### Desafio

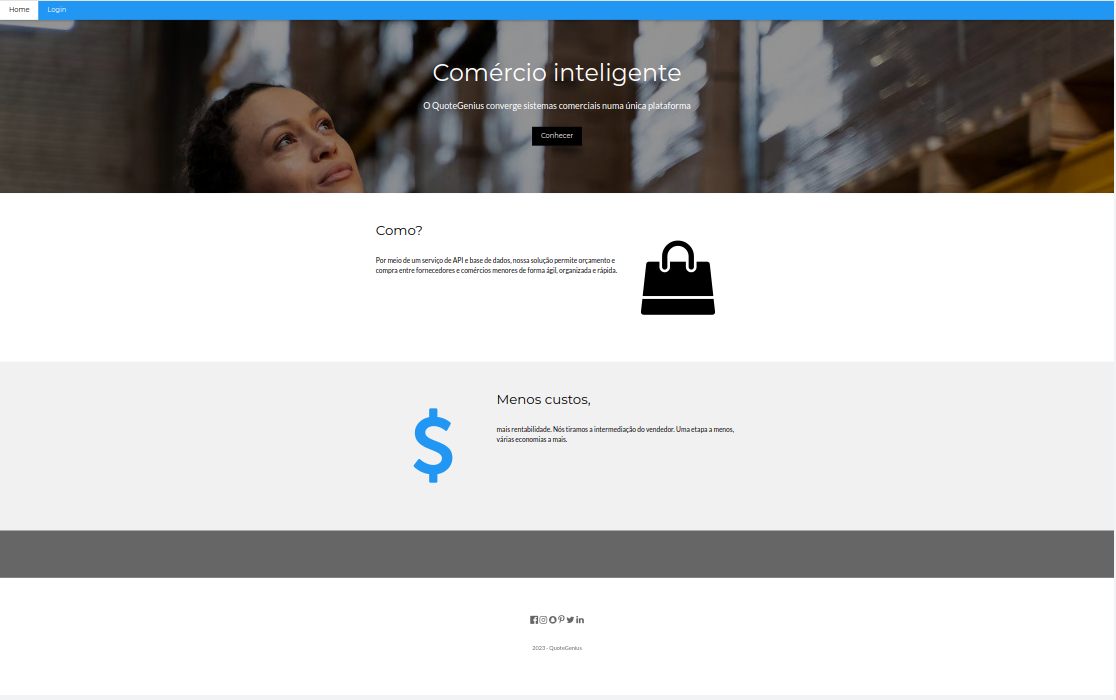
No dia a dia dos comércios, uma das tarefas que mais consome tempo é a de fazer compras para reposição. É comum ter que preparar uma listagem dos produtos, consultar vários fornecedores, comparar preços, tudo manualmente. Além disso, muitos vendedores não fornecem uma lista de todos os produtos disponíveis. Some isso ao desencontro entre as informações dos produtos entre as base de dados dos comerciantes, o que mostra o tamanho do problema.

O QuoteGenius atua como uma plataforma que visa unificar e padronizar as transações comerciais entre fornecedores e comércios menores, por meio de uma API REST, com a qual os desenvolvedores de sistemas de ERP iriam integrar seus softwares, provendo uma funcionalidade útil, lucrativa e produtiva para todos os participantes.

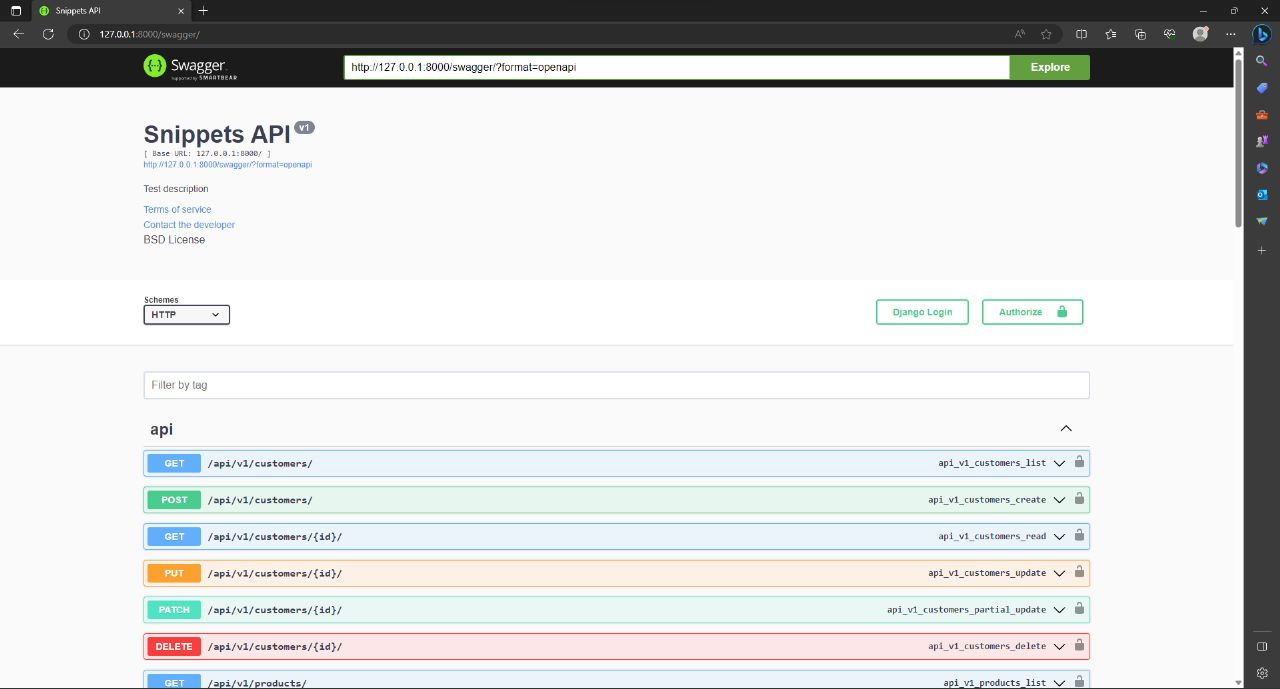
Outros detalhes estão nas páginas a seguir.

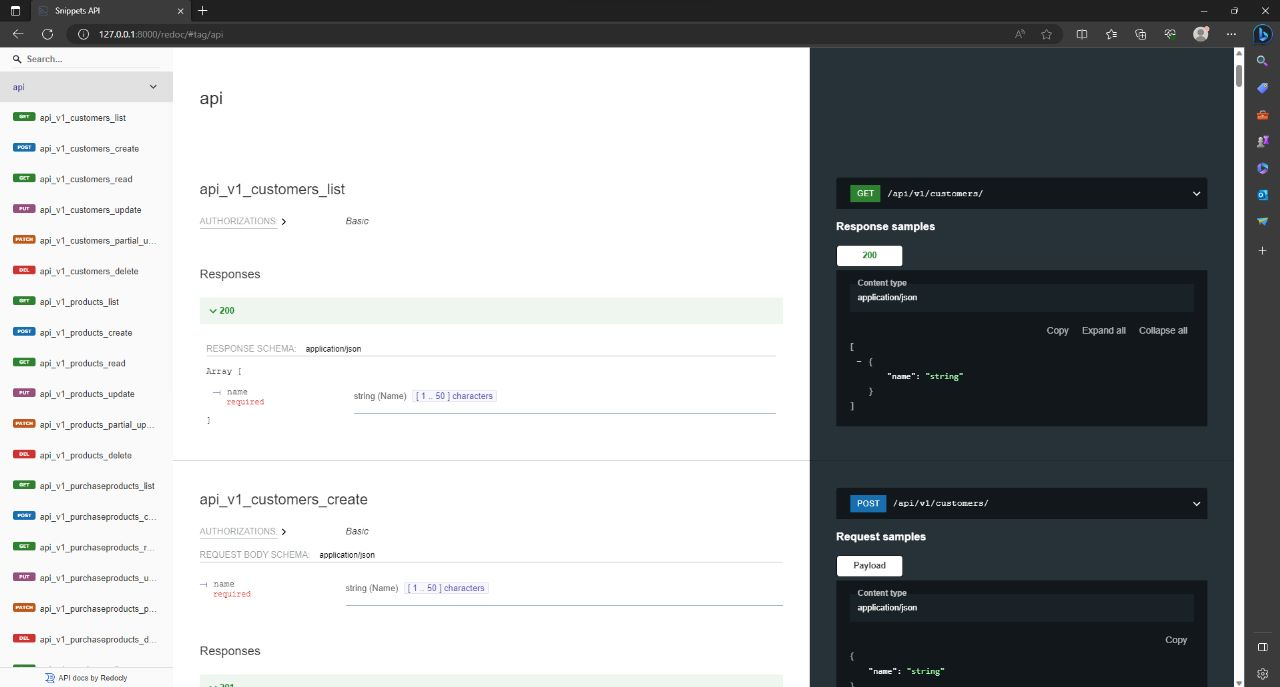
Nossa solução, por ser principalmente um backend voltado para outras empresas (Business to Business), não tem necessidade um front-end tão complexo.

Nossa página web inicial visa informar às empresas, que seriam usuárias da solução, do que se trata a nossa startup.

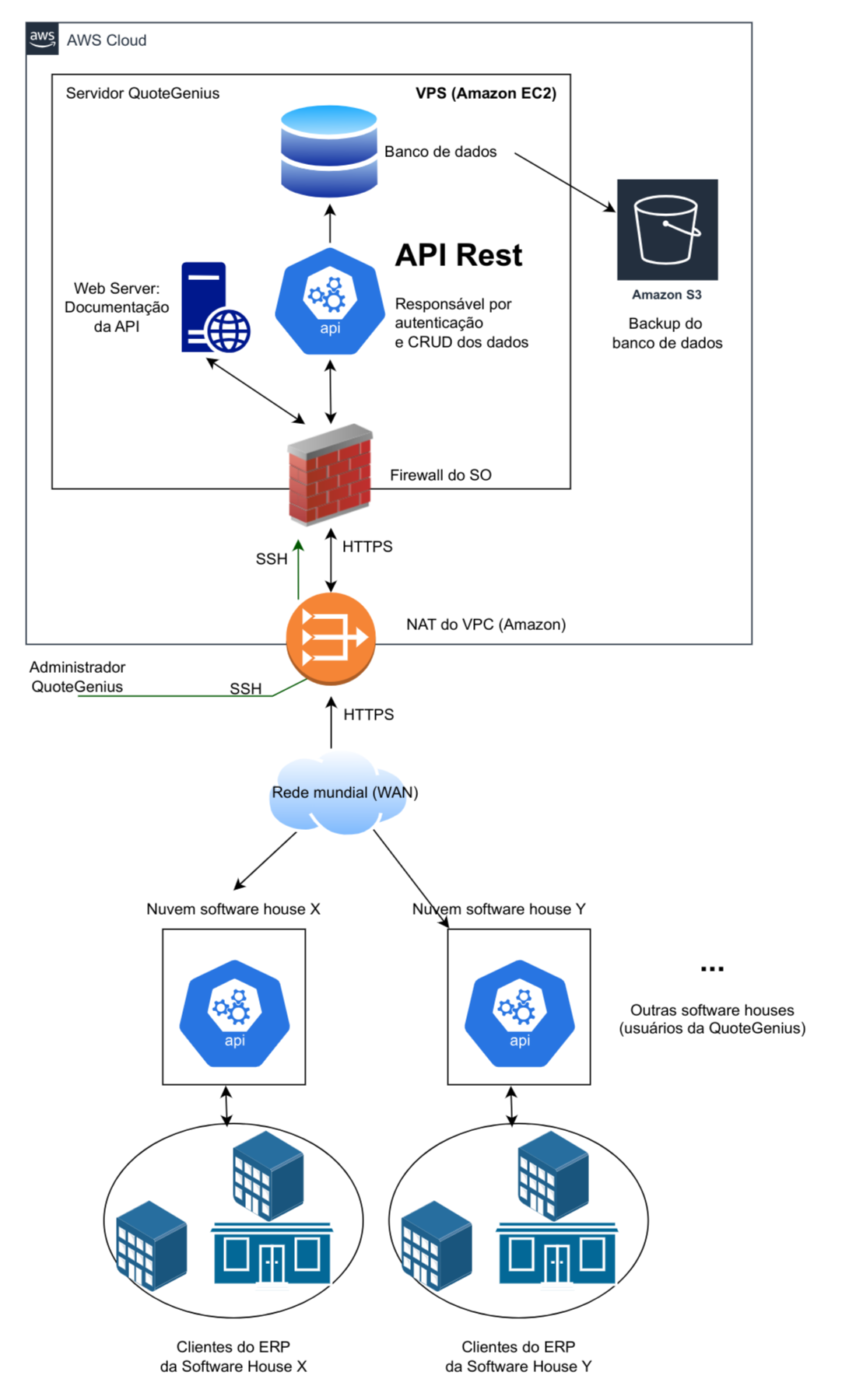


A documentação da API, porém, é rica e interativa:



Outra parte da documentação:

A arquitetura do nosso backend está a seguir.



Optamos por uma nuvem do tipo IaaS Infrastructure as a Service para hospedar nossa API REST, que no caso seria o serviço Amazon EC2.

No entanto, os backups são guardados externamente, no serviço de armazenamento de objetos Amazon S3. Essa é a primeira medida de segurança, para que evitar que os backups sejam perdidos caso a máquina virtual da Amazon seja comprometida.

O firewall do sistema operacional estará programado para somente permitir requisições HTTPS de entrada, que seriam os usuários da API, e SSH, que seria somente para nossas manutenções.

A Amazon também tem um sistema de rede virtual, com regras de firewall customizáveis. Essas regras seriam as mesmas em relação ao firewall do sistema operacional.

Parte das funcionalidades de segurança estão descritas acima. A outra funcionalidade que conhecemos e estamos prestes a implementar diz respeito a granularidade das permissões para cada token da API que os usuários iriam gerar. A ideia é que diferentes chaves sejam usadas pelos clientes, com variados níveis de privlégios.

Nosso banco de dados não fica exposto a rede mundial. A conexão direta a ele só pode ser feita dentro da mesma máquina, no caso, pelo nosso backend web.