

# PLATAFORMA ARCHIVE FIXITY ANCHOR (AFA)

Este manuscrito documenta informações pertinentes ao conhecimento das informações sobre a implementação da Plataforma, seu funcionamento e quais as instruções para execução de uso dos recursos ofertados.

Nesse momento a plataforma AFA permite registrar manualmente os pacotes de informação (Archive Information Package – AIP) de um acervo com seus identificadores associados, assim como os eventos relacionados às mudanças nesses pacotes e viabiliza a auditoria da fixidez desses pacotes a partir da consulta de valores confiáveis de referência ancorados em *blockchain*.

## Implementação

Para a implementação foram desenvolvidos dois *scripts* na linguagem de programação Python (*backend*) e um terceiro *script* que se divide em trechos da mesma linguagem e se relaciona com outros escritos em HTML (*frontend*) para proporcionar uma interface *web* que permite a interação do administrador do arquivo com a plataforma.

Os dois primeiros *scripts* (*afa\_dlt.py* e *afa\_log.py*) implementam as funcionalidades dos módulos AFA-DLT e AFA-LOG, enquanto o terceiro *script* (*afa\_index.py*) surge para viabilizar uma interação com o AFA-LOG de forma amigável ao administrador do arquivo. Os *scripts* desenvolvidos somam pouco mais de 600 linhas de códigos com comentários e estão disponíveis no [Github](#).

Pela existência de experiência prévia a linguagem de programação Python foi a escolhida para a implementação, uma vez que ainda há nessa linguagem relevante acessibilidade e popularidade, dispondo de bibliotecas (nativas ou não) que auxiliaram a implementação de funções essenciais, a exemplo do gerenciamento de Árvores de Merkle e transações em *blockchain*.

Para essa versão implementada ainda não há redundância e nem a possibilidade da escolha dos algoritmos de *hash* e das redes *blockchains*, devendo a rede escolhida para utilização ser definida estaticamente a partir de ajustes para tal no AFA-DLT. Nesse

contexto, foi definida a versão de testes da rede Bitcoin (testnet), escolhida pela sua credibilidade, ausência de custos<sup>1</sup> e simplicidade de uso.

O armazenamento de dados na rede Bitcoin se dá pelo uso de uma instrução especial nas transações, chamada de `OP_RETURN`<sup>2</sup>, considerado o modo mais eficiente e apropriado para armazenamento de pequenas quantidades de dados nessa rede, onde é possível o armazenamento de até 80 bytes de dados.

A execução das transações varia de acordo com a implementação da solução, mas essencialmente é necessário um endereço de carteira válido na rede e que haja fundos disponíveis nessa carteira para pagamento da taxa de processamento da transação<sup>3</sup>.

O limite de 80 bytes para armazenamento de dados nessa rede se mostra mais do que o suficiente para armazenar um valor *hash* SHA-256 (32 bytes.), definido como o único algoritmo *hash* disponível nas operações da plataforma nesse momento e que gera uma cadeia de 64 caracteres hexadecimais. A popularidade desse algoritmo reflete seu custo-benefício, que alia robustez e eficiência, sendo um algoritmo amplamente difundido e bem suportado e, por isso, escolhido como algoritmo *hash* primário nessa solução.

Já para o armazenamento local dos dados de controle da plataforma, o JSON foi o formato escolhido para a estruturação desses dados, escolha também guiada por sua reconhecida acessibilidade e popularidade.

Dentre esses dados de controle estão os documentos de páginas (*p.json*, onde *p* se refere ao número da página), organizados de acordo com o padrão sugerido na arquitetura da plataforma e um documento de estado do sistema (*status.json*) que armazena algumas informações úteis à eficiência de execução da plataforma, como qual é a página aberta para novos registros e quais páginas apresentaram inconsistência nos dados com base no último processo de auditoria<sup>4</sup>.

---

1 Apesar do custo de transação ser inerente ao protocolo, a versão testnet da Bitcoin funciona de forma semelhante a sua vertente oficial, mas sem valor real. É destinada a interessados em testar o protocolo sem custos reais e para isso existem portais (*faucets*) que disponibilizam recursos gratuitamente para uso exclusivo nessa rede.

2 Uma das instruções especiais disponíveis na linguagem de *script* da Bitcoin.

3 [https://en.bitcoin.it/wiki/Miner\\_fees](https://en.bitcoin.it/wiki/Miner_fees)

4 Nesse primeiro momento esses documentos estão configurados para serem armazenados em um único diretório (*afa\_pages*), no entanto, se espera que em futuras versões, já para ambientes de produção, os documentos de páginas sejam distribuídas em vários subdiretórios sob critérios ainda não definidos.

Já considerando o uso da rede *blockchain* de testes da Bitcoin (Bitcoin testnet), ainda são listados a seguir os elementos de *software* cujo uso apoiou a condução da implementação dessa solução ao estágio atual de desenvolvimento e que, portanto, são considerados requisitos para a execução e funcionamento da plataforma:

- Interpretador Python ( $\geq 3.6$ ) e bibliotecas:
  - `cgi` – Permite a interação do administrador do arquivo com a plataforma através de métodos HTTP.
  - `datetime` – Permite manipular formatos de data e hora.
  - `hashlib` – Viabiliza as operações com *hash*.
  - `json` – Possibilita conversão de e para documentos no formato de dados JSON.
  - `time` – Gera intervalos de tempo ao consultar informações sobre transações (e evita que as requisições sejam identificadas como maliciosas — *spam*).
  - `urllib` – Gerencia requisições de URL por HTTP.
  - `bitcoinlib`<sup>5</sup> ( $\geq 0.5$ ) – Gerencia carteiras e transações na rede Bitcoin.
  - `merkletools`<sup>6</sup> – Viabiliza a geração de Árvores de Merkle, assim como a verificação a partir das provas de verificação.

De acordo com esses requisitos a solução pode ser considerada multiplataforma (a depender da disponibilidade do interpretador Python) e suas poucas diretrizes a serem ajustadas devem ser definidas manualmente com a mudança de valores iniciais de variáveis nos *scripts* referentes aos módulos da plataforma, a saber: o nome da carteira e rede Bitcoin a ser usada no módulo AFA-DLT e diretório raiz para armazenamento das páginas no AFA-LOG.

## Configuração e execução

Para a execução da solução o administrador do servidor do arquivo deve posicionar os *scripts* da plataforma (conteúdo do *afa\_sources*) no devido espaço para a execução de *scripts* Python e deve renomear o *script* da interface de gerenciamento

---

5 <http://github.com/1200wd/bitcoinlib>

6 <https://github.com/Tierion/pymerkletools>

(*afa\_index.py*) ou adaptar as configurações de seu servidor de acordo com suas preferências de acesso ao serviço.

Como já adiantado, o estágio atual da solução apenas prevê o registro e consulta na rede *blockchain* Bitcoin e, por padrão, o módulo de registro e consulta em DLT é previamente configurado para criar uma carteira Bitcoin na rede de testes, que ao ser invocado para seu primeiro registro, irá informar o endereço da carteira que deverá receber fundos para efetivar os registros pelo módulo.

O módulo de registro local e auditoria é configurado para gerenciar as páginas do Livro de Registro AFA em um diretório irmão por padrão (*afa\_pages*), mas que deve ser ajustado de acordo com as políticas da infraestrutura local e de parâmetros do servidor<sup>4</sup> (e permitir acesso de leitura e escrita para o *afa\_log.py*).

Da mesma forma, devem ser previstos os ajustes para o devido acesso do módulo aos pacotes que deverão ser cadastrados no sistema e monitorados por ele. Um diretório irmão (*afa\_data*) acompanha os *scripts* para servir de exemplo, podendo o administrador do servidor utilizá-lo como um atalho para o diretório real dos pacotes (facilitando o processo de registro, ainda manual). Pois, ainda deve se considerar que nessa versão, apenas o armazenamento local é suportado, ou seja, a plataforma deve estar sendo executada sob o mesmo sistema que os pacotes de informação a serem preservados e esses devem estar acessíveis para leitura pelo AFA-LOG.

A Figura 1 representa a interface inicial da AFA a qual o usuário da plataforma<sup>7</sup> deve se deparar em seu primeiro acesso. Nessa interface está contido um menu de opções e uma área de informações sobre o estado do livro de registros. Ambos os itens estão presentes em todas as interfaces da plataforma e são os únicos a aparecer no caso de uma execução inicial (sem ações definidas) ou quando há alguma ação definida com parâmetros ou resultados inválidos (nesse caso, uma mensagem adicional informativa sobre o problema é apresentada).

---

<sup>7</sup> Ressalta-se que a partir desse momento o administrador do arquivo será referenciado apenas como “usuário”, uma vez que essa passa a ser sua posição de relação com a plataforma.

Figura 1 — Interface inicial da plataforma Archive Fixity Anchor

# Archive Fixity Anchor

## Operation: Audit full collection

Menu

1. [Execute full audit \(audit entries and pages\)](#)

2. 

Edit or audit entries on page: 0

3. [Create new entries](#)

Book status

Total closed pages: 0

Not registered closed pages: 0 =>

Registered pages awaiting confirmation: 0 =>

Inconsistent pages: 0 => []

Inconsistent entries (Page ID: [Entries ID]): {}

Fonte: Próprio autor.

A área “Menu” apresenta as três únicas ações disponíveis na plataforma:

1. Executar uma auditoria completa do acervo registrado (todas as páginas do livro).
2. Editar registros (apontar registros substitutos) ou auditar páginas individuais (auditoria parcial).
3. Criar novos registros (registrar novos pacotes ou eventos relacionados).

Na área de informações do livro (book status)<sup>8</sup> é apresentada uma estatística sobre os registros na plataforma:

- **Total closed pages** – Contabiliza todas as páginas fechadas na plataforma.

<sup>8</sup> Para não necessitar calcular e gerar as informações sempre que necessário, as informações do livro são sempre salvas no documento *status.json*, no diretório *afa\_pages*, que serve para controle e é criado por padrão na primeira execução da plataforma.

- **Not registered closed pages** – Contabiliza e aponta todas as páginas fechadas na plataforma que ainda não tiveram seu *hash* registrado em DLT.
- **Registered pages awaiting confirmation** – Contabiliza e aponta todas as páginas fechadas na plataforma que ainda não tiveram o registro de seu *hash* em DLT confirmado (aguardando reforço do encadeamento do bloco onde se encontra a transação).
- **Inconsistent pages** – Contabiliza e aponta as páginas auditadas que apresentaram inconsistência em sua fixidez.
- **Inconsistent entries** – Aponta os registros auditados (e suas respectivas páginas) que apresentaram inconsistência em sua fixidez.

Pode ser percebido pelas informações do livro listadas que nessa implementação o fechamento da página não implica automaticamente no registro dessa em DLT. O fechamento da página nessa versão da implementação é dividida em três fases:

1. Fechamento da página para novos registros – A página não permite novos registros e ganha seu *hash* raiz da Árvore de Merkle, criada a partir dos registros na página.
2. Registro em DLT da página fechada – A página fechada fica aguardando o usuário solicitar o registro em DLT e continuará disponível para esse registro em caso de algum erro na requisição anterior de registro.
3. Confirmação do registro em DLT – A transação referente ao registro da página ainda não obteve a quantidade de confirmações necessárias para ser considerada imutável. A plataforma se encarrega de atualizar o estado de confirmação dessas transações, mas de toda forma, as páginas listadas nessa fase também ficam passíveis de sofrerem um novo pedido de registro pelo usuário, uma vez que a transação relacionada pode ter sofrido algum impedimento a sua aceitação, difusão ou confirmação.

Todas as ações referentes ao fechamento e registro das páginas são facilmente acessíveis a partir de links que ficam disponíveis ao usuário na área de informações

sobre o estado do livro e na área de criação de registros (para o fechamento inicial da página), como será apresentado na próxima subseção.

### Criando novos registros

Ao optar por criar novos registros, o usuário é encaminhado a uma interface dedicada a essa ação, onde poucos dados são necessários para o registro, demandando apenas o Identificador Global referente ao pacote (gerado pelo *software* de repositório) e um texto, que descreve um evento relacionado ao pacote ou descreve o caminho para acesso ao pacote no sistema.

A Figura 2 apresenta áreas adicionais que a interface para a criação de novos registros da plataforma oferece (além da área de menu e de informações sobre o estado do livro, apresentados anteriormente e omitidos na Figura). Nessas áreas estão contidas as informações sobre a página atual e os registros já realizados e os campos para preenchimento necessários a um novo registro.

Figura 2 — Interface para a criação de novos registros

Page information

ID: 3

Anchors: Not available yet

Timestamp: Not available yet

Consistent? Yes

Create entry

ID 4

Global ID

Description (text or file path)

Description is a file path? ☐ Yes

Register entry

Clear form

OR [Close page](#)

Entries information

Local ID	Global ID	Description (Text or File path)	Hash
3	94d6b806-6f56-4e4c-8190-4b5064a77007	path:afa_data/pb_2009-2013_1.zip	9a0a6543a8e2f4bc4363edfb00ade539b445c3a470f14e874296aea5f910ea4b
2	73ce8e23-5fcd-47aa-a423-ae2d9a8a7591	path:afa_data/pb_2009-2013_2.zip	2f37311e657fe26863f7f98e51d4272175940ee2dae8df87de858b7b61d9d815
1	e253ecf2-59ad-4a89-b970-043a34774e50	This is a text about an event...	8822923d5cd75d46c4269d1b1f4bcf098724e02e37b340ba6e6b536d7a9ba8ec
0	73ce8e23-5fcd-47aa-a423-ae2d9a8a7591	path:afa_data/pb_2009-2013_1.zip	0a52867c6f83bdacb540e351f45fd53124dc2bac5eb34a232829320987cb09c3

Fonte: Próprio autor.

Uma vez que a página ainda se encontra em construção, além de seu identificador (ID), a mesma ainda não dispõe de dados relevantes para a exibição, o que só ocorrerá após seu fechamento e confirmação de registro em DLT<sup>9</sup>.

Na área para criação de novos registros (create entry) estão dispostos os seguintes campos:

- **Local ID** – Um número inteiro (iniciando em zero) sequencial e único na página que identifica o registro na plataforma junto ao número da página. É criado automaticamente e não é permitido sua alteração pelo usuário.
- **Global ID** – Um campo de texto para a inserção de uma identificação única atribuída pelo sistema de repositório ao pacote em questão. Identifica o pacote no sistema de repositório e é uma das informações necessárias a formação do *hash* referente ao pacote em questão.
- **Description (text or file path)** – Um campo de texto para inserir uma referência ao pacote, pela descrição do caminho no sistema para obtê-lo (*file path*) ou um texto que justifique um evento referente a um registro anterior correspondente ao mesmo Identificador Global em questão. Quando o texto for utilizado, o *hash* desse texto será usado para a criação do *hash* correspondente ao respectivo registro, do contrário, o *hash* do próprio pacote será utilizado para isso.
- **Description is a file path?** – Se marcado (Yes), instrui a plataforma a utilizar o valor inserido no campo anterior como um caminho no sistema para acessar o pacote e considerar esse parâmetro para tratar esse registro adequadamente (gerando *hash* para o pacote ou para o texto).
- **Register entry / Clear form** – Botões de ação para efetivar o registro utilizando os dados informados ou para limpar os campos de informações.
- **Close page** – *Link* que conduz a plataforma ao fechamento da página em questão de acordo com os procedimentos já apresentados, levando a página para estágios de fechamento apresentados ainda nesta seção.

Na área de informações dos registros (entries information) são listados em ordem decrescente os registros já realizados na página atual, trazendo além do Local ID,

---

<sup>9</sup> Sempre que não houver uma página aberta (no caso de uma primeira execução), ao optar por criar novos registros um documento *p.json* é criado no diretório *afa\_pages*, onde *p* é o número identificador da página (que nesse caso é 0).



Global ID e Description, o seu valor *hash*, gerado a partir do *hash* da concatenação do valor *hash* do Identificador Global (Global ID) e do valor *hash* da Descrição do registro (Description).

Ao selecionar a opção por fechar a página (*link* “Close page”), o usuário é redirecionado a mesma interface de criar novos registros, mas agora será exibida uma mensagem relatando o sucesso nesse primeiro estágio de fechamento da página e o usuário poderá perceber que tal página já se mostra contabilizada no campo “not registered closed pages”.

O próximo passo para dar prosseguimento ao processo de registro de uma página é selecionar seu ID (como um *link*), exposto no campo “not registered closed pages”, que requisitará o registro dela em uma *blockchain*. Após isso, caso não haja fundos suficientes, a plataforma alertará o usuário sobre essa situação e informará a rede e um endereço da carteira nessa rede no qual deve ser depositado um valor para suprir o custo da transação<sup>10</sup>. Caso contrário, o usuário é notificado sobre a efetivação de registro da página e o ID dessa é migrado do campo “not registered closed pages” para “registered pages awaiting confirmation”, onde permanecerá até que a plataforma receba a sua confirmação pela rede<sup>11</sup> e notifique o usuário de que o registro da referida página foi confirmado, encerrando o ciclo de fechamento da página e a tornando passível de auditoria.

## Editando registros

A partir da opção no menu de editar ou auditar registros em uma página (edit or audit entries), o usuário obterá as informações e registros de uma página, que já deve estar fechada e registrada, o que lhe permitirá auditar os registros dessa página, assim como adicionar dados de registros substitutos àqueles registros de pacotes que tenham sofrido alguma alteração e que, por isso, precisam agora referenciar sua versão

---

10 A taxa para registrar transações na rede varia ao longo do tempo, mas nessa implementação abordada (utilizando a rede de testes) uma quantia mínima de 100µBTC foi definida estaticamente como montante mínimo necessário para efetivar as transações, visando oferecer uma certa margem para avaliar a ferramenta, pois nesse período de testes essa taxa girou em torno de 6,5µBTC.

11 A confirmação nessa rede se dá quando o bloco que alocou a transação é sucedido por mais de 6 blocos subsequentes sem que ocorra divergências de blocos.

atualizada ou a descrição de um evento relacionado a si. Essa interface para edição ou auditoria de registros em uma página está representada na Figura 3.

Figura 3 — Interface para editar ou auditar registros em uma página

Page information

ID: 3  
Anchors: Root Hash (dc5271ac6f788058fb43a1e82b1e5789530c36e0e5d5ae6ebfb012b7d9b46c008) anchored on [Bitcoin testnet](#)  
Timestamp: 2020-12-17T02:39:37Z  
Consistent? Yes ([Audit this page](#))

Local ID	Global ID	Description (Text or File path)	Hash	Replacement Entry (Page ID/Local ID)	Consistent
0	73ce8e23-5fcd-47aa-a423-ae2d9a8a7591	path:afa_data/pb_2009-2013_1.zip	0a52867c6f83bdacb540e351f45fd53124dc2bac5eb34a232829320987cb09c3	Entry replaced? <input type="checkbox"/> Yes None / None Update entry	Yes
1	e253ecf2-59ad-4a89-b970-043a34774e50	This is a text about an event...	8822923d5cd75d46c4269d1b1f4bcf098724e02e37b340ba6e6b536d7a9ba8ec	Entry replaced? <input type="checkbox"/> Yes None / None Update entry	Yes
2	73ce8e23-5fcd-47aa-a423-ae2d9a8a7591	path:afa_data/pb_2009-2013_2.zip	2f37311e657fe26863f7f98e51d4272175940ee2dae8df87de858b7b61d9d815	Entry replaced? <input type="checkbox"/> Yes None / None Update entry	Yes
3	94d6b806-6f56-4e4c-8190-4b5064a77007	path:afa_data/pb_2009-2013_1.zip	9a0a6543a8e2f4bc4363edfb00ade539b445c3a470f14e874296aea5f910ea4b	Entry replaced? <input type="checkbox"/> Yes None / None Update entry	Yes

Fonte: Próprio autor.

Diferente da criação de registros, nessa fase já é possível consultar informações sobre o registro da página e sua fixidez através dos seguintes campos na área de informações da página (Page information):

- **ID** – Um número inteiro (iniciando em zero) sequencial e único na plataforma para identificar a página. É utilizado para referenciar a página e registros substitutos dentro da plataforma.
- **Anchors** – Valor *hash* raiz da Árvore de Merkle gerada a partir dos registros da página (*hashes* dos registros) e um *link* para consultar esse valor ancorado na *blockchain* (através do nome da rede e o identificador da transação nela)<sup>12</sup>.
- **Timestamp** – Um carimbo de tempo que marca o momento em que a transação com o valor âncora foi aceita pela rede (a partir da mineração do bloco que a continha).

<sup>12</sup> Nessa implementação é utilizado o serviço de exploração de blocos da rede de testes da Bitcoin Blockstream (<https://blockstream.info/testnet/>).

- **Consistent?** – Marcação da fixidez da página a partir da realização da última auditoria. Antes de passar por uma primeira auditoria todas as páginas são consideradas consistentes por padrão. Nesse campo ainda é dada uma opção ao usuário para requisitar a auditoria da página (auditoria parcial) através do *link* “Audit this page”.

Na área de informação dos registros (Entries information), além dos campos já previamente apresentados na interface de criação de novos registros (Local ID, Global ID, Description e Hash), estão contidos os seguintes campos:

- **Replacement Entry (Page ID/Local ID)** – Permite a marcação de que o respectivo registro deve ser considerado substituído por um registro atualizado, o qual é referenciado por um determinado identificador (Local ID) em uma determinada página (Page ID). No caso de atualização do registro, os respectivos campos devem ser preenchidos e em seguida deve ser acionado o botão “Update entry” para gravar as informações na página.
- **Consistent** – Marcação da fixidez do registro a partir da realização da última auditoria. Antes de passar por uma primeira auditoria todos os registros são considerados consistentes por padrão.

## Auditando registros

Nessa implementação da plataforma é possível optar pelo processo de auditoria total (todas as páginas) ou parcial (páginas individuais). A requisição de uma auditoria total pode ser feita a qualquer momento a partir do *link* “Execute full audit (audit entries and pages)” disponível na área de menu que acompanha todas as interfaces da plataforma. Enquanto a requisição de uma auditoria parcial deve ser solicitada a partir da interface de edição da página alvo através do *link* “Audit this page”.

O processo de auditoria implementado só funcionará se não houver páginas com registro pendente e tem funcionamento similar em ambos os modos de auditoria, no entanto, como esperado, a auditoria total iniciará pela última página fechada e registrada e só encerrará ao processar a página de ID 0.

Seja qual for o tipo de auditoria, o seu resultado será exposto pela plataforma em mensagem ao usuário, que poderá consultar a qualquer tempo quais páginas e registros foram identificados como inconsistentes no último processo.

Dentre os resultados dos processos de auditoria na plataforma, há um comportamento que merece destaque devido sua excepcionalidade, que ocorre quando ao menos um registro em uma página se mostra inconsistente, mas o *hash* original da página confere com o valor âncora, demonstrando que houve um distúrbio nos elementos relacionado àquele(s) registro(s) (corrupção de pacote ou das informações de fixidez da plataforma), nesse caso a plataforma terá o seguinte comportamento:

- A página não é marcada como inconsistente e assim não será marcada como tal no campo “Inconsistent pages” na área de informação do livro (Book status).
- O(s) registro(s) é(são) marcado(s) como inconsistente(s), sendo apontado(s) como tal no campo “Inconsistent entries” junto com a página da qual faz(em) parte.
- Na interface “edit or audit entries” ambos, registro(s) e página são mostrados como inconsistentes.

A marcação de inconsistência nos registros e páginas persistem por padrão até que o problema interferente seja sanado e um novo processo de auditoria seja executado contemplando a respectiva página ou registro.

## **Considerações finais**

Nessa versão documentada a plataforma ainda apresenta uma interface rudimentar, mas suficientemente usual, pois poucas são as ações disponíveis ao usuário e os rótulos são autoexplicativos, mas de toda forma, este capítulo visou documentar todas as opções disponíveis, o fluxo de execução das ações na plataforma e o seu funcionamento.

No estágio atual de implementação a plataforma já entrega os recursos necessários ao registro e auditoria de um acervo de pacotes de informação que atenda as premissas da solução previamente apresentadas, permitindo uma prova de conceito e avaliações experimentais iniciais, mas ainda carece de opções que incrementam

robustez a proposta, como as opções de redundância de redes *blockchains* e de algoritmos de *hash*. Dessa forma, apesar de possível, ainda não é recomendável o uso da implementação apresentada neste capítulo para ambientes de produção<sup>13</sup>.

Além dos recursos de redundância, se deseja que outras melhorias sejam introduzidas na plataforma, adicionando recursos e funcionalidades não implementados neste primeiro momento, como a automatização de registro dos pacotes, a busca de pacotes fora do sistema de execução da plataforma, o tratamento de pacotes de informação não organizados e contidos em recipientes e a avaliação e possíveis adequações para uso da plataforma com pacotes do tipo DIP.

---

<sup>13</sup> Recomenda-se ao leitor que consulte as notas de versão referente ao momento em que estiver consultando este trabalho para conhecer o estágio atualizado da implementação.