



# Controle de versão com Git

Leila Rodrigues  
Engenharia de Software  
UFC - 2020.1

# "FINAL".doc



# O que é controle de versão?

# O que é controle de versão?

O controle de versão é um sistema que registra alterações em um arquivo ou conjunto de arquivos ao longo do tempo, para que você possa recuperar versões específicas mais tarde.

# Sistema de Gerenciamento de Versões

- Soluções Comerciais
  - SourceSafe
  - TFS
  - ClearCase
- Soluções Open-source
  - Concurrent Version System (CVS)
  - Subversion (SVN)
  - Git
  - Mercurial

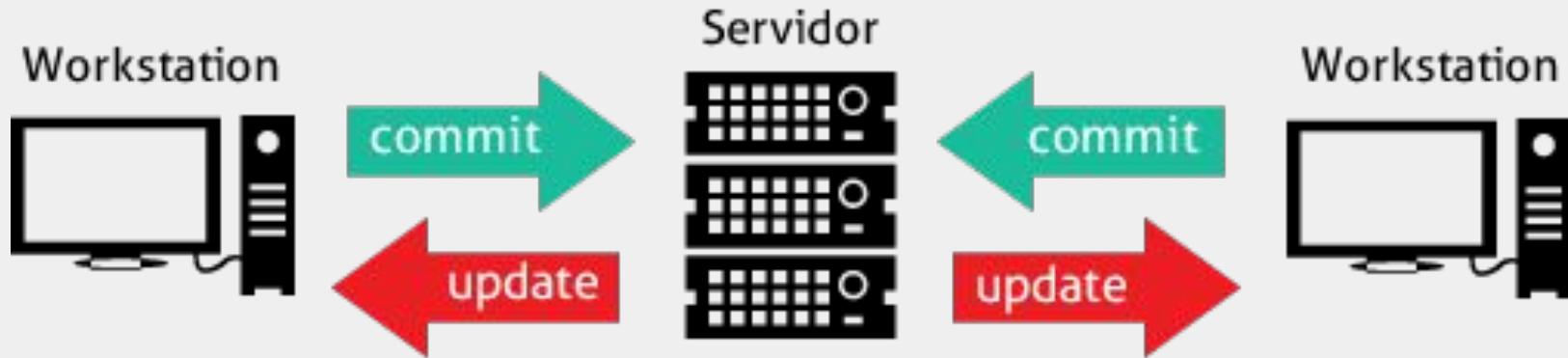
# Por que usar?

- Facilidade para verificar as mudanças no código entre versões
- Facilidade em restaurar versões
- Permite unir alterações feitas por diversos desenvolvedores
- Evita acumular arquivos numerados (ou com data da versão)
- Histórico das alterações

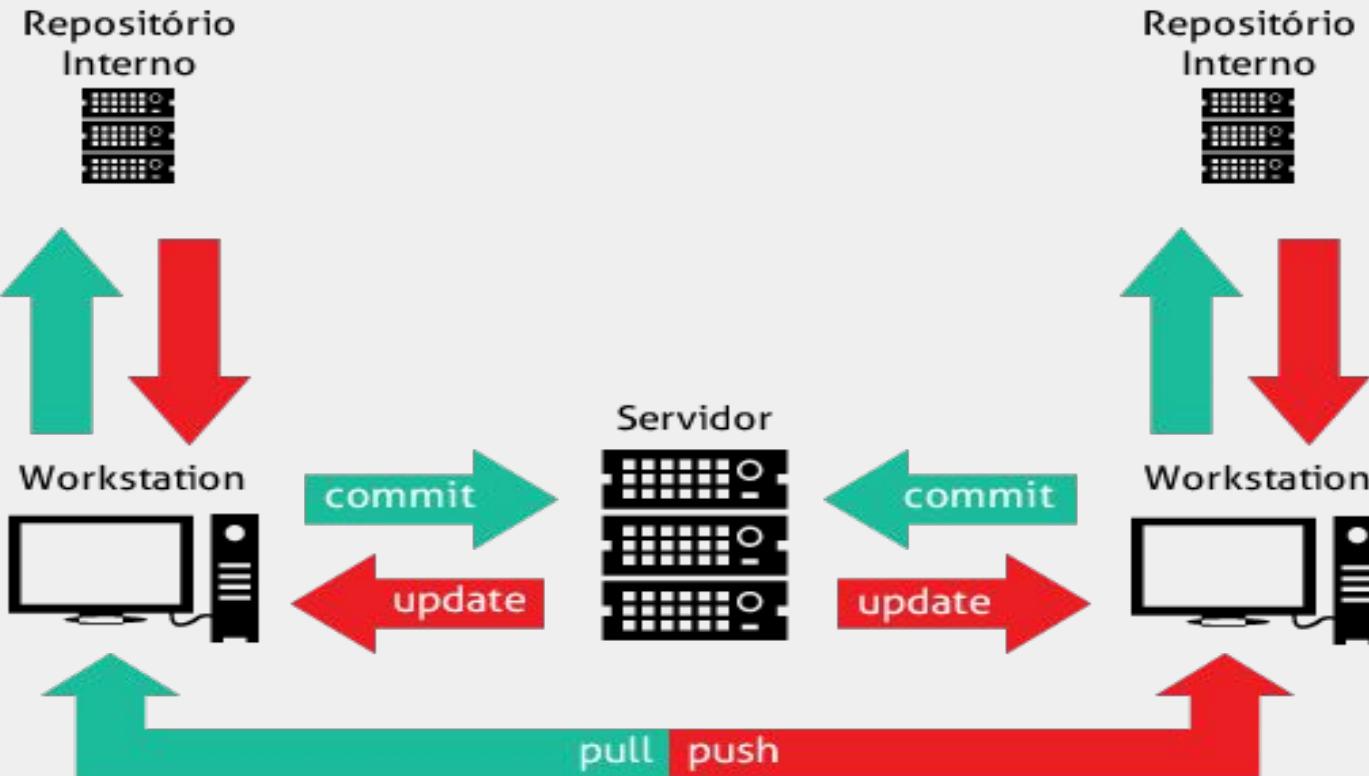
# Como funciona?

- O projeto fica armazenada no servidor (repositório remoto)
  - Armazena histórico e modificações de cada versão
- Essa sincronização de arquivos entre a estação de trabalho e o servidor é feita através dos comandos **commit** e **update**.

# Controle de Versão Centralizado



# Controle de Versão Distribuído



Fonte: [Controle de versão Git, Github e Bitbucket: afinal o que é tudo isso](#)

# Contexto histórico

- O BitKeeper era utilizado por Linus para gerenciar o código de suas versões do Linux.
- Ocorre um problema com a licença do BitKeeper.
  - Linus gostava das funcionalidades
  - E não gostava de outras ferramentas como CVS e SVN
- 05/04/2005: 1<sup>a</sup> versão do Git
- 15/06/2005: Git é usado para controle de versão do kernel do Linux.

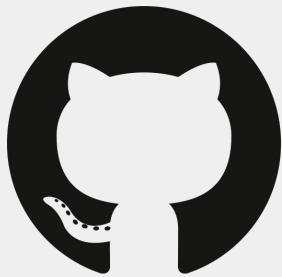
# Vantagens do Git

- Sistema open-source de gerenciamento de versões
- Pode ser usado para controlar versões de qualquer formato (geralmente arquivos de texto)
- Consistência
- Velocidade
  - Muito mais rápidos que sistemas de controle de versão convencionais (CVS e Subversion)
- Simplicidade
- Espaço
  - Algoritmos de compressão eficientes que analisam “o todo”

# Desvantagens do Git

- Documentado quase exclusivamente através de páginas geradas via “man”.
- Maior massa de dados em operações push/pull do que em commits/updates de versões centralizadas.

# Ferramentas populares



Github



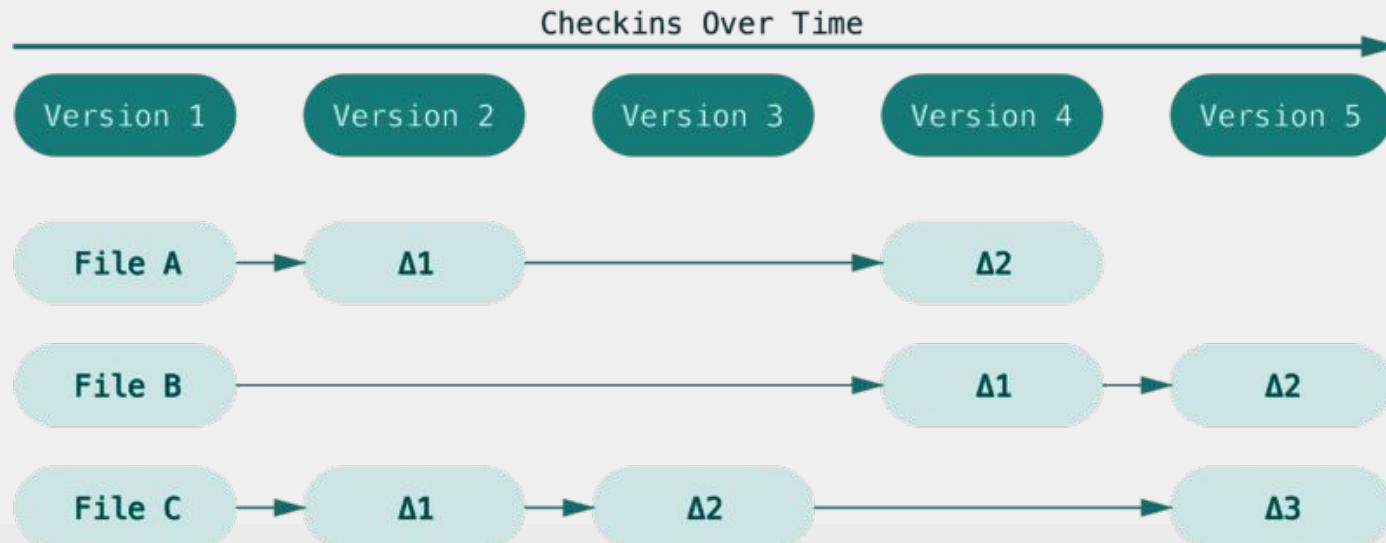
Gitlab



Bitbucket

# *Snapshots*

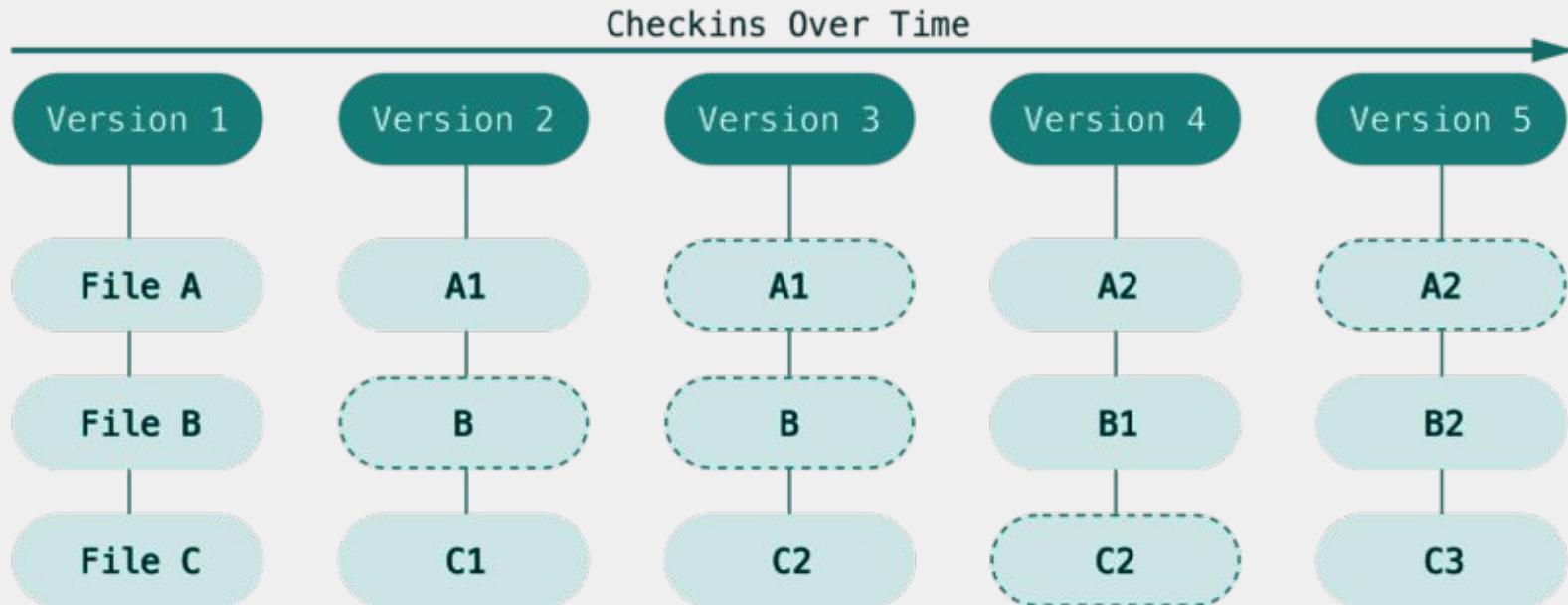
- Outros sistemas armazenam as informações como uma lista de alterações baseadas em arquivos.
- Controle de versão baseado em delta.



# Snapshots

- O Git registra “momentos”, os quais são chamados de snapshots, contendo uma espécie foto dos arquivos.
- A cada commit, o Git basicamente tira uma foto da aparência de todos os seus arquivos naquele momento e armazena uma referência a essa captura.
- O Git pensa em seus dados mais como um fluxo de snapshots.

# Snapshots



# Operações locais

- Quase todas operações do Git são feitas de forma local.
- Você tem todo o histórico do projeto no seu disco local, a maioria das operações parece quase instantânea.
- Isso significa que há muito pouco que você não pode fazer se estiver offline.

# Integridade

- A integridade das informações de cada projeto são asseguradas por um checksum (chave de verificação de integridade).
- Tudo no Git é referido por checksum.
- Isso significa que é impossível alterar o conteúdo de qualquer arquivo ou diretório sem o Git saber disso.

# Integridade

- Para realizar checksum, o Git usa uma implementação de hash SHA-1  
24b9da6552252987aa493b52f8696cd6d3b00373
- Git armazena tudo em seu banco de dados não pelo nome do arquivo, mas pelo valor de hash do seu conteúdo

# Três estados

- **Modificado:** alterou arquivo, mas ainda não foi confirmado no banco de dados.
- **Preparado:** marcou um arquivo modificado na versão atual para entrar no próximo snapshot de confirmação.
- **Confirmado:** os dados foram armazenado com segurança no banco de dados local.

# Armazenamento

- Pasta .git
  - Apenas no diretório raiz do projeto.
  - Contém todos os objetos, commits e configurações do projeto.
  - .git/config: arquivo com configurações específicas do repositório.
- .gitignore
  - Arquivo texto que indica os arquivos que devem ser ignorados

# Instalação

- Linux
  - Para Debian, Ubuntu e derivados:
    - **apt-get install git**
- Windows
  - <https://gitforwindows.org>

# Configurações iniciais

- O git config é a ferramenta do Git usada para configurações.
- Um dos passos iniciais é identificar o nome de usuário e e-mail para o git.

```
$ git config --global user.name "seunome"
```

```
$ git config --global user.email seunome@email.com
```

# Configurações iniciais

- Se desejar que essas informações sejam diferentes para projetos específicos.
- É necessário apenas digitar o mesmo comando sem a diretiva **--global** no diretório em que foi criado um projeto .Git.

# Repositório Git

Transformar um diretório local que atualmente não está sob o controle de versão:

- Linux:

```
$ cd /home/user/project
```

```
$ git init
```

- Windows:

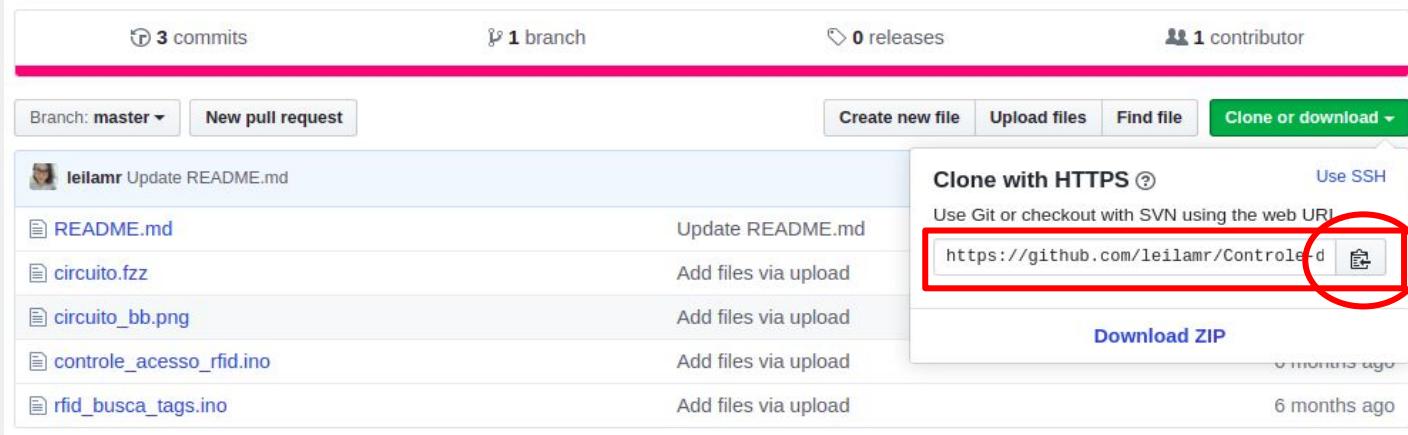
```
$ cd C:\Users\nameusuario\Desktop
```

```
$ git init
```

# Repositório Git

Clonar um repositório Git existente

- Copie a URL do repositório



- Digite o código abaixo no terminal:  
\$ git clone <url-do-repositório>

# Adicionar arquivos à lista de commit

- Adicionar um ou mais arquivos:  
\$ git add <nome-arquivo>  
\$ git add \*.extesão  
\$ git add \*
- Verificar alterações  
\$ git status

# **Ignorar arquivos à lista de commit**

- Aplicar commit apenas arquivos fonte e ignore arquivos derivados
- Por exemplo, ignore executáveis, arquivos gerados durante a compilação e arquivos durante a execução dos testes de código.

# Ignorar arquivos à lista de commit

- Arquivo `.gitignore`

```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg
MANIFEST
```

```
# ignore all .a files
*.a

# but do track lib.a, even though you're ignoring .a files above
!lib.a

# only ignore the TODO file in the current directory, not subdir/TODO
/TODO

# ignore all files in any directory named build
build/

# ignore doc/notes.txt, but not doc/server/arch.txt
doc/*.txt

# ignore all .pdf files in the doc/ directory and any of its subdirectories
doc/**/*.pdf
```

# Commit

- Confirmar alterações  
\$ git commit  
\$ git commit -m “sua mensagem”
- Desfazendo um commit específico  
\$ git revert 58cffab

# Remover arquivos

- Para remover arquivos presentes no seu repositório .git, é preciso removê-lo não só do diretório em que ele se encontra, mas também é removê-lo da área de seleção e então efetuar o commit.

```
$ git rm arquivo.extensão
```

```
$ git commit
```

# Histórico

- Para verificar o histórico dos commits efetuados em determinado repositório

\$ git log

```
leilla@leila-rodrigues:~/Documentos/mathematical-morphology$ git log
commit c857f5858c9596fb95d35ec70df6f30745f95355 (HEAD -> master, origin/master,
origin/HEAD)
Author: leila <leilarodrigues@alu.ufc.br>
Date:   Fri Oct 11 23:13:08 2019 -0300

    Outputs

commit 3caad5ed8a85d0812ea779b3cfffa7da7869afe5
Author: leila <leilarodrigues@alu.ufc.br>
Date:   Fri Oct 11 15:29:25 2019 -0300

    etapa 2

commit f7ae55c9bc253ae5fd511ff9552c247c9662f703
Author: leilarodrigues <leilarodrigues@alu.ufc.br>
Date:   Mon Oct 7 11:13:02 2019 -0300

    Inputs
```

# Repositórios remotos

- Quando se deseja compartilhar um diretório Git com outras pessoas, é necessário entender e aprender como gerenciar um repositório remoto do Git.
- Para ver quais servidores remotos você configurou, você pode executar o comando:  
`$ git remote`

# Repositórios remotos

- Enviar modificações  
    \$ git push origin master
- Pegar modificações feitas em um repositório  
    \$ git fetch origin master (pega as atualizações e coloca em outro branch)  
    \$ git pull origin master (pega as atualizações e já faz merge)

# Ramificações

- Ramificação significa que você diverge da linha principal de desenvolvimento e continua a trabalhar sem mexer nessa linha principal.
- A ramificação isso cria um novo ponteiro para movimentação no repositório Git.

`$ git branch <nome>`

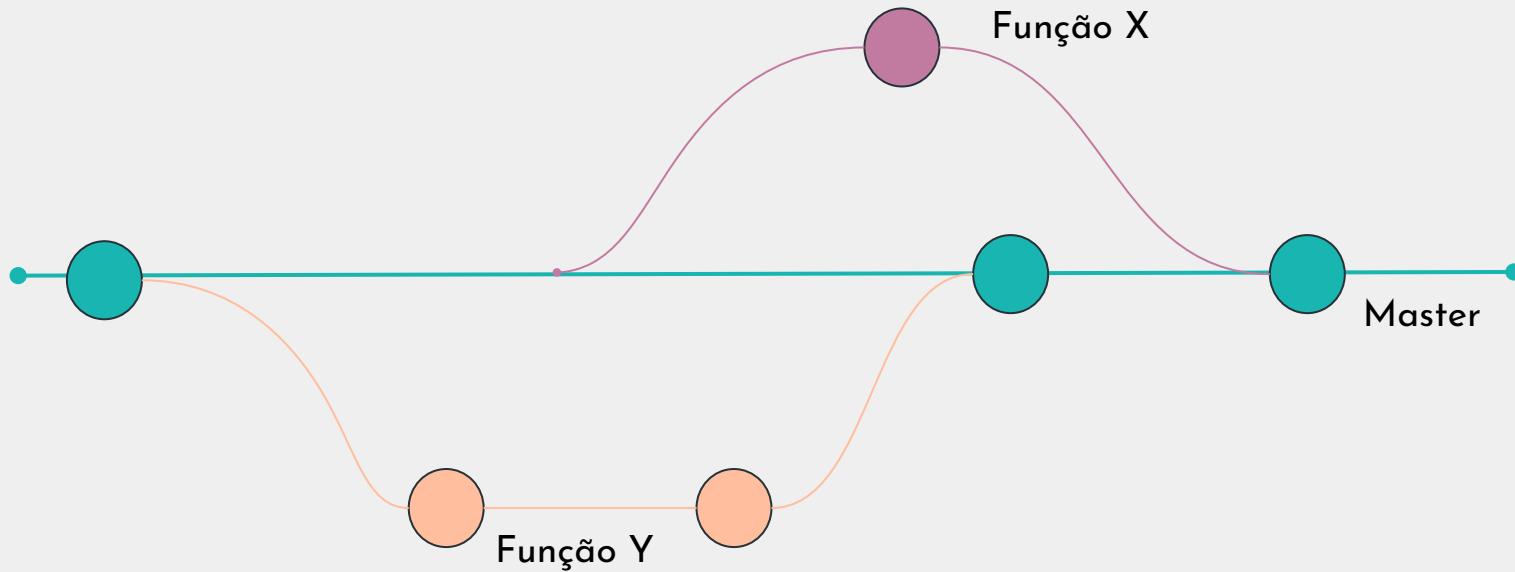
- Alterando a ramificação

`$ git checkout <nome>`

- Visualiza branches

`$ git branch`

# Ramificações



# Merge

- Mesclar a ramificação criada com o ramo master  
\$ git checkout master  
\$ git merge <nome>

# Problemas

- Problemas (bugs) ou sugestões de tarefas para o código?

The screenshot shows a GitHub issue page for a repository named "suhas-nithyanand / Image-Segmentation-using-Region-Growing". The issue is titled "amazing job #1" and is marked as "Open". It was created by "iamyourdaddy" on 20 Jun 2017 with 1 comment.

The first comment, from "ghost", asks: "can you explain what is find\_region() doing?".

The second comment, from "akkshita", provides a traceback:

```
Traceback (most recent call last):
File "region_growing.py", line 100, in
if arr_out[i][j] >25:
IndexError: index 481 is out of bounds for axis 0 with size 481
```

The GitHub interface includes sections for Assignees (No one assigned), Labels (None yet), Projects (None yet), and Milestone (No milestone). The Notifications section indicates that notifications are not being received from this thread. There is also a "Comment" button at the bottom right.

# Licenças



**Eu quero uma licença simples e permissiva**

A [Licença MIT](#) é uma licença permissiva que é concisa e vai direto ao ponto. Ela permite que as pessoas façam o que quiserem com seu código, desde que forneçam uma atribuição de volta para você e não lhe responsabilize.

[jQuery](#) e [Rails](#) usam a Licença MIT.



**Eu estou preocupado com patentes**

A [Licença Apache](#) é uma licença permissiva, similar à Licença MIT, mas que também provê uma concessão expressa de direitos de patente dos contribuintes para os usuários.

[Apache](#), [SVN](#), e [NuGet](#) usam a Licença Apache.



**Eu me preocupo em compartilhar melhorias**

A GPL ([V2](#) ou [V3](#)) é uma licença "copyleft" que exige que quem distribui o seu código ou uma obra derivada deve disponibilizar o fonte sob os mesmos termos.

[Linux](#), [Git](#), e [WordPress](#) usam GPL.

# README.md

README.md



## Dataset Iris plant

Iris flower classification with MLP using MATLAB.

### Atribute informations

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class: Iris Setosa, Iris Versicolour and Iris Virginica.

### Class coding

- Setosa = [1 0 0 ]
- Versicolor = [0 1 0]
- Virginica = [0 0 1]

### Network settings:

```
Train = 70%, Validation = 15% and Testing = 15%
Number hidden of nodes = 4
Epochs = 1000
Trainng Function = trainlm
Transfer Function (layer 1) = tansig
Trasnfer Function (layer 2) = purelin
```

```
Accuracy = 99.3%
```

# Sugestões de leitura

- Documentação oficial:
  - <https://book.git-scm.com/book/en/v2/>
- Guia rápido:
  - [https://rogerdudler.github.io/git-guide/index.pt\\_BR.html](https://rogerdudler.github.io/git-guide/index.pt_BR.html)
- Guia rápido sobre licenças:
  - <http://escolhaumalicenca.com.br/>



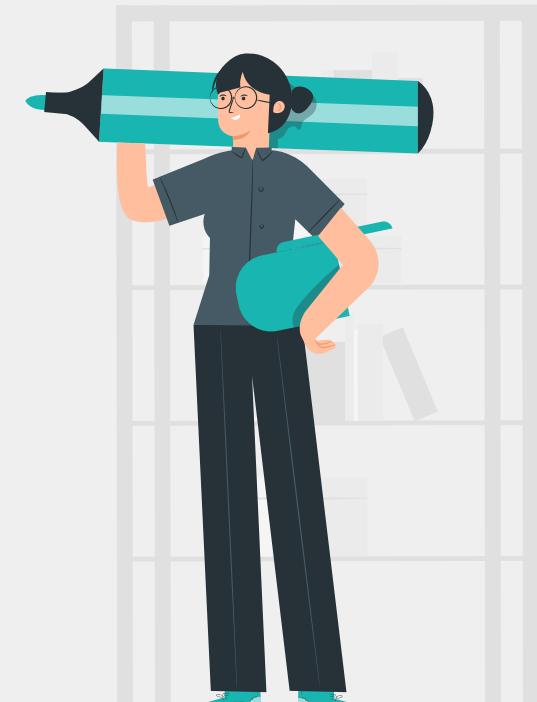
# CONTENTS OF THIS TEMPLATE

Here's what you'll find in this **Slidesgo** template:

A slide structure based on a project proposal, which you can easily adapt to your needs. For more info on how to edit the template, please visit **Slidesgo School** or read our **FAQs**.

1. An assortment of illustrations that are suitable for use in the presentation can be found in the **alternative resources** slide.
2. A **thanks** slide, which you must keep so that proper credits for our design are given.
3. A **resources** slide, where you'll find links to all the elements used in the template.
4. **Instructions for use**.
5. Final slides with:
  - a. The **fonts and colors** used in the template.
  - b. More **infographic resources**, whose size and color can be edited.
  - c. Sets of **customizable icons** of the following themes: general, business, avatar, creative process, education, help & support, medical, nature, performing arts, SEO & marketing, and teamwork.

You can delete this slide when you're done editing the presentation.



# TABLE OF CONTENTS

01

## About the Project

Here you could describe the topic of the section

02

## Major Requirements

Here you could describe the topic of the section

03

## Project Goals

Here you could describe the topic of the section

04

## Sneak Peek

Here you could describe the topic of the section

05

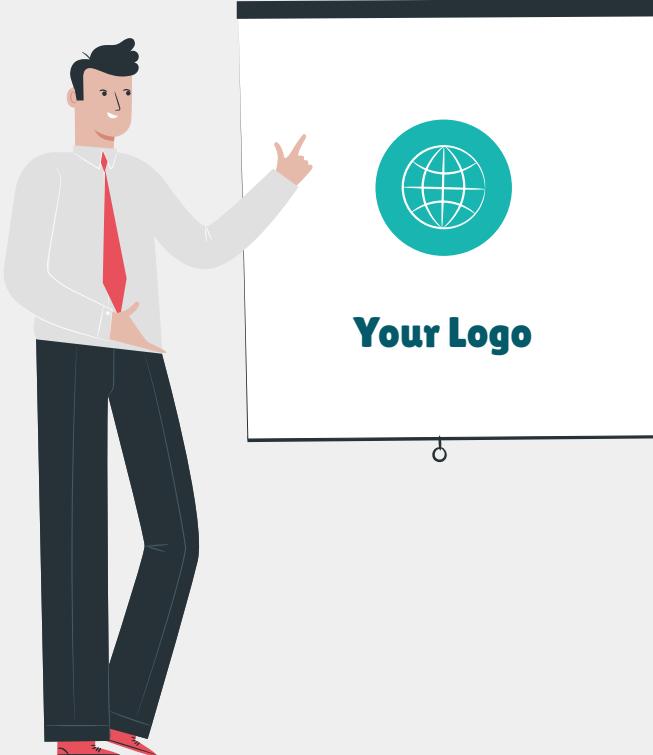
## Project Stages

Here you could describe the topic of the section

06

## Our Team

Here you could describe the topic of the section



## OUR COMPANY

Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than our Moon. The planet's name has nothing to do with the liquid metal, since it was named after the Roman messenger god, Mercury



# WHAT WE ARE WORKING ON

## Neptune

Neptune is the fourth-largest planet in our Solar System



## Jupiter

Jupiter is a gas giant and the biggest planet in our Solar System



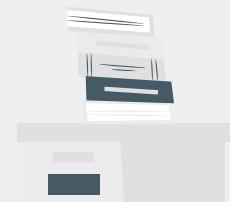
## Mercury

Mercury is the smallest planet in our Solar System



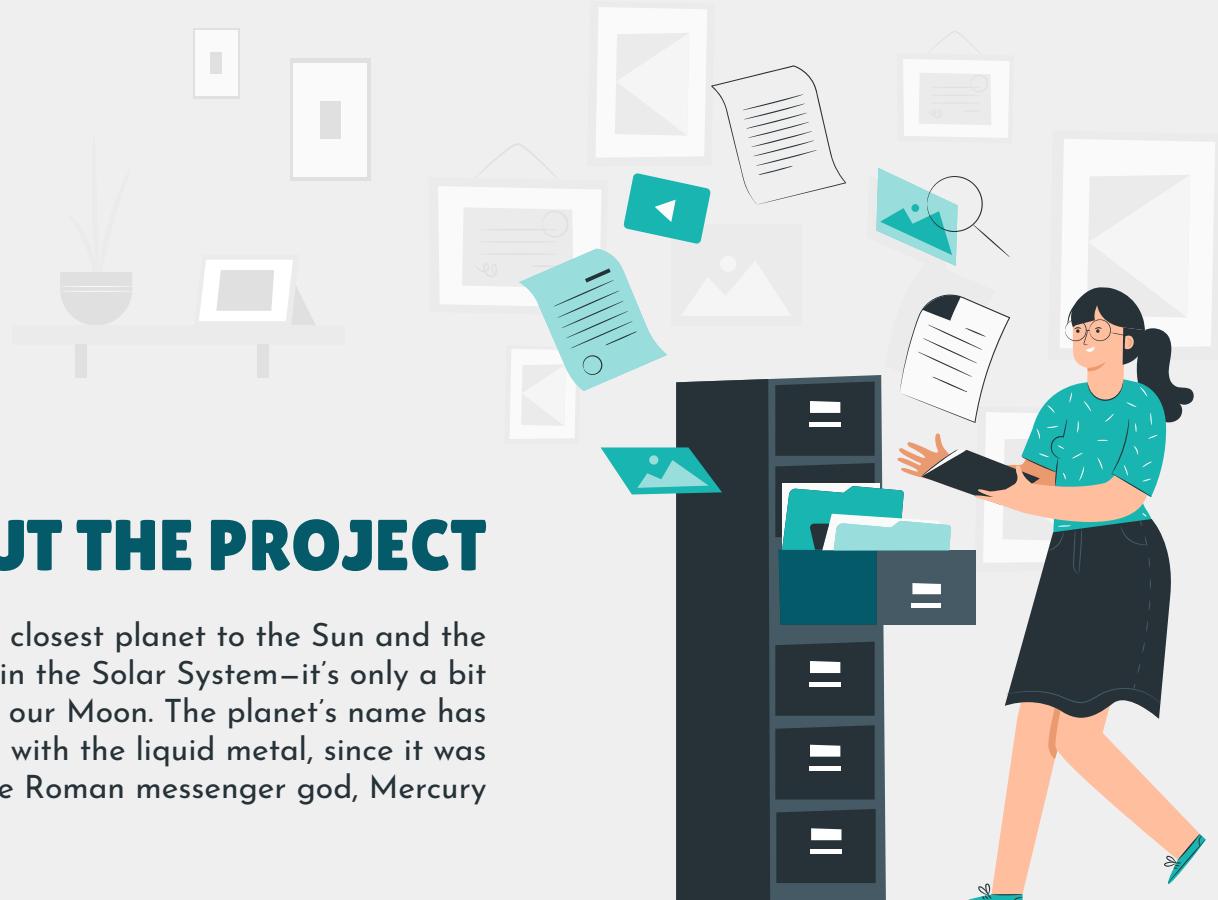
## Saturn

Saturn is composed mostly of hydrogen and helium



# ABOUT THE PROJECT

Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than our Moon. The planet's name has nothing to do with the liquid metal, since it was named after the Roman messenger god, Mercury



# ABOUT THE PROJECT



Venus has a beautiful name and is the second planet from the Sun. It's terribly hot—even hotter than Mercury—and its atmosphere is extremely poisonous. It's the second-brightest natural object in the night sky after the Moon

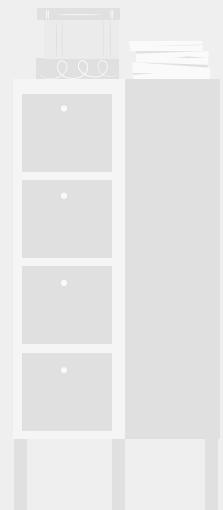
Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than our Moon. The planet's name has nothing to do with the liquid metal, since it was named after the Roman messenger god, Mercury

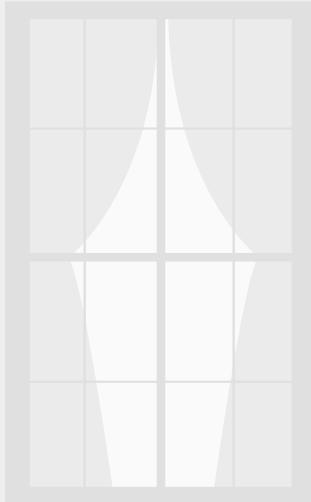
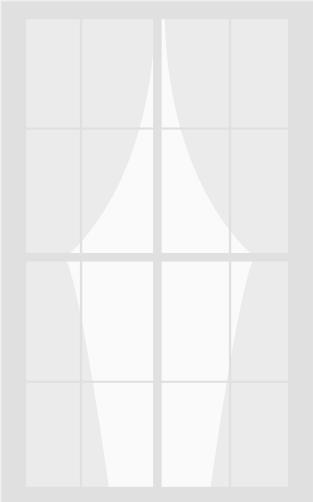




“This is a quote. Words full of wisdom that someone important said and can make the reader get inspired”

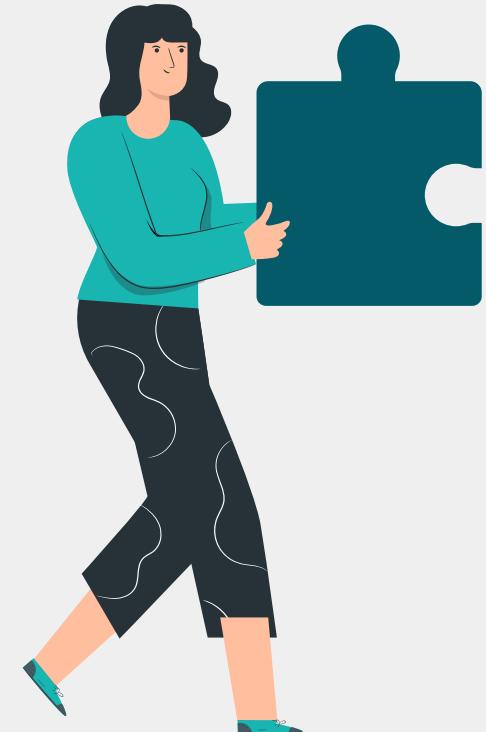
**-SOMEONE FAMOUS**

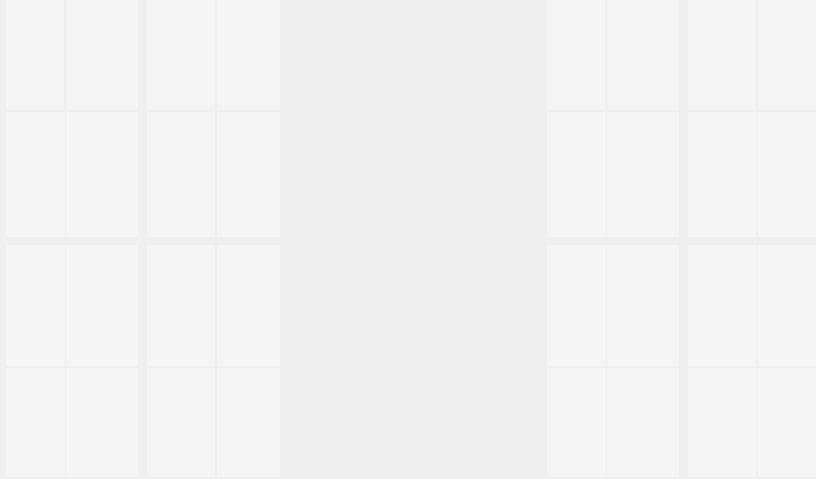
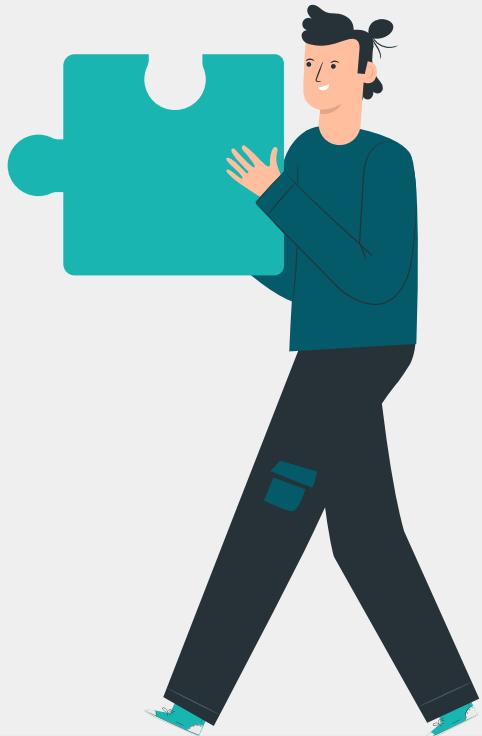




Venus has a beautiful name and is the second planet from the Sun. It's terribly hot—even hotter than Mercury—and its atmosphere is extremely poisonous. It's the second-brightest natural object in the night sky after the Moon

**NOW**



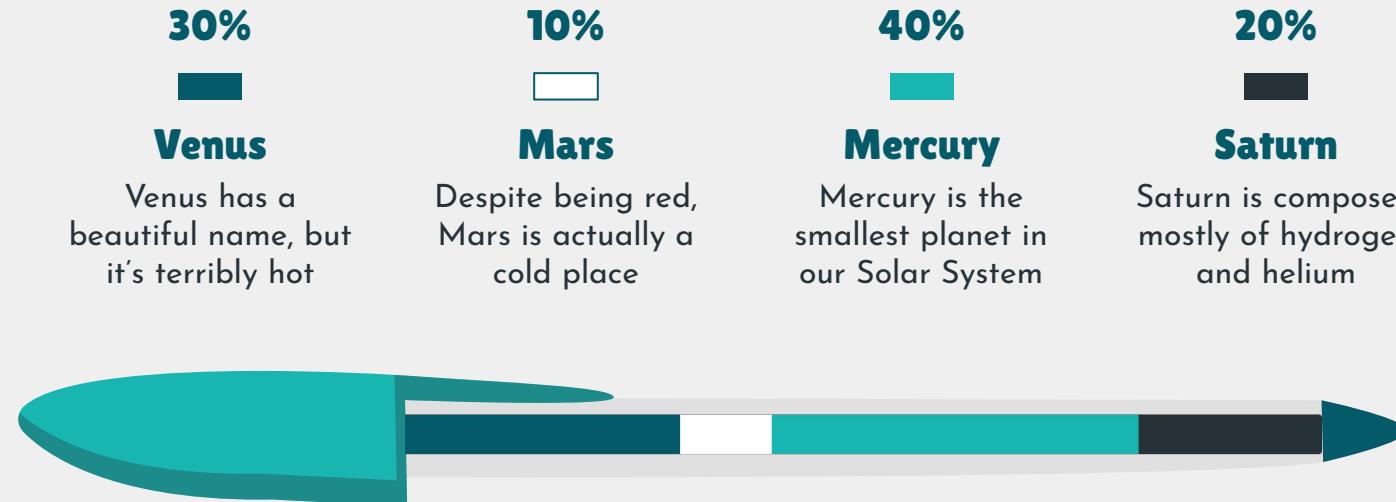


Mercury is the closest planet to the Sun and the smallest one in our Solar System—it's only a bit larger than our Moon. The planet's name has nothing to do with the liquid metal, since it was named after the Roman messenger god

## FUTURE



# BUDGET



**€6,000,000**

If you want to modify this graph, click on it, follow the link, change de data and replace it

# PROJECT GOALS



# PROJECT GOALS



## Mercury

Mercury is the smallest planet in our Solar System



## Venus

Venus has a beautiful name, but it's terribly hot



## Saturn

Saturn is composed mostly of hydrogen and helium



## Neptune

Neptune is the farthest planet from the Sun



## Mars

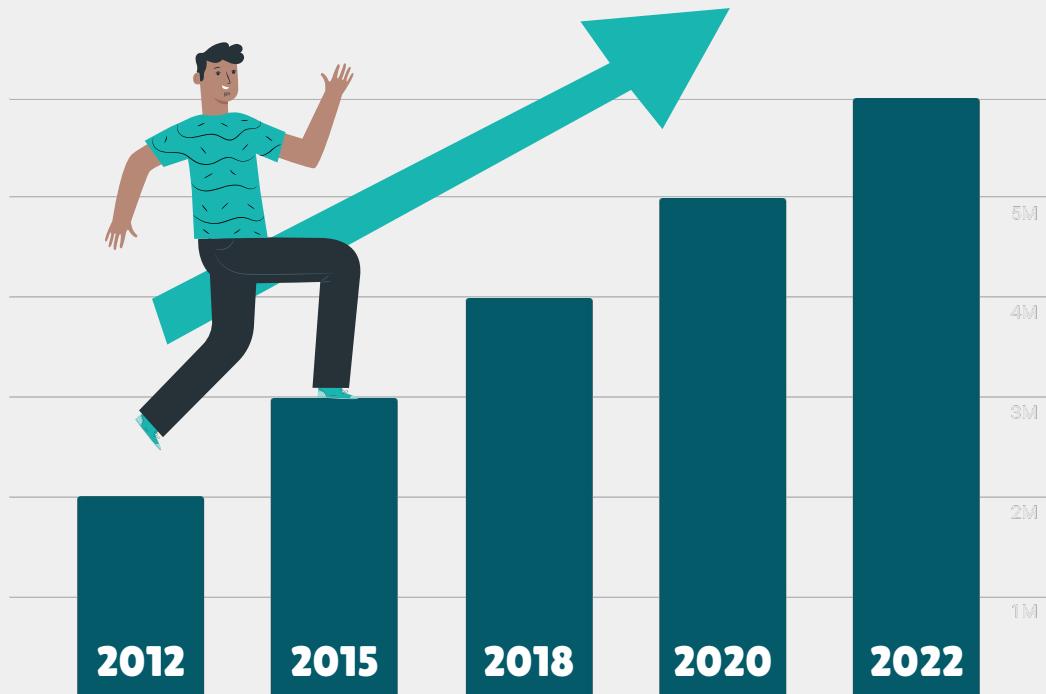
Despite being red, Mars is actually a cold place



## Jupiter

Jupiter is the biggest planet in our Solar System

# PREDICTED RESULTS



**\$5M**

Mars is a cold place

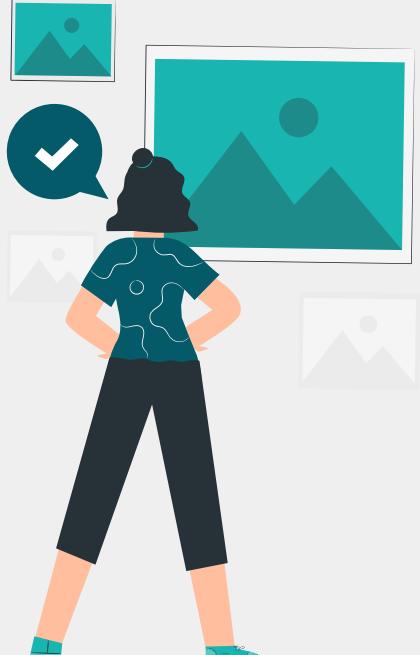
**100**

Venus has a beautiful name

**150**

Saturn is the ringed planet

To modify this graph, click on it,  
follow the link, change de data  
and replace it



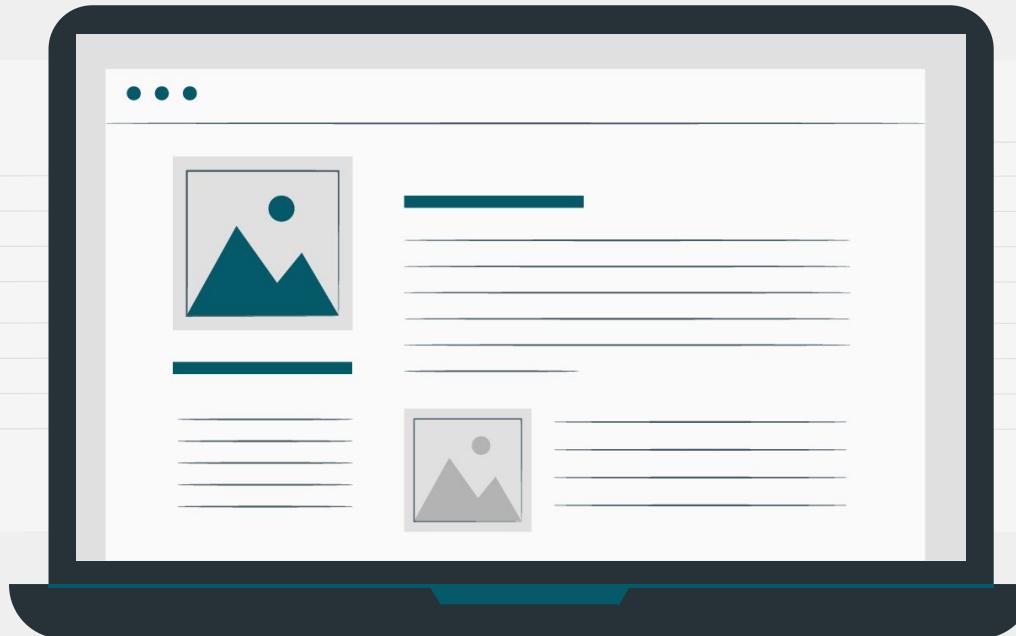
## SNEAK PEEK

Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than our Moon. The planet's name has nothing to do with the liquid metal, since it was named after the Roman messenger god, Mercury



# SNEAK PEEK

Insert your multimedia content here. You can replace the image in the screen with your own work. Just delete this one and add yours

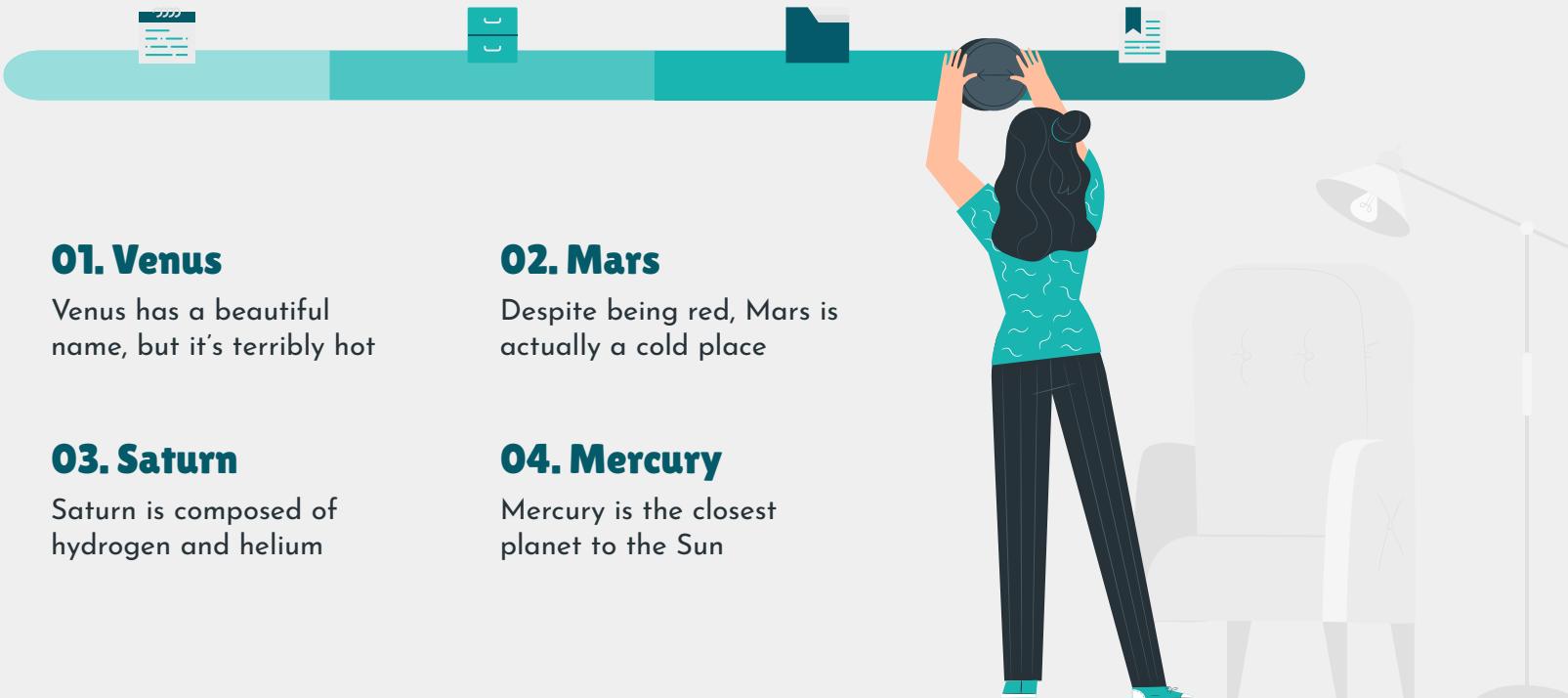


A PICTURE  
ALWAYS  
REINFORCES  
THE CONCEPT



**A PICTURE  
IS WORTH A  
THOUSAND  
WORDS**

# PROJECT STAGES



# PROJECT STAGES



## Mercury

Mercury is the closest planet to the Sun

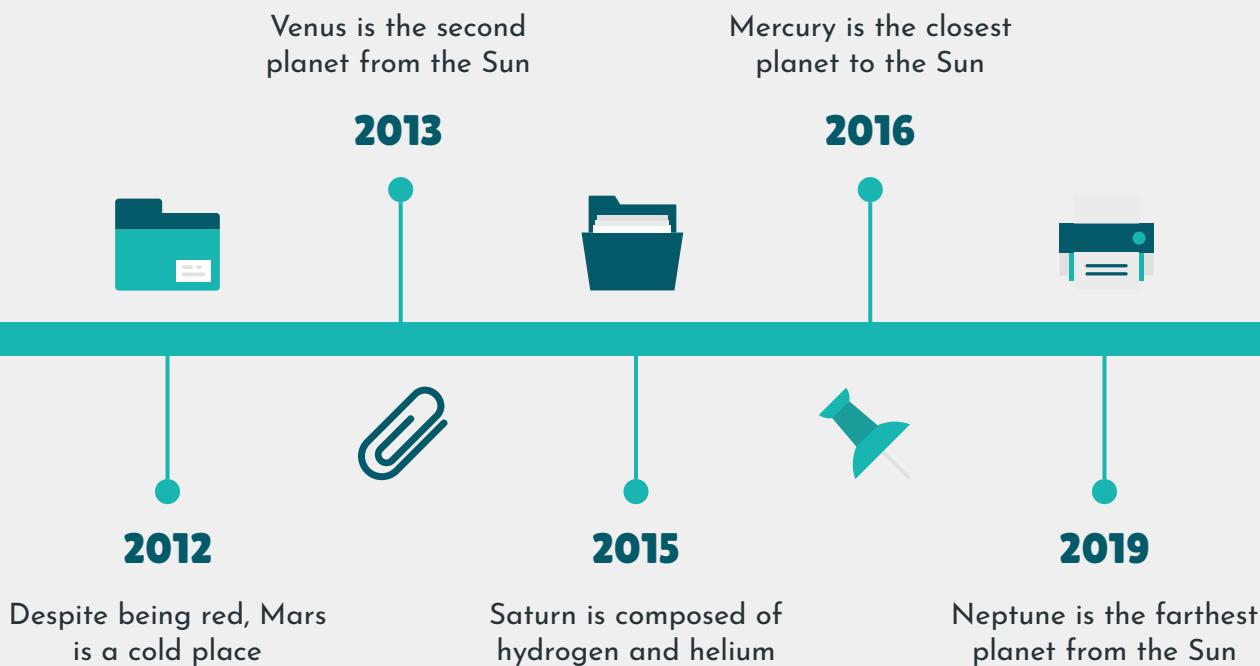
## Neptune

It's the farthest planet from the Sun

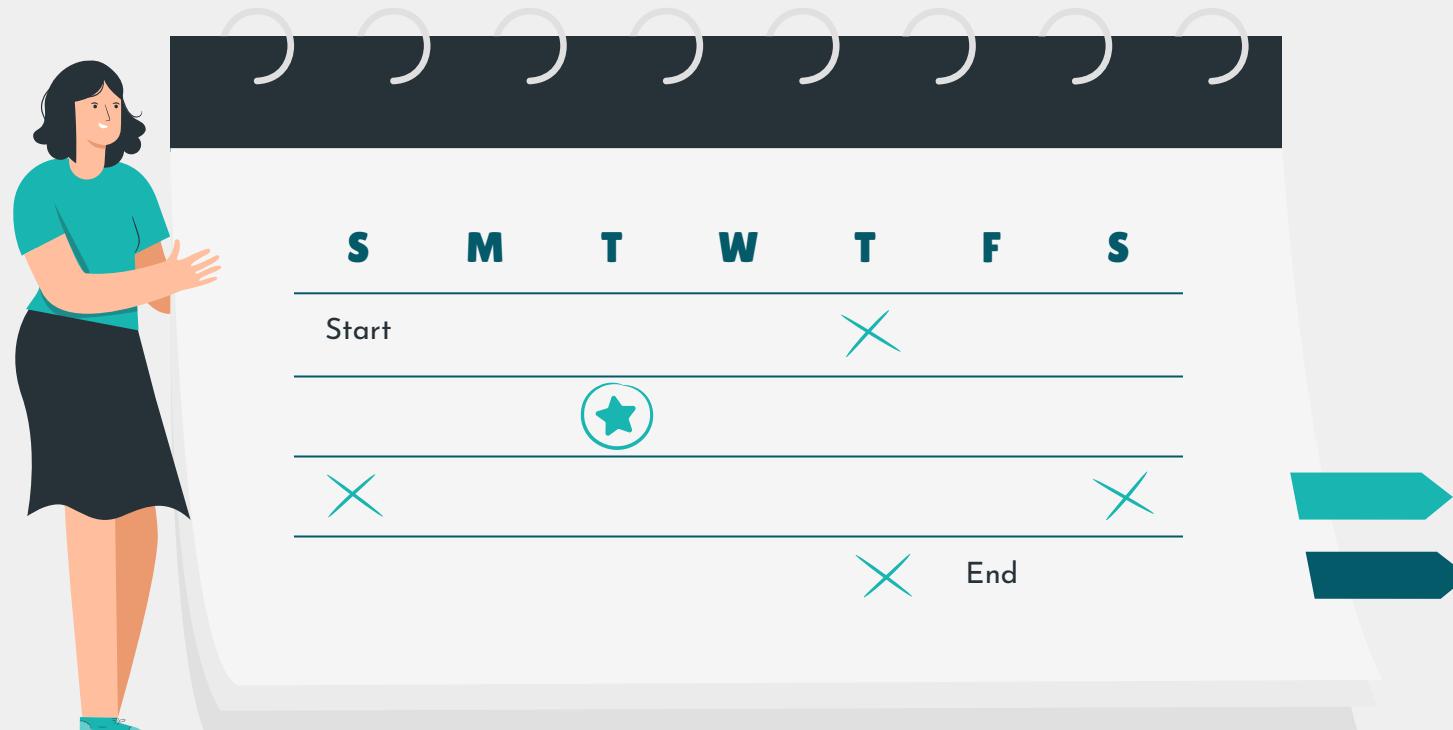
## Saturn

It's composed of hydrogen and helium

# TIMELINE



# TIMELINE





## OUR PARTNERS



### Venus

Venus is the second planet from the Sun



### Neptune

Neptune is the farthest planet from the Sun



### Saturn

Saturn is composed of hydrogen and helium

# OUR TEAM



## John Doe

You can replace the  
image on the screen  
with your own

## Jenna Doe

You can replace the  
image on the screen  
with your own

## Thomas Doe

You can replace the  
image on the screen  
with your own

# THANKS

Do you have any questions?

[youremail@freepik.com](mailto:youremail@freepik.com)

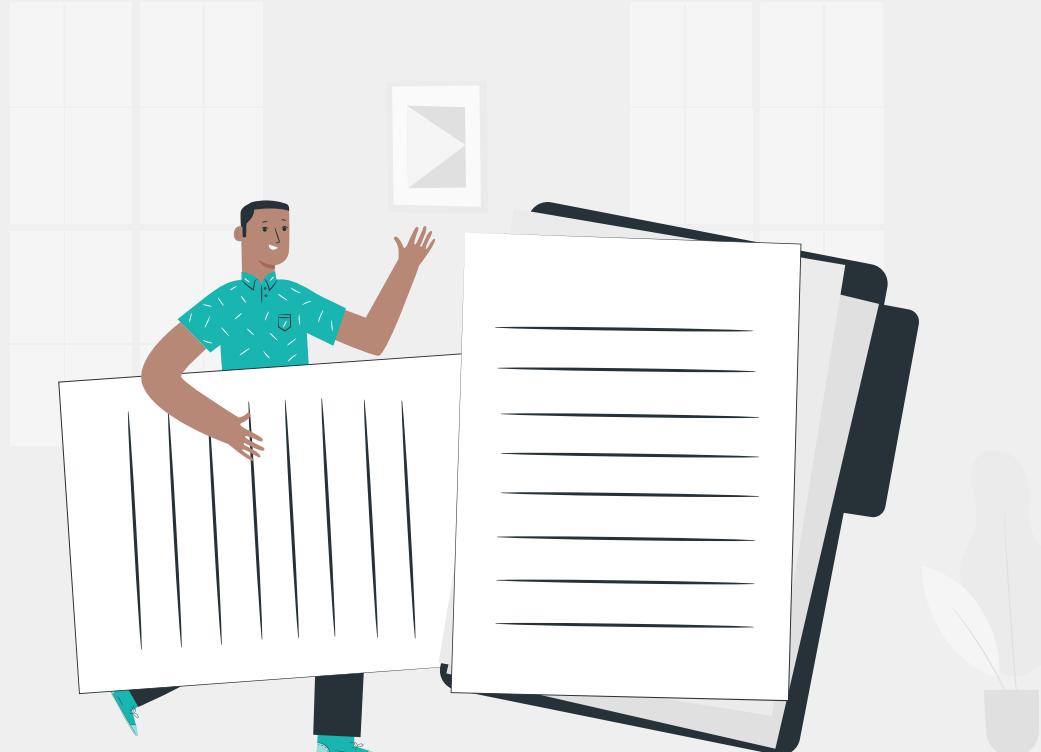
+91 620 421 838

[yourcompany.com](http://yourcompany.com)



CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik

Please keep this slide for attribution.



# ALTERNATIVE RESOURCES



# RESOURCES

## Vectors

- Development
- Calendar
- Charts
- Documents
- Hiring
- Collaboration
- Working
- Secure data
- Growing
- Images
- Devices

- Grades
- Search engine
- Connected
- Add files
- Notebook
- File searching
- Experts
- Resume folder
- Organizing projects
- Brainstorming
- Checklist

## Photos

- Book stack with copy space
- Colleagues looking charts

## Icons

- Essential collection icon pack



# Instructions for use

In order to use this template, you must credit **Slidesgo** by keeping the **Thanks** slide.

## You are allowed to:

- Modify this template.
- Use it for both personal and commercial projects.

## You are not allowed to:

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

<https://slidesgo.com/faqs> and <https://slidesgo.com/slidesgo-school>

# Fonts & colors used

This presentation has been made using the following fonts:

**Lilita One**  
(<https://fonts.google.com/specimen/Lilita+One>)

**Josefin Sans**  
(<https://fonts.google.com/specimen/Josefin+Sans>)

#045a68

#19b5b1

#263238

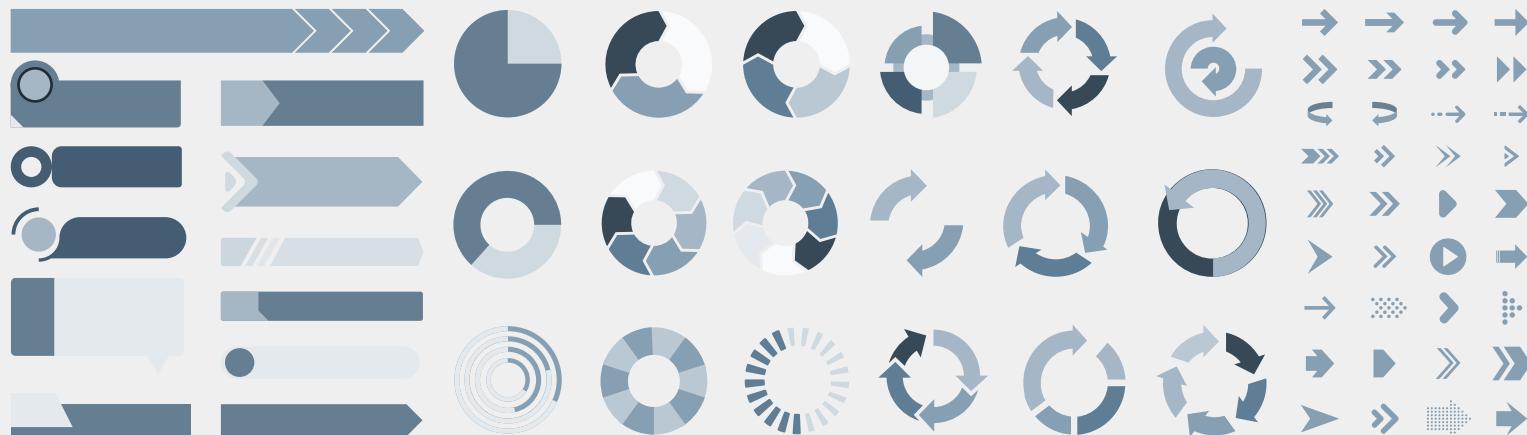
#ffbe9d

#ebebeb

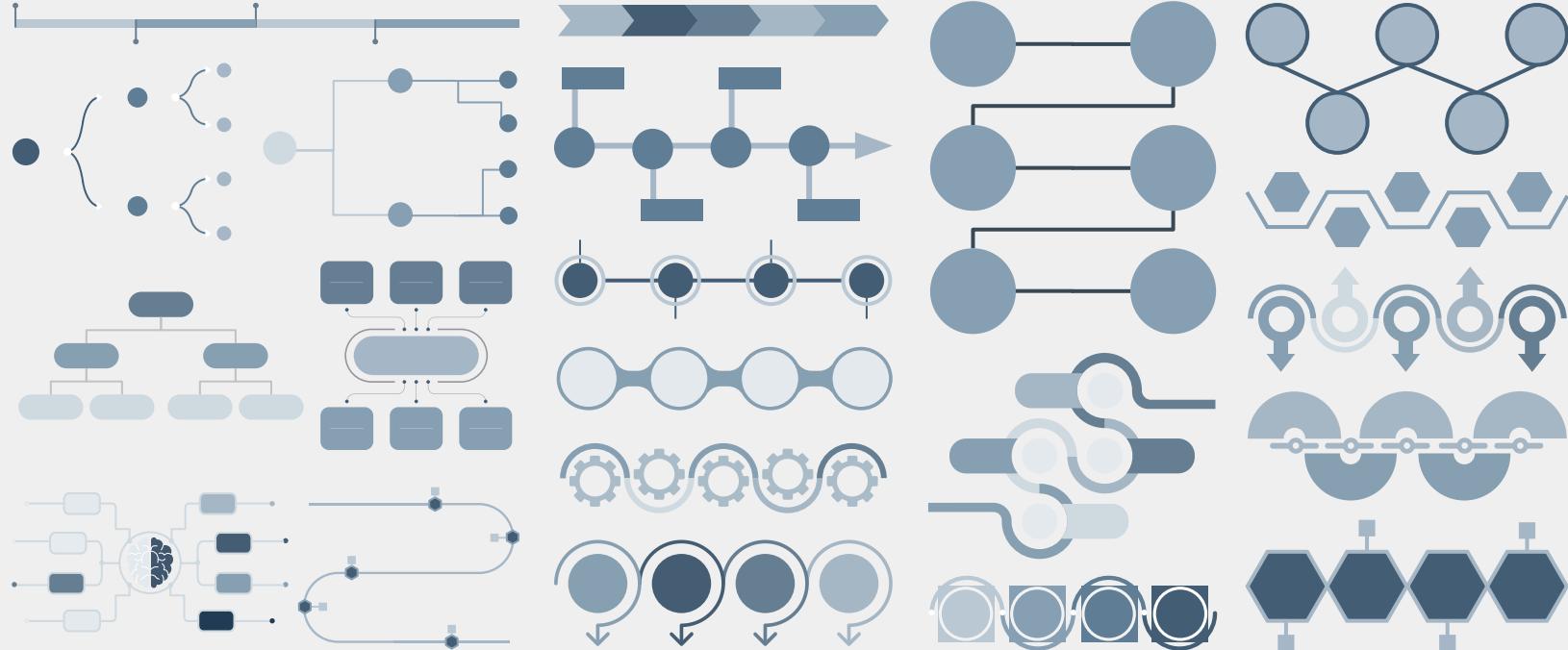
#e0e0e0

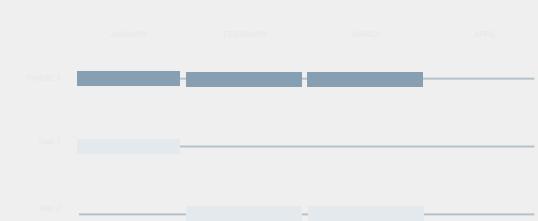
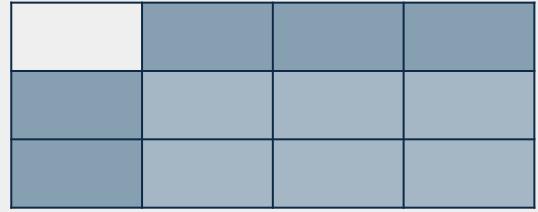
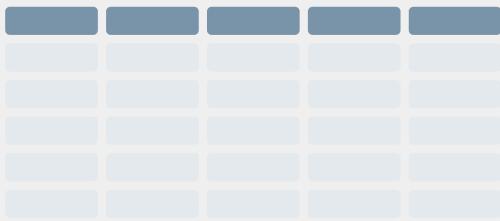
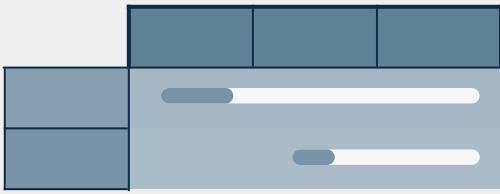
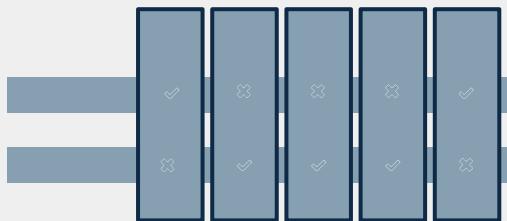
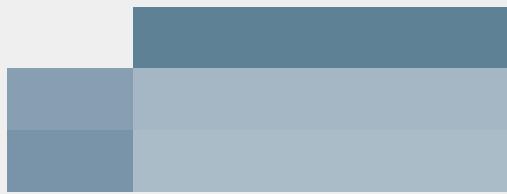
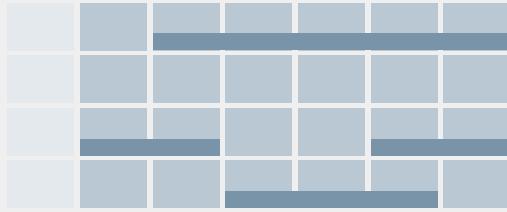
## Use our editable graphic resources...

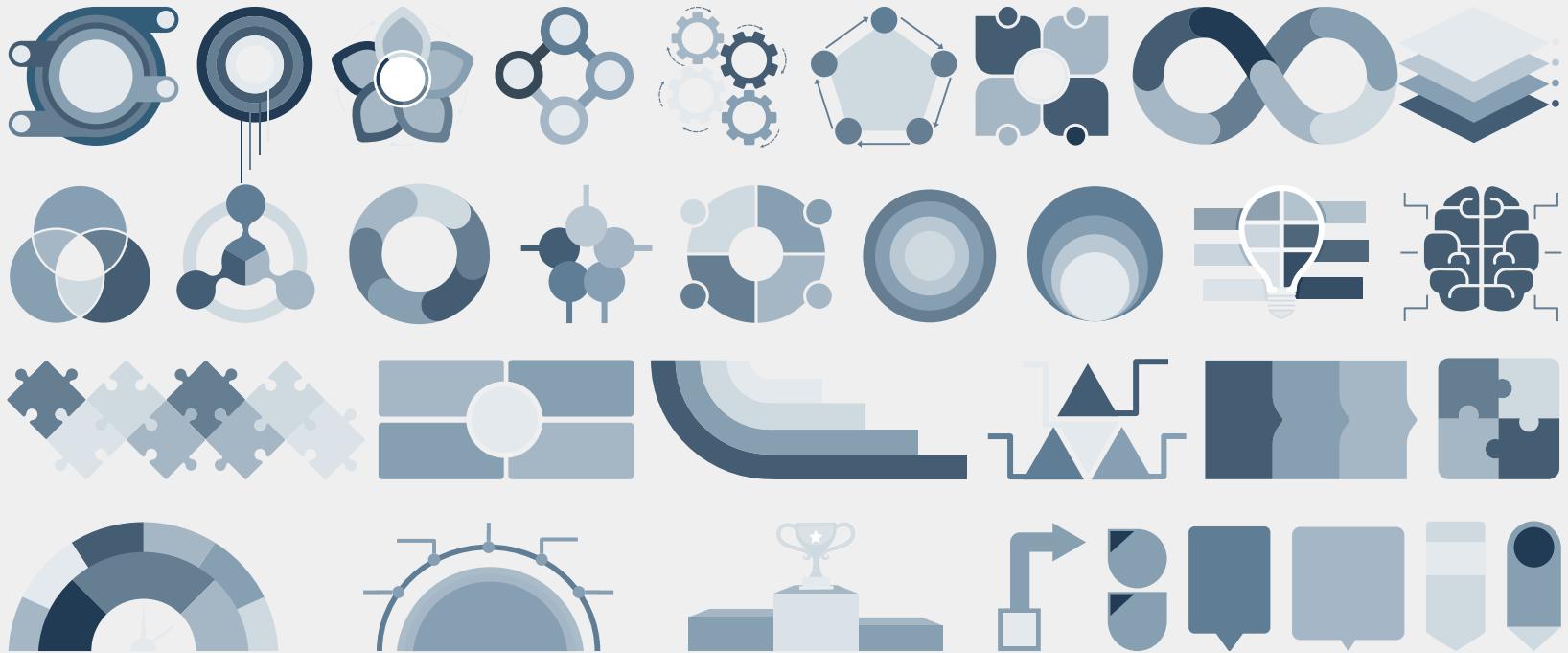
You can easily resize these resources, keeping the quality. To change the color, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want. Don't forget to group the resource again when you're done.

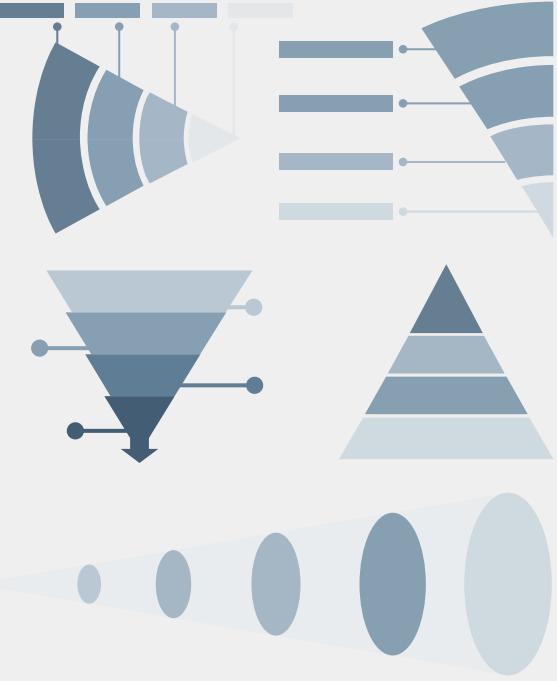
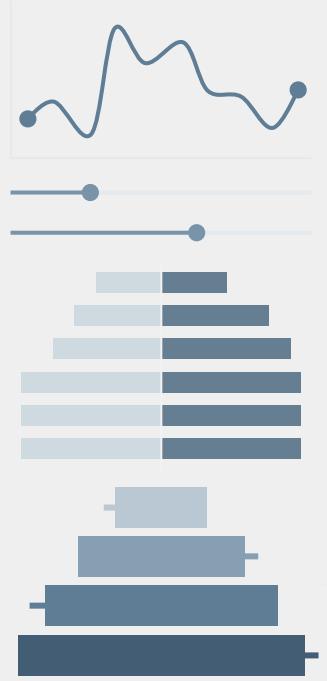
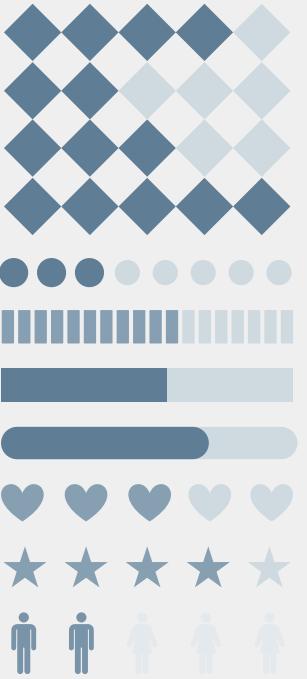












## ...and our sets of editable icons

You can resize these icons, keeping the quality.

You can change the stroke and fill color, just select the icon and click on the paint bucket/pen.

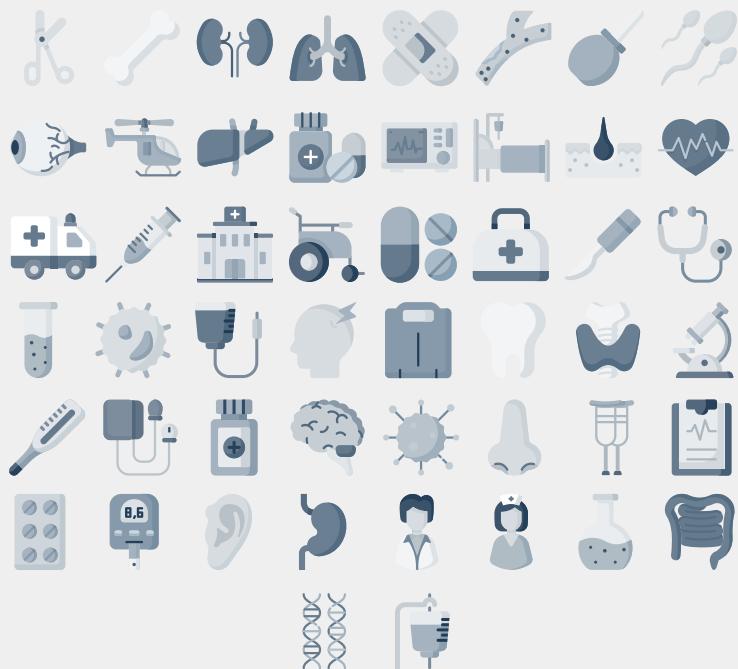
In Google Slides, you can also use Flaticon's extension, allowing you to customize and add even more icons.



## Educational Icons



## Medical Icons



## Business Icons



## Teamwork Icons



## Help & Support Icons



## Avatar Icons



## Creative Process Icons



## Performing Arts Icons



## Nature Icons



# SEO & Marketing Icons



