



FRAMEWORK

Alunos:

Gustavo Alexandre Dias RA 2052229

Camila de Souza Gonçalves RA 926604

Gabriel Pancaldi RA 1914758

Igor Bregagnoli RA: 2053730



Principais características

- O AdonisJS é um framework Node.js segue o padrão MVC
- Framework back-end para Node.js, escrita em TypeScript.
- Pastas e ambiente pre configurado
- Possui recursos como CLI, File Upload simples, validações
- Pacotes que enriquecem o ecossistema como ORM, AUTENTICACAO e Autorização

Objetivo do Framework

- **Facilitar a criação de aplicações web e API's em Node.js.**
- **Desenvolvedores podem se concentrar em criar recursos de negócios em vez de lidar com a infraestrutura e configurações de baixo nível.**
- **Promover as melhores práticas e padronização no desenvolvimento web. Incentivar os desenvolvedores a seguir um conjunto de convenções e diretrizes.**
- **Facilitar o processo de testes, autenticação de software e configuração de banco de dados em uma aplicação web ou API's.**



Introdução

Quando usar ?

- Projetos mais complexos.
- Exigem estruturação e organização robusta e padronizada.
- Projetos que possui requisitos técnicos prontos embutidos no framework.

História

Harinder Virk (o criador e principal mantenedor da estrutura), juntamente com a ajuda dos seguintes membros da equipe principal.

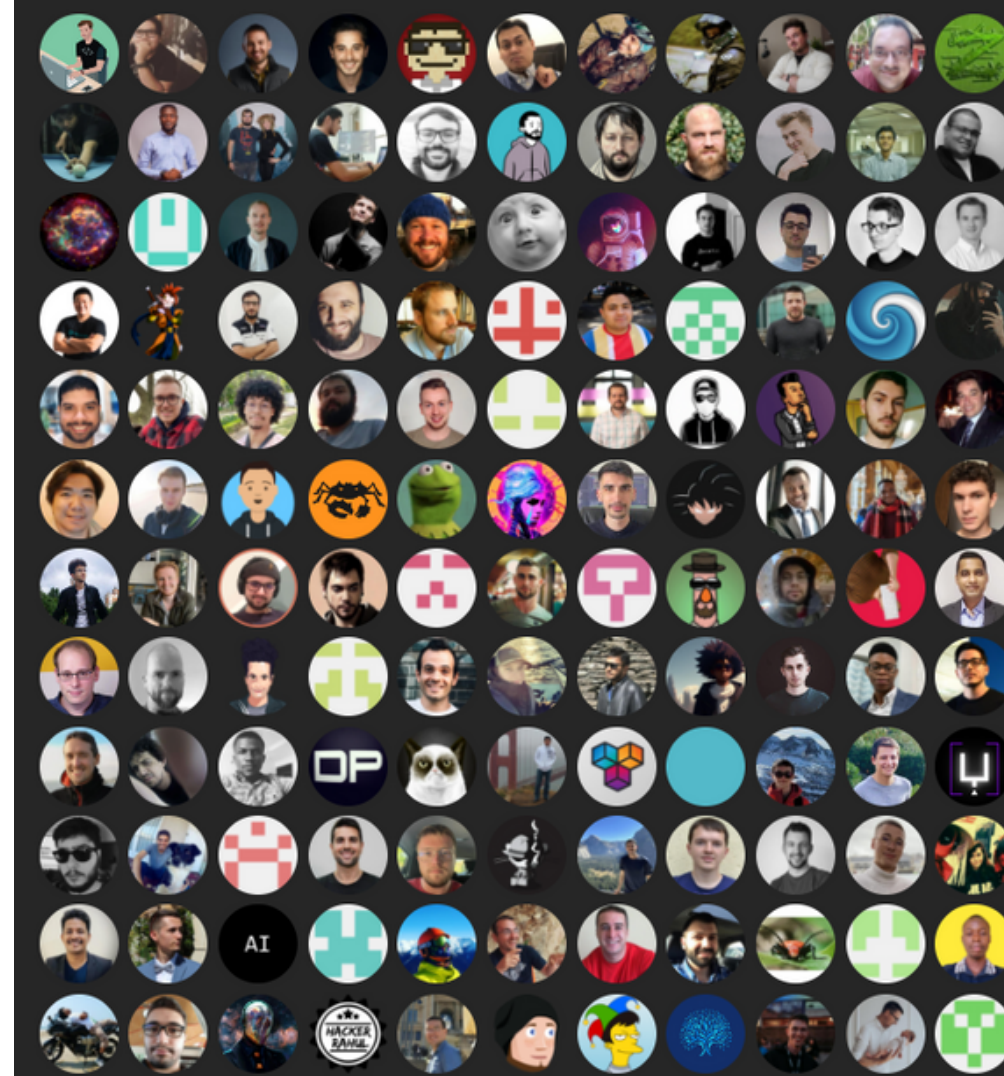
Romain Lanz é um desenvolvedor full-stack da FIVB. Ele acredita que o conhecimento deve ser gratuito e acessível a todos, e está trabalhando para isso.

Michaël Zasso é um engenheiro de software de pesquisa científica e membro do Node.js Technical Steering Committee.

Julien Ripouteau é um desenvolvedor francês apaixonado por Javascript e Open Source. Ele também é o criador da extensão AdonisJS VsCode , AdonisJS Sail e do site packages.adonisjs.com .

Colaboradores (140 e crescendo)

AdonisJS é o resultado do amor e do trabalho de seus colaboradores.





ADONIS


Documentação




Site Oficial

- Fundamentos
- HTTP
- Visualização e modelos
- Validador
- Base de dados
- ORM
- Autenticacao
- Seguranca
- teste

<https://docs.adonisjs.com/guides/introduction>

 adonis

DocumentationLearnResources 

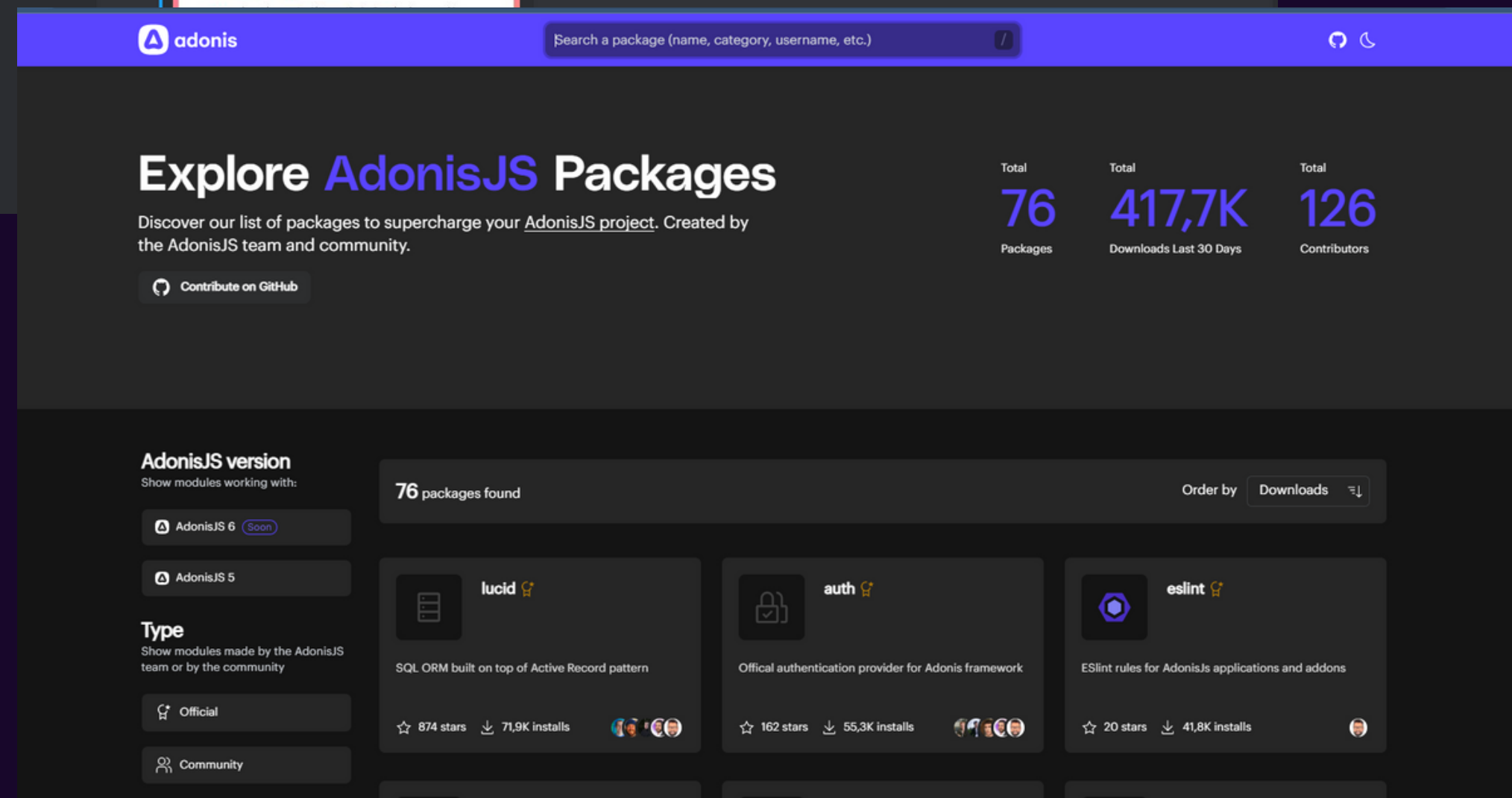
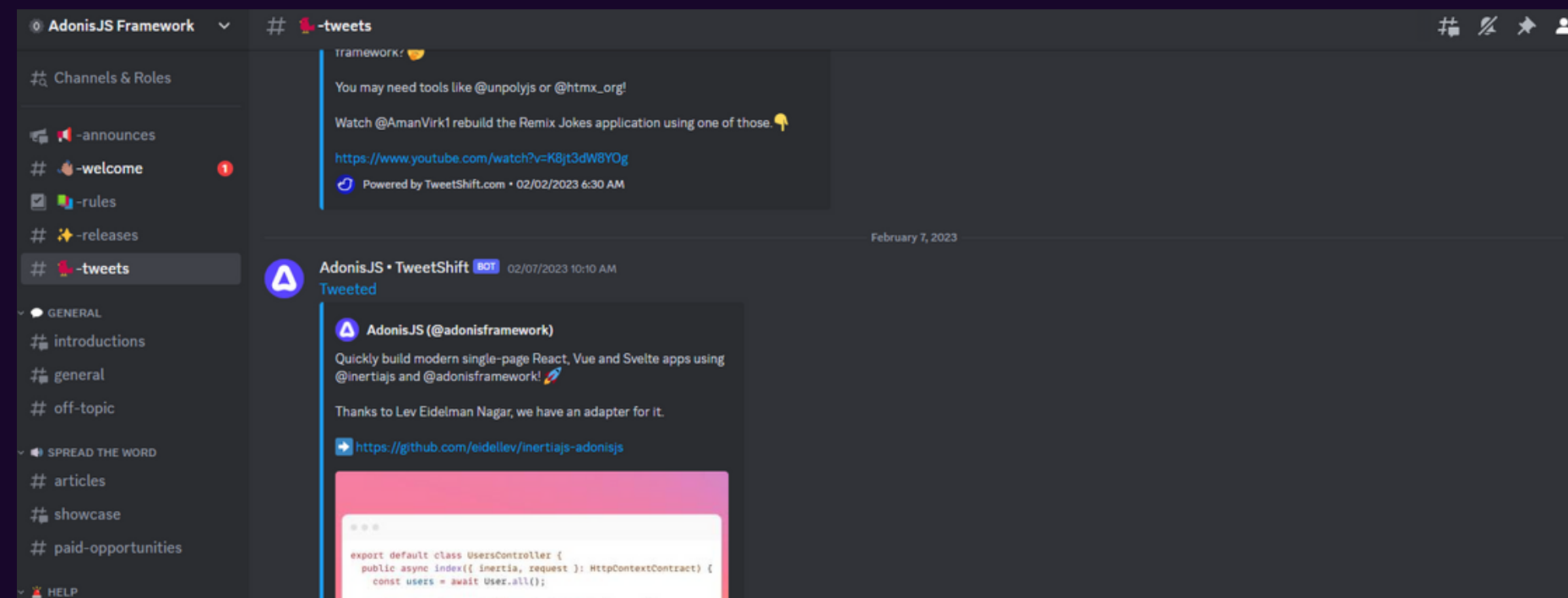
A fully featured web framework *for* Node.js

AdonisJS includes everything you need to create a fully functional web app or an API server. So stop wasting hours downloading and assembling hundreds of packages — Use AdonisJS and be productive from day one.

[Get Started](#)[AdonisJS at a Glance](#)

Comunidade

- **Newsletter** - Receba atualizações da estrutura e da comunidade diretamente em sua caixa de entrada. Sem spam. Máximo de 1 e-mail por mês.
- **Twitter** - Mantenha-se atualizado.
- **Extensão do VS Code** - Uma extensão oficial do VS Code para aumentar sua experiência de desenvolvimento.
- **Awesome AdonisJS** - Uma coleção de pacotes, tutoriais e aplicativos criados pelos membros da comunidade.





Comparação / Exemplo de uso

GET

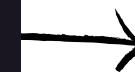
```
Route.get('simple/olaMundo', ({response}) => {  
  response.send('Ola mundo')  
})
```



```
router.get('simple/olaMundo', (req, resposta) => {  
  resposta.send('Ola mundo')  
})
```

POST

```
Route.post('simple/lista/:nome/:idade', ({request, response}) => {  
  // const {name, age} = request.body()  
  const nome = request.param('nome')  
  const idade = request.param('idade')  
  list.push({nome: nome, idade: idade})  
  response.status(200).send(list)  
  console.log(list)  
})
```



```
router.post('simple/list/:nome/:idade', (req, res) => {  
  let nome = req.params.nome  
  let idade = req.params.idade  
  list.push({nome: nome, idade: idade})  
  res.status(200).send(list)  
  console.log(list)  
})
```



Comparação / Exemplo de uso

PUT

```
Route.put('simple/alteraLista/:nome/:nomeAlterar', ({request, response}) => {  
  // const nome = request.params()[0]  
  const nome = request.param('nome')  
  const nomeAlterar = request.param('nomeAlterar')  
  for(let i = 0; i < list.length ; i++) {  
    if(list[i].nome == nome) {  
      list[i] = {nome: nomeAlterar, idade: list[i].idade}  
    }  
  }  
  response.status(200).send(list)  
})
```

You 2 hours ago • enviando projeto Adonis CRUD

```
router.put('simple/alteraLista/:nome/:nomeAlterar', (req, res) => {  
  const nome = req.params.nome  
  const nomeAlterar = req.params.nomeAlterar  
  for(let i = 0; i < list.length; i++) {  
    if(list[i].nome == nome) {  
      list[i] = {nome: nomeAlterar, idade: list[i].idade}  
    }  
  }  
  res.status(200).send(list)  
})
```

DELETE

```
Route.delete('/simple/deletaLista', ({response}) => {  
  list.pop()  
  response.status(200).send(list)  
})
```

```
router.delete('/simple/deletaLista', (req, res) => {  
  list.pop()  
  res.status(200).send(list)  
})
```




Middleware

Route

```
.get('/users/:id', async () => {  
  return 'Show user'  
})  
  
.middleware(async (ctx, next) => {  
  console.log(`Inside middleware ${ctx.request.url()}`)  
  await next()  
})
```

```
Server.middleware.registerNamed({  
  auth: () => import('App/Middleware/Auth')  
})
```

Route

```
.get('dashboard', 'DashboardController.index')  
.middleware('auth') // 👉
```



ADONIS

Instalação

```
# check node.js version
```

```
node -v
```

Criando um novo projeto

```
node init adonis-ts-app@latest adonisJS
```

Iniciando servidor de desenvolvimento

```
node ace serve --watch
```

CUSTOMIZE PROJECT

```
> Select the project structure ... Press <ENTER> to select
> api (Tailored for creating a REST API server)
  web (Traditional web application with server rendered templates)
  slim (A smallest possible AdonisJS application)
```



CUSTOMIZE PROJECT

```
> Select the project structure · api
> Enter the project name · adonisJS
> Setup eslint? (y/N) · true
> Setup prettier? (y/N) · true
```

RUNNING TASKS

```
> Scaffold project 63 ms
> Install dependencies
  [ wait ] eslint, eslint-plugin-adonis, prettier, eslint-config-prettier, and
> Configure installed packages
```



Pacotes

ORM

```
npm i @adonisjs/lucid
```

CONFIGURAÇÃO

```
npm ace configure @adonisjs/lucid
```

```
is\adonisJS>  
is\adonisJS> node ace make:model Disciplina -m|
```

```
onisJS> node ace migration:run|
```

```
s\adonisJS> node ace make:controller Disciplina  
-Controller to
```

```
PS C:\Users\diasg\Desktop\projeto_Adonis\adonisJS> node ace configure @adonisjs/lucid  
> Select the database driver you want to use ... Press <SPACE> to select  
( ) SQLite  
( ) MySQL / MariaDB  
( ) PostgreSQL  
( ) OracleDB  
( ) Microsoft SQL Server
```



ADONIS

{ code }

Considerações finais





OBRIGADO!
DÚVIDAS ??

