

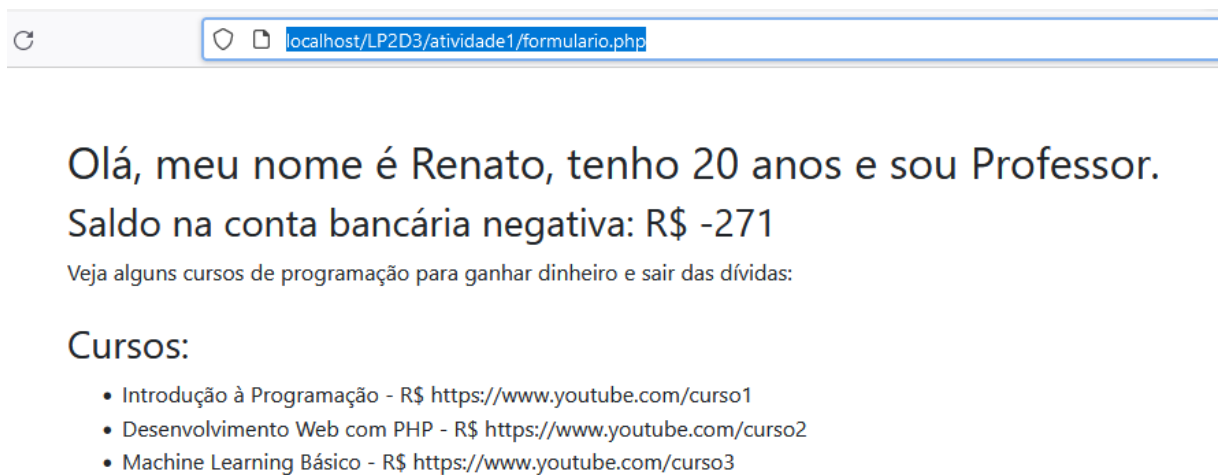
Crie um formulário com os campos nome, idade e profissão e um botão enviar, conforme figura abaixo:



A screenshot of a web browser window. The address bar shows 'localhost/LP2D3/atividade1/formulario.php'. The form contains three input fields: 'Nome:' with the value 'Renato Bueno', 'Idade:' with the value '20', and 'Profissão:' with the value 'Professor'. Below the fields is a blue button labeled 'Entrar'.

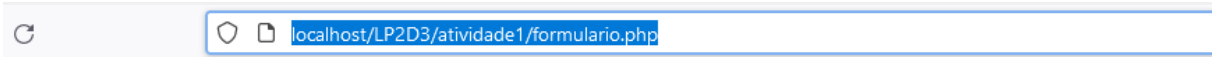
Ao ser clicado no botão enviar será mostrada a seguinte tela dependendo do saldo em conta:

- Para o saldo em conta negativo:



A screenshot of a web browser window showing a message. The address bar shows 'localhost/LP2D3/atividade1/formulario.php'. The message reads: 'Olá, meu nome é Renato, tenho 20 anos e sou Professor. Saldo na conta bancária negativa: R\$ -271'. Below this, it says 'Veja alguns cursos de programação para ganhar dinheiro e sair das dívidas:'. A section titled 'Cursos:' follows, with a list of three items: 'Introdução à Programação - R\$ https://www.youtube.com/curso1', 'Desenvolvimento Web com PHP - R\$ https://www.youtube.com/curso2', and 'Machine Learning Básico - R\$ https://www.youtube.com/curso3'.

- Para o saldo em conta positivo



Olá, meu nome é Renato, tenho 20 anos e sou Professor.  
Seu saldo da conta bancária é de: R\$ 9602

Veja algumas ofertas de produtos para você e alguns cursos:

### Cursos:

- Introdução à Programação - R\$ <https://www.youtube.com/curso1>
- Desenvolvimento Web com PHP - R\$ <https://www.youtube.com/curso2>
- Machine Learning Básico - R\$ <https://www.youtube.com/curso3>

### Produtos com desconto:

- Produto A - R\$ 50
- Produto B - R\$ 75
- Produto C - R\$ 120

*Para o saldo em conta bancária utilizem uma função para randomizar o valor como a função **rand();***

### Exercício 1: Classe Pessoa

Crie uma classe chamada `Pessoa` com os atributos `nome`, `idade` e `profissao`. Implemente um método chamado `apresentar` que retorna uma string no formato "Olá, meu nome é [nome], tenho [idade] anos e sou [profissao].".

Instruções:

1. Cada estudante deve criar uma classe `Pessoa` com atributos únicos (nomes, idades e profissões diferentes).
2. O método `apresentar` deve ser implementado de forma personalizada para cada objeto.
3. Os estudantes podem compartilhar suas soluções ao final da aula para comparar as diferentes abordagens.

### Exercício 2: Conta Bancária

Crie uma classe chamada `ContaBancaria` com os atributos `titular` (nome do titular), `numero` (número da conta) e `saldo`. Implemente métodos para realizar depósito e saque na conta. Ao realizar o saque, verifique se o saldo é suficiente para a operação.

Instruções:

1. Cada estudante deve criar sua própria classe `ContaBancaria` com atributos exclusivos.
2. Implemente os métodos de depósito e saque de acordo com a lógica de cada estudante.
3. Cada estudante deve realizar algumas operações de depósito e saque em sua conta.

### **Exercício 3: Catálogo de Produtos**

Crie uma classe chamada `Produto` com os atributos `nome` e `preco`. Cada estudante deve criar um array de objetos da classe `Produto`, representando um catálogo de produtos de uma loja. Em seguida, crie uma função que exiba o nome e o preço de cada produto do catálogo.

Instruções:

1. Cada estudante deve criar sua própria classe `Produto` com atributos exclusivos (nomes e preços diferentes).
2. Cada estudante deve criar um catálogo personalizado de produtos usando um array de objetos.
3. Implemente uma função que percorra o array de produtos e exiba as informações de cada produto.

Dessa forma, cada estudante terá a oportunidade de criar suas próprias classes e soluções únicas para os exercícios, promovendo a compreensão individual dos conceitos da Programação Orientada a Objetos.

### **Exercício 4: Catálogo de cursos**

Crie uma classe chamada `Curso` com os atributos `nome` e `link`. Cada estudante deve criar um array de objetos da classe `Curso`, representando um catálogo de cursos do youtube. Em seguida, crie uma função que exiba o nome e o link de cada curso do youtube.

Instruções:

1. Cada estudante deve criar sua própria classe `Curso` com atributos exclusivos (nomes e links diferentes).
2. Cada estudante deve criar um catálogo personalizado de cursos usando um array de objetos.
3. Implemente uma função que percorra o array de cursos e exiba as informações de cada curso.

Dessa forma, cada estudante terá a oportunidade de criar suas próprias classes e soluções únicas para os exercícios, promovendo a compreensão individual dos conceitos da Programação Orientada a Objetos.

### **Exercício 5: Tela de cadastro e login**

Criar uma tela de cadastro e login do sistema que vocês deverão entregar durante o semestre. Utilizem o modelo de projeto em <https://github.com/renatobdo/LP2D3/tree/main/Projetos> com a tela de cadastro e login que estamos criando durante as aulas. Modifiquem o frontend!

**Obs.:** Os exercícios devem ser entregues via **arquivo zipado** no **moodle** e o link do seu repositório do **git**. Enviem os dois tanto o arquivo zipado como o link do git. Podem colocar um arquivo txt com o link do git dentro do arquivo zipado.

**Formato de envio:** **GU21579x\_nome\_sobrenomedoaluno.zip**