

# ✓ PLANO DE TESTE – API de Monitoramento (MongoDB + Dashboard)

**Funções testadas:** Deploy via Docker Compose, Login e Criação de Documentos

**Data:** 24/11/2025

**Responsável:** *Gabriel dos Santos Rodrigues*

---

## 1. TESTE DO DEPLOY VIA DOCKER COMPOSE (Função 10)

### 1.1 Objetivo

Validar que o sistema pode ser executado dentro de containers Docker usando uma imagem hospedada no Docker Hub do usuário.

### 1.2 Pré-condições

- Imagem no DockerHub: **kamisama157/fiware-dashboard:latest**
  - Arquivo `docker-compose.yml` funcional.
  - Docker Desktop ativo.
- 

### 1.3 Casos de Teste

---

#### ***Caso 1 – Levantar ambiente via docker-compose***

**Passos:**

1. Criar arquivo `docker-compose.yml` contendo:

```
services:
  dashboard:
    image: kamisama157/fiware-dashboard:latest
    container_name: fiware_dashboard
    ports:
      - "3000:3000"
    restart: always
```
2. Executar:

```
docker compose up -d
```

**Resultado esperado:**

- Container sobe sem erros.

- Logs mostram: “Servidor rodando na porta 3000”.
  - Acessar no navegador: ➔ `http://localhost:3000/`
- 

## ***Caso 2 – Testar login dentro do container***

### **Passos:**

1. Acessar `localhost:3000`.
2. Repetir o teste de login (função 1).

### **Resultado esperado:**

- Autenticação funciona do mesmo jeito que em ambiente local.
- 

## ***Caso 3 – Testar criação de documentos via container***

### **Passos:**

1. Logar dentro do container.
2. Criar documento via dashboard.

### **Resultado esperado:**

- Documento é criado no MongoDB real.
  - A tabela mostra imediatamente.
- 

## ***Caso 4 – Reiniciar container***

`docker compose restart`

### **Resultado esperado:**

- Session store funciona.
  - Login pode ser exigido novamente.
  - Funções CRUD seguem ativas.
- 

## ***Caso 5 – Parar e remover o ambiente***

`docker compose down`

### **Resultado esperado:**

- Container é removido.
  - Nenhum dado no MongoDB é afetado (pois ele é externo).
-

## 2. TESTE DA FUNÇÃO DE LOGIN (API + Interface Web)

### 2.1 Objetivo

Validar se o usuário consegue autenticar-se corretamente no sistema informando credenciais MongoDB válidas, e se o backend estabelece conexão com o banco antes de liberar acesso ao dashboard.

### 2.2 Pré-condições

- Servidor Node.js em execução **localmente**.
- MongoDB ativo e acessível na rede.
- Credenciais válidas de acesso ao banco.
- Navegador funcionando.

### 2.3 Dados de Teste

Caso	Usuário	Senha	Resultado Esperado
Login válido	admin	senhaCorreta	Redirecionar para /dashboard
Login inválido	admin	errado	Exibir “Falha ao autenticar no MongoDB”
Campos vazios	—	—	Exibir erro e impedir envio

---

### 2.4 Casos de Teste

#### Caso 1 – Login bem-sucedido

##### Passos:

1. Abrir `http://localhost:3000/`.
2. Digitar usuário e senha corretos.
3. Clicar em “Entrar”.

##### Resultado esperado:

- A sessão é criada e armazenada.
  - O servidor conecta ao MongoDB.
  - O usuário é redirecionado para o dashboard.
- 

#### Caso 2 – Credenciais inválidas

##### Passos:

1. Inserir usuário correto e senha incorreta.
2. Tentar autenticar.

**Resultado esperado:**

- Página de login recarrega.
  - Mensagem: “Falha ao autenticar no MongoDB.”
  - A sessão **não é criada**.
- 

**Caso 3 – Requisição ao dashboard sem sessão****Passos:**

1. Abrir diretamente `http://localhost:3000/dashboard` sem login.

**Resultado esperado:**

- O middleware `requireAuth` bloqueia.
  - Redireciona para a página de login.
- 

## 3. TESTE DA FUNÇÃO DE CRIAÇÃO DE DOCUMENTOS (API)

### 3.1 Objetivo

Garantir que o sistema registre corretamente novos documentos no MongoDB com os campos:

- `recvTime` (gerado automaticamente)
- `attrName`
- `attrType`
- `attrValue`

### 3.2 Pré-condições

- Usuário deve estar autenticado.
  - Dashboard acessível.
  - Conexão com MongoDB ativa.
- 

### 3.3 Dados de Teste

Campo	Valor de Teste	Validade
<code>attrName</code>	“luminosity”	válido
<code>attrType</code>	“Integer”	válido
<code>attrValue</code>	42	válido
<code>attrValue</code>	“abc”	inválido

---

## 3.4 Casos de Teste

### Caso 1 – Criação válida

**Passos:**

1. No dashboard, localizar o formulário de criação.
2. Preencher:
  - Nome: luminosity
  - Tipo: Integer
  - Valor: 42
3. Enviar.

**Resultado esperado:**

- O servidor cria o documento com sucesso.
  - recvTime é gerado automático.
  - Aparece na tabela imediatamente.
- 

### Caso 2 – Valor inválido

**Passos:**

1. Preencher attrValue = “abc”.

**Resultado esperado:**

- Mongoose rejeita ou converte incorretamente.
- O sistema deve exibir erro ou não criar registro.

*(Caso deseje, posso fazer um teste negativo mais específico.)*

---

### Caso 3 – Campos vazios

**Passos:**

1. Enviar formulário sem preencher nada.

**Resultado esperado:**

- Formulário impede o envio.
- Nada é criado.