

TheCodingMachine

# La revue de PR

Janvier 2023

Revue de PR : comment la rendre efficace et pertinente ?



# SOMMAIRE

1:// LEXIQUE

2:// CYCLE DE VIE

3:// COMMENT EFFECTUER SA REVUE

4:// CONCLUSION

01

LEXIQUE



## Lexique (1/2)

- **PR :**
  - Représentation d'une **branche souhaitant être mergée dans une autre branche**
  - Affichage visuel des différences entre deux branches
  - Nomenclature instanciée par les outils de gestion de Git
  - Pull Request (GitLab)
  - Équivalent Merge Request (GitHub, Bitbucket)
- **Assignee :**
  - Personne responsable de la création de la PR
  - En général, auteur de la branche
- **Reviewer :**
  - Personne responsable de relire la PR, différente de l'assignee
  - Peut-être responsable du merge également, mais ce n'est pas un obligation

## Lexique (2/2)

- **Revue :**
  - Lecture du code par une personne différente de l'auteur
  - Identification des risques liés à intégrer une branche dans une autre
- **Branche source :**
  - Branche qui demande à être mergée dans une autre
- **Branche cible :**
  - Branche qui va accueillir une autre branche
- **Pipeline :**
  - Ensemble d'opérations qui visent à automatiser des processus
  - Notamment utilisée pour effectuer des vérifications syntaxiques, des tests, des builds et des déploiements
  - Peut être exécutée lors de changements apportés sur la branche source d'une PR

02

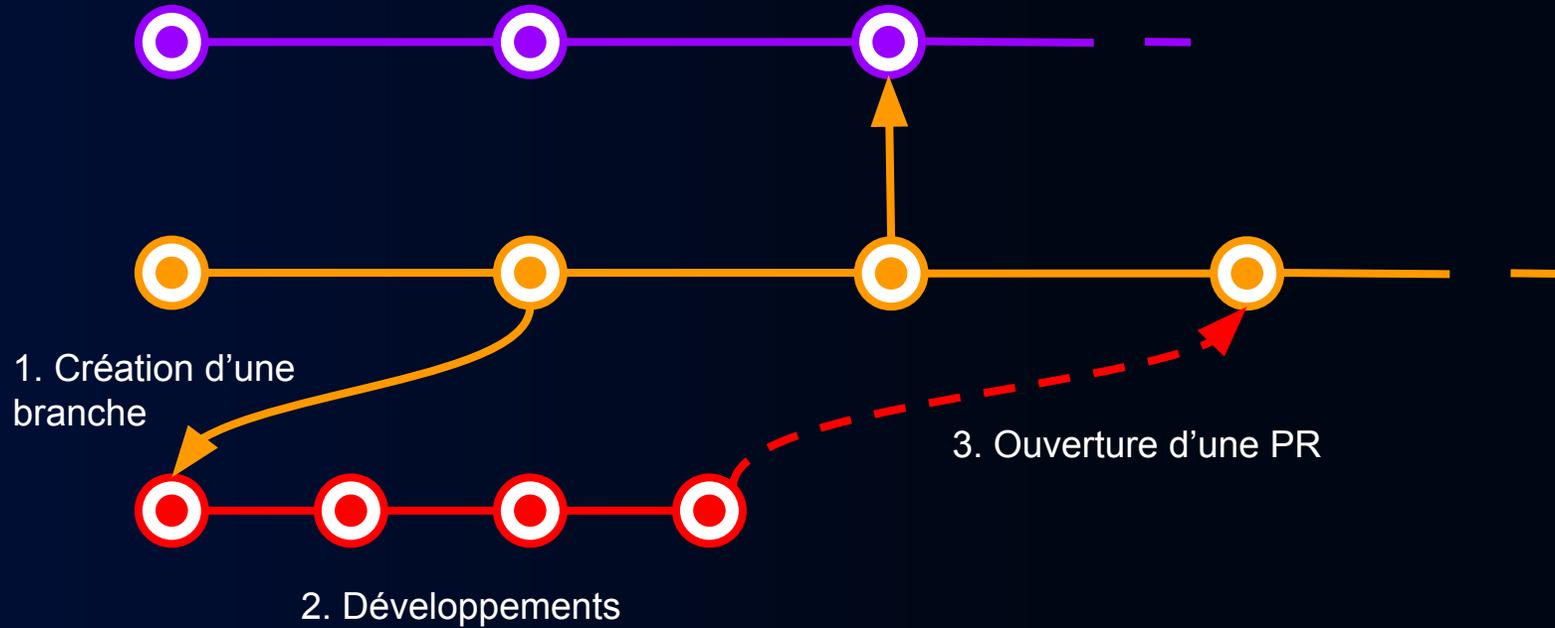
CYCLE DE VIE



main

develop

feature



## 1. Créer une branche locale

- **Choisir avec soin la branche initiale** d'où est tirée cette nouvelle branche
  - Depuis **develop** pour une fonctionnalité
  - Depuis **master** pour un hotfix
- Etre **vigilant** sur la création d'une **branche depuis une autre branche non mergée**
  - Éviter de créer un sac de noeud
  - En cas de nécessité :
    - **Alerter le reviewer** et demander la revue de la branche initiale
    - Ouvrir la PR **entre la branche initiale et la nouvelle branche**
    - Ajouter une description sur la PR pour décrire le comportement et le possible ordre de merge

## 2. Développer la fonctionnalité ou le correctif

- **Committer régulièrement** son code
  - Permettra de **faciliter** le travail du reviewer pour comparer des versions de commit si nécessaire
  - Facilitera les potentiels revert

### 3. Ouvrir la PR (1/2)

- **Push la branche** sur le dépôt
  - Le **CLI** peut directement **afficher un lien d'ouverture** (ou consultation) de la PR associée
- **Ouvrir la PR** sur GitLab
  - **Bien choisir la branche cible**, selon la branche initiale depuis laquelle provient la branche source
  - Il est possible d'ouvrir une PR même si le développement n'est pas terminé, avec le statut "**Draft**"
- Choisir **un nom de PR pertinent**
  - Le nom par défaut peut être parfois confusant
- Ajouter des **commentaires nécessaires**
  - Préciser de potentiels manquements
  - Indiquer des doutes
  - Donner une marche à suivre de relecture

### 3. Ouvrir la PR (2/2)

- Vérifier la **validité de la pipeline**
  - Si une quelconque étape de la pipeline déclenchée par le push est en **échec, analyser le comportement et le corriger**
  - **Résoudre les erreurs** avant de nouveau push la branche
- **Relire son code à tête reposée**
  - S'assurer que les modifications correspondent à ce qui est attendu
- Ajouter un **assignee** et un **reviewer**
  - S'ajouter en tant qu'assignee
  - Affecter une personne tierce pour effectuer la revue et demander un approve
  - Le reviewer n'est **pas forcément une personne plus compétente techniquement**, mais qui apporte un **œil externe**
  - Optionnellement, ajouter un label "**Need Review**" (PR2D2)

## 4. Reviewer : lire la PR

- **Lire et étudier la PR qui m'est assignée** en tant que reviewer
- **Apporter des commentaires** ou des ajustements
  - En cas d'ajustement à apporter par l'assignee, ajouter un label **"Need Changes"**
  - **Refaire une revue** après les changements apportés
- **Approuver** la PR et la **merger**
  - Merger le plus rapidement possible pour **éviter d'empiler les PR**



03

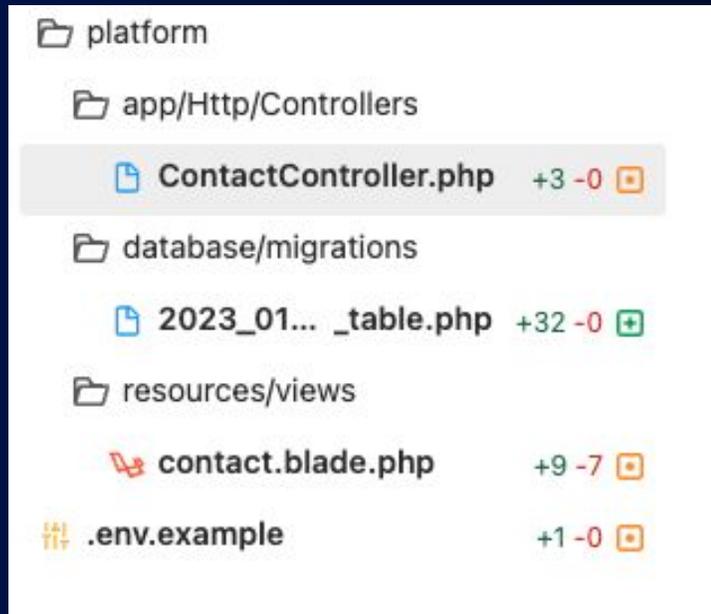
# COMMENT EFFECTUER SA REVUE

## Vérifier les métriques de base

The screenshot shows a GitHub pull request interface. At the top, there are navigation tabs: Overview (0), Commits (4), Pipelines (3), and Changes (4). The main content area displays 'PR à supprimer.' and 'Tests pour le point du vendredi'. A merge request summary shows 'Request to merge test-to-remove/pdv into develop' with a note that the source branch is 5 commits behind. A failed pipeline notification is visible: 'Pipeline #284310 failed for 02d8d86c on test-to-remove/pdv 10 minutes ago'. On the right sidebar, the assignee is Aurélien Mutin (@a.mutin) and the reviewer is Jérémy Gautrais (@JEG). The milestone is 'None' and time tracking shows 'No estimate or time spent'. A 'Delayed' label is present at the bottom of the sidebar.

- **OBJECTIF : prendre la mesure du temps à allouer à la revue**
- S'assurer que la **pipeline est au vert**
- S'assurer qu'**aucun conflit de soit présent**
- Analyser le **nombre de commits**
- Analyser le **nombre de changes**
- Vérifier la présence de **commentaires spécifiques**
- Vérifier la présence de **labels spécifiques**

## Analyser la présence de fichiers clés



- **OBJECTIF : identifier les premiers risques et ajustements sensibles**
- Fichiers contenant des variables d'environnement
  - prévoir une modification des variables du serveur, ajout dans la CI, etc.
- Impacts sur la **base de données** (migration Laravel, etc.)
  - prévoir les impacts lors du déploiement sur une base de donnée existante
- Dépendances tierces (package.json, composer.json)
  - étudier la nécessité d'ajouter un paquet
- Modification des fichiers .lock
  - s'assurer que les modifications sont réellement souhaitées (composer update trop rapide, suppression du fichier pour gestion de conflits, etc.)
- Modification de fichiers de configuration
  - s'assurer de l'impact
- Fichier liés à la configuration des containers
  - attention à la modification de volumes, ou à l'ajout de container non désiré
  - attention à la montée de version des images

# Désactiver l'affichage des différences concernant les espaces

OBJECTIF : ne pas faire de cauchemars chaque nuit après avoir lu une PR

Avec l'affichage des

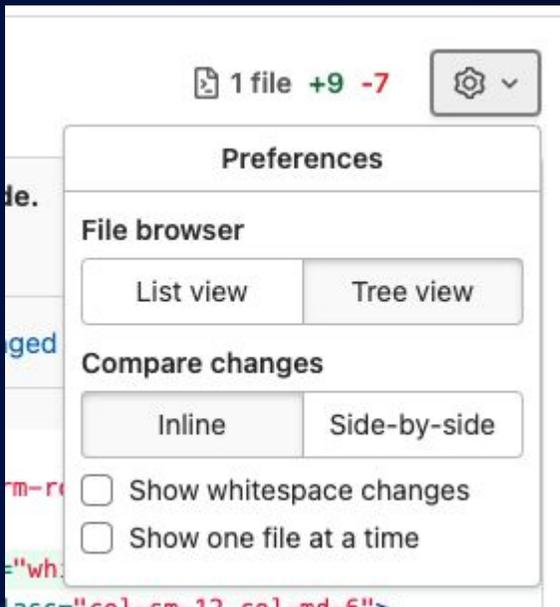
```
platform/resources/views/contact.blade.php +9 -7 Viewed
@@ -5,13 +5,15 @@
5      @csrf
6      <div class="form-row">
7
8      -      <div class="col-sm-12 col-md-6">
9      -      <x-ui.input
10     -      name="firstname"
11     -      label="user.firstname"
12     -      value="{{ old('firstname') }}"
13     -      >
14     -      </x-ui.input>
15     +      <div class="whitespace">
16     +      <div class="col-sm-12 col-md-6">
17     +      <x-ui.input
18     +      name="firstname"
19     +      label="user.firstname"
20     +      value="{{ old('firstname') }}"
21     +      >
22     +      </x-ui.input>
23     +      </div>
24     +      <div class="col-sm-12 col-md-6">
25     +      <x-ui.input
```

Sans l'affichage des

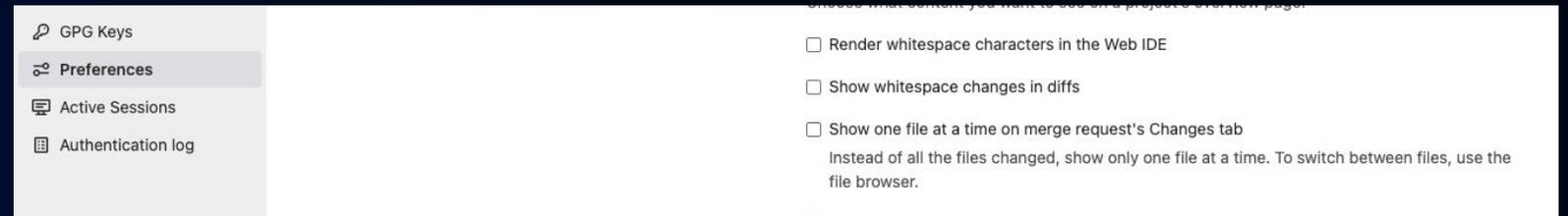
```
platform/resources/views/contact.blade.php +9 -7 Viewed
@@ -5,6 +5,7 @@
5      @csrf
6      <div class="form-row">
7
8      +      <div class="whitespace">
9      <div class="col-sm-12 col-md-6">
10     <x-ui.input
11     name="firstname"
12
13     >
14     </x-ui.input>
15     </div>
16     </div>
17     +      </div>
18     <div class="col-sm-12 col-md-6">
19     <x-ui.input
20     name="lastname"
```

## Désactiver l'affichage des différences concernant les espaces

### Depuis la PR



### Depuis ses préférences



## Analyser en première lecture le code produit

```
@@ -31,6 +32,8 @@ class ContactController extends Controller
 31   32   */
 32   33   public function send(ContactRequest $request): RedirectResponse
 33   34   {
 35   35 +   Http::get('http://localhost/testing');
 36   36 +
 34   37   Notification::route('mail', env('MAIL_FROM_ADDRESS'))->notify(new ContactMessage((object)$request));
 35   38   Notification::route('mail', $request->email)->notify(new ContactMessage((object)$request, true));
 36   39   return Redirect::to('/homepage')
```

- **OBJECTIF** : faire un premier retour (si nécessaire) et retirer une pollution de la lecture
- Vérifier la présence de comportements “hard-coded”
- S’assurer que la **roue n’ai pas été réinventée**
- Vérifier la présence d’écriture de **log non désiré** (console.log(‘coucou’))
- Vérifier la présence de **fautes d’orthographe**
- S’assurer que des **fichiers non désirés** aient été commit
- S’assurer que **tous les fichiers souhaités** aient été commit (oublie d’un add)

## Faire une revue technique détaillée

The screenshot shows a code review interface. At the top, there's a code editor with the following code:

```
@@ -306,6 +307,11 @@ class OrderController extends Controller
306 307
307 308     public function deliver(Request $request): JsonResponse
308 309     {
310 +         $order = Order::query()->find($request->input('order_id'));
```

Below the code editor, there's a comment box with the text: "On pourrait ici préférer l'utilisation de `Order::query()->findOrFail` qui lancera lui même une `ModelNotFoundException` si le modèle n'existe pas". The comment box has a "Write" tab and a "Preview" tab. Below the comment box, there are three buttons: "Start a review", "Add comment now", and "Cancel".

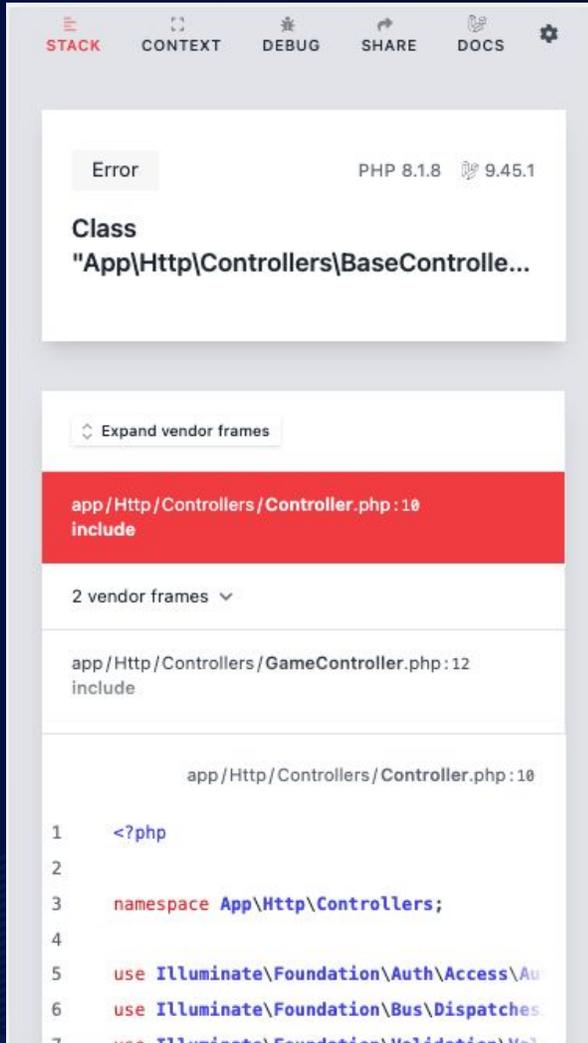
- **OBJECTIF** : identifier les erreurs ou les améliorations possibles
- Démarrer une revue (Start Review)
- **Lire attentivement** chaque modification et prendre du recul
- Proposer des ajustements ou des corrections, en utilisant un langage approprié
  - <https://conventionalcomments.org/>
- **Soumettre** la revue contenant tous les commentaires en attente

The screenshot shows a notification box with the title "2 pending comments". It lists two pending comments:

- `billing/app/Http/C...`: +310  
On pourrait ici préférer l'utili...
- `.env.example`: +85  
Penser à faire passer la varia...

At the bottom of the notification box, there are two buttons: "Pending comments ^" and "Submit review 2".

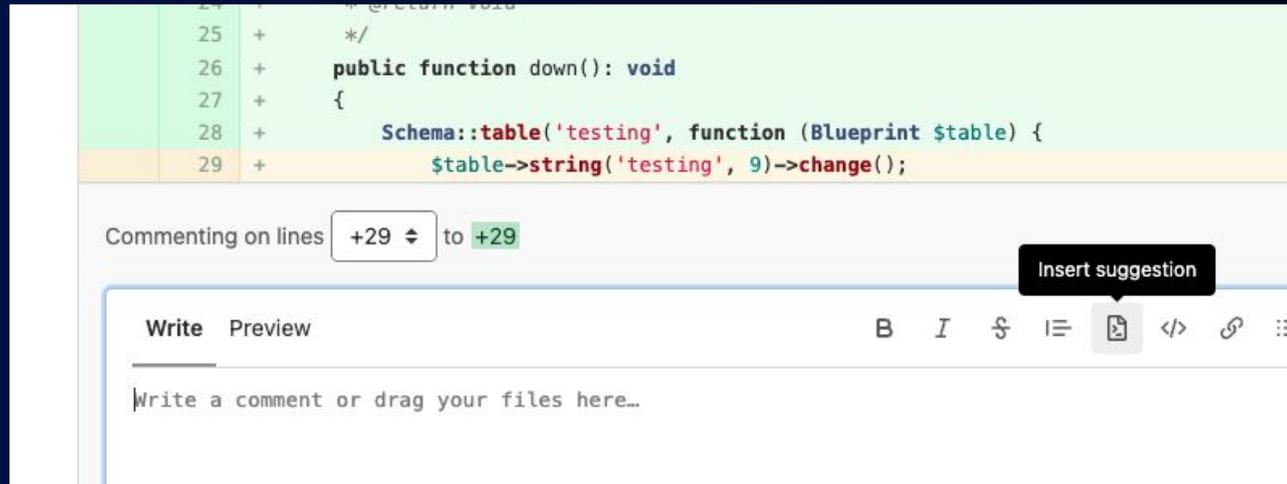
## Tester la plateforme en local



### OBJECTIF : s'assurer de la bonne réalisation fonctionnelle

- S'assurer que la **plateforme est bien fonctionnelle**
  - Une erreur 500 difficile à lire en revue
- Tester les **comportements spécifiques**
  - Tester plusieurs cas d'usage et les **effets de bords potentiellement ciblés**
- Tester des **résolutions différentes**
  - Affichage responsive difficile à lire en revue
- Tester un **environnement local** différent (mac, linux, etc.)
  - Penser à tous les prochains développeurs qui auront à intervenir sur le projet

## Utiliser les suggestions



The screenshot shows a code editor with a PHP function `down()`. Line 29 is highlighted in yellow. Below the code, a comment box is open for line 29. The comment box has a toolbar with icons for bold, italic, link, list, insert suggestion, code, link, and menu. The text "Write a comment or drag your files here..." is visible in the comment box.

```
25 +  */
26 +  public function down(): void
27 +  {
28 +      Schema::table('testing', function (Blueprint $table) {
29 +          $table->string('testing', 9)->change();
```

**OBJECTIF : Compléter les explications par du code concret**

- Un bon croquis vaut mieux qu'un long discours  
*#Napoléon*
- Possibilité d'**appliquer rapidement une solution simple**
  - simplification des process sur un hotfix



The screenshot shows the same code editor as above, but with a suggested change. The suggested change is shown in a box with a "Suggested change" header and an "Apply suggestion" button. The change is a diff showing the original code (line 29) and the suggested change (line 29).

```
25 +  */
26 +  public function down(): void
27 +  {
28 +      Schema::table('testing', function (Blueprint $table) {
29 +          $table->string('testing', 9)->change();
```

Suggested change ⓘ

29	-	\$table->string('testing', 9)->change();
29	+	\$table->string('testing', 10)->change();

Apply suggestion ▾

## Et encore...

- Vérifier la **mise à jour des documentations (Readme)**
- Proposer le **développement de tests** en cas de complexité trop grande
- Analyser le **rapport de code coverage**

**READY TO MERGE !**

04

CONCLUSION



## Ah, c'qui compte c'est pas l'arrivée c'est la quête

- **Facteur clé de succès : le temps**
  - une revue qui traîne, c'est une **branche cible qui prend de l'avance et des conflits à venir**
  - attendre avant de merger, c'est **créer un sac de noeud de branches et de PRs**
- Ne pas avoir peur de **merger**
  - plus de problème à attendre qu'à avancer
  - être attentif aux hotfix
- Utiliser le statut **Draft**
  - permet d'avoir un **état en temps réel de l'avancement** et du code produit
  - permet au **reviewer d'anticiper** son travail
- Privilégier les petites PR donc les petites features

**La revue n'est pas du flicage , mais du partage de connaissances**

**Merci !**  
**Xie xie !**  
**Obrigado !**





**TCM: //** | PARIS

56 rue de Londres  
75008 Paris



**TCM: //** | LYON

35 Rue de Marseille  
69007 LYON



**TCM: //** | HONG-KONG

Level 15  
The Lee Garden Two 28 Yun Ping Road  
Causeway Bay, Hong Kong



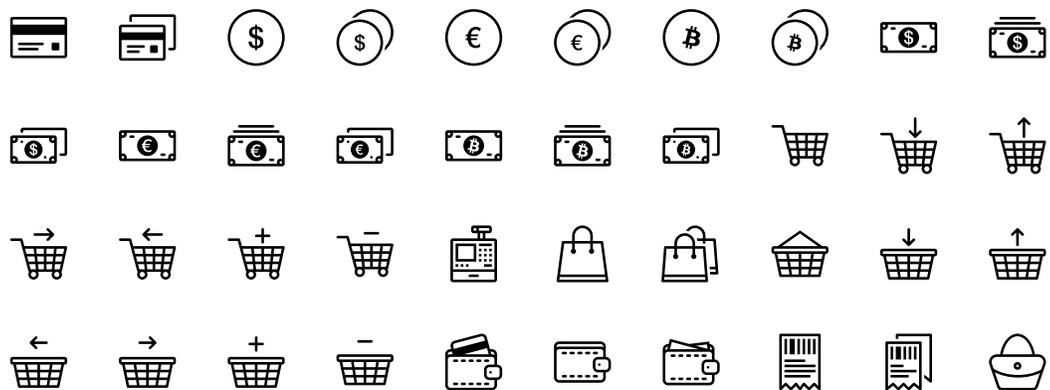
**TCM: //** | LISBOA

Rua da Palma, 219, 3ºEsq  
1100-391 Lisboa

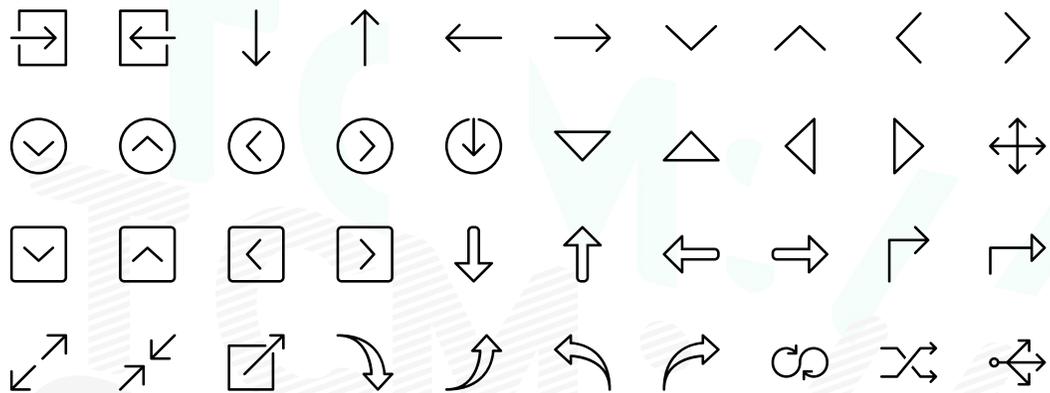


# ICÔNES MODIFIÁVEIS

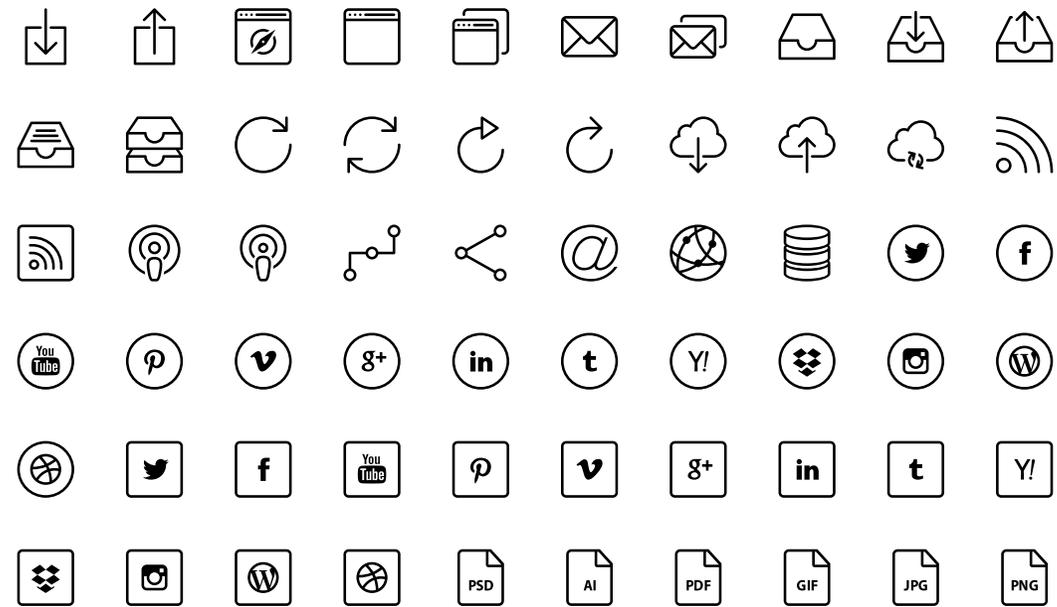
E-Commerce



Arrows



Web



Weather

