



# E-book Módulo 1 - Iniciando o Desenvolvimento com Dados

-  **O que são Dados?**
-  **O que é uma Entidade de Banco de Dados?**
-  **Como criar uma Entidade de Banco de Dados?**
-  **Apresentando Visualmente Dados Salvos**
-  **Importando Dados de um Arquivo Excel para uma Entidade de BD**
-  **Importando Novas Entidades de BD a partir de um Arquivo Excel**
-  **Conhecendo Entidades Estáticas**
-  **Encerramento**

## O que são Dados?

---

## O que são Dados?

*“Conhecimento que se tem sobre algo, usado para solucionar uma questão, fazer um julgamento, criar ou colocar em prática um pensamento, uma opinião.”*

Fonte: Dicionário

Se formos pensar em software, diversos tipos de dados podem ser guardados, como por exemplo, dados pessoais: nome completo, gênero, endereço, email e telefone para contato.

Outros exemplos de dados que podem estar presentes em soluções desenvolvidas podem ter relação com mercado financeiro, compras de produtos, jogos eletrônicos, restaurantes, eventos, e muito mais.

## CRUD

Softwares são criados para certos contextos. Assim, esperasse que algumas ações possam ser feitas com os dados envolvidos, como por exemplo, criar, recuperar, atualizar, e deletá-los. É comum essas ações estarem presentes em diferentes softwares.

Há um acrônimo chamado CRUD, que representa essas ações mencionadas. Cada letra desse acrônimo representa uma das ações, assim como indicado abaixo.

- **C - create** (*criar*)
- **R - Read** (*ler*)
- **U - Upgrade** (*atualizar*)
- **D - Delete** (*deletar*)

## Importância da dados para softwares

Abaixo, são apresentadas três razões que exemplificam a importância de dados para softwares.

1) PERSONALIZAÇÃO	2) TOMADA DE DECISÃO	3) ARMAZENAMENTO E RECUPERAÇÃO
<p>Diversas soluções usam dados coletados para personalizar alguma experiência para algum usuário.</p> <p>Assim, se o usuário tem certas preferências, ou quem sabe certo perfil, softwares podem usar isso como base para recomendar algo ou oferecer alguma experiência baseada nos dados do usuário.</p> <p>Exemplos que podem ser mencionados para exemplificar, seria: recomendar séries ou filmes baseado no que uma pessoa tem assistido, ou quem sabe recomendar produtos baseado nas preferências do usuário.</p>		

1) PERSONALIZAÇÃO	2) TOMADA DE DECISÃO	3) ARMAZENAMENTO E RECUPERAÇÃO
<p>Softwares usam dados para tomar decisões e assim executar certas ações. Isso também tem relação com personalização, mas também se enquadra para outras situações que envolvam a necessidade de ter o dado para decidir e executar algo.</p> <p>Um exemplo simples seria identificar um pagamento estranho usando o cartão de crédito de alguém. Notificar o dono do cartão o quanto antes, a fim de confirmar se essa compra foi realmente feita por ele, poderia ser uma ação escolhida para lidar com a situação identificada.</p>		

1) PERSONALIZAÇÃO	2) TOMADA DE DECISÃO	3) ARMAZENAMENTO E RECUPERAÇÃO
<p>Quando uma pessoa, por exemplo, faz um login em uma rede social, recuperar informações relacionados a ela é algo importante. Não somente em redes sociais, mas em outros aplicativos, é esperado que quando um app é fechado e depois reaberto, certas informações sejam recuperadas e apresentadas como esperado.</p> <p>Portanto, armazenar informações e recuperá-las corretamente, é algo importante para permitir o bom uso de soluções oferecidas.</p> <p>Vale destacar que armazenamento de dados também pode ser mencionado como persistência de dados. Assim, persistir algum dado representa salvar, manter armazenado algum dado.</p>		

## O que é um banco de dados?

É uma forma estruturada, organizada e otimizada que permite o acesso e agrupamento de informações sobre assuntos relacionados.

Saiba que há diversas vantagens que motivam projetos usarem banco de dados para armazenar dados. Abaixo, são mencionadas duas delas:

1

### **Oferecer uma centralização dos dados.**

Permitindo que dados estejam em um mesmo local, facilitando a organização deles, assim como seu acesso.

2

### **Ajudar na construção de um ambiente seguro.**

Hoje em dia, existem diversos recursos que ajudam nessa construção, como por exemplo, a possibilidade de fazer backup de dados com alguma frequência, dentre diversos outros.

## O que é uma Entidade de Banco de Dados?

---

## O que é uma Entidade de Banco de Dados?

---

É um elemento que permite **armazenar** e **acessar informações** que são necessárias para alguma solução construída. Cada entidade deve ter **atributos**, que representam os **tipos de informações** que poderão ser armazenados nela.

Para ajudar no entendimento, poderíamos ter, como exemplo, uma entidade chamada **Pessoa**, onde são armazenadas informações de pessoas cadastradas em algum sistema.

Exemplos de atributos que poderiam fazer parte da entidade **Pessoa** poderiam ser:

- Nome
- Endereço
- Data de nascimento
- Email
- Celular
- Etc.

Vale destacar que na plataforma OutSystems, como padrão, quando é criada uma nova entidade, um atributo Id, que representa o identificador único de cada registro, também é criado de forma automática pela plataforma.

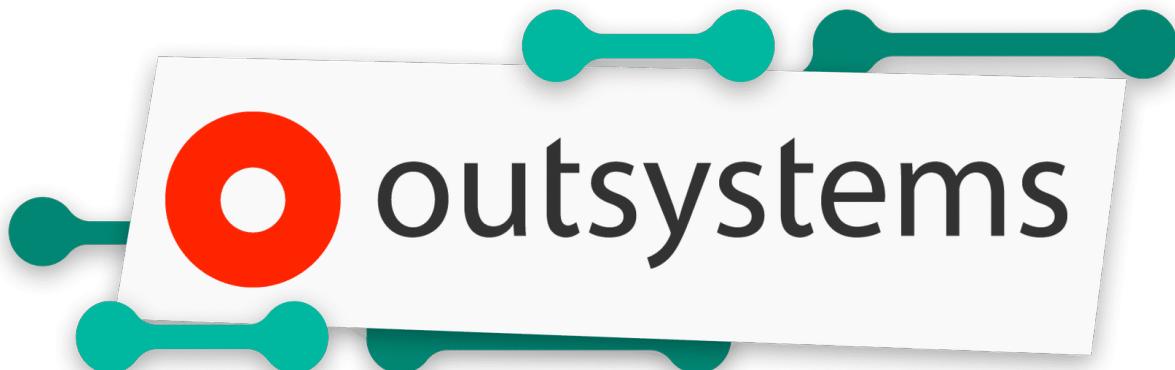
Caso o desenvolvedor deseje, ele pode definir qual tipo de informação será esse identificador único, como por exemplo, os cpfs das pessoas cadastradas.

## O que é um Registro?

A plataforma da *OutSystems* considera que cada cadastro realizado em alguma entidade é um Registro, ou seja, um **Record**.

Uma outra forma de mencionar um **Registro** é usar o termo **Instância** ou *Instance*. Com isso, tanto registro como instância, podem ser usados para referenciar cadastros presentes em alguma entidade.

Com isso, se houver duas pessoas cadastradas em uma entidade Pessoa, isso quer dizer que há dois registros/instâncias presentes.



## O que são Entity Actions?

Entity Actions são ações criadas de forma automática pela plataforma da OutSystems, quando uma nova entidade é criada. Essas ações executam operações básicas de dados: CRUD.

Com isso as Entity Actions permitem que desenvolvedores facilmente possam usar ações capazes de criar, ler, atualizar e deletar dados de alguma entidade.

---

Esse é um recurso muito útil e que traz **agilidade** para o desenvolvimento de **soluções Low-Code**.

## Como criar uma Entidade de Banco de Dados?

---

# Como criar uma Entidade de Banco de Dados?

A plataforma da OutSystems oferece recursos que ajudam a armazenar, recuperar e manipular dados que possam estar presentes em soluções Low-Code desenvolvidas.

Para permitir o uso desses recursos, deve ser criada uma entidade de banco de dados.

---

**Mas como criá-la?**



## **Criando uma entidade de banco de dados**

Aqui há um passo a passo que explica como criar alguma entidade.

## Passo 1

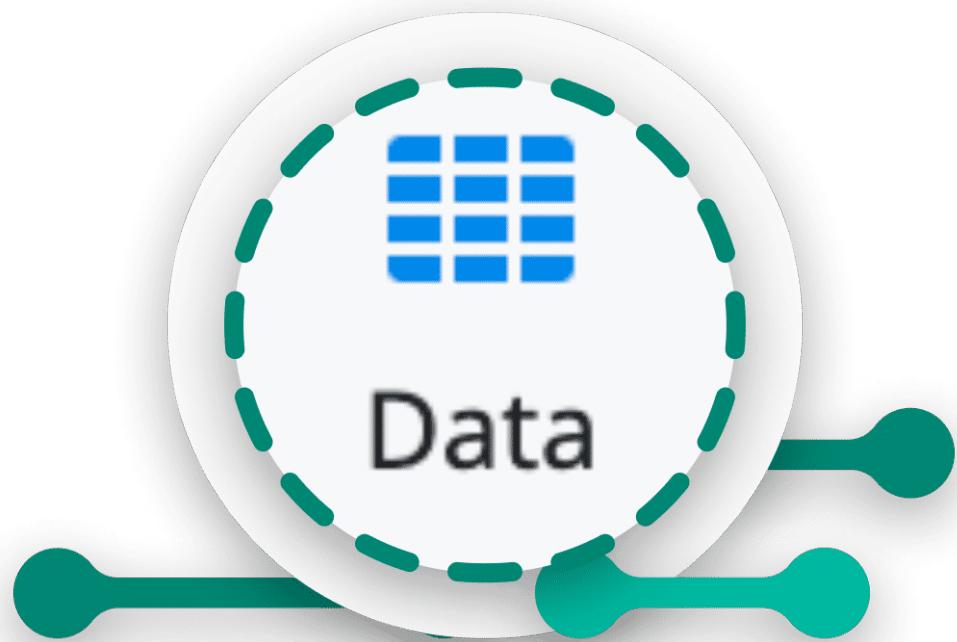
Crie ou acesse algum projeto de desenvolvimento Low-Code.

Use a plataforma **Service Studio**.

Caso não tenha essa plataforma instalada, baixe a partir do link:

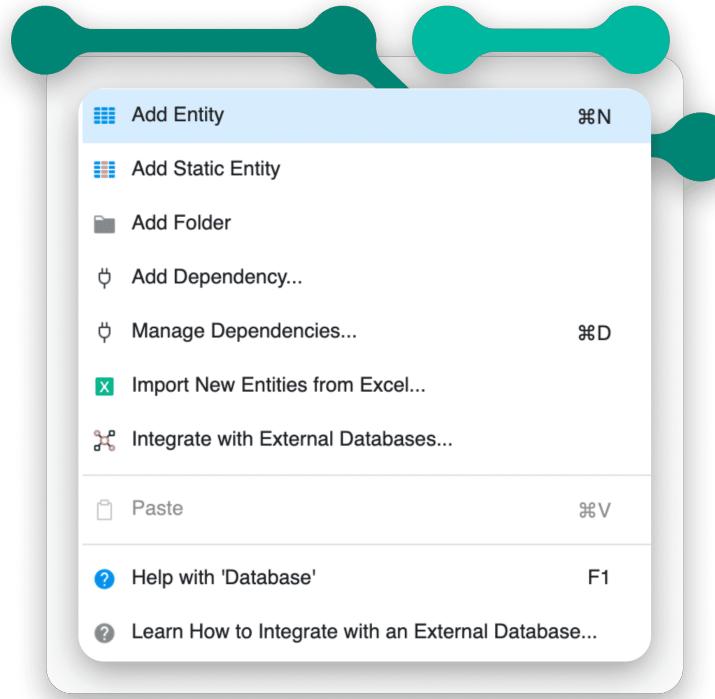
[https://www.outsystems.com/Portal/Trial\\_Portal](https://www.outsystems.com/Portal/Trial_Portal)

**Passo 2**



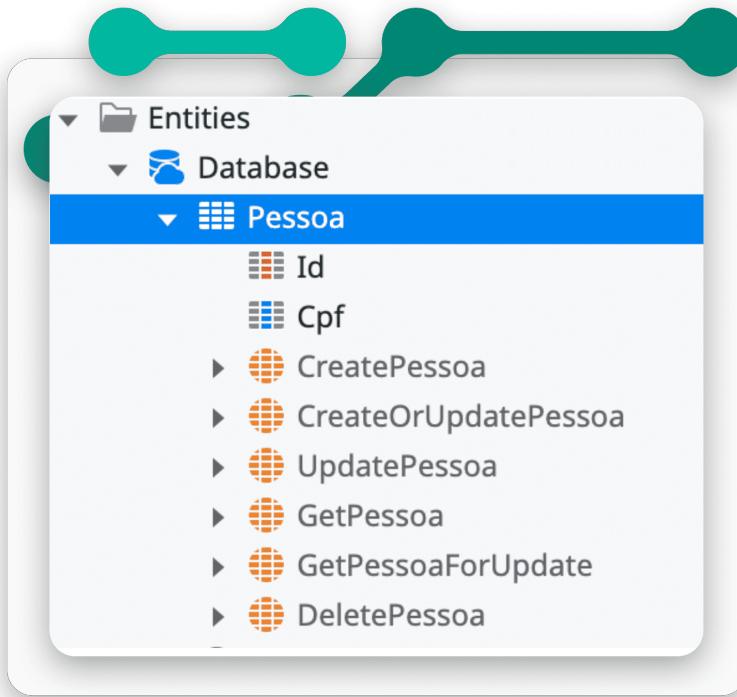
Acesse a **aba Data**.

### Passo 3



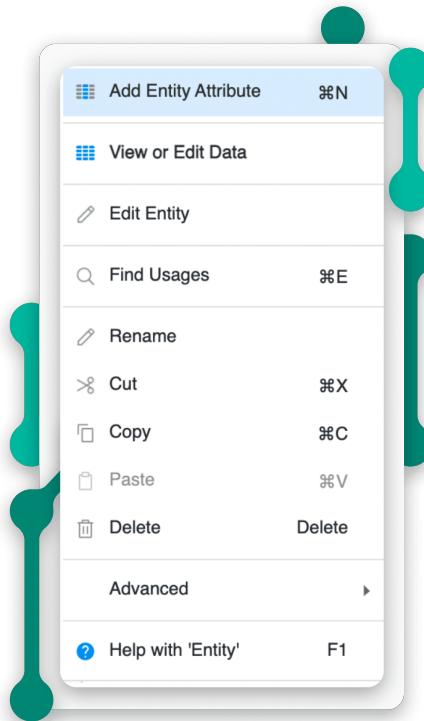
Clique com o botão direito do mouse em "**Database**" e escolha a opção "**Add Entity**".

#### Passo 4



Forneça o nome da sua entidade (p.ex. Pessoa). Perceba que um atributo Id foi criado de forma automática, assim como um conjunto de Entity Actions (p.ex. CreatePessoa, CreateOrUpdatePessoa, ..., DeletePessoa).

## Passo 5



Crie novos atributos conforme precisar, clicando com o botão direito do mouse na entidade desejada e escolhendo a opção “**Add Entity Attribute**”.

## **Pronto!**

Com estes passos você deve conseguir entidades e seus atributos.

## **Configurando atributos**

Conforme novos atributos forem criados em alguma entidade, o desenvolvedor poderá definir qual tipo de dado será armazenado no atributo correspondente.

A plataforma OutSystems oferece um conjunto de tipos que podem ser usados, como por exemplo, texto (text), números inteiros (Integer), Email, telefone (Phone Number), data (Date) etc.

Para escolher o tipo de dado a ser usado, basta escolhê-lo a partir da propriedade Data Type do atributo correspondente, assim como ilustrado no exemplo ao lado.



Cpf ?

Entity Attribute

Name Cpf

Description ...

Label Cpf

Data Type Text

Length

Is Mandatory

Default Value

*Created by ticemtrilhas@*

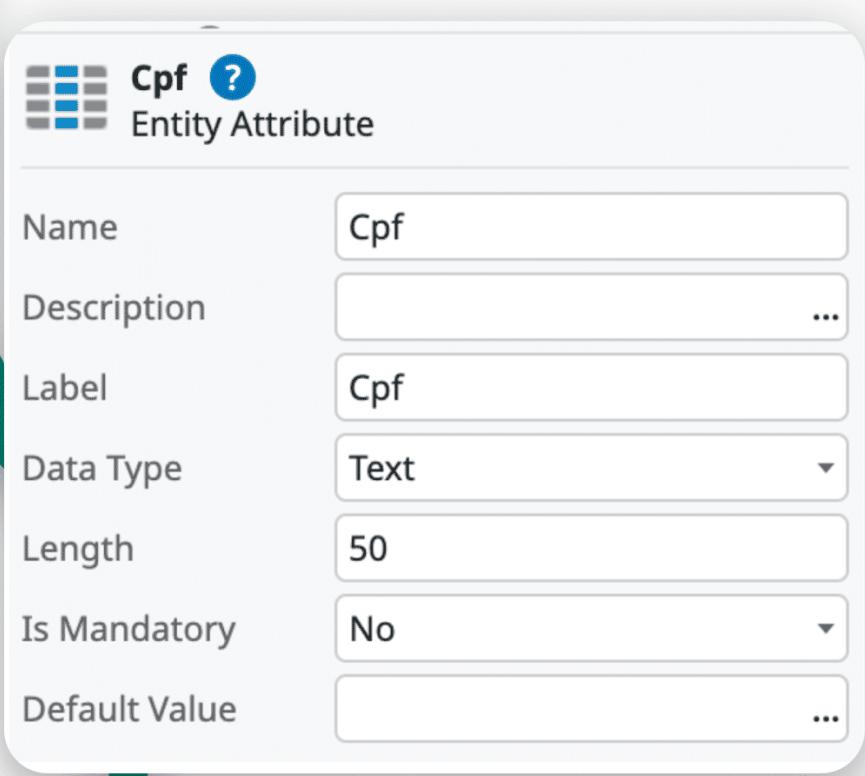
*Last modified by ticemtri*

#### Basic Types

- Text
- Integer
- Long Integer
- Decimal
- Boolean
- Date Time
- Date
- Time
- Phone Number
- Email
- Binary Data

Dependendo do tipo de dado definido, certas configurações adicionais podem ser realizadas, **como por exemplo:**

- A descrição (Description) que descreve o que é o atributo.
- Se o campo é ou não obrigatório (Is Mandatory).
- Qual seu valor padrão (Default Value).
- Caso seja do tipo texto (Text), poderá ser informado o tamanho (Length) limite de caracteres (p.ex. letras, números e caracteres especiais) que pode ser fornecido pelo usuário.



**Cpf** ?

Entity Attribute

Name	<input type="text" value="Cpf"/>
Description	<input type="text" value="..."/>
Label	<input type="text" value="Cpf"/>
Data Type	<input type="text" value="Text"/> ▼
Length	<input type="text" value="50"/>
Is Mandatory	<input type="text" value="No"/> ▼
Default Value	<input type="text" value="..."/>

## Apresentando Visualmente Dados Salvos

---

**Apresentando visualmente  
dados salvos em uma  
página web**

Para apresentar dados de registros salvos em alguma entidade de banco de dados em uma página web, uma ótima maneira de permitir essa apresentação é usar o componente visual (Widget) List.

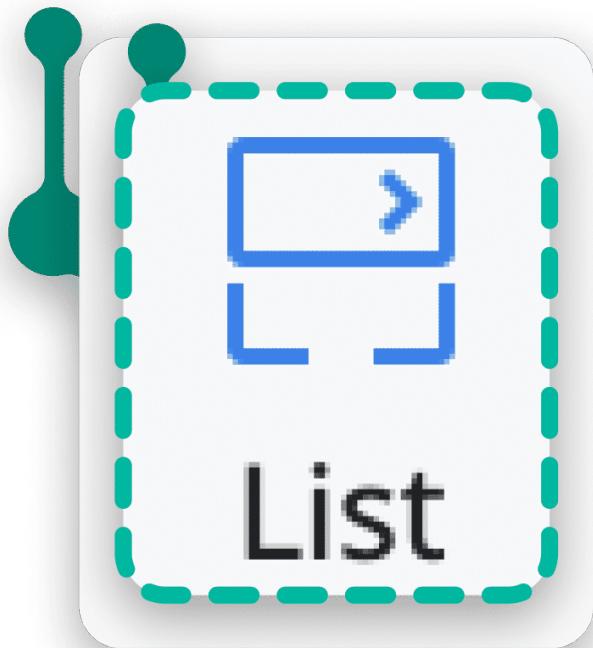
*Abaixo, encontra-se um passo a passo que explica como criar uma página web usando tal componente.*

## Passo 1

Crie uma nova tela (Screen) em um projeto já criado a partir do Service Studio.

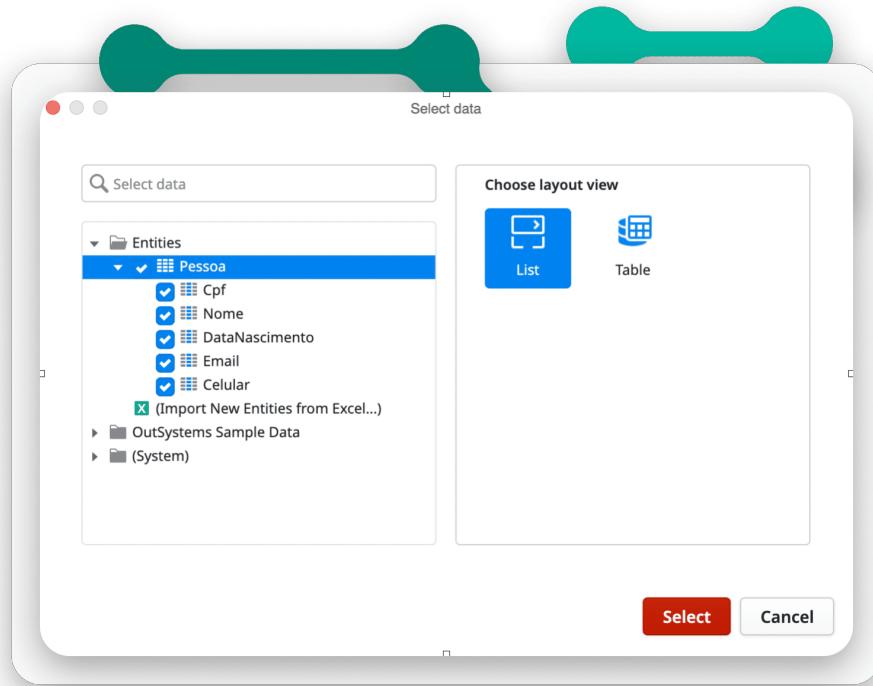
Use inicialmente o template "Empty" nessa criação.

## Passo 2



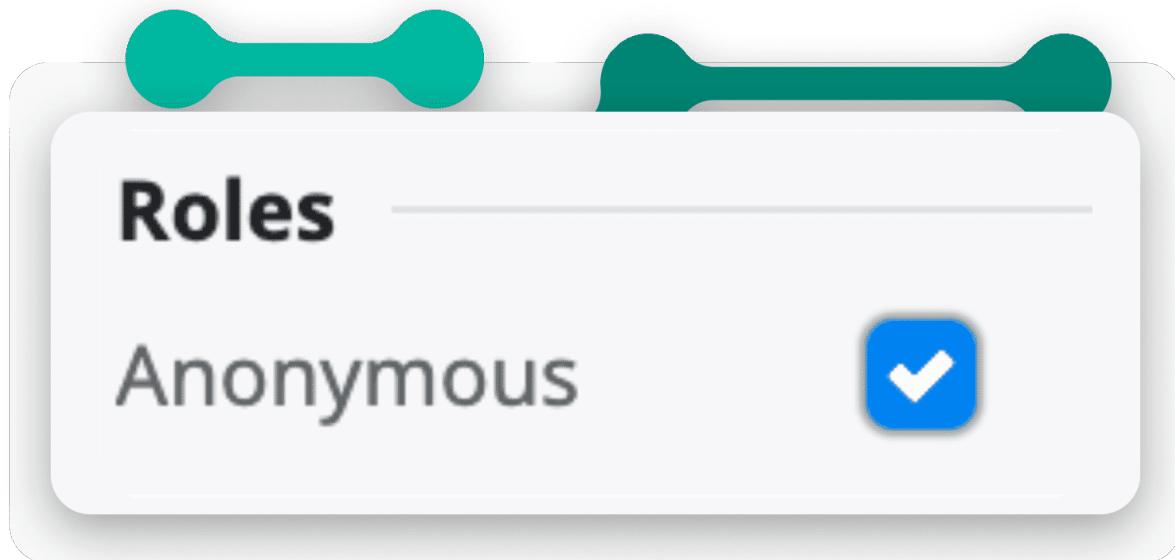
Selecione o componente **List** e arraste para a nova tela criada.

### Passo 3



Clique duas vezes onde o componente foi adicionado na tela, selecione quais atributos deseja apresentar seus dados, e escolha o formato visual List ou Table a ser usado.

Passo 4



Caso não tenha um controle de acesso por login criado, coloque como anônima a página que irá apresentar os dados, assim como ilustrado na imagem.

## Passo 5

### Publique a solução.

Ao publicar, abra a página clicando em “*Open in browser*”



**Dica:** Experimente criar duas páginas, sendo que cada uma deve usar um formato diferente de apresentação: List ou Table.

## Importando Dados de um Arquivo Excel para uma Entidade de BD

---

# Importando Dados de um Arquivo Excel para uma Entidade de Banco de Dados

Para importar dados de um arquivo Excel para alguma entidade de banco de dados já criada, os seguintes passos devem ser realizados.

1

Ter o arquivo Excel com os dados que deseja importar.

2

Caso não tenha uma entidade de banco de dados criada, onde os dados serão importados, o desenvolvedor deverá criá-la com os

atributos correspondentes. Procure manter a consistência dos nomes dos atributos com as colunas presentes no arquivo Excel a ser usado.

3

Na entidade de banco de dados a ser usada, clique com o botão direito e escolha a opção “Advanced > Create Action to Bootstrap Data from Excel”. Na sequência escolha o arquivo Excel que será importado e confirme.

---

## Pronto!

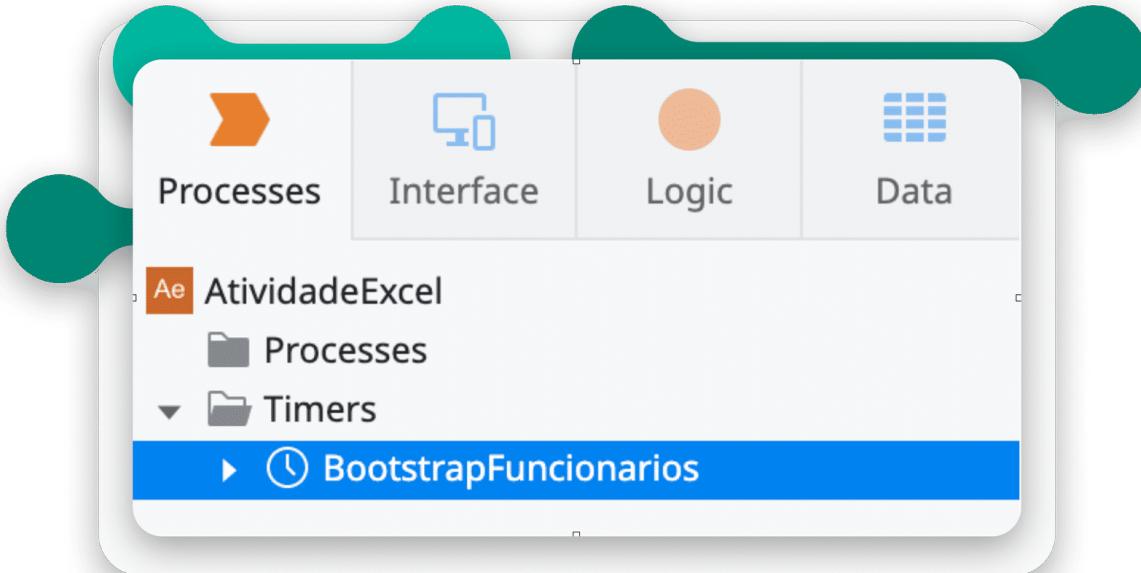
Com isso será possível importar dados para alguma entidade existente.

## Conhecendo o Timer

Timer é um background job, ou seja, uma ação que fica rodando por trás do software em momentos pré-definidos.

Quando uma importação de dados é criada, um Timer é criado na plataforma da OutSystems, assim como ilustrado na imagem ao lado. Perceba que o Timer neste caso foi chamado de

**BootstrapFuncionarios.**



Como padrão, somente são importados dados de um arquivo Excel caso não hajam dados inseridos na entidade considerada. Além disso, somente quando um projeto é publicado, que o Timer faz essa verificação para analisar se irá ou não inserir os dados.

## Importando Novas Entidades de BD a partir de um Arquivo Excel

---

# Importando Novas Entidades de Banco de Dados a partir de um Arquivo Excel

Para permitir a importação de novas entidade de banco de dados a partir de um arquivo Excel, os seguintes passos devem ser realizados.

## **Passo 1**

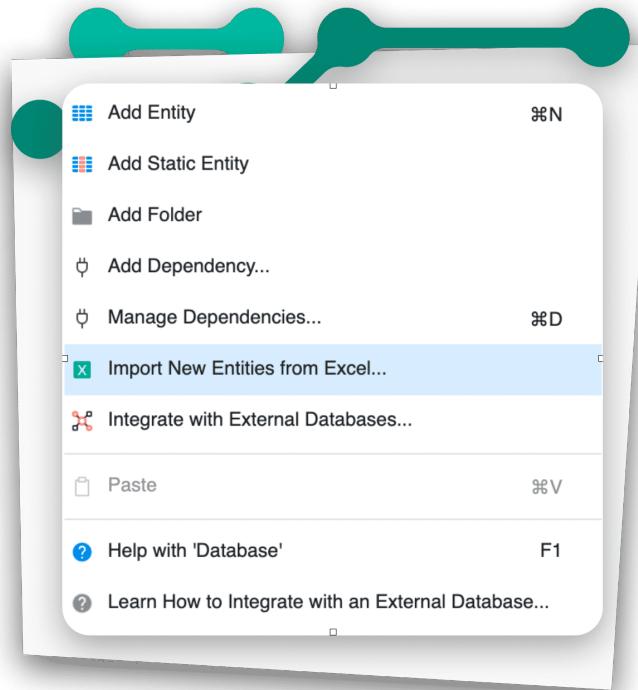
Tenha um arquivo Excel com as informações desejadas.

Lembre-se que cada aba desse arquivo, será considerada para criar uma nova entidade.

## Passo 2

Acesse a aba *Data* de um projeto Low-Code criado no **Service Studio**.

### Passo 3



Clique com o botão direito do mouse em *Database*, e escolha a opção “*Import New Entities from Excel...*”, como ilustrado na imagem.

#### **Passo 4**

Selecione o arquivo Excel desejado e confirme a importação.

## Passo 5

Assim que as novas entidades forem criadas, realize a publicação da solução.

Com isso, você poderá visualizar os registros de cada entidade importada, caso no arquivo Excel, houvessem dados presentes por entidade.

## Conhecendo Entidades Estáticas

---

# Conhecendo Entidades Estáticas

Caso precise definir alguma lista de informações que não deve mudar no sistema, como por exemplo, o estado de quartos em um hotel, uma Entidade Estática pode ser usada. Nesse exemplo de quartos em hotel, os estados possíveis poderiam ser: **Livre e Ocupado**.

Assim, para representar valores que não serão alterados em alguma solução, Entidade estática pode ser usada. Esse é um tipo especial de entidade que cria um conjunto pré-definido de valores, ou seja, uma lista de registros, e que podem ser usados em soluções Low-Code.

Cada entidade estática tem um conjunto de atributos e registros. Cada registro possui os seguintes atributos :

- **Id**  
Identificador único de cada registro.
- **Label**  
Informação que será salva para caracterizar e ajudar a entender o que é cada registro.
- **Order**  
Indica qual a ordem de apresentação de cada registro
- **Is\_Active**  
Informa se o registro está ou não está ativo.

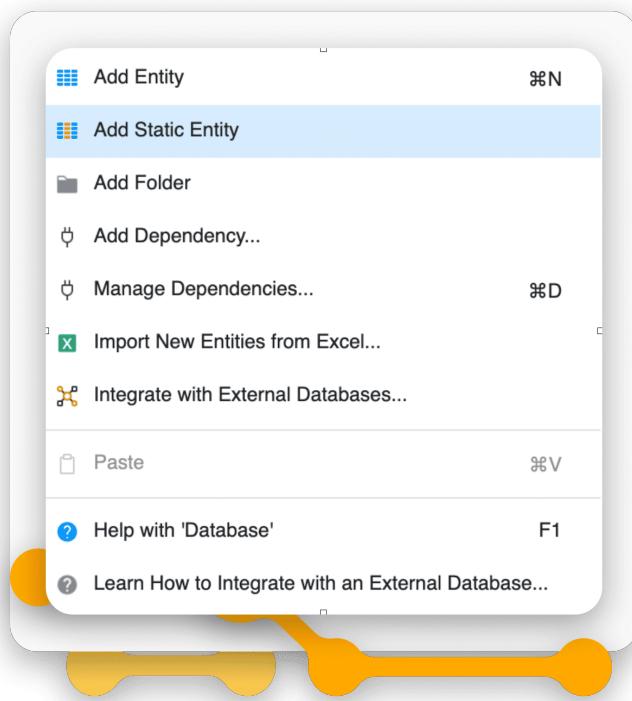
## Criando uma entidade estática

Para criar um entidade estática, abaixo encontram-se os passos necessários para criá-la.

## Passo 1

Acesse um projeto Low-Code criado a partir do Service Studio, e na sequência acesse a *aba Data*, visível geralmente ao lado direito da plataforma.

## Passo 2

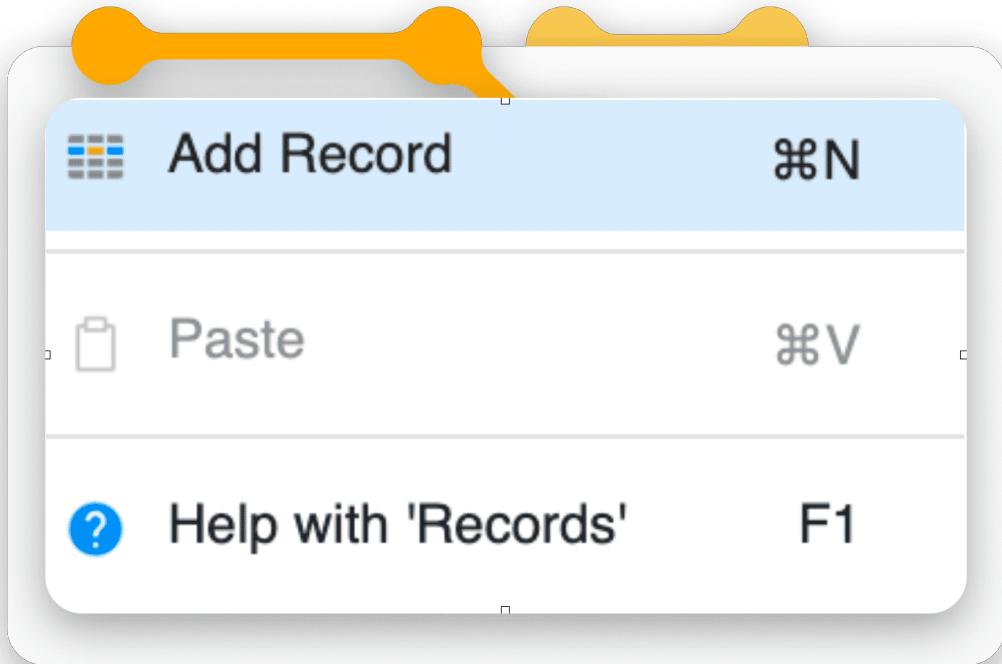


Clique com o botão direito do mouse em Database, e selecione a opção “Add Static Entity”, como ilustrado acima.

### **Passo 3**

Forneça um nome para a entidade criada.

#### Passo 4



Clique com o botão direito do mouse na pasta Records e crie novos registros clicando na opção “Add Record”. Forneça por registro criado, na propriedade Label, alguma informação que o caracterize. Isso ajudará em sua identificação.

## Passo 5

**Opcional:** Caso deseje criar algum novo atributo na entidade considerada, clique com o botão direito do mouse sobre ela e selecione a opção “Add Entity Attribute”. Na sequência, acesse cada registro já criado, e coloque a informação correspondente nesse novo atributo disponível.

## Passo 6

*Opcional:* Caso deseje alterar o nome de algum atributo criado, fique a vontade em aplicar essa mudança.

## Passo 7

Relacione outra entidade de banco de dados com a entidade estática criada.

**Assim, cada registro presente na entidade de banco de dados usada estará associado a algum registro da entidade estática.**

# Encerramento

---

*Clique na seta abaixo para fazer o download deste E-book.*



## File Attachment Block

No file added





**PARABÉNS**

*Você completou este Módulo*