

Manual de instalação

Manual de instalação

Como deixar o CEBI Tester rodando numa máquina nova. Os comandos são copiáveis; execute um de cada vez, na ordem.

Para usar a ferramenta depois de instalada, veja o [Manual do usuário](#).

1. O que a máquina precisa ter

Item	Versão	Para quê
Windows	10 ou 11	É onde a ferramenta roda hoje
Node.js	20 ou superior (testado no 24.15)	Executa a ferramenta
Git	qualquer recente	Baixa e atualiza o código
Espaço em disco	~2 GB	Dependências, navegador de teste e evidências

Além disso, a máquina precisa **alcançar pela rede**:

- os endereços dos ambientes dos clientes (o Balcão, a Baixa, a Hidrometria e o módulo Usuários);
- os bancos Oracle desses clientes, se for usar a conferência no banco (veja o passo 6).

Para conferir o que já está instalado:

```
node -v
```

```
git --version
```

Se o Node não estiver instalado, baixe a versão LTS em <https://nodejs.org> e repita o comando.

2. Baixar o CEBI Tester

Escolha uma pasta de trabalho e clone o repositório:

```
git clone https://bitbucket.org/cebi/cebi.qualityassurance.git
```

Entre na pasta criada:

```
cd cebi.qualityassurance
```

Todos os comandos seguintes são executados **dentro dessa pasta**.

3. Instalar as dependências

```
npm install
```

Demora alguns minutos na primeira vez. Ao final, aparece a quantidade de pacotes instalados.

4. Instalar o navegador de teste

A automação usa um navegador próprio, separado do Chrome do dia a dia:

```
npx playwright install chromium
```

5. Criar o arquivo de credenciais

As senhas ficam num arquivo chamado `.env`, que **nunca** é enviado ao repositório. Comece pelo modelo que acompanha o projeto:

```
cp .env.example .env
```

Abra o `.env` num editor de texto e preencha os blocos dos clientes que você vai usar. Cada cliente tem três linhas principais:

```
POCOS_BASE_URL=http://devlocal.cebi.com.br:40054
POCOS_USUARIO=
POCOS_SENHA=
```

Duas observações:

- **Não precisa preencher tudo.** Quem usa a interface web digita usuário e senha a cada execução; o `.env` serve para rodar pela linha de comando sem digitar toda vez.
- O `.env` **é só desta máquina.** Ao atualizar a ferramenta, ele não é tocado.

6. (Opcional) Ligar a conferência no banco

Parte dos casos confere na base o que a tela mostrou — é o que diferencia "a tela disse que gravou" de "gravou mesmo". Para isso a ferramenta usa o **Dicionário de Dados**, que é um produto à parte, com repositório próprio.

Sem ele, a ferramenta roda e aprova a maioria dos casos; os que dependem de conferência no banco ficam **bloqueados**, com o motivo escrito no resultado.

Para ligar:

1. Instale o Dicionário de Dados na mesma máquina.
2. Cadastre nele a conexão de cada banco que será usado — **o cadastro é feito por quem tem a senha**, no aplicativo do Dicionário, e fica protegido pelo usuário do Windows que o cadastrou. Por isso o cadastro precisa ser refeito em cada máquina nova.
3. No arquivo de cada ambiente, em `config/ambientes/<cliente>-master.json`, confirme que o nome em `basePreparation.opcoes.banco` é exatamente o nome cadastrado no Dicionário.

7. Subir a interface

```
npm run servidor -- --http --porta 4480 --host 127.0.0.1
```

A tela mostra o endereço, os ambientes carregados e dois avisos — que não há autenticação e que, em HTTP, a senha trafega em texto claro. Abra no navegador:

```
start http://localhost:4480
```

Para que outras pessoas da rede interna acessem, troque o host:

```
npm run servidor -- --http --porta 4480 --host 0.0.0.0
```

“ **Mantenha dentro da rede interna.** A interface não tem login e dispara escrita em base de cliente — foi uma decisão de projeto, não um esquecimento.

Enquanto esse comando estiver aberto, o servidor está no ar. Fechá-lo derruba a interface.

8. Conferir se ficou tudo certo

Liste os ambientes que a instalação enxerga:

```
npm run clientes
```

E peça a versão publicada de um deles, sem executar nada:

```
npm run validar-versao -- --ambiente guaratingueta-qa --sistema balcao --somente-versao
```

Se vier a versão do ambiente, a instalação está completa: a ferramenta alcançou o cliente, autenticou e leu a versão.

9. Acrescentar um ambiente novo

Cada cliente é um arquivo em `config/ambientes/`. O caminho mais seguro é copiar um que já funciona e trocar o endereço, o nome e o banco:

```
cp config/ambientes/guaratingueta-qa.json config/ambientes/novocliente-qa.json
```

Depois edite, no arquivo novo: `slug`, `nome`, `cliente`, `baseUrl` e, se for usar conferência no banco, `basePreparation.opcoes.banco`. Reinicie o servidor para que ele apareça na lista.

10. Problemas comuns

Sintoma	Causa provável	Solução
<code>node</code> ou <code>git</code> não é reconhecido	Não instalado, ou terminal aberto antes da instalação	Instale e abra um terminal novo
<code>npm install</code> falha ao baixar	Rede ou proxy da empresa	Falar com a infraestrutura
A interface não abre no navegador	O servidor não está no ar	O comando do passo 7 precisa ficar aberto
"porta já em uso"	Já existe um servidor rodando	Use outra porta (<code>--porta 4481</code>) ou feche o anterior
Nenhum ambiente na lista	Nenhum arquivo em <code>config/ambientes/</code>	Veja o passo 9
Tudo bloqueado por banco	Dicionário de Dados não instalado ou sem a conexão cadastrada	Passo 6

Revisão #1
Criado 2026-10-20 14:12:47 UTC por Stephanie Correa
Atualizado: 2026-10-20 14:13:32 UTC por Stephanie Correa