

## **RELATÓRIO DO PROJETO DE MODELAGEM DE SISTEMAS - API REST**

Esta API é uma plataforma inovadora que visa promover a participação e o engajamento das crianças por meio de um sistema de pontuação personalizado. Os pais ou responsáveis podem utilizar essa API para criar um ambiente divertido e motivador, onde as crianças são incentivadas a realizar tarefas domésticas e educativas, enquanto acumulam pontos e conquistam recompensas.

A funcionalidade central da API é a capacidade de criar e gerenciar tarefas. Os pais têm a liberdade de adicionar uma variedade de tarefas adequadas à idade e ao desenvolvimento de seus filhos. Essas tarefas podem variar desde atividades simples do dia a dia, como arrumar o quarto ou lavar a louça, até tarefas educativas, como ler um livro ou resolver problemas matemáticos. Cada tarefa é atribuída a uma pontuação específica, levando em consideração seu grau de dificuldade. Isso permite que as crianças percebam o valor e o esforço envolvidos em cada tarefa.

Além de criar tarefas, os pais podem definir prazos para a conclusão de cada uma delas. Isso ajuda a promover um senso de responsabilidade e gerenciamento do tempo nas crianças, enquanto elas trabalham para atender aos prazos estabelecidos. Um marcador em formato de barra de progresso é exibido nas atividades, mostrando o tempo restante para o prazo da tarefa. Dessa forma, as crianças têm uma visão clara do progresso e são incentivadas a concluir as tarefas dentro do prazo estipulado.

À medida que as crianças concluem as tarefas, suas pontuações são automaticamente atualizadas e registradas pela API. Os pais podem acompanhar o progresso de seus filhos, visualizando as pontuações acumuladas. Com base nas pontuações, as crianças podem resgatar recompensas pré-definidas pelos pais. As recompensas são projetadas para serem atraentes e proporcionais aos esforços das crianças. Por exemplo, recompensas de maior valor exigem um número maior de pontos acumulados. Esse sistema de recompensas estimula as crianças a se esforçarem e a persistirem na realização das tarefas.

A API oferece recursos adicionais para aprimorar a experiência dos usuários. Os pais recebem notificações sempre que uma tarefa é concluída ou uma recompensa é resgatada. Essas notificações permitem que os pais acompanhem de perto o progresso de seus filhos e tomem as medidas necessárias para entregar as recompensas.

## 1 - Escopo da API REST:

**Tela Inicial:** A API começa com uma tela de login que utiliza um usuário padrão para autenticação.

**Responsável:** Os responsáveis têm acesso para designar tarefas às crianças, definindo o prazo, o grau de dificuldade e a quantidade de pontos que cada tarefa vale. Por exemplo, tarefas difíceis podem valer 50 pontos, médias 25 e fáceis 10.

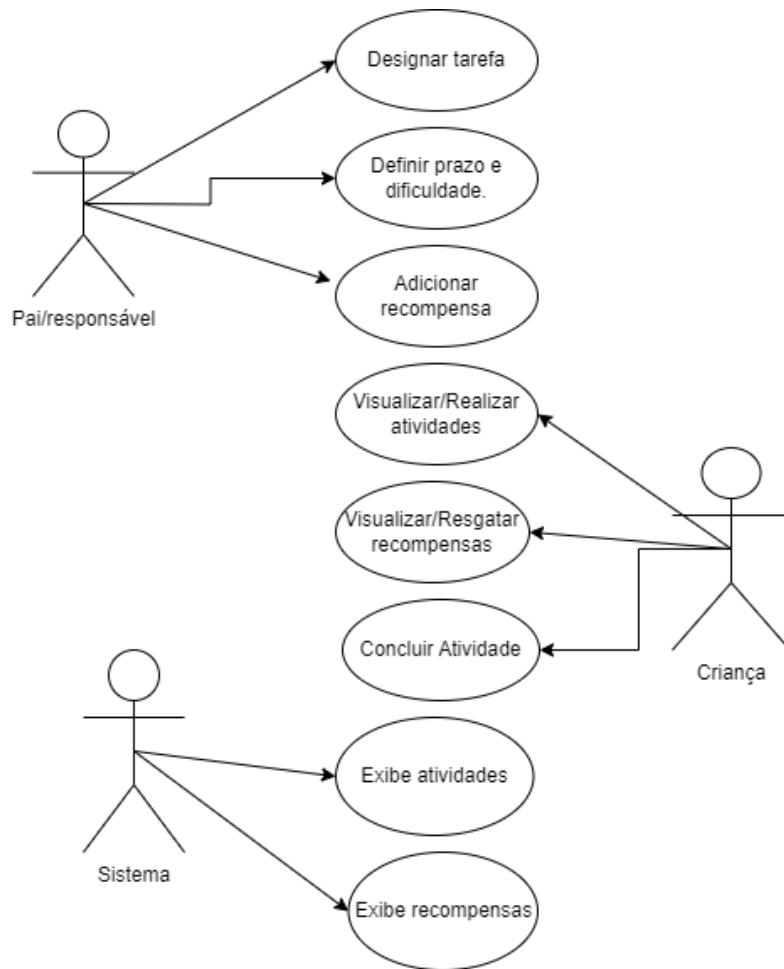
**Aluno:** As crianças podem acessar apenas as abas de atividades e recompensas para visualizá-las.

**Aba de Atividades:** A aba de atividades exibe a descrição, prazo, grau de dificuldade e pontos de recompensa associados a cada tarefa. Haverá também uma barra de progresso mostrando o tempo restante para a conclusão da tarefa. Nessa aba, as atividades já concluídas também serão exibidas.

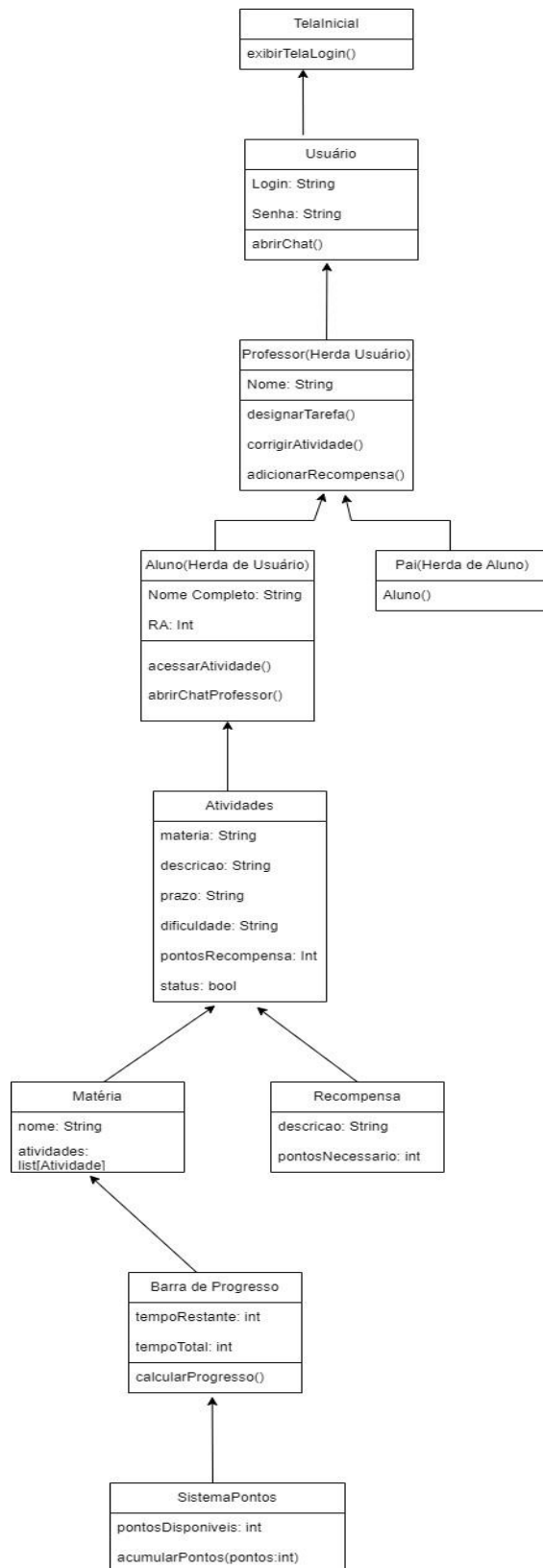
**Aba de Pontos:** Os pontos estarão disponíveis no cabeçalho para visualização, e à medida que as atividades forem concluídas, os pontos serão acumulados até que uma recompensa seja resgatada.

Todas as recompensas serão mostradas, mas apenas aquelas com uma pontuação menor do que a pontuação acumulada pela criança estarão disponíveis para resgate. Quando uma recompensa for resgatada, os responsáveis serão notificados para que possam entregá-la.

## 2.0 - Diagrama de Caso de Uso:



## 2.1 - Diagrama de Classes:



## **2. 2 - Classes:**

### **TelaInicial**

Método: `exibirTelaLogin()`

### **Usuario**

Atributos: `nome (string)`, `senha (string)`

### **Responsável (herda de Usuario)**

Métodos:

`designarTarefa(aluno: Aluno, tarefa: Atividade, prazo: string, dificuldade: string)`

`adicionarRecompensa(descricao: string, pontos: int)`

### **Aluno (herda de Usuario)**

Métodos:

`acessarAtividades()`

### **Atividade**

Atributos: `descricao (string)`, `prazo (string)`, `dificuldade (string)`, `pontosRecompensa (int)`, `status (bool)`

### **BarraProgresso**

Atributos: `tempoRestante (int)`, `tempoTotal (int)`

Métodos: `calcularProgresso()`

### **SistemaPontos**

Atributos: `pontosDisponiveis (int)`

Métodos: `acumularPontos(pontos: int)`

### **Recompensa**

Atributos: `descricao (string)`, `pontosNecessarios (int)`

### **Sistema**

Atributos: `usuarios (list[Usuario])`, `atividadesConcluidas (list[Atividade])`

Métodos:

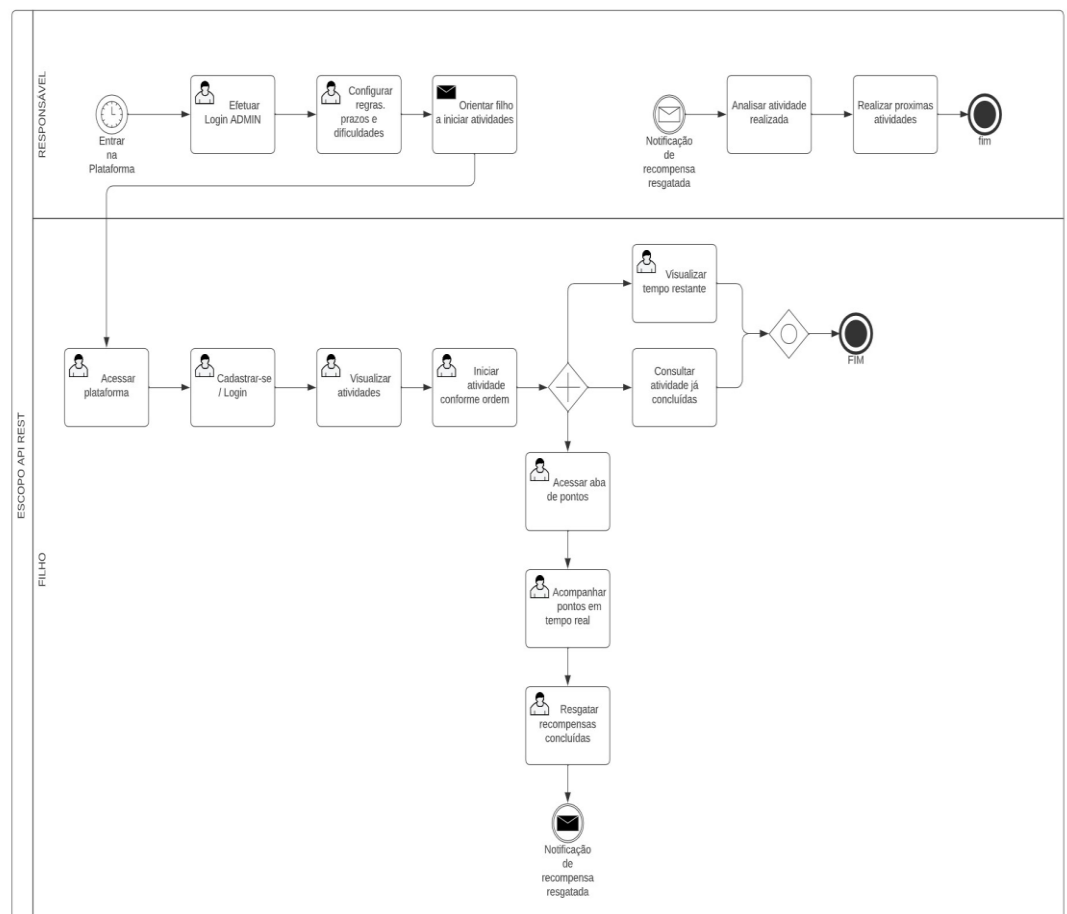
login(usuario: string, senha: string)

exibirAtividades(aluno: Aluno)

exibirRecompensas(aluno: Aluno)

ResgatarRecompensas.

## 2.3 - Diagrama BPMN:



### 3 - Evidência do Teste Unitário:

```
Recompensa 'Assistir a um filme' adicionada ao aluno 'Maria' com sucesso!
Recompensa 'Passear no parque' (30 pontos) adicionada para o aluno
Recompensa 'Passear no parque' adicionada ao aluno 'Maria' com sucesso!
Atividades do Aluno:
Atividade: Descrição: Fazer exercícios
Prazo: 30/05/2023
Dificuldade: Fácil
Pontos de Recompensa: 10
Status: Pendente, Prazo: 30/05/2023, Dificuldade: Fácil
Atividade: Descrição: Ler um livro
Prazo: 15/06/2023
Dificuldade: Média
Recompensa 'Assistir a um filme' adicionada ao aluno 'Maria' com sucesso!
Recompensa 'Passear no parque' (30 pontos) adicionada para o aluno
Recompensa 'Passear no parque' adicionada ao aluno 'Maria' com sucesso!
Atividades do Aluno:
Atividade: Descrição: Fazer exercícios
Prazo: 30/05/2023
Dificuldade: Fácil
Pontos de Recompensa: 10
Status: Pendente, Prazo: 30/05/2023, Dificuldade: Fácil
Atividade: Descrição: Ler um livro
Prazo: 15/06/2023
Dificuldade: Média
Pontos de Recompensa: 20
Status: Pendente, Prazo: 15/06/2023, Dificuldade: Média
Atividades do aluno 'Maria':
- Descrição: Fazer exercícios
  Prazo: 30/05/2023
  Dificuldade: Fácil
```

Como dito antes, o programa foi feito em Python, os arquivos necessários para rodar a API estarão no repositório Git deste Projeto.

[https://github.com/IgorBizutti/Projeto\\_Modelagem/blob/main/unitest.py](https://github.com/IgorBizutti/Projeto_Modelagem/blob/main/unitest.py)

O teste unitário foi feito e rodado sem problemas, para recriar o cenário de teste, baixe o arquivo unittest.py do repositório Git, e faça o comando – “python –m unittest unittest.py”.

Matheus D'Avilla do Carmo, RA - 056903

Lucas Vinícius de Paula Carvalho, RA - 061174

Igor Bizutti, RA - 062166

Otávio Augusto Beividas de Souza, RA - 061338

**20 de junho de 2023 - Sistemas de Informações 5º Semestre – Modelagem de Sistemas.**