

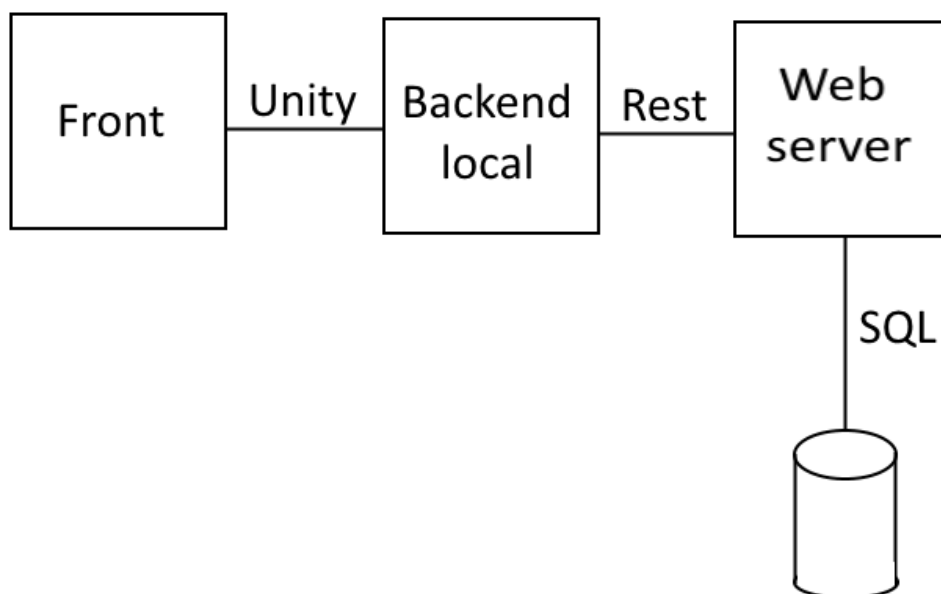
Grupo: Jean Walper e Pedro de la Peña

Descrição do projeto

O projeto consiste em uma simulação de itens, equipamentos e personagens em um jogo. Neste projeto é possível: criar uma conta com login, senha e apelido, criar até 3 personagens por conta, comprar itens de uma loja e também trocar itens entre usuários.

Os itens adquiridos podem ser equipados fazendo com que os atributos do personagem sejam melhorados. Além disso, os itens não são únicos: mais de um usuário pode possuir uma cópia do mesmo item ao mesmo tempo e, visto que os itens possuem raridades diferentes - e quanto mais raro mais difícil de conseguir e caro ele se torna - a troca entre os jogadores é um mecanismo ideal para aqueles que desejam atingir o seu pico de poder ou simplesmente gostam de colecionar itens. Contudo, não é possível que um mesmo usuário equipe dois de seus personagens com os mesmos itens: será preciso desequipar um para poder equipar outro.

Diagrama da arquitetura do sistema



Dicionário Database

Tabela Users

---- Inputs do usuário ----

Login: string única de 16 caracteres utilizada para entrar no jogo.

Senha: string de 64 caracteres utilizada em conjunto com o login para entrar no jogo.

Nick: string única de 16 caracteres que identificará os jogadores em jogo (apelido).

---- Inputs do servidor ----

Join_date: data de criação da conta.

Last_login_date: data do último acesso do usuário.

Active: booleana (flag) que determina se o usuário logou recentemente.

Banned: booleana (flag) que identifica se as contas estão restringidas ou não de acessar o jogo.

Money_spent: int que representa o dinheiro obtido pelo usuário em jogo.

Tabela Item template

---- Inputs do servidor ----

Template_id: int única responsável para que o servidor identifique o item.

Item_name: string que os players utilizarão para identificar o item no jogo.

Item_type: string que os players utilizarão para identificar o tipo e utilidade do item no jogo.

Amount_1 e Amount_2: int que representa um campo genérico que pode ser utilizado em diversos tipos de itens para determinar seus efeitos. A partir destes campos é possível determinar a durabilidade de uma espada ou armadura e também a regeneração de vida de uma poção.

Rarity: int utilizada para determinar a raridade de um item; quanto mais alto o valor, mais raro o item será.

Tabela Item

---- Inputs do servidor ----

Item_id: int auto incrementado que identifica um item quando este é adicionado a conta de um jogador.

Template_id: int que faz referência à chave da tabela item_template.

Login: string que faz referência ao login do usuário dono do item.

Equip: booleana que determina se o item está equipado ou não por um personagem.

Create_date: data em que o item foi adicionado à conta.

Tabela Character

---- Inputs do servidor ----

Character_id: int auto incrementado que identifica os personagens criados.

Login: string que faz referência ao dono do personagem.

Strenght: int que representa o atributo força do personagem.

Dexterity: int que representa o atributo destreza do personagem.

Vigor: int que representa o atributo vida do personagem.

Intelligence: int que representa o atributo inteligência do personagem.

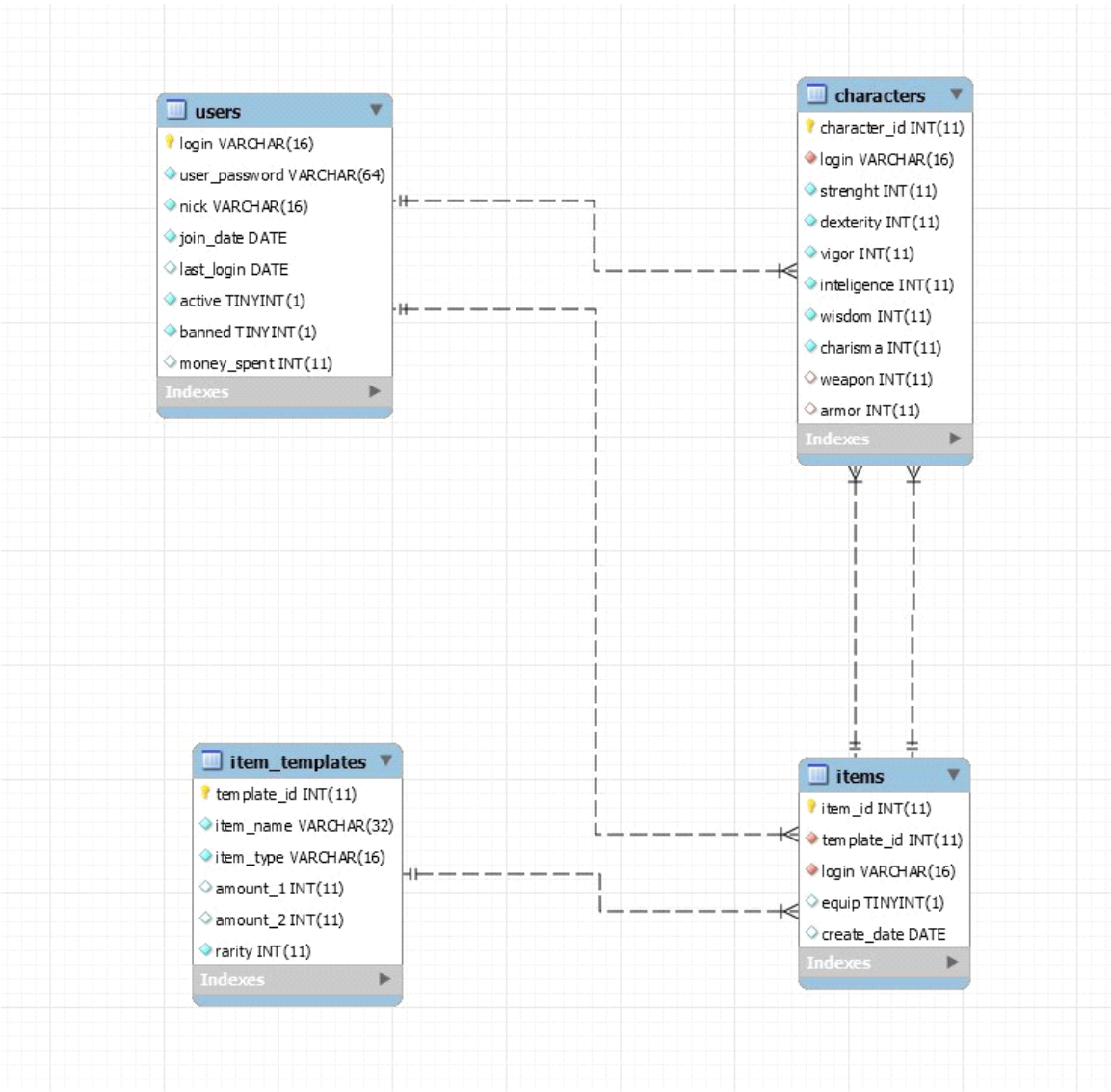
Wisdom: int que representa o atributo sabedoria do personagem.

Charisma: int que representa o atributo carisma do personagem.

Weapon: int que faz referência ao item utilizado como arma do personagem, aumentando seu ataque.

Aarmor: int que faz referência ao item utilizado como armadura do personagem, diminuindo a vida perdida quando o mesmo recebe dano.

Diagrama E-R



REST API

Recurso\Método	GET	POST	PUT	DELETE
/users	<p>Retorna lista de todos os usuários</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errormsg</p>			
/newuser		<p>Adiciona um novo usuário, seguindo o modelo /newuser/:name/:password/:nick/:joindate/:lastlogin/:active/:ban/:money</p> <p>Return: Success: Status 200, Json array</p> <p>Fail: Status 400, errormsg</p>		
/user/:username	<p>Retorna as informações do usuário especificado</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errormsg</p>			
/user/:username/characters	<p>Retorna os personagens criados pelo usuário especificado</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errormsg</p>			
/newcharacter		<p>Adiciona um novo personagem para o usuário, seguindo o modelo /newcharacter/:ownername/:str/:dex/:vig/:int/:wis/:cha</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errormsg</p>		

/putcharacter/			<p>Altera os atributos do personagem com base em seu id, , seguindo o modelo /putcharacter/:id/:strength/:dexterity/:vigor/:intelligence/:wisdom/:charisma</p> <p>Return: Success: Status 200, Json array</p> <p>Fail: Status 400, errmsg</p>	
/userlogin/:username/:date		<p>Altera a última data em que o jogador se conectou ao servidor (login mais recente)</p> <p>Return: Success: Status 200, ok</p> <p>Fail: Status 400, errmsg</p>	<p>O método originalmente era um patch/put, porém o Unity não estava aceitando tal método.</p>	
/deletecharacter/:id				<p>Deleta um personagem com base em seu ID</p> <p>Return: Success: Status 200, ok</p> <p>Fail: Status 400, errmsg</p>
/newitem/:template/:login/:date		<p>Adiciona um novo item a um usuário e a data de sua criação.</p> <p>Return: Success: Status 200, Json array</p> <p>Fail: Status 400, errmsg</p>		
/deleteitem/:itemid				<p>Deleta um item com base em seu ID</p> <p>Return: Success: Status 200, ok</p> <p>Fail: Status 400, errmsg</p>

/item/:username	<p>Retorna todos os itens possuídos por um usuário</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errmsg</p>			
/item/:username/weapons	<p>Retorna todos os itens do usuário que são do tipo “weapon”</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errmsg</p>			
/item/:username/armor	<p>Retorna todos os itens do usuário que são do tipo “armor”</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errmsg</p>			
/templates	<p>Retorna todos os templates de item já criados no jogo.</p> <p>Return: Success: Status 200, json array</p> <p>Fail: Status 400, errmsg</p>			
/trade/start		<p>Inicia uma transação</p> <p>Return: Success: Status 200</p> <p>Fail: Status 400, errmsg</p>		
/trade/rollback		<p>Realiza um rollback</p> <p>Return: Success: Status 200</p> <p>Fail: Status 400, errmsg</p>		

/trade/commit		Realiza um commit Return: Success: Status 200 ok Fail: Status 400, errmsg		
/trade/:id/:login		Altera os itens de um usuário com base no id do item Return: Success: Status 200 ok Fail: Status 400, errmsg		

Bugs conhecidos

Quando um usuário possui mais de um personagem e ele tem itens equipados neles, não há feedback em relação aos atributos dos itens equipados. Contudo, quando o personagem só tem um item equipado, a mudança de equipamentos possui feedback instantâneo.