

MANUAL

NOVO FUNCIONAMENTO FORMS

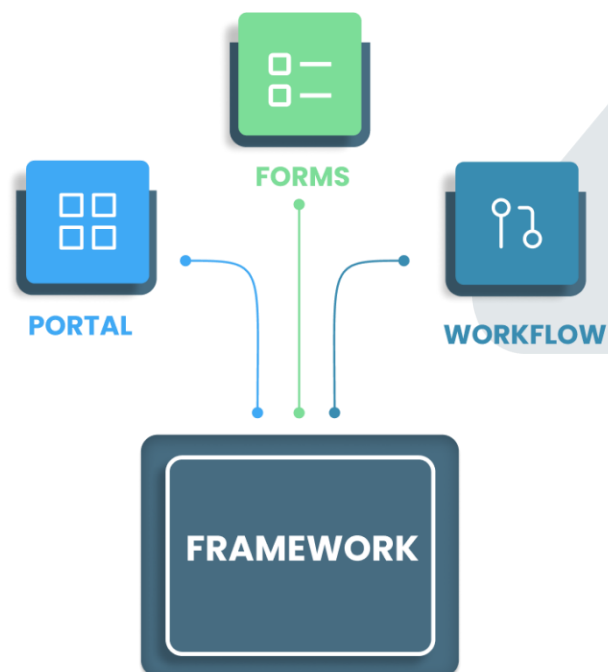


Forms

Novo engine de funcionamento

Neste manual trataremos no forms e seu novo engine de funcionamento.

Abordaremos as principais mudanças e atualizações de performance e melhorias de experiência e uso.



Sumário

1. Instalação do forms

1. Atualizar o produto para a versão que possui esse arquivo
- 1.2 Copiar o conteúdo de novoforms\portalfiles para o diretório de instalação do Zuri
- 1.3 Copiar o conteúdo de novoforms\servicefiles para o diretório de instalação do serviço do Zuri
- 1.4 Colocar a tag `<add key="Zuri.Forms.UseZuriForms" value="true" />` no APPSETTINGS do Web.Config
- 1.5 Clicar em instalar o novo forms na configuration.aspx
- 1.6 Dar um refresh na página e vão aparecer outros botões executar na seguinte sequência:
 - 1.6.1 Migrar os formulários (botão Migrar formulários)
 - 1.6.2 Migrar os módulos de página: forms e reports (botão Migrar módulos de formulários)
 - 1.6.3 Migrar os módulos do workflow (botão Migrar formulários em processos)
- 1.7 Republicar todas as páginas do portal

2. Novo Engine de funcionamento

- 2.1 Exclusão de componentes antigos
- 2.2 Grid Fórmulas
- 2.3 Namespace: Zuri.Forms

3. Melhorias gerais e padronização das configurações dos campos

- 3.1 Modernização dos componentes
- 3.2 Remoção do AjaxControl Toolkit
- 3.3 Atualização do controle de variáveis de contexto

4. Atualização de bibliotecas e persistência

- 4.1. Atualização das bibliotecas de acesso a dados
- 4.2. Atualização do modelo de persistência de configurações
- 4.3. Remoção da tabela FMOD_MODULO
- 4.4. Gravação das informações de configuração no banco
- 4.5. Mudança na gravação de dados no banco

5. Alterações no processo de geração e testes

- 5.1. Testes sem publicação
- 5.2. Testes sem gravação de dados
- 5.3. Validação mais completa dos campos
- 5.4. Controle de versões para formulário
- 5.5. Execução de formulários e controle de acesso

6. Melhorias na configuração dos campos

- 6.1. Valor inicial para campos
- 6.2. Propriedade título para todos os campos
- 6.3. Troca de tipo de campo

7. Expressões**8. Melhorias no Form Filho**

- 8.1. Substituição do form grid pelo form filho
- 8.2. Seleção de filhos de formulários
- 8.3. Abertura de form filho
- 8.4. Novas ações para abrir um form filho

9. Melhorias no Campo DIV

- 9.1. Possibilidades de configuração de modo de exibição

- 9.2. Controle de arquivo

- 9.3. Campos do tipo texto com AutoComplete

10. Melhorias na execução do formulário

- 10.1. Alteração do texto do botão "OK"

- 10.2. Preenchimento parcial do formulário

- 10.3. Controle de eventos e dependência

- 10.4. Validação do formulário por datasource

- 10.5. Validação de obrigatoriedade

- 10.6. Campos de funcionamento como hidden

11. Melhorias na fonte de dados

- 11.1.1. Execução de fontes de dados

- 11.1.2. Inserção de nome da fonte de dados

- 11.1.3. Configuração de Cache

- 11.1.4. Preenchimento de form filho como gatilho

12. Melhoria nas fórmulas

- 12.1. Carregamento por MEF

- 12.2. Novas fórmulas

- 12.3. Adição de novas fórmulas

13. Módulo Forms no Portal

- 13.1. Múltiplas ações ao preencher o formulário

- 13.2. Configuração de seções

- 13.3. Chamadas de integração com Workflow

14. Módulo Reports no Portal

- [14.1. Melhoria na configuração](#)
 - [14.2. Opção de filtro](#)
 - [14.3. Formatação decimal](#)
 - [14.4. Ordenação de campos](#)
 - [14.5. Definição de grupos](#)
 - [14.6. Inclusão de gráficos](#)
- 15. **Módulo Forms no Workflow**
 - [15.1. Substituição de módulos antigos](#)
 - [15.2. Preenchimento parcial do formulário](#)
 - [15.3. Seleção de registro](#)
 - [15.4. Exibição de arquivos nos anexos](#)
- 16. **Módulo Formdata no Workflow**
 - [16.1. Seleção de registros](#)
 - [16.2. Geração de anexo](#)
- 17. **Módulo SelectForm no Workflow**
- 18. **Formulário Anônimo**
- 19. **Comentários Gerais**
 - [19.1. Alteração de tipo e alias de campo](#)
 - [19.2. Manutenção de arquivos XML e procedures](#)
 - [19.3. Configurações adicionais e tags](#)
- 20. **Uso Interno**
 - [20.1. Tag Zuri.Forms.AllowDBAlter](#)
 - [20.2. Uso simultâneo de forms antigo e novo](#)

[20.3. Migração de templates e CSS](#)



1. FORMS

INSTALAÇÃO

2. Atualizar o produto para a versão que possui esse arquivo

- 1.8 Copiar o conteúdo de novoforms\portalfiles para o diretório de instalação do Zuri
- 1.9 Copiar o conteúdo de novoforms\servicefiles para o diretório de instalação do serviço do Zuri
- 1.10 Colocar a tag `<add key="Zuri.Forms.UseZuriForms" value="true" />` no APPSETTINGS do Web.Config
- 1.11 Clicar em instalar o novo forms na configuration.aspx
- 1.12 Dar um refresh na página e vão aparecer outros botões executar na seguinte sequência:
 - 1.12.1 Migrar os formulários (botão Migrar formulários)
 - 1.12.2 Migrar os módulos de página: forms e reports (botão Migrar módulos de formulários)
 - 1.12.3 Migrar os módulos do workflow (botão Migrar formulários em processos)
- 1.13 Republicar todas as páginas do portal

2. NOVO ENGINE DE FUNCIONAMENTO

Melhorias no tratamento de datasource, melhorias no editor, controle de versão (publicada e editada).

Para que seja possível realizar a migração para o Novo Forms, alguns **pré-requisitos** devem ser estabelecidos, sendo eles:

2.1 – Exclusão dos componentes antigos

- Exclusão do anterior (Webfoundations.WFForms)
- Exclusão do Forms Mobile (WF.Ninki)

2.2 – Grid Fórmulas

- Caso existam, as Grid fórmulas deverão ser reconfiguradas para o novo modelo.
- Fórmulas utilizadas na versão antiga passarão a existir como WebserviceSOAP após a migração.

2.3 – Novo Namespace

- Novo Namespace: Zuri.Forms

3. MELHORIAS GERAIS E PADRONIZAÇÃO DAS CONFIGURAÇÕES DOS CAMPOS

3.1 – Modernização dos componentes (bibliotecas externas)

3.2 – Remoção do AjaxControl Toolkit

O acesso as variáveis de contexto do processo foram atualizadas e a seleção passa a ser por categoria do módulo juntamente com as variáveis que competem a cada um deles.

3.3 – Atualização das variáveis de contexto

Foi realizada a atualização do controle de variáveis utilizando o novo esquema de contexto para o forms, portal e para o workflow. Esse último possui a particularidade de atualização por categoria conforme mencionado anteriormente. Abaixo segue todas as funções existentes:

1. AddDays(NumberOfDays) [%#ADDDAYS(2)#%] – dias corridos
2. EncryptNumber(value) [%#ENCNUMBER(1)#%] – número encriptado
3. GetDownloadLink(url;expminutes;accesslist) [%#GETDOWNLOADLINK(/WFResources/Themes/images/zuri.png;15;-1)#%] – link para download com tempo determinado de disponibilidade e de acesso parametrizado

4. `GetNewGUID()` [%#GETNEWGUID()#%] – novo GUID
5. `GetUserEmail(UserId)` [%#GETUSEREMAIL(%WFUser.Id%)#%] – e-mail do usuário a partir do ID do usuário, por exemplo.
6. `GetUserName(UserId)` [%#GETUSERNAME(%WFUser.Id%)#%] – nome do usuário a partir do ID do usuário, por exemplo.
7. `IsInRoles(AccessList)` [%#INROLE(-1)#%] – testa se o usuário logado faz parte de um grupo específico
8. `MonthEnd` [%#MONTHEND()#%] – último dia do mês corrente
9. `MonthStart` [%#MONTHSTART()#%] – primeiro dia do mês corrente
10. `RoleUsersMail(RoleId)` [%#ROLEUSERSMAIL(-1)#%] – lista de e-mails de usuários de um grupo
11. `Spaces(value)` [%#SPACES(1)#%] – inserir espaço
12. `ToString` [%#TOSTR(99999;#;###.00)#%] – formato de número. Observação: válido para qualquer máscara que o .NET interpretar.
13. `WeekEnd` [%#WEEKEND()#%] – último dia da semana
14. `WeekStart` [%#WEEKSTART()#%] – primeiro dia da semana



4. Atualização das Bibliotecas de Persistência

4.1 Atualização de todas as bibliotecas de acesso a dados (DataAccess / DBMapper).

O forms até o momento ainda utilizava as bibliotecas antigas do Zuri Framework por questões de compatibilidade de arquitetura, com a melhoria na arquitetura foi feita a atualização para a utilização das versões mais modernas já utilizadas pelo Zuri Workflow e agora existe uma maior padronização entre os dois módulos.

4.2 – Atualização do modelo de persistência de configurações para ORM.

O forms utilizava o modelo de gravação de dados no banco através de procedures (fd_*), na nova arquitetura este formato de gravação foi atualizado e passou a utilizar o Entity Framework da Microsoft, facilitado a consulta, gravação e manutenção de informações.

4.3 – Remoção da tabela FMOD_MODULO (mudança para MEF)

A tabela FMOD_MODULO tinha como função realizar a montagem do editor do forms antigo e ela retorna quais módulos existem. Para esse forms houve a mudança para MEF que é uma tecnologia que carrega automaticamente de acordo com os

módulos que possuem no diretório e precisam ser carregados. Nesse caso cada módulo é independente, ou seja, independente da criação de uma nova funcionalidade para o forms, não terá impacto nas outras já existentes.

4.4 - Gravação das informações de configuração diretamente no banco (exclusão dos XMLs)

Remoção de XMLConfig.XML, XMLDatasources.XML e XMLLayout.XML passando a ser na tabela de formulário. Isso facilita a migração de ambientes, uma vez que não há mais a necessidade da APPDATA.

4.5 - Mudança na forma de gravação de dados no banco passando a utilizar o Entity Framework

A forma como ocorre a gravação e leitura de dados do formulário na tabela foi alterada passando da utilização de procedures para informação na coluna de configuração, passando a ser registro no banco. Ao realizar a migração de um formulário, automaticamente os XMLs serão apagados do diretório e partir de então tudo funcionará com o Entity Framework. Caso o usuário precise retornar os XMLs devido alguma customização específica, deve-se colocar a tag `Zuri.Forms.KeepCompatibleConfigurations = true` no web.config e ela irá manter os XMLs após a migração de um formulário e caso tenha algum versionamento desse mesmo formulário, ele irá regerar as procedures normalmente.

- **Remoção do campo FOR_N_VINCULA_USUARIO**

Esse campo tinha a função de trazer o mesmo registro para um usuário que retorna para o mesmo formulário que ele já realizou algum preenchimento

Alterações na gravação de dados no banco: Foram realizadas alterações na gravação de dados no banco. São elas:

- **CodigoAutomatico**

Passa a se chamar **Id** (mantido com o nome original em migrações)

- **CodigoUsuario**

Passa a se chamar **ZUserId** (no caso de migrações, o nome original permanece)

- **ZWF_RequestId**

Novo campo para guardar informação do item workflow

Novo campo para guardar informação do ID processo quando possui integração com Workflow - workflow ZWF_RequestId

- **ZWF_RequestSPId**

Novo campo para guardar informação do subprocesso

Novo campo para guardar informação do ID do subprocesso quando possui integração com o Workflow - workflow ZWF_RequestSPId

Quando houver a gravação desses campos, sempre será considerado o último que teve atualização, ou seja, ele sempre será atualizado conforme tiver qualquer modificação no registro. A informação vem do contexto compartilhado.fo

Para que esses dois campos sejam, é preciso selecionar a opção de “incluir relacionamento com processos” na criação do formulário conforme imagem.

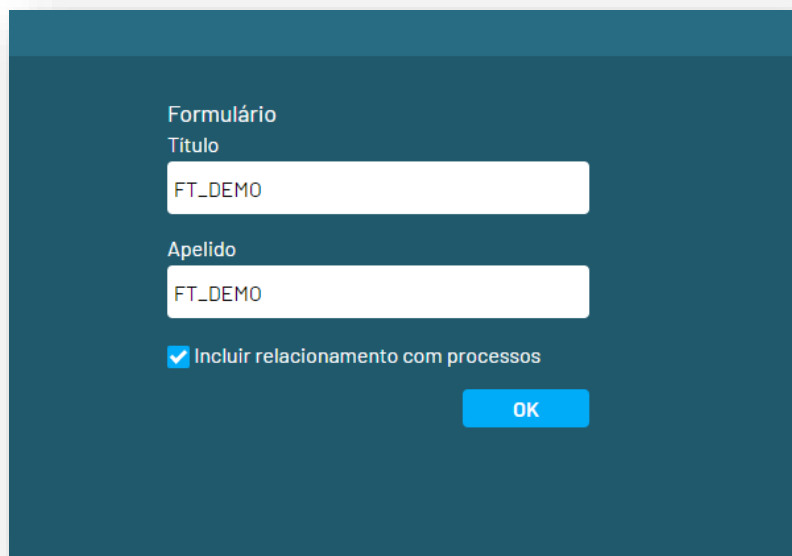


Figura 1 – Cadastro de novo formulário

Caso o usuário não selecione essa opção na criação do formulário ou queira remover, ele poderá realizar essa alteração dentro do formulário conforme imagem:

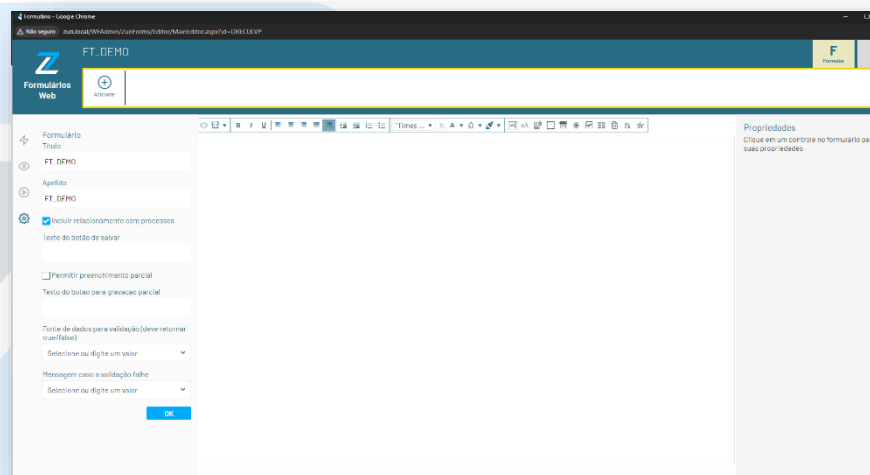



Figura 2 – Alterando as informações de formulário.

5. – Alterações no processo de geração e testes



5.1 – Testes sem publicação

O formulário pode ser testado sem ser publicado, desde que esteja corretamente configurado.

O Teste sem publicação é feito ao clicar no ícone  na barra de ferramentas e não gera gravação de dados no banco.

5.2 – Testes sem gravação de dados.

Ao se executar um teste do formulário publicado por padrão nenhum dado é gravado no banco, somente é exibida na tela a informação no formato que ela seria gravada. Caso seja necessária a gravação de informações no banco pode ser utilizada a tag de configuração:

Zuri.Forms.SaveDBValueInTestMode = true no web.config.

5.3 – Validação mais completa dos campos

Considerando o processo de publicação do formulário, a validação dos campos se tornar mais completa, validando configurações específicas de cada campo e também as referências quebradas de fontes de dados. Como por exemplo: validação de colunas definidas de um **form filho** ou **validação de itens de um dropdown**.

- Quando o usuário clicar no **'gerar formulário'** aparecerá um aviso na tela sinalizando que existem campos e/ou **datasources** que não estão sendo utilizados. Nesse caso ocorrerá normalmente a geração do formulário.

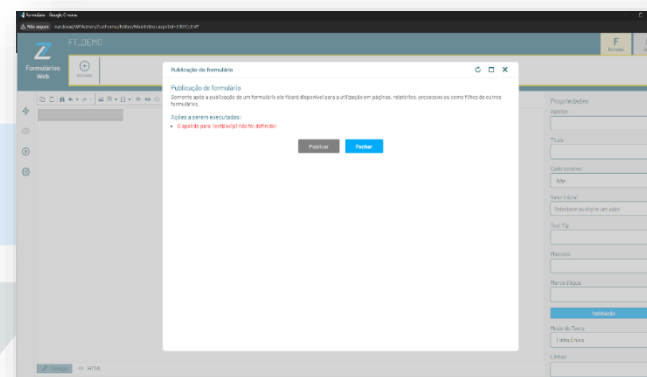


Figura 3 – Processo de Publicação e geração de testes

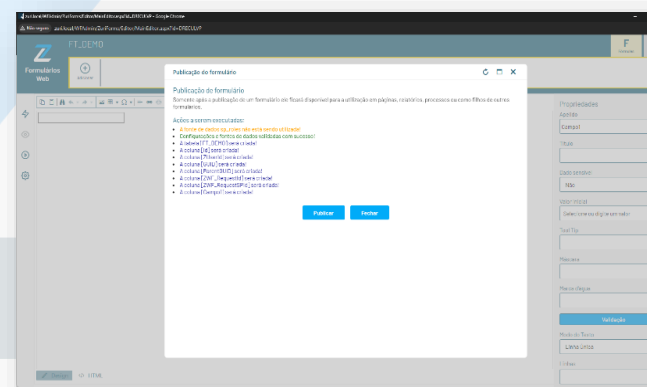


Figura 4 – Processo de Publicação e geração de testes

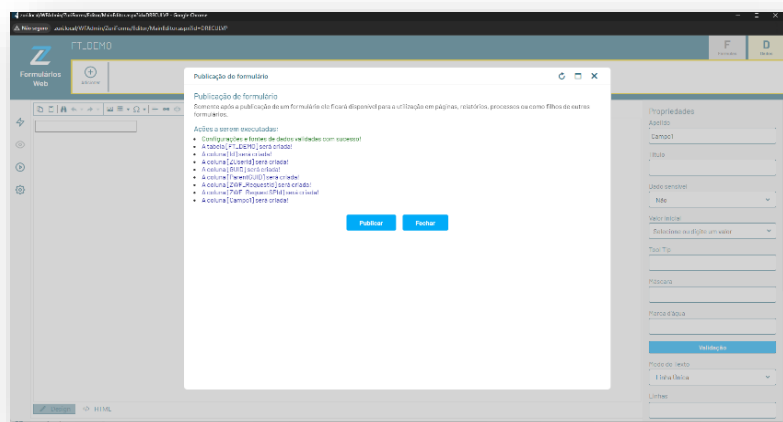


Figura 5 – Processo de Publicação e geração de testes

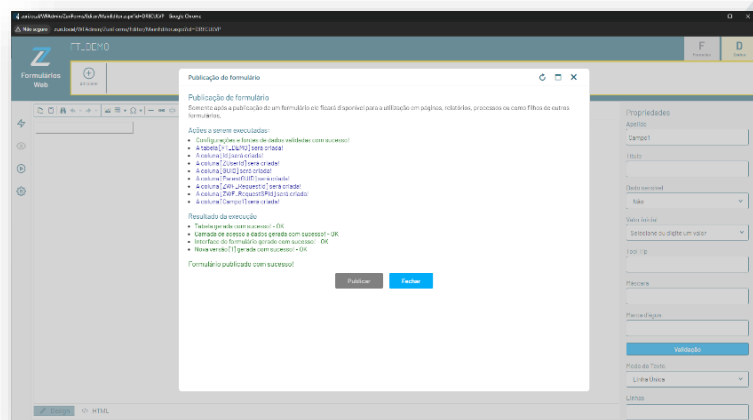


Figura 6 – Processo de Publicação e geração de testes

5.4- Controle de Versões para formulário

Existência de um controle de versões para formulário A cada publicação uma nova versão do formulário é gerada. Para cada nova versão gerada do formulário terá um layout, a configuração de todos os campos, **datasources**, **execconfig** e o **FRONT e BACK** do módulo gerado. A versão zero sempre será a versão em desenvolvimento

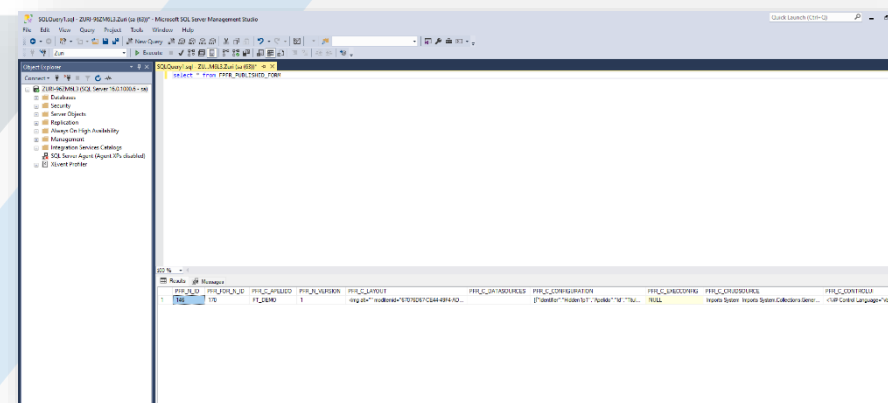


Figura 7 – Versionamento no banco de dados

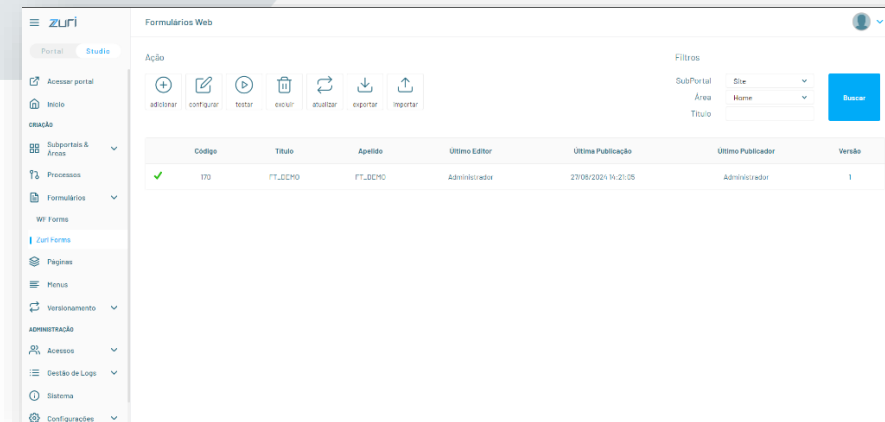


Figura 8 – Versionamento no Zuri

5.5 - Execução de formulários e controle de acesso.

Após formulário já estar gerado, em ambiente de DEV e a depender a licença aplicada, é possível apagar um determinado campo e ao clicar para regerar, aparecerá um aviso na tela sinalizando que é possível excluir a coluna que foi apagada do formulário.

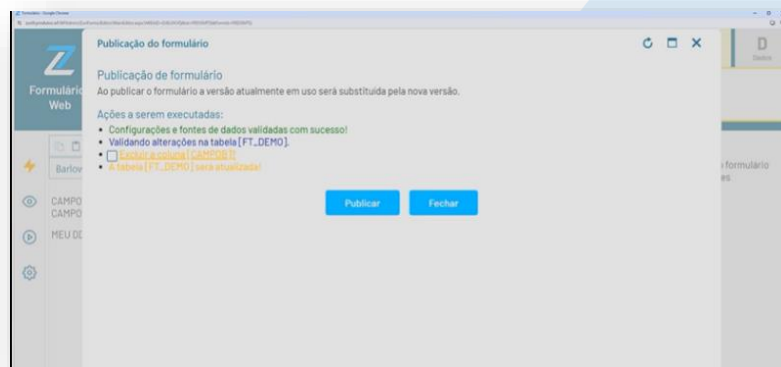


Figura 9 – Validação de tabelas do formulário

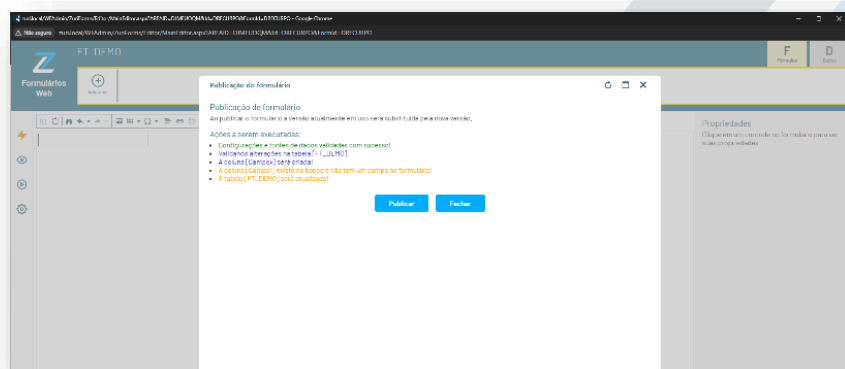


Figura 10 – Confirmação de publicação de formulário

- Cada criação de um novo formulário, haverá um prefixo pré-estabelecido que será o FT_ e ele será o mesmo para todos. Não poderá ser excluído, porém pode ser alterado utilizando a Tag de configuração Zuri.Forms.TablesPrefix no web.config.
- Possibilidade de criação de formulário sem campo, somente com form filho ou form grid. Nesse caso, o formulário pai tem somente os campos Hidden.
- Possibilidade de alterar o tamanho do campo para maior. Caso o usuário tente realizar a diminuição do campo, por exemplo, digitar 90 ao invés do 100 padrão, ao gerar o formulário aparecerá um aviso na tela sinalizando que não é possível realizar a ação. Observação: para realizar a alteração do campo o usuário deve entrar na validação das configurações no campo "máximo caracteres".

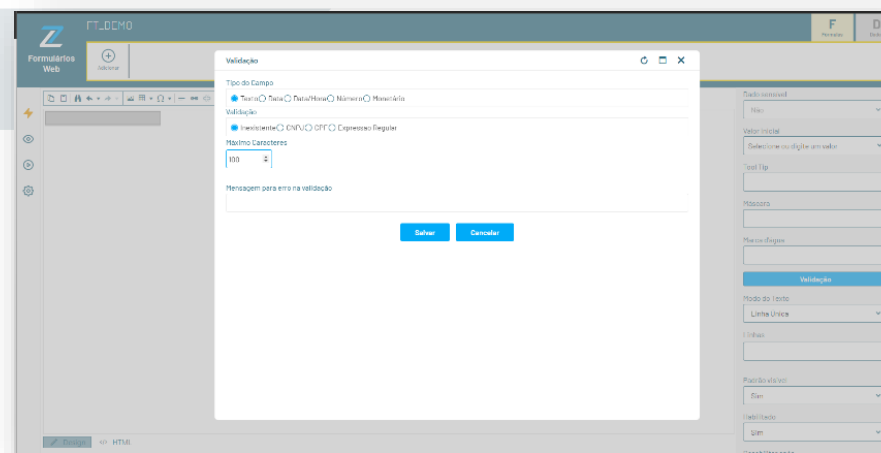


Figura 11 – Alterando o tamanho do campo

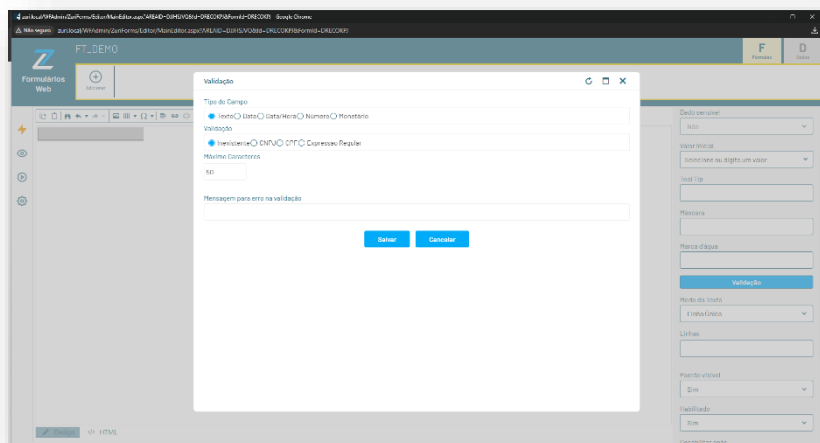


Figura 12 – Alterando o tipo de campo

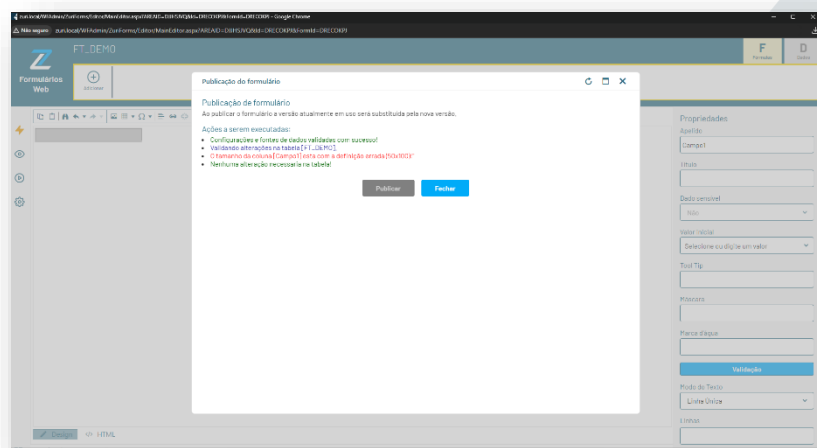


Figura 12 – Cenário de erro de definição de tamanho do campo.

6 – Melhorias na configuração dos campos (geral):

6.1 – Valor inicial para campos.

Possibilidade de utilização de valor inicial (default) para os campos: **TextBox**, **Label**, **Dropdownlist**, **Radiobutton**, **Checkboxlist**, **ListasConectadas**, **HTML**, **Rating**. Para a definição do valor inicial pode ser usado um valor fixo, variável de contexto ou fonte de dados.

6.2 – Propriedade título para todos os campos.

Inclusão da propriedade título para todos os campos, deixando de ser no HTML. Ela servirá para identificar campo (API) e para vir como sugestão para relatórios.

6.3 – Troca tipo de campo.

Possibilidade de troca de tipo de campo sem necessidade de exclusão após troca de apelido

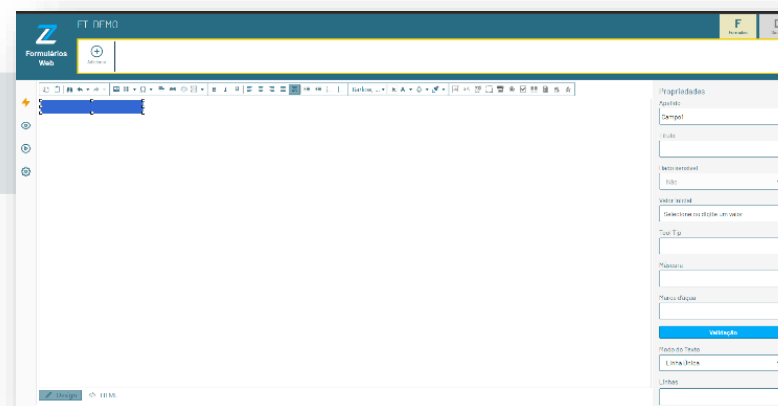


Figura 13 – Seleção de valor Default para campos

7 - Expressões

- Expressões com textos são comparadas de forma case insensitive sempre.
- As expressões só são executadas quando todos os seus parâmetros têm valor preenchido.

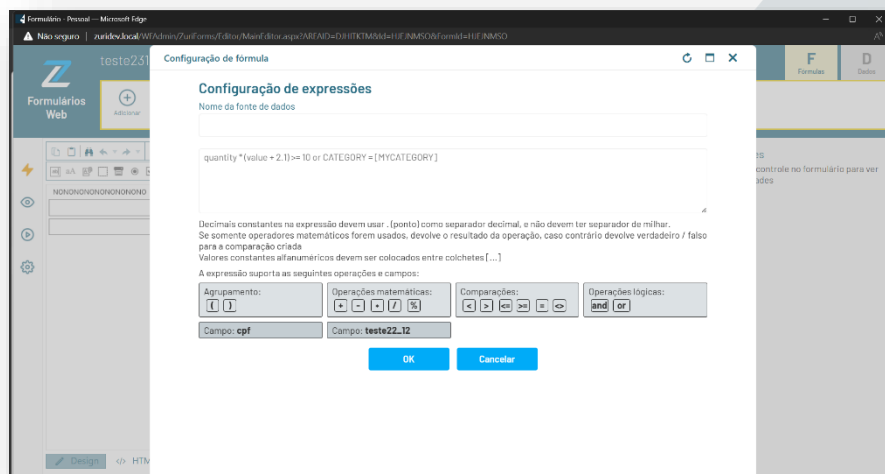


Figura 14 – Configuração de Expressões em fórmulas.

8 - Melhorias no Form Filho

8.1 – Substituição do form grid pelo form filho

Form grid foi substituído pelo form filho após a migração. E o form filho agora possui a abertura como grid, juntamente com o funcionamento de todas as suas **funcionalidades**.

8.2- Seleção de filhos de formulário

- Seleção dos filhos somente de formulários pertencente a mesma área. Observação: na migração mantém se houver um externo, mas não pode ser alterado.
- Alteração das ordens de colunas por meio de drag-drop

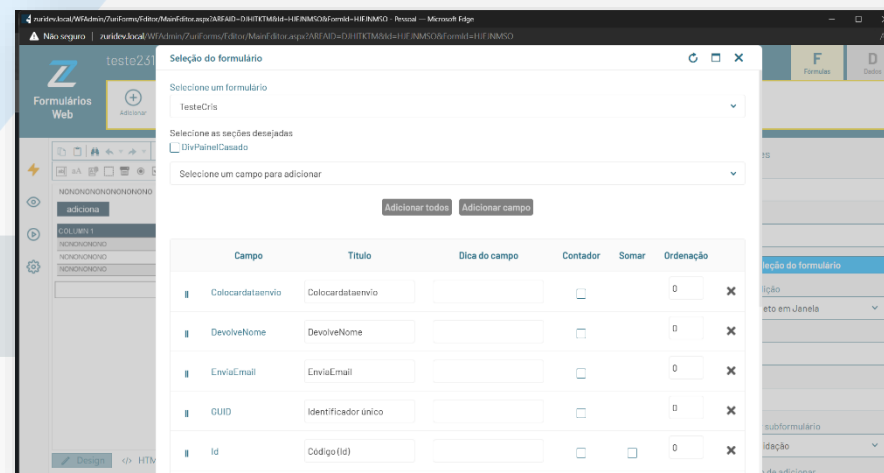


Figura 15 – Seleção de formulário filho

8.3 – Abertura de form filho

Funcionamento de todas as funcionalidades do form filho no formato grid

- O form filho, se tiver título, apresenta um fieldset em torno da área de funcionamento
- Form filho pode ser chamado de quatro formas, sendo elas:

➤ **Popup** (passam a ser exibidos em uma janela modal)

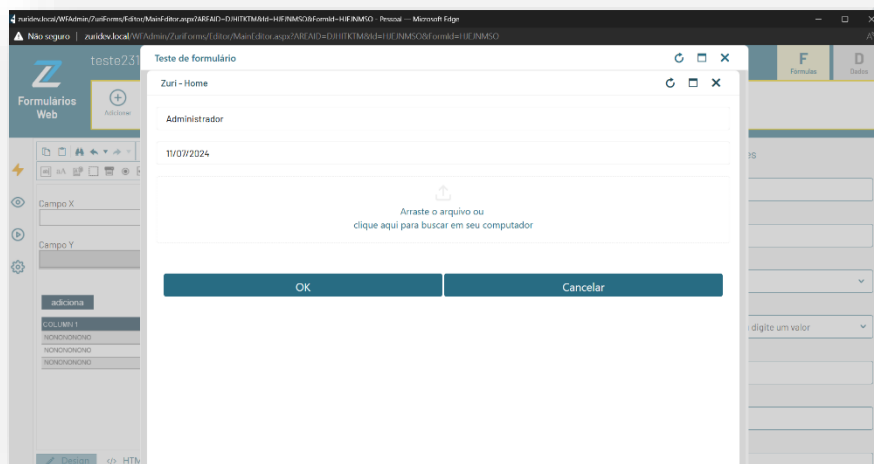


Figura 16 – Pop Up

➤ **Inline no topo da listagem** (são executados no mesmo frame do formulário.)

➤ **Inline sobre a listagem** (são executados no mesmo frame do formulário.)

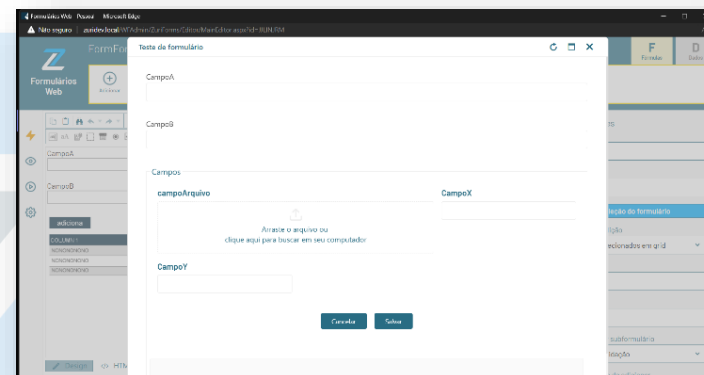


Figura 17 – Inline no topo da listagem

➤ **Grid** (são executados no mesmo frame do formulário.)

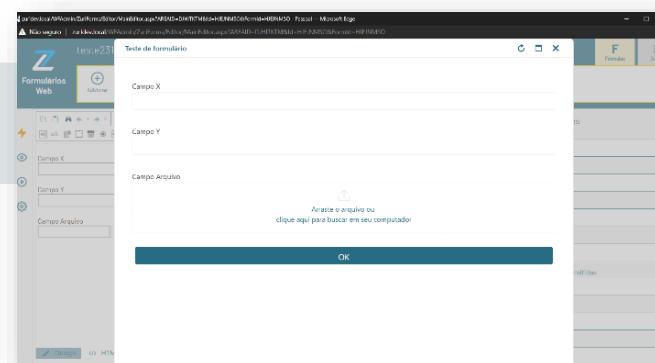


Figura 18 – Form Grid

- Formulários chamados como filho passam a apresentar um botão de cancelar sua execução.

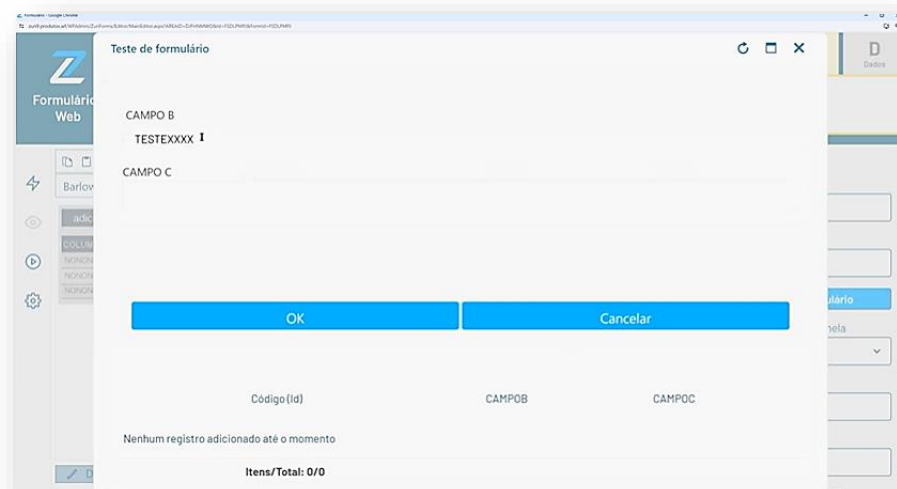


Figura 19 – Formfilho

- Ao selecionar “não” na opção de “Abrir o filho na mesma janela” os formulários que antes abriam em popup passam a ser exibidos em uma janela modal.
- Formulários inline são executados no mesmo frame do formulário principal.
- Para “formulários filho” (não grid) podem ser selecionadas somente um ou mais DIVs para o preenchimento.

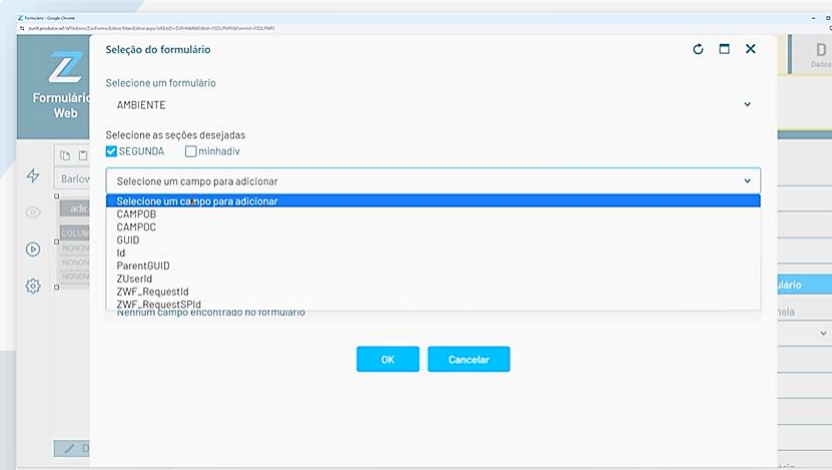


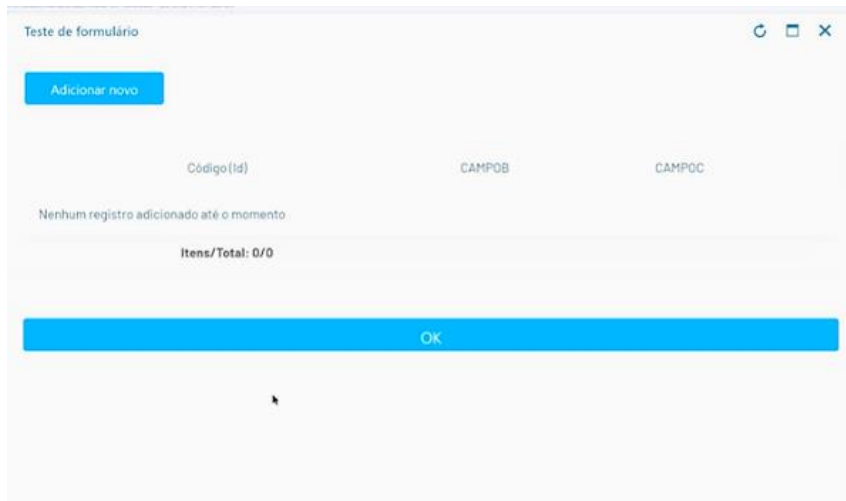
Figura 20 – Seleção de Divs em formulário filho

8.4 – Novas ações para abrir um filho

Agora temos novas possibilidades para abrir um form filho, sendo elas:

- **Não salvar**
- **Salvar com validação**
- **Salvar sem validação**

- Pode ser definido o texto do botão de adicionar



Teste de formulário

Adicionar novo

Código (id) CAMPOB CAMPOC

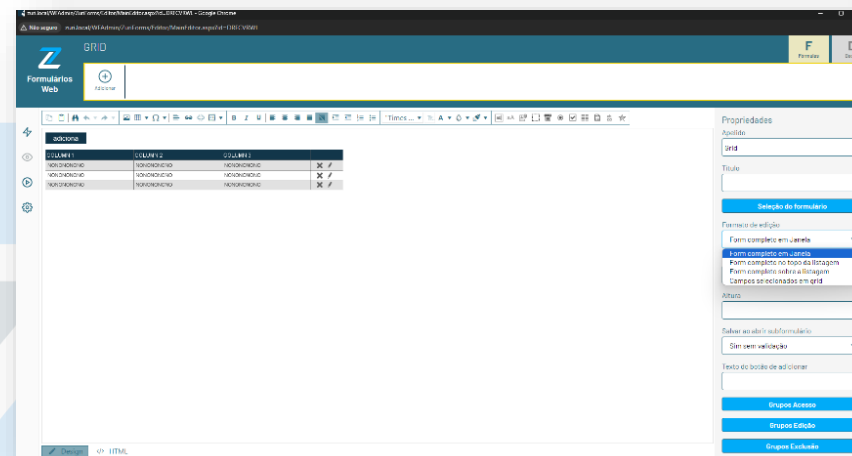
Nenhum registro adicionado até o momento

Itens/Total: 0/0

OK

Figura 21 – Definição de texto do botão

- O controle de acesso ao campo passa a controlar os itens exibidos no grid
- Form filho no formato GRID utilizam o título dos campos para nomear as células



GRD

Formulários Web

Selecione

Selecione	CAMPOB	CAMPOC	Selecione
Selecione	Selecione	Selecione	X
Selecione	Selecione	Selecione	X
Selecione	Selecione	Selecione	X

Propriedades

Apelido

grid

Título

Selecione de formulário

Formulário de edição

Form completo em janela

Form completo no topo da tela

Form completo sobre a tela

Campos selecionados em grid

Ativar

Salvar ao alterar subformulário

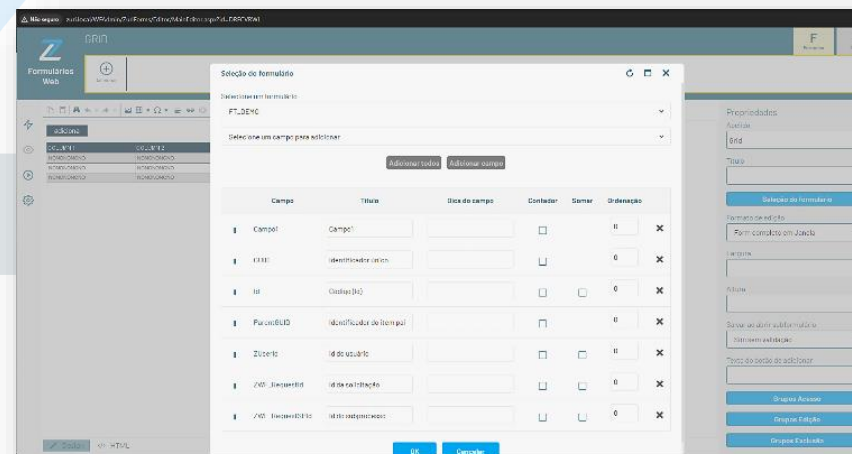
Sim sem validação

Texto do botão de adicionar

Grupos Aceitos

Grupos Excluídos

Figura 22 – Controle de itens exibidos no grid



Seleção de formulário

Formulário de edição

Form completo em janela

Form completo no topo da tela

Form completo sobre a tela

Campos selecionados em grid

Ativar

Salvar ao alterar subformulário

Sim sem validação

Texto do botão de adicionar

Grupos Aceitos

Grupos Excluídos

Figura 23 – Controle de itens exibidos no grid

9.0 – Melhorias no campo DIV

9.1 – Possibilidade de configuração de modo de exibição

Os campos DIV (seção) passa a ter três possibilidades de configuração de modo de exibição, sendo elas:

- **Fieldset** (selecionando a opção “emoldurado”);
- **Hidden** (selecionando a opção “seção escondida”. Nesse caso passa a não aparecer em listagens);
- **Condicional** (possibilidade de inserção de uma fonte de dados com valores para true/false. Exemplo: se um datasource retornar **TRUE** a **DIV** será mostrada, caso ele retorne FALSE a DIV não será mostrada)

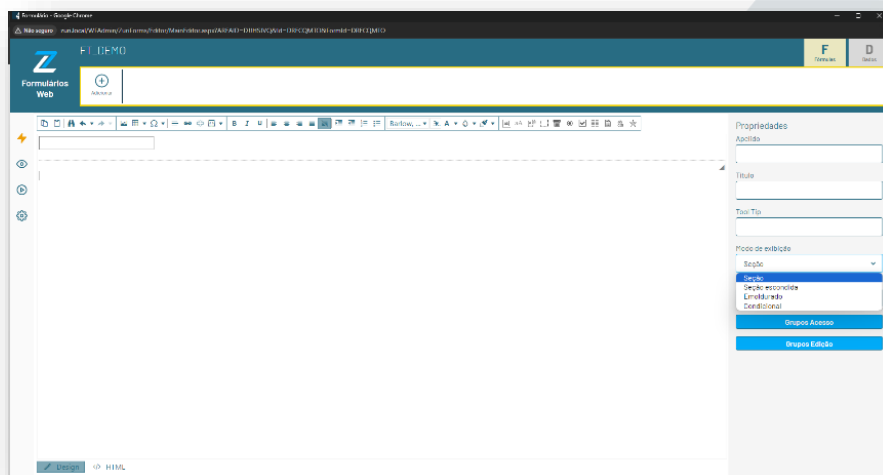


Figura 24 – Condicional “modo de exibição”

9.2 – Controle de arquivo

No controle de arquivo se for definido um filename sem extensão, o arquivo tem o seu nome mudado e a extensão original preservada ;

9.3 – Campos tipo texto com Autocomplete

Campos do tipo texto com **AutoComplete** serão substituídos por **dropdownlists** com autocomplete na migração.

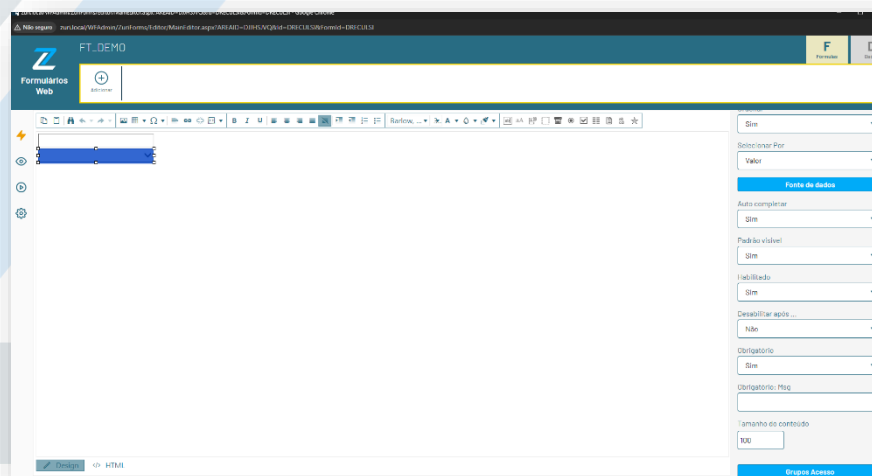


Figura 25 – Campos do tipo texto com AutoComplete

10 – Melhorias na execução do formulário

10.1 – Alteração do texto botão “ok”

Possibilidade de alteração do texto do botão “OK”.

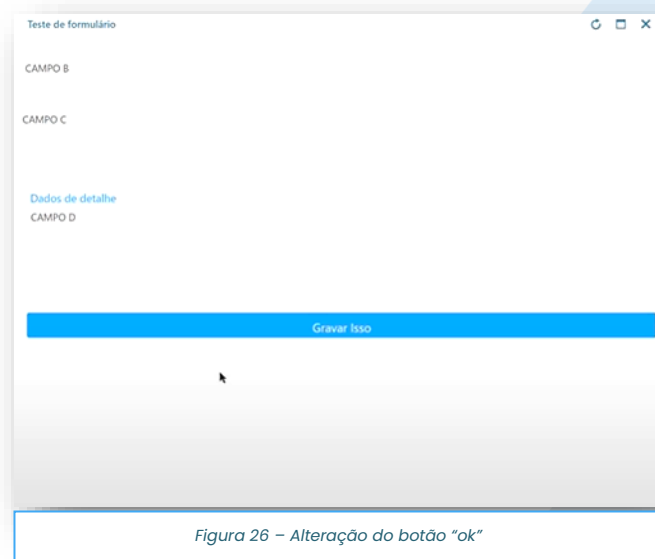


Figura 26 – Alteração do botão “ok”

10.2 – Preenchimento parcial do formulário

Possibilidade de permitir preenchimento parcial do formulário, com possibilidade de alteração do texto do botão “Salvar parcialmente”.

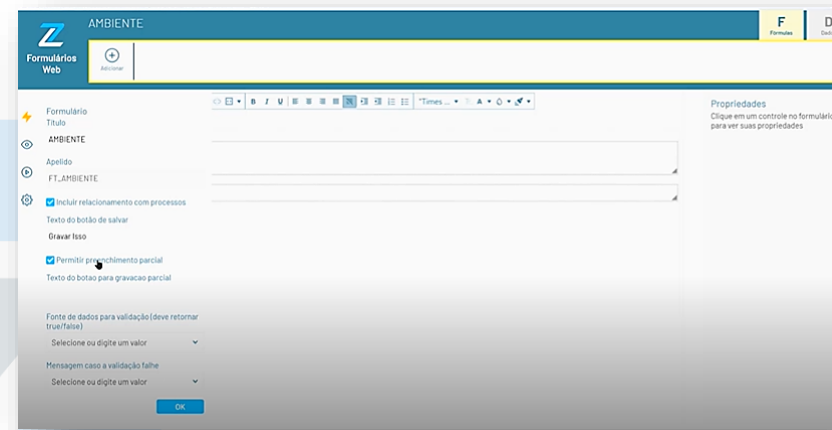


Figura 27 – preenchimento parcial do formulário

- No formulário anterior, na criação do formulário, o Id, **UserId**, **GUID** e **ParentGuid** ficavam somente na memória, agora eles existem dentro do formulário como campo e é possível obter esse valor com um **JavaScript**, pois possui um campo **Hidden** dentro do formulário com esse valor

10.3 – Controle de eventos e dependência para evitar loops infinitos.

A execução das fontes de dados ganhou mais inteligência e eficiência na nova versão e ela impede que referências circulares (uma fonte de dados dependendo de outra que depende dela) acabem fazendo o formulário parar de funcionar ou até comprometendo o ambiente. Quando houver este tipo de situação

ele para a execução da fonte de dados e grava uma mensagem de log para que a configuração seja ajustada.

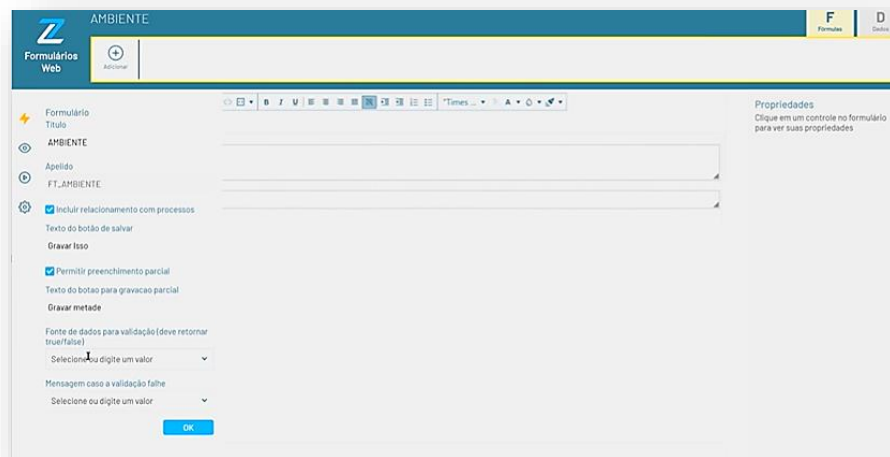


Figura 28 – Execução de datasource

10.4 – Validação do formulário por datasource antes da gravação de dados

Possibilidade de execução de um datasource para validação (true/false) que define se haverá ou não a conclusão do formulário. Caso o retorno seja o "false" retornará a mensagem de erro e não irá salvar. A mensagem de erro pode ser alterada.

10.5 – Validação de obrigatoriedade

Esta validação agora funciona para todos os tipos de campo.

10.6 – Campos de funcionamento como hidden

As informações de campos de funcionamento (ID, UserId, GUID,...) passam a ser disponibilizadas em campos hidden no formulário durante a sua execução. Para que isso ocorra, é necessário clicar a opção "Executar depois de gravar os dados no banco".

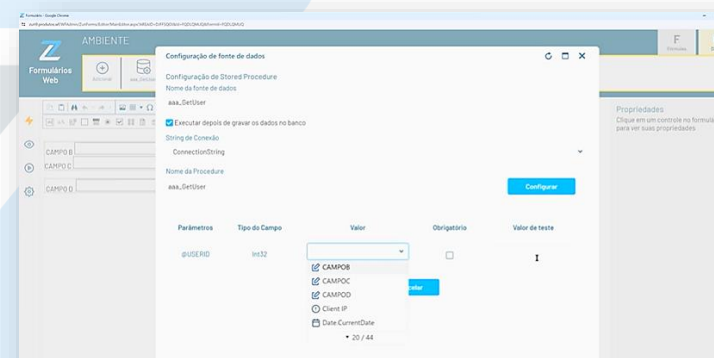


Figura 29 – campos de funcionamento (ID, UserId, GUID,)

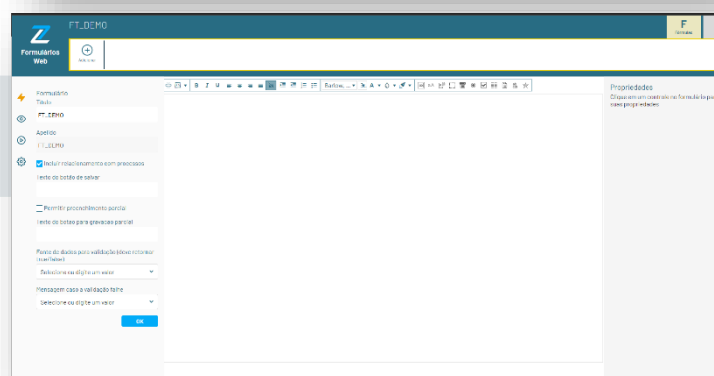


Figura 30 – campos de funcionamento (ID, UserId, GUID,)

11 - Melhorias na fonte de dados

Melhoria na arquitetura com isolamento no funcionamento e configuração e declaração por MEF

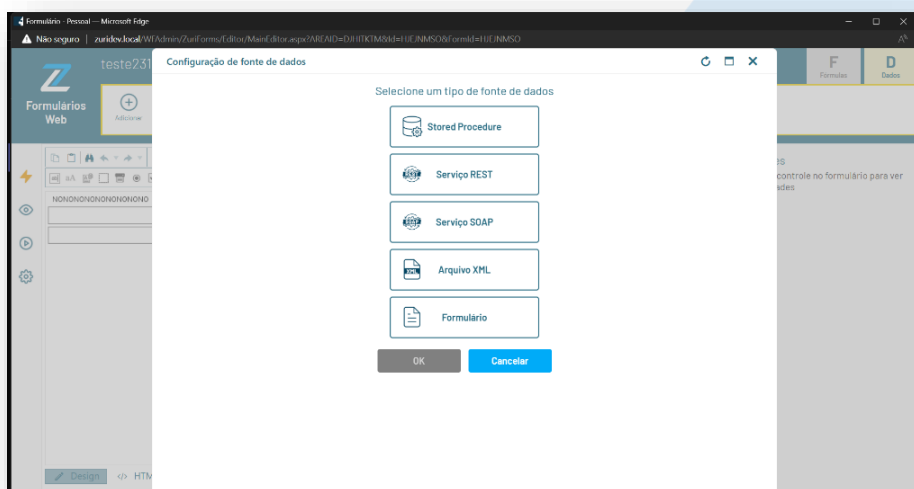


Figura 31 – Fonte de dados

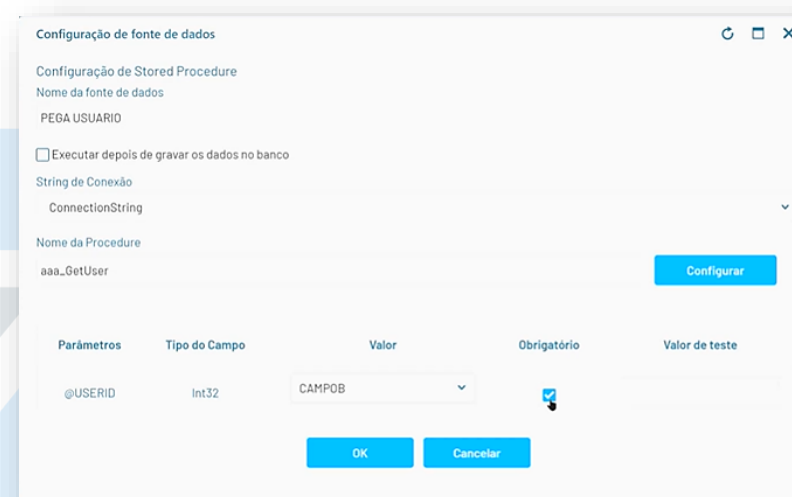


Figura 32 – Configurando Fonte de dados

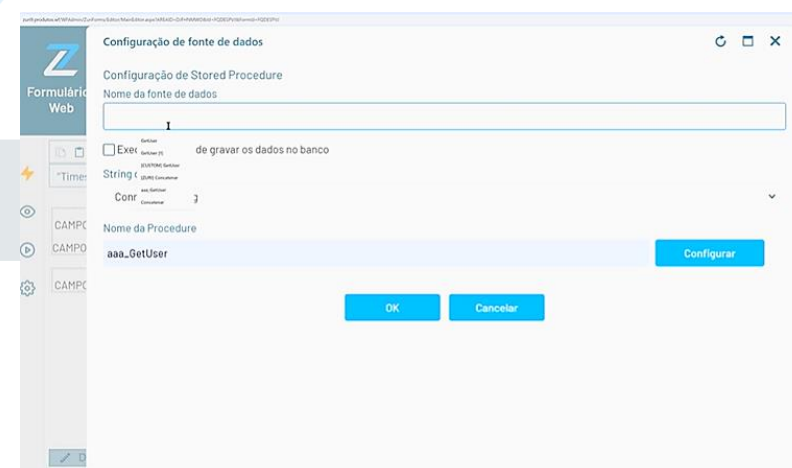


Figura 33– Configurando Fonte de dados “inserindo nome”.

11.1 – Execução da fonte de dados

As fontes de dados (Fórmulas, Procedure, REST ou SOAP) podem ser definidas para executar somente se todos os campos que estiverem configurados como obrigatórios tiverem valor.

11.2 – Inserção de nome da fonte de dados

Possibilidade de inserção de nome da fonte de dados dentro de configurações.

11.3 – Configuração do Cache

Todas as fontes de dados passam a contar com a configuração de Cache. Para realizar a configuração do tempo, basta utilizar a variável **Zuri.Forms.FormDataSourceCacheSeconds**.

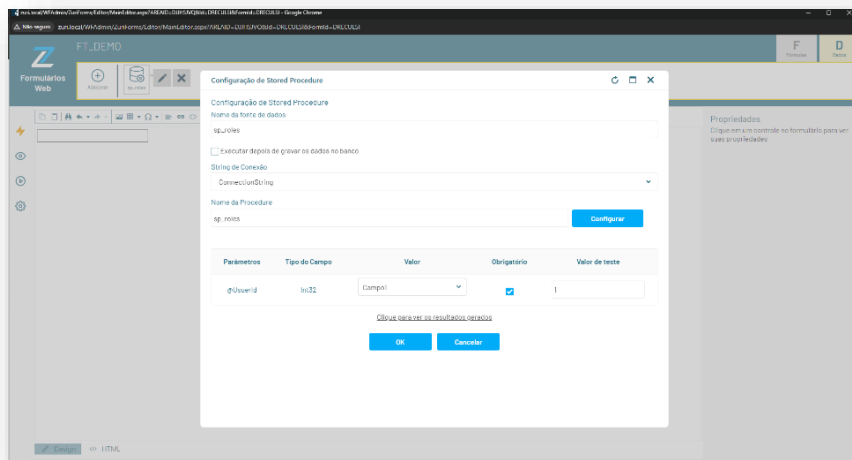


Figura 34– Configurando fonte de dados – Valor de controle

11.4 – Preenchimento do Form filho como gatilho

O preenchimento de um filho agora pode ser utilizado como gatilho para uma fonte de dados passando o número de filhos como o valor do controle.

12 – Melhorias nas fórmulas

12.1 – Carregamento por MEF

As fórmulas passam a ser carregadas por MEF, ou seja, possuem uma execução mais rápida, pois já estão compiladas dentro da DLL.

12.2 – Novas fórmulas:

- **CopiaValor** (para realizar cópia de um valor de um campo para outro). Essa fórmula é mais eficiente que a fórmula de concatenar.
- **Valor mínimo**
Seleciona o menor valor entre os parâmetros passados

12.3 – Adição de novas fórmulas

Novas fórmulas podem ser adicionadas inserindo classes baseadas em **Zuri.Forms.Providers.FormulaProvider**

13 – Módulo Forms no portal

13.1 – Múltiplas ações ao preencher o formulário

O módulo de forms no portal agora suporta múltiplas ações ao ser preenchido o formulário, estas ações são implementações da classe **Zuri.Forms.Providers.ActionProvider** e podem ser:

- **Texto de encerramento (texto em HTML)**

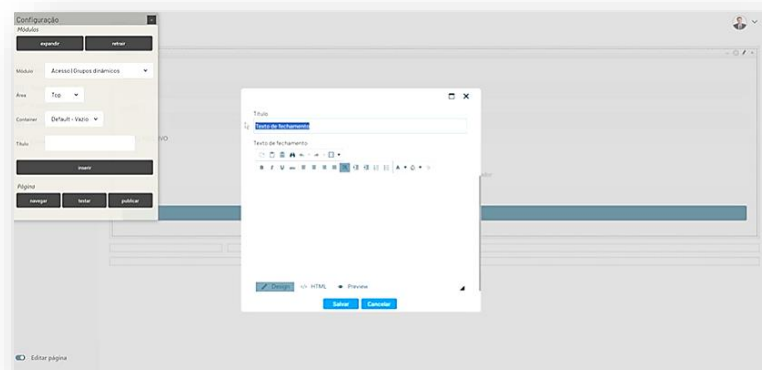


Figura 35– Texto encerramento (HTML)

- **Redirecionamento para outra página**

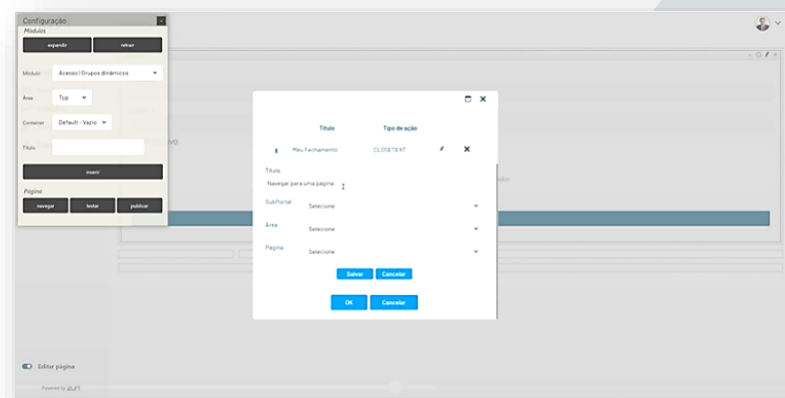


Figura 36– Redirecionamento para outra página.

- **Chamada de processo**
- **Envio por email do PDF**

A ordem da execução das ações é feita de acordo com a sua configuração. A cada configuração é preciso publicar a página para que ela seja realizada.

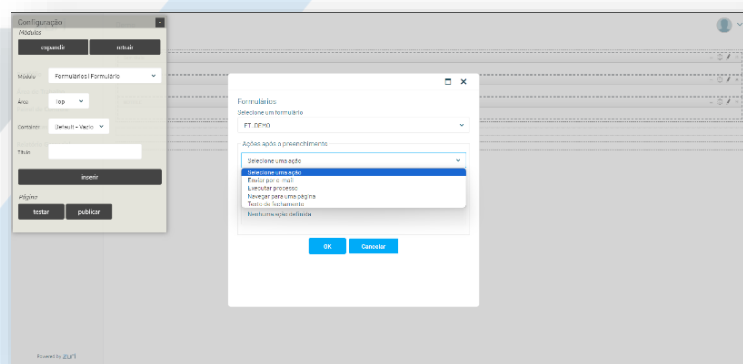


Figura 37– Ações pós preenchimento.

13.2 – Configuração de Seções

A partir desta versão é possível configurar uma ou mais seções para a apresentação ao abrir um formulário na página. Se caso nenhuma ação for definida ou se caso todas as ações não possuírem interface, aparecerá somente uma mensagem de formulário preenchido com sucesso.

A partir desta versão é possível configurar uma ou mais seções para a apresentação ao abrir um formulário na página.



Figura 38- Configuração de Seção

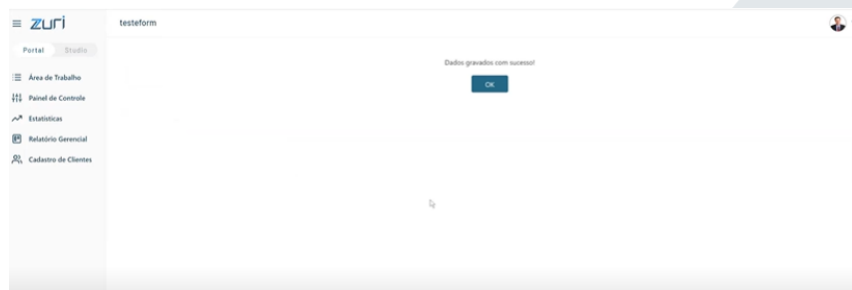


Figura 39- Confirmação de preenchimento.

13.3 Chamadas de integração com Workflow

Ao chamar uma página com o módulo de forms pode ser passado como parâmetro na query string o id do registro que deve ser visualizado. O nome do item na querystring é **FORMITEMID** e deve conter o ID do item com o EncNumber. Se o id for passado com o valor negativo ($-1 * ID$) então o formulário será apresentado desabilitado.

- Para chamadas de integração com o Workflow pode ser gerado o link passado o item do itemworkflow usando o comando **ITEMWORKFLOWID** (com o valor encriptado pela EncNumber) ou então pode ser passado o Id da Etapaltem, passando o comando **STEPITEMID** (com o valor encriptado pela EncNumber)



14 - Módulo Reports no Portal

14.1 - Melhoria na configuração

Melhorias no processo de configuração (eliminação das configurações por metadados.)

14.2 - Opção de filtro

Possibilidade de escolha de somente uma seção para ser preenchida ou visualizada.

> A sugestão de título dos campos já vem preenchida com base no que foi preenchido no formulário

• **Opção de filtro simples, faixa e seletor.**

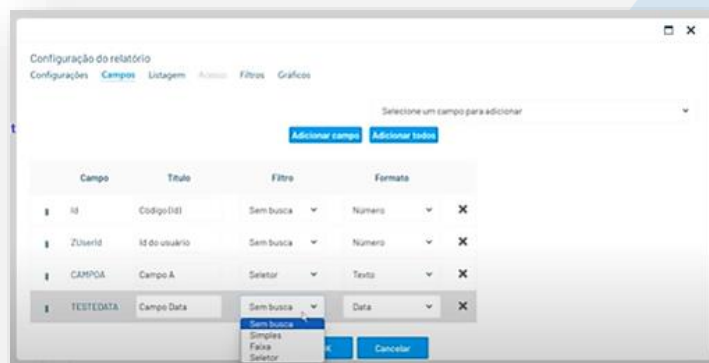


Figura 40- Configuração do relatório no portal

14.3 – Formatação Decimal

Formatação com a inserção da opção de decimal com precisão considerando 4 casas decimal.

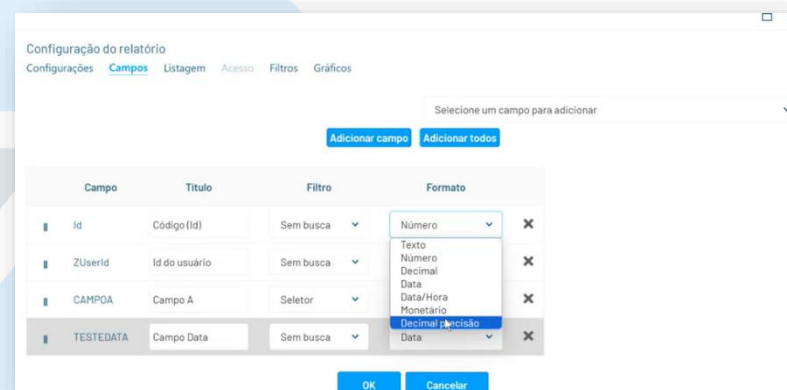


Figura 43- Formatação de decimal

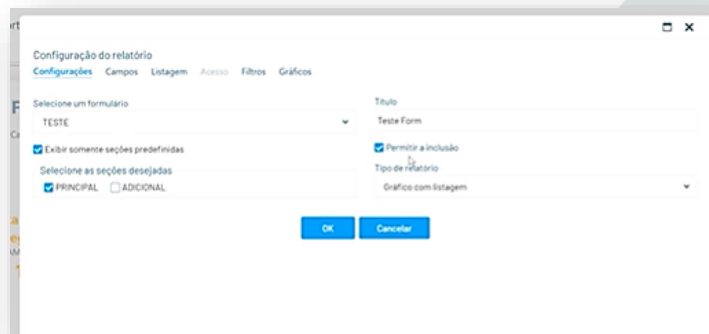


Figura 41- Seleção de filtro simples

14.4 – Ordenação de campos

Novos modos de geração da listagem:

- Somente Listagem
- Listagem com edição em popup
- Listagem com edição Inline
- Listagem com edição no grid
- Gráficos sem drill down
- Gráficos com drill Down

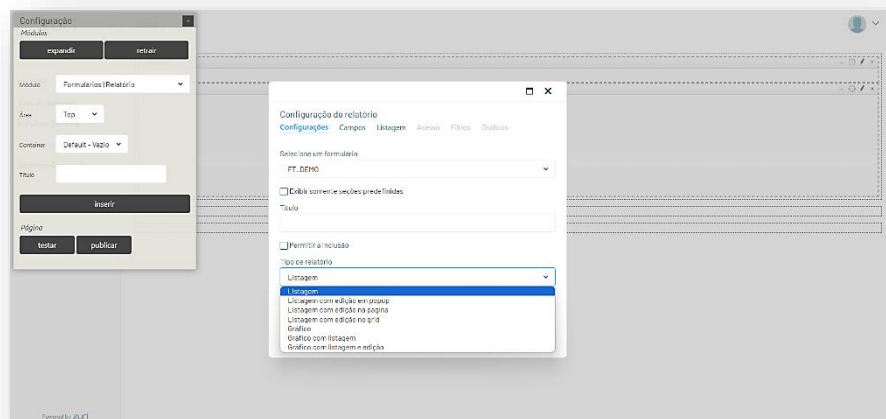


Figura 44- Geração de Listagem

14.5 – Definição de Grupos

Definição de grupos distintos para Inclusão, Edição e Exclusão de registros.

- Possibilidade de utilização das variáveis e funções de contexto nos pré filtros.
- Possibilidade de inclusão de 1 ou mais gráficos no relatório facilitando a criação de dashboards.
- Aplicação de CSS para cada gráfico.

14.6 – Inclusão de Gráficos

Inclusão dos seguintes tipos de gráficos:

- Tabela: que apresenta o resumo dos valores agrupados
- Valor: que apresenta um valor único (soma, contador, média, ...) obtido pelo agrupamento.

- Melhoria no esquema de configuração dos campos para permitir reordenar através de drag e drop e separação da configuração de ordenação, com possibilidade de ordenação Ascendente ou descendente.

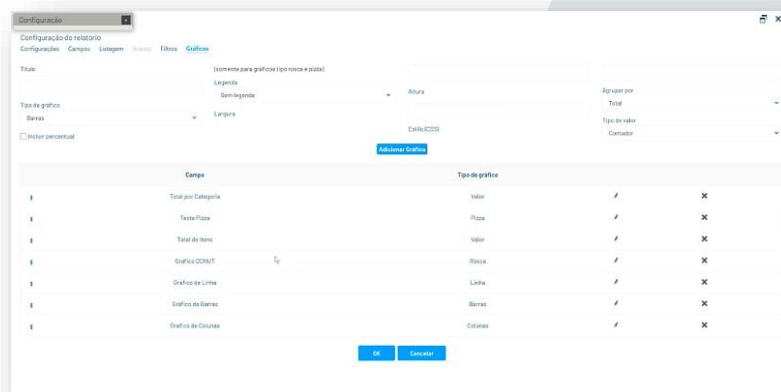


Figura 45- Ordenação

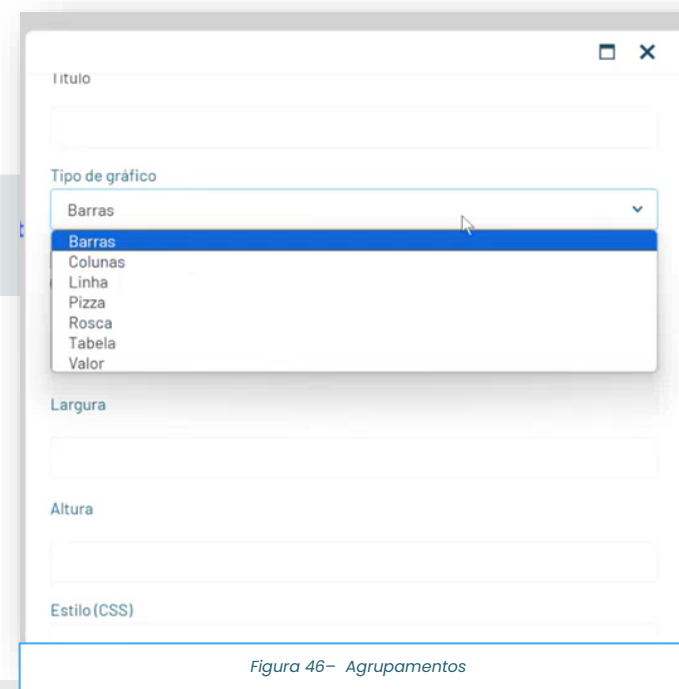


Figura 46- Agrupamentos



15 – Módulo Forms no Workflow

15.1 – Substituição de módulos antigos

Os módulos Formulário e FormFiller foram substituídos pelo Módulo ZuriFormFill que contém as funcionalidades dos dois módulos combinadas, inclusive permitindo a execução simultânea das duas funcionalidades.

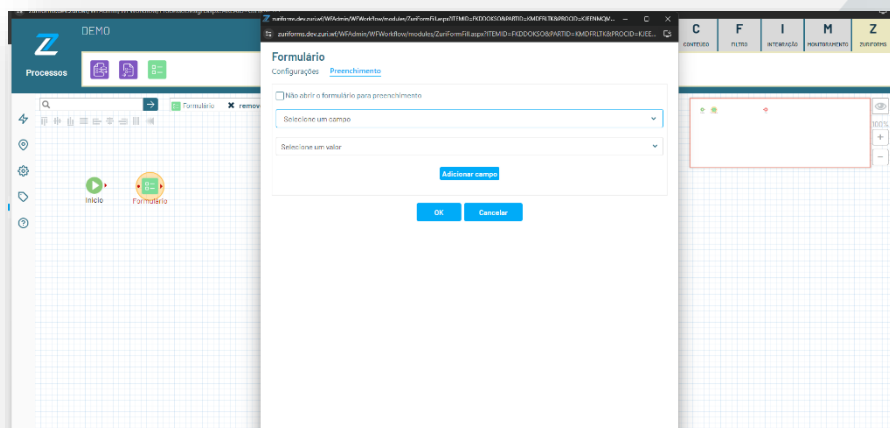


Figura 47– Preenchimento

15.2 – Preenchimento parcial do formulário

O módulo de preenchimento de formulário não tem mais o conceito de form múltiplo, este conceito foi alterado pelas opções

de seleção de registro e conjunto com a possibilidade de usar um nome de anexo diferente.

- O módulo permite trocar o apelido do anexo gerado possibilitando a criação de vários anexos diferentes, mas com os mesmos campos.

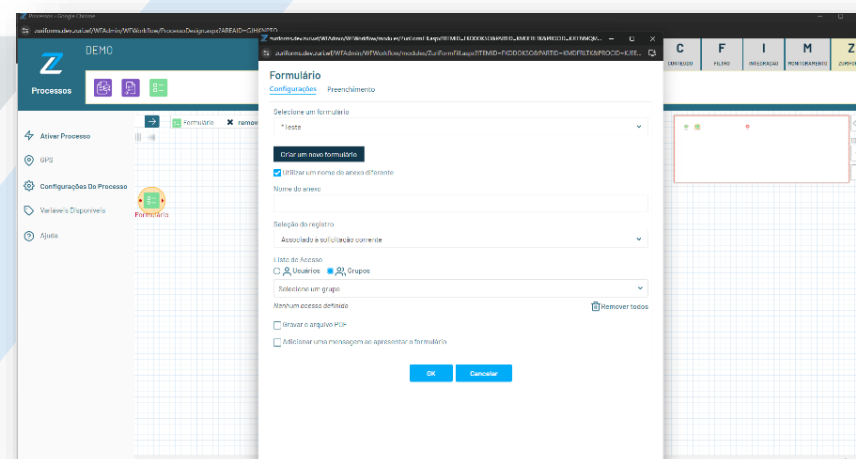


Figura 48– Configurações “criação de anexos”.

15.3 – Módulo Forms no Workflow

O módulo permite a definição da forma de seleção do registro entre as opções :

- Associar à solicitação corrente: o mesmo registro é sempre mantido para a mesma solicitação.

- Sempre criar vazio: sempre vem com o formulário vazio com o apelido sendo o mesmo. Todas as vezes que tiver tentativa de acessar esse registro, ele trará o último que foi preenchido, criando um registro novo na base de formulário.
- Associado a outro Id de solicitação: Permite trazer o formulário preenchido em um outro item passando o ItemWorkflowId.

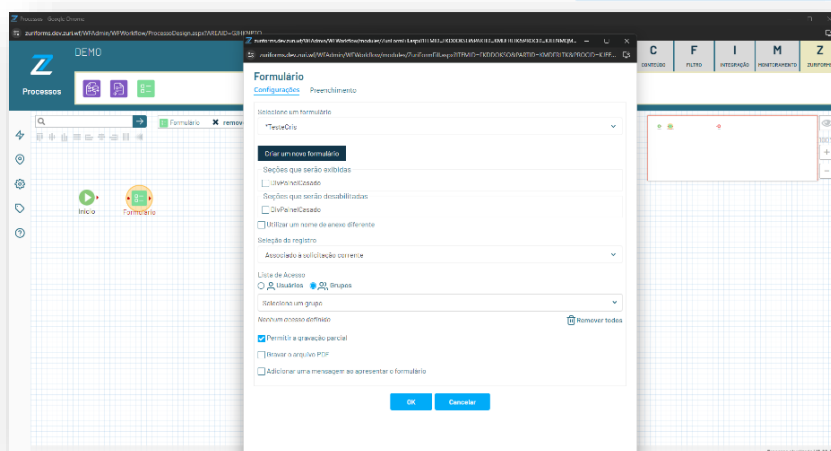


Figura 49- Configurações – “Seleção de registro”

- Selecionar o Id do formulário: Neste caso é recuperado um registro existente na base do formulário para o preenchimento (FormItemId).
- O módulo permite o preenchimento parcial do formulário (se estiver com essa funcionalidade habilitada) e neste caso cria uma saída adicional para o formulário no processo.

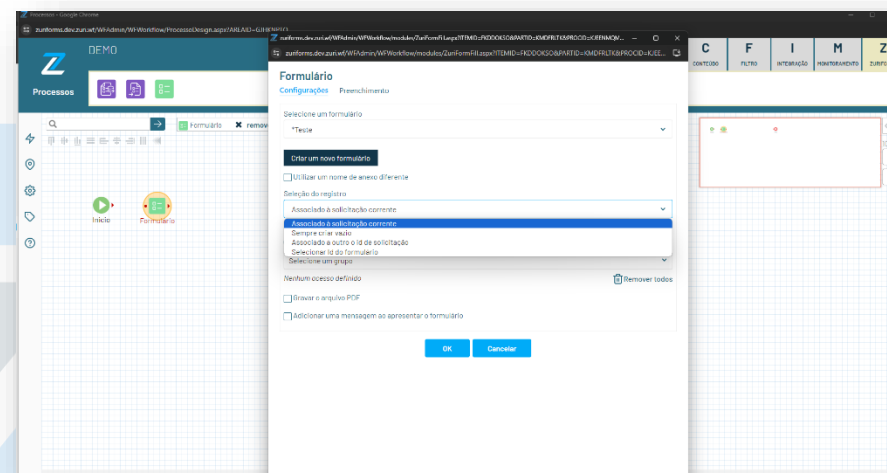


Figura 50- Configurações – “preenchimento parcial”

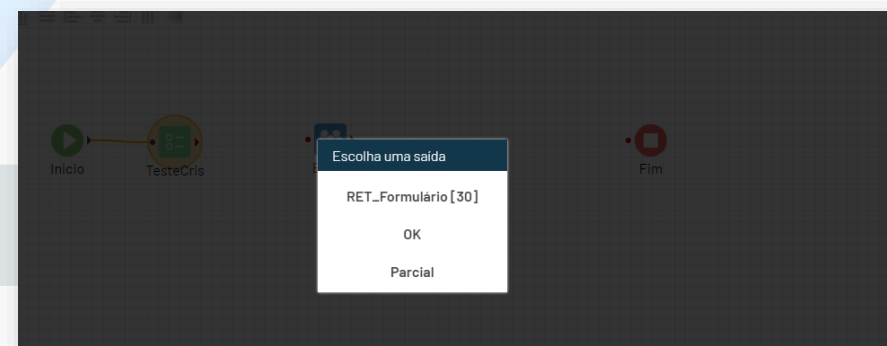


Figura 51- Configurações – “preenchimento parcial”

15.4 – Quando o formulário tem arquivos eles passam a ser listados como arquivos nos anexos.

- A documentação do formulário passa a mostrar nas dependências todos os nomes de fontes de dados externas consumidas pelo formulário, bem como seus filhos e as fontes de dados dos filhos.



16 – Módulo Formdata no Workflow

16.1 – Seleção de registros

O novo módulo FormData substitui os antigos módulos de Anexa Formulário, Anexa SubFormulário e Exporta JSON

O módulo permite a seleção de:

- Um ou mais registros do formulário e gera um anexo no formato JSON,
- Um anexo com um registro único
- Um anexo com uma coleção (resultado da busca)

16.2 – Geração de anexo

O anexo gerado conta também com um campo que devolve a quantidade de registros encontrados.



17 – Módulo SelectForm no Workflow

O módulo SelectForm foi substituído pelo módulo ZuriFormSelect. A funcionalidade foi mantida idêntica.

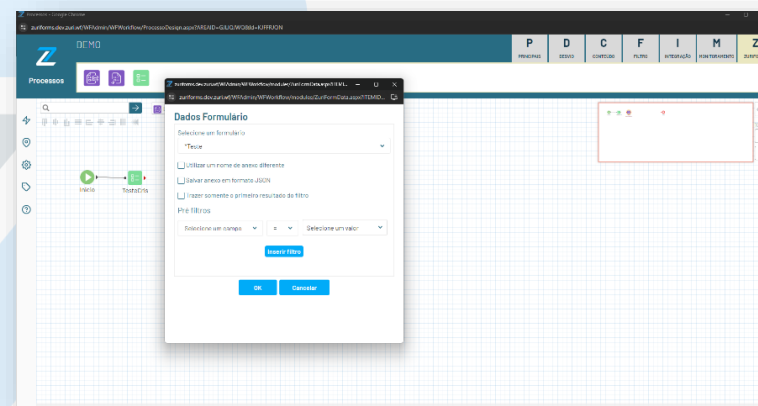


Figura 52 – Configurações – “Dados do formulário”

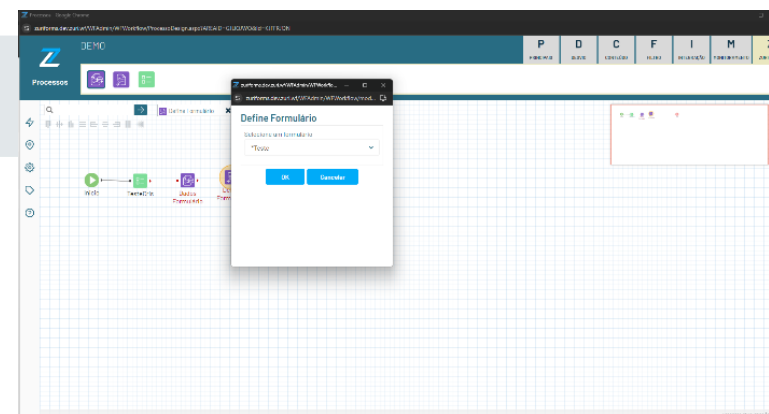


Figura 53 – Define Formulário “select form”.



18 – Formulário Anônimo

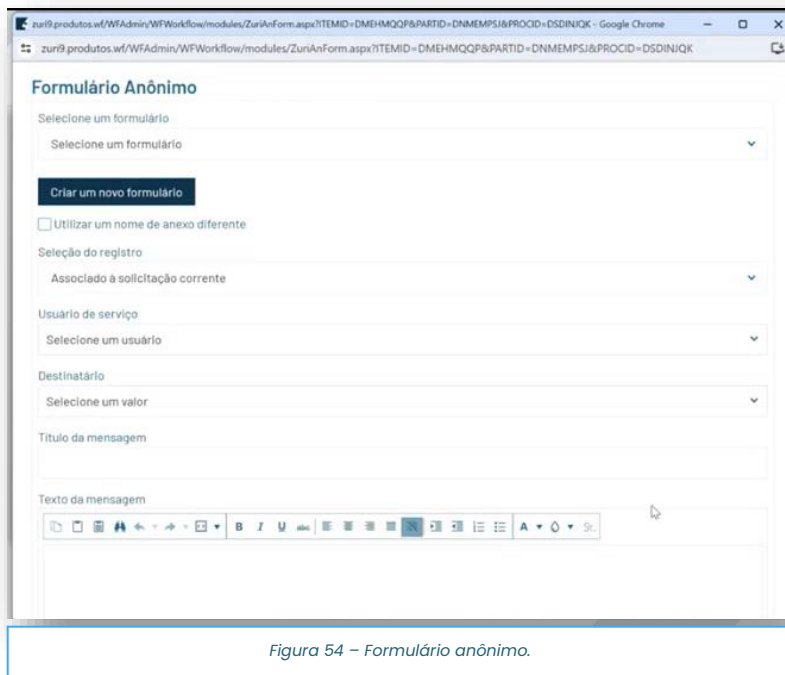


Figura 54 – Formulário anônimo.

- Para inserir o link de preenchimento do formulário, basta selecionar a opção “link para preenchimento”. O texto pode ser editável.

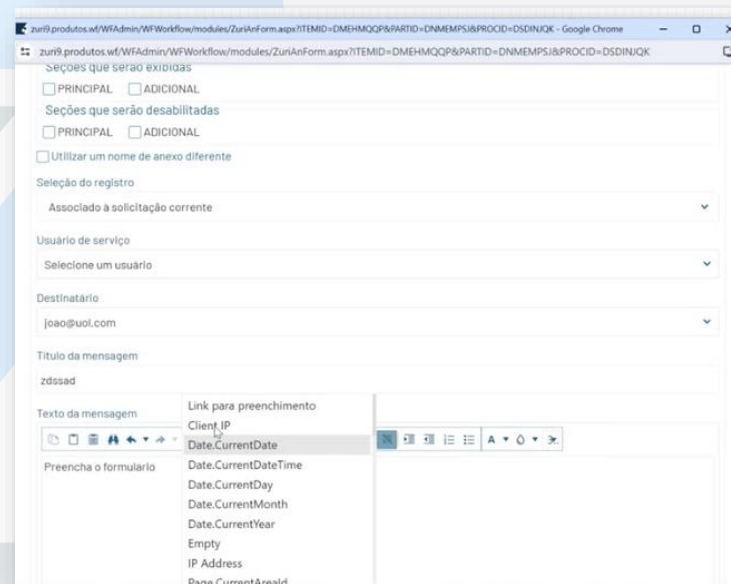


Figura 55 – Link de Preenchimento

- Não há mais a necessidade de escolher a atividade. O código irá olhar para o processo e irá achar qualquer atividade que tenha esse formulário e irá associar esse formulário aquela atividade.
- Nas configurações do formulário anônimo, se o formulário escolhido tiver DIV, é possível escolher quais seções serão exibidas.



Em caso de dúvidas, entre em contato:

suporte@gozuri.com