

## SISTEMA DE CONTROLE DE DESPESAS

Vitor Pellisoli da Silva

Silvio Cesar Viegas

### RESUMO

O presente artigo tem como objetivo principal demonstrar os estudos necessários e os passos para a construção de um aplicativo em OutSystems. Este estudo também apresenta um pouco sobre a evolução do planejamento financeiro pessoal e familiar nos últimos anos. No decorrer deste trabalho é descrito como foi planejado e desenvolvido a aplicação para o controle de despesas e as tecnologias utilizadas, com a intenção de oferecer praticidade aos usuários da ferramenta, através do auxílio da tecnologia.

**Palavras-chave:** OutSystems, Controle de Despesas, Planejamento Financeiro.

### 1. INTRODUÇÃO

No Brasil, a hiperinflação das décadas de 1980 e 1990 criou um cenário de instabilidade econômica, impedindo que os brasileiros desenvolvessem hábitos de planejamento financeiro. Com a estabilização econômica trazida pelo Plano Real em 1994, o consumo aumentou, mas a falta de educação financeira levou ao endividamento da população. Apesar da estabilização, o controle financeiro ainda era um desafio, pois famílias, escolas e governo não ensinavam educação financeira. Gradualmente, o planejamento financeiro pessoal e familiar ganhou espaço, mas dados recentes do Serasa apontam para um aumento da inadimplência, com 72,89 milhões de brasileiros em situação de dívida em março de 2024. Diante desses desafios, este texto apresenta uma solução: a planilha de orçamento familiar. Essa ferramenta ajuda na organização das despesas e permite uma visão mais clara sobre o destino do dinheiro, auxiliando na prevenção de dívidas e no planejamento de investimentos a longo prazo.

## 1.1 Tema

Desenvolvimento de um Sistema de Controle de Despesas

## 1.2 Delimitação do tema

No primeiro momento, o projeto possui foco no desenvolvimento de um sistema básico, com as demais funções sendo incrementadas com o tempo ou conforme a necessidade de futuros usuários.

## 1.3 Problema

O número de famílias endividadas no Brasil cresce em ritmo rápido, segundo a Pesquisa de Endividamento e Inadimplência do Consumidor (PEIC) e da Federação do Comércio de Bens, Serviços e Turismo do Estado de São Paulo (FecomercioSP) realizada em 2021, só na capital, 2,98 milhões de lares possuíam algum tipo de dívida, um número recorde até então. Outra pesquisa mais recente, mostra que a principal modalidade de dívida das famílias é o cartão de crédito (86,9%), seguida por carnês (15,8%), crédito pessoal (9,9%), financiamento da casa (8,7%) e do carro (8,6%) e crédito consignado (6%).

## 1.4 Justificativa

Observando os números apresentados, é notório que as pessoas precisam de um melhor planejamento financeiro, e um sistema que possa organizar e atender as demandas necessárias, iriam auxiliar bastante essas pessoas.

## 1.5 Objetivo geral

Desenvolver um sistema de controle de despesas para ajudar as pessoas a terem um controle melhor de suas dívidas e poderem realizar com mais praticidade o próprio planejamento financeiro.

## 1.6 Objetivos específicos

- a) Cadastrar uma despesa com data, descrição, valor e o tipo de recorrência;
- b) Editar os dados caso haja algum erro;
- c) Filtrar as despesas por diária, mensal ou anual;
- d) Mostrar o número atual de parcelas.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nas décadas de 80 e 90, o brasileiro estava acostumado com as elevadas taxas de recorrentes da hiperinflação, onde os preços aumentavam e diminuía quase que diariamente, devido a esse cenário, o brasileiro não criou hábitos de planejamento financeiro. Foi somente com a implantação do Plano Real no Brasil, em 1994, que iniciou-se um processo de estabilização econômica, possibilitando assim que as pessoas passassem a consumir mais, porém devido à falta de hábito de planejar as

finanças pessoais, a população brasileira se endividou, assim as pessoas endividadas, sem dinheiro para cumprir com os seus compromissos passaram a ter diversos problemas.

Mesmo após a estabilização financeira, as pessoas ainda apresentavam dificuldades em realizar seus controles financeiros, pois as famílias, as escolas e o governo não ensinavam as crianças a se educarem financeiramente. Com o tempo, se tornou possível efetuar projeções quanto ao valor do dinheiro no futuro, e assim, aos poucos o planejamento financeiro familiar e pessoal passou a fazer parte da vida dos brasileiros.

Ainda assim, possuímos dados alarmantes quanto ao endividamento da população em geral, segundo o último levantamento do Serasa, em março de 2024, a inadimplência no Brasil cresceu pelo segundo mês consecutivo. O aumento é de 1,19% em relação ao mês anterior, que corresponde a um acréscimo de 855 mil no número de consumidores inadimplentes. Ao todo, são 72,89 milhões de brasileiros em situação de inadimplência.

Diante de todos esses desafios, ter uma planilha de orçamento familiar pode ajudar na organização financeira. A planilha de gastos é uma maneira de organizar e somar todas as despesas em um só lugar, o que ajuda a enxergar com muito mais clareza para onde está indo o dinheiro.

A principal vantagem de fazer a gestão de gastos com uma planilha de orçamento doméstico é poder manter as contas em dia e não entrar em dívidas maiores do que se pode pagar. Outra grande vantagem é poder planejar melhor os investimentos a longo prazo.

## **2.1 Tecnologias Utilizadas no Desenvolvimento**

A seguir serão apresentados as tecnologias e os recursos utilizados para o desenvolvimento da aplicação que tem como objetivo facilitar o planejamento financeiro não só de famílias, mas também das pessoas que moram sozinhas.

### **2.1.1 OutSystems**

A OutSystems surgiu em 2001, em Portugal, cidade de Lisboa. Trata-se de uma plataforma de desenvolvimento Low-Code que fornece ferramentas para as empresas desenvolverem, implementarem e gerirem aplicações empresariais. Atualmente, contam escritórios pelo mundo todo, mais de 700.000 membros na comunidade, mais de 500 parceiros e milhares de clientes ativos distribuídos em 79 países e 21 setores.

No último relatório “Quadrante Mágico” realizado pela Gartner para 2023-2024 sobre as ferramentas Low-Code, a Outsystems foi classificada como “Líder”, atrás apenas da Mendix. O principal desafio enfrentado por seus usuários é a estrutura de preços, que é considerado elevado para PMEs (Pequenas e Médias Empresas), o modelo de precificação do OutSystems pode ser um obstáculo para organizações com orçamentos mais limitados.

**Figura 1 - Relatório Quadrante-Mágico realizado pela Gartner em 2023**



Fonte:

<https://s4-digital.com.br/revolucao-low-code-insights-do-quadrante-magico-da-gartner-2023-2024/>

Com a plataforma pode-se criar tanto aplicações web como mobile, no caso deste projeto, iremos fazer uma aplicação Web.

### 3 METODOLOGIA

Nesta etapa, vamos abordar a metodologia que foi utilizada no desenvolvimento do projeto, o método de pesquisa utilizado no início deste projeto até então foi a pesquisa científica, que, segundo Casarin e Casarin (2012, p.27) “a pesquisa científica é fundamentada em observações e evidências científicas e se caracteriza pelo embasamento no conhecimento acumulado.”, onde foram usadas as técnicas de coleta de dados para fazer os levantamentos da Fundamentação Teórica, e na segunda parte pro desenvolvimento da aplicação, a pesquisa aplicada, que segundo Mascarenhas (2018, p.48) “Usamos a pesquisa aplicada para estudar o problema em um contexto, buscando soluções para os desafios enfrentados nesse ambiente específico.”.

Quadro 1 - Cronograma de ação

<b>1. Definição</b>	1.1 Tema e Delimitação do tema 1.2 Escolha da ferramenta a ser utilizada
2.1 Regras de negócio	<b>2. Conhecimento</b>

<b>3. Desenvolvimento</b>	3.1 Desenvolvimento da Core Service 3.2 Desenvolvimento das Telas 3.3 Etapa de Testes
4.1 Análise	<b>4. Resultado</b>

Fonte: Elaborado pelo autor, 2024

## 4. DESENVOLVIMENTO

Neste tópico, será apresentado as etapas de desenvolvimento da aplicação, iniciando desde o desenvolvimento da Core Service, passando do Back-end e finalizando no Front-end (que devido a aplicação ser uma versão inicial, não será desenvolvido, e utilizará apenas a formatação padrão do OutSystems).

O projeto foi dividido em duas camadas, a primeira, onde foi feito os Serviços Principais (*Core Services*) junto com as Ações de Serviço (*Service Actions*) e a segunda, onde foi feito o Front-End, com a criação do formulário de preenchimento e a tabela que mostra os dados gravados

### 4.1 Criação da Core Service

A documentação do OutSystems diz o seguinte sobre as *Core Services*:

“Os **Core Services** devem ser sistemas agnósticos (Independentes de uma plataforma específica, o que permite que seja utilizado em uma grande variedade de dispositivos):

**APIs Externas:** O conhecimento de sistemas externos que consomem os serviços é limitado ao nível da **API Externa**. Qualquer alteração, adição ou remoção de um consumidor externo deverá impactar apenas a **API Externa**.

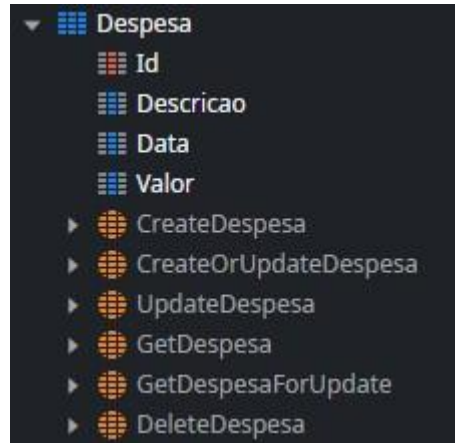
**Serviços de Integração:** Se o **Core Service** estender um sistema externo de registros (um produtor externo), isso também deverá ser abstraído pelo nível dos **Serviços de Integração**. Se um produtor externo mudar ou for substituído, desde que essa mudança seja abstraída pelos **Serviços de Integração**, os Core Services não serão afetados.”

Fonte: [https://success.outsystems.com/documentation/best\\_practices/architecture/designing\\_the\\_architecture\\_of\\_your\\_outsystems\\_applications/service\\_oriented\\_architectures\\_for\\_outsystems\\_applications/](https://success.outsystems.com/documentation/best_practices/architecture/designing_the_architecture_of_your_outsystems_applications/service_oriented_architectures_for_outsystems_applications/)

#### 4.1.1 Criação das Entidades do Banco de Dados

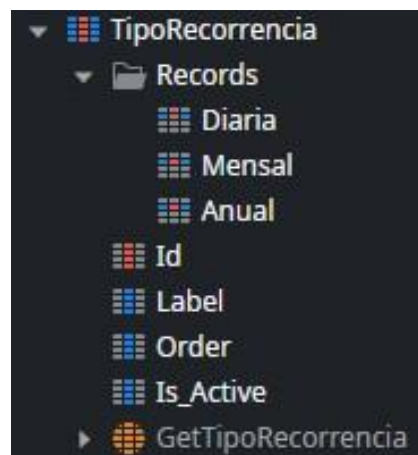
Após a criação da *Service*, dentro do Banco de Dados foram criadas duas Entidades, uma chamada “*Despesa*”, que guarda a Descrição, a Data e o Valor, e uma Entidade Estática chamada “*TipoRecorrecia*” que possui gravada nela três dados (Diaria, Mensal e Anual).

**Figura 2 - Tabela *Despesas***



Fonte: Elaborado pelo Autor, 2024.

**Figura 3 - Tabela *TipoRecorrendia***



Fonte: Elaborado pelo autor, 2024.

#### 4.1.2 Criação das Service Actions

No OutSystems, expor uma *Service Action* gera uma forte dependência do módulo consumidor ao módulo produtor, de uma forma fortemente acoplada.

A lógica da *Service Action* é executada como se estivesse definida no módulo consumidor, rodando em um único processo com a mesma requisição e transação.

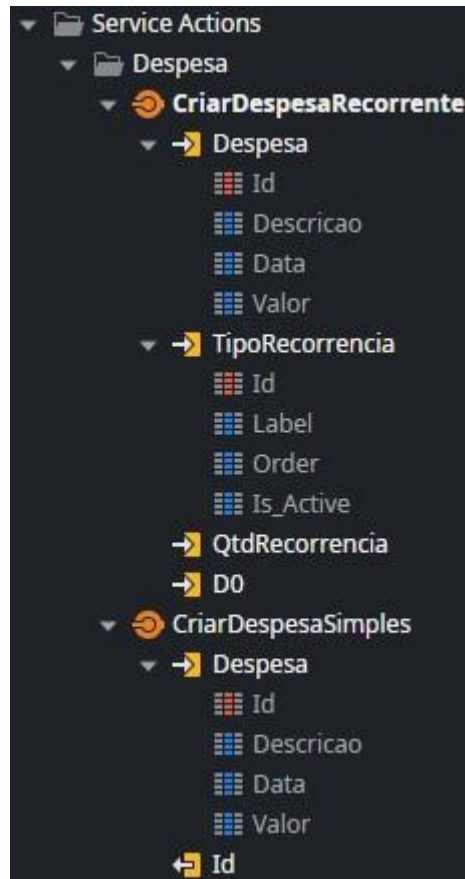
Sempre que uma implementação de uma *Service Action* exposta muda, os módulos consumidores devem ser atualizados e republicados para começarem a usar a última versão.

Neste projeto, foram criadas duas *Service Actions*, a “*CriarDespesaRecorrente*” e a “*CriarDespesaSimples*”.

Na primeira, foram criados 4 parâmetros de entrada (“Despesa”, “TipoRecorrendia”, “QtdRecorrendia” e “D0”).

Já na segunda, foram criados um de Entrada (Despesa) e um de Saída, que retorna o Id criado.

**Figura 4 - Todas Service Actions**



Fonte: Elaborado pelo autor, 2024.

### 4.1.3 Criação das Server Actions

Em OutSystems, uma *Server Action*, é uma ação que executa no lado do servidor.

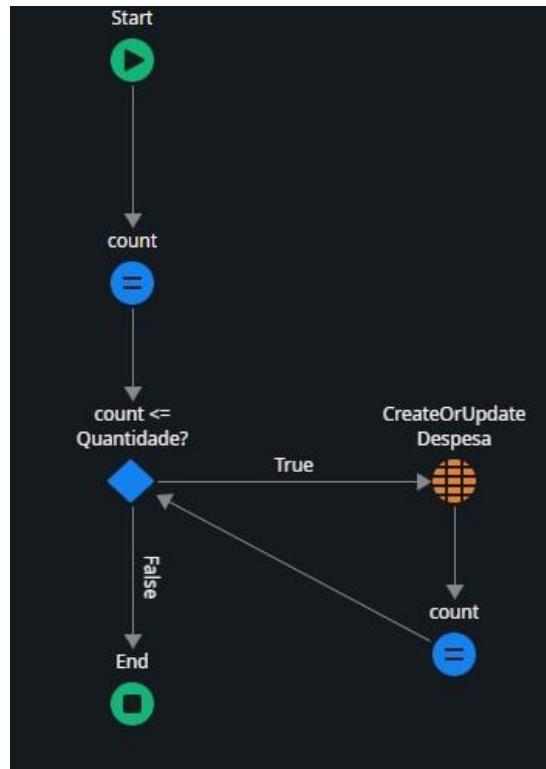
Elas não podem ser expostas em 2 casos:

- Quando possuem parâmetro que é definido através de uma Entidade/Estrutura que não está exposta.
- Quando possuem parâmetro que é definido através de uma Entidade/Estrutura que é reaproveitada de outro módulo.

Caso a execução dos módulos Produtor e Consumidor esteja sob diferentes Provedores de Usuários, os módulos possuem sessões diferentes. Neste caso, as variáveis associadas à sessão podem conter valores diferentes entre os módulos.

Dentro do projeto, foram criadas 3 *Server Actions*, a lógica de todas é praticamente a mesma, apenas alterando o tipo de validação da data para caso ela seja uma despesa Anual, Mensal ou Diária.

**Figura 5 - Modelo lógico de validação das Server Actions**



Fonte: Elaborado pelo Autor, 2024.

Dentro das *Server Actions*, foi criada uma variável local chamada “count”, que foi atribuída ao campo “Assign” (representado pelo “=” ) dentro do Modelo Lógico.

A Variável “Count”(Assign) atribui a ela o campo D0 (Despesa única), para fazer o teste dentro no IF.

Dentro do IF, é feito o seguinte teste lógico “Count <= Quantidade”, caso seja true, ele vai gravar a despesa no banc através do *CreateOrUpdateDespesa*, e seguir para o segundo Count onde ele adiciona 1 ao Count, e voltará para o if, seguindo assim para o End e finalizando o processo.

Dentro do *CreateOrUpdateDespesa* é onde ocorre a variação entre as 3 *Server Actions*.

O atributo Data muda de acordo com a *Service Action*, em despesa diária, é utilizado o *AddDays*, em mensal, o *AddMonths* e no Anual, o *AddYears*.

Figura 6 - Configuração do *CreateOrUpdateDespesa*

CreateOrUpdateDespesa ?	
Run Server Action	
Name	CreateOrUpdateDespesa
Action	CreateOrUpdateDespesa
Source	
Id	NullIdentifier()
Descrição	Despesa.Descricao
Data	AddDays(Despesa.Data,count)
Valor	Despesa.Valor

Fonte: Elaborado pelo Autor, 2024.



#### 4.1.4 Finalização das Service Actions

Após finalizar as *Server Actions*, foi finalizada a configuração das *Service Actions* criadas anteriormente.

Dentro do *CriarDespesaSimples* foi criado o seguinte esquema lógico:

Figura 7 - Modelo Lógico do *CriarDespesaSimples*

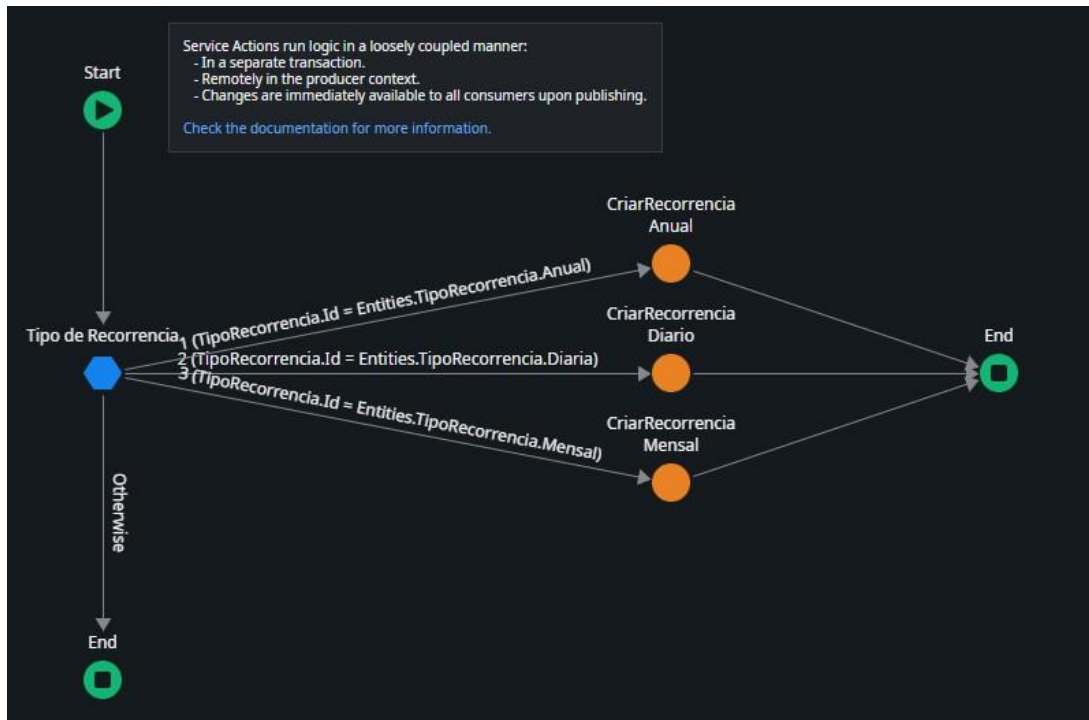


Fonte: Elaborado pelo Autor, 2024.

O *Assign* recebe a valor do ID e grava no banco de dados a despesa simples.

Já dentro do *CriarDespesaRecorrente* foi criado o seguinte esquema lógico:

Figura 8 - Modelo Lógico do *CriarDespesaRecorrente*



Fonte: Elaborado pelo Autor, 2024.

De forma resumida, o Switch realiza a validação para o tipo da recorrência da Despesa, e feito o teste, ele manda para a *Server Action* respectiva, que realiza o processo e grava no banco, e com isso, finalizamos o nosso Back-End.

## 4.2 Criação do Front-End

Após finalizar o Back-End, passamos para a criação das telas.

Após a criação da aplicação principal, vamos aplicar o módulo Core que criamos anteriormente para esse novo, e após isso, foi criada a única tela do projeto chamada “Despesas”.

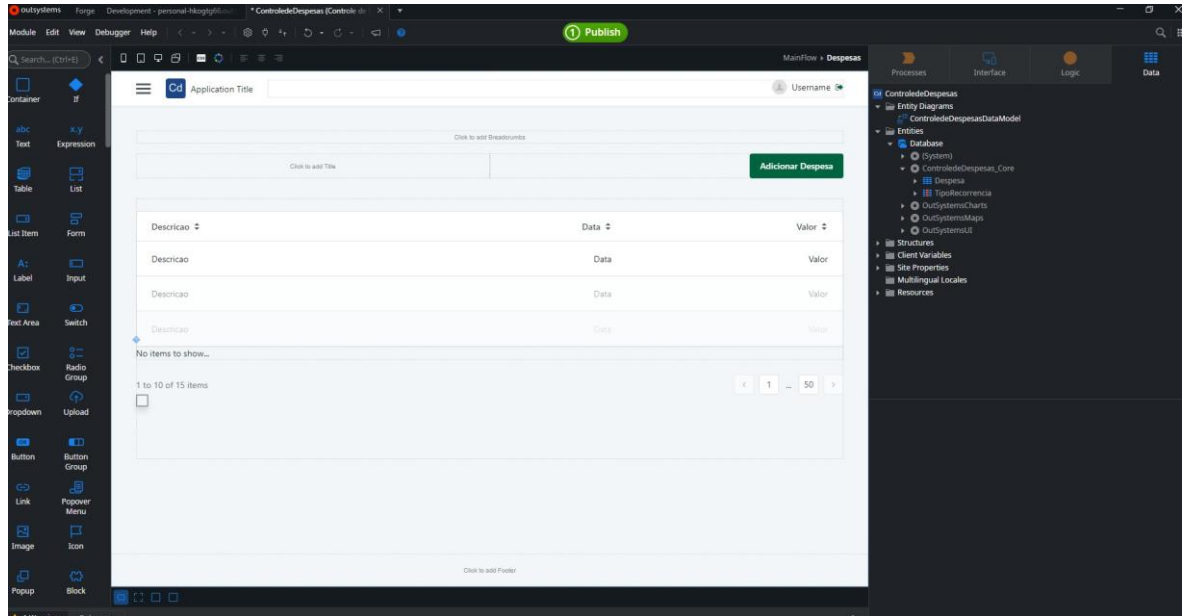
Dentro dessa tela, foi estabelecido 3 coisas e serem desenvolvidas:

- Uma tabela ligada com o DB para mostrar as despesas;
- Um Popup onde as despesas vão ser cadastradas;
- Um filtro na tabela para pesquisá-las por data e ano. Feito essa separação, partimos para o desenvolvimento.

### 4.2.1 Criação da Tabela

O OutSystems nos permite uma criação fácil e rápida de tabelas, que consiste em um simples “Arrastar e soltar”.

Figura 9 - Tela com a tabela dentro da IDE do OutSystems



Fonte: Elaborado pelo Autor, 2024.

#### 4.2.2 Criação do Popup

Após a criação da tabela, foi criado o botão “Adicionar Despesa”, este botão abre um popup para cadastrar a despesa e a gravar dentro do banco de dados.

**Figura 10 - Popup de cadastro dentro da IDE**

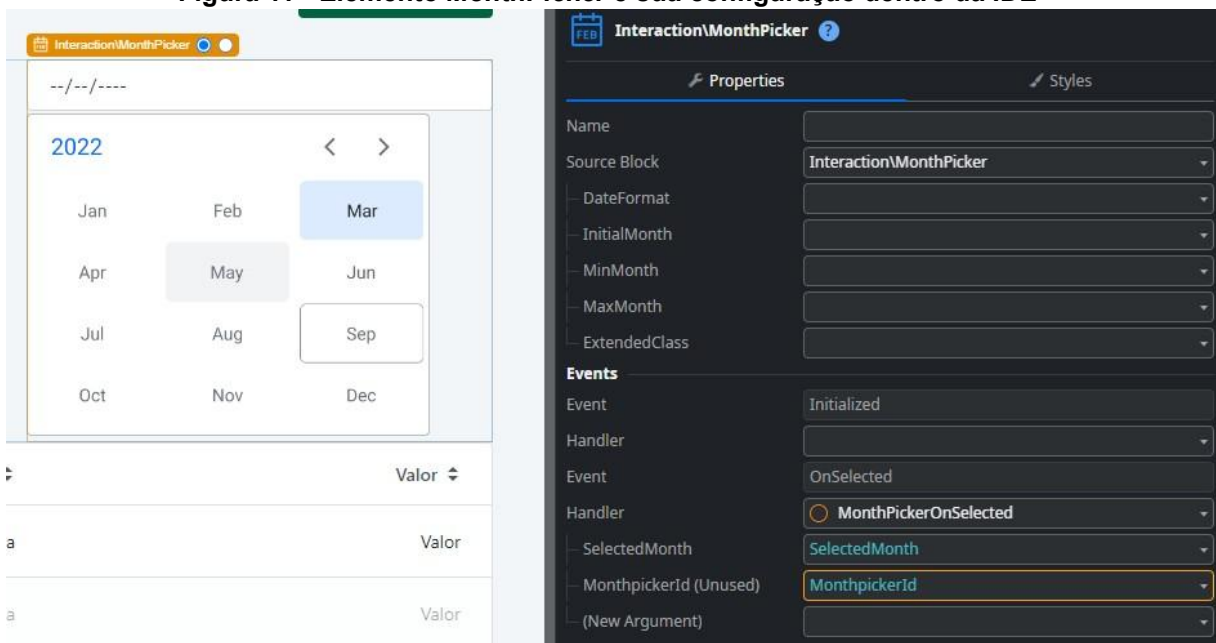
Fonte: Elaborado pelo Autor, 2024.

### 4.2.3 Criação do Filtro da tabela

Dentro de várias ferramentas low-code, temos diversos plugins prontos para agilizar o nosso trabalho e evitar retrabalho ou mão de obra desnecessária.

Nesse projeto, utilizamos um elemento próprio da OutSystems, o *MonthPicker*, que permite criar um filtro para selecionar diretamente por mês e ano, que é exatamente o que queremos nesse projeto.

**Figura 11 - Elemento *MonthPicker* e sua configuração dentro da IDE**



Fonte: Elaborado pelo Autor, 2024.

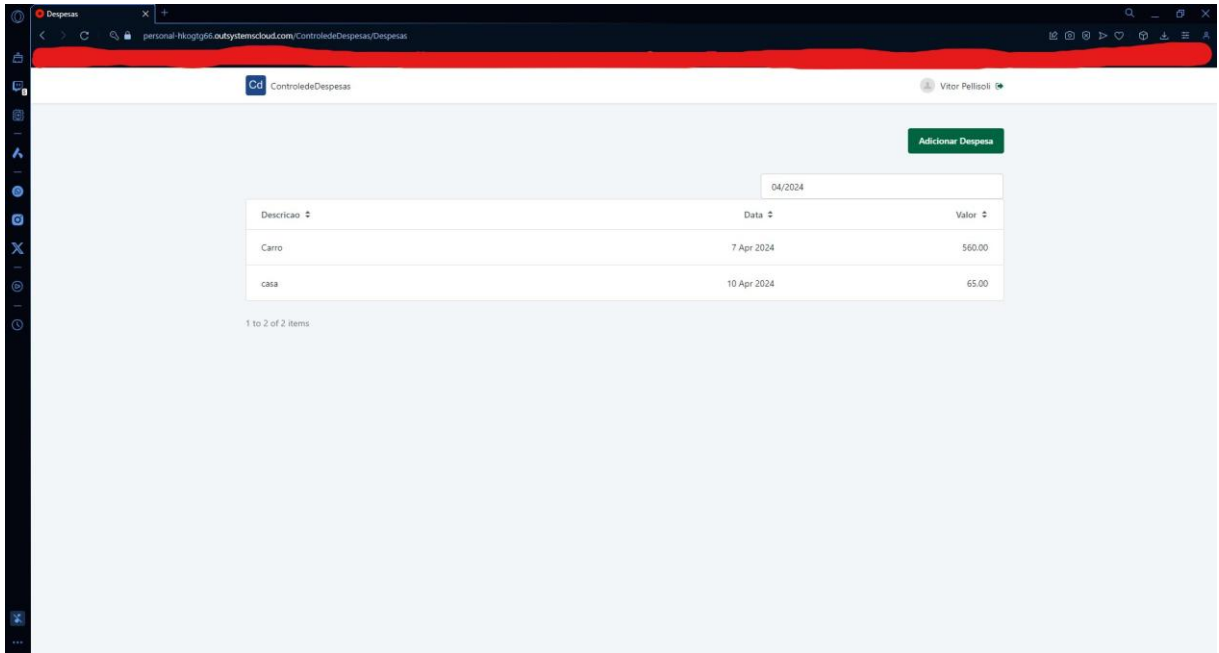
Feito essa última configuração, o projeto proposto está finalizado.

### 4.3 O Projeto Finalizado

Em OutSystems, os projetos ficam armazenados em uma Cloud pessoal de cada usuário, e só podem ser acessados para realizar os testes pelo link do projeto.

O projeto finalizado atingiu as necessidades propostas no início desse projeto.

**Figura 12 - Captura de tela da aplicação funcionando em um navegador (junto de valores teste)**



Descricao	Data	Valor
Carro	7 Apr 2024	560.00
casa	10 Apr 2024	65.00

Fonte: Elaborado pelo Autor, 2024.

## 5 ANÁLISE DE RESULTADOS

Conforme foi mostrado na etapa de desenvolvimento, o projeto obteve um resultado satisfatório, com uma interface intuitiva, disponibilizando as informações de forma clara sem a necessidade de uma grande curva de aprendizado por parte do usuário final.

Os testes obtiveram sucesso quanto a interação entre a aplicação e o banco de dados, por ter sido desenvolvido utilizando o Low-code, o tempo de desenvolvimento foi demasiado rápido, consegui desenvolver essa aplicação inteira em menos de 1:30h, pois graças ao low-code, conseguimos sintetizar uma grande gama de processos que tomam bastante tempo na forma tradicional em apenas pouco minutos, o que permite aos times gastarem mais tempo resolvendo os problemas do cliente do que desenvolvendo o projeto em si.

## **6 CONSIDERAÇÕES FINAIS**

Utilizando como base os objetivos definidos no início do projeto, o resultado final obtido é condizente com o que foi proposto nas expectativas iniciais, porém, é importante salientar que existem melhorias a serem acrescentadas no sistema para que o projeto atinja um nível maior de qualidade.

Algumas funcionalidades, como permitir adicionar o valor total com o número de parcelas e fazer a separação automaticamente está no plano de longo prazo do projeto.

E também obviamente, uma melhor interface gráfica a ser criada na aplicação, que está rodando com o design padrão do OutSystems.

## 7 REFERÊNCIAS

ORTIZ, Elaine. Saiba por que uma planilha de orçamento familiar é importante. **Serasa Score**, 2023. Disponível em <https://www.serasa.com.br/score/blog/saiba-por-que-uma-planilha-de-orcamento-familiar-e-importante/>. Acesso em 23/04/2024.

GONÇALVES, Felipe Pereira; AGNER, William Ricardo Fialka. SFD - Sistema de Finanças Domésticas. **Faculdade Guairacá Guarapuava**, 2013. [https://www.semanaacademica.org.br/system/files/artigos/sfd\\_-\\_sistema\\_de\\_financas\\_domesticas.pdf](https://www.semanaacademica.org.br/system/files/artigos/sfd_-_sistema_de_financas_domesticas.pdf). Acesso em 23/04/2024.

MOURA, Bruno de Freitas. Endividamento e inadimplência do consumidor caem em fevereiro. **Agência Brasil**, 2024. Disponível em <https://agenciabrasil.ebc.com.br/economia/noticia/2024-03/endividamento-e-inadimplencia-do-consumidor-caem-em-fevereiro#:~:text=De%20acordo%20com%20a%20CNC,%25%20para%2078%2C8%25>. Acesso em 23/04/2024.

OUTSYSTEMS. Bem-vindo à OutSystems, Página Inicial. Disponível em <https://www.outsystems.com/pt-br/company/>. Acesso em 26/04/2024.

Revolução Low-Code: Insights do Quadrante Mágico da Gartner 2023-2024. **S4 Digital**, 2023. Disponível em <https://s4-digital.com.br/revolucao-low-code-insights-do-quadrante-magico-da-gartner-2023-2024/>. Acesso em 26/04/2024

CASARIN, Helen de Castro Silva; CASARIN, Samuel José. Pesquisa científica: da teoria à prática. 1. ed. Curitiba: Intersaberes, 2012. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 28/04/2024

MASCARENHAS, Sidnei Augusto (org.). Metodologia científica. 2. ed. São Paulo, SP: Pearson, 2018. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 28/04/2024

ALGAR, Jonathan. Server Action, **OutSystems**, 2024. Disponível em [https://success.outsystems.com/documentation/11/reference/outsystems\\_language/logic/implementing\\_logic/logic\\_elements/server\\_action/](https://success.outsystems.com/documentation/11/reference/outsystems_language/logic/implementing_logic/logic_elements/server_action/). Acesso em 14/05/2024

ALGAR, Jonathan. Use Services to Expose Functionality, **OutSystems**, 2024. Disponível em [https://success.outsystems.com/documentation/11/building\\_apps/reusing\\_and\\_refactoring/use\\_services\\_to\\_expose\\_functionality/](https://success.outsystems.com/documentation/11/building_apps/reusing_and_refactoring/use_services_to_expose_functionality/). Acesso em 14/05/2024

ALGAR, Jonathan. Service-oriented architectures for OutSystems applications, **OutSystems**, 2024. Disponível em [https://success.outsystems.com/documentation/best\\_practices/architecture/designing\\_the\\_architecture\\_of\\_your\\_outsystems\\_applications/service\\_oriented\\_architectures\\_for\\_outsystems\\_applications/](https://success.outsystems.com/documentation/best_practices/architecture/designing_the_architecture_of_your_outsystems_applications/service_oriented_architectures_for_outsystems_applications/). Acesso em 14/05/2024