

```
STATUS
ACCOUNT
STATUS
class Banner
  attr_accessible :horiz, :link, :visible, :image, :position
  has_attached_file :image, styles: { vert: '220'
  before_create :assign_position
end
def assign_position
  max = Banner.maximum(:position)
  self.position = max ? max + 1 : 0
end
```

FULL STACK DEVELOPMENT

LOW CODE DEVELOPMENT

# AULA 04

## SUMÁRIO

O QUE VEM POR AÍ? .....	3
HANDS ON .....	4
SAIBA MAIS .....	5
O QUE VOCÊ VIU NESTA AULA? .....	10
REFERÊNCIAS .....	11

EXEMPLO

## O QUE VEM POR AÍ?

Na aula passada nós construímos a camada de backend de uma solução com a plataforma Outsystems. Isso foi feito para podermos entender como é possível essa separação em OutSystems, e, além disso, criamos uma API para disponibilizar os serviços para outras aplicações.

Nesta aula, nós avançaremos os nossos estudos na parte de frontend: aprenderemos como importar o projeto backend e como trabalhar com a parte de frontend criando uma aplicação Web, e entenderemos como funciona o consumo de uma API.

## HANDS ON

Nesta aula prática, nós desenvolveremos um aplicativo completo para Web, aprenderemos como montar as telas, manipular estilos, utilizar Javascript e importar o plugin da forge para o nosso App.



## **SAIBA MAIS**

Conforme aprendemos anteriormente, a plataforma OutSystems é conhecida por sua abordagem de baixo código (low code), que permite que as empresas acelerem a criação de aplicativos empresariais personalizados.

Uma das áreas mais vitais do desenvolvimento de aplicativos é a parte Web, que abrange a criação de interfaces de usuário atraentes e funcionais para aplicativos Web.

Nesta aula, exploraremos como a OutSystems simplifica e otimiza o desenvolvimento da parte Web, permitindo que as empresas criem aplicativos de alta qualidade de maneira rápida e eficiente.

Vamos trazer alguns dos motivadores que te levarão a estudar esta plataforma:

### **1. Componentes predefinidos**

A OutSystems fornece uma biblioteca rica de componentes predefinidos, o que acelera o desenvolvimento e garante uma aparência consistente em todo o aplicativo.

As pessoas desenvolvedoras podem personalizar esses componentes de acordo com as necessidades do projeto.

### **2. Responsividade**

O desenvolvimento de aplicativos web responsivos é essencial para atender a uma variedade de dispositivos e tamanhos de tela.

O Service Studio facilita a criação de interfaces de usuário que se adaptam automaticamente a diferentes resoluções, garantindo uma experiência consistente em dispositivos móveis, tablets e desktops.

Quem já trabalha com desenvolvimento de sistemas sabe que esse é um dos maiores desafios quando desenvolvemos os nossos apps.

### **3. Integração simples**

A plataforma OutSystems oferece integração perfeita com sistemas de terceiros, bancos de dados e serviços, permitindo que os aplicativos web acessem e compartilhem dados de maneira eficiente.

#### 4. Fluxos de trabalho intuitivos

O Service Studio permite a criação de fluxos de trabalho de usuário por meio de um sistema visual. Isso simplifica a criação de aplicativos com uma lógica de negócios complexa.

#### 5. Teste e implantação fáceis

A plataforma oferece ferramentas de teste e implantação integradas que simplificam a validação, o teste e a implantação de aplicativos web.

Quem já trabalha há alguns anos com desenvolvimento frontend, sabe que esses são alguns dos problemas enfrentados (independente do framework); seja Angular, React, Vue, etc.

Agora que sabemos das vantagens, vejamos também algumas das desvantagens:

- **Custo:** embora seja uma plataforma poderosa, o custo associado ao uso do Outsystems pode ser alto, especialmente para organizações menores ou projetos com orçamentos limitados.
- **Dependência da plataforma:** uma vez que você começa a desenvolver com Outsystems, pode tornar-se dependente da plataforma para realizar atualizações e manutenções futuras. Mudar para outras tecnologias pode ser desafiador.
- **Personalização limitada:** enquanto o Outsystems oferece muitas funcionalidades, pode haver restrições em personalizações avançadas. Em alguns casos, certas customizações específicas podem ser difíceis de implementar.
- **Curva de aprendizado:** para desenvolvedores(as) acostumados com outras linguagens ou frameworks, a transição para o ambiente de desenvolvimento visual do Outsystems pode exigir tempo e esforço para se acostumar.
- **Desempenho:** embora a plataforma seja poderosa, o desempenho dos aplicativos desenvolvidos pode não ser tão otimizado em comparação com soluções feitas com linguagens de programação mais diretas.

- **Limitações de integração:** pode haver limitações ao integrar certos sistemas externos ou tecnologias específicas que não são diretamente suportadas pelo Outsystems.
- **Personalização de interface limitada:** mesmo que o Outsystems ofereça ferramentas para construir interfaces de usuário, a liberdade de design pode ser limitada em comparação com o desenvolvimento manual.

Agora que conhecemos as vantagens e desvantagens de se trabalhar com Outsystems no frontend, vamos conhecer a forge, comunidade on-line e repositório oficial de componentes, aplicativos e recursos disponíveis para a plataforma Outsystems.

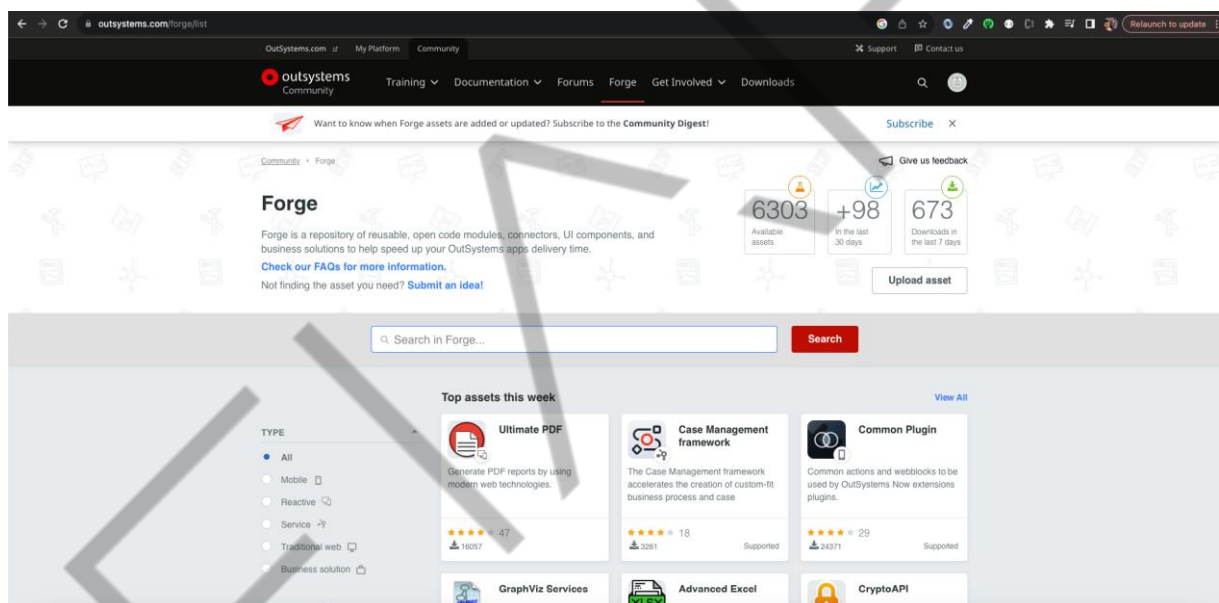


Figura 1 - Forge  
Fonte: elaborado pelo autor (2023)

Ela é uma espécie de "mercado" onde as pessoas desenvolvedoras podem compartilhar e baixar soluções prontas, acelerando o desenvolvimento de aplicativos.

A Forge oferece uma ampla gama de componentes prontos para uso, desde elementos de interface de usuário até integrações com serviços externos.

Além de componentes, você pode encontrar aplicativos de exemplo que demonstram o uso de funcionalidades específicas ou mostram boas práticas de desenvolvimento.

Existem templates e blocos de construção que ajudam a iniciar novos projetos ou a adicionar funcionalidades específicas rapidamente. A figura 2 – “Forge – lista de templates” tem uma lista de alguns desses templates:

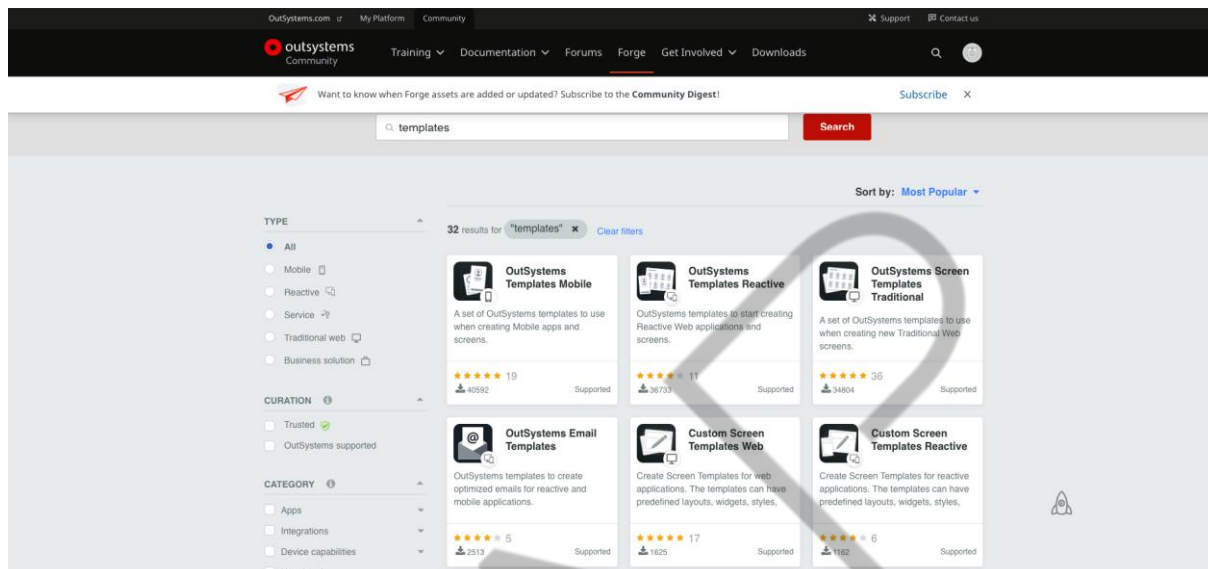


Figura 2 - Forge - lista de templates  
Fonte: Elaborado pelo autor (2023)

Além disso, você também pode consultar a lista diretamente no site da OutSystems. Lá é possível visualizar uma comunidade ativa onde os(as) desenvolvedores(as) compartilham suas criações, colaboram em projetos e oferecem suporte uns aos outros.

Cada componente ou aplicativo geralmente possui avaliações e comentários de outros(as) usuários(as), ajudando na escolha de soluções de alta qualidade.

A seguir, você tem uma imagem de um componente já testado, demonstrando a quantidade de reviews e de downloads dele:

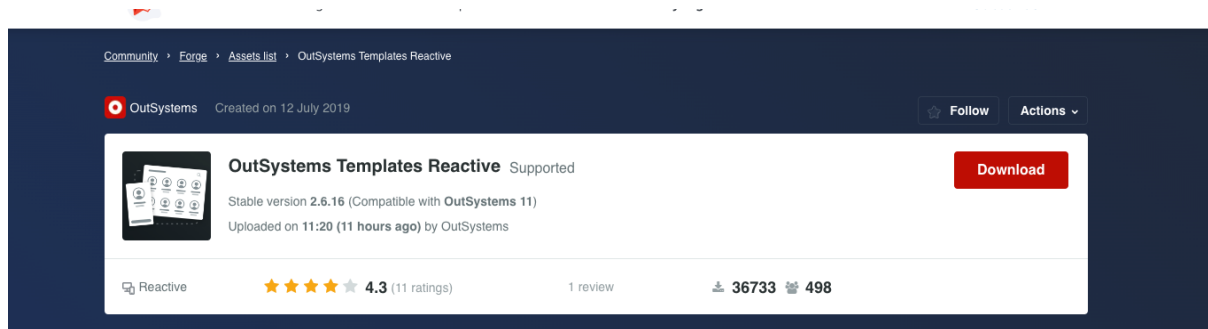


Figura 3 - Forge reviews  
Fonte: elaborado pelo autor (2023)



A Forge é atualizada regularmente com novos componentes e atualizações de versões existentes, refletindo a evolução contínua da plataforma.

### **Como usar a Forge?**

- Você pode pesquisar por categorias, palavras-chave ou funcionalidades específicas para encontrar o que precisa.
- Após encontrar um componente ou aplicativo desejado, é possível baixá-lo e incorporá-lo diretamente ao seu projeto Outsystems.
- Se você desenvolveu algo útil, pode contribuir para a Forge compartilhando seus próprios componentes ou aplicativos.
- Ao usar recursos da Forge, é sempre bom dar feedback, avaliar e compartilhar experiências para ajudar outros desenvolvedores e desenvolvedoras.

## O QUE VOCÊ VIU NESTA AULA?

Nesta aula, apresentamos a plataforma Outsystems. Você aprendeu sobre o ecossistema que compõe esta plataforma junto de um overview no Service Studio, a ferramenta de desenvolvimento da plataforma. Além disso, criamos um novo app passando pelos pontos básicos da plataforma, como:

- Criar um novo APP (Reactive).
- Importar backend.
- Montar todas as telas.
- Publicar o projeto e mostrar a tela de login.
- Cadastrar os dados.
- Manipular estilos.
- Utilizar JavaScript.
- Importar componente da Forge.

## REFERÊNCIAS

OutSystems. **Extend Your Mobile and Reactive Apps Using JavaScript**. Disponível em:

<[https://success.outsystems.com/documentation/11/extensibility\\_and\\_integration/javascript/extend\\_your\\_mobile\\_and\\_reactive\\_apps\\_using\\_javascript/](https://success.outsystems.com/documentation/11/extensibility_and_integration/javascript/extend_your_mobile_and_reactive_apps_using_javascript/)>. Acesso em: 21 mar. 2024.

OutSystems. **Front-End Architecture Best Practices**. Disponível em:

<[https://success.outsystems.com/documentation/best\\_practices/development/front\\_end\\_architecture\\_best\\_practices/](https://success.outsystems.com/documentation/best_practices/development/front_end_architecture_best_practices/)>. Acesso em: 21 mar. 2024.

OutSystems. **Use a Forge Component Made by the Community**. Disponível em:

<[https://success.outsystems.com/documentation/11/getting\\_started/use\\_a\\_forge\\_component\\_made\\_by\\_the\\_community/](https://success.outsystems.com/documentation/11/getting_started/use_a_forge_component_made_by_the_community/)>. Acesso em: 06 fev. 2024.

## **PALAVRAS-CHAVE**

Outsystems, Frontend, Forge.

EMANIP

# POSTECH