


Manual de Equipe



A background image of Kakashi Hatake from the anime Naruto. He is shown from the chest up, wearing his signature black headband with the Hidden Leaf Village symbol, a black mask covering the lower half of his face, and a green vest over a dark blue shirt. He has spiky silver hair. He is holding an orange book with a red circle and a horizontal line through it on its cover. The background is a dark, textured green, suggesting foliage.

🔥 REGRAS DO TIME



REGRAS DO TIME

1. **É pra usar o Slack, pronto acabou:** evitem me mandar mensagem no privado e enviem no channel do time, não por que não gosto, mas para que todos do time possam estar cientes e uns ajudarem os outros. **Deem *reaction* em tudo!**
2. **Confiem uns nos outros:** nós somos um time e precisamos construir uma amizade para atingirmos nossos objetivos. Para isso, não tenham medo de se expor, se você está doente não fale “tive um compromisso” ou “estive ocupado mas recupero o que atrasei”, conte para a gente, guardaremos seu segredo!
3. **Avisem sobre tudo!** Se você for faltar numa *weekly*, estive com problemas em uma tarefa, tiver compromissos **NÃO ESPERE A WEEKLY PRA AVISAR**, avise a gente assim que souber pra te ajudarmos!



REGRAS DO TIME

4. Se você terminou uma tarefa, de o **check** no Trello, avise no channel do **Slack** e mande para o repositório: outras pessoas podem estar dependendo da sua tarefa e precisam das suas alterações, e também pra saber quem preciso ficar cobrando pra fazer as tarefas.
5. Mantenham o **TimeSkip** e a **Planilha de Disponibilidade** atualizados.
6. Sempre que for mandar para o repositório, **verificar se a tarefa já está responsiva**.
7. Tentar usar **flexbox** sempre que possível! Se nao for usar, garanta que esteja fazendo direito
8. Mesmo um código ruim pode funcionar. Mas se ele não for limpo, pode acabar com uma empresa de desenvolvimento. **Mantenham o codigo limpo.**



Git Messages



Git Messages



nova feature: “feature: [...]”



estilo/UI: “style: [...]”



bugfix: “bugfix: [...]”



refatoração: “refactoring: [...]”



arquitetura/ambiente: “architecture: [...]”



deploy/tag de versão: “deploy: [...]”



Git Workflow



Git Workflow

- Começar uma nova tarefa:
 - vá para a branch development:
`$ git checkout development`
 - verifique se existem novas mudanças:
`$ git pull`
 - crie uma nova branch:
`$ git checkout -b nome_da_tarefa`



Git Workflow

- Trabalhe em sua tarefa, COMMITANDO CONTINUAMENTE em intervalos regulares para manter o rastro daquilo que você fez:
 - adicione suas modificações:

```
$ git add diretorio_do_arquivo_modificado
```
 - ou para adicionar todas as modificações:

```
$ git add .
```
 - commit as modificações:

```
$ git commit -m "mensagem do commit"
```

ATENÇÃO: mensagens de commits devem ser escritas em inglês, serem claras e seguirem o padrão definido pela equipe



Git Workflow

- Enviando seus commits para o repositório remoto:
 - vá para a branch development:

```
$ git checkout development
```
 - verifique se existem novas mudanças:

```
$ git pull
```
 - se sim:
 - volte para a branch da sua tarefa:

```
$ git checkout nome_da_tarefa
```
 - de um rebase:

```
$ git rebase development
```
 - resolva os conflitos, se houverem. (não hesite em pedir ajuda caso necessário)
 - volte para a branch development e repita o processo:

```
$ git checkout development
```



Git Workflow

- se não:
 - `$ git merge --no-ff nome_da_tarefa`
 - `$ git push`