



lucianoedipo / react-lgpd-consent



Code

Issues 26

Pull requests 1

Discussions

Actions

Projects 1

Conversation 9

Commits 6

Checks 3

Files changed 24



lucianoedipo commented 11 minutes ago

Owner

Adiciona RECIPES.md com 8 receitas práticas, WORKFLOWS.md documentando CI/CD completo, novos workflows de versionamento/publicação/docs, correção de versão do TypeDoc e atualização da documentação em inglês.

Closes #62



1

mention @copilot in a comment to make changes to this pull request.

lucianoedipo added 5 commits 1 hour ago

docs: adicionar receitas e referências práticas na documentação 1792a9d

feat: add CI/CD workflows for versioning, publishing, and documentation ... 72ed30c

docs: atualizar documentação com receitas práticas e links para workf... 9aeedb3

test: corrigir espaçamento no teste de comportamento do ConsentScript... fcd5394

docs: adicionar RECIPES.md com 8 receitas práticas e WORKFLOWS.md doc... bddfa12

lucianoedipo requested a review from Copilot 11 minutes ago

lucianoedipo self-assigned this 11 minutes ago

lucianoedipo linked an issue 11 minutes ago that may be closed by this pull request

⌚ Copilot started reviewing on behalf of **lucianoedipo** [10 minutes ago](#)

[View session](#)

🕒 Copilot finished reviewing on behalf of **lucianoedipo** [8 minutes ago](#)

👤  **lucianoedipo** [chore: atualizar lockfile e adicionar changeset para documentação e C...](#) [...](#) ● [80e1cf9](#)

👁️ Copilot AI reviewed [6 minutes ago](#)

[View reviewed changes](#)

👤  **Copilot** AI left a comment

[Contributor](#)

## Pull request overview

This PR adds comprehensive documentation and restructures the CI/CD workflows to improve the release process. It introduces practical recipes for common use cases, documents the complete CI/CD pipeline, and splits the previous monolithic release workflow into three focused workflows for versioning, publishing, and documentation deployment.

## Key Changes

- Added **RECIPES.md** with 8 practical recipes covering Next.js App Router, CSP/nonce, Google Consent Mode v2, subdomains, and more
- Added **WORKFLOWS.md** documenting the complete CI/CD pipeline with detailed explanations
- Split release workflow into three separate workflows: `version.yml` (version bumping and tagging), `publish.yml` (NPM publishing), and `docs.yml` (documentation deployment)
- Created `sync-versions.mjs` script to automatically synchronize package versions
- Updated `run-typedoc.mjs` to inject version numbers dynamically into TypeDoc output
- Updated documentation references across README, QUICKSTART, and CONFORMIDADE files

## Reviewed changes

Copilot reviewed 21 out of 24 changed files in this pull request and generated 7 comments.

- ▶ Show a summary per file
- ▶ Files not reviewed (1)
- ▶ Comments suppressed due to low confidence (1)

You can also share your feedback on Copilot code review for a chance to win a \$100 gift card. [Take the survey.](#)

 Implement suggestion



scripts/sync-versions.mjs

```
48 + }
49 +
50 + function parseVersion(version) {
51 +   const [major, minor, patch] = version.split('.').map(Number)
```



Copilot AI [6 minutes ago](#)

The `parseVersion` function doesn't handle invalid version strings (e.g., missing parts, non-numeric values, or versions with pre-release tags like "1.0.0-beta"). This could cause the script to crash with `Nan` values if it encounters an unexpected version format.

Consider adding validation:

```
function parseVersion(version) {
  const parts = version.split('.')
  if (parts.length !== 3) {
    throw new Error(`Invalid version format: ${version}. Expected format: x.y.z`)
  }
  const [major, minor, patch] = parts.map(Number)
  if (isNaN(major) || isNaN(minor) || isNaN(patch)) {
    throw new Error(`Invalid version format: ${version}. Parts must be numeric.`)
  }
  return { major, minor, patch }
}
```



Suggested change

```
51 - const [major, minor, patch] = version.split('.').map(Number)
51 + // Aceita apenas formato x.y.z onde x, y, z são inteiros
52 + // Rejeita pre-release/build tags (ex: 1.0.0-beta, 1.0.0+build)
53 + if (typeof version !== 'string') {
54 +   throw new Error(`Formato de versão inválido: ${version}. Esperado
      string no formato x.y.z`)
55 +
56 + // Regex: três grupos de dígitos separados por ponto, sem extras
57 + const match = version.match(/^(\d+)\.(\d+)\.(\d+)/)
58 + if (!match) {
59 +   throw new Error(`Formato de versão inválido: ${version}. Esperado x.y.z
      (sem pre-release ou build)`)
60 + }
```

```
61 +     const major = Number(match[1])
62 +     const minor = Number(match[2])
63 +     const patch = Number(match[3])
64 +     if ([major, minor, patch].some(n => isNaN(n))) {
65 +       throw new Error(`Formato de versão inválido: ${version}. Partes devem
66 +       ser numéricas.`)
```

Commit suggestion ▾

Add suggestion to batch



Copilot uses AI. Check for mistakes.



Reply...

Resolve conversation

.github/workflows/version.yml

Comment on lines +53 to +63

```
53 +     - name: Check if version PR was merged
54 +       id: check-merge
55 +       run: |
56 +         # Verifica se estamos na main após merge de PR de version
57 +         if git log -1 --pretty=%B | grep -q "chore(release): version packages"
58 +           echo "merged=true" >> $GITHUB_OUTPUT
59 +           echo "✅ PR de version foi mergeado, criando tag..."
60 +         else
61 +           echo "merged=false" >> $GITHUB_OUTPUT
62 +           echo "ℹ️ Aguardando merge do PR de version"
63 +         fi
```



Copilot AI 6 minutes ago

The workflow logic for detecting a merged version PR relies on checking the last commit message for "chore(release): version packages". However, this approach has a potential race condition: if the workflow runs on any push to main (not just after the PR merge), it might create tags prematurely or miss them.

The changesets action already provides outputs ( `hasChangesets` , `published` , `publishedPackages` ) that should be used to determine workflow state. Consider using these outputs instead of parsing git log messages. Additionally, the current logic will run the tag creation step on every push to main where the last commit message matches, which could lead to duplicate runs.

Consider restructuring to separate the "create PR" workflow from the "create tag after merge" workflow, or use the changesets action outputs more effectively.

Cloud Implement suggestion



Copilot uses AI. Check for mistakes.



Reply...

Resolve conversation

RECIPES.md

```
54 + 'use client'  
55 +  
56 + import { ConsentProvider, ConsentScriptLoader,  
           createGoogleAnalyticsIntegration, createGoogleTagManagerIntegration } from  
           'react-lgpd-consent'  
57 + import { useEffect } from 'react'
```



Copilot AI 6 minutes ago

The example code imports `useEffect` from 'react' but never uses it. This unused import should be removed to avoid confusion.

Remove line 57:

```
import { useEffect } from 'react'
```



Suggested change

```
57 - import { useEffect } from 'react'
```

Commit suggestion ▾

Add suggestion to batch



Copilot uses AI. Check for mistakes.



Reply...

Resolve conversation

RECIPES.md

```
438 + return (  
439 +   <html lang="pt-BR">  
440 +     <head>  
441 +       <meta httpEquiv="Content-Security-Policy" content={`script-src 'self'  
           'nonce-${nonce}`}` />
```



Copilot AI 6 minutes ago

The CSP implementation has a potential security issue: setting the CSP header via a `<meta>` tag in the body (line 441) is less secure and has limited support compared to setting it as an HTTP header. The meta tag CSP also overrides/conflicts with the header CSP set in the middleware (line 418-419), which could lead to unexpected behavior.

The CSP should only be set via HTTP headers in the middleware. Remove the meta tag from the layout:

```
// Remove this line:  
<meta httpEquiv="Content-Security-Policy" content={`script-src 'self' 'nonce-${nonce}`}`>
```

The middleware already sets the proper CSP header, so the meta tag is redundant and could cause conflicts.

Suggested change

441 -     ~~<meta httpEquiv="Content-Security-Policy" content={`script-src 'self' 'nonce-\${nonce}`}`>~~

[Commit suggestion](#)

[Add suggestion to batch](#)



Copilot uses AI. Check for mistakes.



Reply...

[Resolve conversation](#)

WORKFLOWS.md

Comment on lines +74 to +85

```
74 + // IMPORTANTE: Compara com git HEAD para detectar mudanças  
75 + function detectBumps() {  
76 +   const coreBump = compareWithGit('packages/core/package.json')  
77 +   const muiBump = compareWithGit('packages/mui/package.json')  
78 +  
79 +   // Aplicar o maior tipo de bump ao agregador  
80 +   if (coreBump === 'major' || muiBump === 'major') return 'major'  
81 +   if (coreBump === 'minor' || muiBump === 'minor') return 'minor'  
82 +   if (coreBump === 'patch' || muiBump === 'patch') return 'patch'  
83 +   return null  
84 + }  
85 + ````
```



Copilot [AI](#) 6 minutes ago

The documentation shows a `compareWithGit` function in the example code (line 76-77), but this function doesn't exist in the actual `sync-versions.mjs` script. The actual implementation uses `getGitVersion` and `getBumpType` functions instead.

Update the example to reflect the actual implementation:

```
// IMPORTANTE: Compara com git HEAD para detectar mudanças
function detectBumps() {
  const coreOldVersion = getGitVersion('packages/core/package.json')
  const muiOldVersion = getGitVersion('packages/mui/package.json')
  const coreBump = getBumpType(coreOldVersion, coreVersion)
  const muiBump = getBumpType(muiOldVersion, muiVersion)

  // Aplicar o maior tipo de bump ao agregador
  if (coreBump === 'major' || muiBump === 'major') return 'major'
  if (coreBump === 'minor' || muiBump === 'minor') return 'minor'
  if (coreBump === 'patch' || muiBump === 'patch') return 'patch'
  return null
}
```



Implement suggestion

Copilot uses AI. Check for mistakes.



Reply...

Resolve conversation

scripts/sync-versions.mjs

Comment on lines +55 to +62

```
55  + function compareVersions(v1, v2) {
56  +   const parsed1 = parseVersion(v1)
57  +   const parsed2 = parseVersion(v2)
58  +
59  +   if (parsed1.major !== parsed2.major) return parsed1.major - parsed2.major
60  +   if (parsed1.minor !== parsed2.minor) return parsed1.minor - parsed2.minor
61  +   return parsed1.patch - parsed2.patch
62  + }
```



Copilot AI 6 minutes ago

The `compareVersions` function is defined but never used in the script. Consider removing it to avoid dead code, or if it's intended for future use, add a comment explaining its purpose.

Suggested change

```
55  - function compareVersions(v1, v2) {
56  -   const parsed1 = parseVersion(v1)
57  -   const parsed2 = parseVersion(v2)
58  -
59  -   if (parsed1.major !== parsed2.major) return parsed1.major - parsed2.major
60  -   if (parsed1.minor !== parsed2.minor) return parsed1.minor - parsed2.minor
61  -   return parsed1.patch - parsed2.patch
```

62 - }

Commit suggestion ▾

Add suggestion to batch



Copilot uses AI. Check for mistakes.



Reply...

Resolve conversation

scripts/run-typedoc.mjs

Comment on lines +33 to +39

```
33 + // Limpar arquivo temporário
34 + try {
35 +   const fs = await import('node:fs/promises')
36 +   await fs.unlink(tempConfig)
37 + } catch {
38 +   // Ignorar erro se não conseguir deletar
39 + }
```



Copilot AI 6 minutes ago

Potential issue with async/await mixing: The cleanup code uses `await import('node:fs/promises')` but this is inside a try-catch at the top level after a sync operation. Since the main logic has already completed and we're just cleaning up, consider using synchronous `unlinkSync` from the already imported `fs` module for consistency with the rest of the file's synchronous file operations, or handle the promise properly.

Example:

```
import { readFileSync, writeFileSync, unlinkSync } from 'node:fs'

// Later, in cleanup:
try {
  unlinkSync(tempConfig)
} catch {
  // Ignorar erro se não conseguir deletar
}
```



Cloud Implement suggestion



Copilot uses AI. Check for mistakes.



Reply...

Resolve conversation



github-advanced-security found potential problems 4 minutes ago

[View reviewed changes](#)

scripts/sync-versions.mjs

```
39 | + function getGitVersion(filePath) {  
40 | +   try {  
41 | +     const relativePath = path.relative(rootDir, filePath)  
42 | +     const content = execSync(`git show HEAD:${relativePath}`, { encoding: 'utf-8' })  
| +   } catch (err) {  
| +     console.error(`Error getting git version for ${filePath}: ${err.message}`)  
| +   }  
| + }  
| + module.exports = getGitVersion
```

Check warning

[Code scanning / CodeQL](#)

**Shell command built from environment values** Medium

This shell command depends on an uncontrolled [absolute path](#).

[Show more details](#)

[Show paths](#)

[Dismiss alert](#)

Open

62 docs receitas #115

lucianoedipo wants to merge 6 commits into [main](#) from [62-docs-receitas](#)

path, especially one from the environment or filesystem) directly into a shell command string. Instead, leverage the `execFileSync` (or `spawnSync`) API, which lets each argument be passed as a discrete element and doesn't invoke a shell.

For the command `git show HEAD:${relativePath}`, the right approach is to use `execFileSync` (from `node:child_process`) as `execFileSync('git', ['show', 'HEAD:' + relativePath])`. This passes the arguments to git directly, with no shell interpretation, making it robust to spaces, etc.

**Required changes:**

- In `getGitVersion`, replace `execSync` with `execFileSync`, separating the command and its arguments.
- Import `execFileSync` instead of or in addition to `execSync` (preferably use only one).
- Ensure the encoding option is preserved (as `execFileSync` accepts it).
- No additional packages or helpers are needed.

---

Suggested changeset 1



scripts/sync-versions.mjs

```
... ... @@ -15,7 +15,7 @@
15 15     * Precisamos comparar com estado anterior (via git) para detectar
       bumps.
16 16     */
17 17
18 - import { execSync } from 'node:child_process'
18 + import { execFileSync } from 'node:child_process'
19 19 import fs from 'node:fs/promises'
20 20 import path from 'node:path'
21 21 import { fileURLToPath } from 'node:url'
...
... @@ -39,7 +39,7 @@
39 39 function getGitVersion(filePath) {
40 40     try {
41 41         const relativePath = path.relative(rootDir, filePath)
42 -     const content = execSync(`git show HEAD:${relativePath}`, {
43 -         encoding: 'utf-8' })
42 +     const content = execFileSync('git', ['show',
43 -         `HEAD:${relativePath}`], { encoding: 'utf-8' })
43 43         return JSON.parse(content).version
44 44     } catch {
45 45         // Se não houver versão anterior (novo pacote), retorna 0.0.0
```

Copilot is powered by AI and may make mistakes. Always verify output.

 Edit

Commit suggestion ▾



Reply...

#### Reviewers

 Copilot



At least 0 approving reviews are required to merge this pull request.

Still in progress? [Convert to draft](#)

#### Assignees

 lucianoedipo



#### Labels

None yet



#### Projects

None yet



#### Milestone



No milestone

---

Development



Successfully merging this pull request may close these issues.

 [Docs] Receitas: Next, subdomínios, Consent Mode, versão, CSP

---

---

2 participants

---

