

# Trabalho Prático 2 - Site Pessoal integrado com API do GitHub e JSONServer

Iniciar tarefa

- Vencimento Quarta-feira por 23:59
- Pontos 30
- Enviando um URL de site ou um upload de arquivo
- Tipos de arquivo zip
- Disponível 27 mai em 0:00 - 26 jun em 23:59

## Introdução

Nesse trabalho, vamos dar vida ao layout desenvolvido no trabalho anterior, apresentando informações reais sobre as atividades realizadas por você no ambiente do **GitHub** e outras informações obtidas OBRIGATORIAMENTE a partir de um servidor JSON Server que oferece uma API e servirá como back end da aplicação.

Você pode evoluir o layout do Trabalho Prático 1 ou começar um projeto de layout do zero. Essa decisão fica a cargo de cada um. Você deverá montar um layout conforme o que foi fornecido para o Trabalho Prático 1 e apresentar funcionalidades que apresentamos neste enunciado.

Para montar a aplicação a ser entregue, você deverá utilizar como base o [projeto JSONServer disponível no Replit](https://replit.com/@rommelpuc/JSONServer) [↗\(https://replit.com/@rommelpuc/JSONServer\)](https://replit.com/@rommelpuc/JSONServer) e se cadastrar no GitHub para que possa fazer a integração, visualizando seus repositórios.

**IMPORTANTE: O trabalho deve ter foco exclusivamente na construção de um site pessoal com informações obtidas do GitHub e do JSONServer. NÃO SERÃO ACEITOS TRABALHOS COM OUTRA TEMÁTICA E COM OUTRAS API.**

## JSONServer

O JSONServer oferece uma API baseada no padrão RESTful para acesso e manipulação de dados dispostos em um arquivo JSON no servidor. [Um exemplo funcional do JSONServer](https://replit.com/@rommelpuc/JSONServer) [↗\(https://replit.com/@rommelpuc/JSONServer\)](https://replit.com/@rommelpuc/JSONServer) está acessível no ambiente do Replit e recomendamos fortemente que você utilize como base para o seu trabalho. Para isso, basta clonar (procedimento de FORK no Replit) e alterar o arquivo db.json, colocando os seus dados.

## API do GitHub

Dentre as funcionalidades providas pela **API do GitHub**, destacamos os seguintes recursos que serão utilizados nesse projeto:

- [Informações do Usuário \(https://docs.github.com/pt/rest/reference/users#get-a-user\)](https://docs.github.com/pt/rest/reference/users#get-a-user) - fornece as informações públicas de um usuário do GitHub;
- [Repositórios do usuário \(https://docs.github.com/pt/rest/reference/repos#list-repositories-for-a-user\)](https://docs.github.com/pt/rest/reference/repos#list-repositories-for-a-user) - fornece uma lista dos repositórios de um usuário;

Navegue pela [documentação da API do GitHub \(https://docs.github.com/pt/rest/reference\)](https://docs.github.com/pt/rest/reference) e encontre outras informações que você poderá utilizar na montagem do seu site pessoal.

## Estruturas de Dados

Você deve estruturar os dados para a montagem da sua aplicação. Utilize o projeto de exemplo do JSONServer, faça uma cópia para o seu trabalho. No arquivo **db.json**, você deve criar as suas estruturas de dados.

Para a montagem do seu trabalho, deve haver, pelo menos, três estruturas distintas: (1) álbuns, (2) fotos e (3) destaques. As informações básicas e OBRIGATÓRIAS de cada uma das estruturas mencionadas são apresentadas a seguir

### ESTRUTURA CONTEÚDOS SUGERIDOS

- Identificador do conteúdo (número sequencial)
- Título do conteúdo
- Descrição do conteúdo
- URL da imagem de capa do álbum
- URL do conteúdo

### ESTRUTURA COLEGAS DE TRABALHO

- Identificador do colega (número sequencial)
- URL da foto do colega
- Nome do colega
- URL do perfil do colega no GitHub

**IMPORTANTE:** Caso você não possua repositórios no GitHub, poderá optar por utilizar uma estrutura adicional de REPOSITÓRIOS no JSONServer para armazenar uma lista de repositórios sugeridos que serão exibidos na parte do layout que corresponde aos repositórios do GitHub. Dessa forma, caso opte por essa estrutura, você deverá montar a estrutura de repositórios no JSONServer e inserir os itens que serão exibidos no seu site. Ainda assim, você deverá cadastrar sua conta no GitHub para exibir as informações da primeira parte do layout.

# Detalhamento do escopo

O trabalho deve contemplar as seguintes telas e suas funcionalidades descritas em seguida.

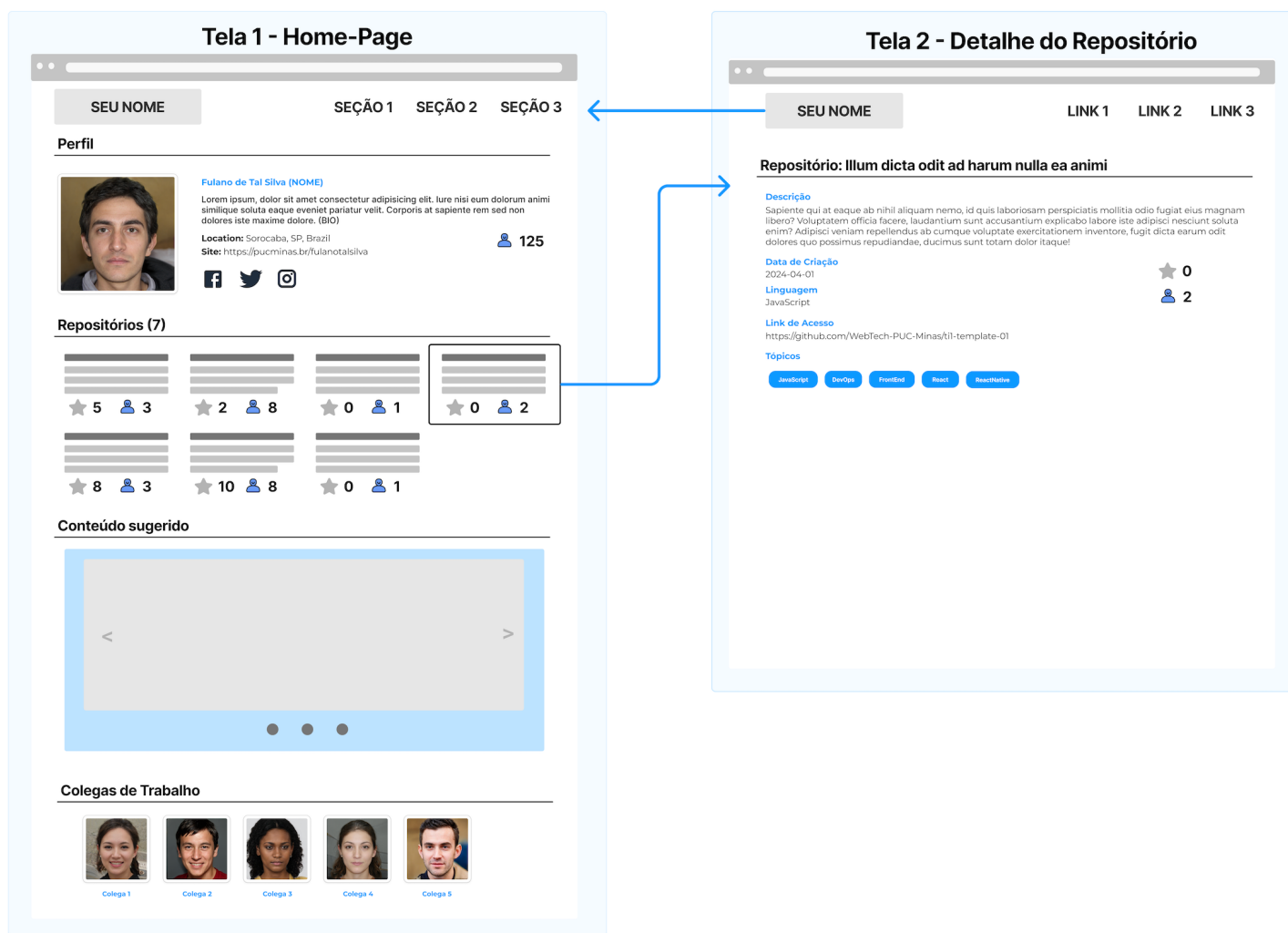
- **Tela principal (index.html)**

- Cabeçalho com seu nome e menu com outras seções do site
- **Seção 1 - Perfil do Usuário** - apresentação do seu perfil no GitHub com foto/avatar, nome e sobrenome, descrição, botões de contato (por exemplo, e-mail, Linkedin, etc)  
⚠ Os dados a serem exibidos devem ser obtidos diretamente da API do GitHub.
- **Seção 2 - Repositórios** - grupo de cards com informações de repositórios mantidos pelo aluno no GitHub ou cadastrados em uma estrutura alternativa no JSONServer. **(Pelo menos 3 itens)**  
ORIENTAÇÃO: use [componente do bootstrap \(https://getbootstrap.com/docs/5.0/components/card/#grid-cards\)](https://getbootstrap.com/docs/5.0/components/card/#grid-cards) (detalhes no link)  
⚠ Os dados a serem exibidos devem ser obtidos diretamente da API do GitHub ou via estrutura no JSONServer.
- **Seção 3 - Conteúdo sugerido** - carrossel/slider com conteúdos e respectivas imagens sugeridos pelo aluno (Artigos, vídeos, infográficos, etc) **(Pelo menos 5 itens)**  
ORIENTAÇÃO: use [componente do bootstrap \(https://getbootstrap.com/docs/5.0/components/carousel/#with-captions\)](https://getbootstrap.com/docs/5.0/components/carousel/#with-captions) (detalhes no link)  
⚠ Os dados a serem exibidos devem ser obtidos de estrutura no JSONServer.
- **Seção 4 - Colegas de Trabalho** - grade com apresentação dos principais colegas de trabalho no curso. **(Pelo menos 3 itens)**  
⚠ Os dados a serem exibidos devem ser obtidos de estrutura no JSONServer.
- Rodapé com informações sobre responsável pelo site (aluno/aluna)

- **Tela de detalhes do REPOSITÓRIO (repo.html)**

- Página única que apresenta, de forma dinâmica, as informações de um **REPOSITÓRIO** específico, cujo código será passado por parâmetro via query string. As informações do repositório devem ser obtidas por meio de API do GitHub e devem contemplar o seguinte: descrição, data de criação, proprietário com imagem/avatar, url de acesso, linguagem principal, lista de tópicos (tags), quantidade de estrelas, observadores e forks, tipo de licença.
- A passagem do parâmetros deve acontecer conforme o seguinte exemplo: **http://servidor/repo.html?id=xxxx**, onde **xxxx** representa o identificador do repositório no GitHub a ser utilizado para recuperação das informações a serem exibidas.

O fluxo de telas que deve ser implementado na aplicação entregue e os wireframes com os detalhes do escopo são apresentados na imagem a seguir.



## Entrega

Você deve, OBRIGATORIAMENTE, enviar um arquivo compactado no formato ZIP com os arquivos necessários para a execução do site. Você poderá também, em caráter ADICIONAL e OPCIONAL, enviar um link para o site publicado no Replit. A entrega do arquivo ZIP é obrigatória e deve conter a seguinte estrutura, conforme projeto JSONServer, disponibilizado no Replit.

```
/db
  db.json          ----- Arquivo com as estruturas de dados e os respectivos dados da aplicação
/public            ----- Pasta onde ficam os arquivos da parte frontend do projeto
  index.html
  repo.html
  /assets
    /css
      styles.css
    /img
      img1.jpg
      img2.jpg
      img3.jpg
    /scripts
```

```
app.js
index.js      ----- Arquivo utilizado para executar o JSONServer de forma alternativa
package.json  ----- Configuração do projeto, utilizado pelo node.js na parte de backend
.gitignore    ----- Arquivo de configuração do git com lista de arquivos a serem ignorados
```

A orientação necessária para executar e testar o JSONServer e a parte de front end no seu próprio computador está disponível na forma de uma videoaula. Verifique para facilitar o processo de desenvolvimento. Você também tem a opção de fazer todo o processo de desenvolvimento e testes a partir do Replit. Isso fica a seu critério.

## Avaliação por pares

O trabalho será avaliado no esquema de avaliação por pares. **Nesse processo, você deve avaliar 3 projetos de outros alunos e isso é obrigatório e impacta na sua pontuação desse trabalho.**

**IMPORTANTE: Os trabalhos devem ter identidade visual própria e código fonte original. Cada aluno deve se identificar na Home-Page e fazer o seu trabalho de forma independente. Será realizada uma verificação de originalidade do código fonte e os trabalhos que tiverem partes significativas de código copiado dos demais colegas serão penalizados na nota.**

Para avaliar os trabalhos e atribuir pontos, siga os critérios apresentados abaixo. Seja honesto na avaliação. A nota final será a média das avaliações dos 3 outros colegas, sendo cancelada pelos professores envolvidos.

### CRITÉRIO 1: Formato e responsividade (3 pontos)

- 3 pontos - **Responsividade funcional:** O site apresenta adaptação e adequação de conteúdos para a versão mobile e desktop. Componentes se adaptam ao tamanho, imagem segue o padrão do grid, conteúdo ajusta a resolução com uso correto de media queries.
- 1 pontos - **Responsividade parcial:** Somente alguns componentes ou elementos HTML se adaptam ao formato móvel, mas o sistema de Grid do Bootstrap está funcional no Desktop.
- 0 pontos - **Sem responsividade** - O site não apresenta recursos de responsividade.

### CRITÉRIO 2: Perfil do usuário no GitHub (**index.html**) (5 pontos)

- 5 pontos - **Seção de perfil dinâmica com dados do perfil carregados via API do GitHub**  
Perfil do usuário com carga dinâmica dos dados, integrada com a API do GitHub, e apresentando: imagem do perfil, nome do usuário, descritivo do perfil (bio), localização geográfica do usuário, site do usuário, links das redes sociais e indicador do número de seguidores;
- 2 ponto - **Seção de perfil dinâmica, porém incompleta ou com alguns defeitos.**  
Perfil do usuário com carga dinâmica dos dados, integrada com a API do GitHub, incompleto em

termos dos dados apresentados no layout proposto ou com alguns problemas de funcionamento.

- 0 ponto - **Seção do perfil do usuário apresentada de forma estática**

Perfil do usuário não integrado com API do GitHub ou apresentando mensagens de erro.

### CRITÉRIO 3: Cards de repositórios do GitHub (**index.html**) (5 pontos)

- 5 pontos - **Seção de repositórios dinâmica com dados do GitHub ou JSONServer**

Relação dos repositórios com carga dinâmica dos dados, integrada com a API do GitHub ou com dados obtidos via JSONServer, e apresentando: nome do repositório, descrição, número de estrelas e número de forks;

- 2 ponto - **Seção de repositórios dinâmica, porém incompleta ou com alguns defeitos.**

Relação dos repositórios com carga dinâmica dos dados, integrada com a API do GitHub, incompleta em termos dos dados apresentados no layout proposto ou com alguns problemas de funcionamento.

- 0 ponto - **Seção do repositórios apresentada de forma estática**

Relação dos repositórios não integrado com API do GitHub ou apresentando mensagens de erro.

### CRITÉRIO 4: Carrousel de conteúdos sugerido (**index.html**) (5 pontos)

- 5 pontos - **Seção de conteúdos dinâmica com dados do JSONServer**

Carrousel de conteúdos com carga dinâmica dos dados, integrada com JSONServer, e apresentando: imagem, título, descrição e com link funcional direcionando para página do conteúdo na Internet;

- 2 ponto - **Seção de conteúdos dinâmica, porém incompleta ou com alguns defeitos.**

Carrousel de conteúdos com carga dinâmica dos dados, integrada com JSONServer, incompleta em termos dos dados apresentados no layout proposto ou com alguns problemas de funcionamento.

- 0 ponto - **Seção do conteúdos apresentada de forma estática**

Carrousel de conteúdos não integrado com JSONServer ou apresentando mensagens de erro.

### CRITÉRIO 5: Lista de colegas de trabalho (**index.html**) (5 pontos)

- 5 pontos - **Seção de conteúdos dinâmica com dados do JSONServer**

Lista de colegas com carga dinâmica dos dados, integrada com JSONServer, e apresentando: imagem, título, descrição e com link funcional direcionando para perfil no GitHub;

- 2 ponto - **Seção de conteúdos dinâmica, porém incompleta ou com alguns defeitos.**

Lista de colegas com carga dinâmica dos dados, integrada com JSONServer, incompleta em termos dos dados apresentados no layout proposto ou com alguns problemas de funcionamento.

- 0 ponto - **Seção do conteúdos apresentada de forma estática**

Lista de colegas não integrada com JSONServer ou apresentando mensagens de erro.

## **CRITÉRIO 6: Página de detalhes do repositório (repo.html) (7 pontos)**

- 7 pontos - **Página de detalhes completa, totalmente funcional e integrada com GitHub**

Página de detalhes com passagem de parâmetro corretamente implementada na home-page (index.html), com recepção do parâmetro correta na página de detalhes (repo.html), carga dinâmica dos dados a partir da API do GitHub e apresentando todas as informações apresentadas no layout proposta: título, descrição, data de criação, linguagem, link de acesso, tópicos, quantidade de estrelas e forks, bem como link funcional direcionando para página no GitHub;

- 3 pontos - **Página de detalhes dinâmica, porém incompleta ou com alguns defeitos.**

Página de detalhes com passagem de parâmetro corretamente implementada na home-page (index.html), com recepção do parâmetro correta na página de detalhes (repo.html), carga dinâmica dos dados a partir da API do GitHub, porém incompleta em termos dos dados apresentados no layout proposto ou com alguns problemas de funcionamento.

- 0 ponto - **Página de detalhes apresentada de forma estática**

Página de detalhes não integrada com GitHub ou apresentando mensagens de erro.

### **Rubrica - Trabalho Prático 2 (2024-1)**



Critérios	Avaliações			Pts
Formato e Responsividade	<b>3 pts</b> <b>Responsividade funcional</b> O site apresenta adaptação e adequação de conteúdos para a versão mobile e desktop Componentes se adaptam ao tamanho, imagem segue o padrão do grid, conteúdo ajusta a resolução com uso correto de media queries;	<b>1 pts</b> <b>Responsividade parcial</b> Somente alguns componentes ou elementos HTML se adaptam ao formato móvel, mas o sistema de Grid do Bootstrap está funcional no Desktop;	<b>0 pts</b> <b>Sem responsividade</b> O site não apresenta recursos de responsividade.	3 pts
Perfil do usuário no GitHub (index.html)	<b>5 pts</b> <b>Seção de perfil dinâmica com dados do perfil carregados via API do GitHub</b> Perfil do usuário com carga dinâmica dos dados, integrada com a API do GitHub, e apresentando: imagem do perfil, nome do usuário, descritivo do perfil (bio),	<b>2 pts</b> <b>Seção de perfil dinâmica, porém incompleta ou com alguns defeitos</b> Perfil do usuário com carga dinâmica dos dados, integrada com a API do GitHub, incompleto em termos dos dados apresentados no layout	<b>0 pts</b> <b>Seção do perfil do usuário apresentada de forma estática</b> Perfil do usuário não integrado com API do GitHub ou apresentando mensagens de	5 pts
Cards de repositórios do GitHub (index.html)	<b>5 pts</b> <b>Seção de repositórios dinâmica com dados do GitHub da JSONServer</b> Relação dos repositórios com carga dinâmica dos dados, integrada com a API do GitHub ou com dados obtidos via JSONServer, e	<b>2 pts</b> <b>Seção de repositórios dinâmica, porém incompleta ou com alguns defeitos</b> Relação dos repositórios com carga dinâmica dos dados, integrada com a API do GitHub, incompleta em	<b>0 pts</b> <b>Seção do repositórios apresentada de forma estática</b> Relação dos repositórios não integrado com API do GitHub ou	5 pts
Carrousel de conteúdos sugerido (index.html)	<b>5 pts</b> <b>Seção de conteúdos dinâmica com dados do JSONServer</b> Carrousel de conteúdos com carga dinâmica dos dados, integrada com JSONServer, e apresentando: imagem, título, descrição e com link	<b>2 pts</b> <b>Seção de conteúdos dinâmica, porém incompleta ou com alguns defeitos</b> Carrousel de conteúdos com carga dinâmica dos dados, integrada com JSONServer, incompleta em termos dos	<b>0 pts</b> <b>Seção de conteúdos apresentada de forma estática</b> Carrousel de conteúdos não integrado com JSONServer ou	5 pts
Lista de colegas de trabalho (index.html)	<b>5 pts</b> <b>Seção de conteúdos dinâmica com dados do JSONServer</b> Lista de colegas com carga	<b>2 pts</b> <b>Seção de conteúdos dinâmica, porém incompleta ou com alguns defeitos</b>	<b>0 pts</b> <b>Seção de conteúdos apresentada de forma estática</b>	5 pts



Critérios	Avaliações			Pts
	dinâmica dos dados, integrada com JSONServer, e apresentando: imagem, título, descrição e com link funcional direcionando para perfil no GitHub;	Lista de colegas com carga dinâmica dos dados, integrada com JSONServer, incompleta em termos dos dados apresentados no layout proposto ou com alguns problemas de funcionamento	Lista de colegas não integrada com JSONServer ou apresentando mensagens de erro.	
Página de detalhes do repositório (repo.html)	<b>7 pts</b> <b>Página de detalhes completa, totalmente funcional e integrada com GitHub</b> Página de detalhes com passagem de parâmetro corretamente implementada na home-page (index.html), com recepção do parâmetro correta na página de detalhes (repo.html), carga dinâmica dos dados a partir da API do GitHub e apresentando todas as informações apresentadas no layout proposta: título, descrição, data de criação, linguagem, link de acesso, tópicos, quantidade de estrelas e forks, bem como link funcional direcionando para página no GitHub;	<b>3 pts</b> <b>Página de detalhes dinâmica, porém incompleta ou com alguns defeitos</b> Página de detalhes com passagem de parâmetro corretamente implementada na home-page (index.html), com recepção do parâmetro correta na página de detalhes (repo.html), carga dinâmica dos dados a partir da API do GitHub, porém incompleta em termos dos dados apresentados no layout proposto ou com alguns problemas de funcionamento.	<b>0 pts</b> <b>Página de detalhes apresentada de forma estática</b> Página de detalhes não integrada com GitHub ou apresentando mensagens de erro.	7 pts
Total de pontos: 30				