

Projeto Teste

Tecnologias WEB

Submetido por:
Gustavo Henrique da Silva, SP

Conteúdo

Introdução.....	3
Glossário.....	3
PHP – 3 horas.....	4
Descrição do projeto e tarefas.....	4
Descrição do formato da entrega.....	4
Causas de incompatibilidade	4
Detalhes da API.....	6
Autenticação e casos gerais	6
Detalhes das Entidades.....	6
API	8
Ficha de Avaliação Simplificada	17

Introdução

Uma empresa de venda e montagem de Computadores na cidade de Maceió/AL está expandindo seus negócios para o restante do Brasil. Para garantir que as máquinas vendidas aos seus clientes possuam peças compatíveis, eles precisam de um *Web Service* que teste a compatibilidade das peças.

O serviço deve ser capaz de:

- Listar peças avulsas;
- Montar máquinas inteiras com peças que sejam compatíveis umas às outras;
- Caso haja incompatibilidade, esta deve ser reportada ao usuário.

O **cadastro** de *templates* de máquinas inteiras deve ser dinâmico para o usuário final.

Glossário

- Soquete (Socket): é o conector, localizado na placa-mãe, que define a interface elétrica para transmissão de dados de e para o processador.
- TDP (Thermal Design Power): é o máximo de calor gerado por um chip de computador ou componente (podendo ser processador, placa de vídeo e etc.) que o sistema de refrigeração do computador foi projetado para dissipar sem nenhuma sobrecarga.
- RAM (Random Access Memory): tipo de memória presente nos computadores para processamentos rápidos e aleatórios. Tipicamente usada para dados ativos durante o trabalho do usuário.
- SLI/Crossfire: tecnologias usadas pelas placas de vídeo da Nvidia e AMD, respectivamente, para trabalharem em conjunto e aumentarem o poder de processamento gráfico total do computador.
- PCI Express (Peripheral Component Interconnect Express): é um padrão de conexão serial de alta velocidade presente nas placas-mãe.

PHP – 3 horas

Descrição do projeto e tarefas

A execução do projeto dura 3 horas. Neste tempo você deve criar uma API utilizando PHP (opcionalmente, com um framework) e uma base de dados relacional.

Você receberá arquivos .json, com dados iniciais, o arquivo de criação do banco de dados e imagens de cada um dos diferentes tipos de entidade que devem ser cadastrados no seu sistema para permitir a execução de testes.

Descrição do formato da entrega

Os seguintes requisitos devem ser cumpridos na entrega do projeto teste:

- O banco de dados deve ser criado com o nome AlatechMachines e deve usar a *collation* utf8mb4;
- A sua API deve ser acessível a partir de /XX/AlatechMachines/api (é sua responsabilidade fornecer o endereço e a porta do *host*. **XX** corresponde à sigla da sua delegação).

Causas de incompatibilidade

Sua API deve informar *incompatibilidade* quando máquinas forem criadas ou editadas nos seguintes casos:

- Tipo de soquete da placa-mãe for diferente do tipo de soquete do processador;
- TDP do processador for maior do que o TDP máximo suportado pela placa-mãe;
- Tipo de memória RAM da placa-mãe for diferente do tipo da memória RAM;
- Quantidade de memórias RAM for maior do que a quantidade de slots presentes na placa-mãe (sempre deve haver ao menos 1 memória RAM na máquina);
- Quantidade de placas de vídeo for maior do que a quantidade de slots PCI Express na placa-mãe (sempre deve haver ao menos 1 placa de vídeo na máquina);
- Quantidade de dispositivos de armazenamento do tipo SATA for maior do que a quantidade de slots SATA na placa mãe;
- Quantidade de dispositivos de armazenamento do tipo M2 for maior do que a quantidade de slots M2 na placa mãe;

- Soma total de dispositivos de armazenamento é igual a zero (ou seja, deve haver ao menos 1 disp. SATA ou 1 disp. M2);
- Quantidade de placas de vídeo for maior que 1 e o modelo de placa de vídeo não suporta SLI/Crossfire;
- Potência da fonte de alimentação for menor do que a potência mínima da placa de vídeo vezes (multiplicada) pela quantidade de placas de vídeo.

Detalhes da API

Autenticação e casos gerais

Por questões de segurança, apenas o *endpoint* de *login* e imagens deve ser público. Ao acessar qualquer outro endereço que não corresponda aos citados anteriormente, a API deve retornar o status code **HTTP 401 Unauthorized** se o *header Authorization* não estiver presente na *request*. Caso haja o *header*, mas este é inválido o *status code* **HTTP 403 Forbidden** deve ser retornado.

A senha do usuário deve ser criptografada no banco de dados com o tipo SHA-256.

A autenticação do usuário na API deve ser feita por meio de um JSON Web Token - JWT que será enviado a cada requisição no header *Authorization*. Esse token deve estar:

- criptografado;
- tipo: `Bearer {token}`; e
- seguro para prevenir a extração da senha do usuário.

Detalhes das Entidades

Após a autenticação do usuário, ele deve ter acesso aos *endpoints* que o permitirão manipular as entidades a seguir (cada entidade deve ter seu *endpoint*) a fim de montar máquinas (que será um *endpoint* separado):

- Listagem de Placas-Mãe
 - Deve ter nome, URL da imagem, marca, tipo do soquete, tipo de memória RAM, quantidade de *slots* de RAM, TDP máximo, quantidade de *slots* SATA, quantidade de *slots* M2, quantidade de *slots* PCI Express.
- Listagem de Processadores
 - Deve ter nome, URL da imagem, marca, tipo do soquete, quantidade de núcleos físicos, frequência base, frequência máxima, quantidade de *cache*, TDP.
- Listagem de Memórias RAM
 - Deve ter nome, URL da imagem, marca, quantidade de memória, tipo de memória RAM, frequência.
- Listagem de Dispositivos de Armazenamento
 - Deve ter nome, URL da imagem, marca, tipo de dispositivo (HDD ou SSD), quantidade de memória, tipo de entrada (SATA ou M2).

- Listagem de Placas de Vídeo
 - Deve ter nome, URL da imagem, marca, quantidade de memória, tipo de memória (GDDR5 ou GDDR6), potência de alimentação mínima, suporta SLI/Crossfire.
- Listagem de Fontes de Alimentação
 - Deve ter nome, URL da imagem, marca, potência (em Watts), classificação 80 Plus (White, Bronze, Silver, Gold, Platinum, Titanium ou não se aplica).
- Listagem de Marcas
 - Deve ter nome.
- Gerenciamento de Máquinas Inteiras
 - Deve ter nome, URL da imagem, placa-mãe, processador, memórias (no mínimo 1 unidade), dispositivos de armazenamento (no mínimo 1 unidade), placas de vídeo (no mínimo 1 unidade), fonte de alimentação.

Todos os produtos devem ter uma foto (as fotos estão fornecidas na mídia) e elas devem ser acessíveis por meio do *endpoint* especificado na seção API a seguir.

API

A API deve sempre receber e retornar dados em formato JSON. Ela deve ter os seguintes *endpoints*, que são relativos à /alatech/api):

POST (login) Autenticação de usuários	
Request	Response
Content-type: application/json Body: <pre>{ username: string <o nome do usuário que se pretenda autenticar> password: string <a senha do usuário que se pretenda autenticar> }</pre>	1. Se tudo OK: Status: 200/OK Content-type: application/json Body: <pre>{ token: string <token que autoriza o usuário a acessar os endpoints do sistema.> }</pre> 2. Se algum dado incorreto: Status: 422/UNPROCESSABLE ENTITY Content-type: application/json Body: <pre>{ message: "Credenciais inválidas" }</pre> 3. Se já houver uma autenticação ativa para o usuário: Status: 403/FORBIDDEN Content-type: application/json Body: <pre>{ message: "Usuário já autenticado" }</pre>

DELETE (logout) Remoção da autenticação de usuários	
Request	Response
Authorization: Bearer string <identificação do usuário se dará pelo header Authorization (assim como os outros endpoints)> Body: vazio	1. Logout confirmado: Status: 200/OK Content-type: application/json Body: <pre>{ message: "Logout com sucesso" }</pre> 2. Caso o token não esteja presente na requisição: Status: 401/UNAUTHORIZED Content-type: application/json Body: <pre>{ message: "Necessário estar autenticado no sistema" }</pre> 3. Caso o token não corresponda a nenhuma autenticação válida: Status: 403/FORBIDDEN Content-type: application/json Body: <pre>{ message: "Token inválido" }</pre>

GET (motherboards, processors, ram-memories, storage-devices, graphic-cards, power-supplies, machines, brands) Listagens de placas-mãe, processadores, memórias RAM, dispositivos de armazenamento, placas de vídeo, fontes de alimentação e máquinas inteiras, respectivamente.	
Request	Response
<p>header: Authorization string <identificação do usuário se dará pelo header Authorization (assim como os outros endpoints)></p> <p>Parâmetros: Query string parameters, ambos opcionais:</p> <ul style="list-style-type: none"> ● pageSize: int <determina o tamanho de cada página de busca> (padrão é 20) ● page: int <determina a página atual da busca> (padrão é 1) 	<p>1. Fluxo normal: Status: 200/OK Content-type: application/json Body:</p> <pre>[{ <propriedades descritas na subseção "Detalhes de Entidades" para a entidade referenciada> }, { <propriedades descritas na subseção "Detalhes de Entidades" para a entidade referenciada> }, ...]</pre> <p>2. Caso o token não esteja presente na requisição: Status: 401/UNAUTHORIZED Content-type: application/json Body:</p> <pre>{ message: "Necessário estar autenticado no sistema" }</pre> <p>3. Caso o token não corresponda a nenhuma autenticação válida: Status: 403/FORBIDDEN Content-type: application/json Body:</p> <pre>{ message: "Token inválido" }</pre>

GET (search/{category}?q={q}) Busca de produtos	
Request	Response
<p>Authorization: Bearer string <identificação do usuário se dará pelo header Authorization (assim como os outros endpoints)></p> <p>Parâmetros obrigatórios:</p> <ul style="list-style-type: none"> ● q: string <corresponde ao termo de busca que deve ser aplicado ao nome dos itens buscados> ● category: string <tipo de peça a ser buscado (deve corresponder ao nome da peça no endpoint, por exemplo motherboards ou processors)> <p>opcionais:</p> <ul style="list-style-type: none"> ● pageSize: int <determina o tamanho de cada página de busca> (padrão é 20) ● page: int <determina a página atual da busca> (padrão é 1) 	<p>1. Fluxo normal:</p> <ul style="list-style-type: none"> - produtos listados - O total de itens de todos os grupos não deve exceder o valor de pageSize <p>Status: 200/OK Content-type: application/json Body:</p> <pre>[{ <propriedades descritas na subseção "Detalhes de Entidades" para a entidade referenciada> }, { <propriedades descritas na subseção "Detalhes de Entidades" para a entidade referenciada> }, ...]</pre> <p>2. Caso o token não esteja presente na requisição: Status: 401/UNAUTHORIZED Content-type: application/json Body:</p> <pre>{ message: "Necessário estar autenticado no sistema" }</pre> <p>3. Caso o token não corresponda a nenhuma autenticação válida: Status: 403/FORBIDDEN Content-type: application/json Body:</p> <pre>{ message: "Token inválido" }</pre>

POST (machines)	
Criar um novo modelo de máquina.	
Request	Response
<p>Authorization: Bearer string <identificação do usuário se dará pelo header Authorization (assim como os outros endpoints)></p> <p>Body: um objeto JSON que deve conter todas as propriedades a seguir:</p> <p>Propriedades do JSON:</p> <ul style="list-style-type: none"> ● motherboardId: ID da placa-mãe ● powerSupplyId: ID da fonte de alimentação ● processorId: ID do processador ● ramMemoryId: ID da memória RAM ● ramMemoryAmount: quantidade de memórias RAM ● storageDevices: array de objetos JSON que contém: <ul style="list-style-type: none"> ○ storageDeviceId: IDs do dispositivo de armazenamento ○ amount: quantidade de dispositivos deste modelo ● graphicCardId: ID da placa de vídeo ● graphicCardAmount: quantidade de placas de vídeo ● imageBase64: imagem do computador, em formato base 64. 	<p>1. Máquina criada com sucesso: Status: 201/CREATED Content-type: application/json Body: <pre>[{ "id": <ID da máquina criada>, ... }]</pre></p> <p>2. Imagem não fornecida, as quantidades de peças foram inválidas ou incompatibilidades entre elas foram encontradas: Status: 422/UNPROCESSABLE ENTITY Content-type: application/json Body: <pre>{ "<propriedade com erro>": "<descrição do erro ou incompatibilidade>", "<propriedade com erro>": "<descrição do erro ou incompatibilidade>", ... }</pre></p> <p>3. Caso o token não esteja presente na requisição: Status: 401/UNAUTHORIZED Content-type: application/json Body: <pre>{ message: "Necessário estar autenticado no sistema" }</pre></p> <p>4. Caso o token não corresponda a nenhuma autenticação válida: Status: 403/FORBIDDEN Content-type: application/json Body: <pre>{ message: "Token inválido" }</pre></p>

PUT (machines/{id}) Alterar um modelo de máquina existente.	
Request	Response
<p>Authorization: Bearer string <identificação do usuário se dará pelo header Authorization (assim como os outros endpoints)></p> <p>Body:</p> <pre>{ <mesmo que POST (machine)> }</pre> <p>Caso imageBase64 não seja fornecido, a imagem da máquina (propriedade "imageUrl") deve permanecer inalterada.</p> <p>Parâmetros obrigatórios:</p> <ul style="list-style-type: none"> id: ID da máquina a ser alterada. 	<p>1. Modelo de máquina atualizado com sucesso: Status: 200/OK Content-Type: application/json Body:</p> <pre>{ <propriedades de máquina descritas na subseção "Detalhes de Entidades"> }</pre> <p>2. As quantidades de peças foram inválidas ou incompatibilidades entre elas foram encontradas: Status: 422/UNPROCESSABLE ENTITY Content-Type: application/json Body:</p> <pre>{ "<propriedade com erro>": "<descrição do erro ou incompatibilidade>", "<propriedade com erro>": "<descrição do erro ou incompatibilidade>", ... }</pre> <p>3. Caso o token não esteja presente na requisição: Status: 401/UNAUTHORIZED Content-type: application/json Body:</p> <pre>{ message: "Necessário estar autenticado no sistema" }</pre> <p>4. Caso o token não corresponda a nenhuma autenticação válida: Status: 403/FORBIDDEN Content-type: application/json Body:</p> <pre>{ message: "Token inválido" }</pre>

DELETE (machines/{id}) Remover um modelo de máquina existente.	
Request	Response
<p>Authorization: Bearer string <identificação do usuário se dará pelo header Authorization (assim como os outros endpoints)></p> <p>Parâmetros obrigatórios:</p> <ul style="list-style-type: none"> ● id: int <ID da máquina a ser deletada> 	<p>1. Modelo de máquina removido com sucesso: Status: 204/NO CONTENT</p> <p>2. Caso o modelo de máquina não exista: Status: 404/NOT FOUND Content-type: application/json Body: <pre>{ message: "Modelo de máquina não encontrado" }</pre></p> <p>3. Caso o token não esteja presente na requisição: Status: 401/UNAUTHORIZED Content-type: application/json Body: <pre>{ message: "Necessário estar autenticado no sistema" }</pre></p> <p>4. Caso o token não corresponda a nenhuma autenticação válida: Status: 403/FORBIDDEN Content-type: application/json Body: <pre>{ message: "Token inválido" }</pre></p>

POST (verify-compatibility) Verificar a compatibilidade entre duas ou mais peças, não necessariamente uma máquina, no entanto deve sempre haver uma placa-mãe e uma fonte de alimentação.	
Request	Response
<p>Authorization: Bearer string <identificação do usuário se dará pelo header Authorization (assim como os outros endpoints)></p> <p>Body: um objeto JSON com algumas propriedades obrigatórias (marcadas com *) e outras opcionais:</p> <p>Propriedades do JSON:</p> <ul style="list-style-type: none"> ● motherboardId*: ID da placa-mãe ● powerSupplyId*: ID da fonte de alimentação ● processorId: ID do processador ● ramMemoryId: ID da memória RAM ● ramMemoryAmount: quantidade de memórias RAM ● storageDevices: array de objetos JSON que contém: <ul style="list-style-type: none"> ○ storageDeviceId: IDs do dispositivo de armazenamento ○ amount: quantidade de dispositivos deste modelo ● graphicCardId: ID da placa de vídeo ● graphicCardAmount: quantidade de placas de vídeo 	<p>1. Verificação bem sucedida: Status: 200/NO CONTENT Body: <pre>{ message: "Máquina válida" }</pre></p> <p>2. Incompatibilidades encontradas: Status: 422/UNPROCESSABLE ENTITY Content-Type: application/json Body: <pre>{ "<propriedade com erro>": "<descrição do erro ou incompatibilidade>", "<propriedade com erro>": "<descrição do erro ou incompatibilidade>", ... }</pre></p> <p>3. Caso o token não esteja presente na requisição: Status: 401/UNAUTHORIZED Content-type: application/json Body: <pre>{ message: "Necessário estar autenticado no sistema" }</pre></p> <p>4. Caso o token não corresponda a nenhuma autenticação válida: Status: 403/FORBIDDEN Content-type: application/json Body: <pre>{ message: "Token inválido" }</pre></p>

GET (images/{id}) Retorna a imagem com o dado “id” (que é o valor correspondente em “imageUrl” nas listagens da API)	
Request	Response
Parâmetros: Obrigatórios: <ul style="list-style-type: none"> ● id: int <ID da imagem buscada> 	1. Fluxo normal: Status: 200/OK Content-type: image/string <string sendo o valor correspondente ao tipo de imagem> Body: A própria imagem 2. Caso a imagem não exista: Status: 404/NOT FOUND Content-type: application/json Body: <pre>{ message: "Imagem não encontrada" }</pre>

Ficha de Avaliação Simplificada

SEÇÃO	CRITÉRIO	PONTOS DE JULGAMENTO	PONTOS DE MEDIÇÃO	TOTAL
C1	Desenvolvimento lado servidor	0,00	10,00	10,00
Total		0,00	10,00	10,00