



UNIVERSIDADE  
VILA VELHA  
ESPÍRITO SANTO

# Disciplina: Desenvolvimento Web - *Back End*

## Temática: Tópicos 2 e 3

Professor: Vinícius da Rocha Motta  
vinicius.motta@uvv.br



# Criando um repositório

- Criar um repositório Git é o primeiro passo para iniciar o versionamento de arquivos de um projeto de software.
  - `git init`

Assim, um repositório é criado e uma pasta `.git` é adicionada, contendo todas as informações que o repositório precisa para controlar as versões do projeto.



# Comando add

Utilize o comando ***add*** para iniciar o monitoramento de novos arquivos que foram criados ou depositados nas pastas do repositório local, assim como para preparar os arquivos que já estão sendo monitorados e que foram modificados após determinado ***commit*** (GIT..., 2019a).

```
git add helloWorld.c
```

```
git add *.c
```

```
git add .
```



# Comando status

- Para visualizar o status dos arquivos do repositório local, entre com a seguinte instrução no prompt de comando (GIT..., 2019h):
  - `git status`



# Comando commit

- Para efetuar a consolidação dos arquivos de um repositório, entre com a instrução ***commit*** no prompt de comando conforme a seguir:
  - `git commit -m "Versao inicial"`

O parâmetro **-m** define uma mensagem que pode servir como identificação para a versão que foi criada, por exemplo, “Versão inicial” quando se tratar do primeiro commit realizado, ou “Implementação do sprint #3”, que se refere a alguma implementação específica.



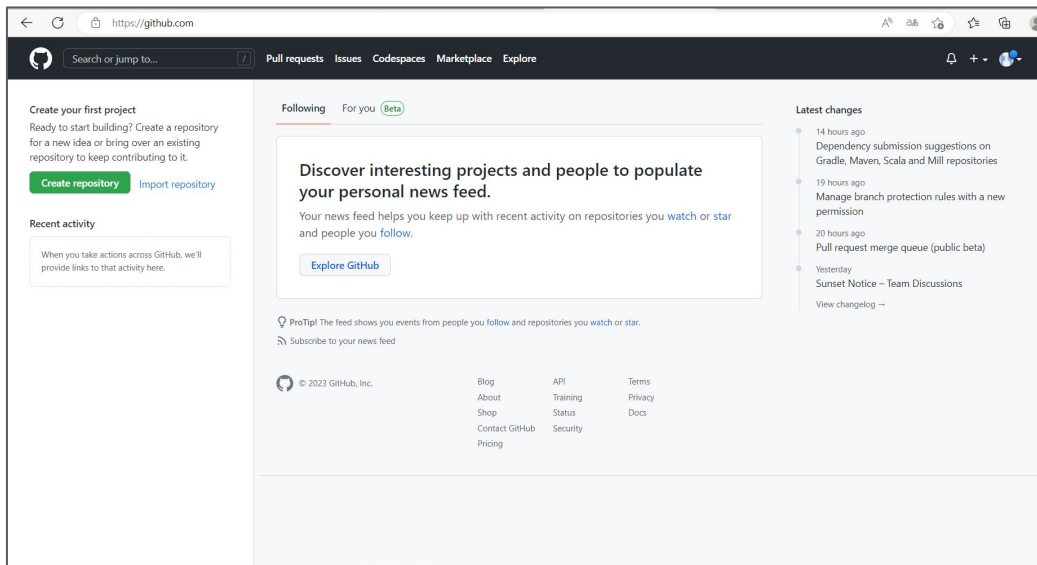
# Comando log

- Pode ser necessário consultar o histórico desses commits para se obter mais detalhes, como a data em que foi realizado e o responsável pela execução. O comando utilizado para acessar a essas informações é log, conforme apresentado a seguir (GIT..., 2019f).
  - git log



# Transferência do repositório remoto para o local

No repositório remoto (GitHub) de um projeto, fica armazenada sua coleção de arquivos e pastas, contendo todas as informações sobre as alterações que foram realizadas, os responsáveis pelas modificações e quando estas foram feitas.





# Clonando repositório

- Para clonar este repositório remoto por linha de comando, abra uma janela de terminal e entre com a seguinte instrução:
  - git clone <https://github.com/ThiagoMartinsSaraiva/pulse-ani-mation.git>

Após a execução do comando veja o histórico de versões, incluindo todos os commits que foram realizados, pode ser conferido por meio do comando ***git log***.





## Atualização do repositório local

Para trazer as atualizações do repositório remoto para o repositório local, dois comandos podem ser utilizados: ***fetch*** e ***pull***.



# Comando remote

- Este comando visualiza a lista de remotos e permite adicionar novos remotos. Quando um clone é realizado, automaticamente o Git coloca o nome de origin para o repositório (GIT..., 2019g)
  - `git remote -v`



## Comando remote

- Para incluir o repositório remoto chamado angular/angular do Github, entre com a seguinte instrução no prompt de comando, informando um nome para ele e seu endereço/caminho. Note que o parâmetro angular após add serve como um apelido para a vinculação dos repositórios, e não precisa necessariamente ser idêntico ao nome do repositório:
  - `git remote add angular https://github.com/angular/angular`



# Comando fetch

- O comando fetch busca todos os dados do repositório remoto que ainda não constam no seu repositório local. No entanto, ele não realiza o merge automaticamente com os seus arquivos do repositório local. Se for necessário fazer o merge, você deverá fazê-lo manualmente (GIT..., 2019d). Para rodar o comando fetch, utilize a seguinte instrução no prompt de comando:
  - `git fetch origin`



## Comando pull

- Assim como fetch, o comando pull traz as atualizações do repositório remoto para o repositório local. A diferença é que pull faz o merge automaticamente com os dados do repositório local (GIT..., 2019f).
  - `git pull origin`



UNIVERSIDADE  
VILA VELHA  
ESPÍRITO SANTO

# Construção de uma aplicação local em C# com SGBD

A **linguagem C#** é amplamente **utilizada** no meio **acadêmico e comercial**.

Com esta linguagem é possível implementar aplicativos que atendem às mais diversas funcionalidades, como **sistemas educativos e comerciais, ferramentas de software de uso geral e jogos**.



# Construção de uma aplicação local em C# com SGBD

O MySQL é um dos **sistemas gerenciadores de bancos de dados mais utilizados no mercado.**

Sua **facilidade** de uso colabora para isso.

Com as **ferramentas do MySQL**, pode-se **modelar e criar bancos de dados** para as mais diversas necessidades, desde sistemas pequenos até grandes bases de dados.



# Construindo uma aplicação em C#

**C# é uma linguagem de programação da plataforma .NET Framework, desenvolvida pela Microsoft.**

**Sua sintaxe lembra as linguagens C, C++, Java e Javascript, e os programas escritos em C# combinam um ou mais arquivos de origem para produzir um *Assembly*, que pode ser um arquivo executável (.exe) ou uma biblioteca dinâmica (.dll)**





# Construindo uma aplicação em C#

- Orientada a componentes: **módulos** que **reutilizam componentes**.
- Orientada a objetos: recursos do **paradigma da orientação a objetos**.
- Fortemente tipada: **as variáveis possuem tipos de dados bem definidos**.
- Ambiente gerenciado: o **gerenciamento de memória dos programas escritos em C#** é realizado pelo **runtime via Garbage Collector (GC)**, garantindo um modo **seguro** de execução.



UNIVERSIDADE  
**VILA VELHA**  
ESPIRITO SANTO

# Vídeo de demonstração

[https://www.youtube.com/watch?v=TgY56\\_8pOyE&ab\\_channel=Programa%C3%A7%C3%A3oPlena](https://www.youtube.com/watch?v=TgY56_8pOyE&ab_channel=Programa%C3%A7%C3%A3oPlena)



UNIVERSIDADE  
**VILA VELHA**  
ESPÍRITO SANTO

# Dúvidas?

[vinicius.motta@uvv.br](mailto:vinicius.motta@uvv.br)