

Sumário

Setup de desenvolvimento. On Boarding e primeiros passos.	2
Instalação de requisitos	2
Padrões de desenvolvimento	4
Tipos HANA vs tipo MODEL chsarp	4
Descrições gerais de nomenclatura	4
Campos de usuário	4
Tabelas	4
Variáveis	5
Nomenclatura de Metodos	5
Implementação de conexões	5
Projeto Automação	5
Exceções	8
Projeto automação	8
Boas práticas de organização dos códigos	8
Considerações importantes:	8
MVC	8
Padrão de criação de models	9
Como depurar serviços do Windows	9
Utilização do GIT	10
Acesso:	10
Como utilizar o GIT:	10
Como utilizar o GIT no Visual Studio.	10
Como commitar o desenvolvimento realizado (pull/merge request)	10
Como criar issue! Padrão de geração para issue	10
Issue de nova model sem tabela	10
Issue de novo zoom	12
Issue de nova model tabela SAP	13
Issue de getter e/ou setter (metodos em geral)	13
Issue de nova Model e tabela customizada	14
Modelo para metodos de delete	15
Criando telas padrões do SAP	16
Vídeo de criação de telas	16
Padrões de testes e simulações	17
Automação	17

Como testar o projeto automação:.....	17
Checklist para simulação	17
Como fazer testes unitários de metodos em background	17
Como utilizar o git nos testes	17
Documento de teste.....	17
Snippets e trechos de codigo padronizado	17
Como usar os snippets	17
Download dos snippets	18
Como importar os Snippets.....	18
Como implementar os snippets	18
Instalação SAP DI API Versão 10.....	18
PreRequisites.....	18
DI API	18
Instalação SDK	19

Setup de desenvolvimento. On Boarding e primeiros passos.

Instalação de requisitos

Para iniciar as atividades de desenvolvimento, utilize as ações indicadas abaixo na sequência indicada.

1. Baixe a IDE de desenvolvimento Visual studio community 2019 (comunidade). Ela pode ser encontrada no site da Microsoft ou pelo link [Baixar o Visual Studio 2019 para Windows e Mac \(microsoft.com\)](#)
 - a. Selecione os componentes WEB utilizados para realizar a instalação.
2. Baixe a IDE de desenvolvimento “Visual studio Code”. Pode ser baixado na URL: [Download Visual Studio Code - Mac, Linux, Windows](#)

3. Crie um usuario no GITLAB. Caso já tenha um usuário não é necessário criar.
 - a. Para criar um usuario acesse o endereço <https://gitlab.com/>
 - b. Envie para o Wellington Previatti o seu usuario do GIT criado. Pode formalizar através do e-mail wellington.previatti@sixconsult.com.br
4. Baixe e instale os .net framework' para iniciar os desenvolvimentos.
 - a. Instale e baixe o .net framework mais atual (developer/sdk).
 - i. [Download .NET Framework 4.8 | Free official downloads \(microsoft.com\)](#)
 - b. Baixe o net core 2.1 e o mais atual.
 - i. [Download .NET Core \(Linux, macOS, and Windows\) \(microsoft.com\)](#)
5. Efetue o download da pasta de requisitos SAP. Ela será muito importante para uso de integração com SAP.
 - a. https://1drv.ms/f/s!Ag_drMqC3F6kn_5Maltux_7rbqFGBA
 - b. Sap 10, hana2.0
 - c. https://1drv.ms/u/s!Ag_drMqC3F6koPttmCYoOuksrkrwA?e=MBuJKo
6. Baixe o GIT no link
 - a. <https://git-scm.com/download/win>
7. Após o download, execute os passos conforme indicado abaixo.

Todas as pastas devem estar no mesmo diretorio e com o nome indicado. Sempre indique hanadb para os hosts

- a. 1-Instalar prerequisites
 - b. 2-Instalar diapi "DI API"
 - c. 3-Instalar diapi 64 como administrador (deve ser como administardor)
"DIAPL.x64"
 - d. 4-Instalar hdb clientes com nome
SAP_HANA_CLIENT_x64_1.00.120.114.1542121741
 - e. 5-Instalar hdb SAP_HANA_CLIENT_x86
 - f. 6-Instalar SDK
 - g. 7-Instalar client.x64
 - h. 8-Descompacte e instale o hana_studio "Instalar prerequisites"
 - i. Para saber como proceder na configuração! Verifique o vídeo <https://lnnk.in/jBah>
8. Efetue o download do aplicativo Discord. Este aplicativo é utilizado para comunicação da equipe de desenvolvimento.
 - a. [DISCORD | Seu Lugar para Papear e Ficar De Boa](#)
 - b. Após o download e criar seu usuario, se inscreva no grupo da SIX através do link abaixo:
 - i. <https://discord.gg/3jXfA6PDxR>

Padrões de desenvolvimento

Tipos HANA vs tipo MODEL chsarp

Este tipo de relação de-para serve para padronizar quais os tipos sua model deve ter para que fiquem equivalentes ao tipo HANA

	A	B	C	D	E	F	G	H	I
1	Regras de conversão de model								
2	Tipo tabela HANA						Tipo model		
3	INTEGER				VIRA		int?		
4	NVARCHAR				VIRA		string		
5	NVARCHAR				Se valor y/n ou s/n		Bool?		
6	TIMESTAMP				VIRA		datetime		
7	SMALLINT				SE FOR HORA		timespan		
8	SMALLINT				Se não hora		int?		
9	DECIMAL				VIRA		decimal		
10					VIRA				

Descrições gerais de nomenclatura

Campos de usuário

Criação de tabelas de usuários dos projetos do **AUTOMAÇÃO** devem obrigatoriamente começar com **AUT_**.

Criações de tabelas de usuário de projetos que não são produtos, devem ser obrigatoriamente começar com **SX_**.

Campos Booleanos

Os campos bool no sap devem ser criados como inteiro. Onde o valor 0 (zero) sera false e 1 (um) será true. O Campo deverá ter o prefixo U_Log, ficando o nome U_Log_[Nomecampo]

Campos data e hora.

Os campos no sap do tipo data, deverão estar como data. Os campos do tipo hora deverão estar marcados como hora.

Tabelas

O complemento dos nomes das tabelas deve ser em letras maiúsculas abreviando no seguinte modelo.

Máximo de caracteres para criação de tabelas 18 caracteres.

Quantidade de palavras	Nome	Tabela
1 palavra	Etiquetas	AUT_ETIQUETAS
2 palavras (4 caracteres) _(4caracteres)	Histórico etiquetas	AUT_HIST_ETIQ

3 palavras (3 caracteres) _(3caracteres) _ (3caracteres)	Histórico etiquetas da contagem	AUT_HIS_ETI_CON
---	---------------------------------------	-----------------

Variáveis

Fica proibido a criação de variáveis somente com Iniciais, tais como gsg, tdd, oqp e outros. Exceto quando utilizado em cursores como for. Ex. i = 0; j=0;

Variáveis devem ter todo ou parte do nome da model ao qual se refere ou então o nome da finalidade a que se destina. Ex.

genericScripts = new GenericScripts(); , expedicao = new ExpedicaoModel(); , contador = 0; , controleDeLeiturasX = 0; controleDeLeitura = false;

Ex. incorretos:

exp = new ExpedicaoModel(); , a = 0; tdd = false;

Nomenclatura de Metodos

Metodos deverão ter um padrão de sufixo e prefixo, onde:

1. Metodos que realizam ações no sistema, tais como transferencias, movimentações ou atualizações compostas que envolvam mais de uma tabela. Deverão ter prefixo “Do”
 - a. CriarAlgo será DoCriarAlgo
2. Metodos que utilizam conexão de banco via ADO.NET. Deverão ter sufixo “ADO”. Ex:
 - a. DoCriarAlgo será DoCriarAlgoADO.
3. Metodos que utilizam conexão ou manipulam o banco via DI’API. Deverão ter sufixo “DIAPI”. Ex:
 - a. DoCriarAlgo será DoCriarAlgoDIAPI
4. Metodo que vai realizar somente uma consulta de dado, seja ela em tabela ou modelos. Deverão começar com Get. Ex:
 - a. ValorRequerido será GetValorRequerido
5. Metodo que vai validar um ou mais campos de uma model, lista ou variavel. Deverá começar com Valída. Ex:
 - a. impressoesValidadas será ValidarImpressao
6. Metodo que vai retornar um boleano com o objetivo de apenas checar se a existência de um registro na tabela, model, lista ou variavel. Deverá ter o prefixo “Tem” ou “Is”

Ex:

 - a. configuracaoExistente será TemConfiguracaoExistente
 - b. configuracaoExistente será IsConfiguracaoExistente

Implementação de conexões

Projeto Automação

O Projeto automação trabalha com tres metodos de conexão com o banco SAP. Sendo eles a camada de serviço em REST denominada “Service layer”, camada de serviço em C# denominada DIAPI e a última que deverá somente ser usada para consultas (tabela padrão sap) ADO.NET. Para uso destas na automação seguem exemplos de implementação.

Conexão via DI/API

O Projeto utiliza uma classe que valida a conexão automaticamente basta instanciar a Classe SQL utilizando using e chamar o método DoQuery ou DoQuerys para Listas e passar como parâmetro um Select.

Como criar metodos na api

https://1drv.ms/v/s!Ag_drMqC3F6koO0IXVl1jr0ullqUWQ?e=hqLW89

Exemplo de uso para implementação da classe select:

[Como implementar a classe SQL \(\\$2083009\) · Snippets · SIX Consult / Geral / Automacao · GitLab](#)

Em Casos em que é necessário fazer Insert, Update ou Delete é obrigatório utilizar o método ExecuteQuery.

Exemplo de Implementação de GET com Tipo Primitivo (apenas uma coluna do select):

```
1 referência | 0 exceções
private int GetLocalização(NFEntradaItensModel nfe)
{
    int AbsEntry = 0;

    using (SQL sql = new SQL())
    {
        string query = string.Format(@"SELECT ""AbsEntry"" FROM ""OBIN""
                                     WHERE ""WhsCode"" = '{0}' AND ""SysBin"" = 'Y'", nfe.WhsCode);

        AbsEntry = sql.DoQuery<int>(query);
    }

    return AbsEntry;
}
```

Exemplo de Implementação GET com Model:

```

0 referências | 0 exceções
private NFEntradaItensModel GetLote(int DocEntry, int ObjType)
{
    NFEntradaItensModel nota;

    using (SQL sql = new SQL())
    {
        string query = string.Format(@"SELECT T0.""DocEntry"", T0.""ItemCode"", T1.""MdAbsEntry"",
                                         T2.""BinAbs"", T2.""SnBMDAbs"",
                                         T3.""DistNumber"", T4.""BinCode""
                                         FROM OITL T0
                                         INNER JOIN ITL1 T1 ON T0.""LogEntry"" = T1.""LogEntry""
                                         LEFT JOIN OBBQ T2 ON T1.""MdAbsEntry"" = T2.""SnBMDAbs""
                                         INNER JOIN OBIN T3 ON T2.""SnBMDAbs"" = T3.""AbsEntry""
                                         INNER JOIN OBIN T4 ON T2.""BinAbs"" = T4.""AbsEntry""
                                         WHERE T0.""DocEntry"" = {0} and T0.""DocType"" = {1}", DocEntry, ObjType);

        nota = sql.DoQuery<NFEntradaItensModel>(query);
    }

    return nota;
}

```

Exemplo de Implementação GET com Lista de Tipo Primitivo (usado para pegar listas com tipos simples):

```

0 referências | 0 exceções
private List<string> GetDepositos()
{
    List<string> depositos = new List<string>();

    using (SQL sql = new SQL())
    {
        string query = string.Format(@"SELECT ""WhsCode"" FROM ""OWHS"" ");

        depositos = sql.DoQuerys<string>(query);
    }

    return depositos;
}

```

Exemplo de Implementação com Lista e Model:

```

1 referência | 0 exceções
private List<NFEntradaItensModel> GetNotaEntrada(string fornecedor, int nota, string serie)
{
    List<NFEntradaItensModel> nfentrada = new List<NFEntradaItensModel>();

    using (SQL sql = new SQL())
    {
        string query = string.Format(@"SELECT
                                         T0.""DocEntry"",T0.""DocNum"",T0.""ObjType"",T0.""CardCode"",T0.""BPLId""
                                         , T0.""BPLName"",T0.""Serial"", T0.""SeriesStr""
                                         , T1.""LineNum"", T1.""ItemCode"", T1.""Description"", T1.""Quantity"", T1.""WhsCode""
                                         , T2.""ManBtchNum"", T2.""PurPackUn""
                                         , T3.""BinActivat""
                                         FROM ""OPCH"" T0
                                         INNER JOIN ""PCH1"" T1 ON T1.""DocEntry"" = T0.""DocEntry""
                                         INNER JOIN ""OITM"" T2 ON T2.""ItemCode"" = T1.""ItemCode""
                                         INNER JOIN ""OWHS"" T3 ON T3.""WhsCode"" = T1.""WhsCode""
                                         WHERE T0.""CANCELED"" != 'Y' AND T0.""CardCode"" = '{0}'
                                         AND T0.""Serial"" = '{1}' AND IFNULL(T0.""SeriesStr"", '') = '{2}', fornecedor, nota, serie);

        nfentrada = sql.DoQuerys<NFEntradaItensModel>(query);
    }

    return nfentrada;
}

```

Exemplo de Insert no Banco:

```
O referências | O exceções
public void InserirEtiqueta(string codigoEtiqueta, int quantidade)
{
    using (SQL sql = new SQL())
    {
        string query = string.Format(@"INSERT INTO ""AUT_Etiquetas"" (""U_CodEtiqueta"", ""U_QtdAtual"")
        VALUES ('{0}', '{1}')" , codigoEtiqueta, quantidade);

        sql.ExecuteQuery(query);
    }
}
```

Exemplo de Delete no Banco:

```
O referências | O exceções
public void DeletarEtiqueta(string codigoEtiqueta)
{
    using (SQL sql = new SQL())
    {
        string query = string.Format(@"DELETE FROM ""AUT_Etiquetas"" WHERE ""U_CodEtiqueta"" = '{0}' ", codigoEtiqueta);

        sql.ExecuteQuery(query);
    }
}
```

Exemplo de Update no Banco:

```
O referências | O exceções
public void AtualizarEtiqueta(string codigoEtiqueta, int quantidade)
{
    using (SQL sql = new SQL())
    {
        string query = string.Format(@"UPDATE ""AUT_Etiquetas"" SET ""U_QtdAtual"" = {0}
        WHERE ""U_CodEtiqueta"" = {1} ", quantidade, codigoEtiqueta);

        sql.ExecuteQuery(query);
    }
}
```

Exceções

Projeto automação

Para exceções dentro do projeto automação, deverá ser utilizado o metodo ColetorException ao invés do Exception

Boas práticas de organização dos códigos

Considerações importantes:

- **É proibido a utilização de implementações de metodos com retorno do tipo Object, exceto em casos especificos de plugins (pcs -> vide conceito de PCS);**

MVC

Projetos desenvolvidos em .net com asp.net mvc deverão ter alguns padrões de organização.

Padrão de criação de models

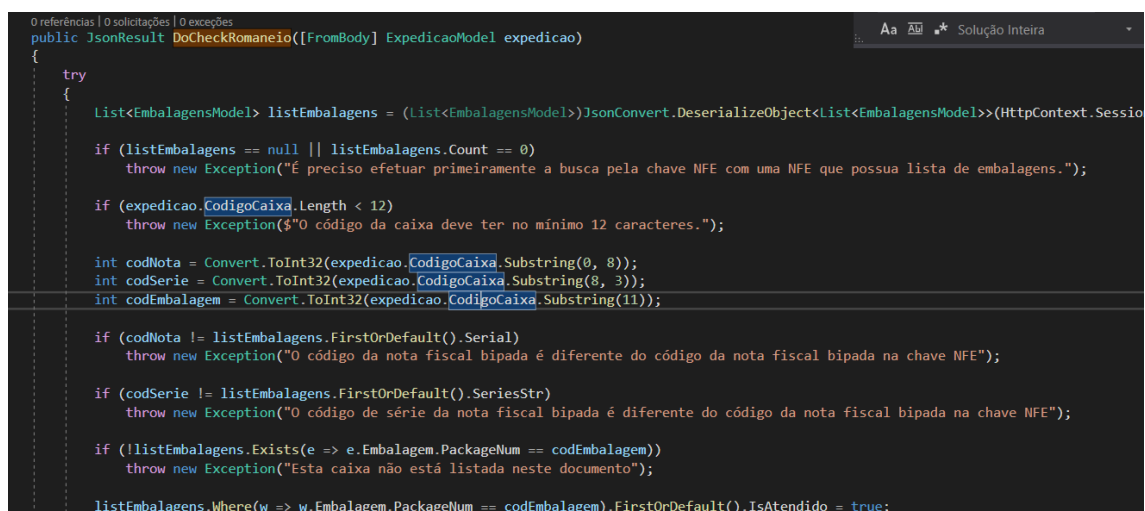
Para a geração de modelos (models) é importante visualizar o vídeo em anexo

https://1drv.ms/v/s!Ag_drMqC3F6koOx6XYPY4VMO5mwZjA?e=jwLQYJ

JsonResult

Métodos com retorno `jsonResult`, ou seja, que são responsáveis por receber e responder json devem ter a única obrigação de receber os métodos e tratar o retorno. Não devem existir regras de negócio ou validações implementadas.

Ex. de um caso errado:



```
0 referências | 0 solicitações | 0 exceções
public JsonResult DoCheckRomaneio([FromBody] ExpedicaoModel expedicao)
{
    try
    {
        List<EmbalagensModel> listEmbalagens = (List<EmbalagensModel>)JsonConvert.DeserializeObject<List<EmbalagensModel>>(HttpContext.Session.GetString("listEmbalagens"));

        if (listEmbalagens == null || listEmbalagens.Count == 0)
            throw new Exception("É preciso efetuar primeiramente a busca pela chave NFE com uma NFE que possua lista de embalagens.");

        if (expedicao.CodigoCaixa.Length < 12)
            throw new Exception($"O código da caixa deve ter no mínimo 12 caracteres.");

        int codNota = Convert.ToInt32(expedicao.CodigoCaixa.Substring(0, 8));
        int codSerie = Convert.ToInt32(expedicao.CodigoCaixa.Substring(8, 3));
        int codEmbalagem = Convert.ToInt32(expedicao.CodigoCaixa.Substring(11));

        if (codNota != listEmbalagens.FirstOrDefault().Serial)
            throw new Exception("O código da nota fiscal bipada é diferente do código da nota fiscal bipada na chave NFE");

        if (codSerie != listEmbalagens.FirstOrDefault().SeriesStr)
            throw new Exception("O código de série da nota fiscal bipada é diferente do código da nota fiscal bipada na chave NFE");

        if (!listEmbalagens.Exists(e => e.Embalagem.PackageNum == codEmbalagem))
            throw new Exception("Esta caixa não está listada neste documento");

        listEmbalagens.Where(w => w.Embalagem.PackageNum == codEmbalagem).FirstOrDefault().IsAtendido = true;
    }
}
```

Ex. de um caso correto:



```
0 referências | 0 solicitações | 0 exceções
public JsonResult JuntarEtiquetasDto([FromBody] WebEtiquetaModel etq)
{
    try
    {
        Library.Etiquetas etiquetas = new Library.Etiquetas(_config, HttpContext);
        AutEtiquetasDto etiqueta = etiquetas.JuntarEtiquetasDto(etq.CodEtiquetas);
        return Json(new ErroModel { ID = 0, mensagem = string.Concat("Etiquetas ", etiqueta.U_CodEtiqueta, ", Gerada com sucesso.") });
    }
    catch (Exception ex)
    {
        var error = new ErroModel { ID = 1, mensagem = ex.Message };
        return Json(error);
    }
}
```

Como depurar serviços do Windows

https://1drv.ms/v/s!Ag_drMqC3F6koPQc33cRL0qvuAd_fQ?e=sCuMwX

Utilização do GIT

Git é uma ferramenta de controle de versão e modificações de código fonte usado pelos programadores.

Caso tenha alguma dúvida do que é um GIT e os conceitos básicos, verifique este vídeo do youtube. O conceito usado pela Six é o mesmo, a única diferença é que usamos uma interface gráfica para rodar os comandos e nosso repositório é na nuvem.

[\(532\) Entenda o GIT em 10 MINUTOS - YouTube](#)

Acesso:

Acesse através do link abaixo:

[Development · Boards · SIX Consult · GitLab](#)

Como utilizar o GIT:

https://1drv.ms/v/s!Ag_drMqC3F6koPUoPCqmt0TpkfCzjQ?e=N555wD

Como utilizar o GIT no Visual Studio.

<https://lnnk.in/fnaM>

Como commitar o desenvolvimento realizado (pull/merge request)

<https://lnnk.in/embl>

Como criar issue! Padrão de geração para issue

De forma geral, lembre-se de que toda issue existente deverá ter uma ação de atuação. E não apenas o título deverá ter o conteúdo indicado abaixo assim como os passos para chegar na solução proposta.

Issue de nova model sem tabela

****Dependência****

#435 , #433 ,#434

****Visão geral****

Criar model para envio e retorno de comunicação entre os metodos de reporte

****Onde gerar****

Esta model deve ser gerada na própria api. Em models da api.

![[image|/uploads/b313460e524a377beffa91c844cb1b3/image.png]]

****Sobre a atividade****

Gerar ****model**** conforme definição:

****Como gerar model****

https://1drv.ms/v/s!Ag_drMqC3F6koOx6XYPY4VMO5mwZjA?e=jwLQYJ

Nome do model: ****ReporteModel****

****Snippet****

AutSnippetMdB

Campo	Tipo 1	Tipo 2	Observação	Onde aplica T=Tabela, M=Model	Proteção
Usuario	string		Valor do usuario da operação	M	Publico
Recurso	string		Codigo do recurso a nivel de usuario (resCode)!	M	Publico
Orsc	ORSC		Valor referente ao objeto recurso apos validação	M	Internal
Ordem	int		Referente ao numero Docentry da ordem	M	publico
Owor	Owor		Referente ao objeto da ordem relacionada	TM	Internal
Itinerario	string		Codigo do itinerario	M	Publico
ORST	ORST		Valor objeto para itinerario	M	Internal
QtdParaReportarBoas	Double		Quantidade a reportar de unidades boas	M	Publico
QtdParaReportarBoasUN2	Double		Quantidade a reportar de unidades boas	M	Publico
Regugos	List de RefuseReportDto		Refugos relacionados ao movimento de reporte	M	Publico
Caminholmpressora	string		Impressora relacionada ao movimento	M	Publico
EtiquetasSuspensaDto	list de string		Relação de etiquetas vinculadas ao reporte suspenso e outros	M	public
EtiquetasSuspensaDto	list de autEtiquetasDto		Relação de etiquetas vinculadas ao reporte suspenso e outros	M	Internal
DtInicial	datetime?		Data inicial do reporte	M	Publico
DtFinal	datetime?		Data final do reporte	M	Publico
Fator	double		Fator nominal do calculo de proporção para MP	M	Publico
_sql	Sql		Contexto de transação para uso futuro	M	Privado

****Valores Defaults****

Criar no construtor vazio do metodo a instancia para evitar null do seguintes valores:

Regugos,Orsc,ORST , EtiquetasSuspensa, EtiquetasSuspensaDto

****Criar metodo de validação****

Criar um metodo com o nome PopulaEValida() publico. Este metodo deverá possuir as seguintes funções

- [] Validar se usuario é null atraves de string.IsNullOrEmpty. Retornar exception coletorException caso vazio com dados informados.
- [] Validar se Recurso é null atraves de string.IsNullOrEmpty. Retornar exception coletorException caso vazio com dados informados.

- [] Efetuar o GetRecursosDiApi e popular o valor de this.Orsc com o resultado, passando o valor da model Recurso como parametro. Issue #435

- [] Validar se Ordem é zero. Retornar exception coletorException caso vazio com dados informados.

- [] Efetuar o GetOrdem pela issue #433 e atribuir o valor de this.Owor com o resultado, passando o valor da model.Ordem como parametro.

- [] Validar se Itinerario é null atraves de string.isNullorEmpty. Retornar exception coletorException caso vazio com dados informados.

- [] Efetuar o GetItinerarioDaOrdem pela issue #434 e atribuir o valor de this.Orst com o resultado, passando o valor da model.Itinerario como parametro.

- [] Caso o valor de EtiquetasSuspensa.count() seja maior que zero E EtiquetasSuspensaDto.count seja zero, deverá realizar foreach e validar cada etiqueta mencionada realizando o GetEtiquetaDiApi(codigoEtiqueta) e atribuir o resultado a lista de EtiquetasSuspensaDto

- [] Validar se DtInicial é diferente de null e DtFinal é diferente de null. Caso sejam null nao fazer nada, Caso sejam diferente de null então validar:

Data final não pode ser inferior a inicial.

****metodo adicional****

Criar metodo internal SetSql(sql sql) com o objetivo de setar a variavel _sql.

****Observação****

****_Colocar todos os descritivos do metodo e atributos na classe conforme imagem abaixo_****

![[image|[/uploads/0e265bc7594a0019a5b33106af3ea97b/image.png]]]

Para facilitar, posicione o cursor acima da linha desejada e digite barra 3 vezes. O Visual studio cria automaticamente a sessão de comentario e voce so precisará preencher o descritivo.

Issue de novo zoom

****Visão geral****

Gerar mecanismo zoom para consultar a respectiva tabela conforme definição do zoom

****Tabela vinculada****

ORST

****Tipo (nome)****

Itinerario

****colunas****

code, desc

****NomeColunas****

Codigo, Nome

****Documentação sobre o ZOOM****

\$2103191

Issue de nova model tabela SAP

```
Escopo base na versão inicial (Verifique no histórico se existem novas referencias):

[ESCOPO_002_-_Programa_de_produção.docx](/uploads/980cb5e8c37399291a330ee585608a4e/ESCOPO_002_-_Programa_de_produção.docx)

**Dependência**

Nenhuma

**Visão geral**

Criar model entity para manipulação de tabelas da ORSC

**Sobre a atividade**

Gerar *model** conforme definição:

**Como gerar model**

https://1drv.ms/v/s!Ag_drMqC3F6koOx6XYPY4VMO5mwZJA?e=jwLQYJ

Nome do model: **ORSC**

**Snippet**

AutSnippetMdDB

Criar no model com o mesmo tipo de dado e os campos indicados na imagem abaixo:

![image](/uploads/b0e6652fa1f7b9a965f1be528cc4a157/image.png)

![image](/uploads/bf6e68f7a3902f24119dbb5bae11f02/image.png)

![image](/uploads/bd2b6af4c2713e9dc176611887ded37/image.png)

Em soma aos campos descritos da imagem. Deverá ser associado o objeto do tipo reserva relacionado a ordem. Este objeto será da model resultante da issue : #367 .

public wor1 LinhasOrdem {Get;set;}
```

Issue de getter e/ou setter (metodos em geral)

```
_Modelo para criação de metodo_

**Visão geral**

Gerar manipuladores para a tabela indicada através dos padrões do automação.

**Dependências**

Esta tarefa so pode ser feita apos a tarefa #343

Voce deve atualizar seus snippets conforme vídeo e links abaixo (mantenha atualizado a cada periodo):

https://1drv.ms/u/s!Ag\_drMqC3F6koPUuOoQiyts9QTqs\_Q?e=T79D71

https://1drv.ms/v/s!Ag_drMqC3F6koOxv3ltfi7dPR0cqPw?e=g1BW29

https://1drv.ms/v/s!Ag_drMqC3F6koOxuGnbxNjRfAX5vVA?e=DMNQbL

**Sobre as alterações**

Deverá ser criado na classe producao.cs ->

![image](/uploads/71d43b08803e983a4d4b2858f16bb875/image.png)

Para criar metodos dentro da classe veja o video abaixo:

https://1drv.ms/v/s!Ag\_drMqC3F6koOOIXV1jr0ullqUWQ?e=hqLW89

Neste arquivo uso o snippet AutSnippetDBBase e crie os seguintes metodos

- public SetAutProgramaDIAP(AUT_PROGRAMA programa) -> Responsável por chamar o metodo internal e iniciar transações. usar snippet AutSnippetDBBase

- internal SetAutProgramaDIAP(SQL sql, AUT_PROGRAMA programa) -> responsável por validar e inserir na tabela pai e filha atraves dos snippets (AutSnippetDBDiApiDoInsertModel, AutSnippetDBDiApiDoInsertModel).

- internal ValidaAutProgramaDIAP(SQL sql, AUT_PROGRAMA programa) -> responsavel por validar o objeto antes da validação. Este nao possui snippet.

Ainda neste arquivo producao.cs crie usando o AutSnippetDBBase o metodo responsavel por get. com parametro docentry que será usado para obter o filtro GetAutProgramaDIAP. Tambem crie o get de lista usando GetAutProgramasDIAP com filtro de status com mais de uma opção de status possiveis para filtrar.
```

Dentro do metodo responsavel pela regra de negocio, use o snippet AutSnippetDBSelectGetDIAPIModel

No metodo get ainda terão campos adicionais e getters adicionais que deverão ser feito para popular campos adicionais da model

- ResName -> nome recurso

- ItinDesc -> nome itinerario

- DescMp -> nome do item de materia prima.

Use os getters já existentes deles para realizar a operação. (não é necessario criar) (GetRecursoByCod, #375)

Com relação as validações, elas deverão ser feitas dentro do metodo ValidaAutPrograma e serão baseadas na model antes de executar os insert.

****Sobre as validações****

- Validar campos bases se estão cadastrados – Item, Depósitos, Recursos ,Itinerario, Unidade Medida, Responsável / Colaborador, Datas no Passado ou incoerentes (**Usar os getters existentes para cada entidade desta para validar! o getter deverá acionar um exception quando nao houver e este deverá ser throw'zado para o usuario**)
- Apresentar mensagens de erros ou de criação/atualização com sucesso
- Deve ter a opção de eliminar campanha e itens, mais eliminar campanha com status de liberado e que tenha quantidade produzida
- Opções de Duplicação Campanha/Linha

Issue de nova Model e tabela customizada

Nome: Solução proposta para a situação

Escopo base na versão inicial (Verifique no histórico se existem novas referencias):

[ESCOPO_002_-_Programa_de_produção.docx](/uploads/980cb5e8c37399291a330ee585608a4e/ESCOPO_002_-_Programa_de_produção.docx)

****Dependência****

Nenhuma

****Visão geral****

Criar tabela no DBSETUP para a entidade mencionada acima. Criar tambem os models envolvidos para manipulação futura.

****Sobre a atividade****

Gerar ****tabela e model**** conforme definição:

****Como gerar model****

https://1drv.ms/v/s!Ag_drMqC3F6koOx6XYPY4VMO5mwZjA?e=jwLQYJ

A tabela deve ser do tipo master do DBSETUP pra ter sequenciais etc

Nome da tabela e do model: ****AUT_PROGRAMA****

****Snippet****

AutSnippetMdDB

****Lembre-se****

Quando o campo que for somente do model, deve ter o atribute ignorecolumn nela. conforme os outros model's existentes.

| Campo | Tipo 1 | Tipo 2 | Observação | Onde aplica T=Tabela, M=Model |

| ----- | ----- | ----- | ----- | ----- |

| Code | db_Alpha | st_None | Campo padrão da tabela! | M |

| docEntry | db_Numeric | st_None | Campo padrão da tabela e código do programa! | M |

| Recurso | db_Alpha | st_None | Valor referente ao recurso | TM |

| ResName | db_Alpha | st_None | Nome do recurso | M |

| Itinerario | db_Alpha | st_None | Valor referente ao Itinerario (code) | TM |

| ItinDesc | db_Alpha | st_None | Descrição do itinerario | M |

| Status | db_Numeric | st_None | 0=planejado, 1=liberado, 2=iniciado, 3=fechado | TM |

| DtPrevista | db_date | st_None | Data planejamento | TM |

| MatPrima | db_Alpha | st_None | Item materia relacionada | TM |

| DescMp | db_Alpha | st_None | Descrição materia prima | M |

| Lote | db_Alpha | st_None | Lote | TM |

| Obs | db_Alpha | st_None | Observação | TM |

| Responsavel | db_Alpha | st_None | Usuario logado que criou o registro | TM |

| Resp_atu | db_Alpha | st_None | Usuario logado que Atualizou o registro | TM |

| AUT_PROG_ITENS | lista de AUT_PROG_ITENS | | Lista do tipo da tabela filha | M |

Nome da tabela e do model: **AUT_PROG_ITENS**

Esta tabela deve ser filha da AUT_PROGRAMA! em caso de duvidas entre em contato com o Sr @alyribas para saber como criar tabelas filhas.

| Campo | Tipo 1 | Tipo 2 | Observação | Onde aplica T=Tabela, M=Model |

| ----- | ----- | ----- | ----- | ----- |

| Sequencia | db_Numeric | st_None | Sequencia auto increment! | TM |

| ItemCode | db_Alpha | st_None | Código do item oitm | TM |

| Qtd_planej | db_Float | st_Quantity | Quantidade planejada | TM |

| Porc_varia | db_Float | st_Quantity | Porcentual de variação | TM |

| Qtd_report | db_Float | st_Quantity | Quantidade já reportada para o item | TM |

| Qtd_Aprov | db_Float | st_Quantity | Quantidade já reportada (SUM) OK para o item (itens bons) | TM |

| Qtd_Ref | db_Float | st_Quantity | Quantidade já reportada (SUM) NOK para o item (itens refugados) | TM |

| Qtd_suc | db_Float | st_Quantity | Quantidade já reportada (SUM) de sucata | TM |

| Qtd_reta | db_Float | st_Quantity | Quantidade já reportada (SUM) de retalho | TM |

| Dt_Entrega | db_Date | st_None | Data prevista da entrega | TM |

| UN_Secondar | db_Alpha | st_None | Unidade de medida secundaria | TM |

| Qtd_Aprov_Secun | db_float | st_Quantity | Quantidade aprovada (sum) da Unidade de medida secundaria | TM |

| DocEntryOpBase | db_Numeric | st_none | Ordem base para estatísticas | TM |

| Fator | db_Float | st_Quantity | Quantidade para fator de DIVIDIR QTDE CONSUMO/QTDE - FUNCAO AUTOMATICA | TM |

| Consumo | db_Float | st_Quantity | Relação de quanto deverá ser consumido para cada porção do reporte | TM |

Modelo para metodos de delete

Modelo para criação de metodo

****Respeite as regions criadas na classe****

****Visão geral****

Gerar manipuladores para a tabela indicada através dos padrões de automação.

****Dependências****

Esta tarefa só pode ser feita após a tarefa #343

Você deve atualizar seus snippets conforme vídeo e links abaixo:

https://1drv.ms/u/s!Ag_drMqC3F6koOxtLf0AVuJK-QU6tg?e=JSH65d

https://1drv.ms/v/s!Ag_drMqC3F6koOxv3ltfi7dPR0cqPw?e=g1BW29

https://1drv.ms/v/s!Ag_drMqC3F6koOxuGnbxNjRfAX5vVA?e=DMNQbL

****Sobre as alterações****

Deverá ser criado na classe `producao.cs` ->

![[image|/uploads/71d43b08803e983a4d4b2858f16bb875/image.png]]

Para criar métodos dentro da classe veja o vídeo abaixo:

https://1drv.ms/v/s!Ag_drMqC3F6koO0IXVl1jrOullqUWQ?e=hqLW89

Neste arquivo uso o snippet `AutSnippetDBBase` e crie os seguintes métodos

- `public DelAutProgramaDAPI(AUT_PROGRAMA programa)` -> Responsável por chamar o método interno e iniciar transações. usar snippet `AutSnippetDBBase`
- `internal DelAutProgramaDAPI(SQL sql, AUT_PROGRAMA programa)` -> responsável por validar e deletar na tabela pai e filha através dos snippets (`AutSnippetDBDoDelete`).
- `internal ValidaDeleteAutProgramaDAPI(SQL sql, AUT_PROGRAMA programa)` -> responsável por validar o objeto antes da operação. Este não possui snippet.

****Crie o SQL de delete baseado no código para `AUT_Programa_prod` e (`code`, `lineId`) para `AUT_PROG_ITENS`****

****Sobre as validações****

- [] Colocar dentro do método `ValidaDeleteAutProgramaDAPI`
- [] Nas modelos de entrada é obrigatório o código na model `AUT_Programa_prod` e `lineId` na `AUT_PROG_ITENS` estarem preenchidos.
- [] Validações óbvias se existe registro
- [] Caso o valor de `AUT_PROGRAMA` seja menor que 2 permitir! se não informar que "O Programa não aceita mais alterações pois seu status já está atualizado".

Criando telas padrões do SAP

Vídeo de criação de telas

Para criação de telas com objetos padrões do SAP, pode fazer através do vídeo abaixo.

<https://lnnk.in/aqfd>

Padrões de testes e simulações

Automação

Como testar o projeto automação:

Opção 1:

[Como testar o automação · Wiki · SIX Consult / Geral / Automacao · GitLab](#)

Opção 2:

<https://lnnk.in/bBcz>

Checklist para simulação

https://1drv.ms/x/s!Ag_drMqC3F6koOxl_E3e4BV9jdgYvg?e=rCHSzt

Como fazer testes unitários de metodos em background

Para auxiliar na realização de testes dos metodos criados no automação, por favor utilize o mecanismo do vídeo abaixo.

https://1drv.ms/v/s!Ag_drMqC3F6koPth30kyMwxslN2N3w?e=wxgH7J

Como utilizar o git nos testes

Caso já tenha visto o vídeo do git não é necessário verificar este vídeo novamente! Caso seja somente um tester e precise saber o procedimento. Verifique o vídeo abaixo

[Utilização do GIT](#)

Documento de teste.

Este documento deve ser preenchido com os problemas encontrados e encaminhado no GIT conforme vide do GIT.

https://1drv.ms/w/s!Ag_drMqC3F6koOwcW6tjALJHAdvvDA?e=kuyIXu

Snippets e trechos de codigo padronizado

Snippets são trechos de códigos padrões para reutilização.

Como usar os snippets

Para utilizar os snippets, voce deverá seguir os passos abaixo

Download dos snippets

Faça o download do arquivo no endereço abaixo.

https://1drv.ms/u/s!Ag_drMqC3F6koPUuOoQiyts9QTqs_Q?e=TZ9D7I

Como importar os Snippets

Para importar verifique o vídeo abaixo.

https://1drv.ms/v/s!Ag_drMqC3F6koOxv3Itfi7dPR0cqPw?e=g1BW29

Como implementar os snippets

Para utilizar os snippets, verifique o vídeo abaixo

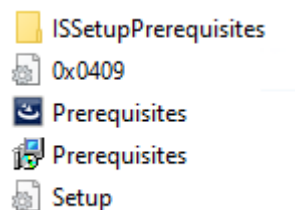
https://1drv.ms/v/s!Ag_drMqC3F6koOxuGnbxNjRfAX5vVA?e=DMNQbL

Instalação SAP DI API Versão 10

A versão do SAP 10, trouxe novas funcionalidades e pra isso precisamos instalar o Client e a DI API e SDK. A partir da versão 10.0 o SAP trabalha apenas com DI API x64.

PreRequisites

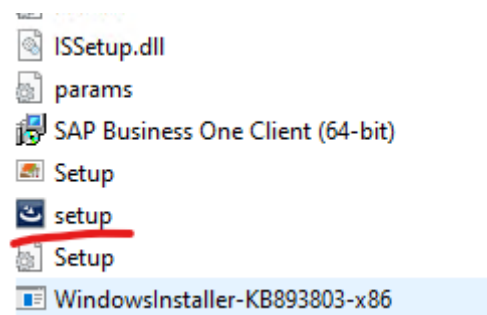
Na Pasta Prerequisites da Versão SAP 10 executar o Instalador Prerequisites.exe.



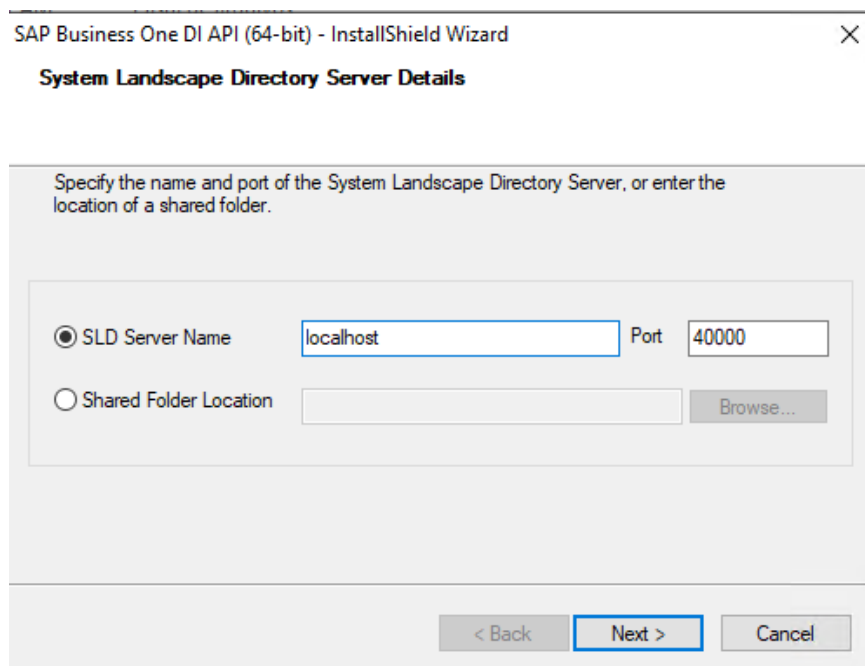
DI API

Para a Instalação da DI API basta apenas executar a instalação do Client DI API encontrado na pasta Packages.x64/Client/setup.exe.

Obs: o Client já instala a DI API x64. Não é necessário instalar posteriormente.



Na tela de configuração do SLD (Servidor de Licença) apenas preencher com localhost.



SAP Business One DI API (64-bit) - InstallShield Wizard

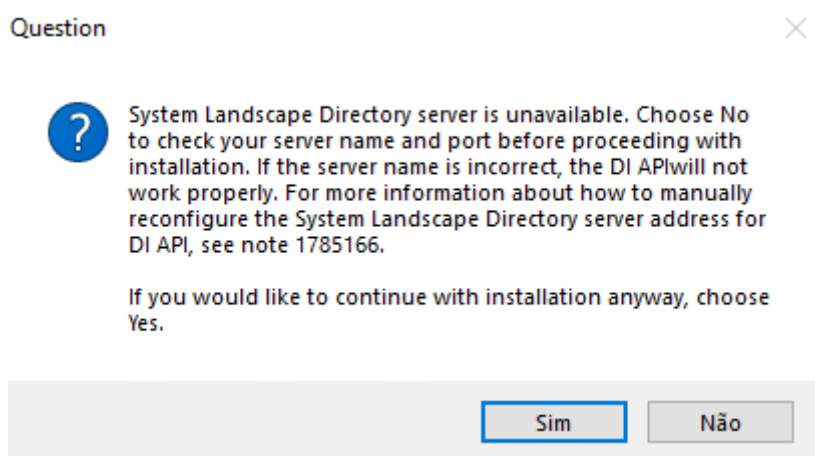
System Landscape Directory Server Details

Specify the name and port of the System Landscape Directory Server, or enter the location of a shared folder.

☒ SLD Server Name Port

☐ Shared Folder Location

Em seguida confirmar a mensagem, apertando em Sim.



Question

?

System Landscape Directory server is unavailable. Choose No to check your server name and port before proceeding with installation. If the server name is incorrect, the DI API will not work properly. For more information about how to manually reconfigure the System Landscape Directory server address for DI API, see note 1785166.

If you would like to continue with installation anyway, choose Yes.

Instalação SDK

Para Desenvolvedores é necessário instalar o SDK para as aplicações desenvolvidas.

O instalador do SDK encontra-se na pasta Packages.x64\SDK apenas execute o instalador setup.exe