**GbClickerV2**

**Resumo:** O GbClickerV2 é um sistema que teve como base um antigo projeto meu (GbClicker) utilizando as mesmas ferramentas (HTML, CSS, JAVASCRIPT, PHP e MYSQL), porém, de uma maneira muito mais simples e desorganizada. É um projeto pessoal de um jogo do tipo “*Clicker/Idle*”, que consiste em o usuário clicar para conquistar dinheiro **virtual e fictício** utilizado para o aprimoramento das ferramentas utilizadas no jogo. Em um primeiro momento, o jogo não necessariamente terá alguma interação entre jogadores, porém pode ser alterado de acordo com o caminhar do projeto. Hoje dia 19/08/2023, resolvi utilizar o padrão de projeto MVC (Model, View and Control) pois com o desenvolver do projeto, notei que os códigos estavam ficando difíceis de serem acessados e alterados. Também não utilizei mais o aplicativo XAMPP para manusear o banco de dados ou criar servidor por meio do Apache, estou utilizando o servidor embutido no próprio PHP.

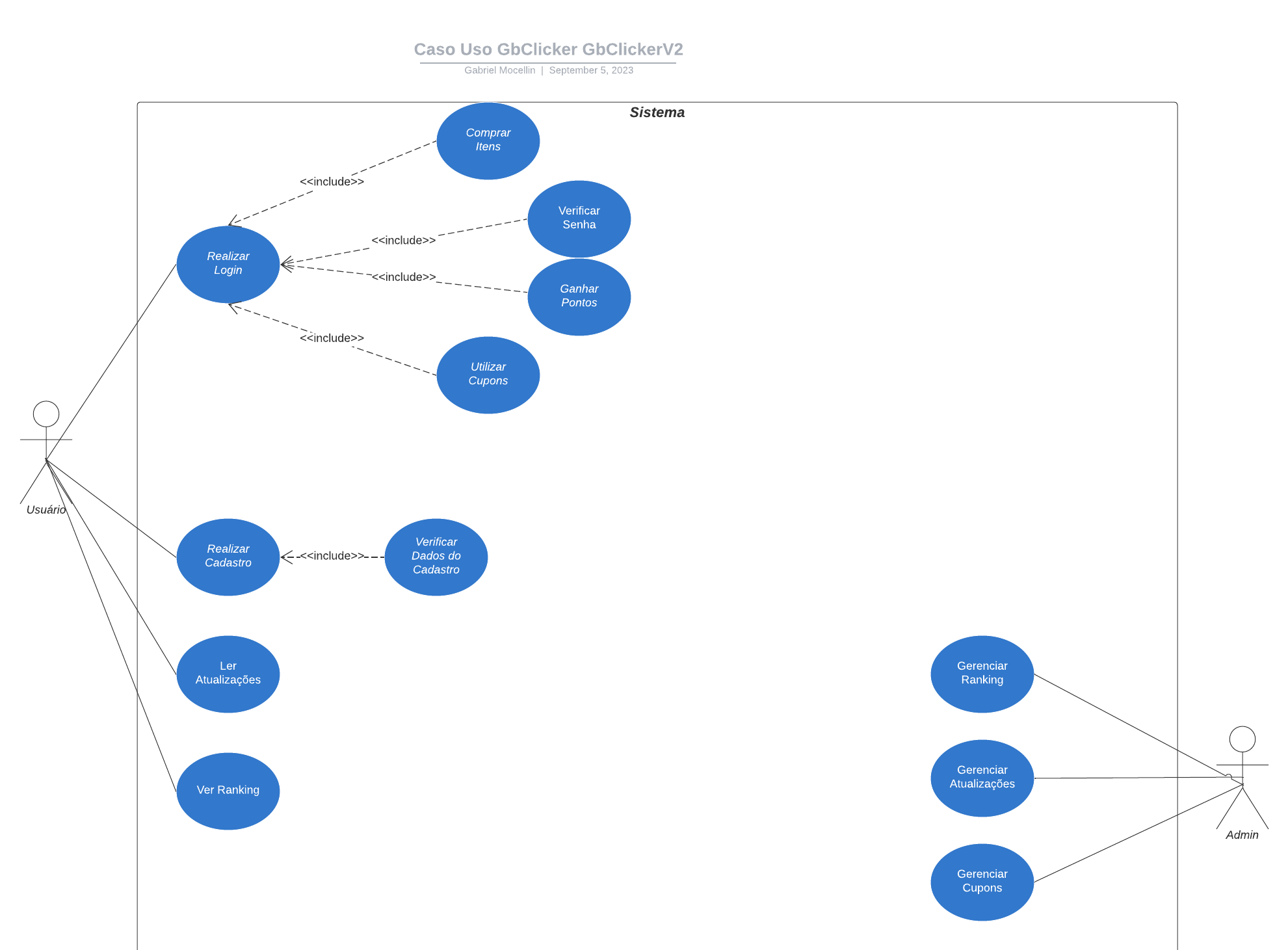
**Requisitos Funcionais**

|  |
| --- |
| **REQUISITOS** |
| **1RF:** O site deverá possuir uma tela inicial (*landing page*) com uma pequena introdução sobre o jogo. |
| **2RF:** A tela inicial (*landing page*) as atualizações feitas com datas e fotos. |
| **3RF:** A página inicial deverá conter um botão que redirecione para a página de login. |
| **4RF:** A página de login deverá conter um botão que redirecione para a página de registro, caso o usuário não possua conta. |
| **5RF:** Os campos presentes na página de registro deverão possuir validações no *back-end* utilizando Javascript antes de serem enviados ao banco de dados. |
| **6RF:** Caso haja uma tentativa de registro com campos inválidos, deve ser informado para o usuário por meio de uma notificação os campos invalidados. |
| **7RF:** Ao usuário realizar login com campos válidos, deve ser redirecionado à página inicial (onde o jogo acontece de fato). |
| **8RF:** Deve ser disponibilizado ao usuário na página do jogo um menu, onde possa ser feita a navegação entre as páginas de máquinas, configurações, loja, página inicial, cupons, perfil e inventário. |
| **9RF:** Ao usuário realizar o login deve ser iniciado o sistema de coleta de dinheiro por *minions*, que são comprados na loja. |

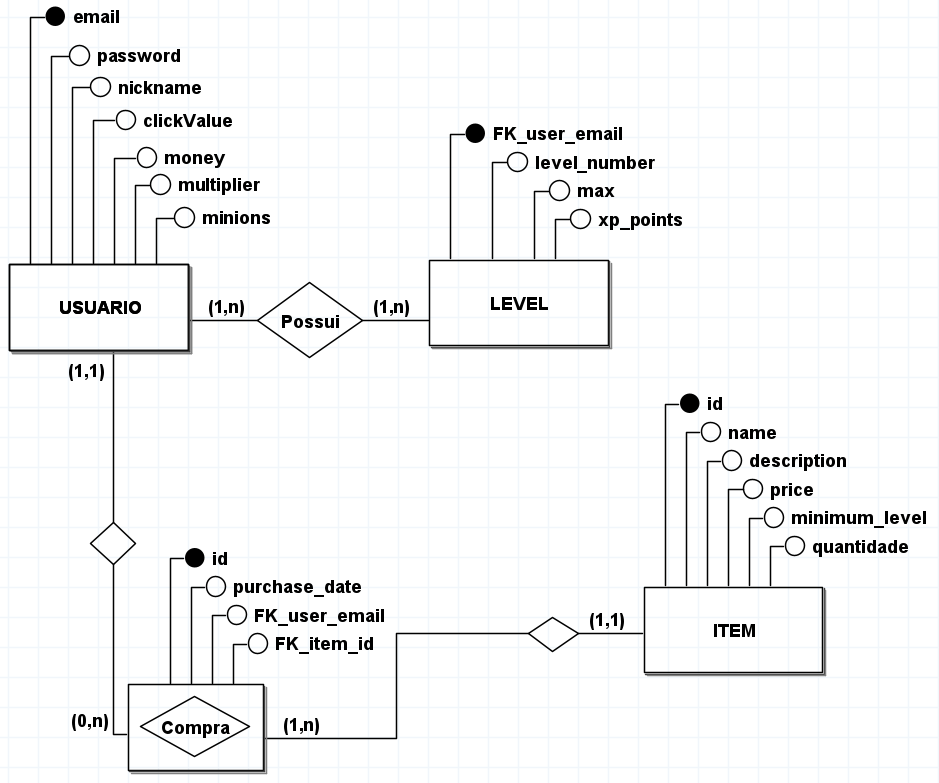
**Requisitos Não Funcionais**

|  |
| --- |
| **REQUISITOS** |
| **1RNF:** O software deverá ser desenvolvido utilizando as linguagens de programação PHP e JAVASCRIPT, para o banco de dados utilizará MYSQL, para estilização e estrutura utilizará HTML e CSS. |
| **2RNF:** A cada no máximo 5 dias, o desenvolvedor deverá verificar se o documento está com os requisitos atualizados referentes ao projeto. |
| **3RNF:** O projeto deverá utilizar o padrão de projeto MVC (*Model, View and Control*). |
| **4RNF:** O usuário deve ter seus dados carregados referentes ao jogo como itens e recursos adquiridos ao realizar o login. |
| **5RNF:** O jogo deve salvar os dados atuais do usuário a cada 1.5 segundos de maneira automática utilizando XMLHttpRequest. |

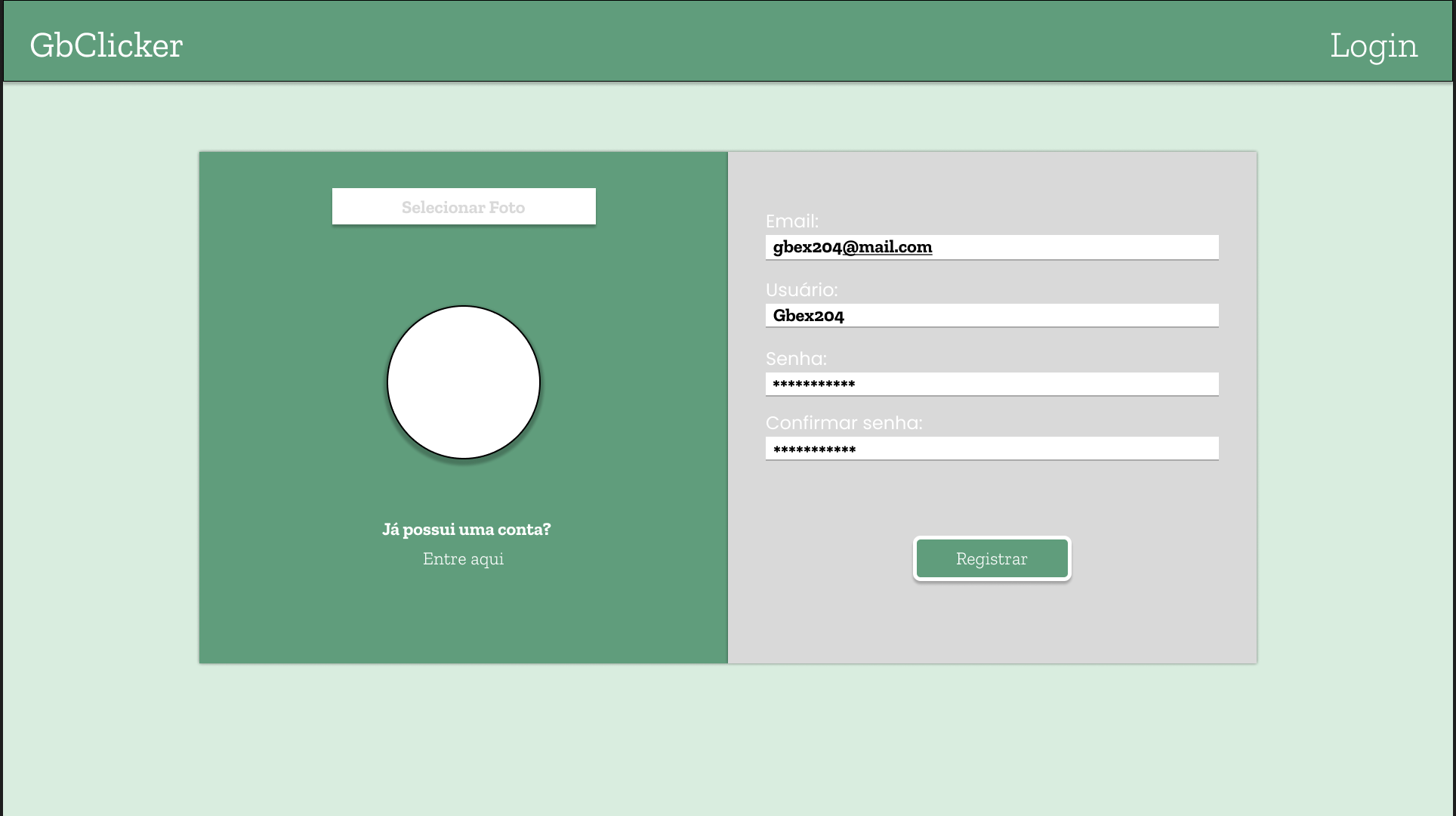
**Caso de uso**

****

**Modelo Relacional**

****

**Prototipação**

****

Página de registro do usuário



Página de login do usuário

**Problemas a serem resolvidos**

* Encontrar uma solução para quando a notificação esteja ativa, não seja necessária a divisão com width:100% para centralizá-la (presente na página de login e registro). Esta utilização impede o usuário de acessar o botão de login por 8 segundos (estipulados no Javascript).
* Além disso, o código ainda está um pouco confuso, principalmente caso seja necessário o aumento de atributos que devem ser salvos ou resgatados do banco de dados relacionado ao usuário (talvez uma atenção a mais seja necessária).
* Ao entrar em outras páginas que utilizam um objeto da classe game, ocorre tentativas de “atualizar” os dados presentes na página atual como ocorre na página inicial do jogo, onde são mostradas as mudanças no dinheiro atual, multiplicador e nível. Para corrigir esse erro deve ser verificado se ao buscar pelos *id’s* os textos utilizados para mostrar os dados são diferentes de nulo. Exemplo:

