Test Mass Handler

Case: Como resolvemos o nosso problema com massa de dados para testes?

/thoughtworks

natura

O mundo é mais bonito com você

Quem somos

Analistas de Qualidade na Thoughtworks Brasil

Ana Ludmila

Nathalia Torres

/thoughtworks

Uma comunidade vibrante e distribuída de tecnologistas

9000+
pessoas

27+
anos

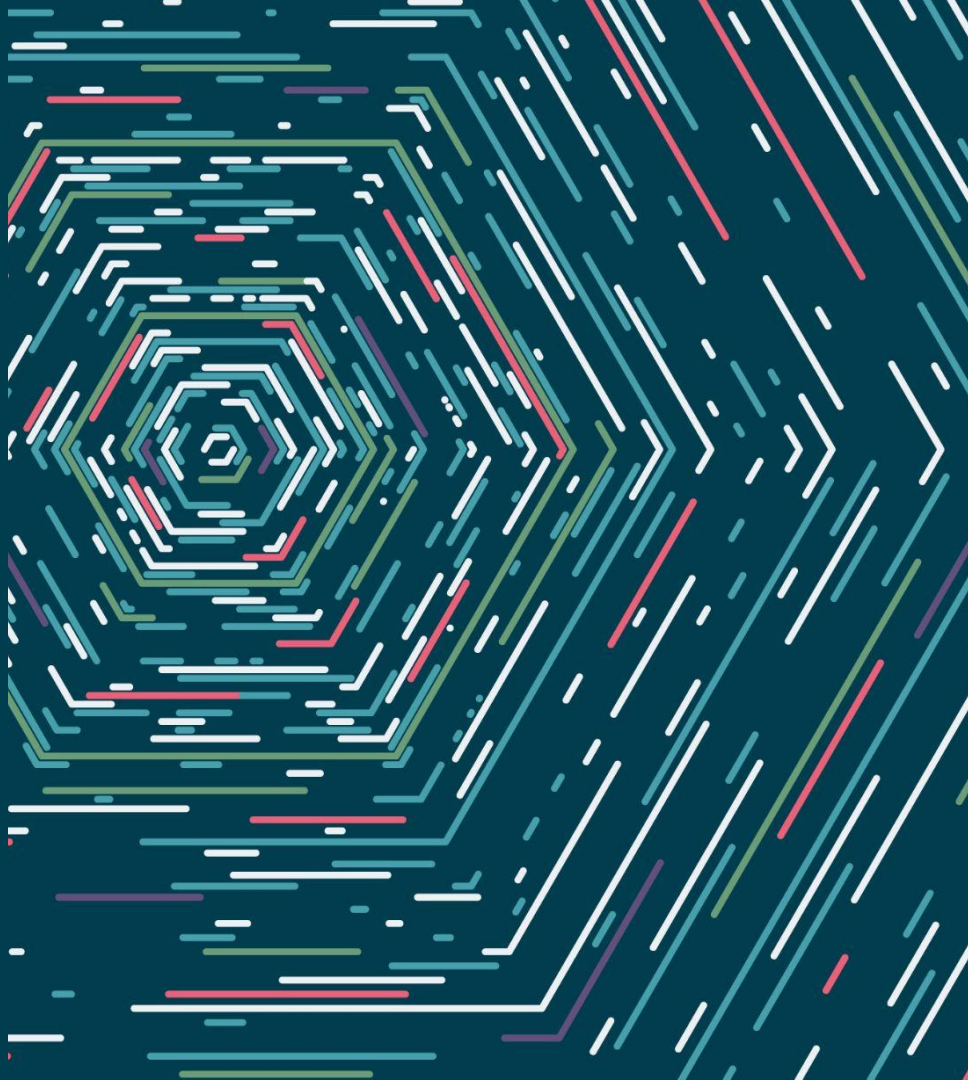
48
escritórios

17
países

Alemanha / Austrália / Brasil
Canadá / Chile / China / Equador
Espanha / Estados Unidos / Finlândia
Holanda / Índia / Itália / Reino Unido
Romênia / Singapura / Tailândia

Agenda

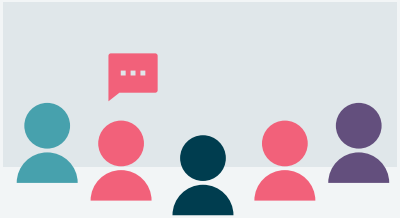
Nosso Contexto



/thoughtworks

QAs atuando dentro de um System Team

O que é um System Team?

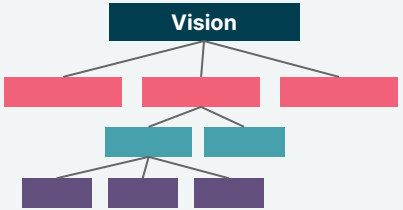


Equipe especializada em construir e dar suporte ao ambiente de desenvolvimento Agile, apoiando a criação e manutenção das Pipelines de Entrega Contínua.

O que ele faz?

Constrói infraestrutura de sistema, integração de solução, testes de ponta a ponta (E2E) e release da aplicação.

Quais são as responsabilidades das pessoas QAs?

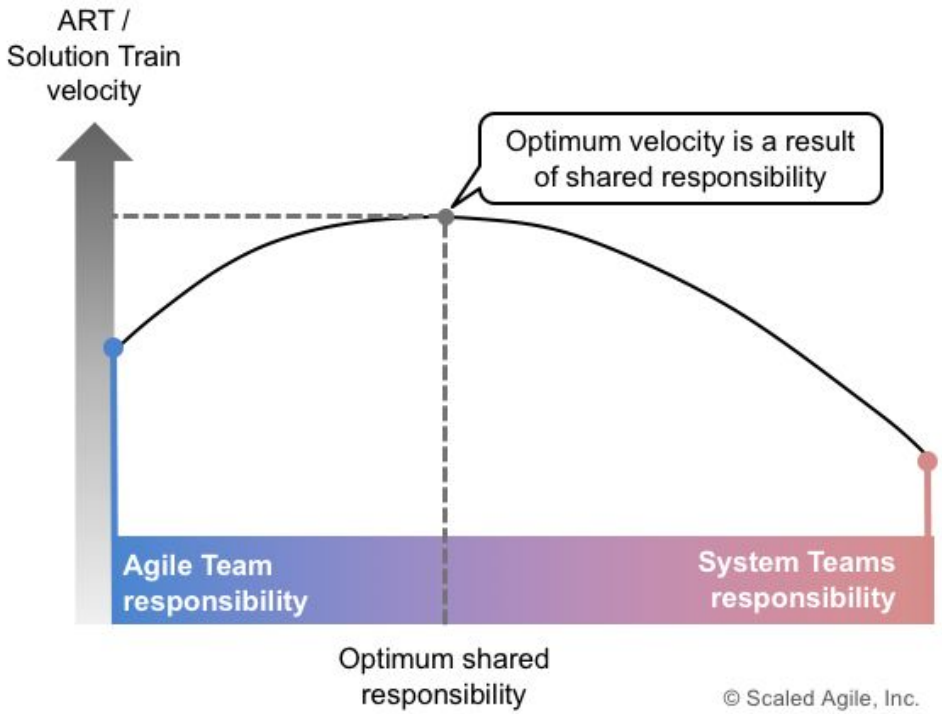


Garantir que os fluxos principais dos usuários estão funcionando corretamente através de Testes E2E.

Releases através de um Agile Release Train

ART alinham as equipes a uma missão compartilhada de negócios e tecnologia que planeja, compromete, desenvolve e implanta em conjunto.

O desenvolvimento de soluções eficientes requer o compartilhamento das melhores práticas entre o System Team e os outros times.



Alguns pontos relevantes sobre nosso contexto

Autonomia limitada como vendors

Enquanto consultoria, é natural que não tenhamos privilégios de acesso irrestrito a todos os dados. Portanto, algumas de nossas ações foram orientadas para nos adaptar a estas restrições.

Foco em resolver um problema pontual, não um problema coletivo

Embora a solução esteja sendo reutilizada por alguns times pontualmente, a idéia inicial era resolver o nosso problema enquanto time, e não um problema geral a nível organizacional.

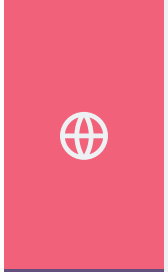
Nos livrar de tarefas repetitivas

Por não possuirmos um ambiente efêmero, a mesma base de dados de homologação é utilizada durante alguns ciclos de teste, o que nos gerava alguns retrabalhos.

O desafio que nos trouxe aqui hoje...

Um contexto de time tão complexo nos traz vários tipos de desafios!

No entanto, vamos focar somente na questão pela qual estamos aqui: **Gerenciar massa de dados para testes pré release.**



Refresh no banco de dados com alta frequência

Massa de dados desatualizada

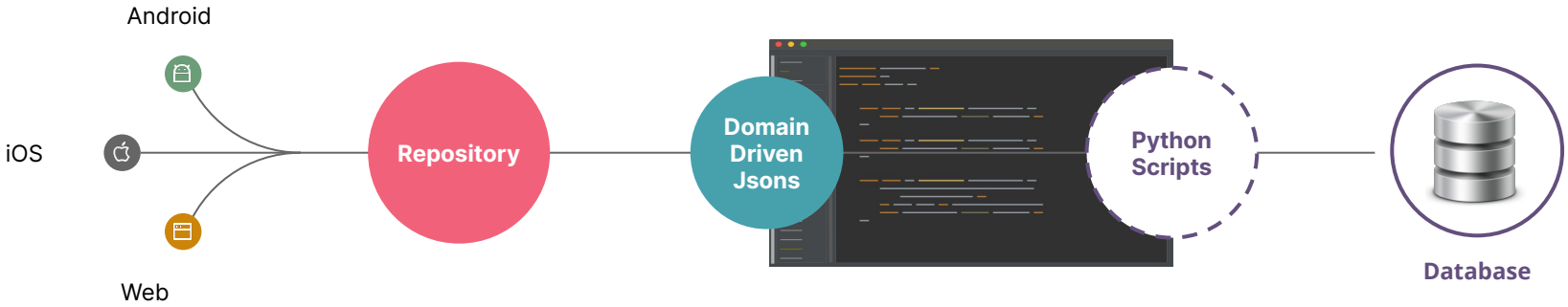
Falta de documentação centralizada para geração de queries

Inserção de massa de dados no código de forma manual

Solução Implementada

/thoughtworks

Arquitetura genérica - Test Mass Handler

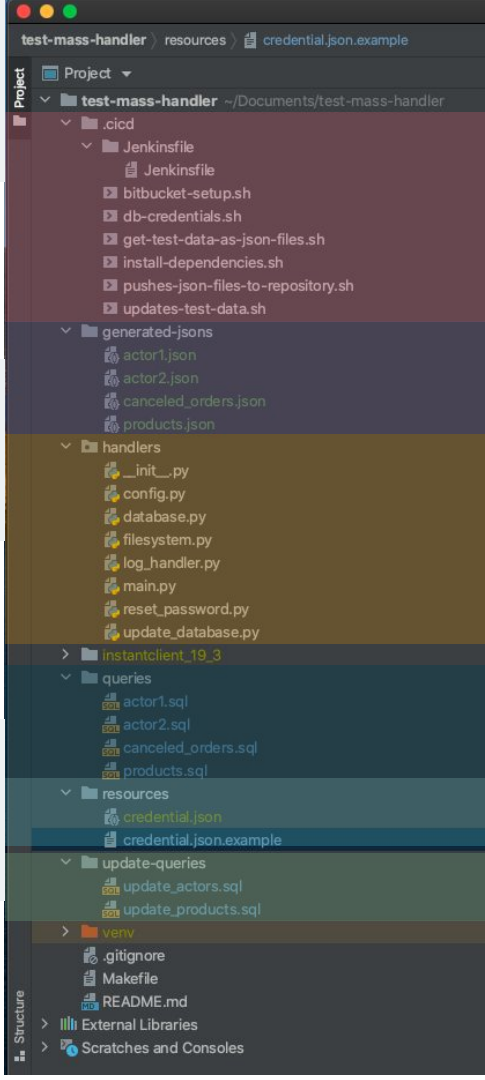


Python Scripts → Database → Domain Driven Jsons → Repository

Estrutura do projeto

O objetivo principal é fornecer arquivos .json com dados de teste confiáveis para as mais diversas necessidades de testes das equipes (automatizado, manual, exploratório, etc).

.cicd	Scripts para automação das tarefas (Pipeline).
generated-jsons	Massa de dados geradas em formato .json através das queries executadas
handlers	Scripts em python para conexão com o banco de dados, execução das queries e saída de dados em forma de .json
queries	**SELECTS** executadas pelos script python na base de dados (.sql)
resources	Armazenamento das credenciais necessárias
update-queries	**UPDATES** executados e em seguida **COMMIT** na base de dados.



Tool Stack

Ferramentas utilizadas pela cliente e adaptadas a nossa necessidade.

Codebase principal

Base de dados e arquivos de consulta utilizados pela codebase

ORACLE®

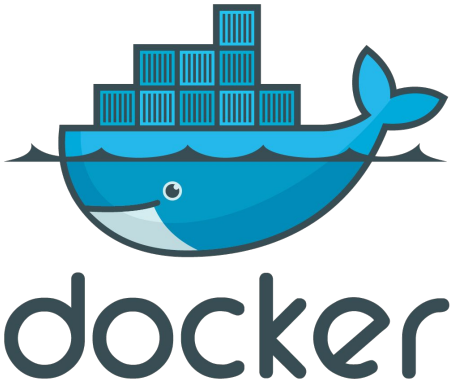
DATABASE

SQL

Armazena a codebase, *pipeline as code* e arquivos de saída .json

Automação de tarefas e orquestração dos scripts .sh

Construção da pipeline e provisionamento de ambiente local.



Automação dos passos da pipeline.

Como garantimos os compliances de segurança?

Acima de tudo, a nossa solução deve respeitar os princípios de integridade, disponibilidade e confidencialidade dos dados.

Para isso, além de usuários com privilégios adequados para realizar apenas as operações que lhe são permitidas, também observamos os seguintes aspectos na arquitetura da nossa solução:

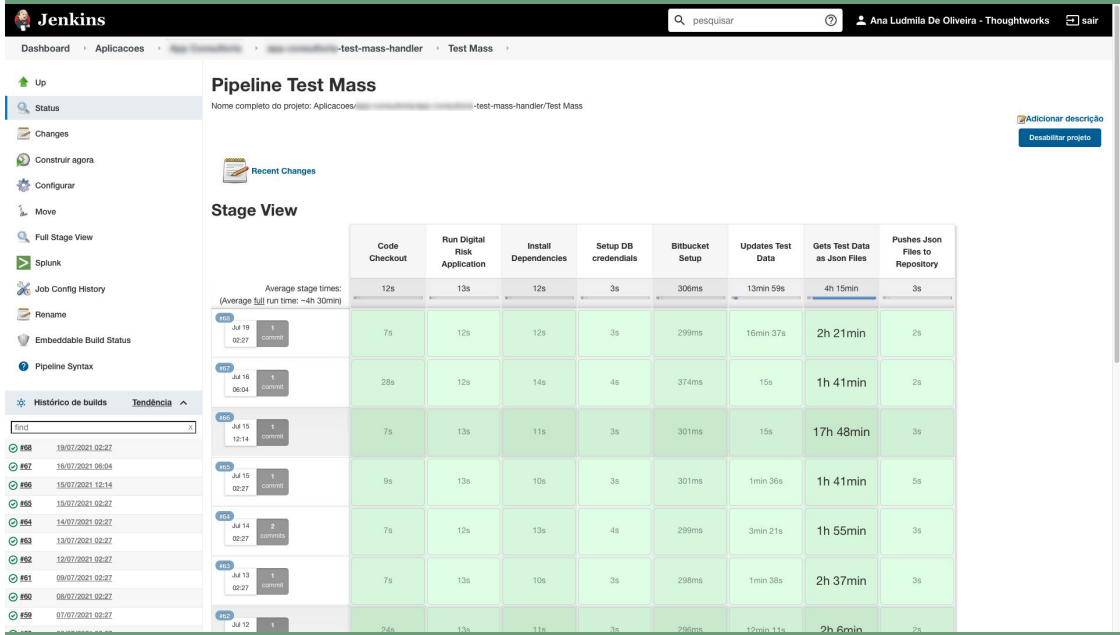
1.
VPN

2.
Credenciais armazenadas em um arquivo local e adicionado ao gitignore

3.
Credenciais criptografadas adicionadas ao Jenkins

Pipeline no Jenkins

Ao criar uma pipeline que executa o ***Test Mass Handler*** todos os dias de madrugada, viabilizamos que todos os dias tenhamos dados aptos a serem utilizados em testes (mesmo que não os façamos) e com isso, dispensamos a necessidade de executar a solução localmente.



O fim das intervenções manuais

Pipeline do Test Mass Handler

Code Check Out

- Inicia o docker container;
- Realiza checkout do código python.

Setup

Dependencies	DB Credentials	Bitbucket
Instala as dependências do ambiente de desenvolvimento necessárias.	Copia as credenciais criptografadas configuradas no servidor do Jenkins para o arquivo **/resources** **/credential.json**	Configurar as credenciais que serão utilizadas para realizar o commit e push dos arquivos ao fim da execução para o Bitbucket

Handler Test Data

Update	Select
Executa todos os arquivos .sql que estão no diretório **/update-queries**, setando os dados que são necessários. **UPDATE / COMMIT**	Executa todos os arquivos .sql que estão no diretório **/queries**, transformando os seus resultados em arquivos .json e armazenando na pasta **/generated-jsons**. **SELECT**

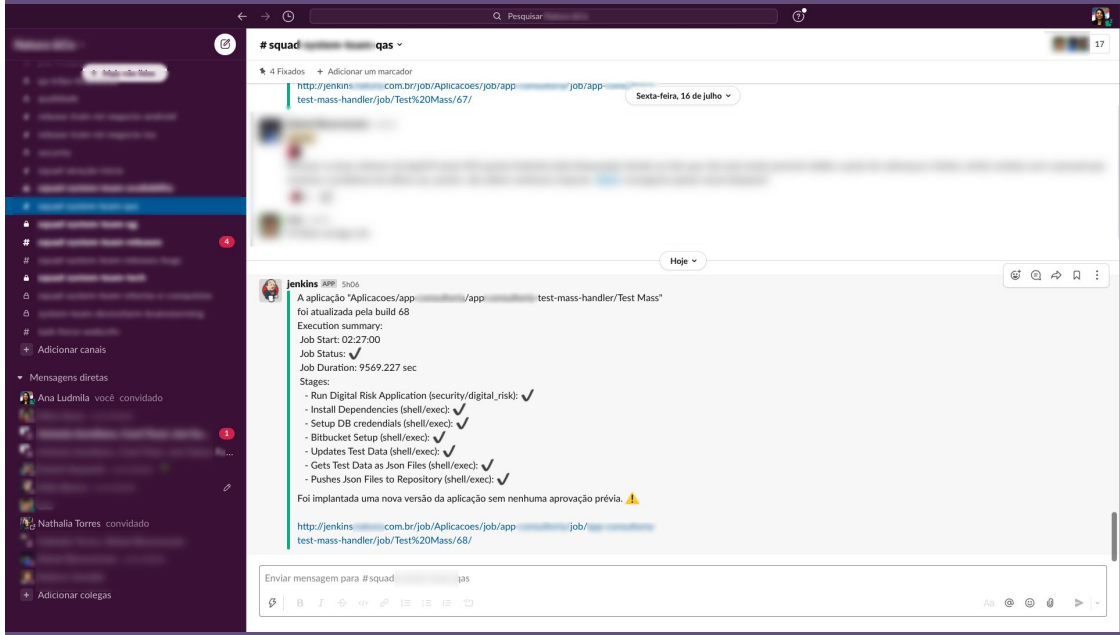
Push

Realiza um commit dos arquivos que estão na pasta generated-jsons, e em seguida realiza push para o repositório no bitbucket.

A pipeline é executada diariamente às 2h da manhã, e ao final da execução envia uma notificação para o time via Slack

Notificações via Slack

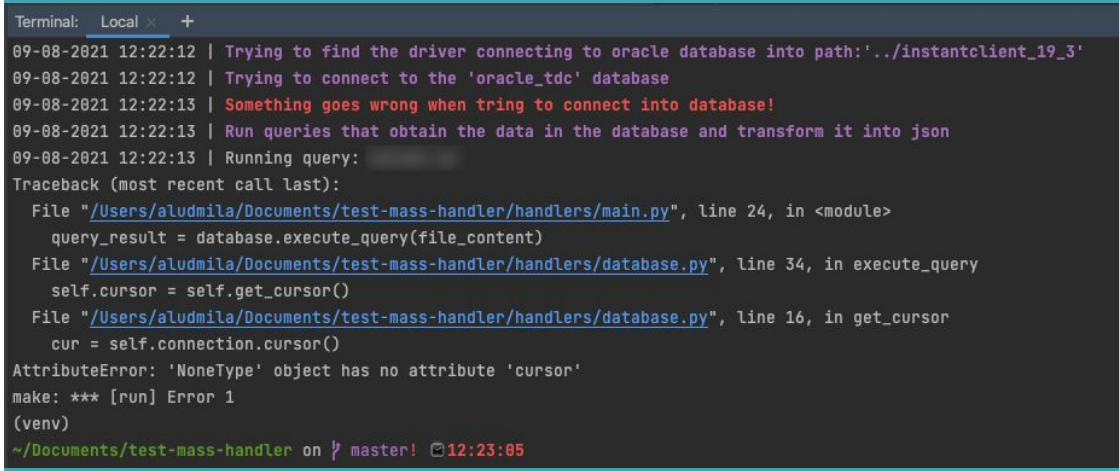
Através de notificações sobre o feedback de execução da pipeline no Slack, foi possível ter uma visibilidade constante para todo o time sobre o sucesso ou falha da obtenção dos dados para teste.



Feedback rápido e propriedade compartilhada

Logs no terminal

Importante também nos casos em que a pipeline falha, pois conseguimos identificar o problema de maneira mais ágil!



Feedback sobre a execução dos scripts Python

Quais foram os nossos ganhos?

/thoughtworks

Reuso e colaboração entre as squads

Outras squads que atuam na mesma jornada podem utilizar os dados gerados pela pipeline para realizar os seus testes antes de integrar seus módulos na super app.

Documentação viva

Código quebrado = Documentação obsoleta

Através da atualização automática da nossa massa de testes, obtemos uma fonte confiável e rápida de documentação.

Feedback em tempo hábil

Através da integração do Jenkins com o Slack, temos um feedback diário sobre a situação das massas de teste.

Podemos verificar massas indisponíveis, queries quebradas e outros incidentes antes de criarmos o Release Candidate.

QA 3.0

O projeto habilitou as QAs a trabalharem na dimensão de negócios, dimensão técnica e dimensão DevOps.

Desta forma, contribuímos com o time a entregar valor de negócio frequentemente e com qualidade, independente dos papéis.

100%

de tempo livre para automatizar outros processos e elaborar tech talks!

Estamos ansiosas para trabalhar com você!

Ana Ludmila
Senior Consultant - Quality Analyst
aludmila@thoughtworks.com

Nathalia Torres
Consultant - Quality Analyst
nathalia.torres@thoughtworks.com

/thoughtworks