

AlphaMathcs

Desenvolvedores

DAVID HENRIQUE SOUZA SANTANA

FELIPE ALAN DE ALMEIDA

FELIPE TESSAROLLO RAMOS

IVAN RAMALHO TONIAL

STENIO VINICIOS DE MEDEIROS

Resumo

Este projeto foi desenvolvido durante o curso de Desenvolvimento Web FullStack Alpha Edtech, promovido pelo instituto Alpha Lumen.

Os desenvolvedores são estudantes do curso e o criaram em uma atividade avaliativa na qual era exigido desenvolver um jogo educativo para crianças.

Público alvo

Crianças entre 6 e 9 anos. Sobretudo recomendamos seu uso como atividade escolar complementar.

Objetivo

O jogo procura desenvolver de forma lúdica e interativa a habilidade da criança de representar quantidades inteiras por meio de blocos representativos (material dourado), bem como efetuar operações aritméticas utilizando essa representação.

No caso de uso como atividade escolar, recomendamos ao(à) professor(a) utilizar para desenvolver as seguintes habilidades listadas na Base Nacional Comum Curricular (BNCC):

- (EF01MA03) *Estimar e comparar quantidades de objetos de dois conjuntos ~~(em torno de 20 elementos)~~, por estimativa e/ou por correspondência ~~(um a um, dois a dois)~~ para indicar “tem mais”, “tem menos” ou “tem a mesma quantidade”.*
- (EF01MA04) *Contar a quantidade de objetos de coleções ~~até 100 unidades~~ e apresentar o resultado ~~por registros verbais e simbólicos~~, em situações de seu interesse, como jogos, brincadeiras, materiais da sala de aula, entre outros.*
- (EF01MA07) *Compor e decompor número de até duas ordens, por meio de diferentes adições, com o suporte de material manipulável, contribuindo para a compreensão de características do sistema de numeração decimal e o desenvolvimento de estratégias de cálculo.*
- (EF01MA08) *Resolver e elaborar problemas de adição e de subtração, envolvendo números de até dois algarismos, com os significados de juntar, acrescentar, separar e retirar, com o suporte de imagens e/ou material manipulável, utilizando estratégias e formas de registro pessoais.*
- (EF02MA01) *Comparar e ordenar números naturais ~~(até a ordem de centenas)~~ pela compreensão de características do sistema de numeração decimal (valor posicional e função do zero).*
- (EF02MA02) *Fazer estimativas por meio de estratégias diversas a respeito da quantidade de objetos de coleções e registrar o resultado da contagem desses objetos ~~(até 1000 unidades)~~.*
- (EF02MA03) *Comparar quantidades de objetos de dois conjuntos, ~~por estimativa e/ou~~ por correspondência (um a um, dois a dois, entre outros), para indicar “tem mais”, “tem menos” ou “tem a mesma quantidade”, indicando, quando for o caso, quantos a mais e quantos a menos.*
- (EF02MA04) *Compor e decompor números naturais de até três ordens, com suporte de material manipulável, por meio de diferentes adições.*
- (EF02MA06) *Resolver e elaborar problemas de adição e de subtração, envolvendo números de até três ordens, com os significados de juntar, acrescentar, separar, retirar, utilizando estratégias pessoais.*
- (EF03MA02) *Identificar características do sistema de numeração decimal, utilizando a composição e a decomposição de número natural de até quatro ordens.*
- (EF03MA03) *~~Construir~~ e utilizar fatos básicos da adição e da multiplicação para o cálculo mental ou escrito.*
- (EF03MA05) *Utilizar diferentes procedimentos de cálculo mental e escrito, inclusive os convencionais, para resolver problemas significativos envolvendo adição e subtração com números naturais.*

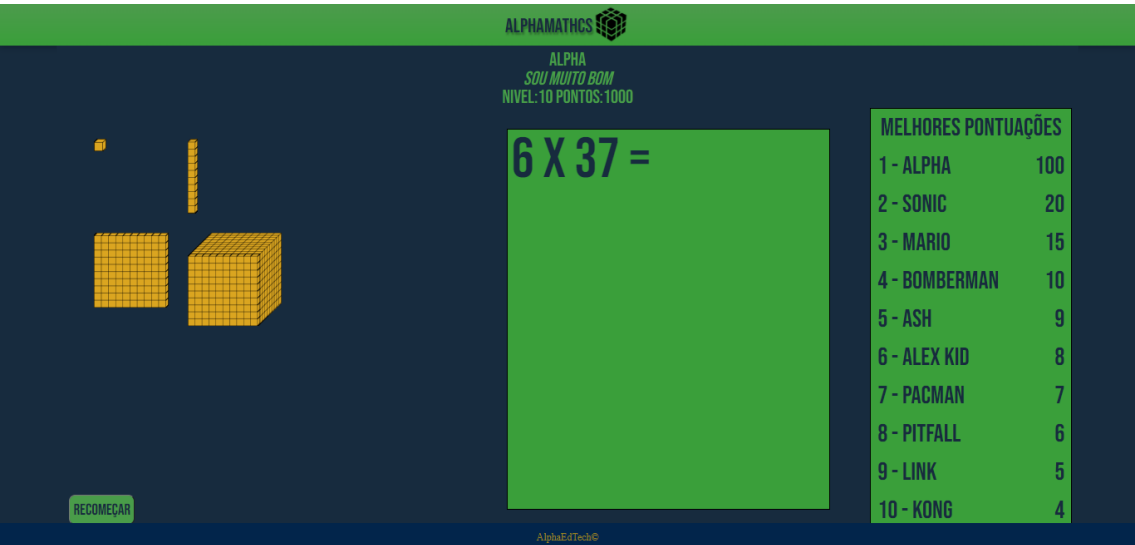
- (EF03MA06) Resolver e elaborar problemas de adição e subtração com os significados de juntar, acrescentar, separar, retirar, comparar e completar quantidades, utilizando diferentes estratégias de cálculo exato ou aproximado, incluindo cálculo mental.
- (EF03MA08) Resolver e elaborar problemas de divisão de um número natural por outro (até 10), com resto zero e com resto diferente de zero, com os significados de repartição equitativa e de medida, por meio de estratégias e registros pessoais.
- (EF04MA01) Ler, escrever e ordenar números naturais até a ordem de dezenas de milhar.
- (EF04MA04) Utilizar as relações entre adição e subtração, bem como entre multiplicação e divisão, para ampliar as estratégias de cálculo.
- (EF04MA05) Utilizar as propriedades das operações para desenvolver estratégias de cálculo.
- (EF04MA06) Resolver e elaborar problemas envolvendo diferentes significados da multiplicação (adição de parcelas iguais, organização retangular e proporcionalidade), utilizando estratégias diversas, como cálculo por estimativa, cálculo mental e algoritmos.
- (EF04MA07) Resolver e elaborar problemas de divisão ~~cujos divisores tenham no máximo dois algarismos~~, envolvendo os significados de repartição equitativa e de medida, utilizando estratégias diversas, como cálculo por estimativa, cálculo mental e algoritmos.
- (EF04MA08) Resolver, com o suporte de imagem e/ou material manipulável, problemas simples de contagem, como a determinação do número de agrupamentos possíveis ao se combinar cada elemento de uma coleção com todos os elementos de outra, utilizando estratégias e formas de registro pessoais.

Referencia: <http://basenacionalcomum.mec.gov.br/abase/#fundamental/matematica-no-ensino-fundamental-anos-iniciais-unidades-tematicas-objetos-de-conhecimento-e-habilidades>

Mecânica do jogo

O jogo consiste em resolver uma serie de operações matemáticas por meio da contagem de blocos (material dourado). O jogador tem a disposição 36 blocos sem 9 para representar unidades, 9 para representar dezenas, 9 para representar centenas e 9 para representar milhares, de forma que é possível compor qualquer valor inteiro entre 0 e 9.999. Para compor o valor desejado o jogador deve “arrastar” para um campo demarcado da tela. Toda vez que conseguir resolver corretamente a operação matemática um novo desafio aparece.

A cada operação respondida corretamente o jogador acrescenta 10 pontos. A cada 5 (cinco) operações respondidas corretamente os dados do usuário são salvos no servidor. Um ranking com os 10 jogadores com maior pontuação registrada no servidor é mostrado na tela, incentivando assim o jogador a alcançar melhores posições.



As operações matemáticas são apresentadas de forma progressiva de dificuldade de acordo com a tabela:

Nível	Subnível	Operações
1	1	Números entre 0 e 9
	2	Números entre 10 e 99
	3	Números entre 100 e 999
	4-5	Números entre 1000 e 9999
2	1	Soma de números entre 0 e 9
	2-3	Soma de números entre 0 e 100
	4-5	Soma de números entre 0 e 1000
3	1-5	Soma e subtração de números entre 0 e 1000
4	1-3	Multiplicação de números entre 1 e 10
	4-5	Multiplicação de números entre 1 e 100
5	1-3	Soma, subtração e multiplicação de números entre 1 e 10
	4-5	Soma, subtração e multiplicação de números entre 1 e 100
6	1	Divisões com dividendo até 10 e divisor até 10
	2	Divisões com dividendo até 100 e divisor até 10
	3	Divisões com dividendo até 1000 e divisor até 20
	4-5	Divisões com dividendo até 10000 e divisor até 20
7 ou mais	1-5	Somas, subtrações, multiplicações e divisões cujo resultado esteja entre 1 e 9999

APIs

O jogo é executado quase na sua totalidade no navegador, porem, os dados dos usuários são salvos no servidor, permitindo assim que o jogador continue a sua progressão em diferentes maquinas, bastando para isso informar o mesmo nome de usuário. Além disso, o servidor permite ranquear todos os jogadores, independentes se utilizar a mesma maquina ou maquinas diferentes.

A comunicação entre o cliente e o servidor é feita por meio de três rotas:

- `http://{ Ip Server }:3000/scores` – Método “GET”: consulta o ranqueamento das 10 melhores pontuações já atingidas no jogo. Retorna um vetor com 10 objetos com o seguinte formato:
`{ "name": "string", "score": integer }`
- `http://{ Ip Server }:3000/status?name={ nome de usuário }`– Método “GET”: consulta os dados do usuário, como pontuação e nível atingidos. Retorna um objeto com o seguinte formato:
`{ "name": "string", "description": "string", "score": integer, "level": integer }`
- `http://{ Ip Server }:3000/register` – Método “POST” com informações no corpo (body) no seguinte formato:
`{ "name": "string", "description": "string", "score": integer, "level": integer }`
Retorna *true* caso o registro seja bem sucedido ou *false* caso contrário:

Tecnologias utilizadas

O BackEnd foi escrito em linguagem JavaScript e executado utilizando NodeJS e Pm2. As bibliotecas Express, Fs e Cors foram utilizadas.

O frontEnd foi escrito utilizando HTML, CSS e JavaScript com o uso da biblioteca JQuery.