


# Quem sou eu?



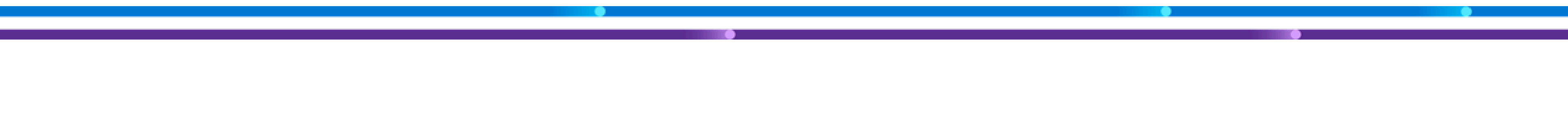
**Renicius Pagotto Fostaini**

- Porto Feliz, SP
- Graduado em Ciência da Computação
- Senior Infrastructure Engineer | Thoughtworks
- Certificações em Azure e GitHub
- Microsoft MVP





# GitHub Codespaces



# Codespace



É um ambiente de desenvolvimento completo hospedados na nuvem que permitem codificar diretamente do navegador, eliminando a necessidade de configuração local e oferecendo acesso instantâneo a todos os recursos do repositório.



# Codespaces | Características

- **Ambientes pré-configurados** diretamente dos repositórios com todas as ferramentas necessárias
- **Infraestrutura em nuvem** eliminando a necessidade de configurações locais
- **Acessibilidade remota** através de qualquer dispositivo com conexão à internet
- **Compatibilidade com editores** como VS Code web, desktop ou JupyterLab

# Codespaces | Ciclo de Vida



- O ciclo de vida começa quando você cria um Codespace e termina quando você exclui
- Criação, Reconstrução, Parada e Exclusão



# Codespace | Ciclo de Vida



- Criação a partir de um repositório
  - <https://github.com/codespaces>
- Timeout de 30 min por padrão, porém configurável
- Auto-save habilitado por padrão para web
- Alterações não serão perdidas em caso de perda de conexão
- Durante exclusão, alterações pendentes não serão salvas





# GitHub Dev Editor



# github.dev editor



- Editor leve que roda inteiramente no navegador
- Gratuito para todos os usuários do GitHub.com
- Permite navegar, editar e fazer commits em repositórios





# github.dev editor



- Pressione "." enquanto navega em um repositório ou PR
- Pressione ">" para abrir em nova aba
- Mude a URL de "github.com" para "github.dev"



# github.dev editor | Limitações



- Sem terminal integrado
- Sem capacidade de compilar ou executar código
- Apenas extensões web são suportadas
- Dados salvos localmente até o commit



# github.dev editor vs Codedspace



- GitHub.dev: gratuito, inicia instantaneamente, sem terminal
- Codespaces: VM dedicada, ambiente completo, terminal integrado





# Demo





# Autenticação, Segurança e Contas



# Two-Factor Authentication

- 2FA (Two-Factor Authentication):
  - Camada extra de segurança que exige dois métodos de verificação.
- Processo de Configuração:
  - Aplicativo autenticador (recomendado), Chave de segurança, GitHub Mobile entre outros.
- Preparação para Recuperação:
  - Salve os códigos de recuperação em local seguro.
- Por que é Importante:
  - Protege contra acessos não autorizados.
  - Obrigatório para todos os contribuidores do GitHub.

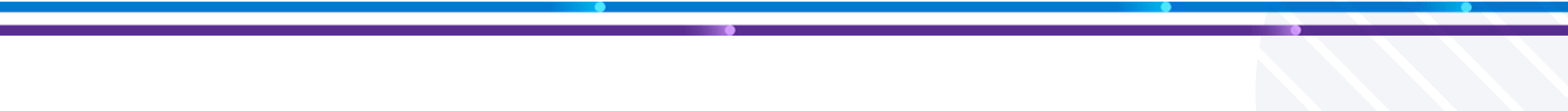
# Permissões de Acesso no GitHub

- Níveis de Acesso:
  - Read, Triage, Write, Maintain, Admin.
- Tipos de Contas:
  - Repositórios pessoais, Organizations, Enterprise.
- Permissões por Função:
  - Diferentes capacidades para cada nível (criar branches, aprovar PRs, gerenciar configurações).
- Importância:
  - Controle de acesso para proteger recursos e código.

# Enterprise Managed Users (EMUs)



- Definição:
  - Contas gerenciadas centralmente via provedor de identidade (IdP).
- Principais vantagens:
  - Gerenciamento centralizado de usuários.
  - Autenticação via IdP externo.
  - Restrição de conteúdo público.
- Disponibilidade:
  - Exclusivo para GitHub Enterprise Cloud.







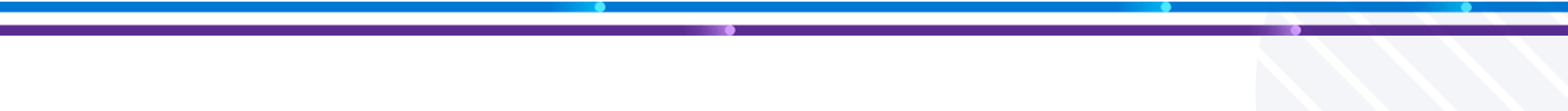
# Administração



# Administração | Organização



- Conta compartilhada no GitHub para colaboração em equipe com gerenciamento avançado de acesso e recursos
- Adicione colaboradores através das Configurações do repositório
- Atribua níveis de permissão apropriados com base nas necessidades
- Gerencie acesso em nível individual ou de equipe
- Funções Principais:
  - Owner (controle completo), Member (padrão), Team maintainer
- Equipes e Hierarquia:
  - Grupos que refletem a estrutura organizacional com permissões em cascata
  - Equipes aninhadas: Equipes pai passam permissões para equipes filhas



# Administração | Permissão e Visibilidade

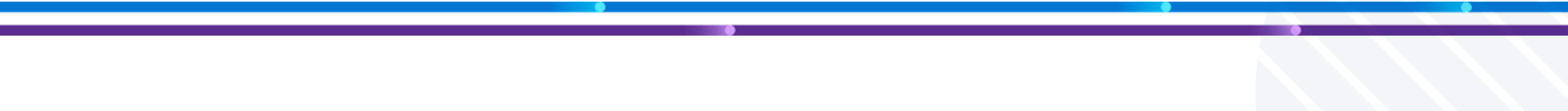


- Habilitando e Desabilitando Recursos
  - Segurança
  - Actions
- Níveis de Permissão de Repositório
  - Contas pessoais: Proprietário do repositório e Colaboradores
  - Contas de organização: Admin, Maintain, Write, Triage, Read
  - Níveis de permissão podem ser atribuídos a indivíduos ou equipes
- Opções de Visibilidade de Repositório
  - Público: Visível para todos na internet
  - Privado: Visível apenas para você e pessoas com quem compartilhar explicitamente
  - Interno: Visível para membros da sua empresa

# Administração | Repositórios



- Proteções de branch:
  - Previnem pushes forçados, exigem revisões específicas
- CODEOWNERS:
  - Designam proprietários para seções específicas de código que devem revisar alterações
- Revisores obrigatórios:
  - Exigem revisão de código de pessoas específicas antes da mesclagem
- Edite regras de proteção de branch para habilitar "Exigir revisão dos Proprietários do Código"
- Insights do Repositório



# Administração | Segurança

- Alertas do Dependabot:
  - Notificam sobre vulnerabilidades em dependências
- Varredura de segredos:
  - Detecta segredos acidentalmente commitados nos repositórios
- Proteção de push\*:
  - Bloqueia pushes contendo segredos
- Varredura de código\*:
  - Identifica vulnerabilidades e erros no código
- Gerenciamento de políticas e configurações de segurança

\* Apenas GitHub Advanced Security

# Obrigado

