REUNIÃO ALL TAX KT – 30/08/2023

Notas: Maycon R. V. Pereira

TIMP – TAX INTELLIGENCE & MANAGEMENT PLATFORM

O TIMP é uma plataforma WEB que funciona usando tecnologias como:

- Javascript
- HTML
- CSS
- XS Engine as the runtime server
- SAP HANA as Database

Existe um processo de minificação que roda no SAP chamado xtrider, que basicamente tem uma tarefa que faz um “Uglify” dos arquivos. Mas nem todas as funções do ultimo Javascript não são suportados pelo Uglify. Por isso sugere-se utilizar a versão 5 do EcmaScript

Uglify: é uma biblioteca JavaScript para a minificação de arquivos JavaScript. Uglificar um arquivo é minificá-lo usando Uglify. A uglificação melhora o desempenho reduzindo, ao mesmo tempo, a legibilidade.

Todo o projeto roda no servidor do banco de dados na versão clássica, pois dependendo do projeto pode haver uma versão clássica e uma versão avançada da ferramenta.

Na Petrobras, utiliza-se a versão clássica que compartilha o servidor do banco e o servidor do aplicativo, logo o código está dentro do HANA.

O TIMP funciona através de chamadas de REST API que enviam e recebem informações em JSON.

O funcionamento da arquitetura é baseado em 3 capas

- 1ª Capa: Business Logic
- 2ª Capa: UI
- 3ª Capa: Banco de dados

COMPONENTES

Um componente é basicamente uma abstração de alguns procedimentos para gerenciar, criar, atualizar, deletar e executar certos objetos. A plataforma TIMP possui 27 componentes.

- Os componentes têm diferentes tipos.
    - Builders/Executor: componentes para criar esquemas de objeto que serão usados com um template na qual a informação de um cliente poderá ser visualizada. (BRB, BCB, DFG, etc)
    - Conteúdo

- Componentes para criar dados sobre a plataforma, como ADM onde é possível fazer a gestão de usuário e assinar regras e privilégios ou LOG onde é possível visualizar as ações que todos os usuários executam na plataforma.
- Componentes que são usados para expandir os dados de um cliente com algumas customizações ou ajustes, onde normalmente são usados para criar ou adicionar informações complementar/específica, como MDR, TCC, ATR

Os componentes não utilizam linhas das tabelas da base dos clientes. Normalmente são adicionados *calculation views* que encapsulam a lógica que agregam os dados como join, groups, totals, etc. e esses dados são vinculados dentro de uma estrutura, que então são utilizadas pelos componentes como uma referência para buscar os dados do banco de dados.

BUILDER/EXECUTE

- BRB: para criar relatório, com colunas e campos calculados
- BRE: para criar regras de negócios. Geralmente preenche-se um valor na coluna de um relatório dependendo de uma lógica. São usados para aplicar regras com determinadas lógicas como criar várias condições por exemplo se a empresa for igual = 100 colocará esses valores na coluna.
- BCB: para criar uma hierarquia para visualizar os dados de vários relatórios nas mesmas páginas totalizando os valores
- TDK: Para criar gráficos e dashboards baseados nos dados transacionais
- BFB: para criar layouts de livros fiscais
- TFB: para criar páginas fiscais usando as páginas criadas em BFB.
- BPMA: criar processos de negócios que irão automatizar algumas operações, como executar um relatório de

CONTENT:

Conteúdo da plataforma: priveleges, roles, meta-dados de objetos como relatórios, log de ações, etc.

- ADM: cria usuários, assina regras e cria parâmetros como customização de diversos cenários
- ATR: cria os meta-dados para as estruturas para ajudar a criar queries em outros componente:
- Calendar: criar atividades específicas e assinam um processo em um dia especifico.
- TBD: para criar e salvar arquivos como imagens e pdfs
- TSB: para criar meta-dados para executar uma simulação
- TCM: para criar uma pacote de conteúdo que pode ser exportado e importado em outros componentes.
- LOG: para traçar as ações de um usuário na plataforma

Conteúdo de taxas:

- TCC: informações de crédito com dados complementares
- MDR: meta-dados como códigos de ajustes e filtros de conta
- TPC: pagamentos que podem ser criados com alguns valores no ERP.

- TAA: ajustes fiscais
- NF: Para visualizar todos os dados de uma NF
- NCM: Para transporte de conteúdo entre componentes.
- TFP: Define períodos fiscais para executar os dados da base de dados do cliente.

INTEGRAÇÕES

São componentes normalmente de conteúdo que precisam acionar diversas atividades no ERP e a conexão é feita por requisições http. Baseado em xml ou formato JSON.

Componentes como:

-TAA: para acionar procedimento de contabilização em um ajuste.

-BSC: para fazer uma correção em valores de uma ERP.

-TPC: para executar pagamentos.

TIPOS DE ARQUITETURA

- **Three Tier Architecture (Arquitetura de Três Níveis) - Classic**

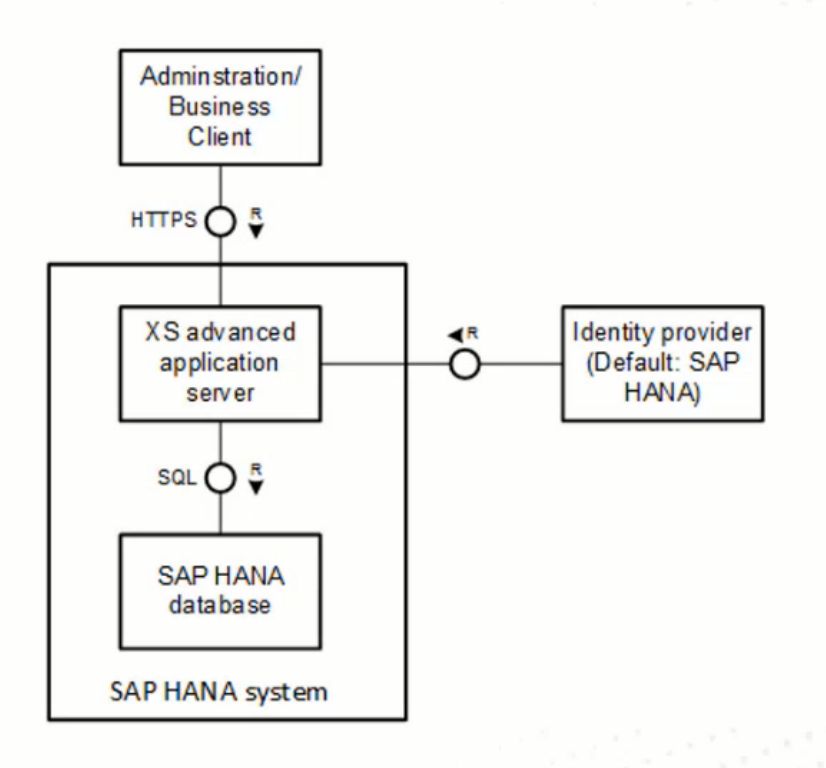

Figura 1 - Classic architecture

- **Three Tier Architecture (Arquitetura de Três Níveis) – Advanced**

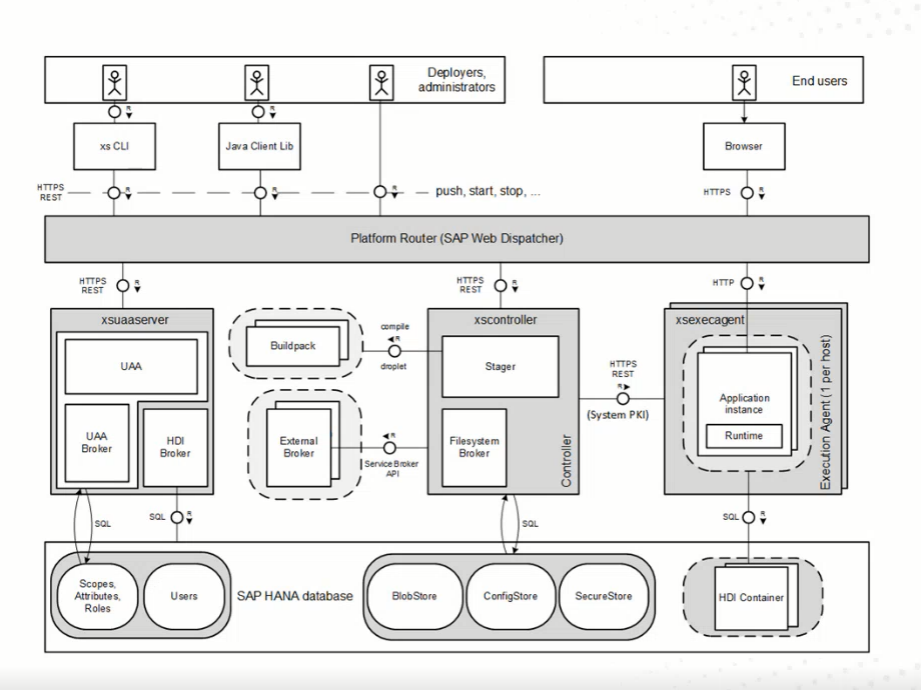

Figura 2 - Advanced architecture

QUESTIONS ❓

Em cenário de erro em uma tabela, o que fazer?

-Abrir o inspecionar, fazer o processo e analisar na aba “Network” a requisição que está mostrando erro.

-Na requisição, pode-se a resposta obtida da requisição na aba “Response”. Na aba Headers, as informações de cabeçalho, tais como URL, status code e cookies podem ser visualizados. Pela URL, identifica-se o endpoint a ser analisado no código.

-Após identificar o endpoint, procura-se no arquivo de endpoint do componente (ex: mdr/server/endpoint.xsjs), o controller que está sendo chamado e então no arquivo controlador, procurar pela função que está dando problema.

-Obs.: No Editor do SAP Hana, quando for feita uma importação de um componente para um arquivo de outro componente, não pode ser feita uma importação direta pois isso pode dar erro na hora de minificar. Com isso, caso seja preciso fazer uma importação de um arquivo de um componente para outro, é preciso importar ele em api/api.xsjslib e então no componente que irá usar o arquivo, importa-se o api e usa-se o elemento importado, através do ponto.

Como encontrar a query que está rodando em uma função

-Coloca-se o breakpoint na linha onde a execução está sendo feita e ao ir entrando nas respectivas chamadas (Step in), deve-se encontrar onde a função da ORM (ex. READ, CREATE..) é chamada e então utilizar o 5º botão do

debugger (EVALUATE EXPRESSION) e adicionar ao objeto, a propriedade simulate passando o valor true.

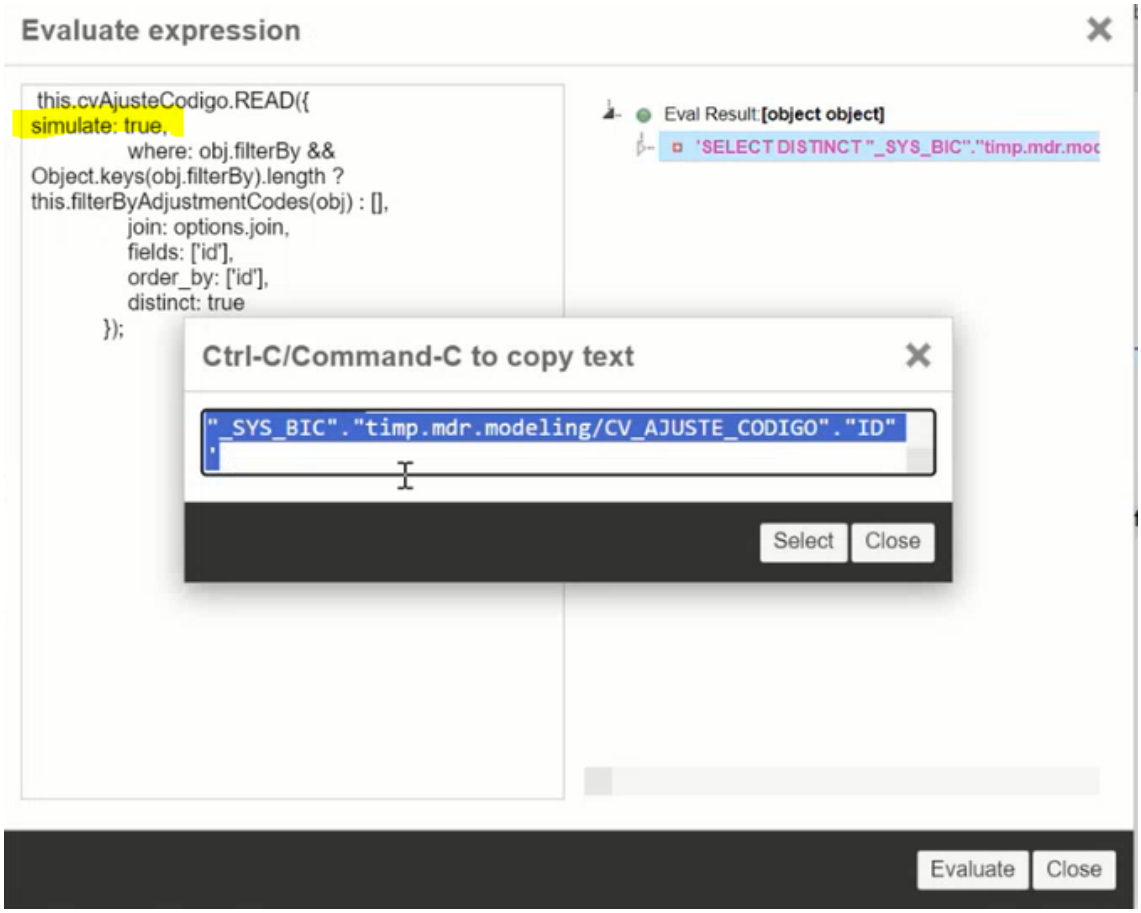

Figura 3 - Using the "simulate" property