

Analise e Desenvolvimento de Sistemas

Práticas DevOps

```
1287     $return[$day->shot_date] = $day;
1288 }
1289 return $return;
1290 }
1291
1292 static function day_images_list($date, $studio) {
1293     global $global_studio_list;
1294     if(!in_array($studio, $global_studio_list)) die("error studio");
1295     $date = mysql::escape($date);
1296     if(mysql::count("image_date", "shot_date = '$date'") < 1) die('date not found');
1297     $studio = intval($studio);
1298
1299     $return = array();
1300
1301     $result = mysql::query("SELECT image.id as image_id, image, image_date WHERE image_date.id=image.day_id AND image_date.shot_date = '$date' AND image.enabled=1 AND image.studio = '$studio'");
1302     while($img = mysql::fetch($result)) {
1303         $img->copyright = metadata::get_copyright($img->image_id);
1304         $img->models = metadata::get_models($img->image_id);
1305         $return[$img->image_id] = $img;
1306     }
1307 }
1308
1309 1273
1310 1274
1311 1275
```

Aula 3
Henrique Gurgacz de Almeida

Explorando a Ferramenta GitHub

Módulo 3

- Versionamento de Código
- Pipelines no GitHub



Versionamento de Código

Unidade 3.1



O GitHub



- Criado em 2008
- Microsoft adquiriu o GitHub por US\$ 7,5milhões
- Possui 36 milhões de usuários
- Possui 100 milhões de projetos
- Possui licença gratuita
- Possui Códigos abertos (públicos) e códigos fechados (privados)



O GitHub

- Armazena, Gere e Versiona inclusive arquivos comuns
- Colaboração e Comunicação
- Documentação e Visibilidade (Wikis, Views, Markdown, etc)
- Páginas de empresas e desenvolvedores (Redes Sociais)
- Integração Com IA Generativa para criar código (CoPilot)
- Actions -> Para CI e CD
- Segurança: Code Scanning



Features of GitHub



- **GitHub Codespaces**
- **GitHub Advanced Code Search**
- **GitHub Actions**
- **GitHub Pages**
- **GitHub CoPilot**
- **GitHub Marketplace**
- **GitHub Pull Requests**
- **GitHub Discussions**
- **GitHub CLI**
- **Security Alerts**
- **Native Mobile Apps**

O GitHub



- Projetos famosos hospedados no GitHub:
 - <https://github.com/WordPress>
 - <https://github.com/mysql>
 - <https://github.com/grafana>
- Pessoas Famosas:
 - <https://github.com/addyosmani>
 - <https://github.com/tonsky>
 - <https://github.com/DanWahlin>



O GitHub

- Documentação Oficial GitHub

<https://docs.github.com/pt>



O GitHub – Cli e Desktop

<https://docs.github.com/pt/github-cli>

<https://docs.github.com/pt/desktop>



O GitHub – Clonar Repositório Público

Angular-HelloWorld Public Watch 2

main 1 Branch 0 Tags

Go to file t Add file <> Code

Dan Wahlin Update index.html

.vscode	Initial commit
src	Update index.html
.editorconfig	Initial commit
.gitignore	Initial commit
README.md	Initial commit
angular.json	Initial commit
package-lock.json	Initial commit 2 years ago
package.json	Initial commit 2 years ago
tsconfig.app.json	Initial commit 2 years ago
tsconfig.json	Initial commit 2 years ago
tsconfig.spec.json	Initial commit 2 years ago

Local Codespaces

Clone ?

HTTPS SSH **GitHub CLI**

<https://github.com/DanWahlin/Angular-HelloWorld>

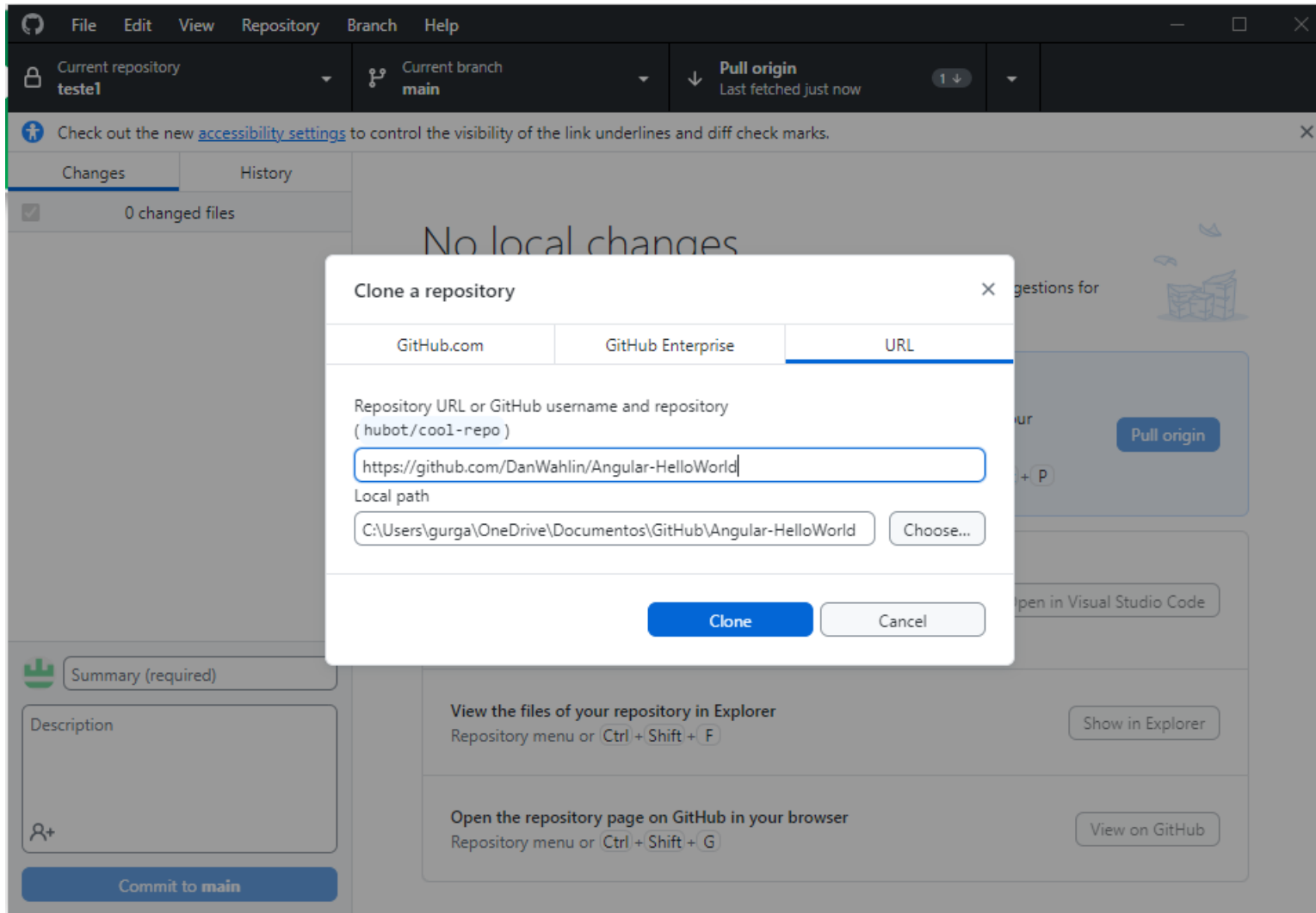
Clone using the web URL.

Open with GitHub Desktop

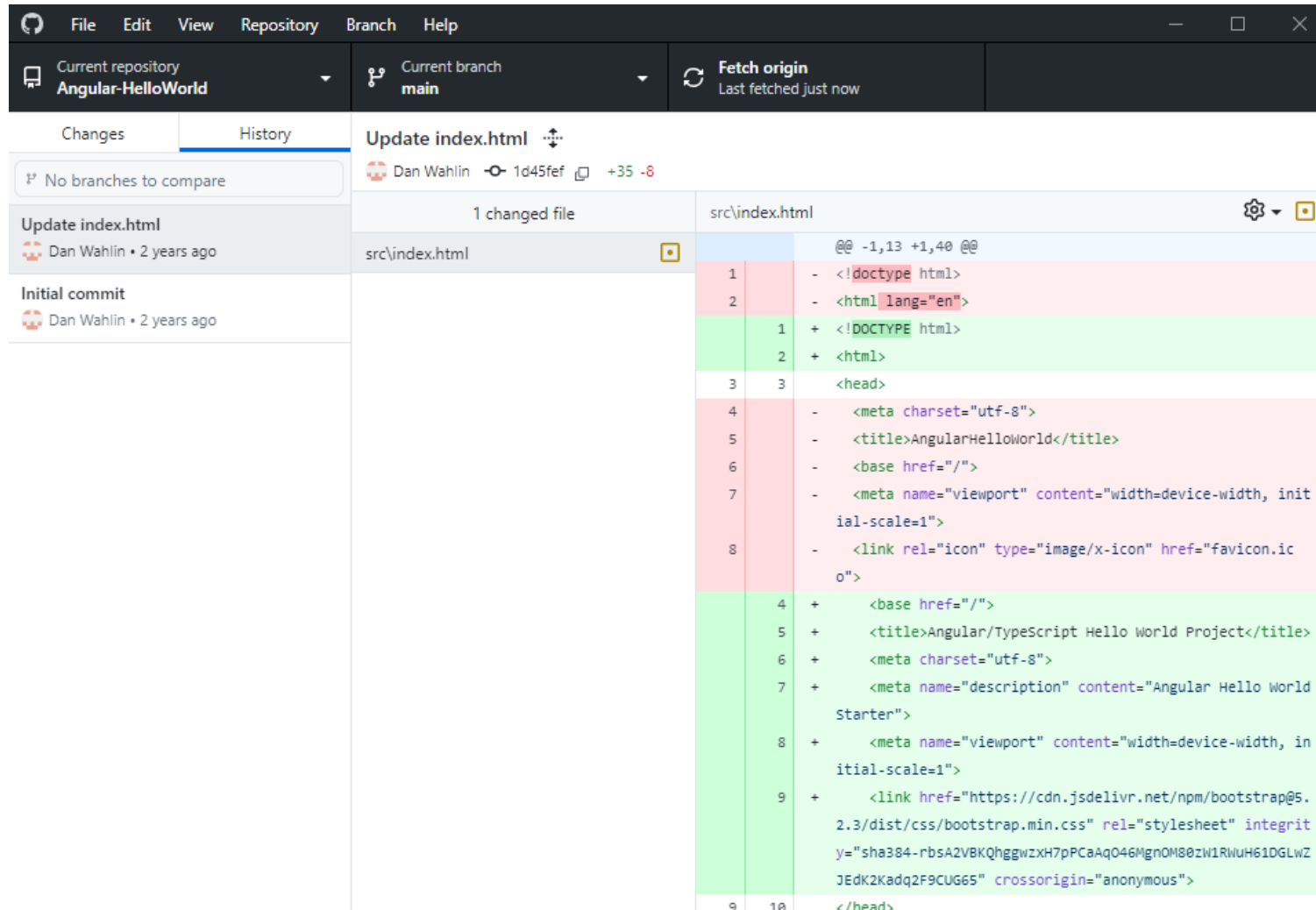
Download ZIP



O GitHub – Clonar Repositório Público



O GitHub – Clonar Repositório Público



Current repository: Angular-HelloWorld | Current branch: main | Fetch origin: Last fetched just now

Changes | History

No branches to compare

Update index.html
Dan Wahlin • 2 years ago

Initial commit
Dan Wahlin • 2 years ago

Update index.html
Dan Wahlin • 1d45fef • +35 -8

1 changed file: src/index.html

```
@@ -1,13 +1,40 @@
- <!doctype html>
- <html lang="en">
+ <!DOCTYPE html>
+ <html>
3   <head>
4     <meta charset="utf-8">
5     <title>AngularHelloWorld</title>
6     <base href="/">
7     <meta name="viewport" content="width=device-width, initial-scale=1">
8     <link rel="icon" type="image/x-icon" href="favicon.ico">
+   <base href="/">
+   <title>Angular/TypeScript Hello World Project</title>
+   <meta charset="utf-8">
+   <meta name="description" content="Angular Hello World Starter">
+   <meta name="viewport" content="width=device-width, initial-scale=1">
+   <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLWZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
9   </head>
```

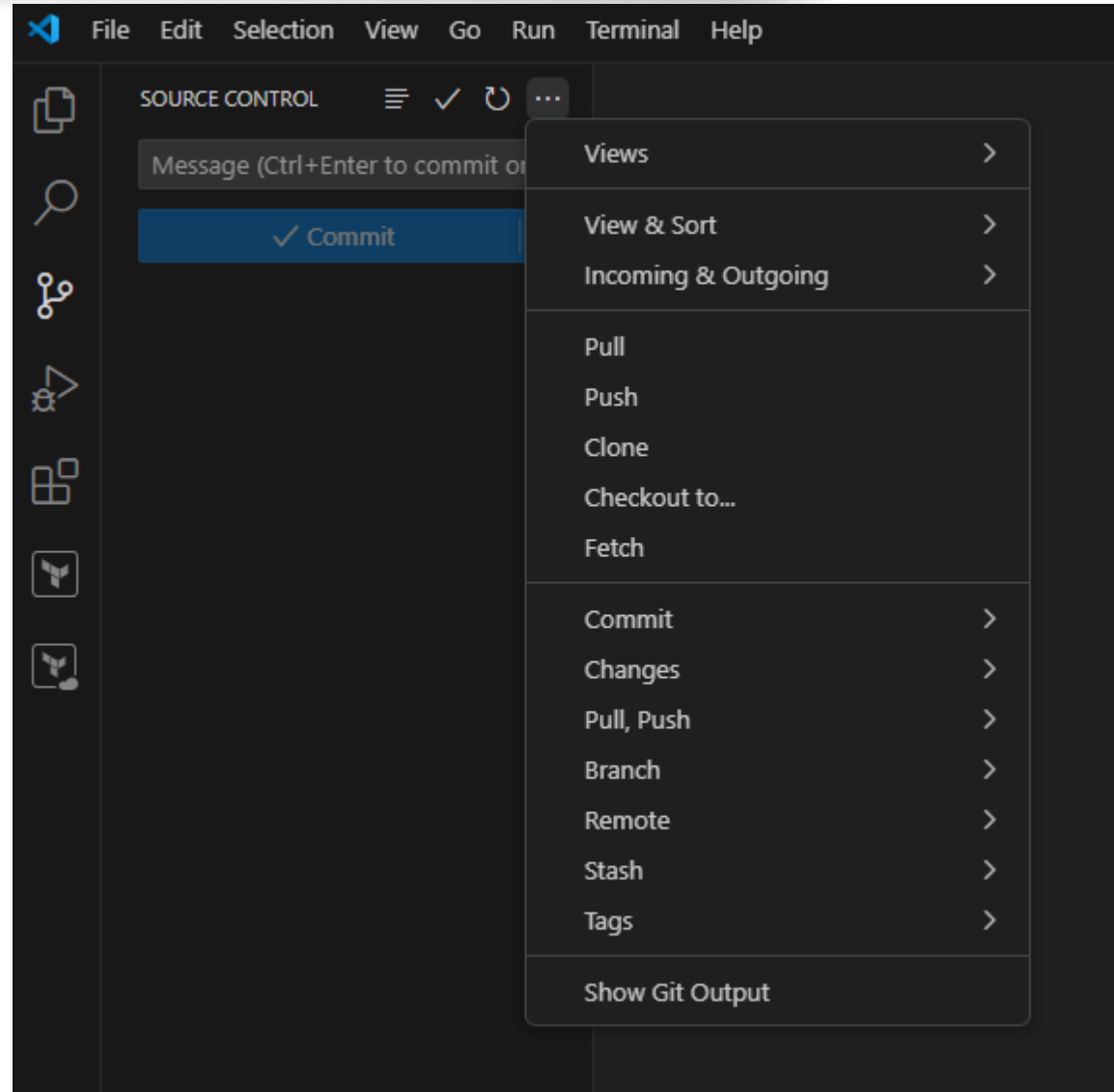


O GitHub – Passo a Passo

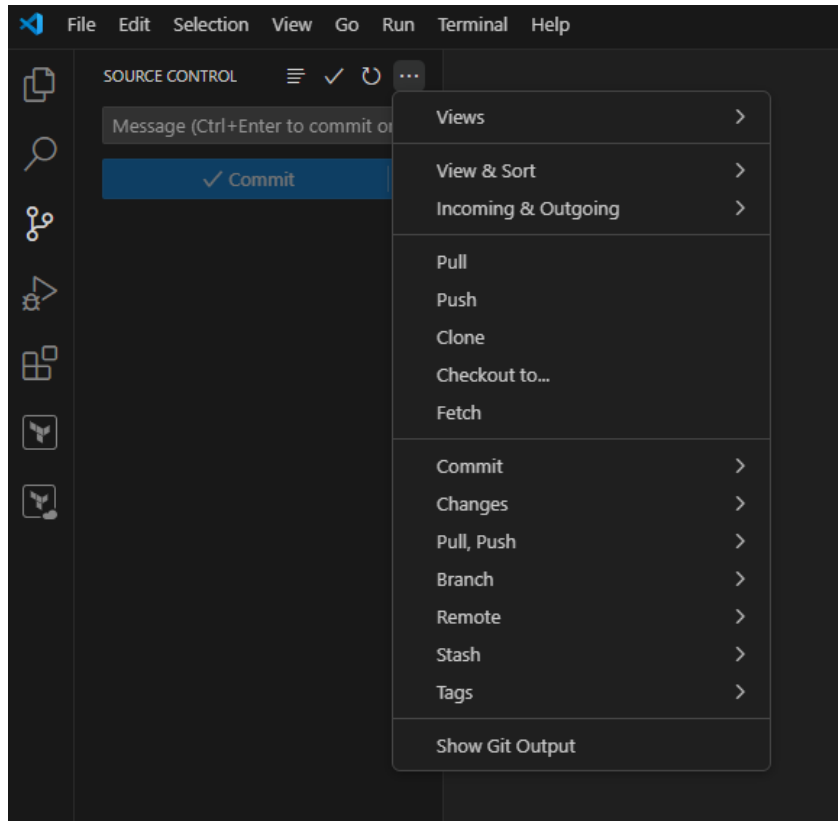
- Criar uma Conta
- Criar um Repositório
- Readme e a Linguagem Markdown
<https://www.markdownguide.org/basic-syntax/>



O GitHub – No VsCode



O GitHub – No VsCode



- **Clone:** Copia o repositório completo para o computador local
- **Commit:** Salva as alterações até o momento no repositório local. Permite voltar a esse ponto (antes do salvamento) caso necessário
- **Push:** Envia as alterações para a nuvem
- **Pull:** Baixa as atualizações de um repositório



O GitHub – No VsCode

- Executando Commit, Push e Pull no GitHub Usando VSCode



O GitHub – No VsCode

- **Resolvendo Conflitos no GitHub com VsCode**



O GitHub – Branches

- Criar um Branch
- Usar um Branch



Pipelines no GitHub

Unidade 3.2



Exemplos de Pipelines

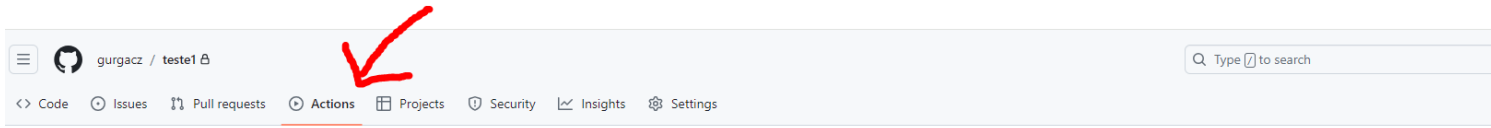


GitHub Actions

- Realizar o Build ou compilação do código para confirmar que não possui erros de escrita de código;
- Realizar uma bateria de testes unitários para identificar se não possui erros funcionais no sistema ;
- Realizar uma análise de padrões e boas práticas, tais como: nome de variáveis, quantidade de comentários, uso indevido de operações ou comandos;
- Realizar a execução do deploy do sistema em um ambiente qualquer;
- Enviar e-mails e alertas de operações;
- Solicitar aprovação de ações realizadas.



GitHub Actions



gurgacz / teste1

<> Code Issues Pull requests **Actions** Projects Security Insights Settings

Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and [set up a workflow yourself](#) →

Search workflows

Suggested for this repository

Jekyll using Docker image

By GitHub Actions

Package a Jekyll site using the jekyll/builder Docker image.

Configure

HTML

Deployment [View all](#)

Deploy Node.js to Azure Web App

By Microsoft Azure

Build a Node.js project and deploy it to an Azure Web App.

Configure

Deployment

Deploy to Amazon ECS

By Amazon Web Services

Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.

Configure

Deployment

Build and Deploy to GKE

By Google Cloud

Build a docker container, publish it to Google Container Registry, and deploy to GKE.

Configure

Deployment

Terraform

By HashiCorp

Set up Terraform CLI in your GitHub Actions workflow.

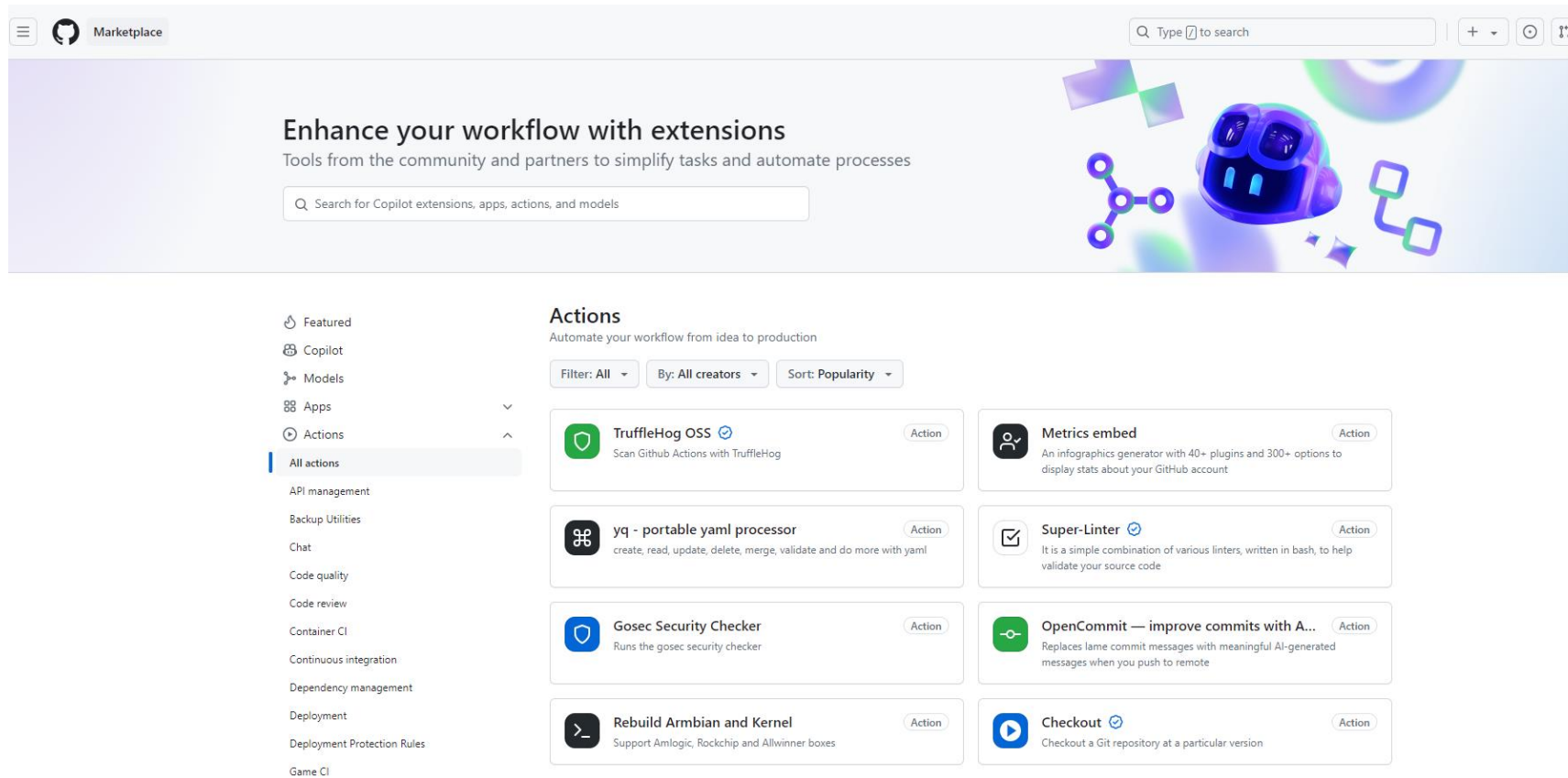
Configure

Deployment



GitHub Actions - Marketplace

<https://github.com/marketplace?type=actions>



The screenshot displays the GitHub Marketplace interface for Actions. At the top, a navigation bar includes the GitHub logo, the word "Marketplace", a search bar with the placeholder "Type to search", and icons for adding, refreshing, and settings. Below the navigation bar is a hero section with the heading "Enhance your workflow with extensions" and the subtext "Tools from the community and partners to simplify tasks and automate processes". A search bar for "Copilot extensions, apps, actions, and models" is also present. To the right of the hero section is a colorful illustration of a robot head and various geometric shapes. On the left side, a sidebar lists categories: Featured, Copilot, Models, Apps, and Actions (which is highlighted). Under the Actions category, a list of sub-categories is shown: API management, Backup Utilities, Chat, Code quality, Code review, Container CI, Continuous integration, Dependency management, Deployment, Deployment Protection Rules, and Game CI. The main content area is titled "Actions" with the subtitle "Automate your workflow from idea to production". It features filters for "Filter: All", "By: All creators", and "Sort: Popularity". Below the filters, a grid of action cards is displayed, each with an icon, name, description, and an "Action" button. The actions shown are: TruffleHog OSS (Scan GitHub Actions with TruffleHog), Metrics embed (An infographics generator with 40+ plugins and 300+ options to display stats about your GitHub account), yq - portable yaml processor (create, read, update, delete, merge, validate and do more with yaml), Super-Linter (It is a simple combination of various linters, written in bash, to help validate your source code), Gosec Security Checker (Runs the gosec security checker), OpenCommit — improve commits with A... (Replaces lame commit messages with meaningful AI-generated messages when you push to remote), Rebuild Armbian and Kernel (Support Amlogic, Rockchip and Allwinner boxes), and Checkout (Checkout a Git repository at a particular version).

Marketplace

Enhance your workflow with extensions
Tools from the community and partners to simplify tasks and automate processes

Search for Copilot extensions, apps, actions, and models

Featured
Copilot
Models
Apps
Actions
All actions
API management
Backup Utilities
Chat
Code quality
Code review
Container CI
Continuous integration
Dependency management
Deployment
Deployment Protection Rules
Game CI

Actions
Automate your workflow from idea to production

Filter: All By: All creators Sort: Popularity

TruffleHog OSS Action
Scan GitHub Actions with TruffleHog

Metrics embed Action
An infographics generator with 40+ plugins and 300+ options to display stats about your GitHub account

yq - portable yaml processor Action
create, read, update, delete, merge, validate and do more with yaml

Super-Linter Action
It is a simple combination of various linters, written in bash, to help validate your source code

Gosec Security Checker Action
Runs the gosec security checker

OpenCommit — improve commits with A... Action
Replaces lame commit messages with meaningful AI-generated messages when you push to remote

Rebuild Armbian and Kernel Action
Support Amlogic, Rockchip and Allwinner boxes

Checkout Action
Checkout a Git repository at a particular version



GitHub Actions - YAML

- GitHub Actions usa a sintaxe YAML para escrever os Pipelines
- Cada pipeline é armazenado como um arquivo com extensão yml e deve ficar salvo no diretório chamado .github/workflows
- O cabeçalho base de qualquer pipeline escrito em YAML no GitHub é:

```
name: nome_do_pipeline
```

```
on: [xxxxx]
```

```
jobs:
```



GitHub Actions – Exemplo de um Pipeline em YAML

```
name: GitHub Pipeline de Exemplo
run-name: Teste de GitHub Actions
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "Job iniciado automaticamente"
      - run: echo "Job executando no ${ runner.os } server hospedado pelo GitHub!"
      - run: echo "Branch ${ github.ref } e repositório ${ github.repository }."
      - name: Copiando código para o ambiente do GitHub Actions
        uses: actions/checkout@v4
      - run: echo "O repositório ${ github.repository } foi clonado no servidor do Github."
      - name: Listando arquivos copiados para o ambiente do GitHub Actions
        run: |
          ls ${ github.workspace }
      - run: echo "Situação do Job ${ job.status }."
      - run: echo "Pipeline Encerrado"
```



GitHub Actions – Exemplo de um Pipeline em YAML

```
name: GitHub Pipeline de Exemplo
run-name: Teste de GitHub Actions
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "Job iniciado automaticamente"
      - run: echo "Job executando no ${ runner.os } server hospedado pelo GitHub!"
      - run: echo "Branch ${ github.ref } e repositório ${ github.repository }."
      - name: Copiando código para o ambiente do GitHub Actions
        uses: actions/checkout@v4
      - run: echo "O repositório ${ github.repository } foi clonado no servidor do Github."
      - name: Listando arquivos copiados para o ambiente do GitHub Actions
        run: |
          ls ${ github.workspace }
      - run: echo "Situação do Job ${ job.status }."
      - run: echo "Pipeline Encerrado"
```

Nome do Pipeline

Quando será executado.

Pull Request
Push
Commit
Create

Onde o pipeline será executado



GitHub Actions – Exemplo de um Pipeline em YAML

- Criar um exemplo do Pipeline com o código anterior
- Mostrar o resultado após o Push

```
name: GitHub Pipeline de Exemplo
run-name: Teste de GitHub Actions
on: [push]
jobs:
  Explore-GitHub-Actions:
    runs-on: ubuntu-latest
    steps:
      - run: echo "Job iniciado automaticamente"
      - run: echo "Job executando no ${runner.os} server hospedado pelo GitHub!"
      - run: echo "Branch ${github.ref} e repositório ${github.repository}."
      - name: Copiando código para o ambiente do GitHub Actions
        uses: actions/checkout@v4
      - run: echo "O repositório ${github.repository} foi clonado no servidor do Github."
      - name: Listando arquivos copiados para o ambiente do GitHub Actions
        run: |
          ls ${github.workspace}
      - run: echo "Situação do Job ${job.status}."
      - run: echo "Pipeline Encerrado"
```



GitHub Actions – Exemplo de um Pipeline Real

```
name: Pipeline Teste
on
  pull_request:
    branches:
      - main
jobs:
  Tarefa1:
    runs-on: ubuntu-latest
    steps:
      - name: Passo1-Puxar última versão do código
        uses: actions/checkout@v2
      - name: Passo2-Instalar Node.JS
        uses: actions/setup-node@v1
        with:
          node-version: "12.x"
      - name: Passo3-Instalar Dependencias
        run: npm ci
      - name: Passo4-Rodar os testes
        run: npm run test:ci
```



Dúvidas???

