**FACULDADE DE TECNOLOGIA SENAC GOIÁS**

**Gestão de Tecnologia da Informação**

LogoSenac

**GERÊNCIA DE CONFIGURAÇÃO**

Componentes: **Victor Hugo P. Costa,**

Goiânia,

2016

**GERÊNCIA DE CONFIGURAÇÃO**

Trabalho apresentado como requisito parcial para obtenção de aprovação na disciplina Auditoria e Qualidade de Software, no Curso de Gestão de Tecnologia da Informação, na Faculdade de Tecnologia Senac Goiás.

Professor Elias Ferreira.

Goiânia,

2016

**SUMÁRIO**

[O que é a gerência de configuração? 4](#_Toc446261585)

[O que são itens de configuração? 4](#_Toc446261586)

[Resultados esperados de acordo com a MPS.BR 5](#_Toc446261587)

[Processo de Gerência de Configuração 6](#_Toc446261588)

[GitHub 8](#_Toc446261589)

[Bibliográfica 16](#_Toc446261590)

**Gerência de Configuração**

# O que é a gerência de configuração?

Os sistemas de software estão em constante evolução. A manutenção do software, isto é, modificações em artefatos existentes, chega a consumir 75% do custo total do seu ciclo de vida. Aproximadamente, 20% de todo o esforço de manutenção é usado para consertar erros de implementação e os outros 80% são utilizados na adaptação do software em função de modificações em requisitos funcionais, regras de negócios e na reengenharia da aplicação. A Gerência de Configuração de Software surgiu da necessidade de controlar estas modificações, por meio de métodos e ferramentas, com o intuito de maximizar a produtividade e minimizar os erros cometidos durante a evolução.

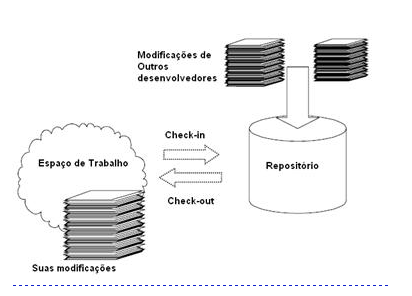
É uma disciplina que controla e notificam as inúmeras correções, extensões e adaptações aplicadas durante o ciclo de vida do software de forma a assegurar um processo de desenvolvimento e evolução sistemático e rastreável, sendo indispensável quando equipes manipulam, muitas vezes em conjunto, artefatos comuns.

Apesar de existir um forte apelo para o uso da Gerência de Configuração de Software durante a etapa de manutenção, a sua aplicação não se restringe somente a essa etapa do ciclo de vida do software. O uso dos sistemas de Gerência de Configuração é fundamental para prover controle sobre os artefatos produzidos e modificados por diferentes recursos desde o planejamento e levantamento de requisitos até a construção e entrega do produto. O motivo da sua importância está geralmente associado aos problemas identificados quando a Gerência de Configuração não é utilizada no desenvolvimento de software.

# O que são itens de configuração?

A melhor forma de explicar é citando exemplo, então vamos lá.

O sistema de controle de versões (**Subversion**, por exemplo) permite que os artefatos sejam obtidos, por meio de uma operação conhecida como check-out, modificados dentro do espaço de trabalho do desenvolvedor e, depois, retornados ao repositório, por meio de uma operação conhecida como check-in, como exemplifica a Figura 1. O repositório é o local de armazenamento dos artefatos que estão sob controle da Gerência de Configuração de Software. **Estes artefatos recebem o nome de itens de configuração.** A cada operação de check-in realizada, a versão do item de configuração é incrementada de uma unidade. Quando o item é adicionado pela primeira vez no repositório, este item passa a ter a versão igual a 1. Para cada item de configuração armazenado, são anexadas informações como: datas da criação ou alteração, comentários e versões.



Neste cenário, não há perdas ou sobreposições porque políticas de trabalho foram estabelecidas, restringindo ou controlando as modificações no repositório. As ferramentas de controle de versões normalmente suportam a definição de diferentes políticas de trabalho. Dentre essas políticas, podemos citar a política pessimista, que enfatiza o uso de check-out reservado, fazendo bloqueio e inibindo o paralelismo do desenvolvimento sobre o mesmo artefato.

# Resultados esperados de acordo com a MPS.BR

São 7 (sete) no total.

1. GCO 1 – Um Sistema de Gerência de Configuração é estabelecido e mantido.
2. GCO 2 – Os itens de configuração são identificados com base em critérios estabelecidos.
3. GCO 3 – Os itens de configuração sujeitos a um controle formal são colocados sob baselines.
4. GCO 4 – A situação dos itens de configuração e das baselines é registrada ao longo do tempo e disponibilizada.
5. GCO 5 – Modificações em itens de configuração são controladas.
6. GCO 6 – O armazenamento, o manuseio e a liberação de itens de configuração e baselines são controlados.
7. GCO 7 – Auditorias de configuração são realizadas objetivamente para assegurar que as baselines e os itens de configuração estejam íntegros, completos e consistentes.

# Processo de Gerência de Configuração

Do ponto de vista gerencial, o processo de Gerência de Configuração de Software é dividido em cinco funções: **identificação da configuração, controle da configuração, acompanhamento da situação da configuração, auditoria da configuração e gerenciamento de entrega**.

A função de **identificação da configuração** tem por objetivo: (1) a seleção de quais artefatos serão itens de configuração; (2) a definição de uma nomenclatura, que possibilite a identificação inequívoca dos itens de configuração, baselines e releases; e (3) a descrição dos itens, tanto física quanto funcionalmente.

A seleção de itens de configuração é feita no início da fase de planejamento e leva em conta: (1) se o artefato é crítico para o projeto; (2) a dependência entre artefatos; (3) o impacto que uma modificação do item tem no produto; (4) se o artefato pode ser modificado por dois ou mais grupos; (5) se é frequentemente alterado devido a sua complexidade e (6) se é gerado manualmente, automaticamente ou ambos.

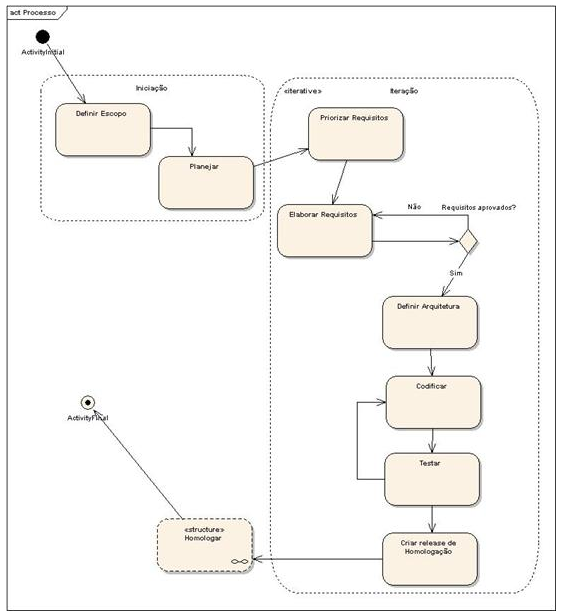
A função de **controle da configuração** é designada para controlar e acompanhar a evolução dos itens de configuração selecionados na função de identificação. Ferramentas como JIRA, Bugzilla, dentre outras, apoiam, em conjunto com as ferramentas de controle de versões, as atividades desta função.

A função de **acompanhamento da situação da configuração** visa: (1) armazenar as informações geradas pelas demais funções; e (2) permitir que essas informações possam ser acessadas em função de necessidades específicas, por exemplo, para a melhoria do processo, para a estimativa de custos futuros e para a geração de relatórios gerenciais. Estas informações podem ser obtidas, no decorrer do projeto, a partir dos sistemas de controle de versões e modificações.

A função de **auditoria da configuração** ocorre geralmente quando uma release deve ser criada. Suas atividades compreendem: (1) auditoria funcional, que abrange a revisão dos planos, dados, metodologia e resultados dos testes, assegurando que a release cumpre corretamente o que foi especificado; e (2) auditoria física, com o objetivo de certificar que a release é completa em relação ao que foi acertado em cláusulas contratuais. A auditoria pode ser feita com base nos relatórios obtidos na função anterior.

Já a função de **gerenciamento de liberação e entrega** descreve o processo formal de: (1) construção e liberação de uma release do produto; e (2) entrega, com informações de como implantar o software no ambiente final de execução. Ferramentas, como Ant, permitem que roteiros de construção sejam escritos e executados no apoio a esta fase.

Para exemplificar, vamos considerar um ciclo de vida iterativo e incremental que propõe inicialmente a identificação do escopo do projeto e a aprovação deste escopo pelo cliente. Posteriormente, nesta abordagem, o software é construído em ciclos sucessivos denominados iterações. A cada iteração, os requisitos são priorizados, detalhados, aprovados e o software são modelados, construído e testado, como ilustra a **Figura 2**. Ao final, um release do produto é entregue para homologação e aprovação do cliente.



Podemos citar como exemplos de ferramentas de mercado: CVS, Subversion, IBM Rational ClearCase e Microsoft Visual Source Safe.

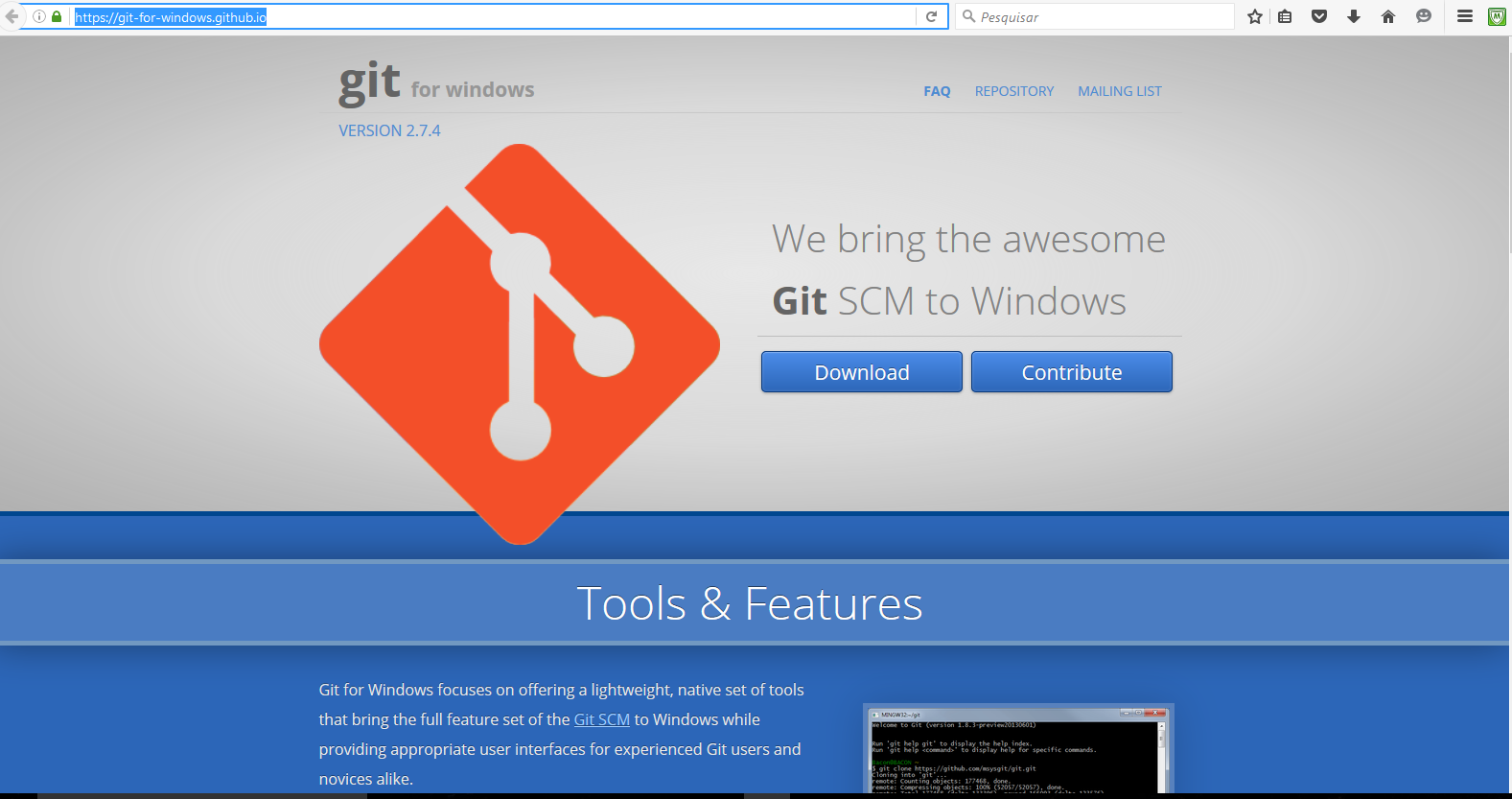
# GitHub

GitHub é um Serviço de Web Hosting Compartilhado para projetos que usam o controle de versionamento Git. É escrito em Ruby on Rails pelos desenvolvedores da Logical Awesome (Chris Wanstrath, PJ Hyett e Tom Preston - Wernder). O GitHub possui planos comerciais e gratuitos para projetos de código aberto.

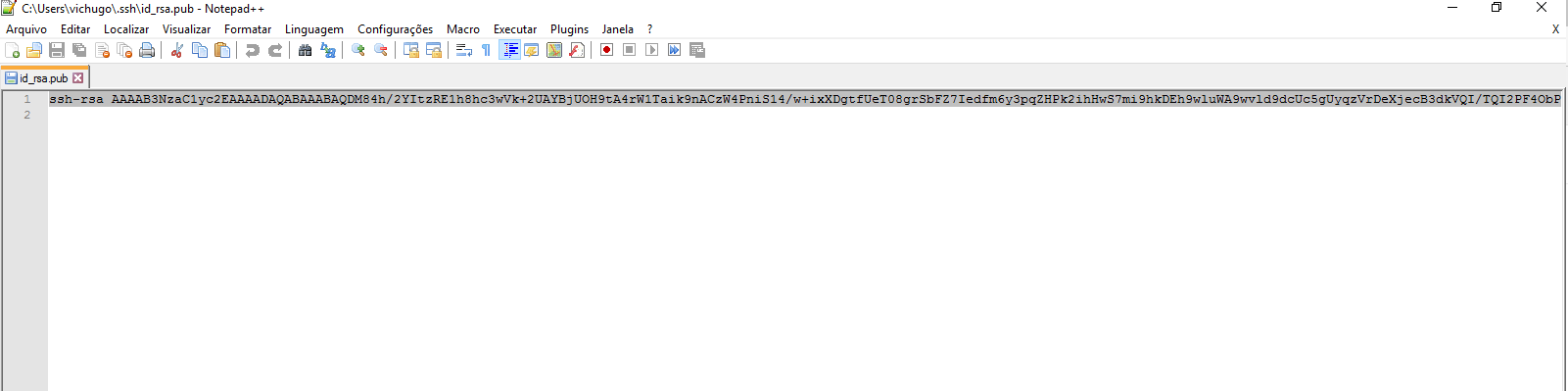
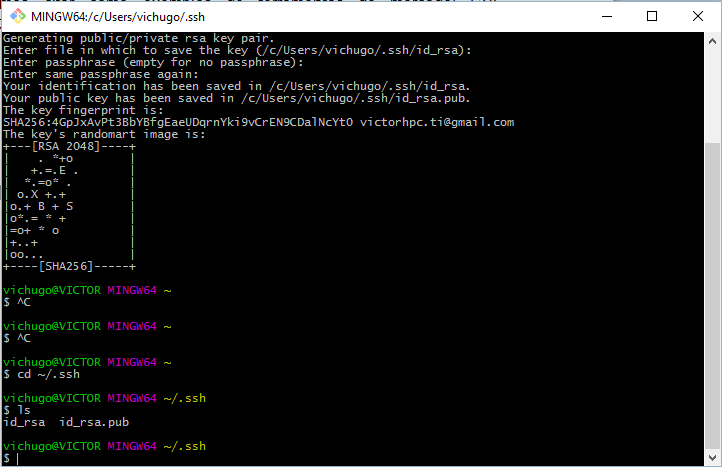
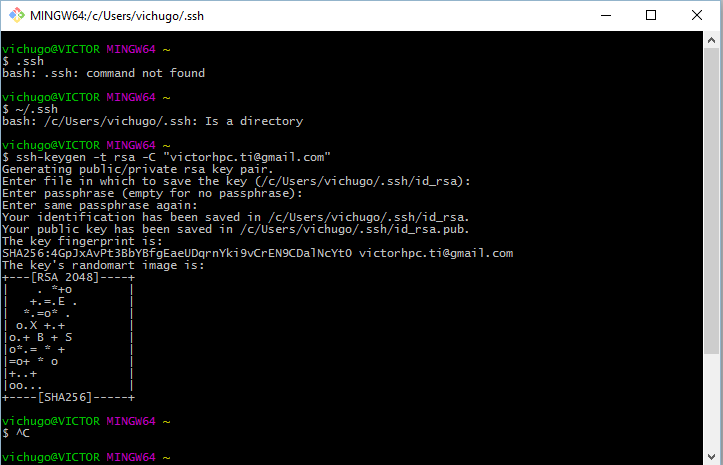
Este site possui funcionalidades de uma rede social como feeds, followers, wiki e um gráfico que mostra como os desenvolvedores trabalham as versões de seus repositórios.

Passos para utilizar.

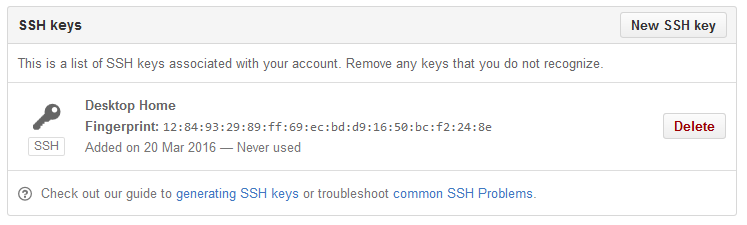
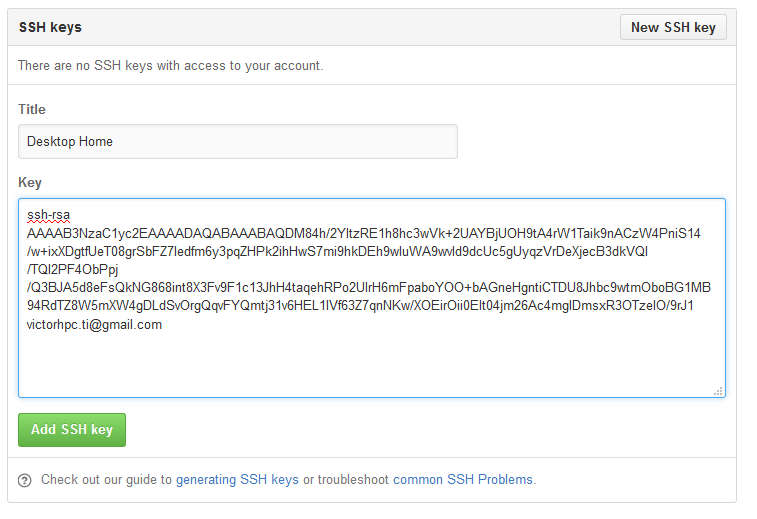
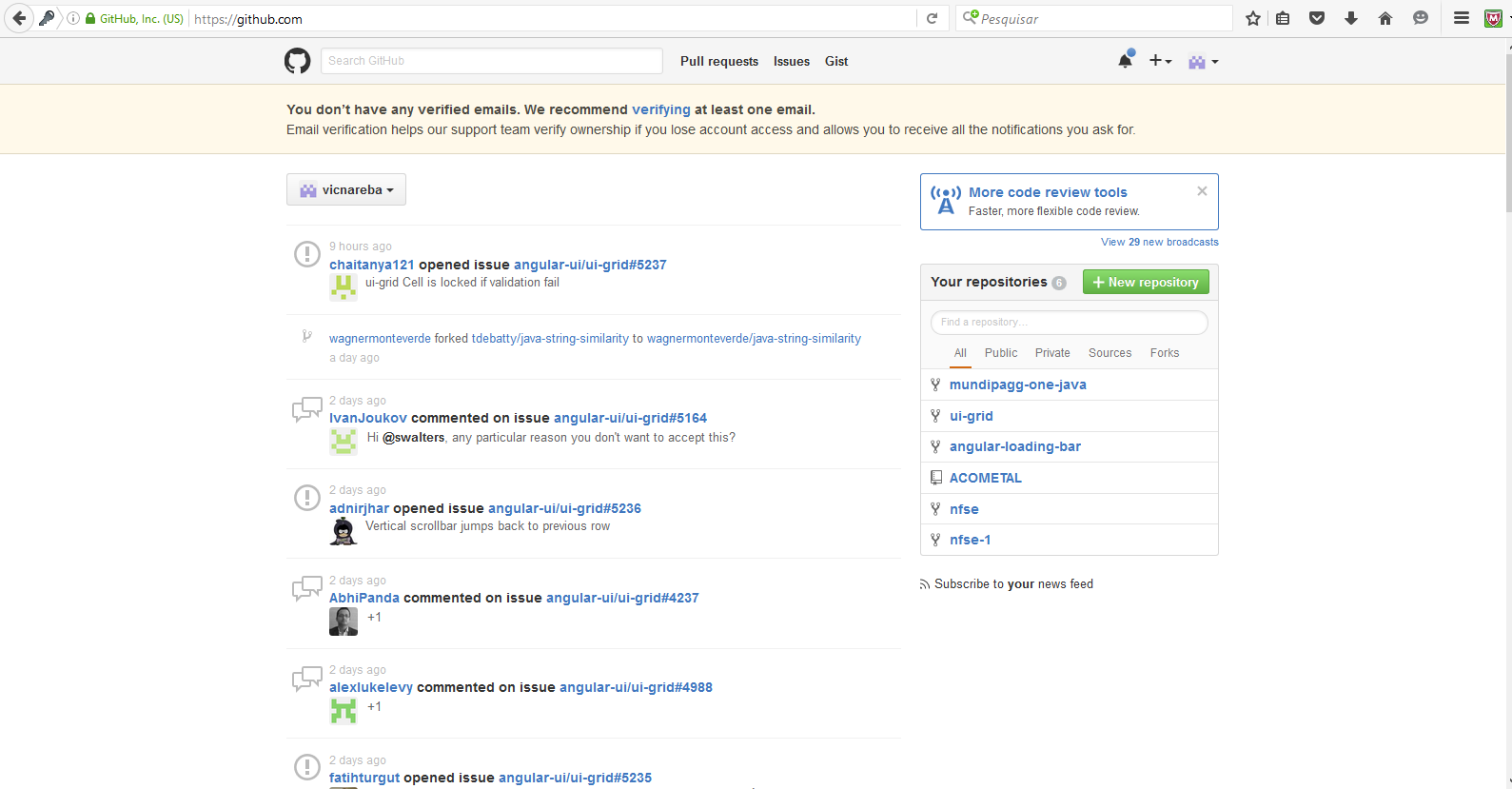
Primeiramente, baixa-lo e instalar utilizando o default.



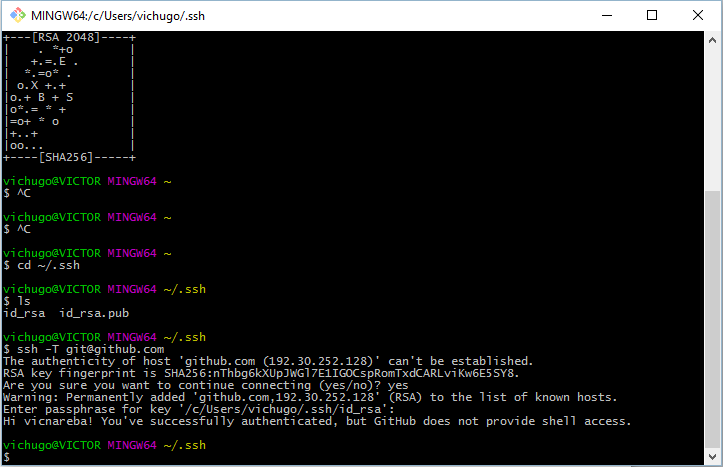
Abrir o GitBash e gerar a chave pública SSH apontando para a conta que você criará no GitHub.



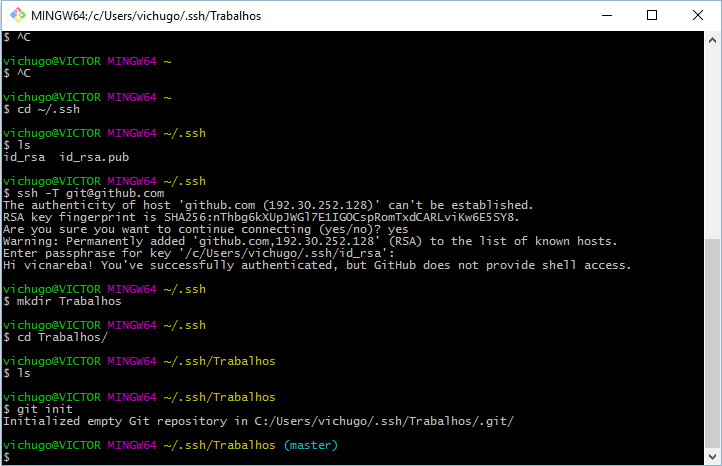
Após feito isso, acessar a sua conta GitHub e ir em Settings/SSH Keys. Em seguida, criar um novo SSH Key e adicionar a Key gerada.



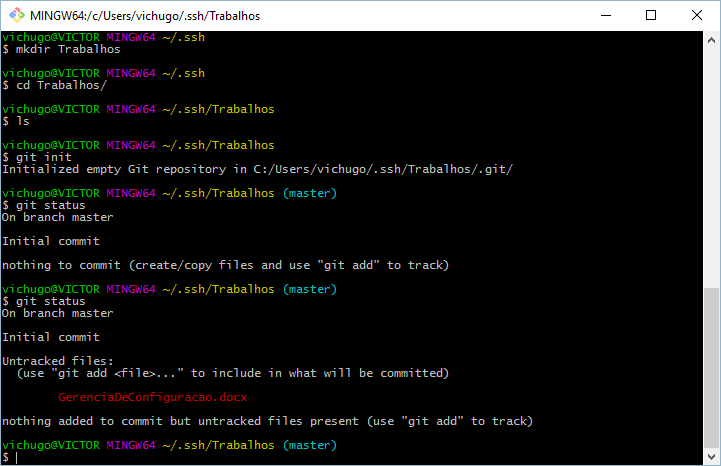
Verificar a conexão com o GitHub.



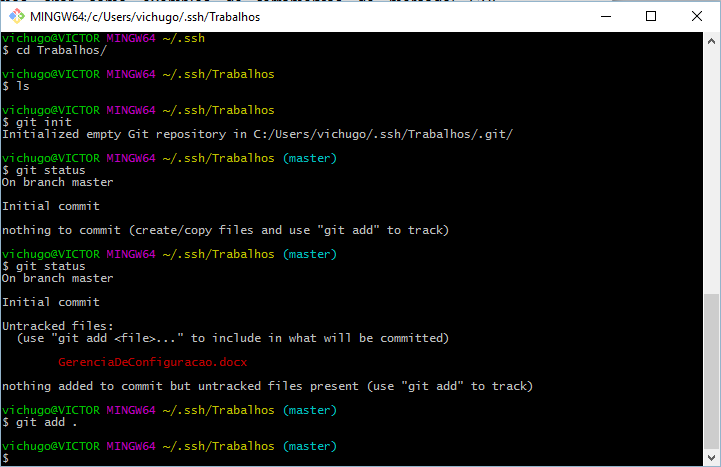
Criar uma pasta que será o seu repositório. Em seguida, utilizar o comando git init para inicializar o repositório.



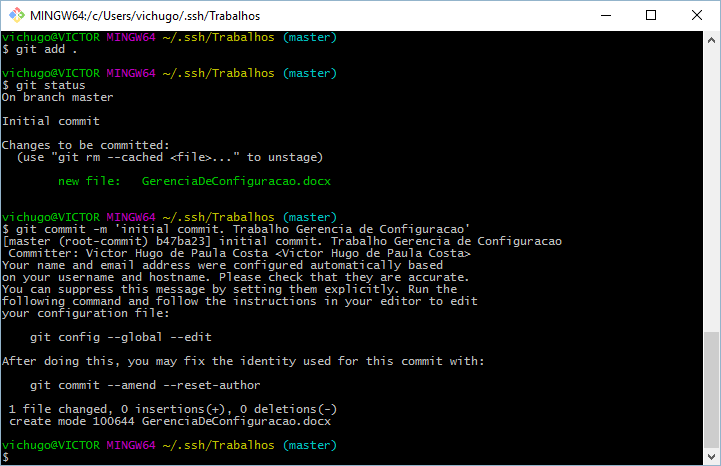
Adicionar no repositório o que você deseja disponibilizar. Utilizar o comando git status para verificar o status do repositório, se à algo para commitar.



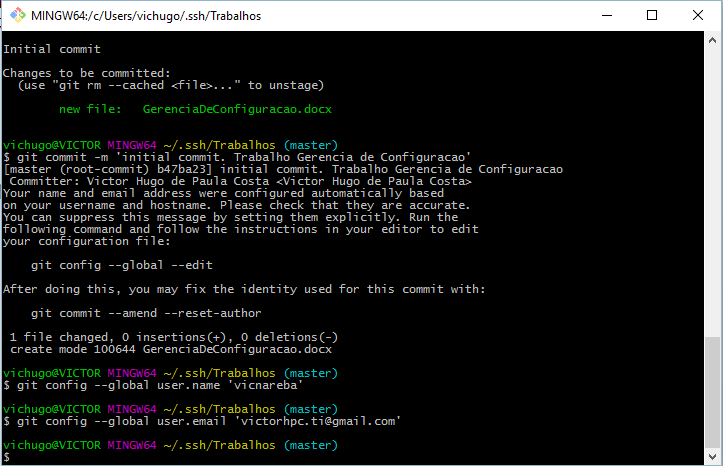
Feito isso, utilizar o comando git add . para adicionar o que será disponibilizado no commit.



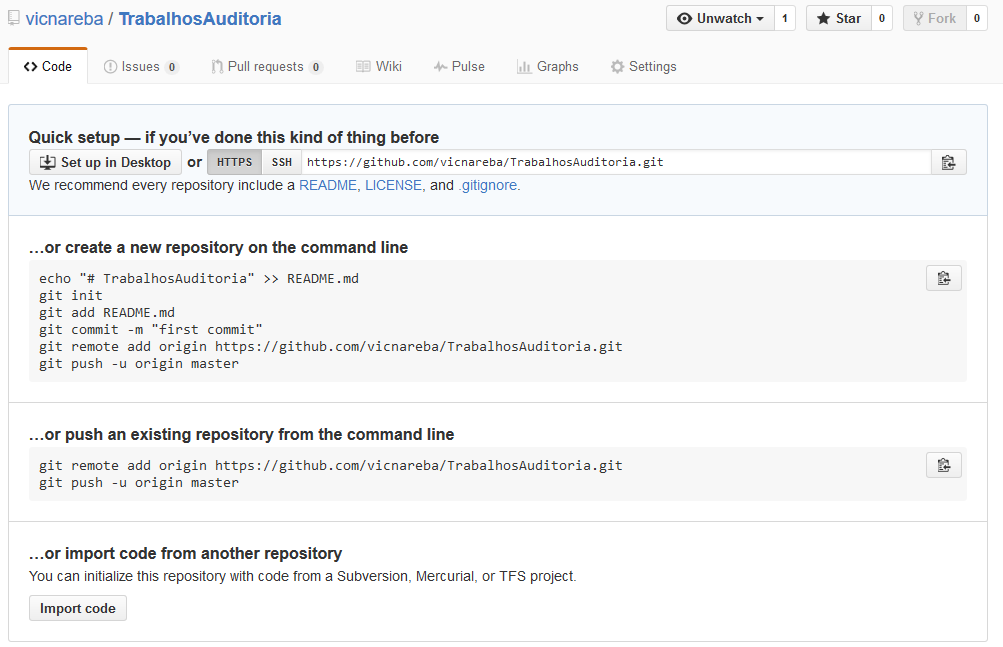
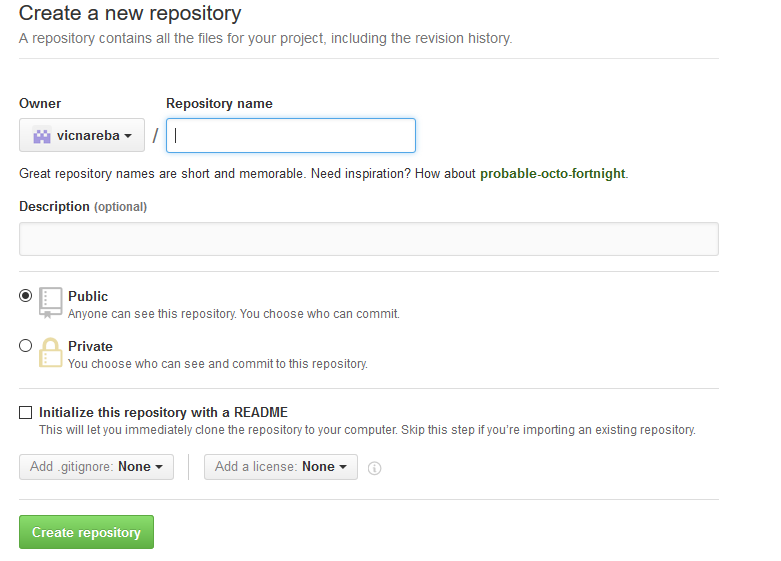
Utilizar em seguida o comando git commit –m ‘COMENTARIO’ para commitar o que deseja.



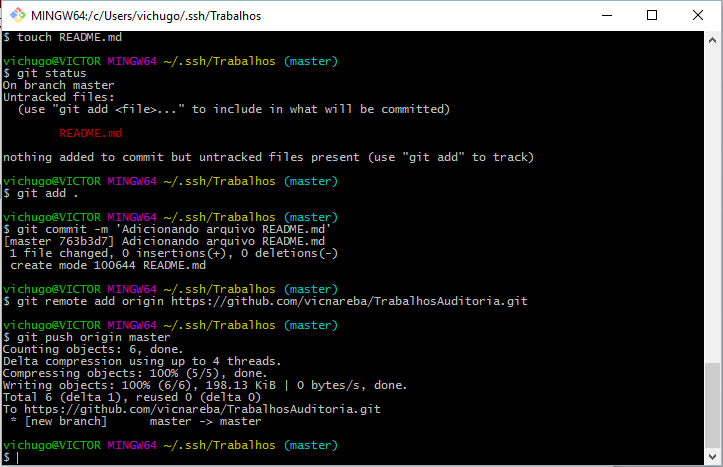
Para identificar o usuário do commit, usar os comandos git config –global user.name ‘usuário’ e git config –global user.email ‘e-mail do usuário’.



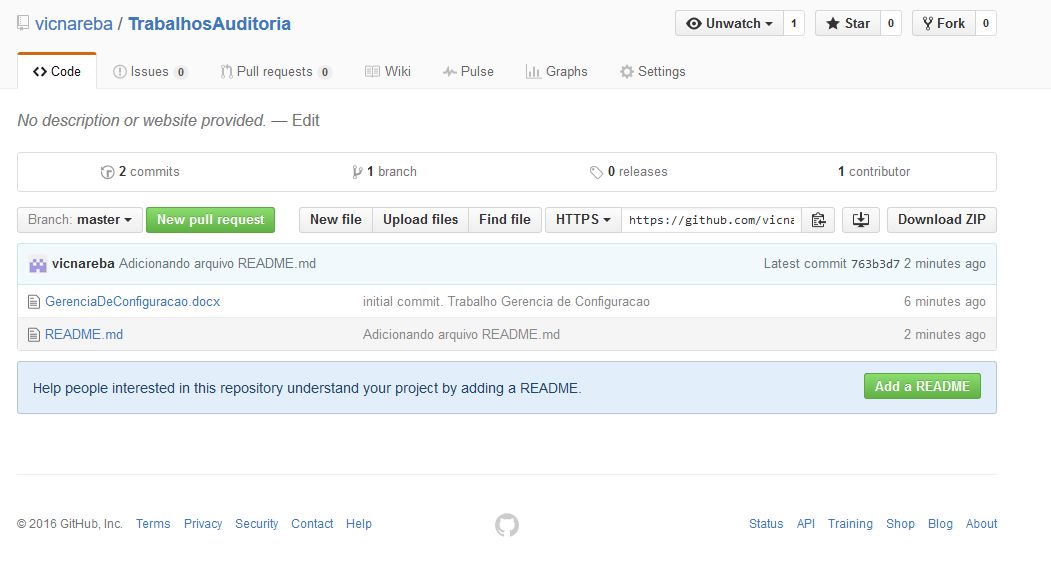
Agora vamos criar um repositório no GitHub.



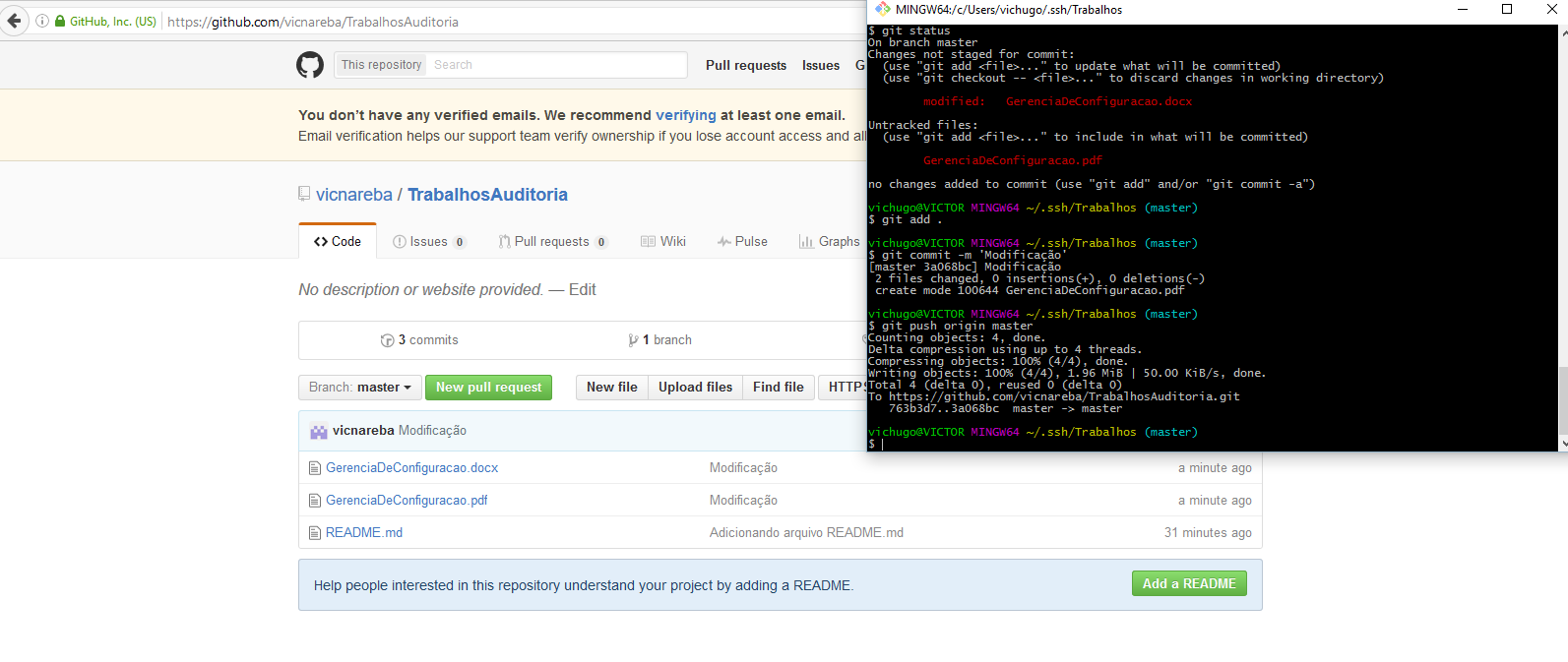
Feito isso, vamos sincronizar o repositório do GitHub com o seu repositório criado na máquina.



E pronto, o mesmo estará tanto na sua maquina, quanto na nuvem.



Qualquer modificação do arquivo, o mesmo identificará uma modificação e exigirá um commit. Utilizar os comandos ensinados para commitar e sincronizar com o repositório na nuvem.



# Bibliográfica

<http://www.devmedia.com.br/gerencia-de-configuracao-de-software/9145>

<https://www.youtube.com/watch?v=_Ci1q53DgvY>

<https://git-for-windows.github.io/>

<http://pt.slideshare.net/vaniltonpinheiro/mps-br-garantia-da-qualidade>