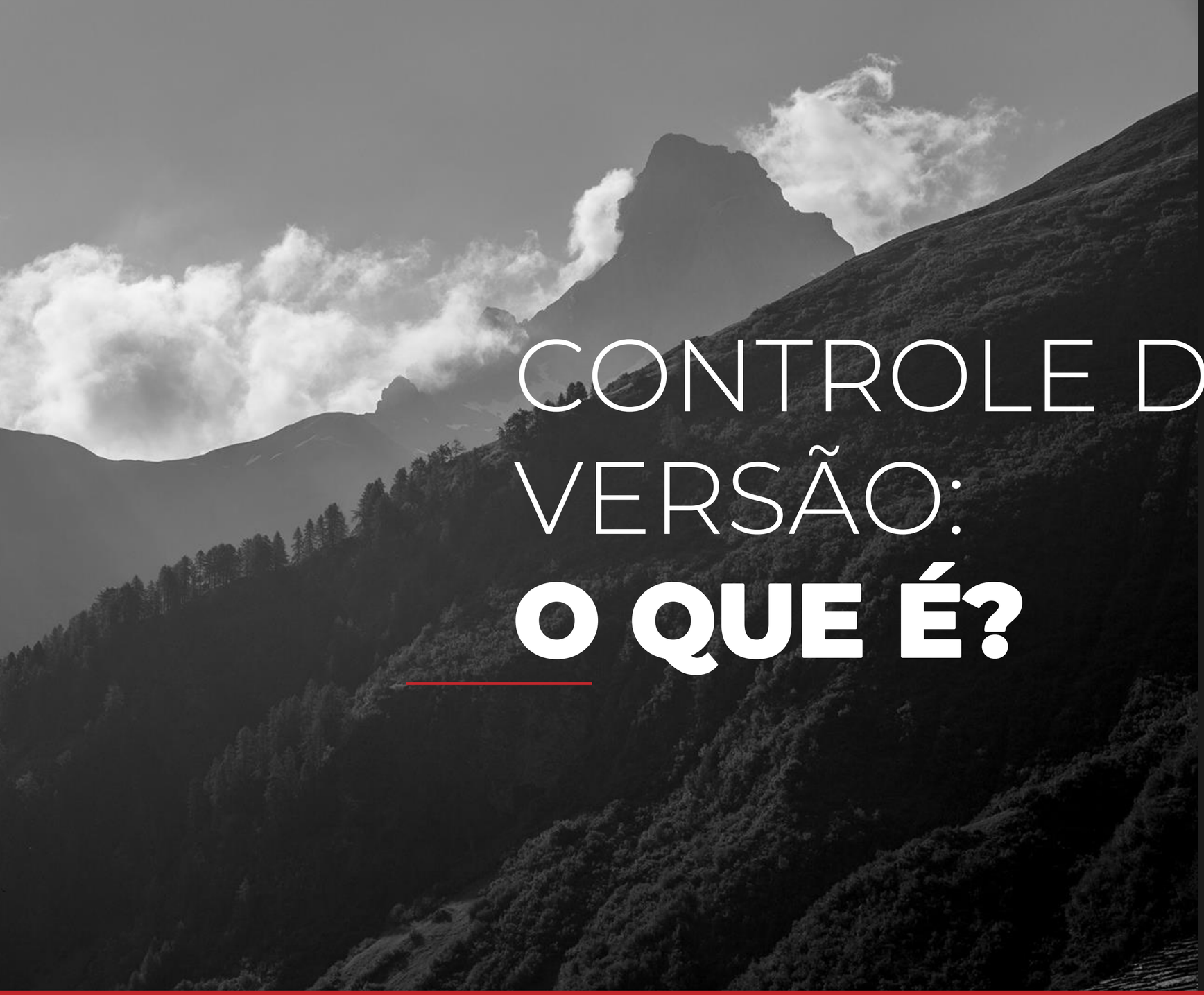

SENAI

CONTROLE DE VERSÃO
TÓPICOS ABORDADOS

- O que é um controle de versão.
- Git e GitHub: a diferença.
- Conexão e autenticação.
- Integração do Git e GitHub no IntelliJ.
  - Configurar o Git no IntelliJ.
  - Criar um novo repositório.
  - Conectar ao GitHub.
- Clonar um repositório remoto.
- Resolvendo problemas.



CONTROLE DE VERSÃO: **O QUE É?**

SENAI

CONTROLE DE VERSÃO
O QUE É?

- É a prática de rastrear e gerenciar mudanças em arquivos ao longo do tempo.
- Também conhecido como versionamento ou controle de fonte.

CONTROLE DE VERSÃO
OBJETIVO

- Permitir acompanhar a evolução do código.
- Facilitar a colaboração entre desenvolvedores.
- Garantir a integridade e a recuperação de projetos.

CONTROLE DE VERSÃO
BENEFÍCIOS

- Colaboração eficiente.
- Histórico completo das alterações.
- Backup seguro do código.
- Organização do projeto.

CONTROLE DE VERSÃO:
GIT E GITHUB

SENAI

CONTROLE DE VERSÃO
GIT

- **Ferramenta local:** É um software instalado no seu computador que permite você controlar as versões dos seus projetos.
- **Controle de versões:** Rastreia as mudanças nos seus arquivos, permitindo que você volte para versões anteriores, compare diferentes versões e veja o histórico de alterações.
- **Operações básicas:** Criar repositórios, adicionar arquivos, fazer commits, criar branches e mesclar mudanças.

CONTROLE DE VERSÃO
GITHUB

- **Plataforma online:** É um serviço baseado em nuvem que oferece hospedagem para seus repositórios Git.
- **Colaboração:** Permite que você compartilhe seus projetos com outras pessoas, colaborar em projetos em equipe e contribuir para projetos open source.

CONTROLE DE VERSÃO
A DIFERENÇA

- **Git:** É a **ferramenta** que você usa para controlar as versões dos seus projetos localmente.
- **GitHub:** É a **plataforma** que você usa para compartilhar seus projetos com o mundo e colaborar com outros desenvolvedores.

CONTROLE DE VERSÃO: **CONEXÃO E AUTENTICAÇÃO**

SENAI

CONTROLE DE VERSÃO

COMO CONECTAR O GIT COM O GITHUB

Tipos de autenticação para conexão entre git e github:

- **Nome de usuário e senha:** A forma mais básica de autenticação.
- **Tokens de acesso pessoal:** Uma string de caracteres que representa as credenciais do usuário.
- **Chaves SSH:** Um par de chaves criptográficas que permite a autenticação sem a necessidade de digitar a senha a cada conexão.

CONTROLE DE VERSÃO
CREDENCIAIS DO AUTOR

Identificando o autor do código.

- A primeira vez que você realiza um commit em um novo repositório, o **Git** precisa saber quem está realizando essa ação. Por isso, ele solicita o seu **nome** e **email**. Essas informações são armazenadas no arquivo de configuração local do repositório (**.git/config**) e são incluídas no cabeçalho de cada commit.
- Existe duas formas de armazenar essas credenciais:
  - De forma **global** e **local.**

CONTROLE DE VERSÃO
CREDENCIAIS **LOCAIS**

Armazenamento:

- As credenciais são armazenadas em um arquivo de configuração específico para cada repositório, geralmente **.git/config**.

Escopo:

- As credenciais são válidas apenas para aquele repositório em particular.

CONTROLE DE VERSÃO
CREDENCIAIS **LOCAIS**

Quando usar:

- **Múltiplos repositórios com contas diferentes:**
  - Se você trabalha com vários repositórios que exigem credenciais diferentes, configurar as credenciais localmente evita conflitos.
- **Projetos colaborativos:**
  - Quando você está colaborando em um projeto onde outras pessoas também têm acesso ao repositório, as credenciais locais ajudam a manter a privacidade das suas informações.

CONTROLE DE VERSÃO
CREDENCIAIS **GLOBAIS**

Armazenamento:

- As credenciais são armazenadas em um arquivo de configuração global do Git, geralmente em **~/.gitconfig.**

Escopo:

- As credenciais são válidas para todos os repositórios que você clona ou cria.

CONTROLE DE VERSÃO
CREDENCIAIS **GLOBAIS**

Quando usar:

- **Única conta para todos os repositórios:**
  - Se você utiliza a mesma conta para todos os seus repositórios, configurar as credenciais globalmente simplifica o processo de autenticação.
- **Projetos pessoais:**
  - Para projetos pessoais, onde você é o único usuário, as credenciais globais são convenientes.

CONTROLE DE VERSÃO

CONFIGURANDO AS CREDENCIAIS:

Comandos para a configuração das credenciais via linha de comando:

- **Global**

```
git config --global user.name "Seu Nome"
git config --global user.email "seuemail@email.com"
```

- **Local**

```
git config user.name "Seu Nome"
git config user.email "seuemail@email.com"
```

CONTROLE DE VERSÃO: GIT/GITHUB NO INTELLIJ

SENAI

CONTROLE DE VERSÃO
PASSO A PASSO PARA INTEGRAÇÃO

1. Habilite o **VCS** - Version Control System (Sistema de controle de versão) no **Intellij** para criar um repositório local do **git** no seu projeto.

Se o Git não estiver instalado: Você verá uma mensagem de erro indicando isso. Clique no link sugerido para baixar e instalar o Git. **Recomendamos instalar o Git manualmente** a partir do site oficial, pois a instalação via IntelliJ pode não funcionar em todos os casos.

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO
PASSO A PASSO PARA INTEGRAÇÃO

SENAI

2. Faça o primeiro **commit** do projeto para o **repositório local** do **git**.
3. Utilize as **credenciais** de forma **local**.

Cuidado ao marcar "*Set Properties globally*" no IntelliJ: Ao marcar essa opção, todos os projetos que você criar e commitar no Git terão suas credenciais associadas automaticamente. Para evitar isso, desmarque essa opção e preencha suas credenciais para cada projeto criado. **Utilize credenciais globais apenas em computadores pessoais**.

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO
GITHUB - TOKEN

4. Gere um **token** para facilitar a integração do **intellij** com o **github**. Esse token gerado não será mais exibido, por isso salve-o em um armazenamento remoto de sua preferência como **Onedrive** ou **Google Drive**.

Importante: Para facilitar os exercícios, estamos gerando um token de acesso do GitHub com permissões amplas. **No entanto, essa não é uma prática segura** em ambientes de produção. Ao trabalhar em projetos reais, siga as diretrizes da sua empresa e crie tokens com **permissões mínimas necessárias** e **data de expiração definida**. Isso limita o acesso de terceiros à sua conta em caso de comprometimento do token.

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO
PASSO A PASSO PARA INTEGRAÇÃO

6. Vincule sua conta do **github** no **intellij** utilizando o **token** gerado.

Alternativa: Autenticação via navegador. Você pode conectar sua conta do GitHub ao IntelliJ usando o navegador. Para isso basta selecionar a opção "***Log in via GitHub***". Essa opção abrirá uma janela do navegador para que você faça o login na sua conta do GitHub e autorize o IntelliJ a acessar seus repositórios. No entanto, essa opção mantém sua conta do GitHub logada e aberta no navegador. **Recomenda-se o uso de tokens de acesso** para maior segurança e praticidade, pois eles não exigem que você mantenha sua sessão do navegador aberta.

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO
PASSO A PASSO PARA INTEGRAÇÃO

7. Crie no **github,** através do **intellij,** um **repositório remoto** para seu projeto.

Tudo certo até aqui? Se você seguiu todos os passos e não houve conflito de credenciais, seu projeto já está no GitHub! Entre lá e confira! Caso contrário, um repositório vazio foi criado e você verá mensagens de erro no IntelliJ. Vamos abordar como resolver essas situações em um tópico futuro.

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO
PASSO A PASSO PARA INTEGRAÇÃO

8. Faça **commit** das atualizações do projeto nos **repositório local** e **remoto** utilizando o botão ***“Commit and Push”.***

Parabéns! Seu projeto já está no GitHub. Agora você pode adicionar novas funcionalidades e fazer commits regularmente. **Cada commit é como um ponto de restauração do seu projeto.** Adicione comentários claros para explicar as mudanças feitas. **A recomendação é fazer um commit sempre que finalizar uma tarefa ou antes de sair do trabalho.**

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO: **CLONAR PROJETOS DO GITHUB**

SENAI

CONTROLE DE VERSÃO
CLONAR PROJETOS DO GITHUB

SENAI

1. Navegue até sua conta do **github** e copie a **URL** do **repositório remoto** do projeto que deseja clonar.

Atenção: **você está usando um computador público!** Para proteger seus dados pessoais, utilize sempre a navegação anônima ou privada. Assim, suas informações de login não serão salvas no computador e outras pessoas não poderão acessá-las. Ao terminar, basta fechar a janela.

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO
CLONAR PROJETOS DO GITHUB

SENAI

2. No **intellij**, abra um projeto através do **VCS** usando a **URL** copiada do **repositório remoto**.

Clonando pela primeira vez? Se você ainda não conectou sua conta do GitHub ao IntelliJ, precisará copiar a URL do repositório e colar no local indicado como apresentado no vídeo a seguir. **Já fez a conexão?** Basta selecionar o repositório desejado na lista ao abrir um novo projeto via VCS.

*Clique no botão para ver o **vídeo** deste passo.*

CONTROLE DE VERSÃO: **RESOLVENDO PROBLEMAS**

SENAI

CONTROLE DE VERSÃO
CONFLITOS DE CREDENCIAIS

Definição:

- São situações em que o Git não consegue autenticar o usuário corretamente, impedindo ações como **push** e **pull**.

Exemplos:

- Usar a senha errada.
- Tokens errados ou desatualizados.
- Ter múltiplas contas em um mesmo serviço e confundir as credenciais.
- Problemas com chaves SSH, como permissões incorretas ou chaves não adicionadas ao agente SSH.

CONTROLE DE VERSÃO

POR QUE OS CONFLITOS OCORREM?

Causas:

- **Configurações incorretas:**
  - Nome de usuário, senha, tokens ou chave SSH configurados de forma errada**.**
- **Cache de credenciais:**
  - O Git pode armazenar credenciais incorretas em cache.
- **Problemas de conexão:**
  - Dificuldades em se conectar ao servidor remoto.

CONTROLE DE VERSÃO
IDENTIFICANDO E DIAGNOSTICANDO PROBLEMAS

Sintomas:

- Mensagens de erro como **"remote: Invalid username or password"** ou **"Permission denied"**.
- Solicitação constante de credenciais.
- Falha ao conectar ao repositório remoto.

CONTROLE DE VERSÃO
IDENTIFICANDO E DIAGNOSTICANDO PROBLEMAS

SENAI

Diagnóstico

- **Verificar as configurações:**
  - Verificar se o nome de **usuário** e **email** estão corretos nos arquivos de configuração.

Comando para a verificação global:

```
git config --global --list
```

Comando para a verificação local:

```
git config --list
```

CONTROLE DE VERSÃO
SOLUÇÕES PRÁTICAS

Verificar e corrigir erros de autenticação:

- **Login:**
  - Verifique se está logado corretamente em sua conta do github no intellij.
- **Token:**
  - Verifique se o token que você está utilizando está válido**. Caso precise,** gere um novo token com as permissões necessárias e guarde-o em local seguro e que seja fácil para seu acesso.

CONTROLE DE VERSÃO
SOLUÇÕES PRÁTICAS

Limpando o cache de credenciais:

- **Comando Global:**

```
git config --global --unset user.name
git config --global --unset user.email
```

- **Comando Local:**

```
git config --unset user.name
git config --unset user.email
```

- **Objetivo:**
  - ➢ Remover as credenciais armazenadas em cache, forçando o Git a solicitar as credenciais novamente.

CONTROLE DE VERSÃO

EXERCÍCIOS:

INTEGRAÇÃO COM **GIT E GITHUB:**

- INICIALIZAR UM REPOSITÓRIO GIT.
- COMMITAR AS ALTERAÇÕES.
- CONECTAR SEU REPOSITÓRIO LOCAL AO GITHUB.
- ENVIAR AS ALTERAÇÕES PARA O GITHUB.

SENAI

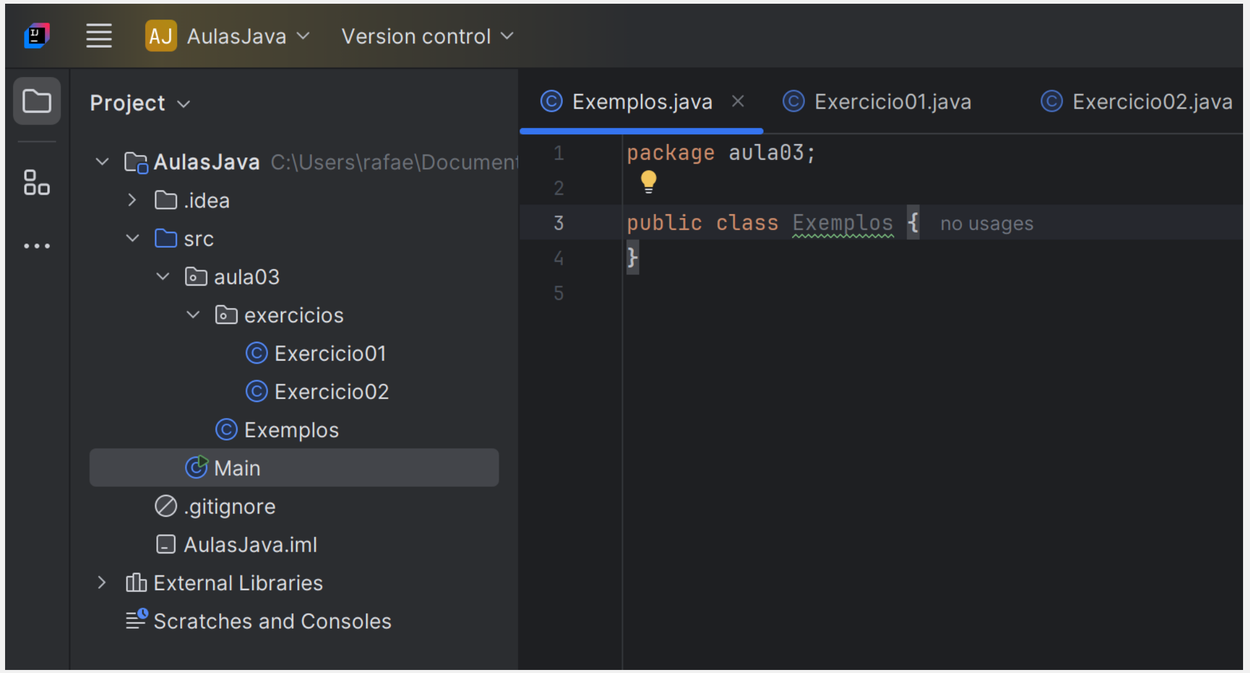
FUNDAMENTOS DE JAVA
EXERCÍCIOS

SENAI

Exercício 1:

Agora é hora de colocar a mão na massa!

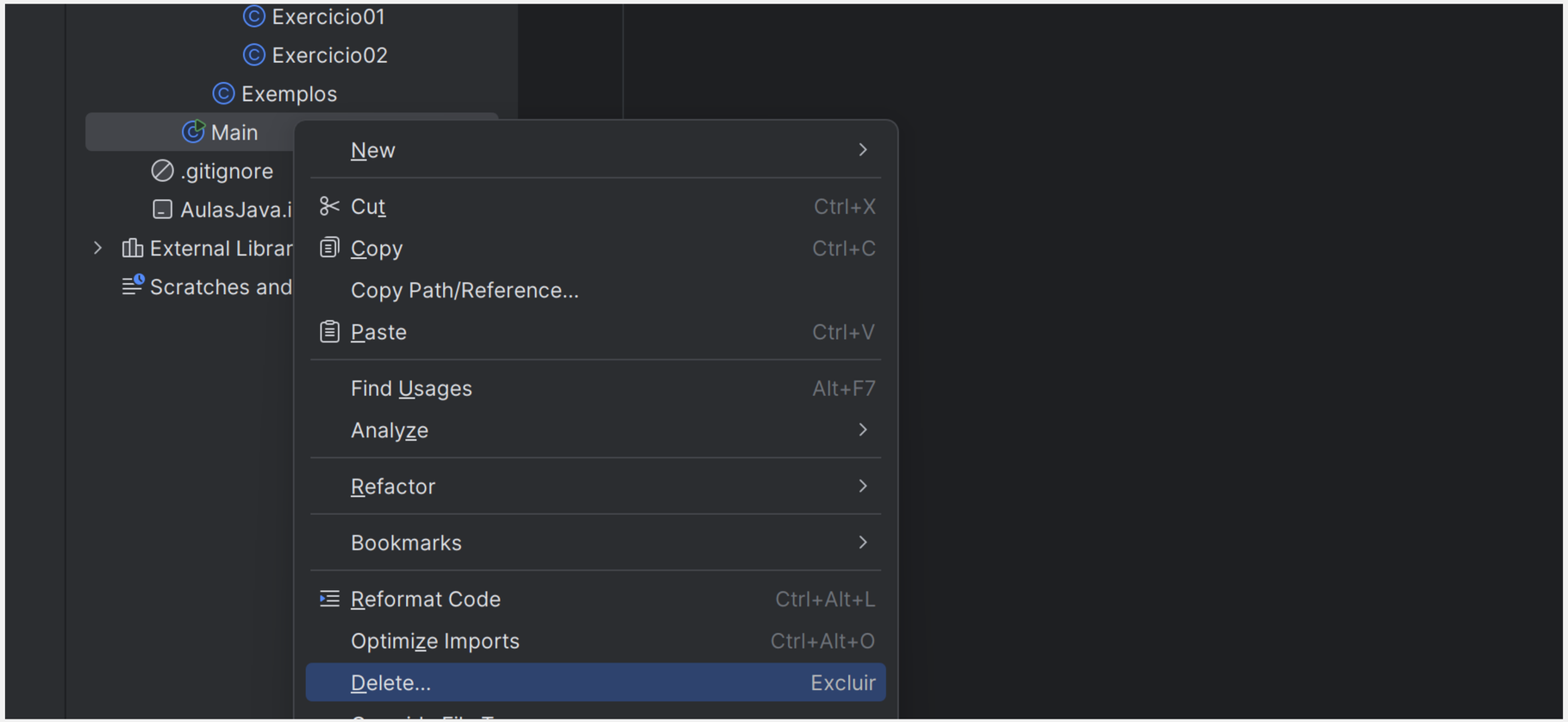
- Crie um novo projeto Java no IntelliJ chamado **AulasJava**, organize sua estrutura da seguinte forma:



FUNDAMENTOS DE JAVA
EXERCÍCIOS

Exercício 1:

- Exclua o arquivo **Main,** nós não iremos usá-lo.

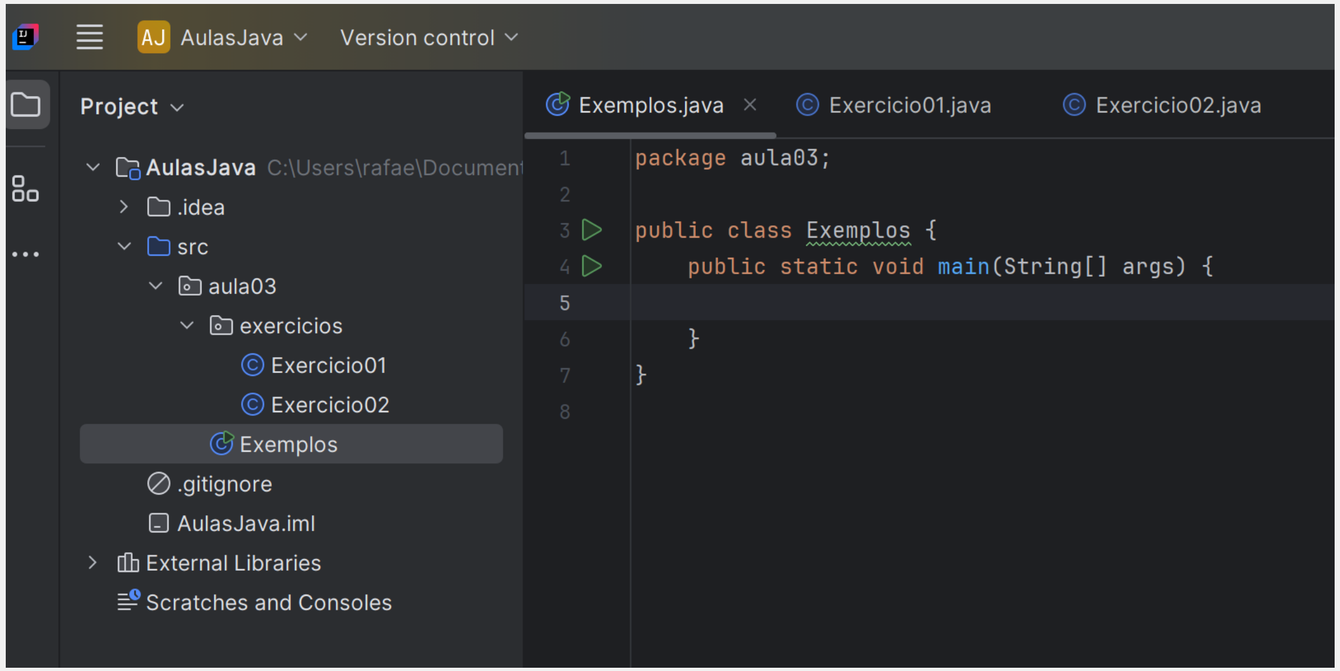


FUNDAMENTOS DE JAVA
EXERCÍCIOS

SENAI

Exercício 1:

- Em cada arquivo **.java** que criou, terá uma classe com o mesmo nome do arquivo, implemente o método **main** para testar seu código (comando **psvm**).



FUNDAMENTOS DE JAVA
EXERCÍCIOS

SENAI

Exercício 1:

Após configurar seu projeto, siga os passos aprendidos em aula para:

- Inicializar um repositório Git.
- Commitar as alterações no repositório local.
- Conectar seu repositório local ao GitHub:
- Enviar as alterações para o GitHub.

Verifique se seu código foi enviado corretamente acessando seu repositório no GitHub.

FUNDAMENTOS DE JAVA
CONSIDERAÇÕES FINAIS

O projeto **AulasJava** será nossa base para todas as aulas.
Manteremos a seguinte estrutura:

- **Um package por aula:** Cada aula terá sua própria pasta para organizar exemplos e exercícios.
- **Padrão de nomenclatura:** Seguiremos o mesmo padrão de nomes para arquivos e pastas em todas as aulas.

FUNDAMENTOS DE JAVA
CONSIDERAÇÕES FINAIS

SENAI

Seu fluxo de trabalho em cada aula:

1. **Clone do projeto:** No início de cada aula, faça um clone do repositório principal para seu computador.
2. **Desenvolvimento:** Crie seus exemplos e resolva os exercícios dentro da estrutura do projeto.
3. **Commits:** Faça commits frequentes, adicionando comentários claros sobre as alterações realizadas.
4. **Push:** No final da aula, envie suas alterações para o repositório remoto.

FUNDAMENTOS DE JAVA
CONSIDERAÇÕES FINAIS

Por que isso é importante?

- **Organização:** A organização do projeto facilita a revisão e a colaboração.
- **Aprendizado:** A prática regular do Git reforça os conceitos aprendidos.
- **Avaliação:** Sua participação e organização do projeto serão avaliadas.

Ao final de cada módulo, eu pedirei para você compartilhar comigo o link do seu repositório para avaliação.


SENAI
DEPARTAMENTO REGIONAL
DE SÃO PAULO
www.sp.senai.br