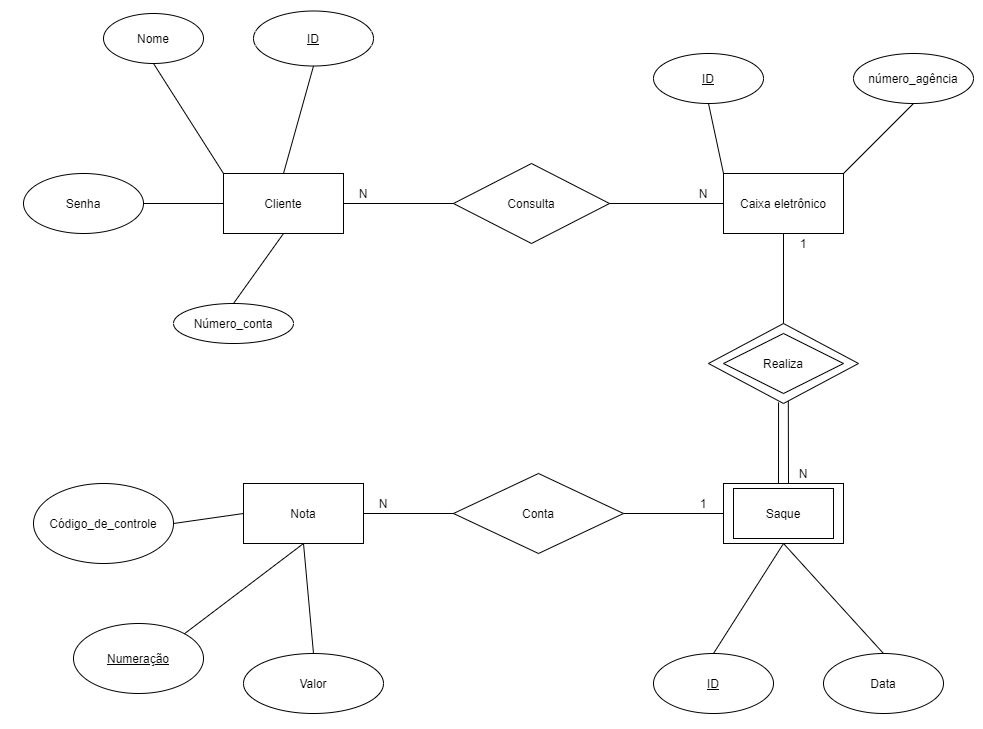
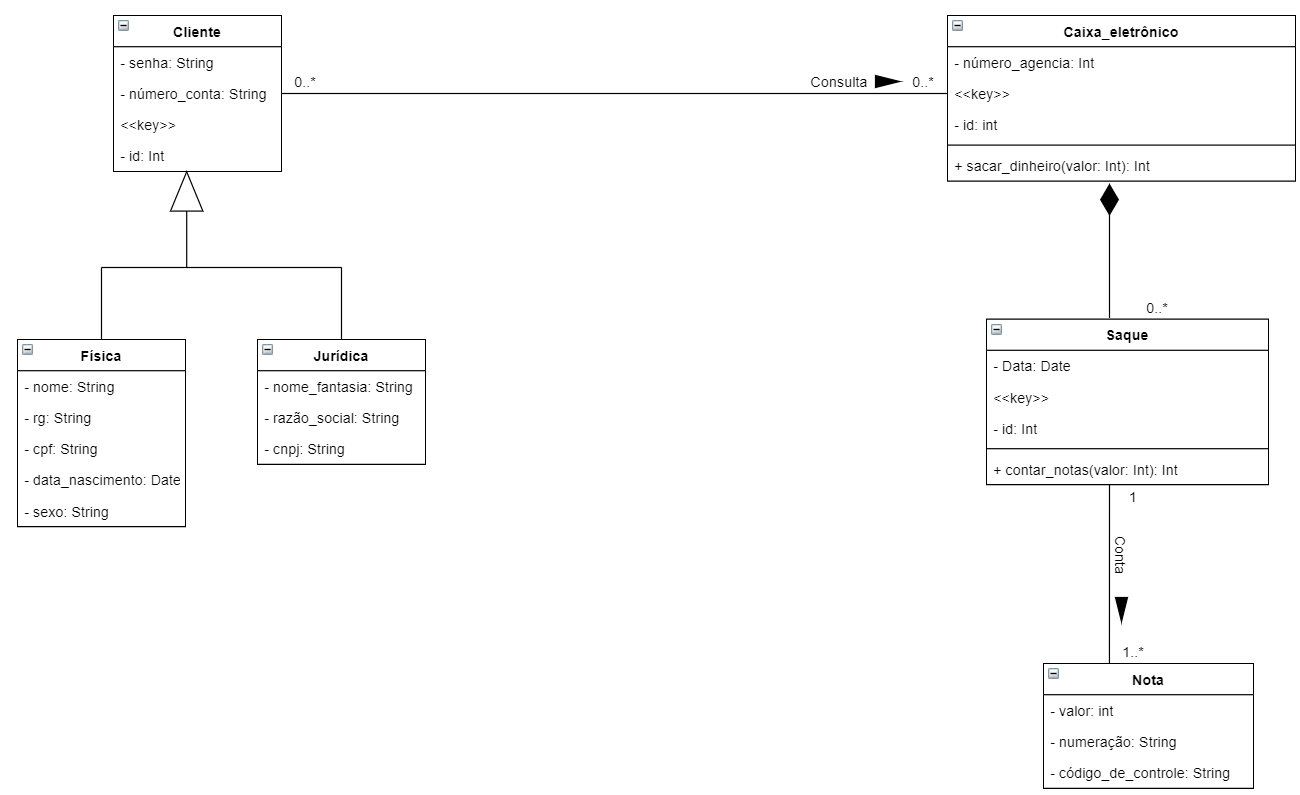
1.



2.



3.

A arquitetura N-tier (ou arquitetura multicamada) é uma arquitetura cliente-servidor onde as camadas de apresentação, processamento de aplicativos e gerenciamento de dados são separados fisicamente. O tipo mais usado é a arquitetura de três camadas nas quais são: **camada de apresentação**, **camada lógica** e **camada de acesso a dados**.

* **Camada de apresentação**: É a camada responsável pela apresentação, interação do usuário e recebimento de dados processados pelo software. Sua principal função é traduzir tarefas e resultados em algo no qual o usuário possa entender.
* **Camada lógica**: Essa é a camada que coordena a aplicação, realiza cálculos, processa os comandos e faz decisões lógicas e avaliações. Ela também move e processa dados entre as camadas de apresentação e a camada de dados.
* **Camada de acesso a dados**: Essa camada é onde as informações são armazenadas e acessadas de um banco de dados ou de um sistema de arquivos. A informação é então levada para a camada lógica para processamento, e então exibida para o usuário.

Deve-se ressaltar que a arquitetura N-tier pode ter mais que 3 camadas, isso irá depender de muitos fatores como regras de negócio entre outros.

4.

O **Hypertext Markup Language (HTML)** é uma linguagem de marcação utilizada para a construção de páginas na web. O HTML descreve como será a estrutura da página e, assim, dando significado a ela. O HTML é formado por uma série de elementos conhecidos como marcadores (tags) nos quais são os comandos de formatação da linguagem. Esses elementos são formados por um nome de marcador, atributos, valores e filhos. Os atributos modificam os resultados padrões dos elementos e os valores caracterizam essa mudança. Os documentos de HTML podem ser interpretados por navegadores.

O **Cascading Style Sheets (CSS)** é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em linguagem de marcação como o HTML. Ou seja, o CSS descreve como os elementos de HTML são exibidos. Assim, o CSS consegue adicionar estilo a uma página web como cor do background, tamanho e estilo das fontes usadas, posição das imagens entre outros exemplos. O CSS tem uma sintaxe simples no qual é composta em 3 partes: um **seletor** no qual declara qual parte do HTML um estilo será aplicado, uma **propriedade** e seu **valor** nos quais consistem em palavras chaves em inglês que darão uma propriedade a um elemento do HTML como tamanho, cor etc.

O **Javascript** é uma linguagem de programação de que permite a implementação de itens complexos em páginas web como por exemplo mostrar um conteúdo que se atualiza em um intervalo de tempo, mapas interativos, gráficos, animações, utilização de APIs entre outros exemplos. Através da API DOM (Document Object Model) podemos manipular o HTML e o CSS através dos códigos do Javascript, assim, podemos criar, remover e alterar os elementos do HTML. Por meio do DOM, o Javascript também permite aplicar novos estilos na página da web ou modificá-los. Portanto, através da API DOM, o Javascript pode modificar dinamicamente o HTML e o CSS.

5.

A diferença das arquiteturas MVC e SPA é que a forma de como elas funcionam. A arquitetura MVC (Model-View-Controller) foca no reuso de código e separa a aplicação em três componentes lógicos interconectados: o componente de modelo (model), o componente de apresentação (view) e o componente de controle (controller).

O componente de modelo consiste na parte lógica da aplicação, que gerencia o comportamento dos dados através de regras de negócios, lógica e funções. Portanto, trata-se do coração da execução. O componente de visualização pode ser qualquer saída de apresentação dos dados, como uma tabela ou um diagrama, ou seja, é onde os dados solicitados do modelo são exibidos. O controlador tem como foco a ação do usuário, onde são manipulados os dados que o usuário insere ou atualiza. O controle envia essas ações para o modelo e para a janela de visualização onde serão realizadas as operações necessárias.

Os aplicativos web feitos a partir da arquitetura MCV, para cada mudança como exibir os dados ou enviar os dados de volta para os servidores, demanda a renderização de uma nova página do servidor para o browser.

Já o SPA (Single-page application) é uma abordagem de design de website onde cada conteúdo novo da página é fornecido, não do carregamento de novas páginas de HTML, mas geradas dinamicamente através da habilidade do Javascript em manipular os elementos do DOM da própria página existente. Uma abordagem SPA permite que o usuário continue a consumir e interagir com a página enquanto novos elementos estão sendo atualizados ou coletados e, assim, resultando em interações e recarregamento de conteúdo muito mais rápidos. Na SPA não há a necessidade de se fazer requisições para carregamento de novas páginas.

Portanto uma das grandes diferenças entre os dois modelos é como as páginas são carregadas.

A vantagem do SPA em relação ao MVC é a possiblidade de otimização, em geral, da performance da aplicação ao deslocar todo o esforço de renderização para o cliente e permitir um tráfego de dados mais leve entre o cliente e o servidor. Outra vantagem é o reaproveitamento de código através de tecnologias como React ou Angular, possibilitando assim o desenvolvimento com menor esforço e mais padronizado. Assim, o SPA permite uma interação mais rica entre a aplicação e o usuário e fornece atualizações em tempo real, sendo útil para aplicações que utilizem notificações, dados de fluxo contínuo e gráficos que trabalhem em tempo real (gráficos da bolsa de valores etc.).

Entretanto para web sites que contenham muitas páginas estáticas, a introdução de SPA pode piorar os tempos de carregamento para o usuário, requerendo ao usuário que este abaixe e execute o payload do Javascript antes de poder visualizar qualquer conteúdo, neste caso uma aplicação em arquitetura MVC seria a mais recomendada. Outra vantagem do MVC em relação ao SPA é que para usuários que ainda utilizem browsers muito antigos ou tenham uma conexão de internet muito lenta, websites feitos em MVC podem funcionar melhor para eles.

Portanto a escolha de cada arquitetura vai depender do tipo de aplicativo web que está sendo desenvolvido.