

CES-65 - Projetos de Sistemas Embarcados

CE-235 - Sistemas Embarcados de Tempo Real

CE-230 - Qualidade, Confiabilidade e Segurança (Safety) de Software

CE-237 - Tópicos Avançados de Teste de Software

Prof. Dr. Adilson Marques da **Cunha**

Prof. Dr. Luiz Alberto **Vieira Dias**



OBJETIVOS

- Mostrar de maneira clara, pragmática e didática o desenvolvimento de um projeto, utilizando-se do software SCADE (*Safety-Critical Application Development Environment*) da Esterel Technologies.
- Servir de base e fonte de apoio para a capacitação e o nivelamento das Turmas de Graduação e de Pós-Graduação do ITA, envolvendo Projetos de Sistemas Computadorizados que utilizam o SCADE.
- Descrever os procedimentos para a construção de um Especificação de um Painel de Gerenciamento de Combustível (*Fuel Management Panel Specification*) de uma aeronave fictícia.

CONTEXTO

Com intuito de prover auxílio técnico ao Ambiente Integrado de Ferramentas de Engenharia de Software Ajudada por Computador do SCADE, para apoiar os alunos integrantes de Disciplinas CES-65, CE-230, CE-235 e CE-237 do ITA, o presente trabalho tem por objetivo servir de um guia ou tutorial dando as condições mínimas, necessárias e suficientes para o

entendimento e a construção do Painel de Gerenciamento de Combustível de uma Aeronave fictícia utilizando o Ambiente SCADE.

Os procedimentos mostrados neste documento não têm a intenção de esgotar o assunto. Eles foram elaborados visando apenas aumentar a clareza e o pragmatismo do projeto, propiciando sua implementação satisfatória.

Voltado para um estudo orientado a um projeto, este trabalho busca selecionar o que há de melhor nos principais guias de estudo dos semestres anteriores, juntamente com diversas discussões em fóruns a respeito do assunto. Portanto, ele representa um produto não somente do seu autor, como também um esforço coletivo dos alunos das turmas dos anos anteriores.

O projeto cria um display de gerenciamento de combustível. O projeto permite a escolha do tipo de combustível que será utilizado pela aeronave, verificar o nível de combustível, possibilita ainda recarga do mesmo e de forma segura, pois este possui um sistema de abertura/travação da tampa do tanque.

Apesar de não ser obrigatório, pois este material buscou ser o mais didático possível, é recomendável que o leitor tenha um conhecimento mínimo de algoritmos e de programação.

Espera-se que, com este guia inicial, a jornada de estudo e produção de cada aluno seja facilitada e otimizada em futuras e novas implementações neste e/ou nos próximos semestres no ITA.

***Nível de Estudo: Pré-intermediário**

“Uma imagem vale mais que mil palavras”

(Confúcio)

Dicionário de Termos

Alguns termos utilizados neste projeto, são peculiares ao uso do Scade ou a área em questão, portanto, é listado abaixo alguns termos que normalmente não aparecem na área comum de programação. São eles:

👉 **Cockpit Display System(CDS)** → Oferece a parte visível (e/ou audível) de um sistema cuja função seja de Interação Humano-Computador(IHC), ou seja, é basicamente a tela que apresenta informações para o utilizador do sistema. Ele pode ser um componente de hardware físico, ou uma emulação desta tela.

CDS's de um “Legacy 500”, aeronave executiva produzida pela EMBRAER



Fonte: sitio da international-pilot

👉 **Scade UAPage Creator/Display** → Módulo do Scade responsável pelo desenvolvimento visual do projeto. Extensão padrão: “.sgfx”

👉 **Scade Suite** → Módulo do Scade responsável pelo desenvolvimento da lógica do projeto. Extensão padrão: “.vsw”

👉 **Definition File(DF)** → Instancia de arquivo que irá definir a parte visual do projeto, ou seja o layout, disposições, tamanhos, cores dentre outros atributos de botões, textos, etc. A GUI(Gaphical User Interface) é definida em binário. O DF é reconhecido pela extensão “.sgfx”, pode ser criada e editada no módulo do Scade Page Creator.

👉 **Widgets** → É um elemento(controle) de interação do usuário com o computador. Este elemento normalmente é disposto dentro de uma interface gráfica de usuário(GUI), com isso o usuário pode interagir de forma direta com o controle/elemento para ler, inserir ou editar informações de um aplicativo. As bibliotecas de Widgets vem acompanhadas de botões, menus, containers,

EditBoxes, listas, radioboxes, barra de progresso e de rolagem, entre inúmeros outros.

👉 **User Application (UA)** → Aplicativos do usuário/ Lógica do sistema.

👉 **WorkSpace** → Instancia de arquivo que irá definir dentro da área no Scade de desenvolvimento de programação, onde é inserida toda a lógica de desenvolvimento e visualização dos objetos, atributos e métodos, além das ferramentas para esse desenvolvimento. Reconhecido pela extensão “.vsw”, pode ser editável no módulo do Scade Suite.

👉 **ARINC 661** → É uma norma de padronização que visa a normalização, comunicação e definição dos elementos de um ou mais CDS's. Em outras palavras, busca fazer a interação e comunicação dos vários CDS's com a UA's do sistema. Dentre diversas aeronaves que utilizam este padrão podemos citar Airbus A380 e Boeing 787. No nosso caso também pode ser entendido como um protocolo de comunicação entre UA e CDS.

Sumário

0	Instalação do Scade	5
1	Desenvolvendo o visual	6
2	Desenvolvendo a lógica	13
3	Reportando a documentação	39
Referências		43

0 Instalação do Scade

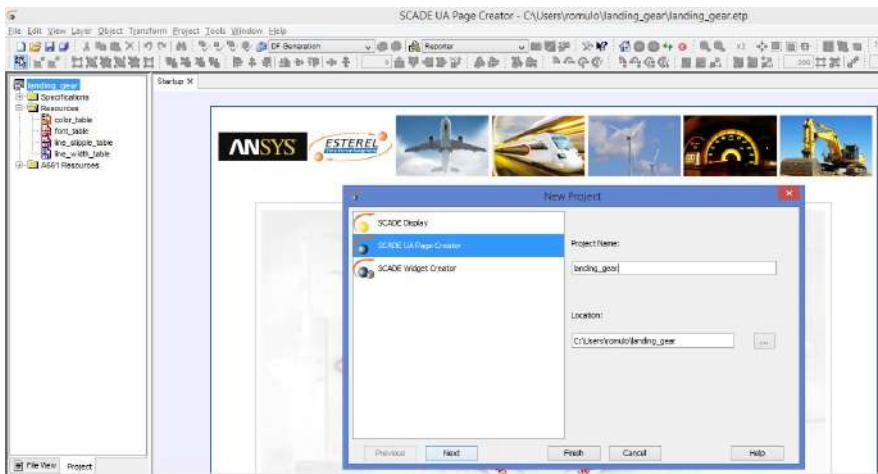
- 👉 Recomenda-se a plataforma Windows 10 para a instalação do Scade.
- 👉 Os procedimentos de instalação podem ser encontrados no portal do Projeto de STAGIHO-TR, ou [clique aqui](#) para ser redirecionado diretamente para a secção referida.

1 Primeiro abra o Scade UA Page Creator:

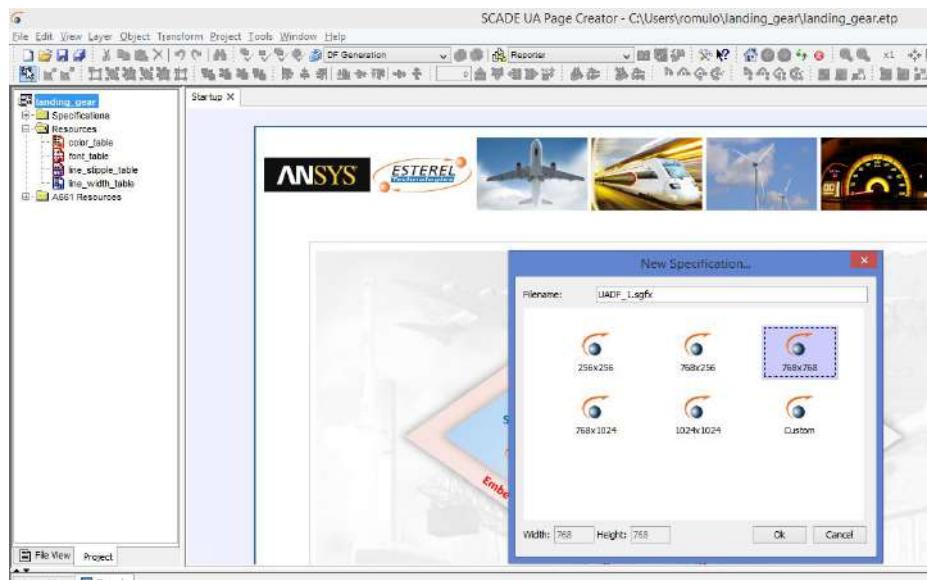


1.1 Crie um projeto clicando em File > New > Project.

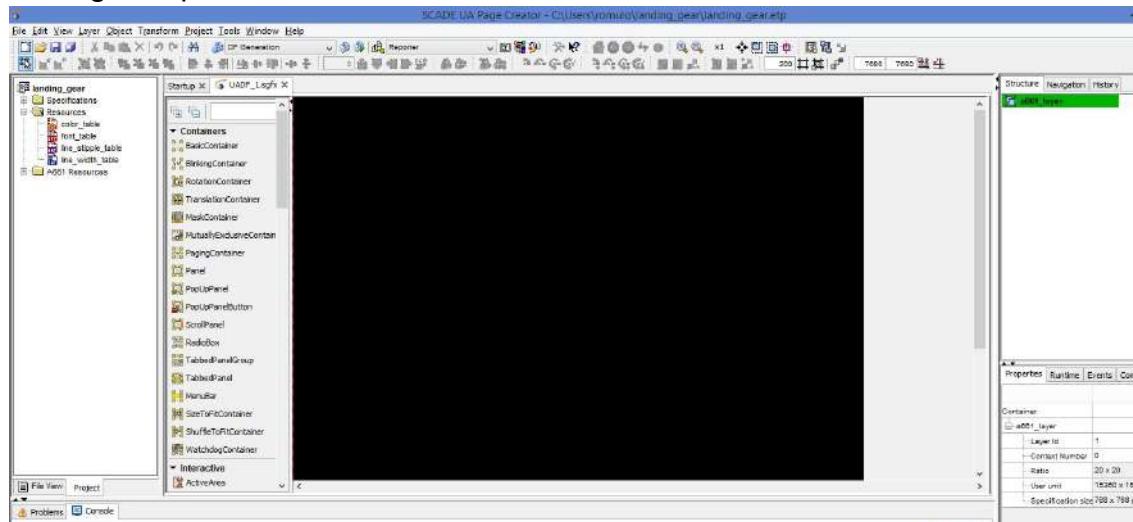
1.2 Selecione a opção SCADE UA Page Creator, e dê o nome a seu projeto.



1.3 Agora crie um Definition File, clicando em File>New>"Specification" e escolhendo a opção 768x768 DF dentro do menu Specification e pressione ok:

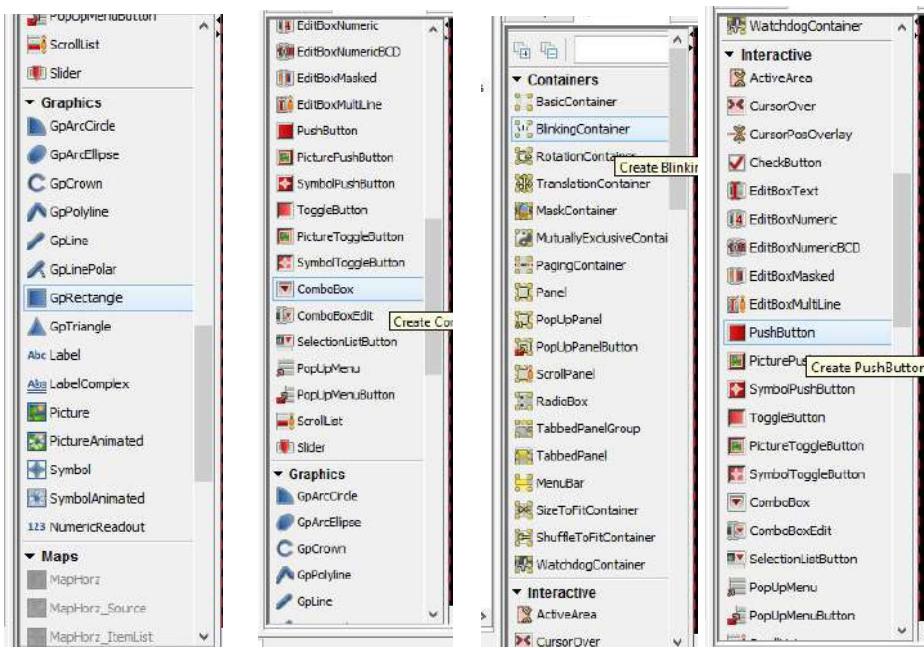


1.4 Agora aparecerá esta tela:

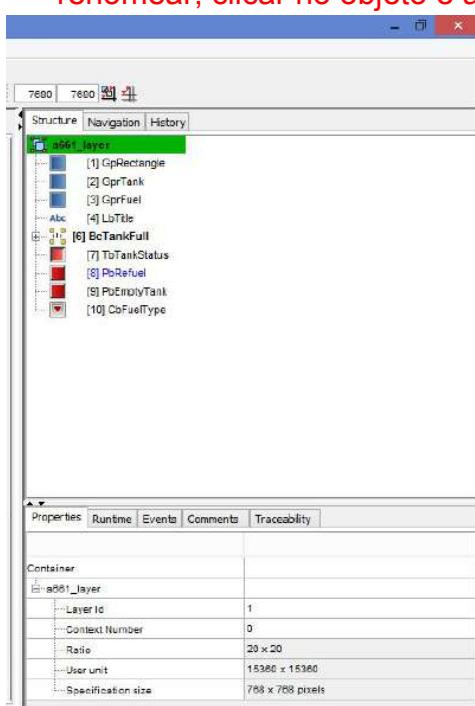


1.5 Para adicionar elementos ao display (área preta da figura anterior) apenas clique no desejado (objetos representados por ícones nos menus) e depois clique no display.

Agora adicione 3 GpRectangle, 2 Label, 1 ToggleButton, 1 ComboBox, 1 BlinkingContainer e 2 PushButtons. (Obs: o BlinkingContainer, não aparecerá na tela, basta clicar nele para cria-lo.)



1.6 Agora, para editar a forma e a posição dos widgets de forma mais precisa, utilizaremos o Properties Box OBS : ARRASTE Label com id 5 PARA DENTRO DO BLINKING CONTAINER como na imagem abaixo
OBS2: Atentar-se para os IDs estarem de acordo com a imagem. Para renomear, clicar no objeto e apertar “F2”.



1.7 Os widgets devem ser nomeados e posicionados de acordo com a tabela abaixo:

Nome	ID	PosX	PosY	SizeX	SizeY
GpRectangle	1	200	14260	14960	800

GprTank	2	9500	2150	5000	10000
GprFuel	3	9550	2200	4900	1000
LbTitle	4	5740	14250	5000	800
LbTankFull	5	870	9400	3000	800
BcTankFull	6	-----	-----	-----	-----
TbTankStatus	7	840	12500	3000	800
PbRefuel	8	10550	12500	3000	800
PbEmptyTank	9	10550	1000	3000	800
CbFuelType	10	840	11020	3000	800

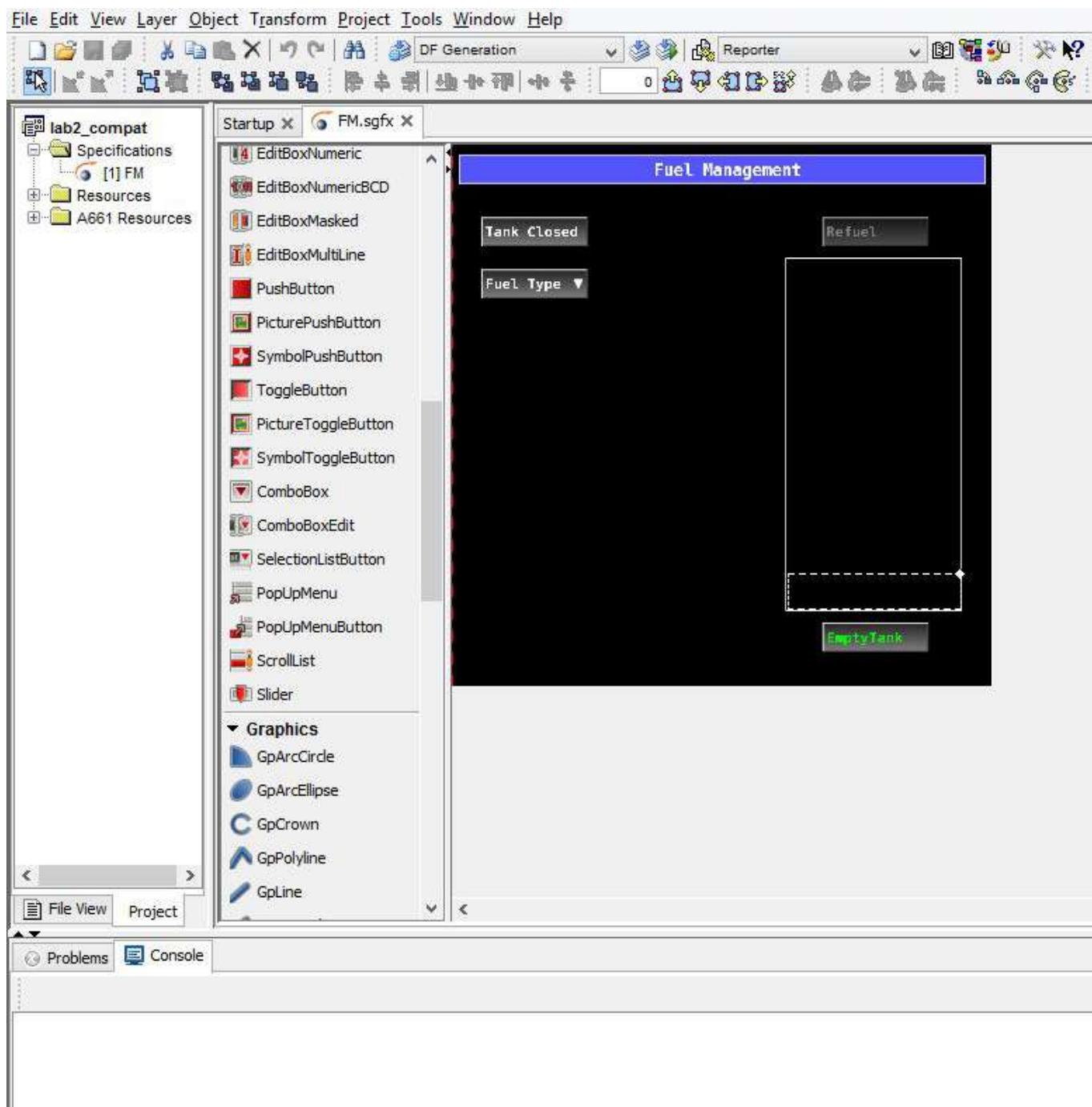
1.8 Agora, mude o texto do Label LbTitle para “Fuel Management” e deixe o tamanho F3, mude o texto do Label LbTankFull para “Tank Full!” e mude o Color index para 21 e mude o tamanho para F3, agora no BcTankfull mude o

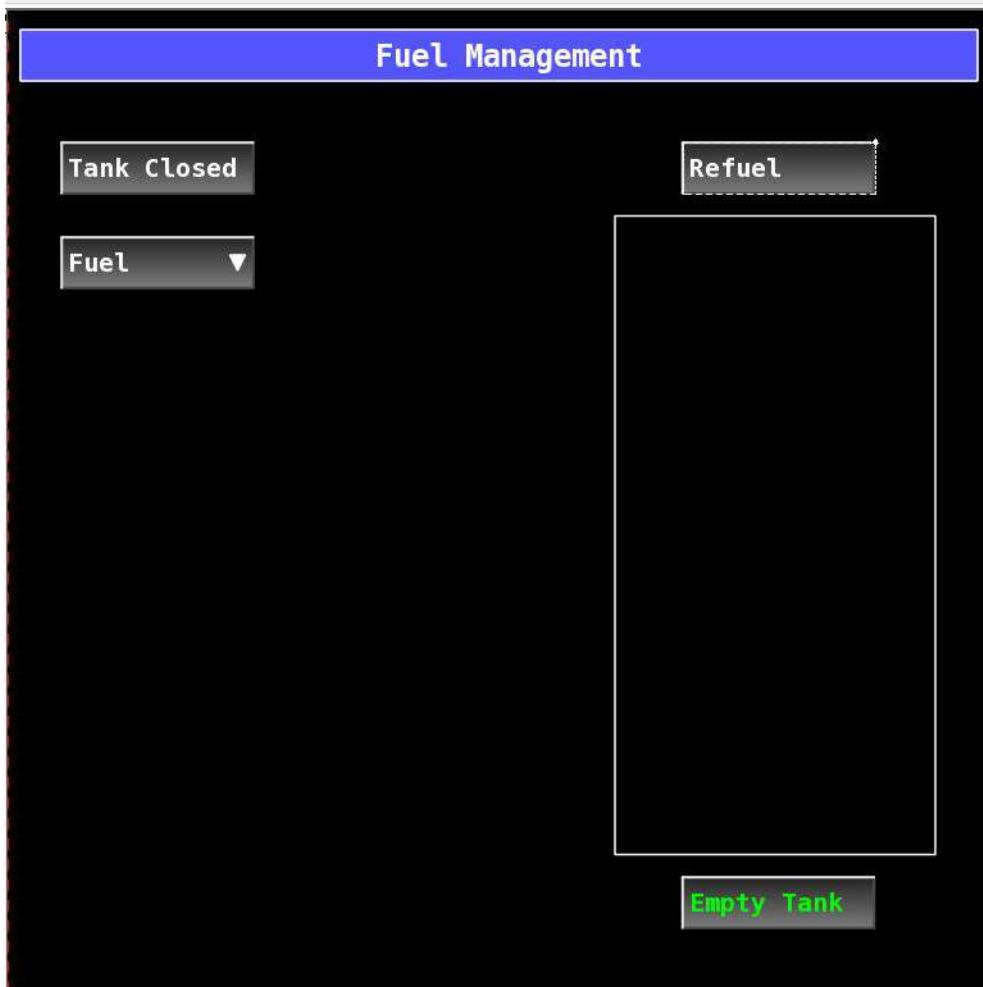
BlinkingType para 2 e o visible para false, Agora no GprTank, mude o Colorindex para 1 e o Fillindex para 0,no GprFuel, mude o Visible para “A661_FALSE” e o colorindex e fillindex para 74. Agora no PbRefuel mude o enable para

“A661_TRUE” e mude o texto para “Refuel”, no PbEmptyTank mude o StyleSet para “112 : RCT_F2_GN” e mude o texto para “Empty Tank”,Agora no TbTankStatus, mude o AlternateFlag para “A661_TRUE” e mude o texto para “Tank Closed”, e mude o AlternateLabelString para “Tank Open”. No

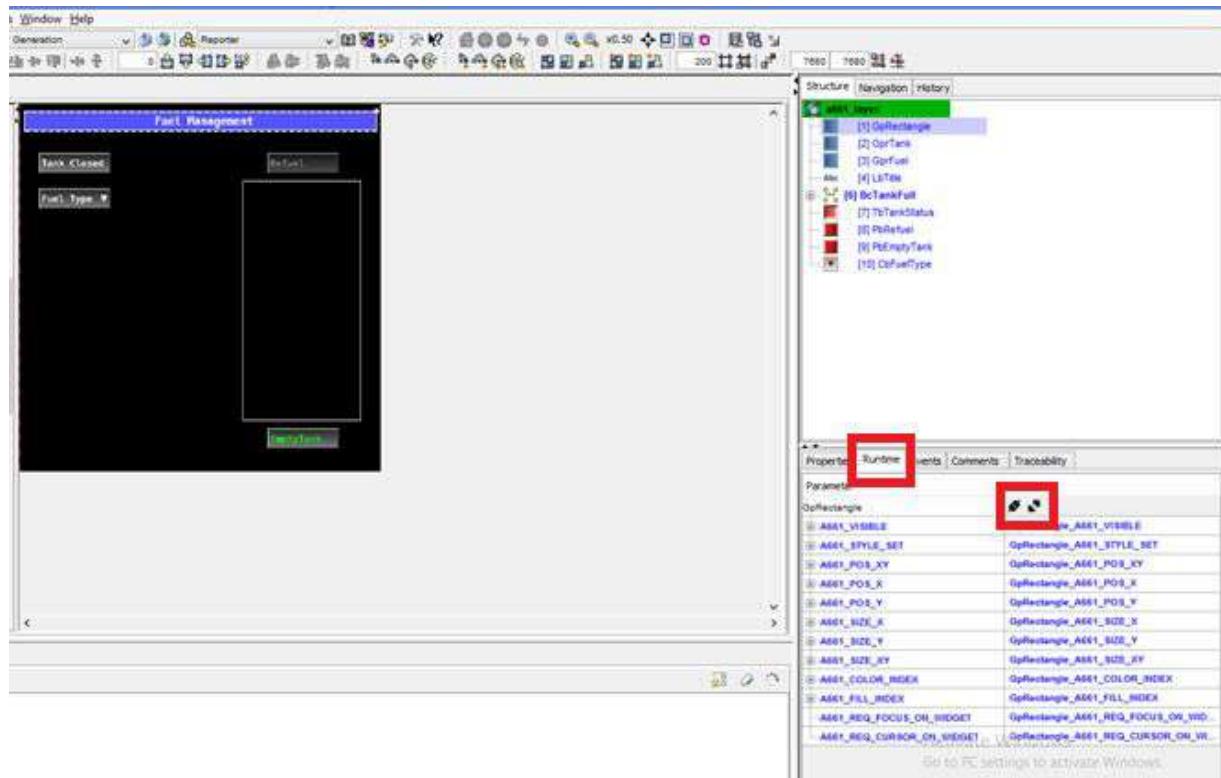
CbFuelType, mude o entry 1 para Fuel Type, o 2 para Alcohol, o 3 para Diesel e o 4 para Gasoline.

1.9 No final, o display deverá ficar desta maneira:





1.10 Ative as mensagens do barramento ARINC 661 para cada componente criado através do menu Runtime, clicando no botão “**Connect All**”, conforme abaixo (ATENÇÃO, não esqueça de conectar tambem o objeto a661_layer):

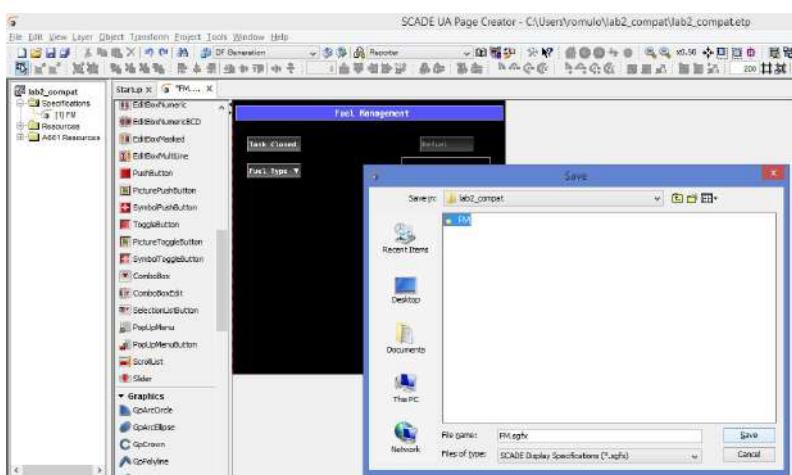
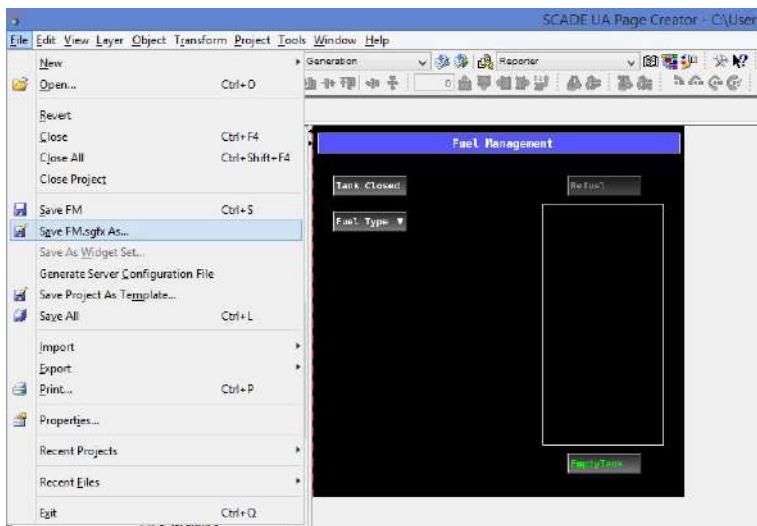


OBS: Lembre-se de repetir o processo para todos os componentes listados na sessão “Structure”.

Obs2: No SCADE R19 a opção para fazer a conexão do objeto **a661_layer** pode não aparecer, para aparecer salve o projeto, feche-o e abra novamente, assim a opção para conectá-lo estará disponível.

1.11 Configure os eventos dos componentes TbTankStatus, PbRefuel, PbEmptyTank e CbFueltype (Assim como no passo anterior, deve-se clicar na aba “events” e depois “connect all” para esses objetos).

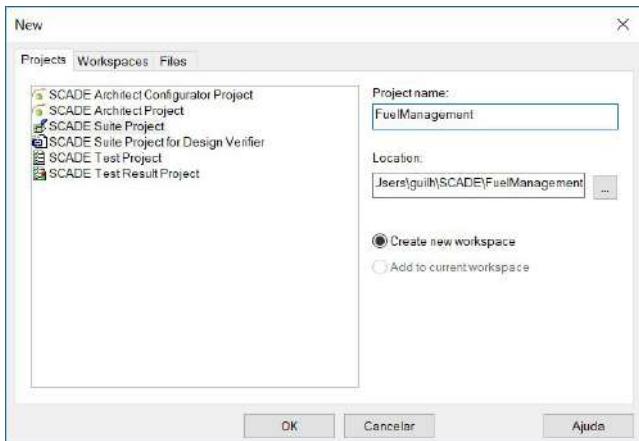
1.12 Agora salve o seu Definition File como “FM”(observe como ficará a extensão do arquivo .sgfx) dentro de uma pasta que você possa achar com facilidade.



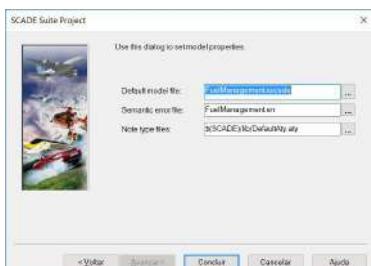
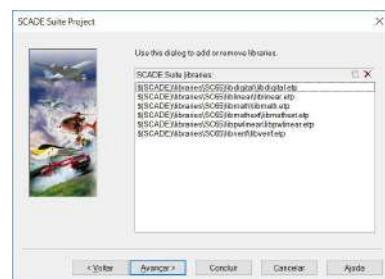
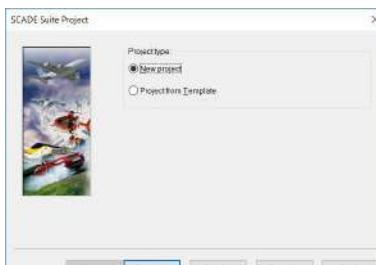
2 Agora, abra o SCADE Suite R19.



2.1 Crie um projeto chamado FuelManagement e salve ele na mesma pasta que o Definition File por conveniência.



Passo 1

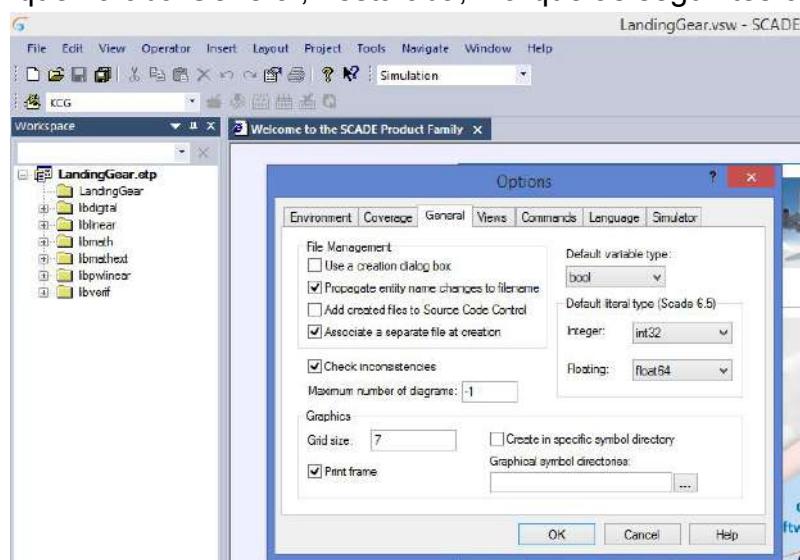


Passo 2

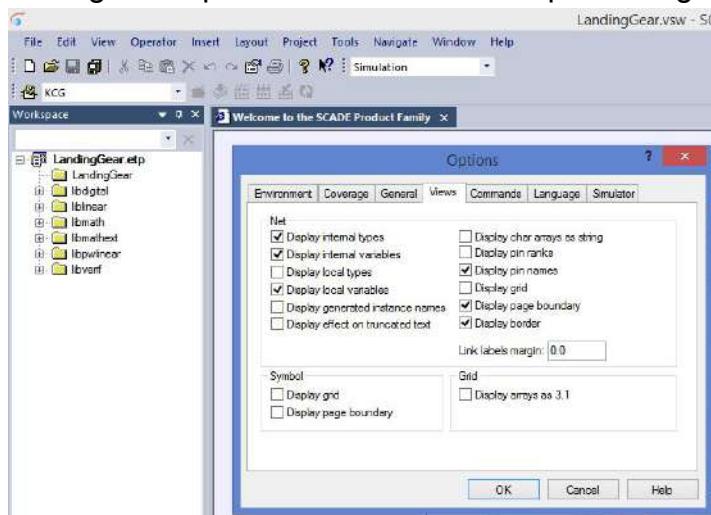
Passo 3

Passo 4

2.2 Agora, configure o SCADE Suite desta maneira: Vá no menu Tools> Options e clique na aba General, nesta aba, marque as seguintes opções:

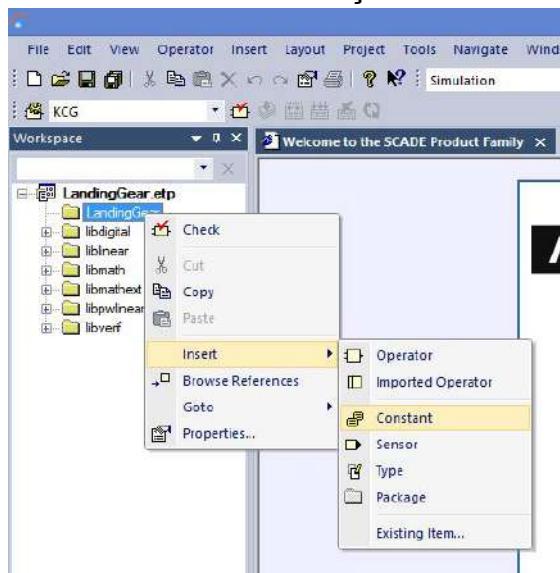


2.3 Agora clique na aba Views e marque as seguintes opções:

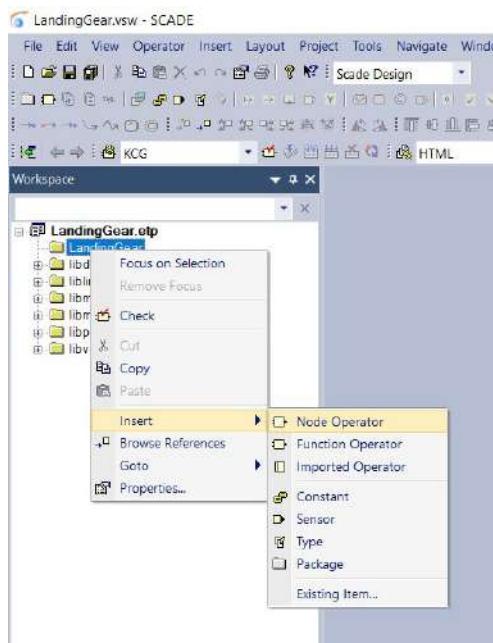


2.4 Segue uma introdução sobre as funcionalidades básicas do Scade Suite.

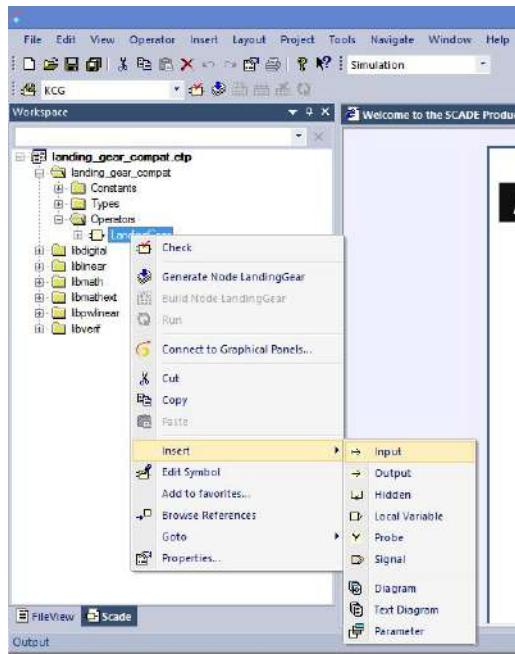
Constantes: São valores de certo tipo de dado que podem ser atribuídos a outros elementos. A criação de uma Constante no Scade é feita desta maneira:



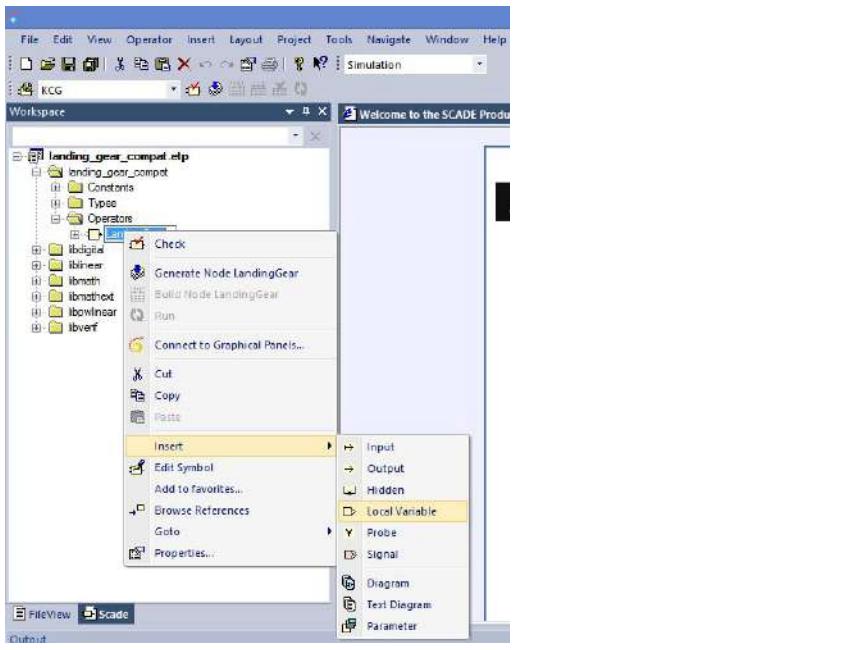
Operadores: são utilizados para construir e executar a lógica de seu programa. A criação de um operador no Scade é feita desta maneira:



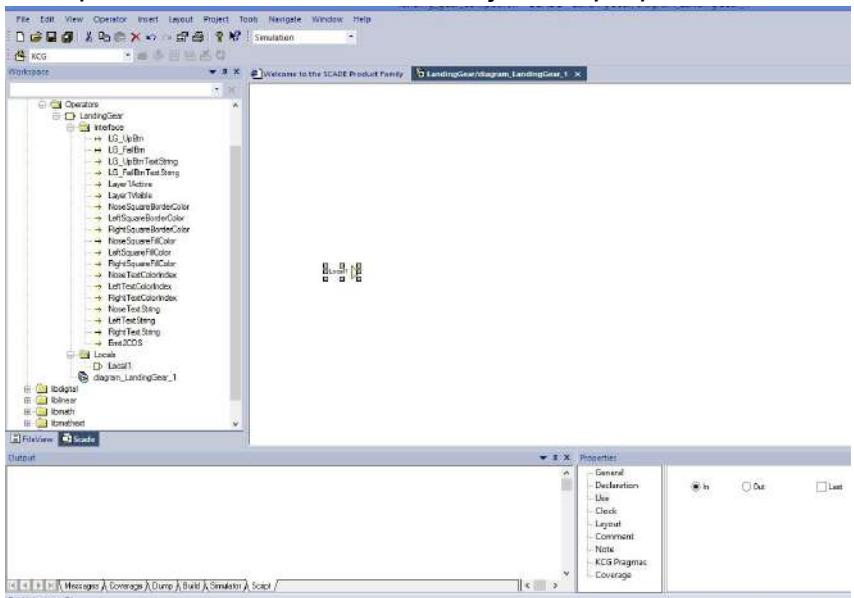
Interfaces Inputs ou Outputs: Podem ser utilizadas para definir o valores de uma variável ao receber/enviar o valor de uma ação. A criação de uma interface é feita desta maneira no Scade:



Variáveis locais: São utilizadas para armazenar valores.
A criação de uma variável local é feita desta maneira no Scade:

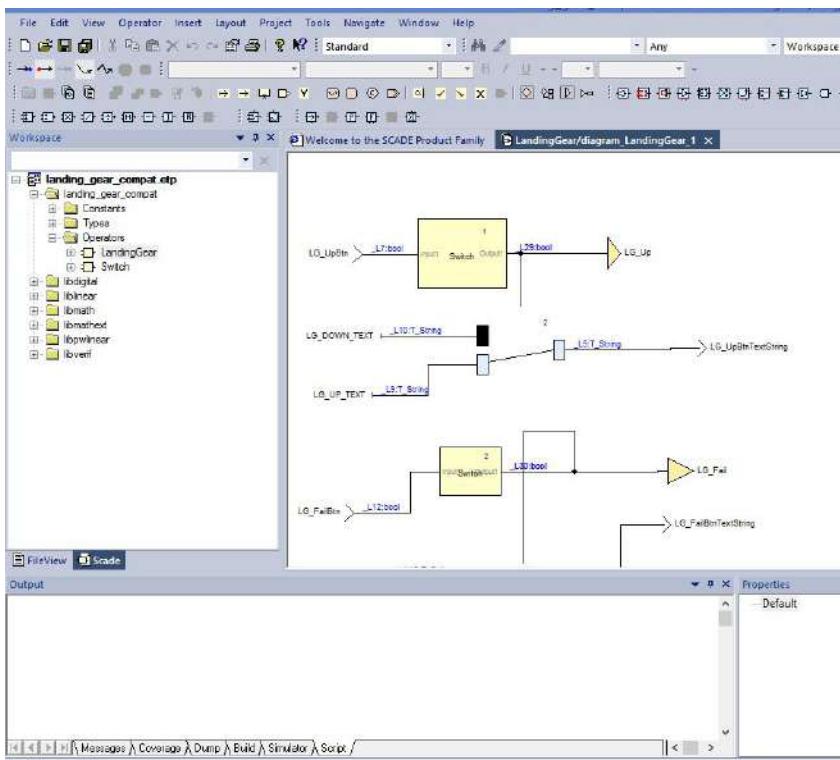


OBS: uma variável pode ser usada tanto como input quanto output, você pode escolher o uso dela na janela “*properties*”.



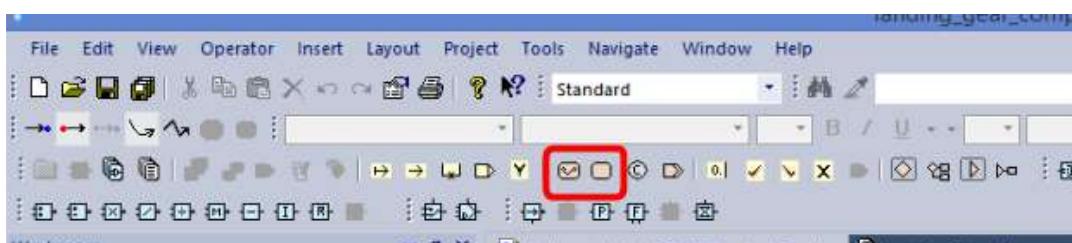
Workspace: Clicando duas vezes em um Operador, surgirá uma tela branca que é utilizada para montar a lógica do programa como um “círcuito”(quem já projetou algo em eletrônica vai se sentir familiarizado o.O) arrastando vários elementos como variáveis, constantes, Máquinas de estado, Inputs, outputs e até outros operadores.

Exemplo de workspace:

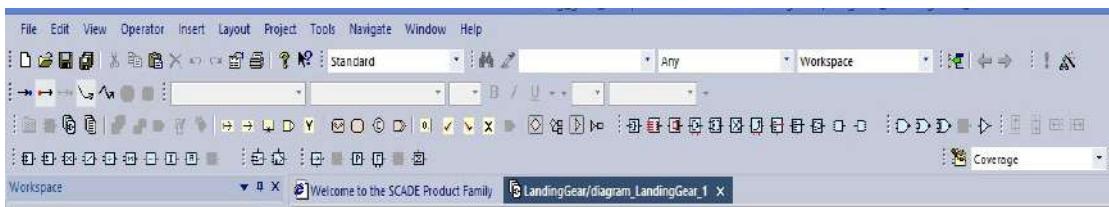


Máquinas de estado e estados: Uma máquina de estado é utilizada para fazer a transição de uma ação para outra utilizando estados que pulam de ativos para desativados dependendo da ação, um estado possui um workspace próprio que é utilizado para inserir lógica assim como no Operador.

A criação de uma Máquina de estado e de um estado é feita clicando nestes ícones:



Operadores lógicos: Os operadores lógicos como IF, FOLLOWEDBY, NOT etc. Estão incluídos no SCADE e podem ser utilizados no workspace, clicando conforme cada um deles, nesta área (As barras disponíveis podem ser configuradas com o clique do botão direito na área abaixo):



Voltando ao projeto...

x2.5 Crie as seguintes constantes (clique duas vezes na pasta constants para ficar mais facil):

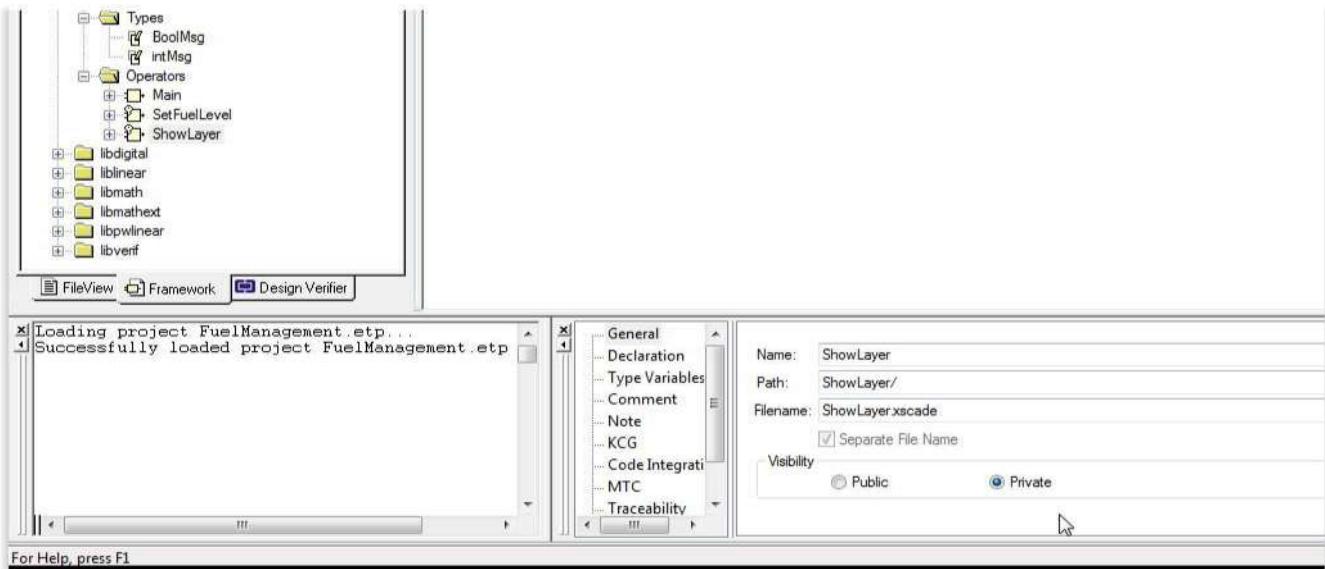
Constant	Type	Value
A661_FALSE	bool	false
A661_TRUE	bool	true
ALCOHOL_CO	uint8	64
CLOSED_ID	uint32	0
DIESEL_COLOR	uint8	47
FUEL_INCREM	uint32	1000
FUEL_LEVEL_CO	uint32	100
GASOLINE_CO	uint8	57
MAX_FUEL_LE	uint32	10000
OPEN_ID	uint32	1

2.6 Agora crie os seguintes tipos e coloque <structure> na definition (ver imagem do próximo passo. Novamente, para ficar mais facil clique duas vezes na pasta types apos criar os tipos):

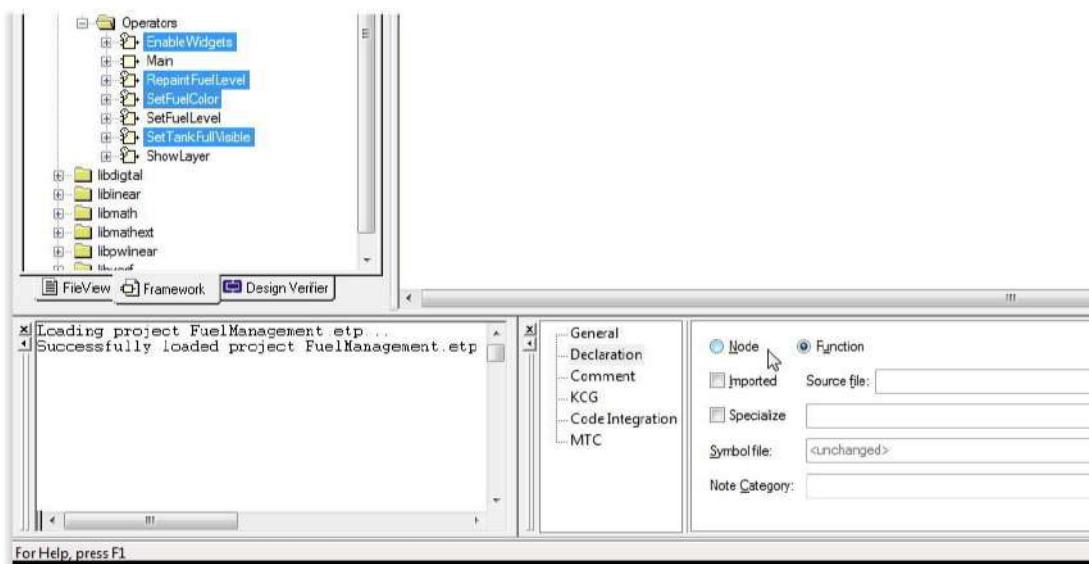
2.7 Mude os valores do tipo para que fiquem iguais aos da figura abaixo:

Type	Definition	Comments
BoolMsg	<structure>	
Emit	bool	
Value	bool	
intMsg	<structure>	
Emit	bool	
Value	uint32	
intMsg2	<structure>	
Emit	bool	
Value	uint8	

2.8 Agora crie 3 operadores chamado Main, SetFuelLevel e ShowLayer, agora, vá nas propriedades dos operadores SetFuelLevel e ShowLayer e, clique no menu general e mude a visibility de Public para Private, apenas o main deverá ficar public:



2.9 Agora, crie mais 4 operadores chamados: EnableWidgets, RepaintFuelLevel, SetFuelColor, SetTankFullVisible agora, vá na propriedade destes operadores e deixe eles private, depois vá no declaration e mude a opção node para Function (somente desses ultimos), novamente o main deverá ser o único público:



2.10 Crie as seguintes interfaces no operador main:

Nome	Type	Kind
TankStatus	bool	Input
FuelType	uint16	Input
Refuel	bool	Input
EmptyTank	bool	Input
LayerVisible	bool	Output
FuelVisible	BoolMsg	Output
FuelLevel	intMsg	Output
RefuelEnable	BoolMsg	Output
FuelTypeEnable	BoolMsg	Output
TankFullVisible	BoolMsg	Output
FuelColor	intMsg2	Output

OBS :Ainda no operador main crie uma variável local do tipo int32 chamada tank_status.

2.11 Crie as seguintes interfaces no operador EnableWidgets:

Nome	Type	Kind
FuelType	uint16	Input
FuelStatus	uint32	Input
FuelLevelValue	uint32	Input
RefuelEnable	BoolMsg	Output
FuelTypeEnable	BoolMsg	Output

2.12 Crie as seguintes interfaces no operador RepaintFuelLevel:

Nome	Type	Kind
Value	uint32	Input
FuelVisible	BoolMsg	Output
FuelLevel	intMsg	Output

2.13 Crie as seguintes interfaces no operador SetFuelColor:

Nome	Type	Kind
FuelType	uint16	Input
FuelColor	intMsg2	Output

2.14 Crie as seguintes interfaces no operador SetFuelLevel:

Nome	Type	Kind
Refuel	bool	Input
Empty	bool	Input
FuelLevelValue	uint32	Output

OBS: Ainda no operador SetFuelLevel crie uma variável local do tipo uint32 chamada value

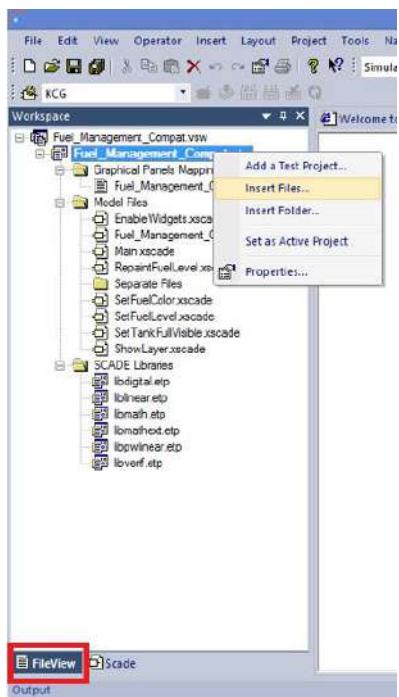
2.15 Crie as seguintes interfaces no operador SetTankFullVisible:

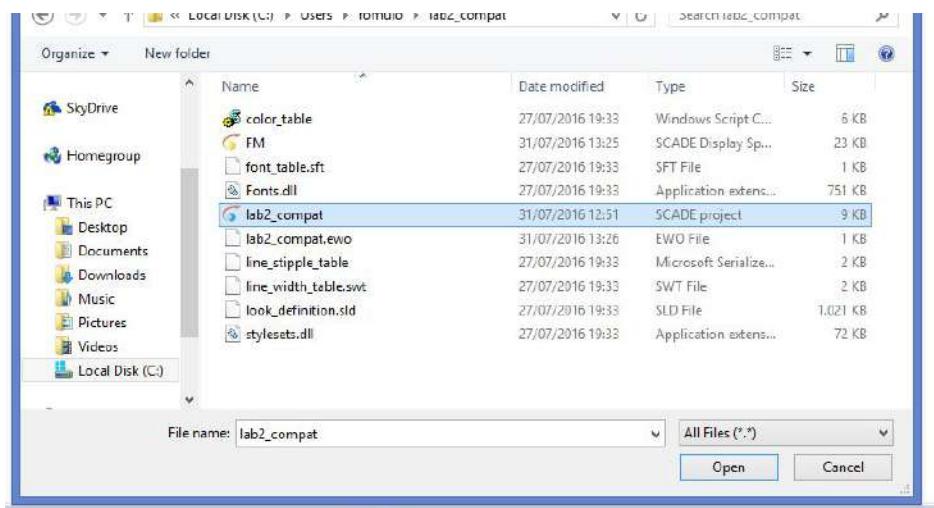
Nome	Type	Kind
FuelLevelValue	uint32	Input
TankFullVisible	BoolMsg	Output

2.16 Crie a seguinte interface no operador ShowLayer:

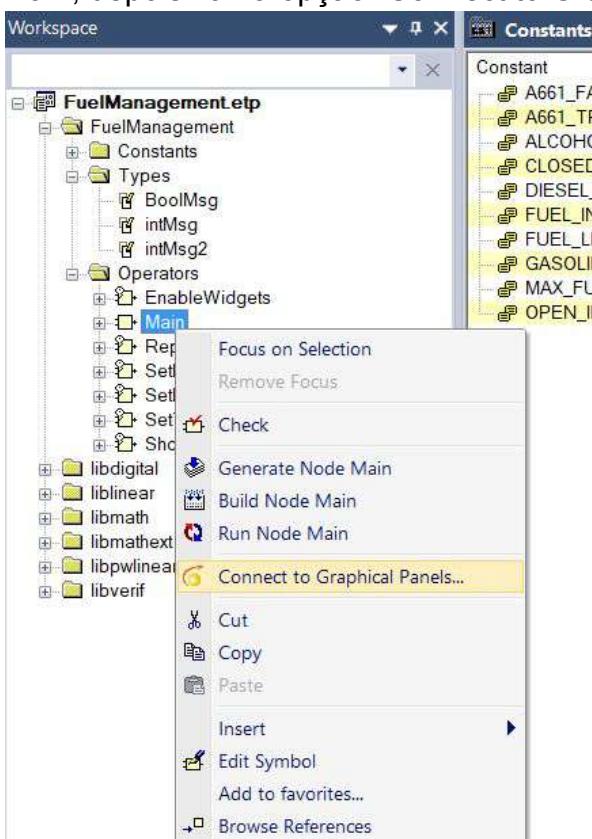
Nome	Type	Kind
LayerVisible	bool	Output

2.17 Agora importe o Definition File (que foi criado anteriormente) clicando na aba “File view” e com o botão direito no arquivo “<nome_do_projeto>.etp” e indo em “insert files” escolha o arquivo Definition file que deve ter a extensão “*.etp” (como já salientado anteriormente sobre o tipo de extensão do DF que é o “*.sgfx”):

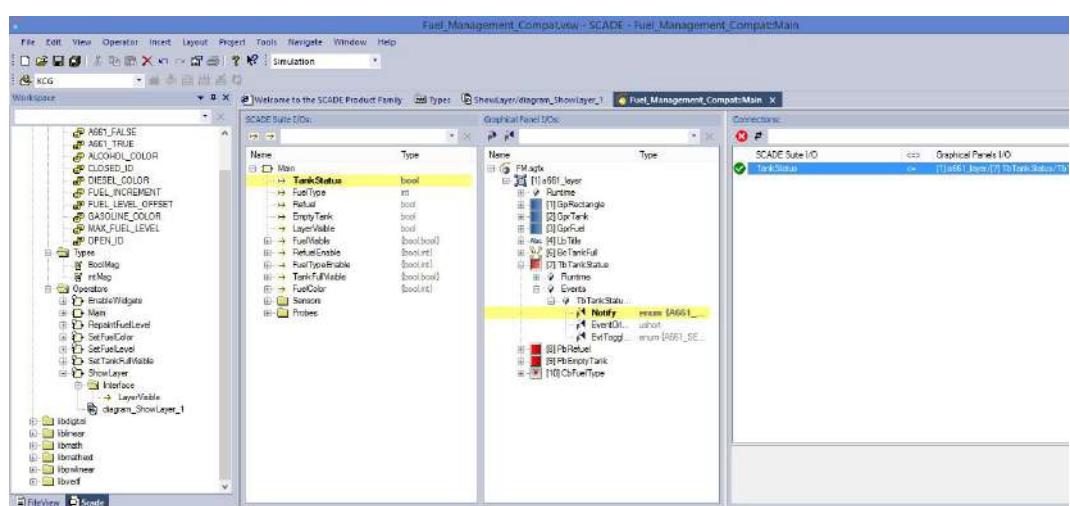
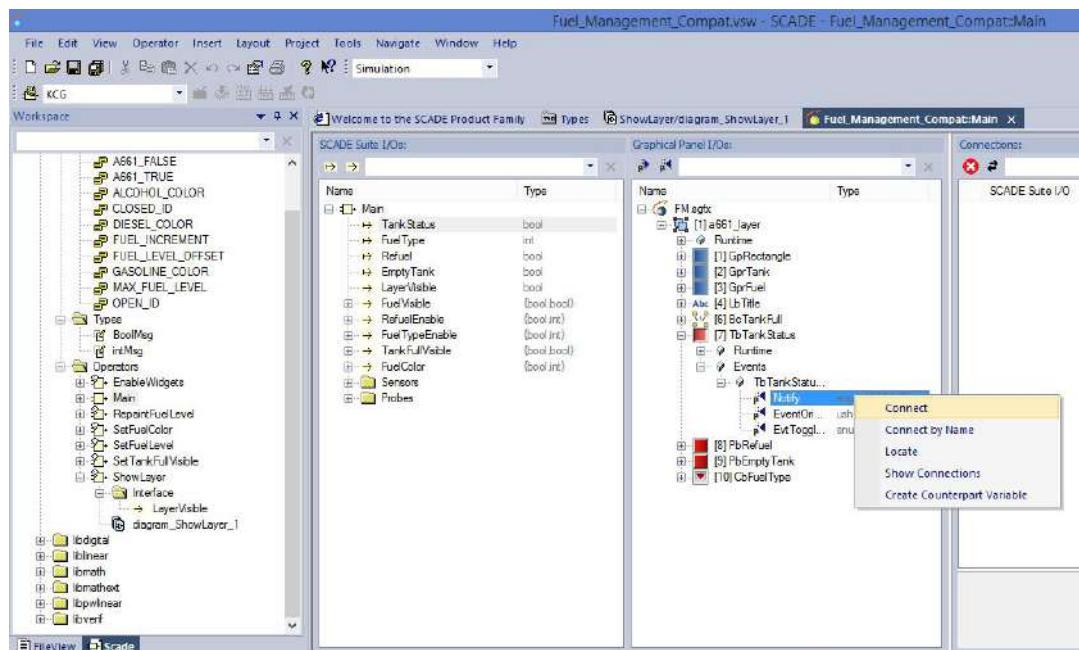
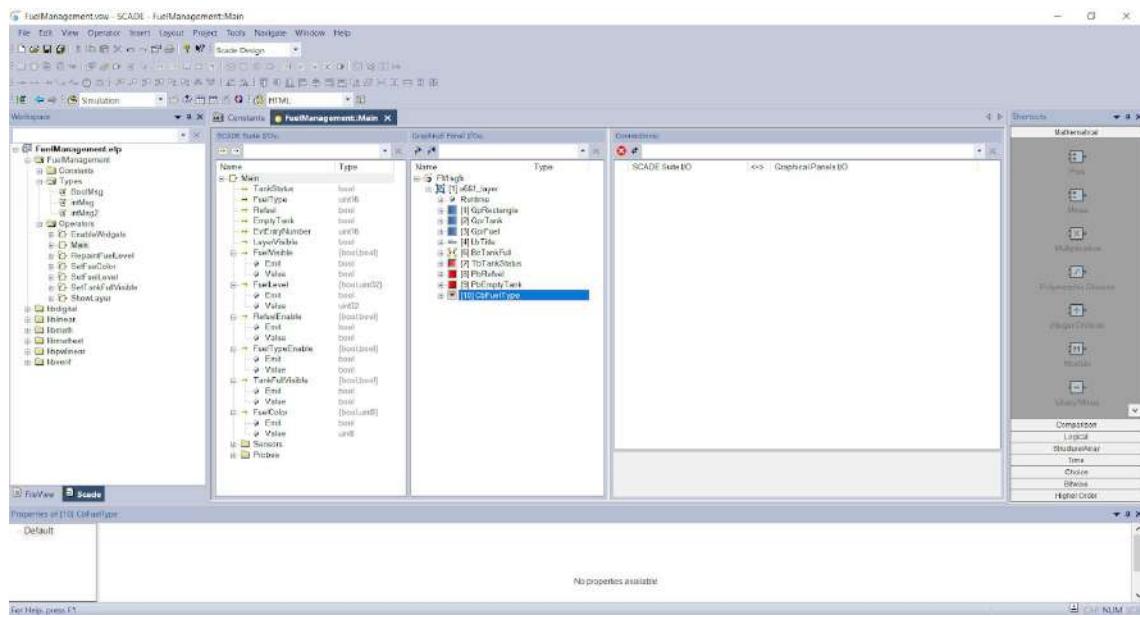




2.18 Agora, será feita a conexão do operador Main com o ambiente gráfico do Definition file, clique na aba framework clique com o botão direito no operador Main, depois vá na opção “Connect to Graphical Panels”:

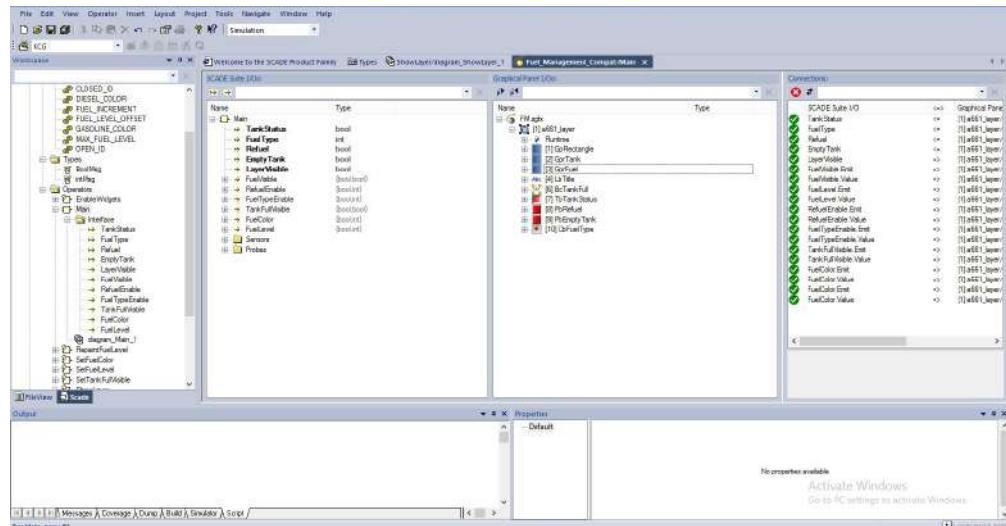


2.19 Nesta tela que foi aberta, será feita a associação entre os inputs e outputs com as interface gráfica. Para fazer uma conexão, clique em uma I/O e clique em um dos valores dos widgets. EX:

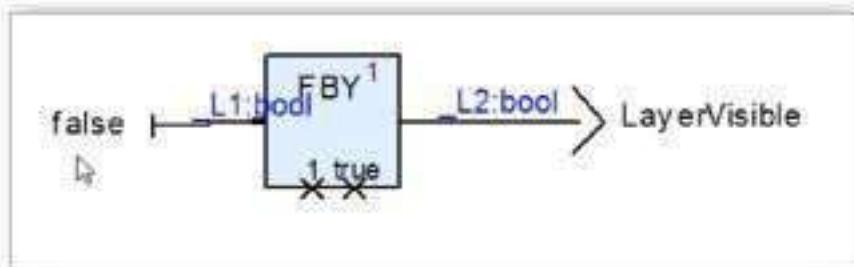


2.20 Agora, crie as conexões especificadas na figura abaixo:

SCADE Suite I/O	<=>	Graphical Panels I/O
TankStatus	<=	[1] a661_layer/[7] TbTankStatus/TbTankStatus_A661_EVT_STATE_CHANGE/Notify
FuelType	<=	[1] a661_layer/[10] CbFuelType/CbFuelType_A661_EVT_SEL_ENTRY_CHANGE/EvtEntryNumber
Refuel	<=	[1] a661_layer/[8] PbRefuel/PbRefuel_A661_EVT_SELECTION/Notify
EmptyTank	<=	[1] a661_layer/[9] PbEmptyTank/PbEmptyTank_A661_EVT_SELECTION/Notify
LayerVisible	=>	[1] a661_layer/a661_layer_A661_REQ_LAYER_VISIBLE/Emit
FuelVisible.Emit	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_VISIBLE/Emit
FuelVisible.Value	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_VISIBLE/Visible
FuelLevel.Emit	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_SIZE_Y/Emit
FuelLevel.Value	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_SIZE_Y/SizeY
TankFullVisible.Emit	=>	[1] a661_layer/[6] BcTankFull/BcTankFull_A661_VISIBLE/Emit
TankFullVisible.Value	=>	[1] a661_layer/[6] BcTankFull/BcTankFull_A661_VISIBLE/Visible
FuelColor.Emit	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_FILL_INDEX/Emit
FuelColor.Emit	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_COLOR_INDEX/Emit
FuelColor.Value	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_FILL_INDEX/FillIndex
FuelColor.Value	=>	[1] a661_layer/[3] GprFuel/GprFuel_A661_COLOR_INDEX/ColorIndex
RefuelEnable.Value	=>	[1] a661_layer/[8] PbRefuel/PbRefuel_A661_VISIBLE/Visible
RefuelEnable.Emit	=>	[1] a661_layer/[8] PbRefuel/PbRefuel_A661_VISIBLE/Emit
FuelTypeEnable.Value	=>	[1] a661_layer/[10] CbFuelType/CbFuelType_A661_VISIBLE/Visible
FuelTypeEnable.Emit	=>	[1] a661_layer/[10] CbFuelType/CbFuelType_A661_VISIBLE/Emit

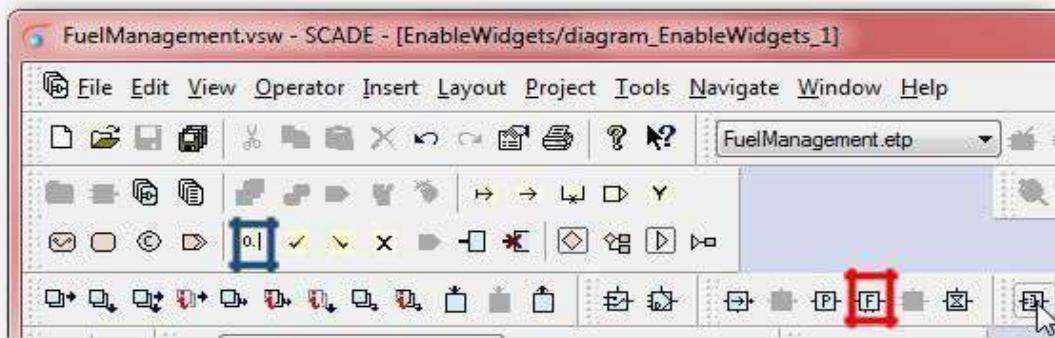


2.21 Agora, com as conexões feitas, clique no operador ShowLayer e adicione o seguinte código (o output LayerVisible está dentro da pasta interface do operador ShowLayer):

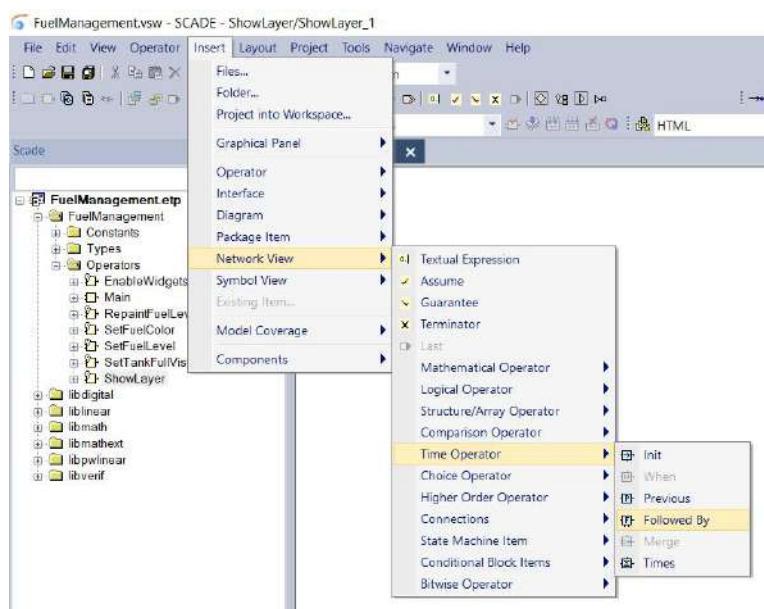


Ele retornara o valor false seguido de true.

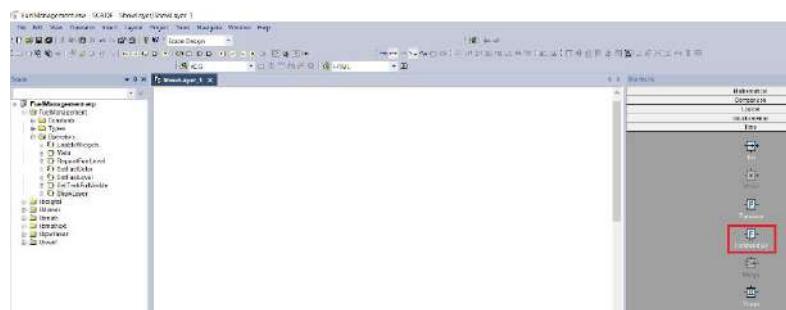
OBS: O bloco utilizado foi o **FollowedBy**, e a ferramenta utilizada para colocar o valor false no workspace está circulada em azul abaixo.



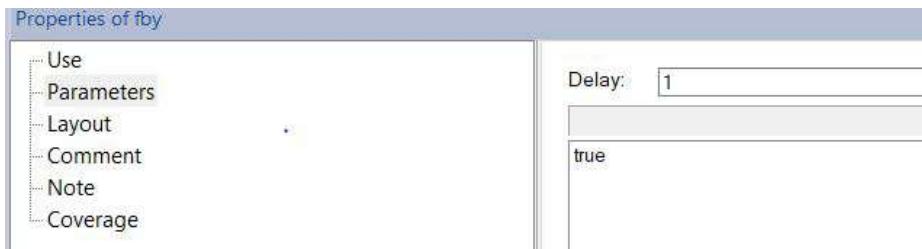
Ou



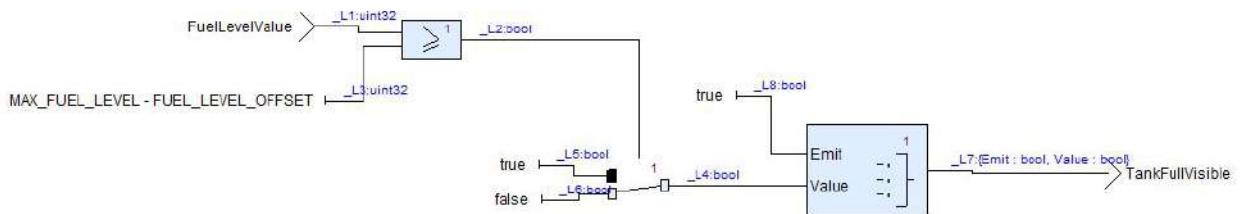
Ou



Obs2: Propriedade do bloco FBY:



2.22 Clique no operador SetTankFullVisible e adicione o seguinte código



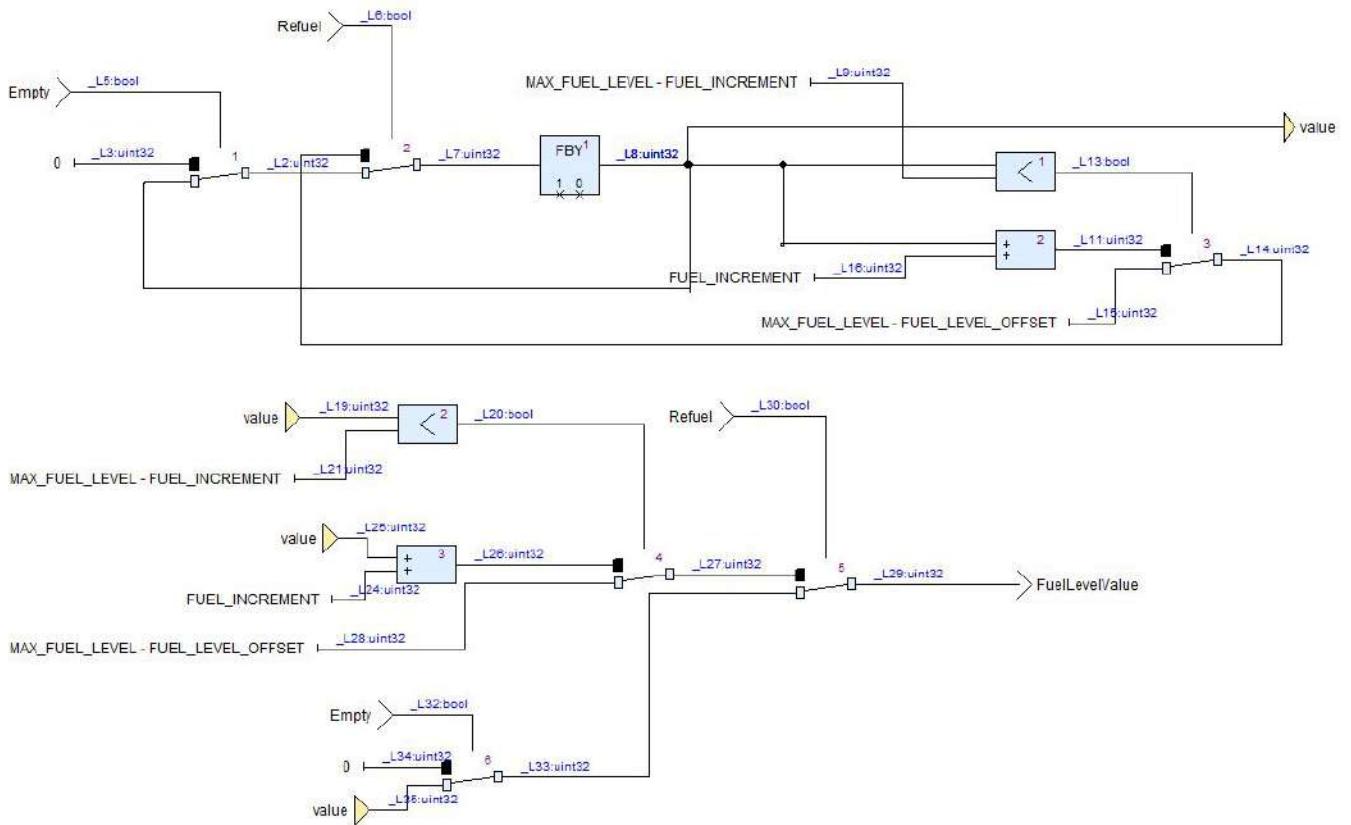
Ele será responsável por mostrar o tanque de combustível cheio por completo

OBS: Os blocos utilizados foram o **greater than or equal**, **data structure** e o **If/then**, para adicionar mais uma entrada no bloco **data structure** igual o da imagem, foi utilizado a ferramenta circulada em verde:

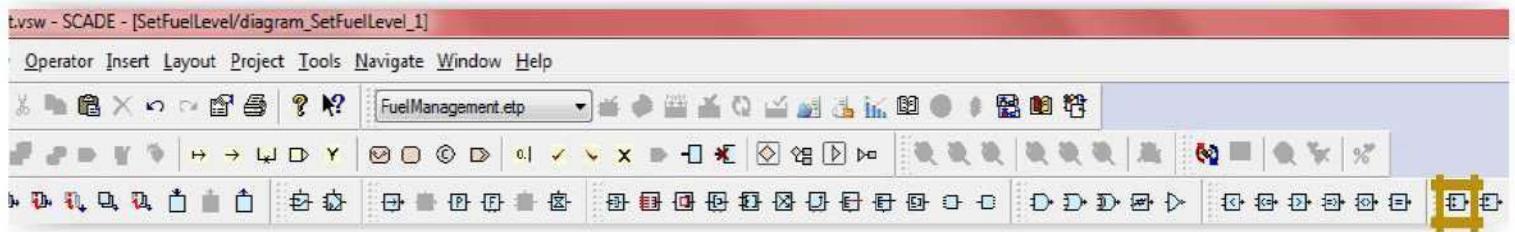
OBS 2: Outra forma de adicionar duas entradas no bloco **data structure** é simplesmente selecionando ele e em "Use" escolher "Input number: 2".



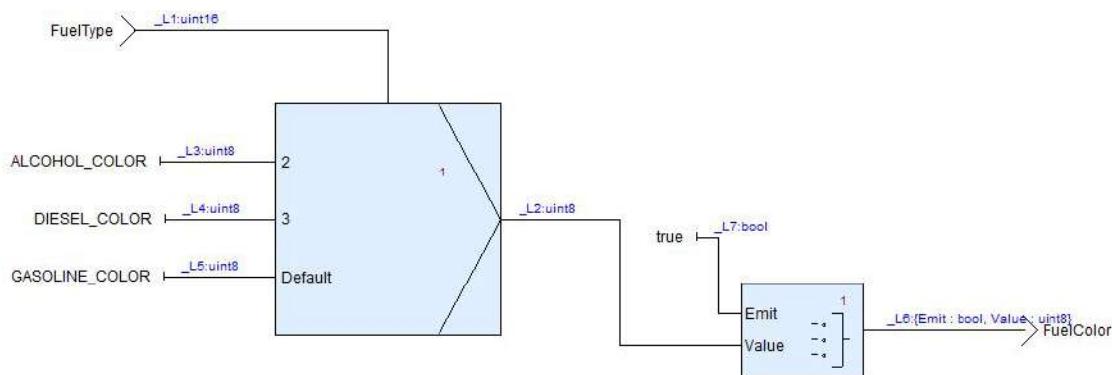
Agora clique no operador SetFuelLevel e adicione os seguintes códigos. (obs, note que o bloco value é a variável local do SetFuelLeve e para trocar para output basta seleciona-lo depois de adiciona-lo, clicar na propriedade “use” e trocar para input ou output) (obs2, os blocos com valor 0 devem ser do tipo uint32)



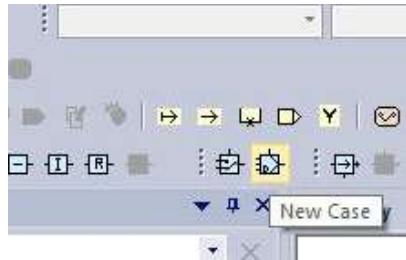
Ele será responsável por definir a quantidade de combustível no tanque O novo bloco utilizado foi o **Plus**.



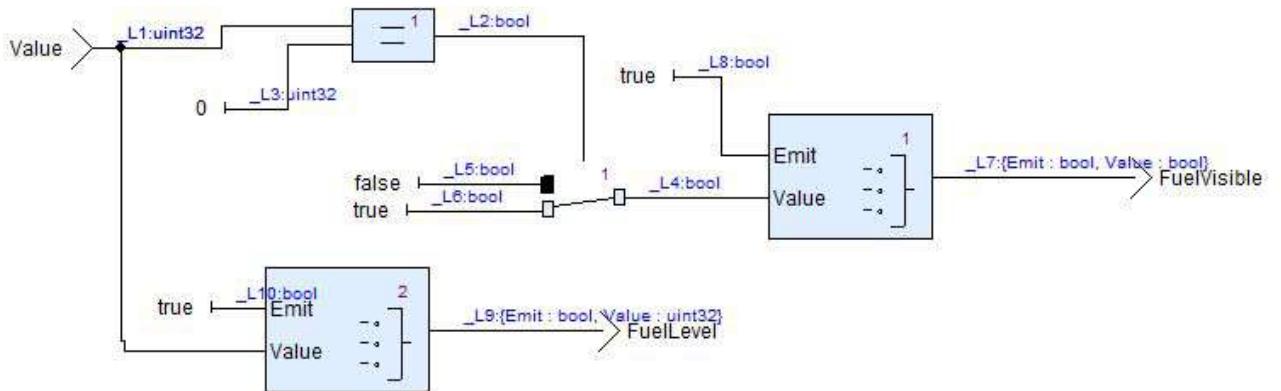
2.23 Agora clique no operador SetFuelColor adicione o seguinte código



Este código será responsável por ler qual combustível o usuário escolheu **OBS:** O novo bloco utilizado foi o **Case**.



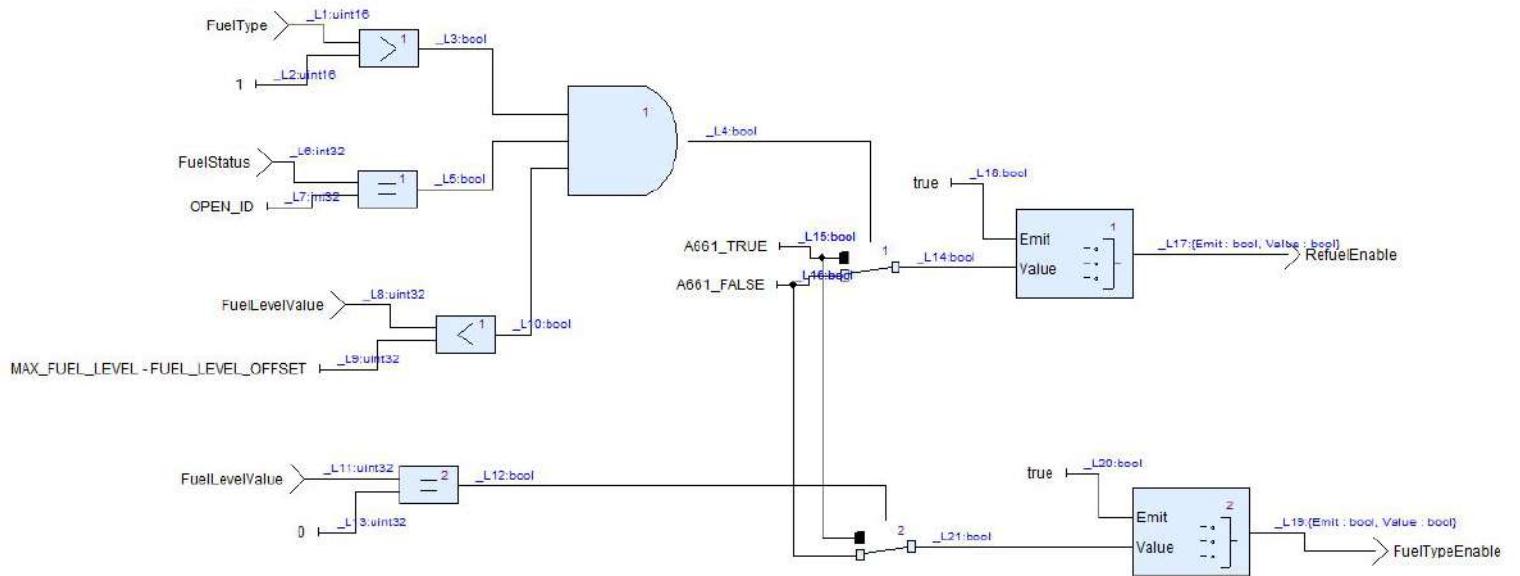
2.24 Agora, clique no operador RepaintFuelLevel adicione o seguinte código:



Este código é responsável por atualizar a quantidade de combustível no tanque.

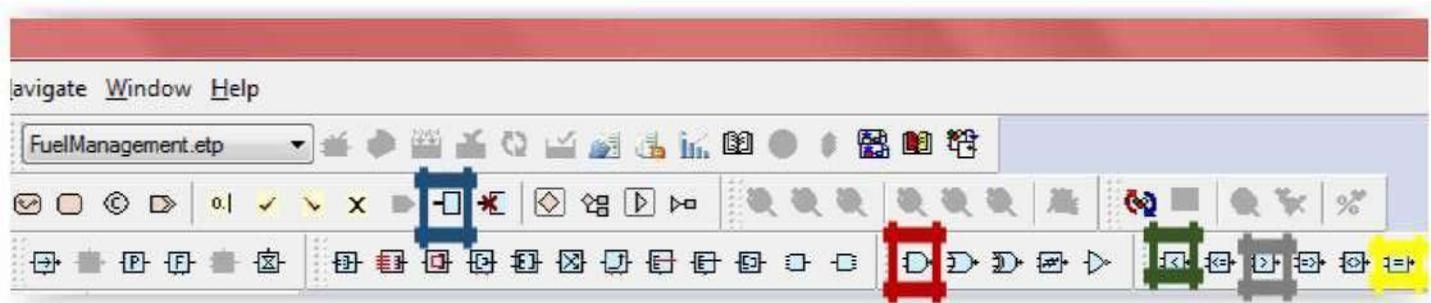
OBS: O bloco novo utilizado foi o “Equal” que fica junto com as demais comparações.

2.25 Clique no operador EnableWidgets adicione o seguinte código:

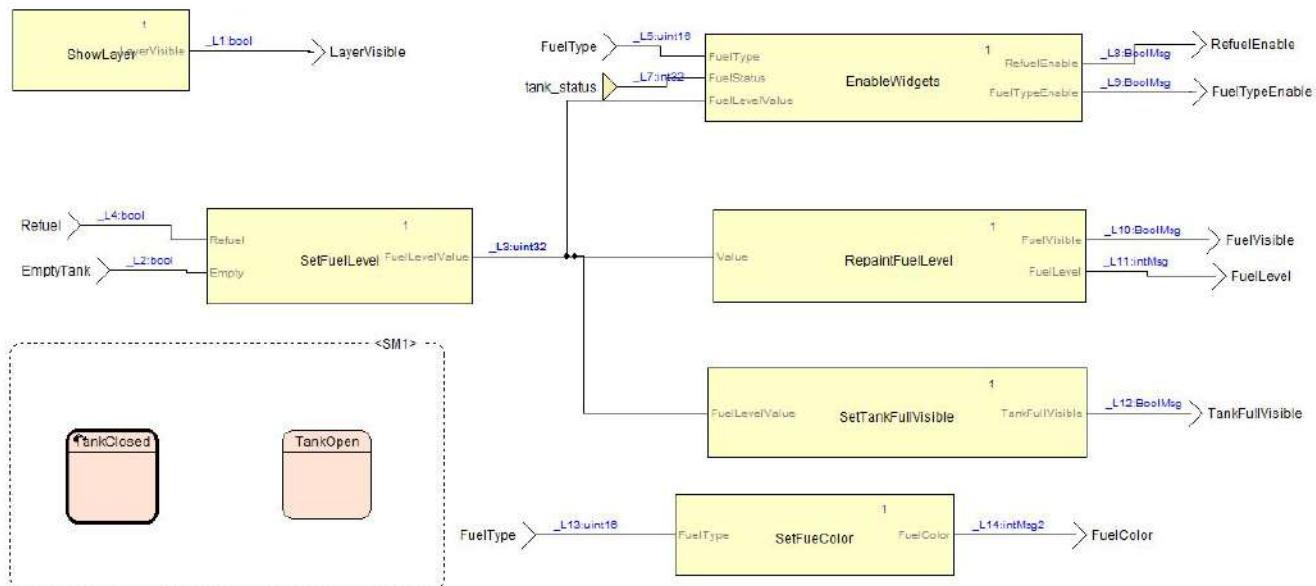


Esse código vai ser responsável por ativar e desativar as opções de reabastecimento.

OBS: Os blocos novos utilizados neste código são Equal, Greater than, Lesser than e And, o AND possui uma interface a mais que foi adicionada utilizando a ferramenta destacada em azul (ou trocar a propriedade input number para 3) (verificar tambem os tipos, alguns sao uint16 e outros uint32):

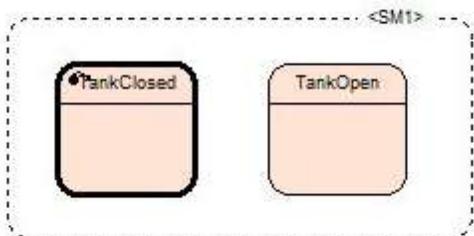


2.26 Agora, adicione o seguinte código no operador Main:

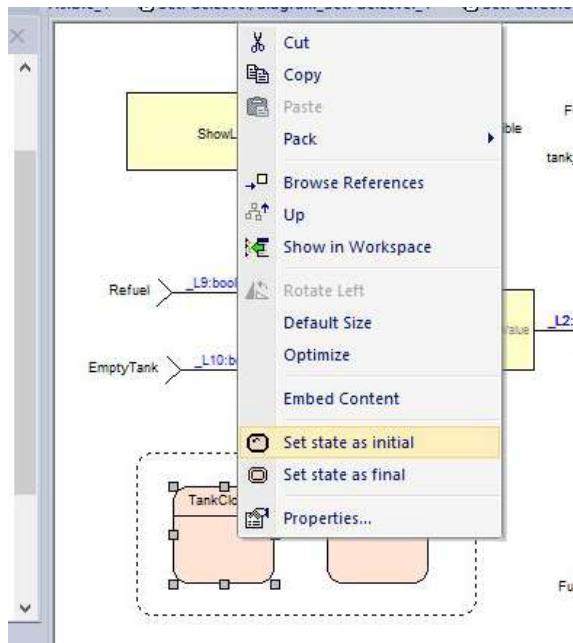


O item <SM1> pode ser adicionado através do menu: Insert -> Network View -> State machine item -> state, fazer esse processo suas vezes na mesma máquina de estado.

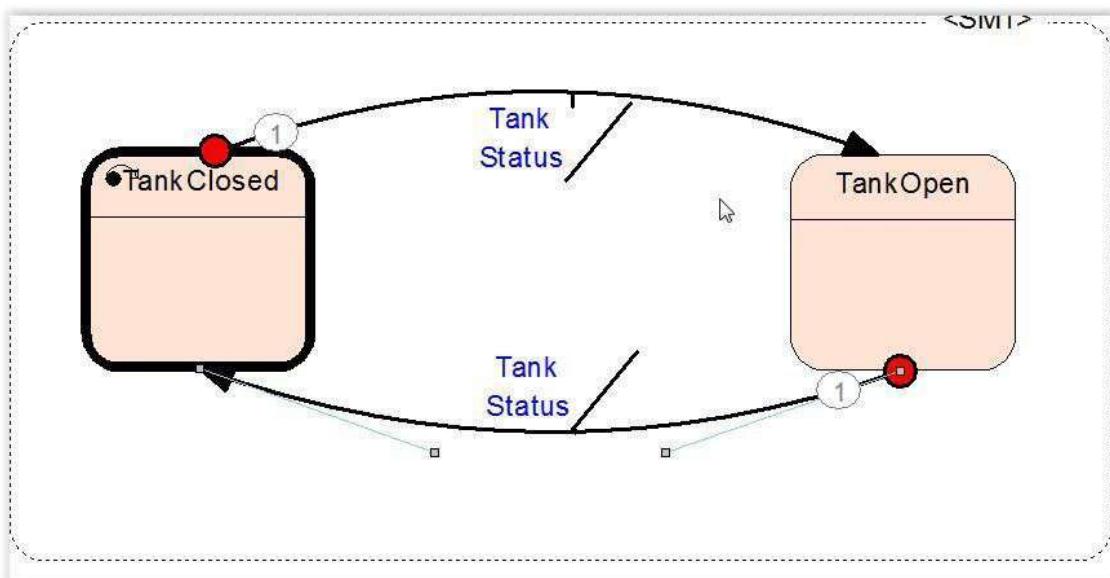
2.27 Clique no estado TankClosed com o botão direito e vá na opção e clique no botão Hide Content, faça o mesmo com o estado TankOpen, o resultado deverá ser como a imagem abaixo:



2.27 Agora, clique com o botão direito no estado TankClosed e escolha a opção Set State as initial, ATENÇÃO : não faça o mesmo com o outro estado !



2.28 Faça a conexão entre os estados como indicado na figura abaixo, e mude o texto azul para TankStatus:



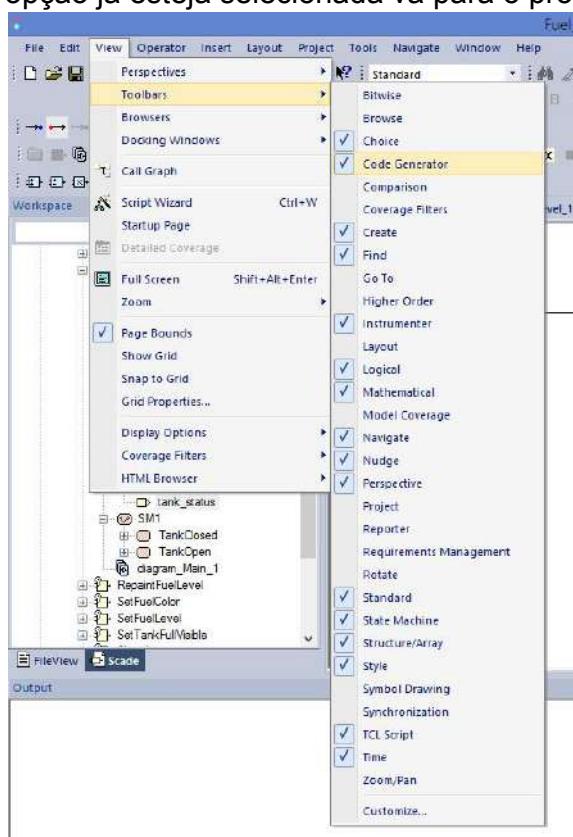
2.29 Clique duas vezes no estado TankClosed e adicione o seguinte código:



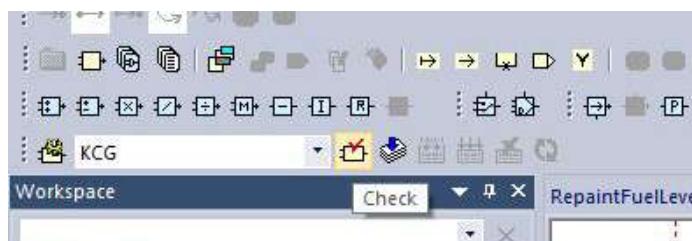
2.30 Agora, clique duas vezes no estado TankOpen e adicione o seguinte código:



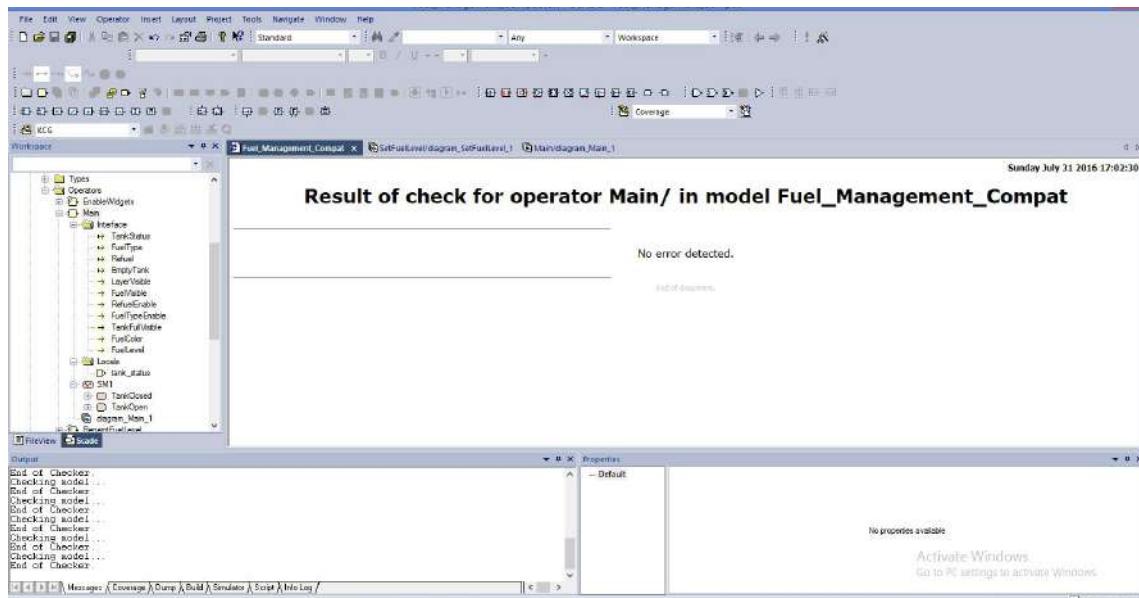
2.31 Vá na opção view/Toolbars e marque a opção Code Generator(Caso a opção já esteja selecionada vá para o próximo passo):



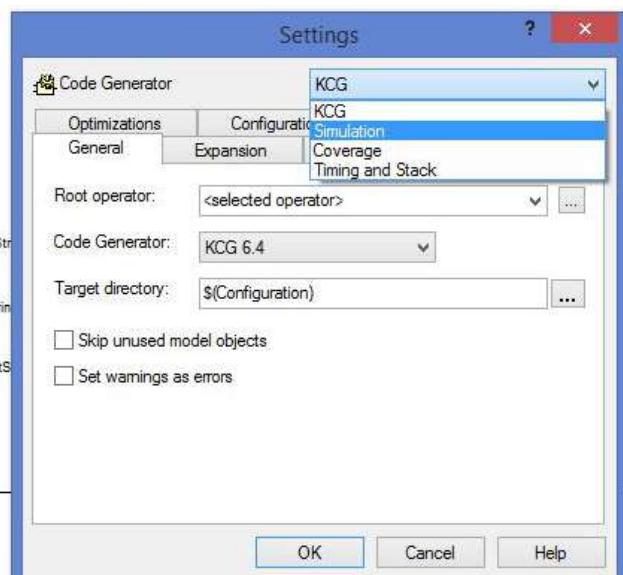
2.32 Clique em um operador “Main” e logo em seguida clique no botão Check para testar se há alguma falha no programa:



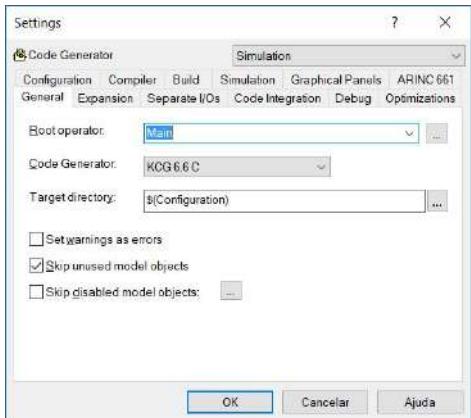
Se tudo der certo esta mensagem deverá aparecer na tela:



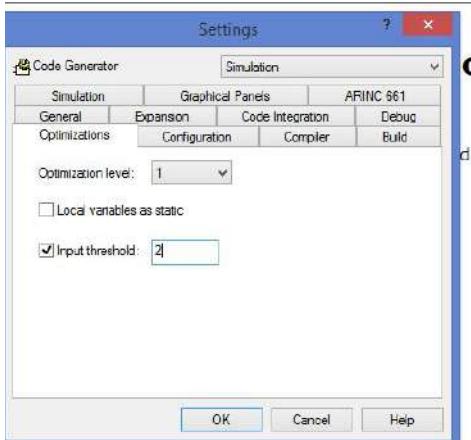
2.33 Agora, você deve configurar algumas coisas, clique na aba Project/code generator/settings e escolha a opção simulation:



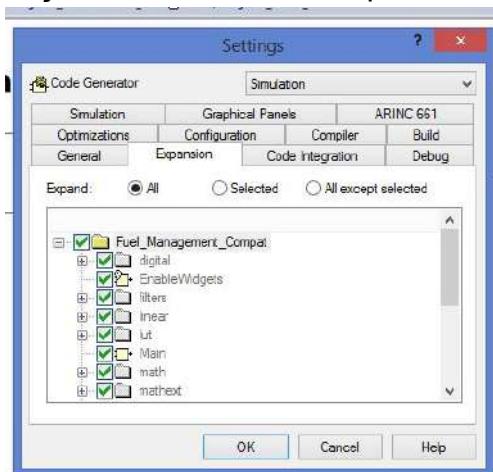
Na aba general, deixe todas as opções iguais a da imagem abaixo:



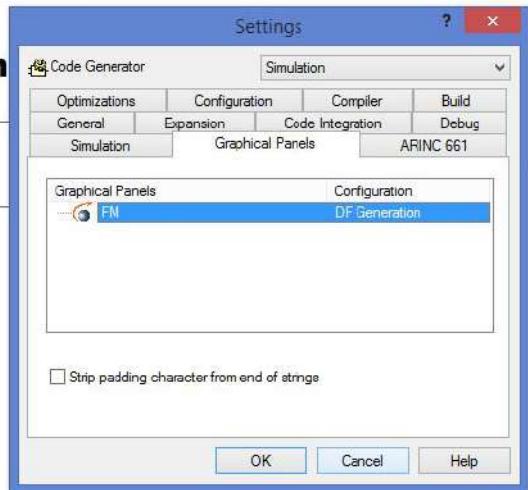
Faça o mesmo com a aba Optimizations:



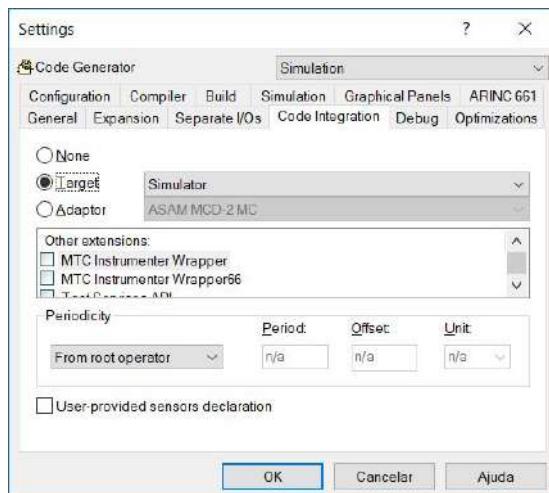
Faça o mesmo na aba Expansion:



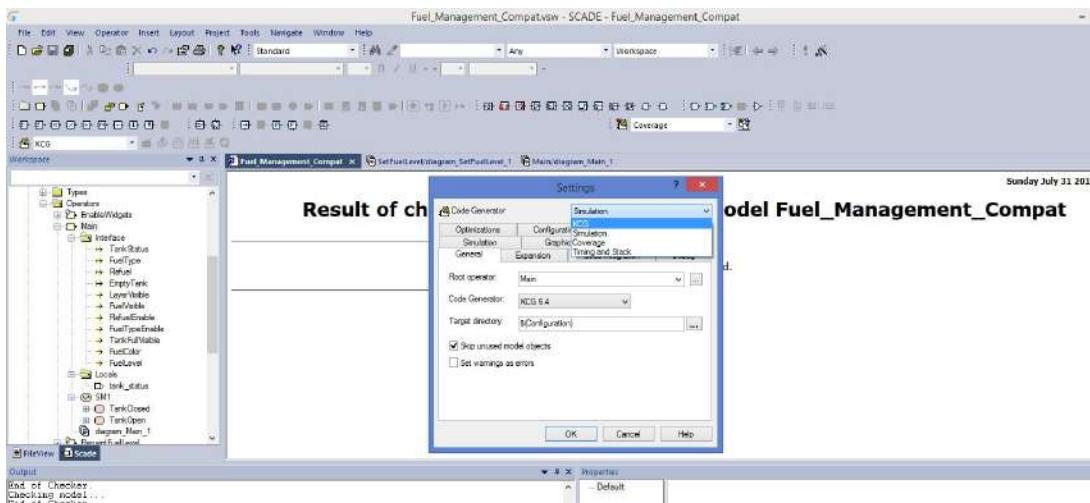
Faça o mesmo na aba Graphical Panels:



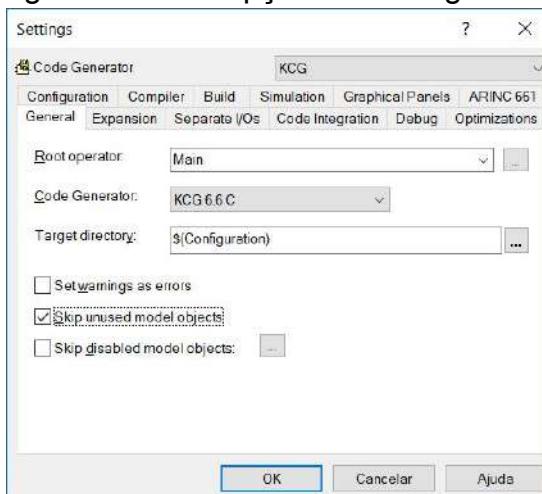
Faça o mesmo na aba Code Integration:



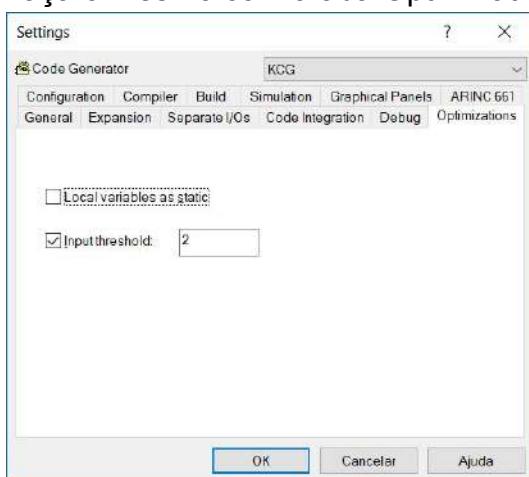
Agora escolha a opção KCG:



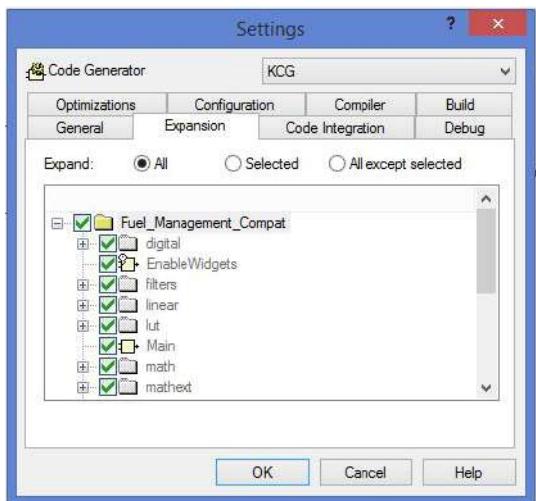
Agora deixe as opções da aba general igual aos da figura abaixo:



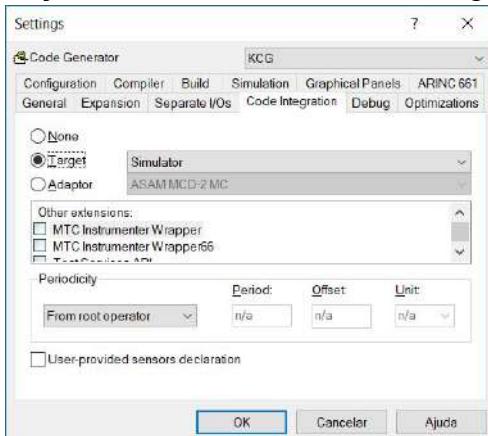
Faça o mesmo com a aba Optimizations:



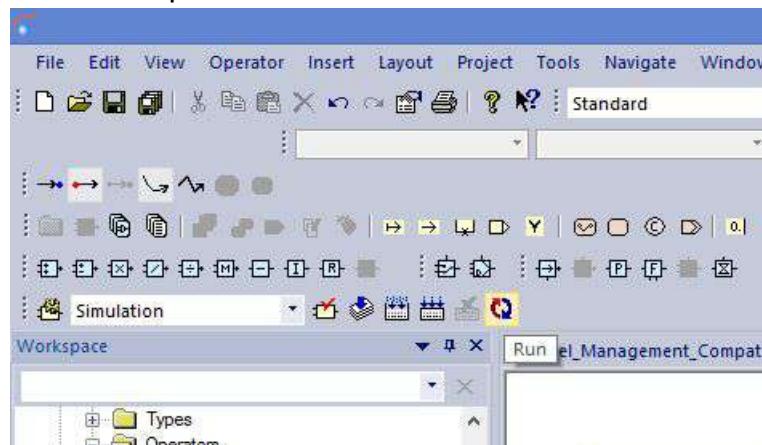
Faça o mesmo com a aba Expansion:



Faça o mesmo na aba Code Integration:



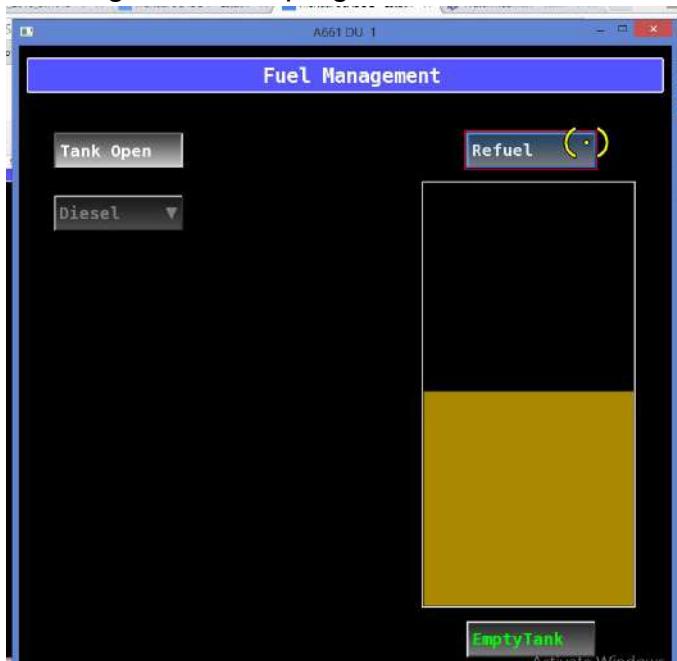
2.34 Agora, simule o programa escolhendo a opção simulation e apertando **rebuild all** e depois **run** neste menu:



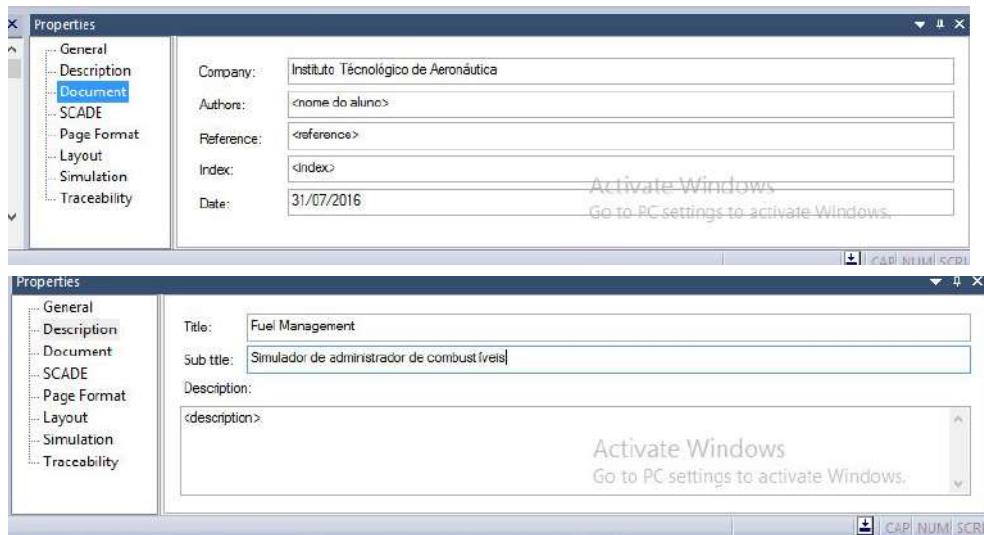
2.35 Após algum tempo, uma tela preta deverá aparecer, clique no botão go para aparecer a interface gráfica:



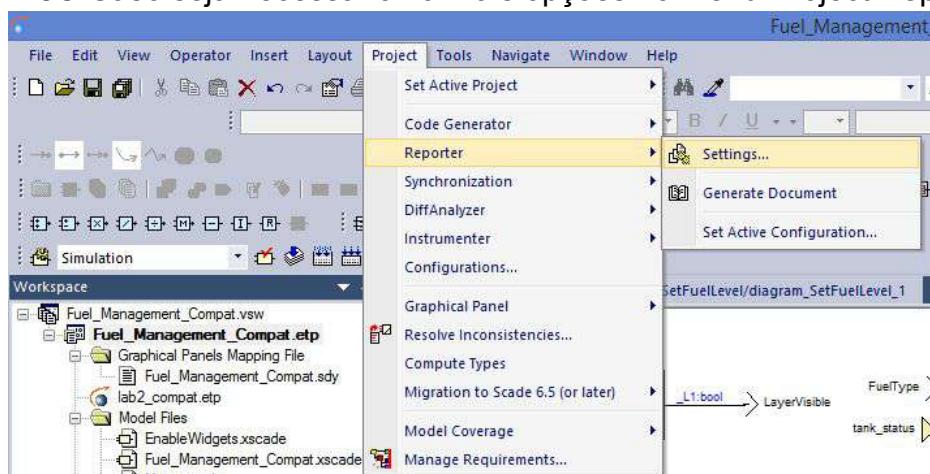
2.36 Agora teste o programa:



2.37 Antes de fazer o report, é necessário preencher algumas informações, clique na aba *File View* e clique com o botão direito em propriedades do arquivo *FuelManagement.etc*, nas propriedades, os menus que devem ser preenchidos são : *Description* e *Document*.

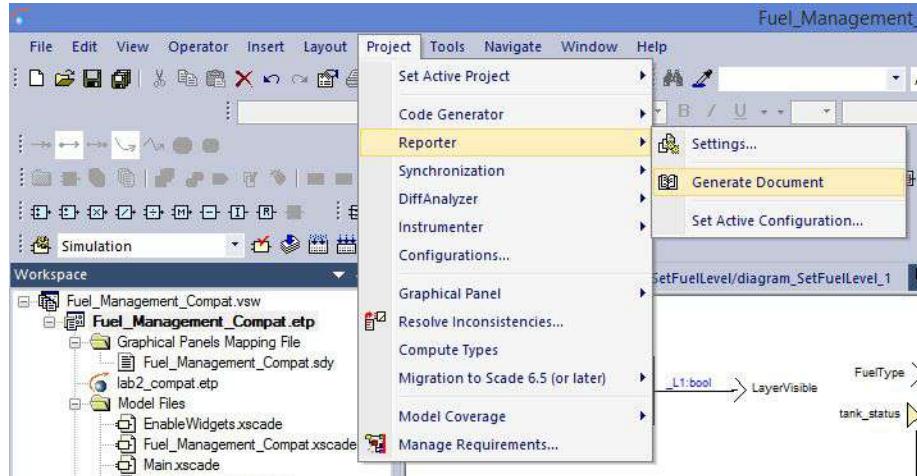


2.38 Caso seja necessário há mais opções no menu Project/Reporter/Settings

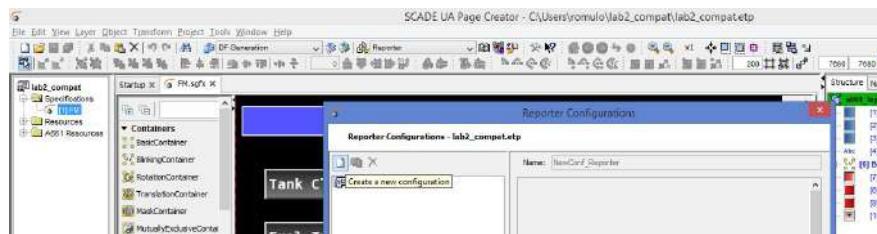


2.39 Agora, clique em *Generate Document*, se tudo der certo um documento

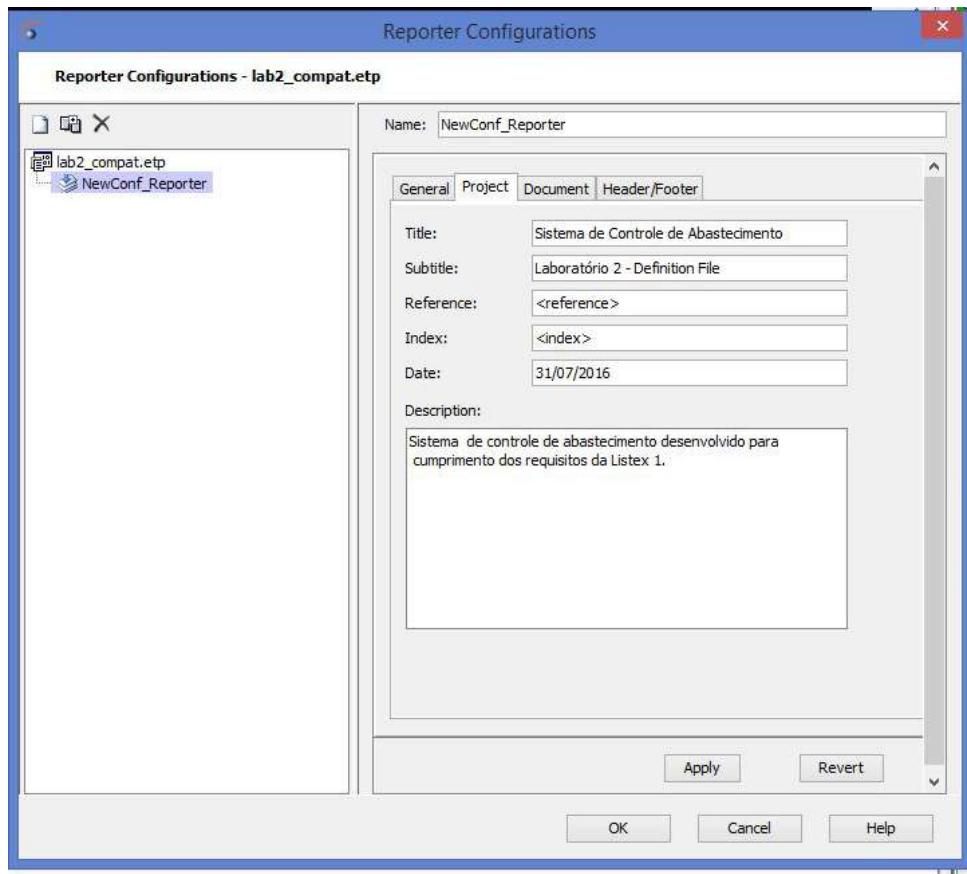
.html será criado na pasta do projeto



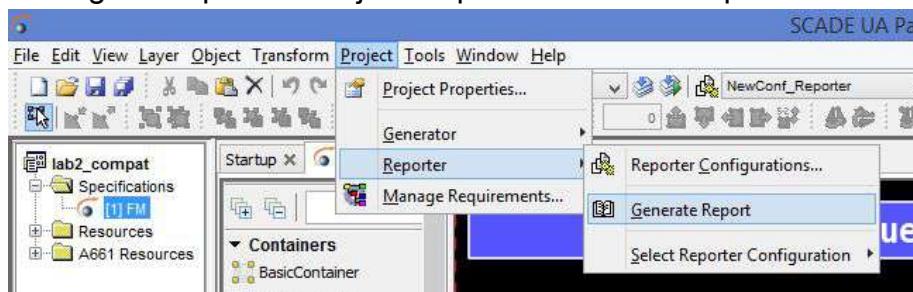
3.4 Agora abra o SCADE Page Creator e abra o a Definition file que forá feito mais cedo,e vá em Project/Reporter/Reporter Configurations e adicione uma nova configuração.



Preencha os campos Project e Document conforme abaixo:



3.5 Agora clique em Project/Reporter/GenerateReport



3.6 Agora vá na pasta do seu projeto, todos os documentos deverão estar lá, segue exemplo do arquivo html gerado:

'FM' SPECIFICATION

SCADE Display R17 Sun Jul 31 18:02:03 BRT 2016 [Printable Document Version](#)

Table Of Contents <ul style="list-style-type: none"> 1. Specification Description <ul style="list-style-type: none"> 1.1. General Description 1.2. Specification Overview 1.3. Color Table Description 1.4. Line Width Table Description 1.5. Line Shape Table Description 1.6. Font Table Description 1.7. Layer List 1.8. Picture Table Description 1.9. Symbol Table Description 2. Layer: #661_layer <ul style="list-style-type: none"> 2.1. Layer Structure 2.2. Layer Content 	<p>1. Specification Description</p> <p>This section describes the configuration of the specification 'FM'.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;">File name</td> <td>C:\Users\romulo\lab2_compete\FM.scd</td> </tr> <tr> <td>Window dimension (pixels)</td> <td>798x768</td> </tr> <tr> <td>Color Mode</td> <td>RGB</td> </tr> </table> <p>1.1. General Description</p> <p>Sistema de controle de abastecimento desenvolvido para cumprimento dos requisitos de Utter 1.</p> <p>1.2. Specification Overview</p> <p>This subsection shows a complete multi-layers overview of the specification 'FM'.</p> 	File name	C:\Users\romulo\lab2_compete\FM.scd	Window dimension (pixels)	798x768	Color Mode	RGB
File name	C:\Users\romulo\lab2_compete\FM.scd						
Window dimension (pixels)	798x768						
Color Mode	RGB						

Referências

Portal do Projeto SI-LANSAB. Disponível em:

<https://sites.google.com/site/projislansab/> Acesso em 10 de julho de 2015

Portal do Projeto SI-GAC. Disponível em:

<https://sites.google.com/site/projetosigac2015/> Acesso em 10 de julho de 2015

Portal da International-pilot. Disponível em:

<http://www.international-pilot.com/first-batch-pilots-trained-embraer-legacy-500/> .

Acesso em 18 de julho de 2015

MIRACHI Samoel; JUNIOR Carlos Lopes Nunes. WarmUp - Display ARINC

661 de um trem de pouso (LandingGear). Campus Monte Negro. 2013

MANUAL DA EMBRAER. Treinamento Scade ITA. 2013

SCADE FOR DUMMIES. 2013

OBS: As imagens dos procedimentos deste manual foram produzidas pelo próprio autor, utilizando-se da ferramenta Scade.