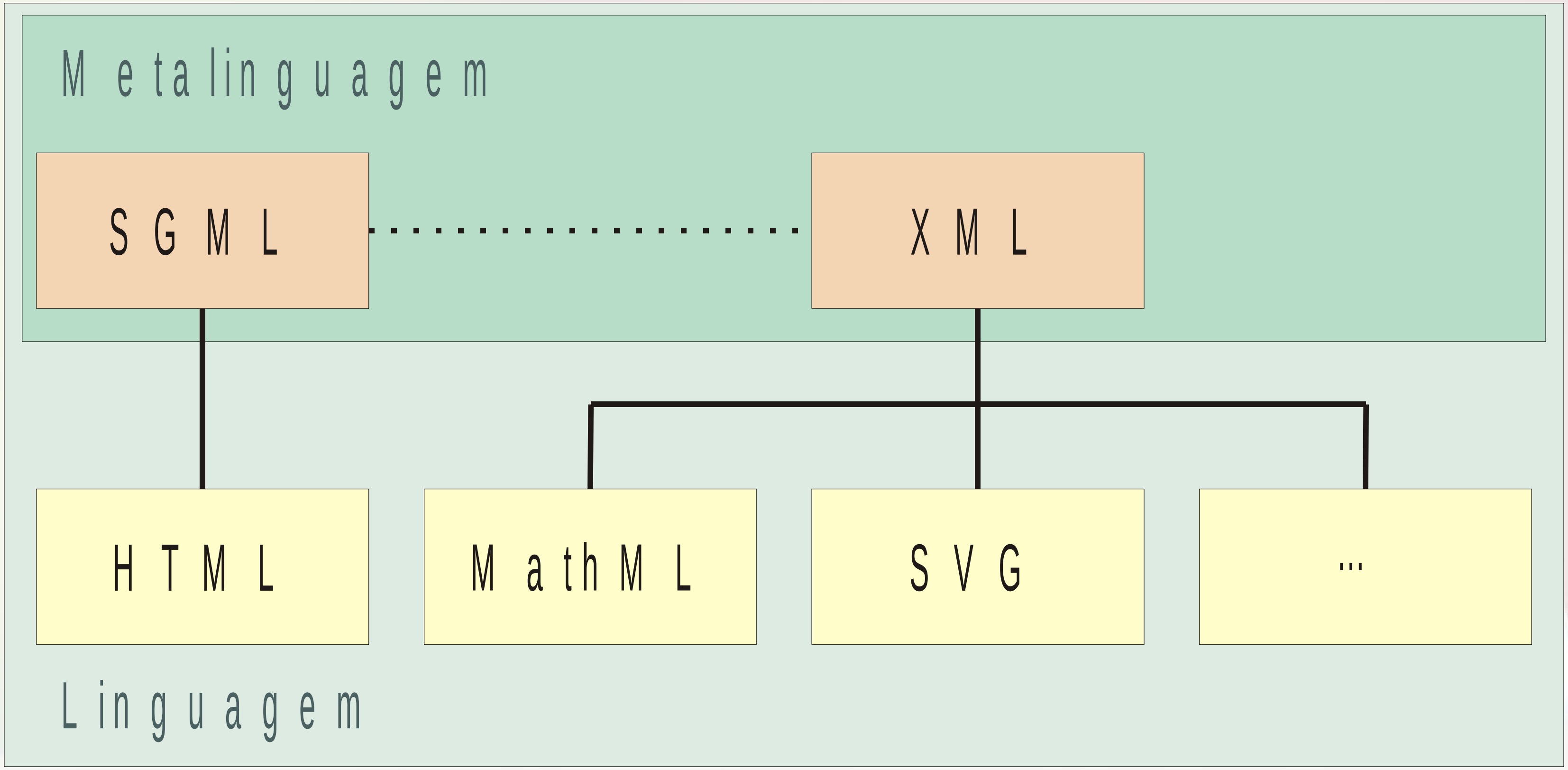
- Especificação / Padrão
  - Webcomponents Model
  - Lightning Web Components
- Implementação
  - Javascript / CSS / HTML
- Empacotamento / Distribuição
  - npm

Lighting Modelo

- Especificação / Padrão
  - Webcomponents Model
  - Lightning Web Components
- Implementação
  - Javascript / CSS / HTML
- Empacotamento / Distribuição
  - npm

HTML

- Hypertext Markup Language
- Publishing language
- Written in SGML



Metalinguagem

SGML

XML

HTML

MathML

SVG

...

Linguagem

Linguagem de Marcação

- Utiliza marcadores para agregar informações adicionais a documentos.
- Tomemos como exemplo a seguinte frase:

Horácio escreveu o livro Vida dos Dinossauros.

- Desejamos agregar informações que identifiquem quem é o **autor** e qual a **ação** realizada.

Linguagem de Marcação

- Os marcadores se diferenciam do conteúdo pelos símbolos “<” e “>” (seguem o mesmo princípio de HTML):

```
<autor>Horácio</autor> <ação>escreveu o livro Vida dos Dinossauros</ação>
```

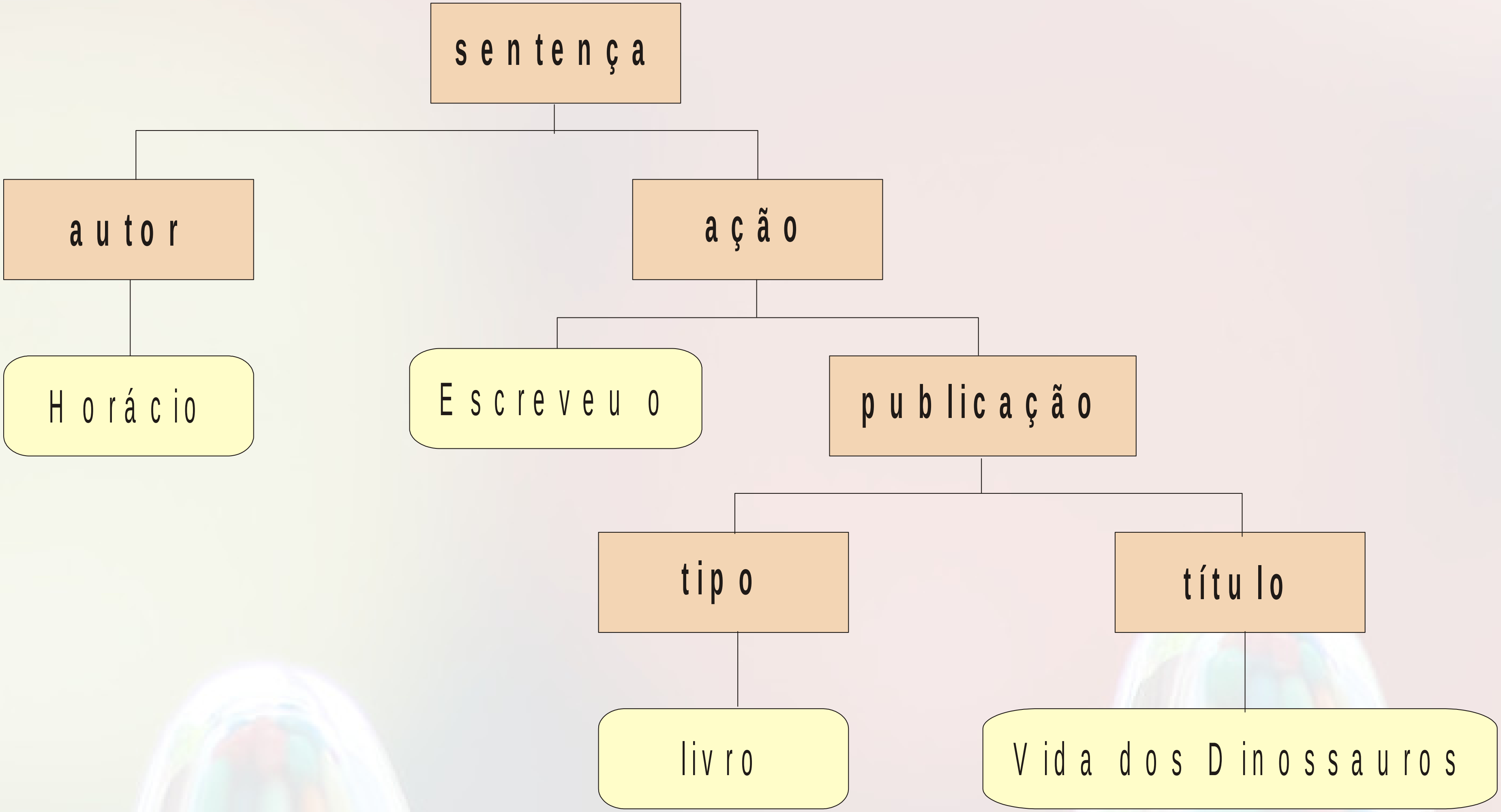
- Os marcadores delimitam unidades estruturais denominadas **elementos**.

Estrutura Hierárquica

- Marcações podem ser agrupadas hierarquicamente.
- Cada documento tem uma única raiz.
- A interpretação de cada marcador está subordinada a seu contexto.

```
<sentença>
  <autor>Horácio</autor>
  <ação>escreveu o
    <publicação>
      <tipo>livro</tipo>
      <título>Vida dos Dinossauros</título>
    </publicação>
  </ação>
</sentença>
```

Modelo de Dados XML



sentença

autor

ação

Horácio

Escreveu o

publicação

tipo

título

livro

Vida dos Dinossauros

Elemento Vazio

- Não contém outros elementos ou texto aninhados

```
<esgotado/>
```

Atributos

- Elementos podem conter atributos
- Não há dois atributos de mesmo nome no mesmo elemento

```
<autor cpf=”487.526.548-74” nascimento=”12/5/1960”> Horácio </autor>
```

Lighting
Modelo

- Especificação / Padrão
  - Webcomponents Model
  - Lightning Web Components
- Implementação
  - Javascript / CSS / HTML
- Empacotamento / Distribuição
  - npm

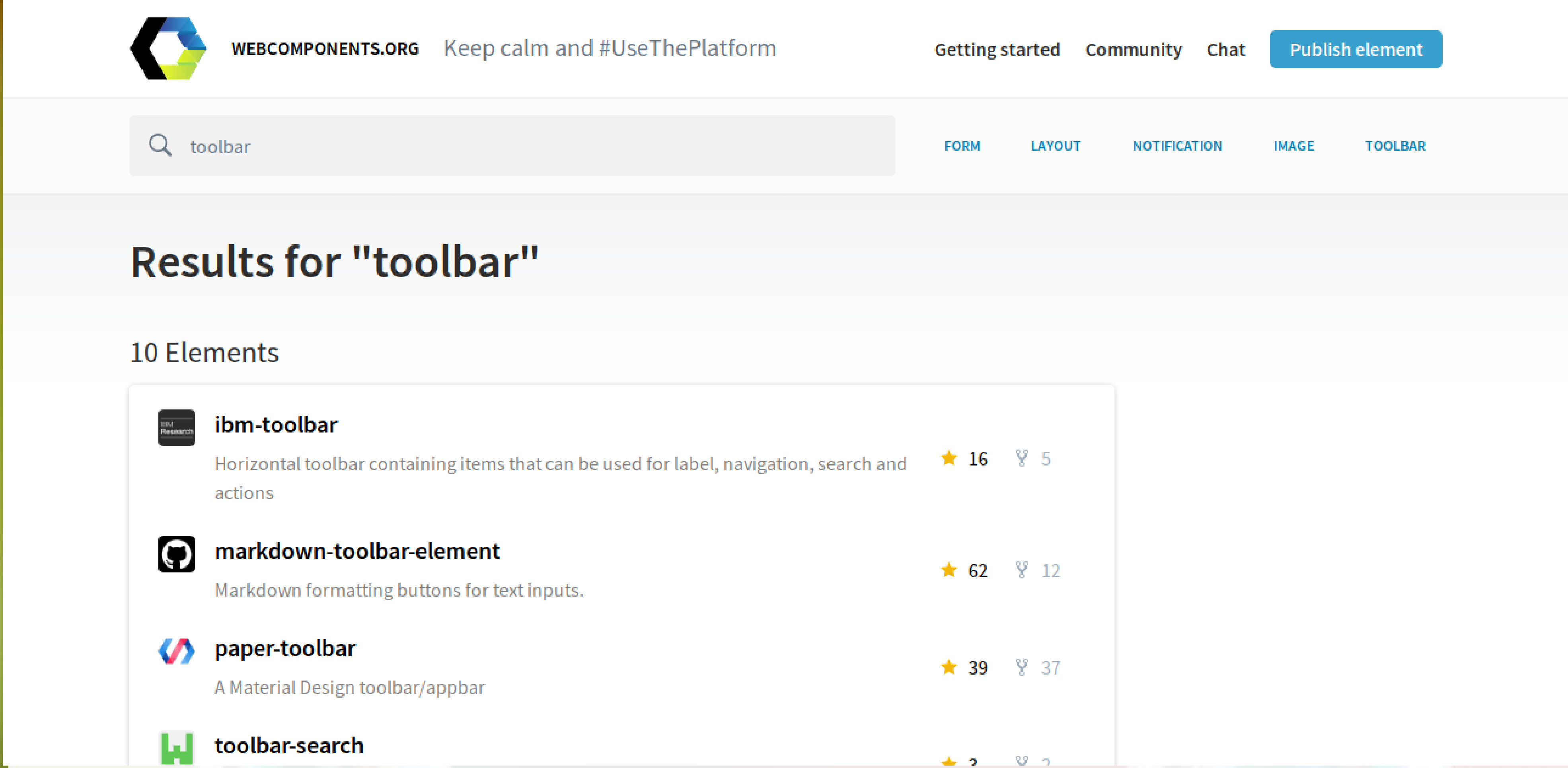
Web Components

Web Components

- Iniciativa do WebApps WG de criar um modelo de componentes para a Web (Cooney & Glazkov, 2013)
- Subdividido em:
    - Templates
    - Decorators
    - Custom Elements
    - Shadow DOM
    - Imports

WebComponents

https://www.webcomponents.org



WEBCOMPONENTS.ORG Keep calm and #UseThePlatform

Results for "toolbar"

10 Elements

ibm-toolbar
Horizontal toolbar containing items that can be used for label, navigation, search and actions
★ 16 ⑂ 5

markdown-toolbar-element
Markdown formatting buttons for text inputs.
★ 62 ⑂ 12

paper-toolbar
A Material Design toolbar/appbar
★ 39 ⑂ 37

toolbar-search

Google Polymer

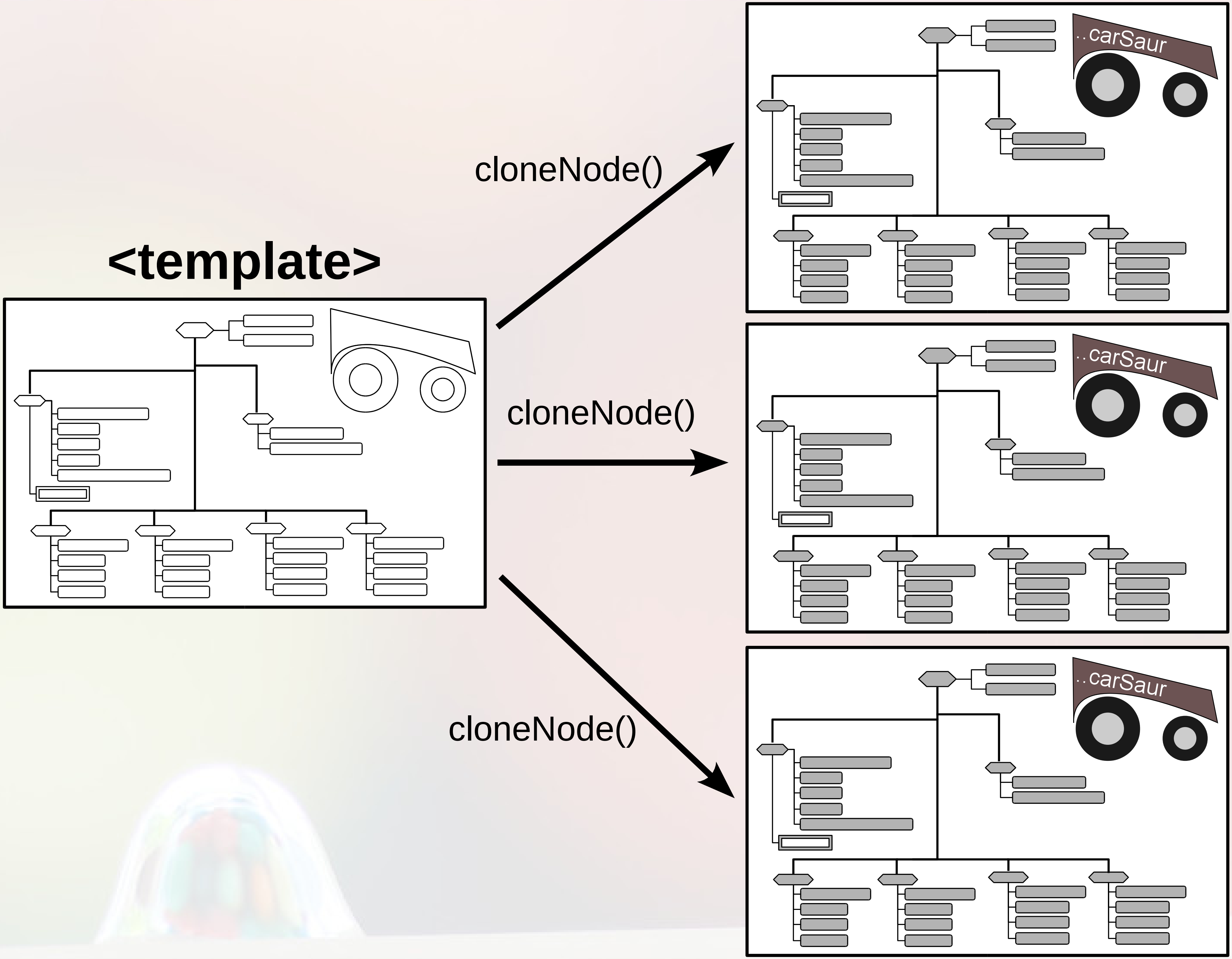
https://www.polymer-project.org

Google Polymer

http://www.polymer-project.org/

Template

<template>



cloneNode()

cloneNode()

cloneNode()

Template

```
<template id="imageSet">
  <style scoped>
    .tyre {
      fill: #1a1a1a;
    }
    .rim {
      fill: #cccccc;
    }
    .frame {
      fill: #6c5353;
      stroke: #000000;
      stroke-width: 1px;
    }
    .nameStyle {
      fill: white;
      font-size: 28px;
      font-family: Arial;
    }
  </style>
```

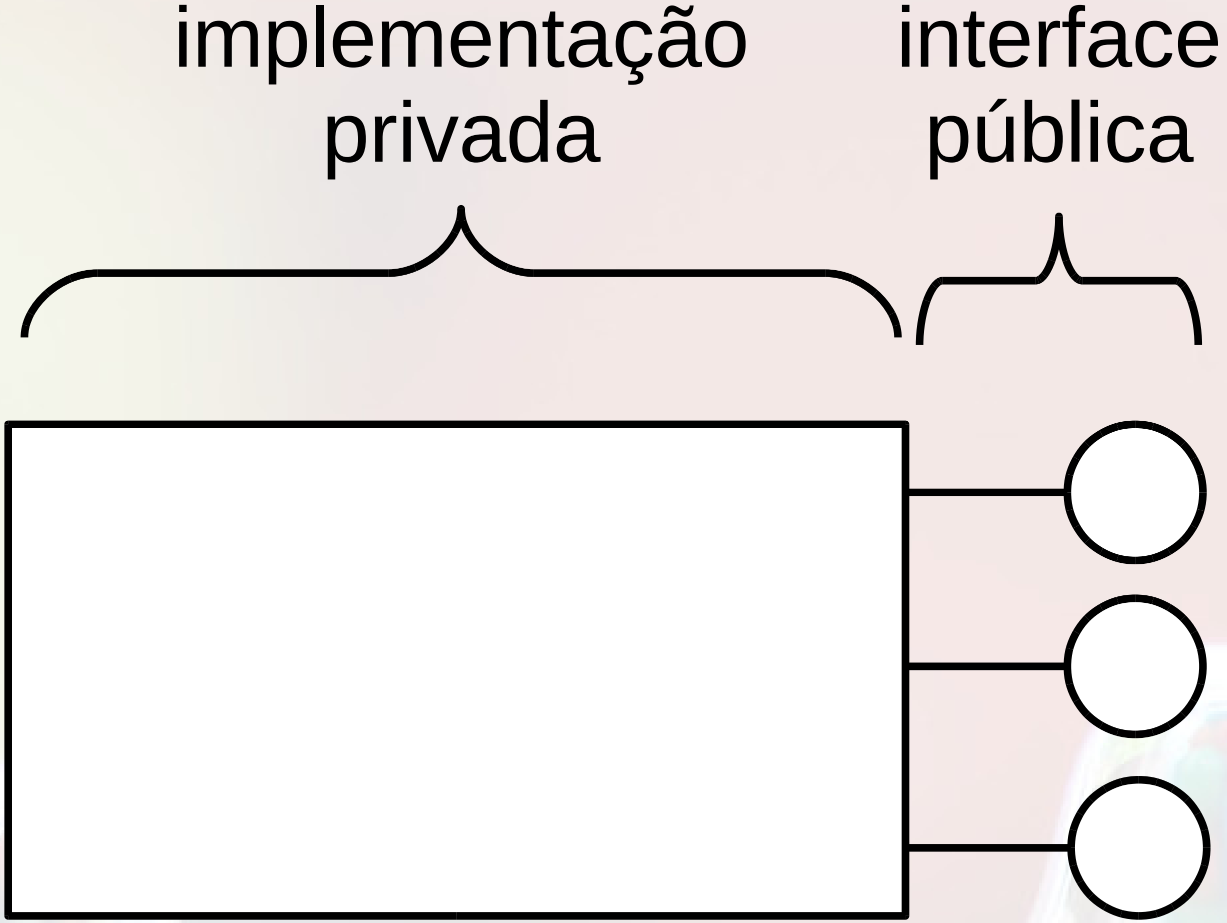
```
  <svg …>
    …
  </svg>

</template>
```

```
<div id="myCar">
  <!-- empty -->
</div>
```

```
var carComponent = document.querySelector("#carComponent");
var myCar = document.querySelector("#myCar");
myCar.appendChild(carComponent.content.cloneNode());
```

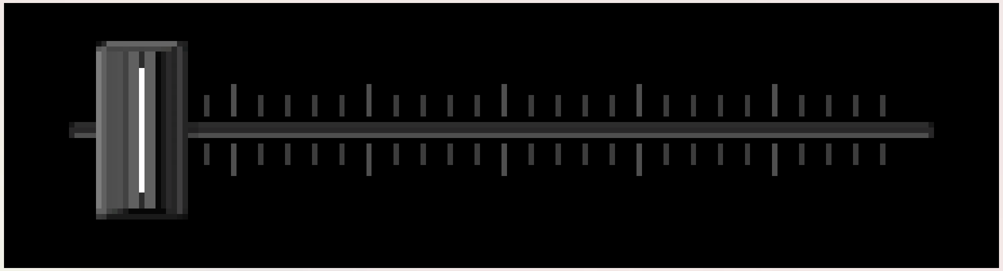
Shadow DOM

Modelo de Componente



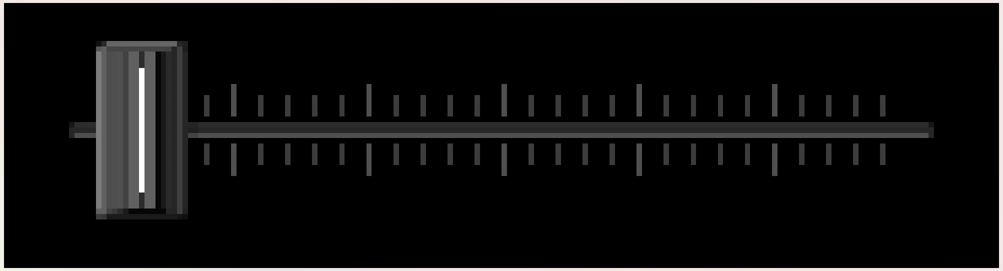
Shadow DOM

Interface

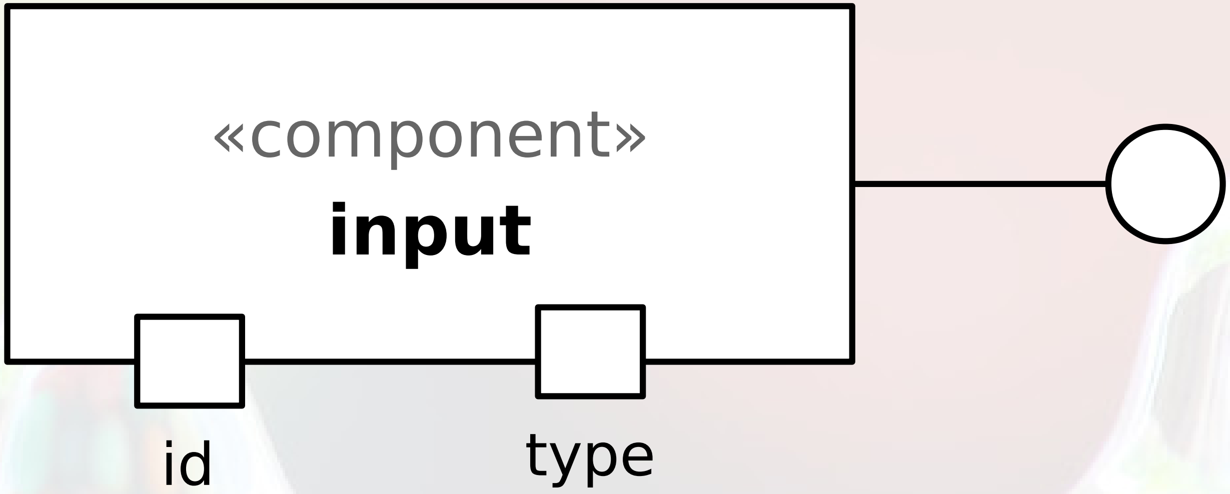


```
<input id="volume" type="range"/>
```

Atributos como Propriedades

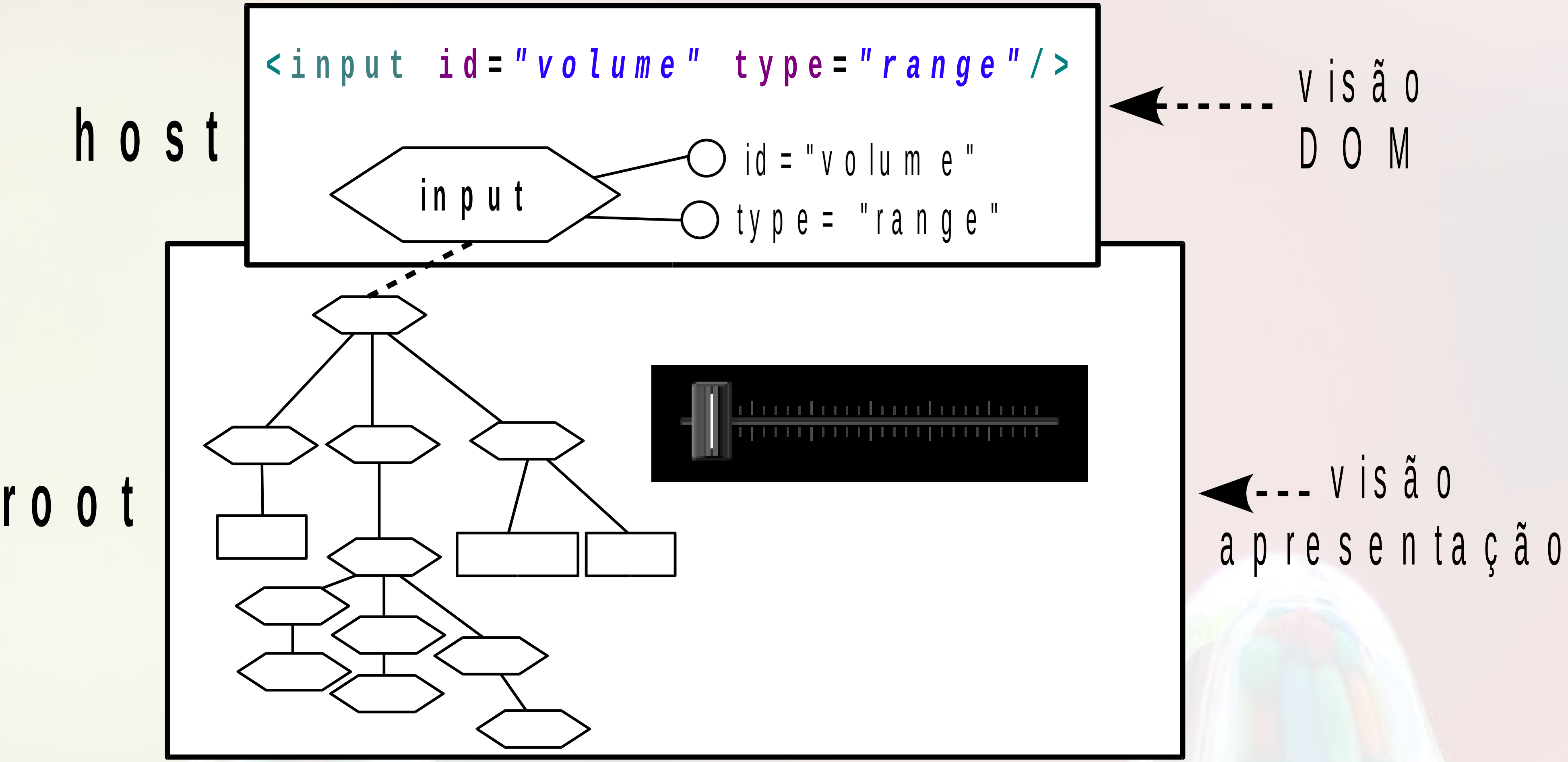


```
<input id="volume" type="range">
```



Shadow DOM

Implementação



```
<input id="volume" type="range"/>
```

host

input

id="volume"

type="range"

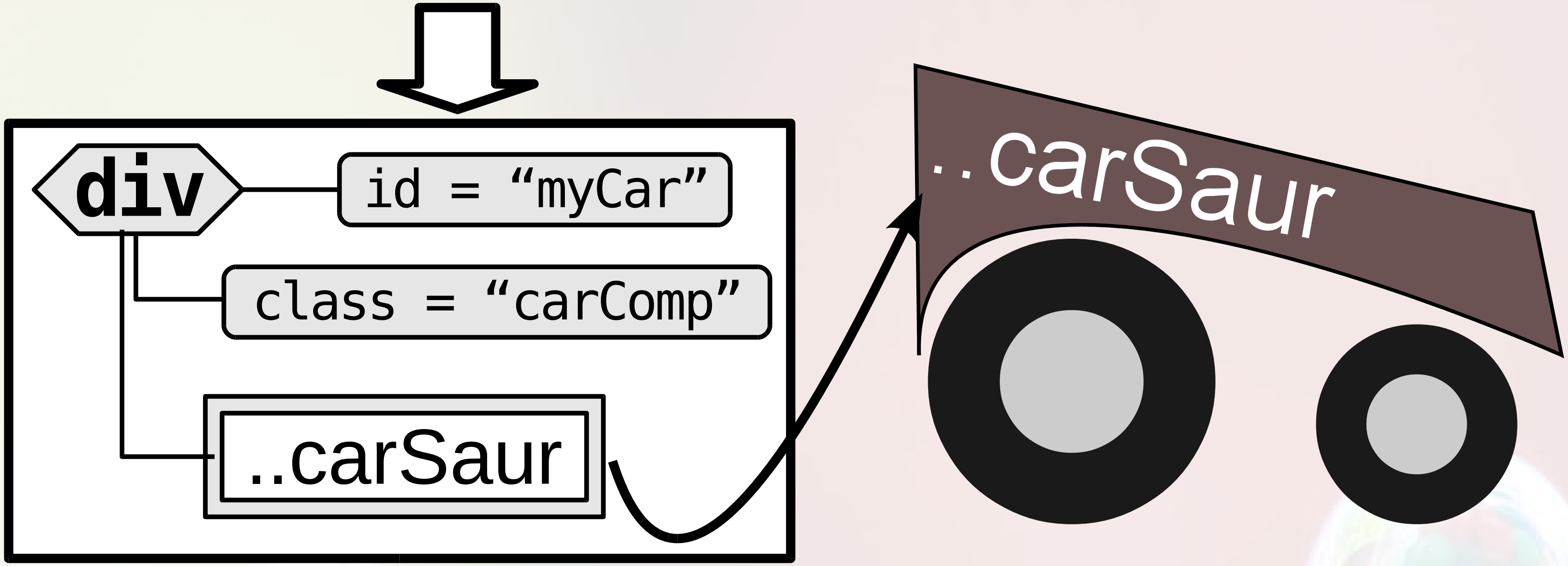
visão DOM

root

visão apresentação

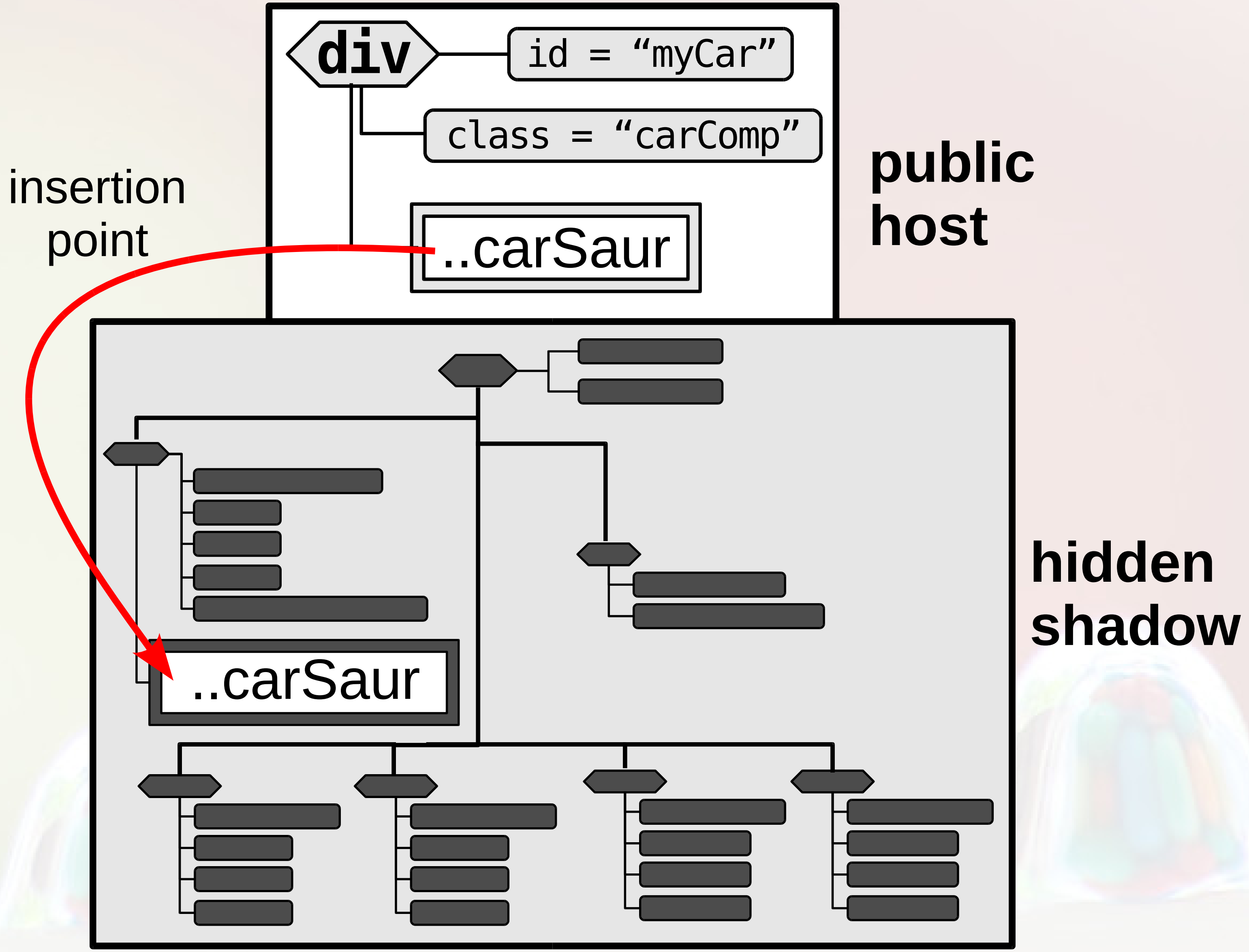
Shadow DOM – Carro
Interface

```
<div id="myCar" class="carComp">..carSaur</div>
```



Shadow DOM – Carro

Implementação



Shadow Car

```
<template
id="carComponent">

  <style scoped>
    …
  </style>

  <svg …>
    …
  </svg>

  <div id="textCarName"
       class="nameStyle"
       width="181px">
    <content></content>
  </div>
</template>
```

```
function applyTemplate()
{
  var myCar =

document.querySelector("#myCar");

  var carComponent =

document.querySelector("#carComponent")
         .content;

  var myCarShadow =
         myCar.webkitCreateShadowRoot();
         // standard: createShadowRoot()

  myCarShadow.appendChild(carComponent);
}
```

```
<div id="myCar" class="carComp">..carSaur</div>
```

Shadow Car

```
<template
id="carComponent">

  <style scoped>
    …
  </style>

  <svg …>
    …
  </svg>

  <div id="textCarName"
       class="nameStyle"
       width="181px">
    <content
select="#carName">
     </content>
  </div>
```

```
function applyTemplate()
{
  var myCar =

document.querySelector("#myCar");

  var carComponent =

document.querySelector("#carComponent")
         .content;

  var myCarShadow =
         myCar.webkitCreateShadowRoot();
         // standard: createShadowRoot()

  myCarShadow.appendChild(carComponent);
}
```

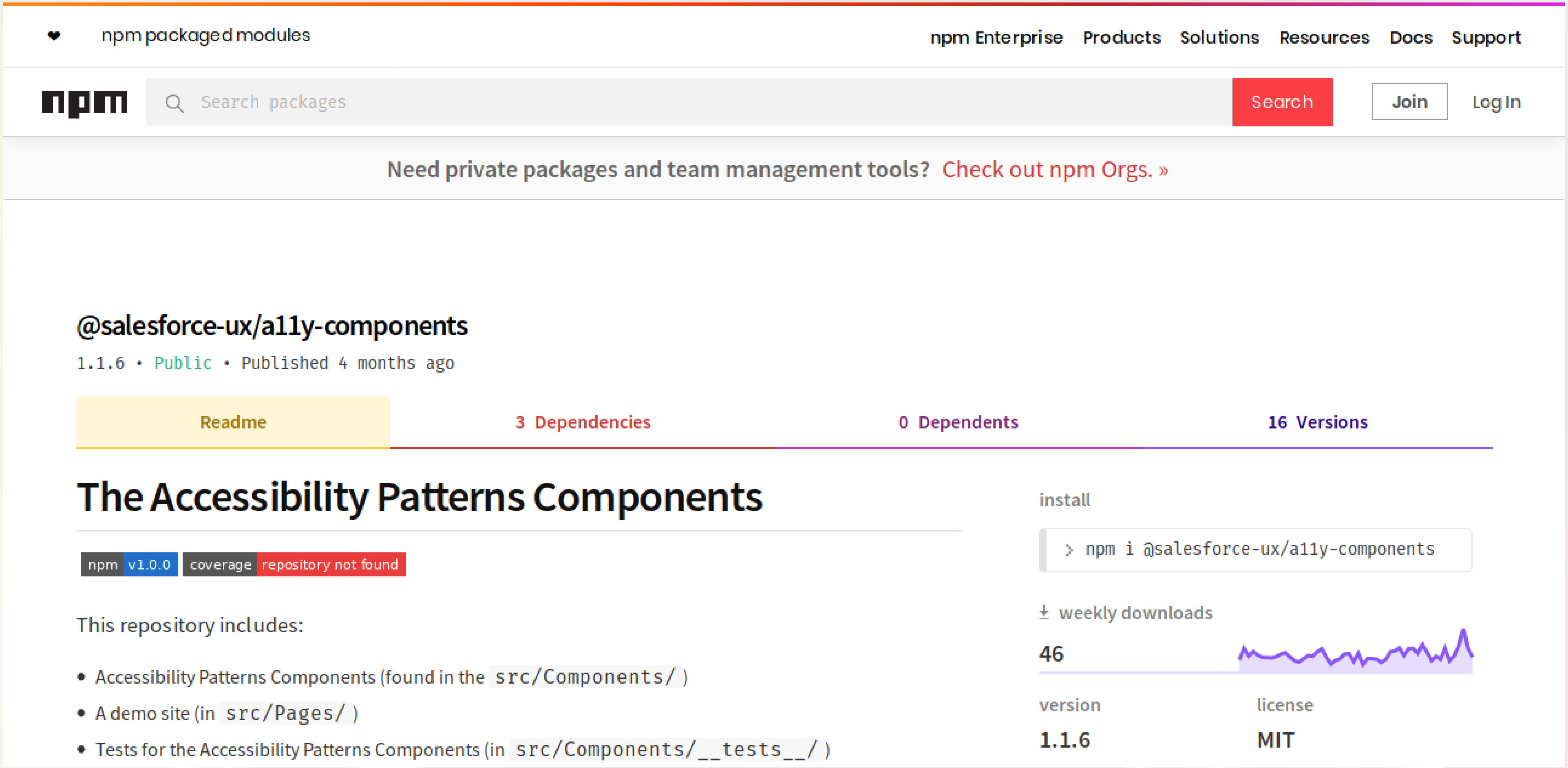
```
<div id="myCar" class="carComp">
  <div id="carName">..carSaur</div>
</div>
```

Lighting
Modelo

- Especificação / Padrão
  - Webcomponents Model
  - Lightning Web Components
- Implementação
  - Javascript / CSS / HTML
- Empacotamento / Distribuição
  - npm

Empacotamento - npm

https://www.npmjs.com



Need private packages and team management tools? Check out npm Orgs. »

@salesforce-ux/a11y-components

1.1.6 • Public • Published 4 months ago

The Accessibility Patterns Components

This repository includes:

- Accessibility Patterns Components (found in the `src/Components/`)
- A demo site (in `src/Pages/`)
- Tests for the Accessibility Patterns Components (in `src/Components/__tests__/`)

install

```
> npm i @salesforce-ux/a11y-components
```

weekly downloads

46

version

1.1.6

license

MIT

npm

- “npm is the world’s largest software registry. Open source developers from every continent use npm to share and borrow packages, and many organizations use npm to manage private development as well.” npm site (03/08/2019)

npm
Especificação do Pacote

Especificação JSON

```
{
  "name": "asdruboides",
  "version": "1.0.0",
  "description": "Big asdruboides package",
  "main": "index.js",
  "repository": {
    "type": "git",
    "url": "https://github.com/user/santanche.git"
  },
  "author": "Asdrubal <asdrubal@xyz.com>",
  "license": "MIT"
}
```

https://docs.npmjs.com/files/package.json

JSON

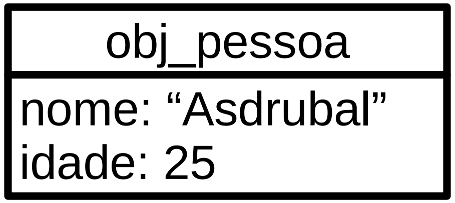
JavaScript

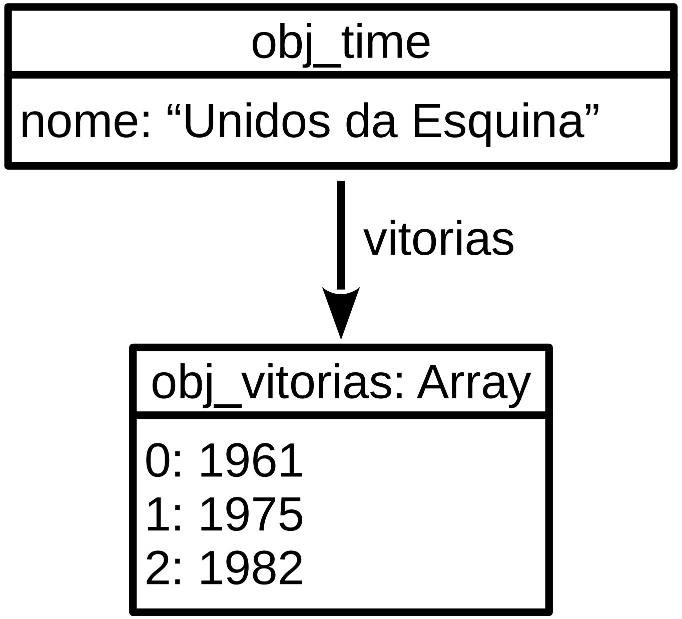
- Inventada por Brendan Eich na Netscape
- Incorporada no Internet Explore como Jscript
- ECMAScript → padronização (Ecma, 2011)
- Originalmente para pequenos scripts no navegador
- Limitações de acesso ao hardware local
  - segurança
  - independência de plataforma

Objetos em JavaScript

Objetos JS

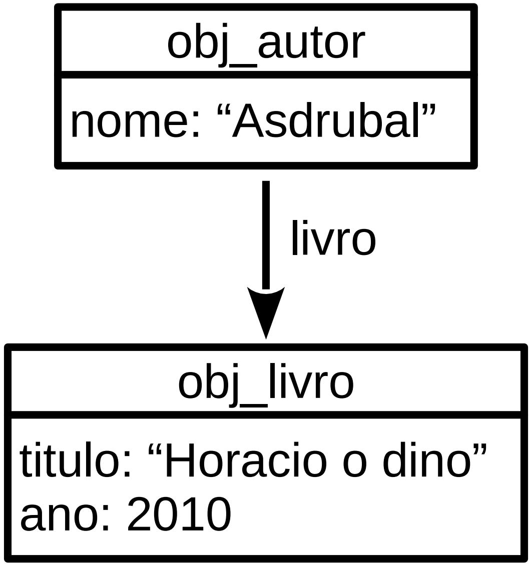
`{ }`	

`{`
`  "nome": "Asdrubal",`
`  "idade": 25`
`}`
	

`{`
`  "nome": "Unidos da Esquina",`
`  "vitorias": [1961, 1975, 1982]`
`}`
	 obj_time
	nome: “Unidos da Esquina”
	vitorias
	obj_vitorias: Array
	0: 1961
	1: 1975
	2: 1982

Objetos JS

```
{
 "nome": "Asdrubal",
 "livro": {
   "titulo": "Horacio o dino",
   "ano": 2010
  }
}
```



obj_autor

nome: “Asdrubal”

livro

obj_livro

titulo: “Horacio o dino”
ano: 2010

JSON
JavaScript Object Notation

- Padrão aberto de intercâmbio de objetos
- Baseado na notação JavaScript
- Incorporado ao ECMAScript (Ecma, 2011)
- Adotado por diversas linguagens (http://json.org/)

PIP e PyPi

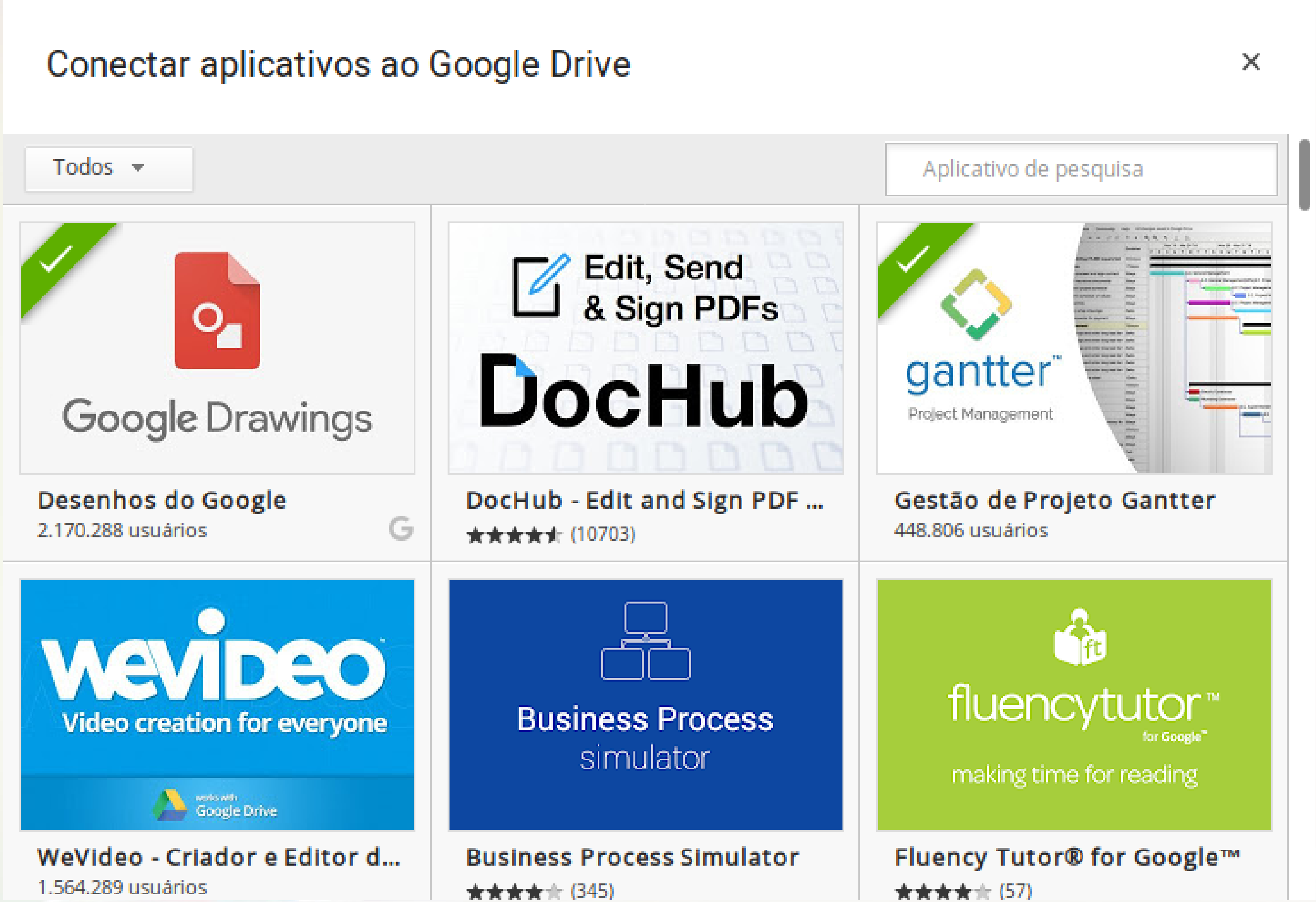
https://pypi.org

- PIP
  - Python Package Manager
- PyPi
  - Python Package Index

Cenário 2
Apps Google Drive

Apps Google Drive

https://drive.google.com



Conectar aplicativos ao Google Drive

Desenhos do Google
2.170.288 usuários

DocHub - Edit and Sign PDF ...
★★★★★ (10703)

Gestão de Projeto Gantter
448.806 usuários

WeVideo - Criador e Editor d...
1.564.289 usuários

Business Process Simulator
★★★★☆ (345)

Fluency Tutor® for Google™
★★★★☆ (57)

Google Drive App
Modelo

- Especificação / Padrão
  - Serviços REST
  - JSON
- Implementação
  - Diversas linguagens
- Empacotamento / Distribuição
  - online

Arquitetura Orientada a Serviços

Arquitetura Orientada a Serviços – *Service Oriented Architecture* (SOA)

Computação Orientada a Serviços

> “Computação orientada a serviços é o paradigma da computação que utiliza serviços como elementos fundamentais para o desenvolvimento de aplicações.” (Papazoglou, 2003)

Service Oriented Architecture (SOA)

Componentes auto-descritivos

Abertos

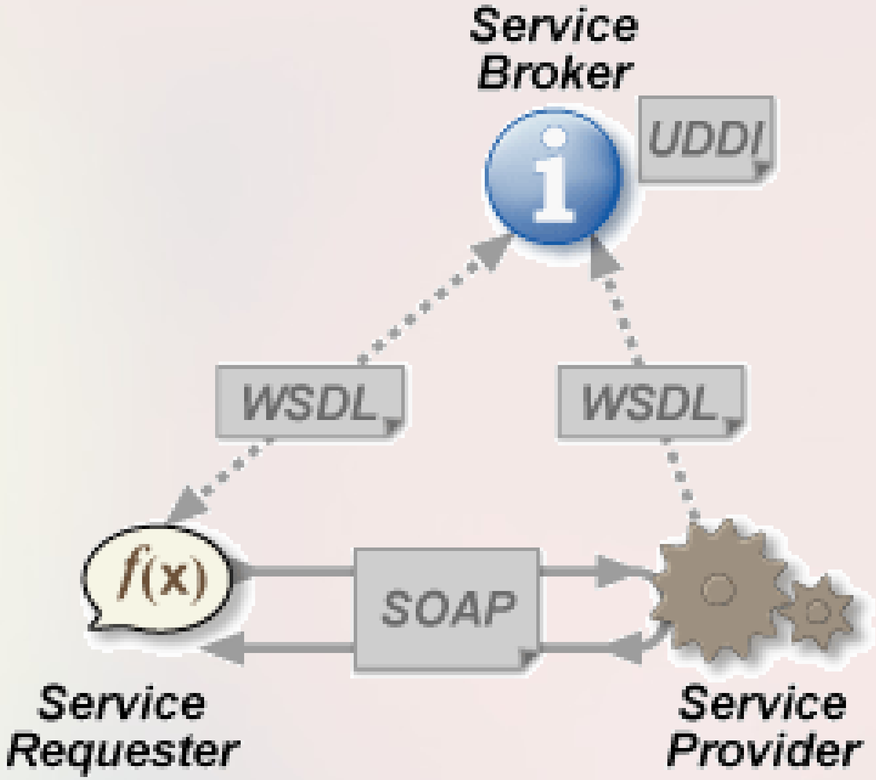
Possibilitam composição rápida e a baixo custo

São providos por provedores de serviços

(Papazoglou, 2003)

“SOA é um estilo arquitetural cujo objetivo é alcançar baixo acoplamento entre agentes de software em interação.” (He, 2003)

Baseada em Mensagens
Web Services

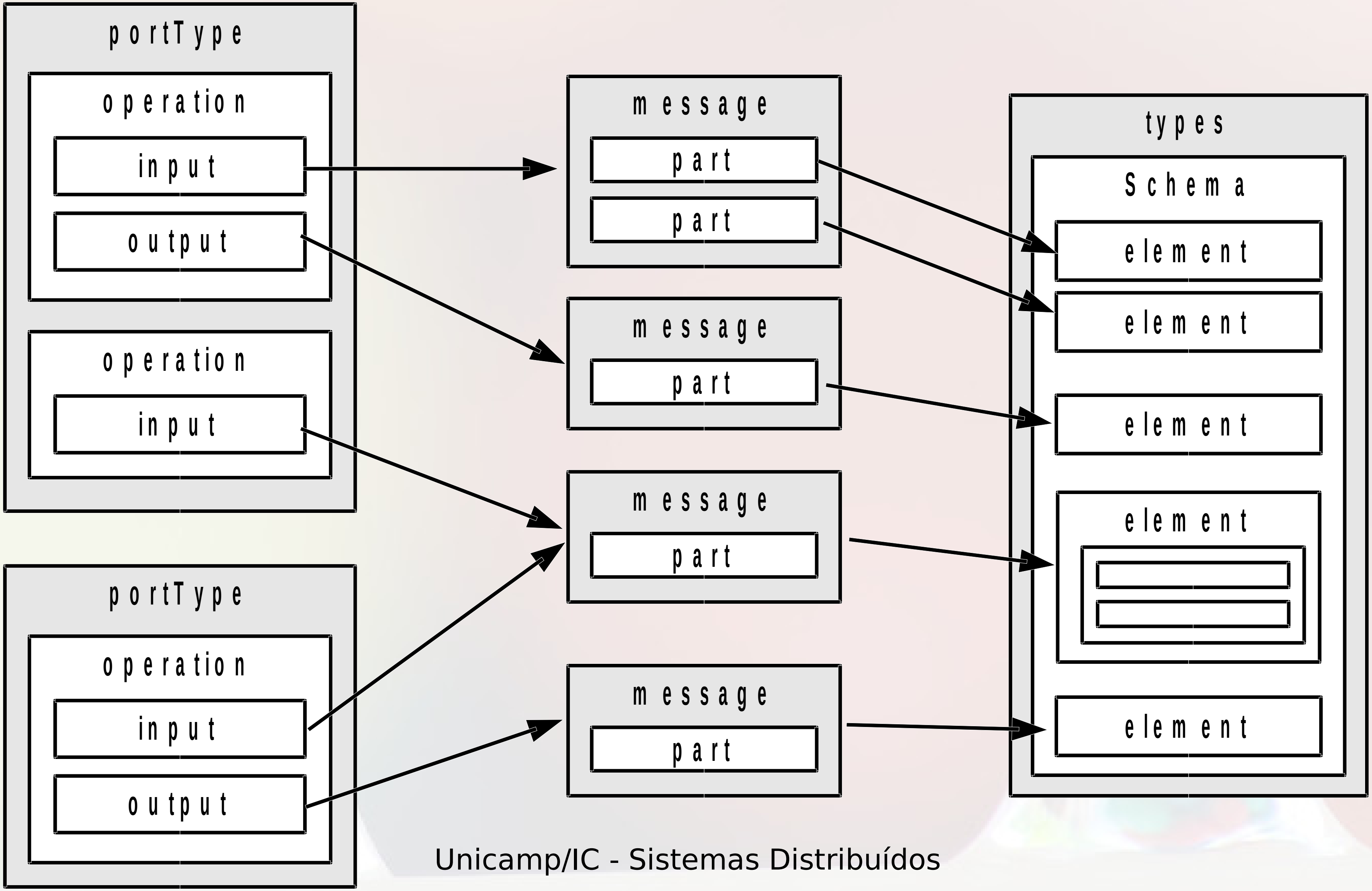


autor: H. Voormann

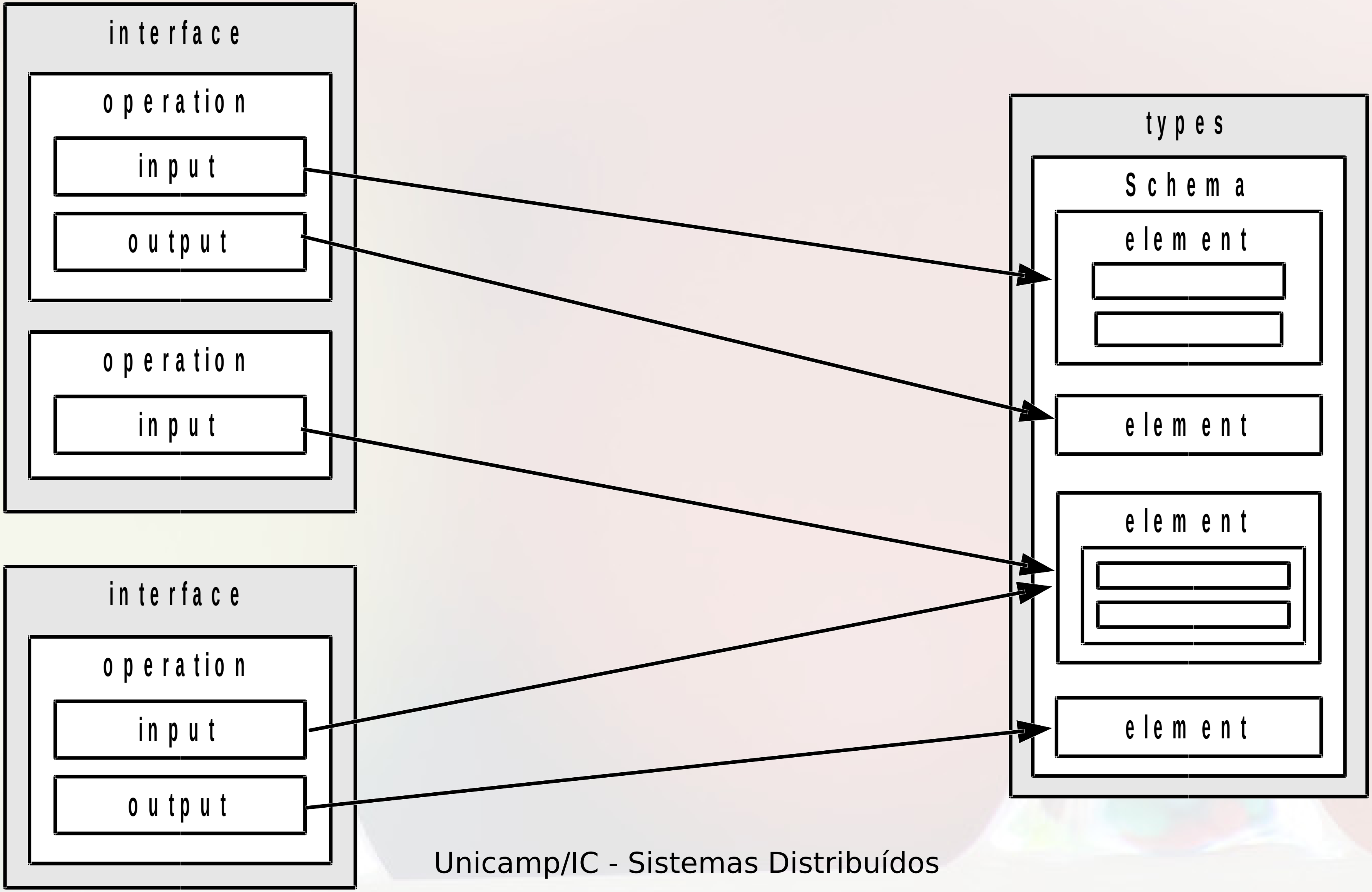
http://en.wikipedia.org/wiki/Image:Webservices.png

Web Services

WSDL 1



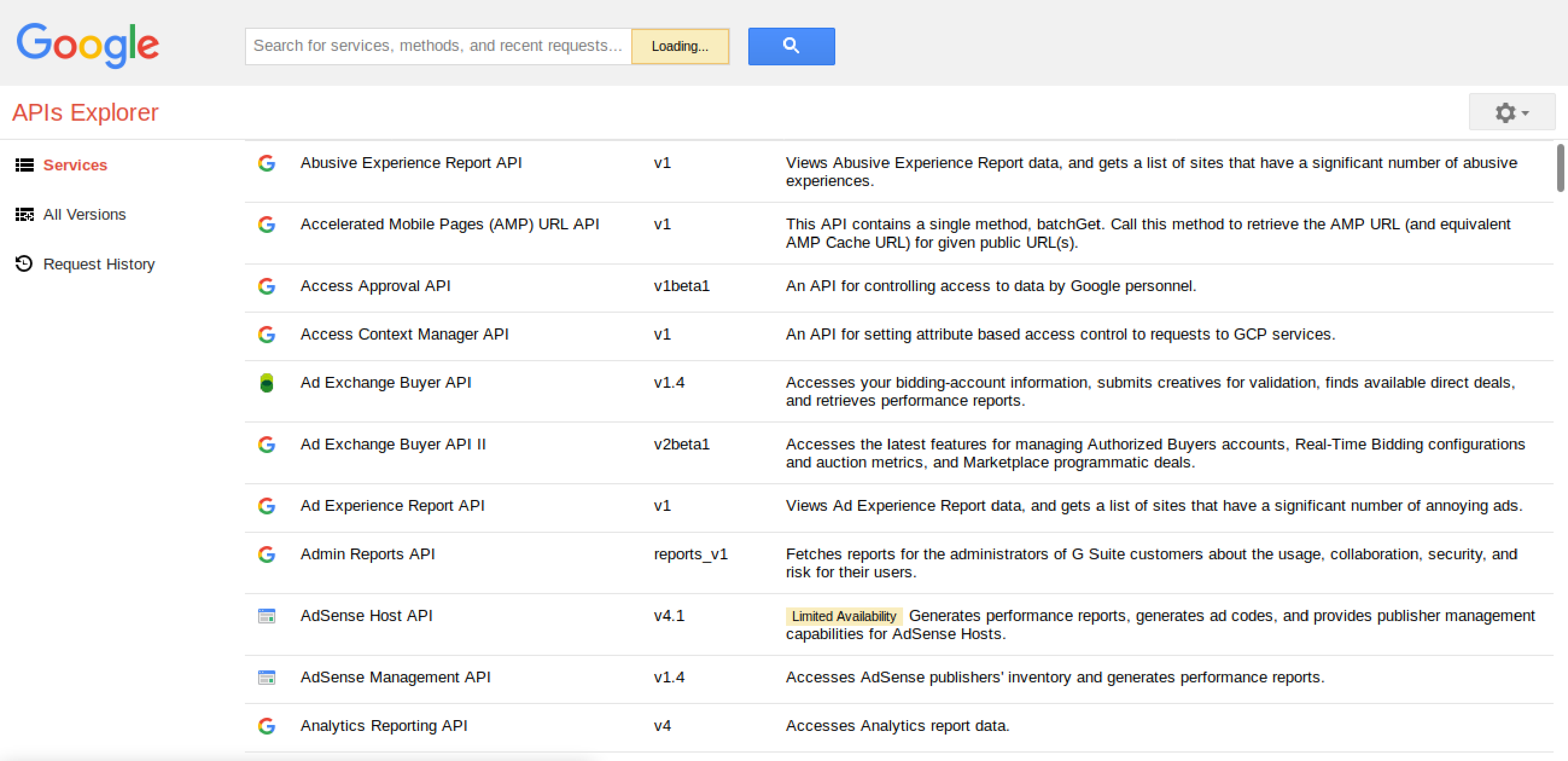
Web Services
WSDL 2



Simplificação REST e JSON

Google API Explorer

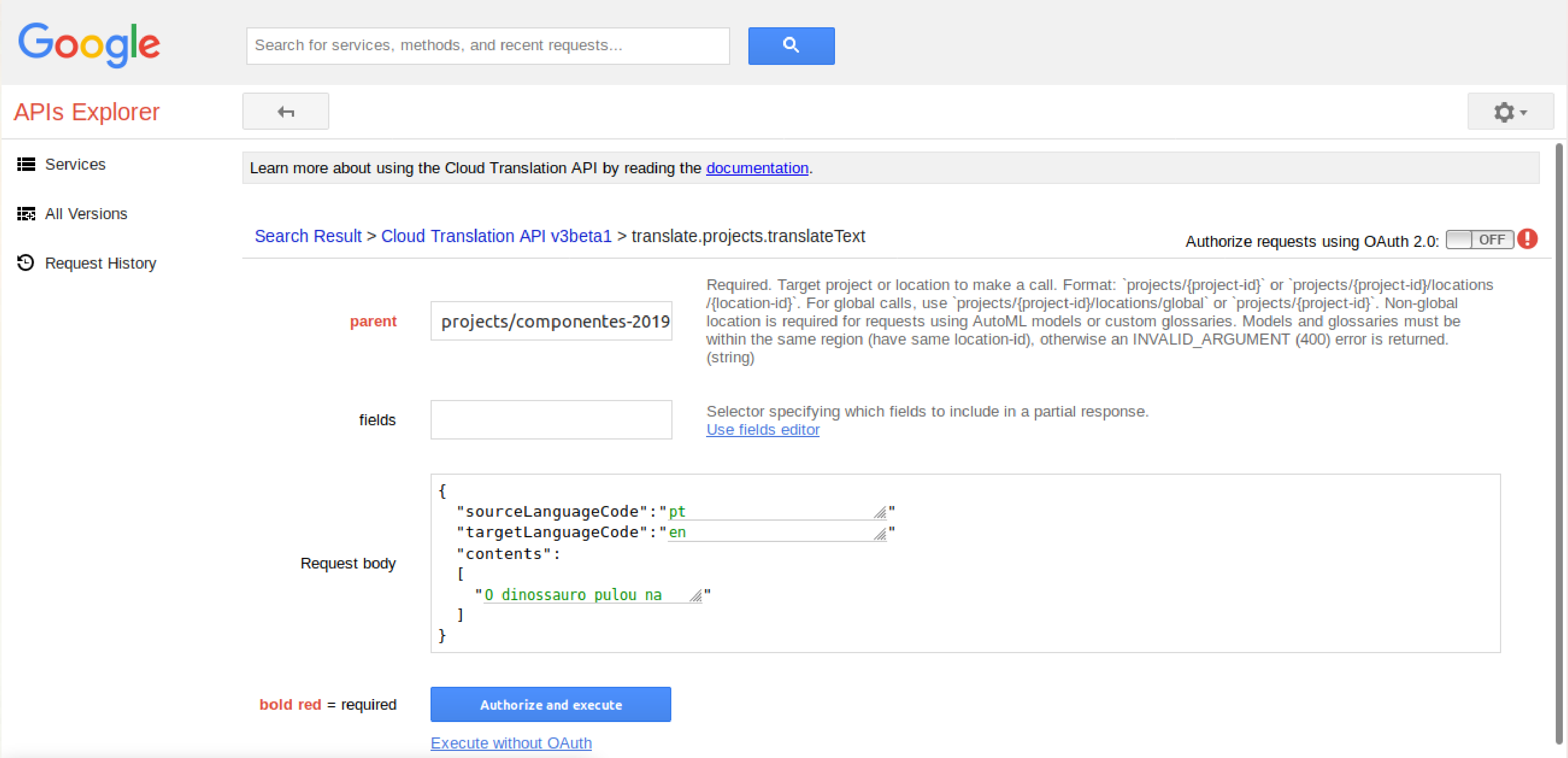
https://developers.google.com/apis-explorer/



Abusive Experience Report API	v1	Views Abusive Experience Report data, and gets a list of sites that have a significant number of abusive experiences.
Accelerated Mobile Pages (AMP) URL API	v1	This API contains a single method, batchGet. Call this method to retrieve the AMP URL (and equivalent AMP Cache URL) for given public URL(s).
Access Approval API	v1beta1	An API for controlling access to data by Google personnel.
Access Context Manager API	v1	An API for setting attribute based access control to requests to GCP services.
Ad Exchange Buyer API	v1.4	Accesses your bidding-account information, submits creatives for validation, finds available direct deals, and retrieves performance reports.
Ad Exchange Buyer API II	v2beta1	Accesses the latest features for managing Authorized Buyers accounts, Real-Time Bidding configurations and auction metrics, and Marketplace programmatic deals.
Ad Experience Report API	v1	Views Ad Experience Report data, and gets a list of sites that have a significant number of annoying ads.
Admin Reports API	reports_v1	Fetches reports for the administrators of G Suite customers about the usage, collaboration, security, and risk for their users.
AdSense Host API	v4.1	Limited Availability Generates performance reports, generates ad codes, and provides publisher management capabilities for AdSense Hosts.
AdSense Management API	v1.4	Accesses AdSense publishers' inventory and generates performance reports.
Analytics Reporting API	v4	Accesses Analytics report data.

Cloud Translation API v3beta1

https://developers.google.com/apis-explorer/#search/translate/translate/v3beta1/translate.projects.translateText



Search for services, methods, and recent requests...

APIs Explorer

Learn more about using the Cloud Translation API by reading the documentation.

Search Result > Cloud Translation API v3beta1 > translate.projects.translateText

Authorize requests using OAuth 2.0: OFF

parent projects/componentes-2019

Required. Target project or location to make a call. Format: `projects/{project-id}` or `projects/{project-id}/locations/{location-id}`. For global calls, use `projects/{project-id}/locations/global` or `projects/{project-id}`. Non-global location is required for requests using AutoML models or custom glossaries. Models and glossaries must be within the same region (have same location-id), otherwise an INVALID_ARGUMENT (400) error is returned. (string)

fields

Selector specifying which fields to include in a partial response.
Use fields editor

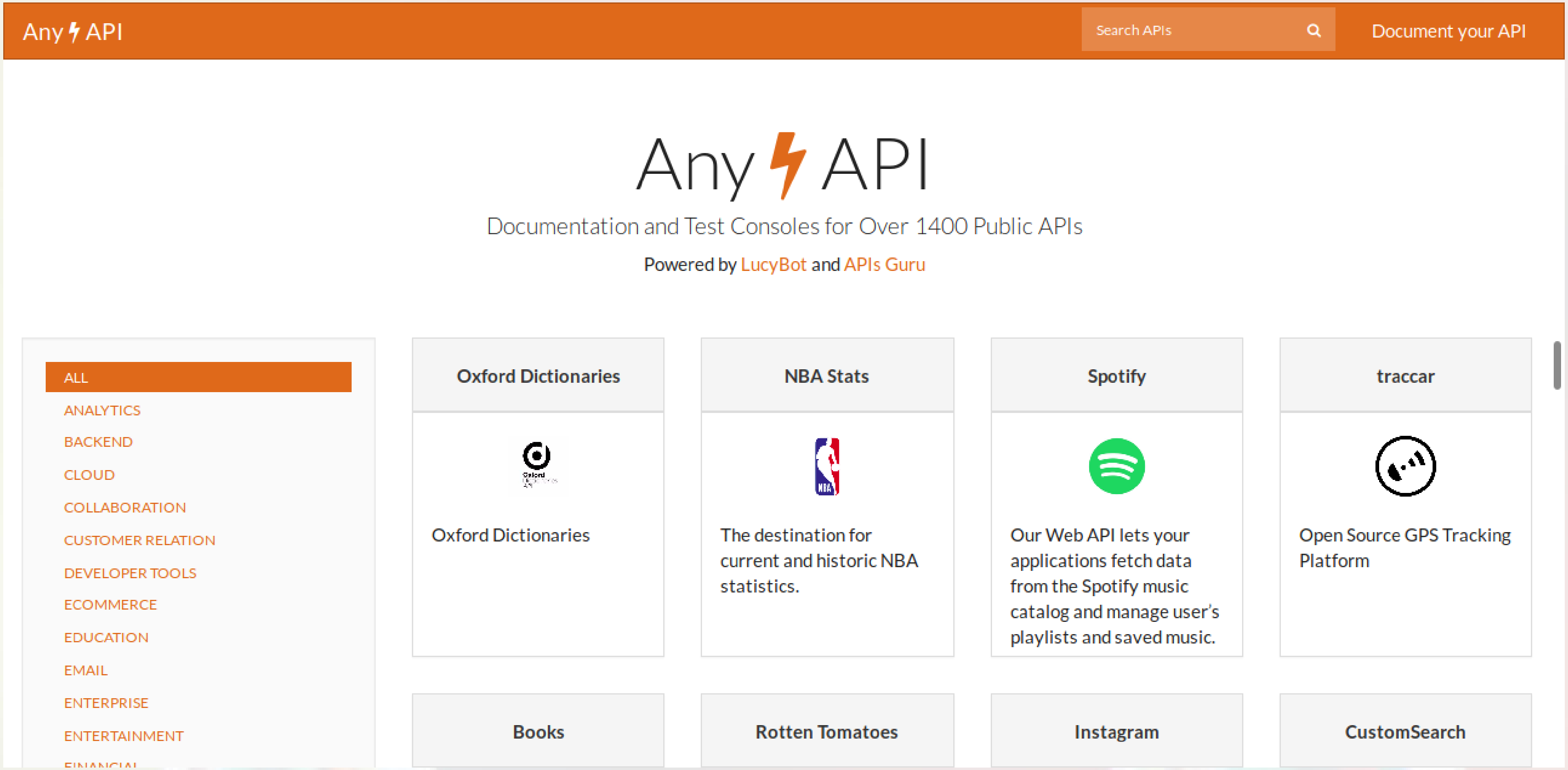
Request body

```
{
  "sourceLanguageCode":"pt"
  "targetLanguageCode":"en"
  "contents":
  [
    "O dinossauro pulou na"
  ]
}
```

bold red = required

Any API

https://any-api.com



Any API

Search APIs

Document your API

Any API

Documentation and Test Consoles for Over 1400 Public APIs

Powered by LucyBot and APIs Guru

- ALL
- ANALYTICS
- BACKEND
- CLOUD
- COLLABORATION
- CUSTOMER RELATION
- DEVELOPER TOOLS
- ECOMMERCE
- EDUCATION
- EMAIL
- ENTERPRISE
- ENTERTAINMENT
- FINANCIAL

Oxford Dictionaries

Oxford Dictionaries

NBA Stats

The destination for current and historic NBA statistics.

Spotify

Our Web API lets your applications fetch data from the Spotify music catalog and manage user's playlists and saved music.

traccar

Open Source GPS Tracking Platform

Books

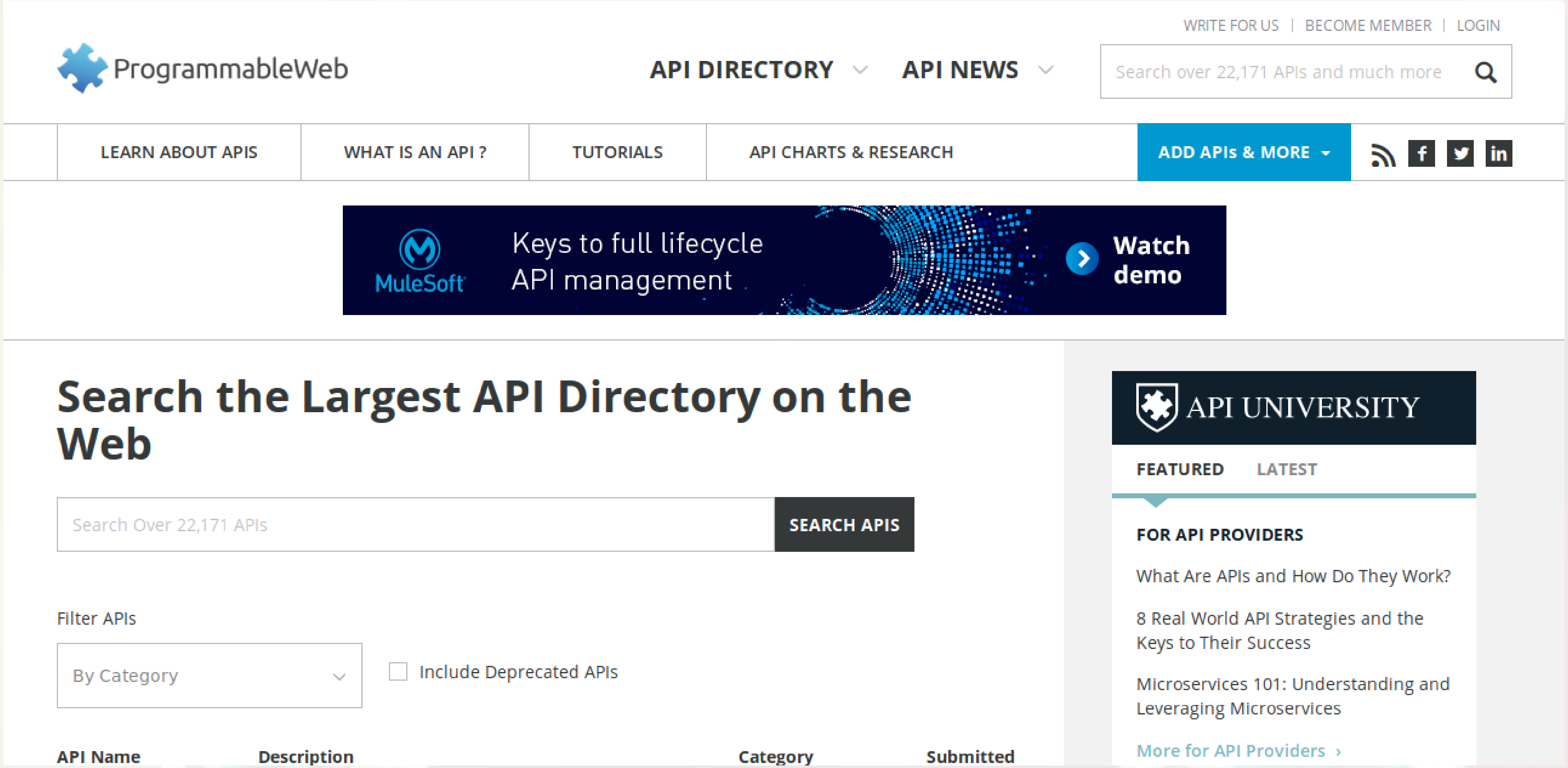
Rotten Tomatoes

Instagram

CustomSearch

ProgrammableWeb

https://www.programmableweb.com



WRITE FOR US | BECOME MEMBER | LOGIN

ProgrammableWeb

API DIRECTORY API NEWS

Search over 22,171 APIs and much more

LEARN ABOUT APIS | WHAT IS AN API ? | TUTORIALS | API CHARTS & RESEARCH | ADD APIs & MORE

Search the Largest API Directory on the Web

Search Over 22,171 APIs SEARCH APIS

Filter APIs

By Category

□ Include Deprecated APIs

API Name | Description | Category | Submitted

API UNIVERSITY

FEATURED LATEST

FOR API PROVIDERS

What Are APIs and How Do They Work?

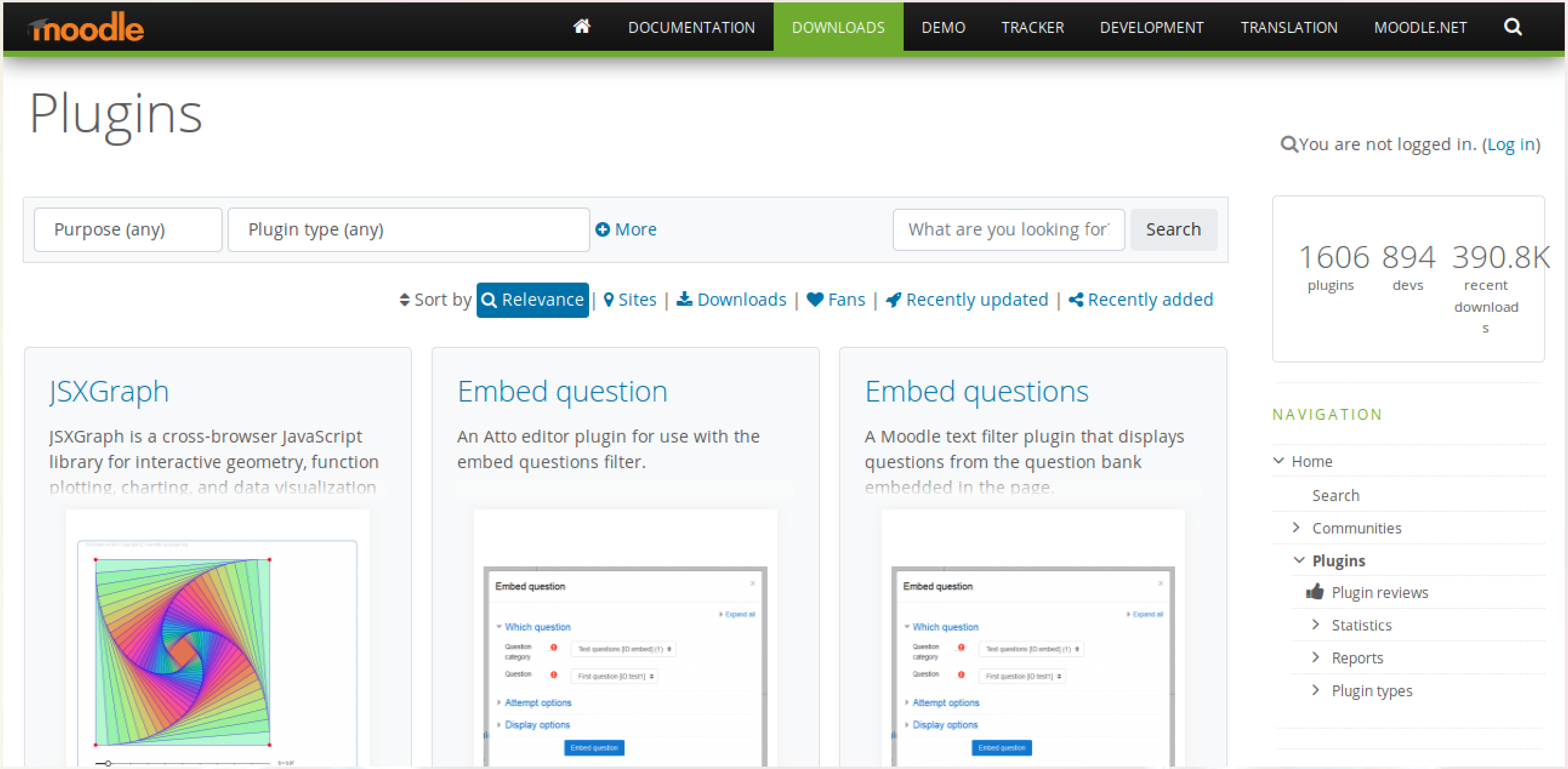
8 Real World API Strategies and the Keys to Their Success

Microservices 101: Understanding and Leveraging Microservices

More for API Providers ›

Cenário 3
Moodle Plugins

Moodle Plugins



Plugins

You are not logged in. (Log in)

Purpose (any) | Plugin type (any) | More | What are you looking for | Search

Sort by Relevance | Sites | Downloads | Fans | Recently updated | Recently added

1606 plugins | 894 devs | 390.8K recent downloads

JSXGraph

JSXGraph is a cross-browser JavaScript library for interactive geometry, function plotting, charting, and data visualization

Embed question

An Atto editor plugin for use with the embed questions filter.

Embed questions

A Moodle text filter plugin that displays questions from the question bank embedded in the page.

NAVIGATION

- Home
  - Search
  - Communities
  - Plugins
    - Plugin reviews
    - Statistics
    - Reports
    - Plugin types

Referências

- Abowd, G. D., Allen, R., Garlan, D. **Formalizing style to understand descriptions of software architecture**. ACM Trans. Softw. Eng. Methodol., ACM Press, 1995, 4, 319-364.
- Bass, L., Clements, P., Kazman, R. **Software Architecture in Practice.** Addison-Wesley, 2003.
- Cheesman, J., & Daniels, J. (2000). **UML Components: A simple process for specifying component-based software**. Addison-Wesley.
- Garlan, D. et al. **Architectural Mismatch (Why It’s Hard to Build Systems Out of Existing Parts)**. Proceedings, 17th Int. Conf. on Software Engineering. Seattle, WA, April 23-30, 1995.
- He, H. **What Is Service-Oriented Architecture**. Setembro 2003. Disponível em http://www.xml.com/pub/a/ws/2003/09/30/soa.html
- ISO/IEC/IEEE 24765:2010 **Systems and software engineering — Vocabulary**

Referências

- Jha, P. C., Bali, V., Narula, S., & Kalra, M. (2014). **Optimal component selection based on cohesion & coupling for component based software system under build-or-buy scheme**. Journal of Computational Science, 5(2), 233–242.
- Papazoglou, M. P., Georgakopoulos, D. **Service-oriented computing.** Commun. ACM, 2003, 46, 25-28.
- Parnas, D. **On the Design and Development of Program Families**. IEEE Transactions on Software Engineering SE-2, 1976, 1, 1-9.

Referências

- Software Engineering Standards Committee of the IEEE Computer Society. **Systems and software engineering – Recommended practice for architectural description of software-intensive systems**, ISO/IEC 42010 IEEE Std 1471-2000 First edition 2007-07-15, Julho 2007.
- Sommerville, I. (2007) **Software Engineering**, 8th. ed. Addison Wesley.
- Stevens, W. P., Myers, G. J., & Constantine, L. L. (1974). **Structured design**. IBM Systems Journal, 13(2), 115–139.
- Taylor, R. N. , et al. **A Component- and Message-Based Architectural Style for GUI Software**. IEEE Trans. Software Engineering, IEEE Press, 1996, 22, 390-406.

Referências

- Comella-Dorda, S. **Component Object Model (COM), DCOM, and Related Capabilities**. Carnegie Mellon University, março de 2001.
- Cook, S., Bock, C., Rivett, P., Rutt, T., Seidewitz, E., Selic, B., & Tolbert, D. (2015). **OMG Unified Modeling Language (OMG UML) - version 2.5**. Needham. Retrieved from http://www.omg.org/spec/UML/2.5/
- Gamma, E. Helm, R. Johnson, R. Vlissides, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995.
- Krueger, C. W. Software Reuse. ACM Comput. Surv., ACM Press, 1992, 24, 131-183.
- Liskov, B. **Keynote address - data abstraction and hierarchy**. OOPSLA '87: Addendum to the proceedings on Object-oriented programming systems, languages and applications (Addendum), ACM Press, 1987, 17-34.
- Martin, R. C. **Design Principles and Design Patterns**. Object

Referências

- Mcilroy, M. D. Naur, P. & Randell, B. (ed.) **Mass Produced Software Components. Software Engineering: Report of a conference sponsored by the NATO Science Committee**, 1968.
- Meyer, B. (1992). **Applying “design by contract.”** Computer, 25(10), 40–51. https://doi.org/10.1109/2.161279
- Meyer, B. (2000) **Object-Oriented Software Construction** (2nd Edition). Prentice Hall.
- Parrish, R. **XPCOM Part 1: An introduction to XPCOM**. DeveloperWorks, fevereiro de 2001, on-line: http://www.ibm.com/developerworks/webservices/library/co-xpcom.html
- Willliams, S. & Kindel, C. **The Component Object Model: A Technical Overview**. Microsoft Corporation, 1994

André Santanchè

http://www.ic.unicamp.br/~santanche

Licença

- Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições: Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença.
- Mais detalhes sobre a referida licença Creative Commons veja no link: http://creativecommons.org/licenses/by-nc-sa/3.0/
- Agradecimento a Doug Wheller [http://www.flickr.com/photos/doug88888/] por sua fotografia “Two drops” usada na capa e nos fundos, disponível em [http://www.flickr.com/photos/doug88888/5817711538/] vide licença específica da fotografia.

Conceito