

Projeto Final

Módulo 1

Culture Code 🟡 🟣 🟠 🟢



Descrição

- **Culture Code** é uma aplicação de gamificação de uma empresa “Exemplo”
- No Projeto Final do Módulo 1 será desenvolvido um recorte dessa aplicação: a funcionalidade de **resgate de itens** a partir de pontos que o usuário possui
- Os dados deverão ser enviados e recuperados da API (A API será executada utilizando o **json-server**)
- As **únicas** bibliotecas externas permitidas no desenvolvimento do projeto são:
 - Biblioteca para os ícones: Font Awesome, Remix Icon , Icons8, Google Icons, Flat Icons, ou outra de preferência do aluno
- Data de entrega: **18/02/2024**
- Os arquivos do projeto deverão estar em um **repositório público no GitHub**
- A aplicação deverá ser publicada na **Vercel e a API no Render**
- **Qualquer tipo de plágio não será tolerado. Em caso de cópia, o aluno será imediatamente desligado do curso.**

Acompanhamento

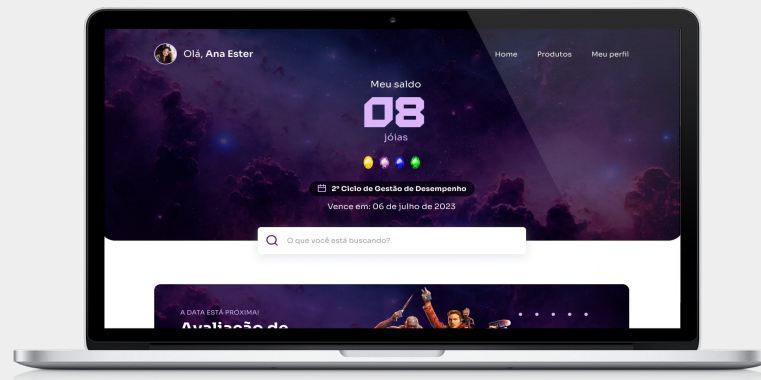
- Não haverá aula nesse período, haverão plantões de 19h às 20:30h para tirar dúvidas e acompanhar o andamento do projeto
- Um professor estará disponível para tirar dúvidas
- Outro professor conduzirá reuniões diárias, seguindo a metodologia SCRUM, chamadas de *daily meetings* de **presença obrigatória**
 - Para essas reuniões, grupos de alunos serão adicionados em uma sala a parte (um grupo por vez)
 - Cada aluno do grupo terá 2 minutos para falar o que fez no dia anterior e o que vai fazer neste
- Enquanto um professor conduz as *daily meetings*, outro estará na sala principal tirando dúvidas.
- Os encontros ocorrerão no link do Meet: [Sala de Aula](#)

Acompanhamento

- O primeiro passo para iniciar o projeto é criar o repositório no GitHub
- Até o dia **22/01/2024**, crie o repositório do GitHub que armazenará seu trabalho
- Publique as alterações feitas no código do projeto com frequência para evitar ter o trabalho perdido caso ocorra algum imprevisto!
- Os professores acompanharão esse repositório durante o mês de projeto

Layout: Figma

[Arquivo Aberto para acessar as especificações](#)



Funcionalidades

- Abertura
- Login
- Início
 - Menu superior
 - Seção de informações
 - Campo de pesquisa
 - Banner avaliação
 - Produtos
 - Resgate
- Meu perfil
 - Meus dados
 - Meus resgates
 - Minhas jóias
 - Sair

Funcionalidades Obrigatórias

- Abertura (*layout estático*)
- Login (*layout estático*)
- Início
 - Menu superior (*layout estático*)
 - Seção de informações (*layout estático*)
 - Campo de pesquisa (*layout estático*)
 - Banner avaliação (*layout estático*)
 - Produtos (*layout com a comunicação com a API*)
 - Resgate (*layout com a comunicação com a API*)
- Meu perfil
 - Meus dados (*layout estático*)
 - Meus resgates (*layout com a comunicação com a API*)
 - Minhas jóias (*layout estático*)
 - Sair (*layout estático*)

Funcionalidades Optativas

- Abertura (*layout estático*)
- Login (*layout estático*)
- Início
 - Menu superior (*layout estático*)
 - Seção de informações (*layout estático*)
 - Campo de pesquisa (*comunicação com a API*)
 - Banner avaliação (*layout estático*)
 - Produtos (*layout com a comunicação com a API*)
 - Resgate (*layout com a comunicação com a API*)
- Meu perfil
 - Meus dados (*layout estático*)
 - Meus resgates (*layout com a comunicação com a API*)
 - Minhas jóias (*layout estático*)
 - Sair (*layout estático*)

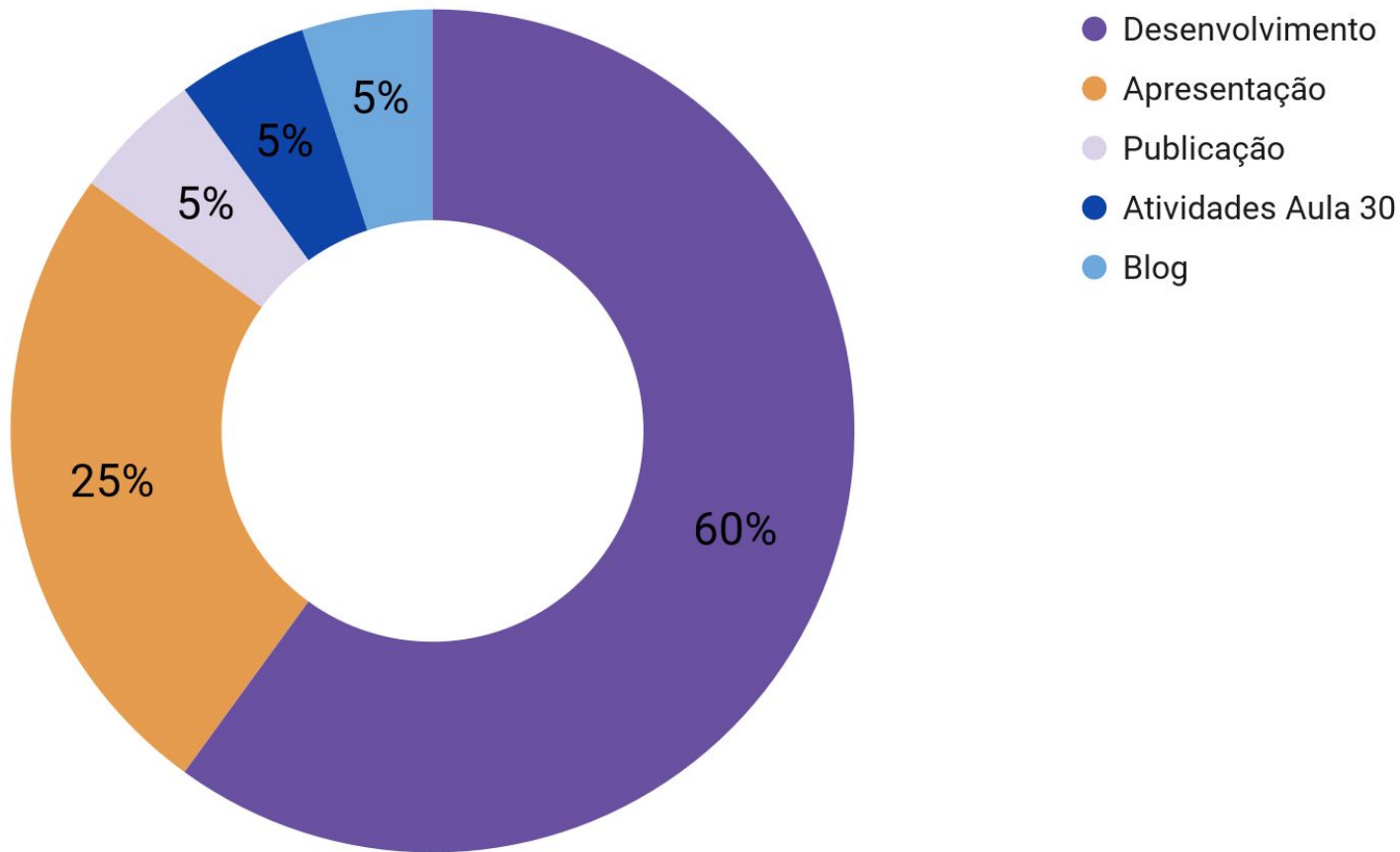
Detalhes da Entrega

- Até o dia **18/12/2024**, responda [este formulário](#) informando: nome completo, e-mail (para onde será enviada a nota final do módulo 1), link do repositório no GitHub e link da aplicação na Vercel
- **Só serão aceitos arquivos incluídos até 23:59h do dia 18/02/2024 (domingo)**

Avaliação do projeto

- A avaliação do projeto será feita com base na **apresentação individual** ao vivo feita pelo aluno, no **código-fonte** e na **aplicação publicada** na Vercel
- As apresentações ocorrerão entre os dias **19/02/2024** e **23/02/2024**
- Serão 5 dias de apresentações, os alunos serão divididos entre esses dias
- A ordem das apresentações será definida por sorteio
- O dia e horário em que cada aluno apresentará o projeto será informado previamente na última semana de projeto
- A aluno será aprovado no módulo se alcançar nota igual ou superior à 70%

Avaliação Módulo 1



**As atividades
obrigatórias
irão compor**

90%

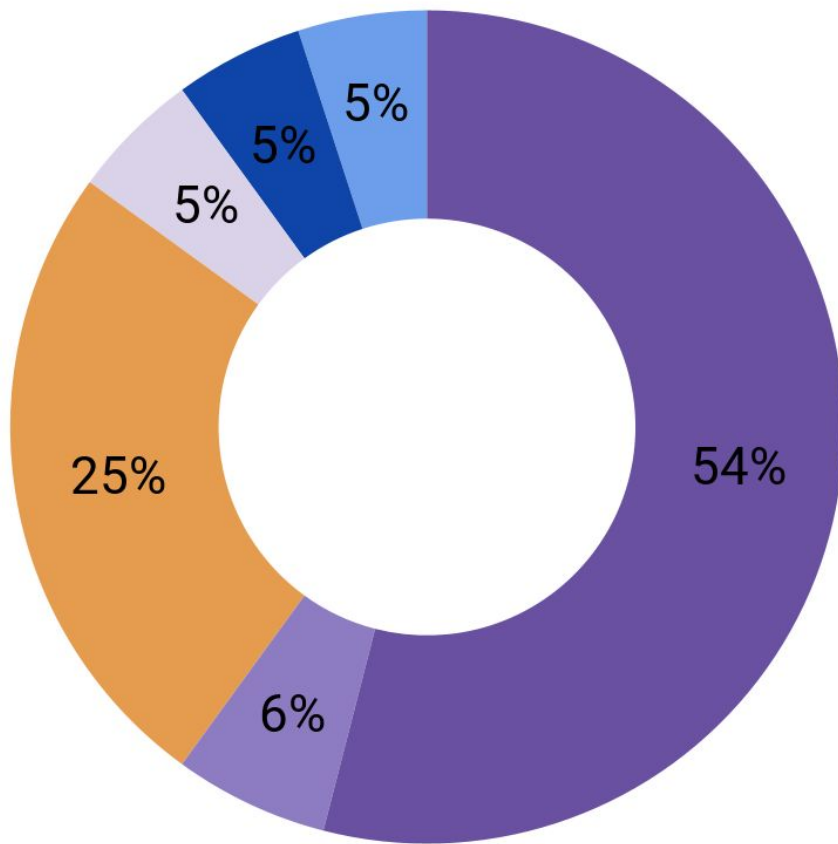
da nota do projeto

**As atividades
optativas
irão compor**

10%

da nota do projeto

Avaliação Módulo 1



- Desenvolvimento Obrigatório
- Desenvolvimento Optativo
- Apresentação
- Publicação
- Atividades Aula 30
- Blog

O que será avaliado no desenvolvimento

- Porcentagem de atividades desenvolvidas
- Fidelidade do *layout*
- Funcionamento
- Qualidade do código
- Organização do código e dos arquivos

Na apresentação

- Demonstre o funcionamento da sua aplicação
- Apresente o código-fonte explicando o fluxo utilizado para o desenvolvimento
- Tempo máximo: **15 minutos**
- Busque equilibrar o tempo entre explicar as funcionalidades e o código-fonte
- **Mostre domínio do que foi desenvolvido**

Detalhamento



Abertura

- Obrigatório
 - Desenvolver toda a parte visual com HTML e CSS
 - Ao clicar no botão “Fazer login” redirecionar para a Tela de Login
- Optativo
 - *Não se aplica*
- Sugestão Extra
 - Não se aplica

Login

- Obrigatório
 - Desenvolver toda a parte visual com HTML e CSS
 - Ao clicar no botão “Entrar” redirecionar para a Tela Inicial
- Optativo
 - *Não se aplica*
- Sugestão Extra
 - Validar o login consultando o usuário previamente incluído na API

Início

- Obrigatório
 - Desenvolver toda a parte visual com HTML e CSS
 - Desenvolver o **Menu superior** que será replicados para as outras páginas
 - O nome e foto do Usuário com dados fixos
 - Ao clicar no Item de Menu “Meu perfil” redirecionar para a página correspondente
 - A **Seção de informações** com dados fixos
 - O campo de **Pesquisa** com layout fixo
 - O **Banner** de avaliação deve ser inserido como imagem (resgatar do Figma)
- Optativo
 - O campo de **Pesquisa** deve filtrar os produtos da lista de acordo com a informação digitada e acionamento da tecla “Enter”
- Sugestão Extra
 - No **Menu superior**, o nome do Usuário deve ser do usuário logado
 - Armazenar o Saldo de jóias do usuário logado na API

Produtos

- Obrigatório
 - Desenvolver toda a parte visual com HTML e CSS
 - Listar os Produtos vindos de uma coleção da API
 - Ao clicar em no botão “Resgatar”, redirecionar para a tela de **Resgate** mostrando o produto selecionado
- Optativo
 - Não se aplica
- Sugestão Extra
 - Não se aplica

Resgate

- Obrigatório
 - Desenvolver toda a parte visual com HTML e CSS
 - Mostrar os dados do Produto vindos de uma consulta à API pelo **id**
 - Ao clicar em Resgatar salvar na API a operação de resgate com a informação do produto (nome, imagem e jóias) e horário
 - Redireciona para a tela de Produto resgatado
- Optativo
 - Não se aplica
- Sugestão Extra
 - Atualizar o Saldo de jóias do Usuário após o resgate

Produto resgatado

- Obrigatório
 - Desenvolver toda a parte visual com HTML e CSS
 - Mostrar os dados do Produto vindos de uma consulta à API pelo **id**
 - Ao clicar em Voltar à página inicial redireciona para o Início
- Optativo
 - Não se aplica
- Sugestão Extra
 - Em “Meu saldo” mostrar o Saldo de jóias real do Usuário após o resgate

Meus dados

- Obrigatório
 - Não se aplica
- Optativo
 - Desenvolver toda a parte visual com HTML e CSS
- Sugestão Extra
 - Mostrar usuário logado vindo da API

Meus resgates

- Obrigatório
 - Desenvolver toda a parte visual com HTML e CSS
 - Listar os Resgates realizados vindos de uma coleção da API
- Optativo
 - Não se aplica
- Sugestão Extra
 - Não se aplica

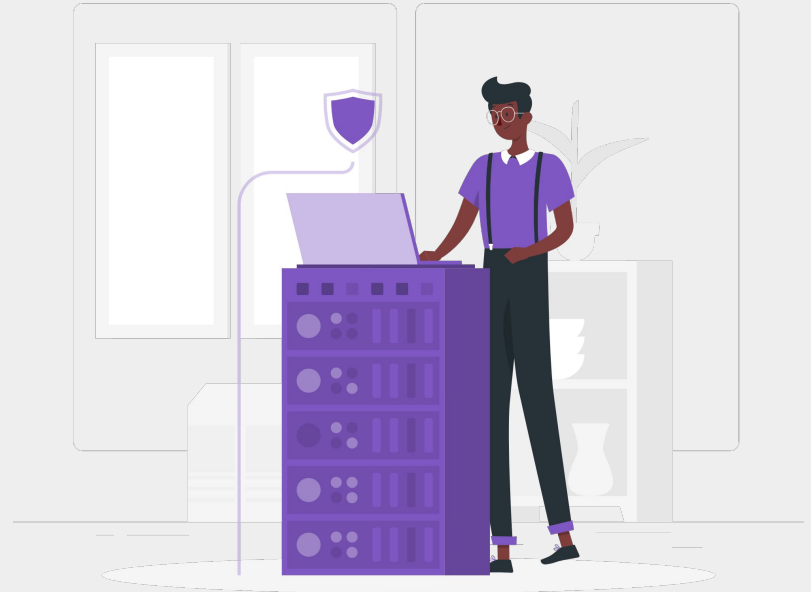
Minhas jóias

- Obrigatório
 - Não se aplica
- Optativo
 - Desenvolver toda a parte visual com HTML e CSS
- Sugestão Extra
 - Não se aplica

Sair

- Obrigatório
 - Ao clicar na opção “Sair”, redirecionar para a tela de Abertura
- Optativo
 - Não se aplica
- Sugestão Extra
 - Não se aplica

Publicação



Vercel

- A Vercel será utilizada para hospedar **a aplicação front-end**
- O código na sua aplicação deve estar em um repositório no GitHub
- Crie uma conta no site da Vercel: [Sign Up – Vercel](#)
- Ao entrar clique em **Add new > Project**
- Conecte o seu GitHub e escolha o repositório na lista e clique em **Import**
- Na próxima tela clique em **Deploy**
- A Vercel fornecerá a URL para acesso à sua aplicação

Render

- O Render poderá ser utilizado para hospedar **a API** (a parte do json-server)
- Clone este repositório para o seu GitHub clicando na opção **Use this template:**
<https://github.com/jesperorb/json-server-heroku>
- Defina no arquivo db.json o conteúdo base. Exemplo:

```
{  
  "produtos": [],  
  "resgates": [],  
}
```

- Entre no site do Render: [Render](#)
- Clique **GET STARTED FOR FREE > Sign up for Render [GitHub]**
- Faça o login no seu GitHub

Render

- Clique em **New Web Service**
- Do lado direito da página, clique em **Configure account** abaixo do símbolo do GitHub, selecione a sua conta e clique em **Install** para vinculá-la ao Render
- Escolha o repositório que deseja publicar (o criado anteriormente) e clique em **Connect**
- Defina um nome para o *web service* e ao final da página clique em **Create Web Service**
- Após o sucesso da publicação o link localizado abaixo do nome do *web service* poderá ser usado
- Use o link gerado na sua aplicação front-end substituindo o <http://localhost:3000>

**A nota final do
Módulo 1 será
entregue até o
dia 23/02/2024**

**através do e-mail informado
no formulário de entrega*



Sugestões



Sugestões para implementações extras

- Comunicação do Login com a API
- Comunicação do Meu Perfil com a API
- Responsividade
- Paginação da listagem
- Auto-contraste
- **Documentação JSON-SERVER:** <https://github.com/typicode/json-server#routes>

Obrigado!

