

Implementação de um Web App de e-commerce de produtos customizáveis dentro da Cloud AWS

Leandro Yasuzawa Praxedes¹, Matheus Simões Pinsdorf¹, MSc. Joaquim Pessoa Filho¹

¹Instituto Presbiteriano Mackenzie, Faculdade de Computação e Informática, Ciências da Computação. Rua da Consolação, 930, CEP-01302-907 – S. Paulo, SP

{leandro.yp,simoes.mat}@hotmail.com, joaquim@mackenzie.br

Abstract. *With the decline in the popularity of monolithic systems, due to the labor market need to develop more flexible, scalable, and less coupling systems, together with the growth of cloud computing services, the study and implementation of an architecture in microservices of an e-commerce of customizable objects (this line of work that expanded in the COVID-19 pandemic), inside the AWS cloud (Amazon Web Services).*

Keywords: *Cloud Computing; Customizable Products; E-Commerce*

Resumo. *Com a queda da popularidade dos sistemas monolíticos, devido a necessidade do mercado de trabalho de desenvolver sistemas mais flexíveis, escaláveis e menos acoplados, aliado ao crescimento dos serviços de computação em nuvem, foi explorado o estudo e implementação de uma arquitetura em microsserviços de um e-commerce de objetos customizáveis (ramo este que expandiu na pandemia COVID-19), dentro da cloud AWS (Amazon Web Services).*

Palavras-chave: *Computação em Nuvem; Produtos Customizáveis; Comércio Eletrônico*

1. Introdução

Desde o decreto pandêmico do COVID-19 pela OMS, a convivência dentro de casa foi intensificada, levando a um aumento das compras on-line, hábito este que tende a se manter, mesmo após o fim da pandemia. Conjuntamente, estudos evidenciam que os clientes dão preferência à produtos customizados acima dos padronizados, de forma a satisfazer individualmente as demandas. Segundo Carpo (2011), tudo que é digital é por princípio variável, o que valida cada vez mais o investimento numa plataforma de e-commerce, seja por pequenas ou grandes empresas, outro fator de suma importância a ser considerado, é o “sétimo P” [Adolpho 1999], esse que é referente à personalização dos produtos, dentro do marketing digital.

Aplicativos de e-commerce vêm se popularizando cada vez mais com toda essa imersão tecnológica global, e junto com ele existem grandes desafios. Um dos tópicos mais importantes seriam os altos e baixos nos canais de venda, estes que podem ser, de forma macro, afetados pela sazonalidade do seu produto, ou simplesmente pelo horário do dia em relação ao seu maior público alvo. Outro problema também é a insegurança do cliente, por estar comprando algo que ele não consegue ver pessoalmente, pode gerar inicialmente uma certa desconfiança na credibilidade da loja e, posteriormente, contribuir para uma possível devolução do produto.

Em outro aspecto tecnológico, décadas atrás o mercado da Computação em Nuvem ainda era emergente [Kim 2009], esta que surgiu da necessidade de criar infraestruturas de TI complexas, onde as organizações podem configurar e administrar um recurso de computação multi-locatário [Fisher 2018], que antes poderia ser organizado apenas por grandes companhias. A principal dificuldade era ter infraestrutura dedicada, já que o custo de aquisição de maquinário é algo que é efetivamente abatido apenas no médio/longo termo.

O consumidor do serviço pode utilizar tudo de acordo com a sua demanda atual, trocando despesas fixas (datacenters, servidores físicos etc.) por despesas variáveis, e pagar apenas pelo que for utilizado [Amazon 2022]. Pequenas corporações se beneficiam da Computação em Nuvem de diversas formas, a principal delas é a de os provedores mais bem-sucedidos (Amazon AWS, Microsoft Azure e Google Cloud Platform) [Yu et al. 2010] de proporcionarem recursos escaláveis no estilo "pague pelo que for usado".

Estas provedoras, detêm mais da metade de todo o market share de Computação em Nuvem, em 2017 foi constatado que a AWS, Azure e Google Cloud dominavam, respectivamente 33 por cento, 11 por cento e 8 por cento do mercado. Atualmente os valores continuam bem semelhantes, apenas a Microsoft que se destacou por quase dobrar sua participação no mercado [Statista 2022]. De acordo com um estudo conduzido pela Grand View Research, o mercado de Cloud está previsto para chegar em um nível de USD 1.554,94 bilhões em torno de 2030, contabilizando todas as provedoras e suas modalidades (pública, privada e híbrida).

2. Referencial Teórico

O referencial teórico do atual artigo foi embasado em artigos científicos e livros, com enfoque no mercado de produtos personalizáveis, seu crescimento durante a pandemia do vírus SARS-CoV-2, e também sobre as infraestruturas de aplicações provenientes da utilização dos artefatos da computação em nuvem e seu impacto na evolução da engenharia de software, realizando alguns contrapontos com metodologias utilizadas anteriormente na história da tecnologia da informação.

2.1. Mercado de E-Commerce e Produtos Personalizáveis

A customização de produtos é a habilidade para fornecer produtos de forma a atender individualmente cada cliente, empresas que focam nesse quesito como estratégia para seus negócios estão buscando a diferenciação de seu produto, de forma que a diferenciação traga melhores alternativas ao consumidor [Bassi, Picchi e Gasparotto 2020].

De acordo com o relatório Next in Personalization de 2021, 71% dos consumidores desejam ter experiências personalizadas quando buscam um serviço e mais de $\frac{3}{4}$ ficam insatisfeitos quando essas expectativas não são atendidas.

Além do mais, nessa mesma pesquisa, 75% dos entrevistados relataram ter procurado uma nova forma de realizar compras durante a pandemia, o que indica que as empresas precisavam estar preparadas, pois para oferecer um ambiente agradável ao cliente, é papel chave que a comunicação que os websites apresentem se baseie na confiança para com os consumidores [Augusto, Santos e Santo 2020].

Concomitantemente, a expansão do e-commerce mundialmente desde a popularização da Internet (como ilustrado na Figura 1) era algo que trazia maior credibilidade às corporações que optavam por esse meio de disponibilizar seus produtos e serviços, já que hoje em dia, praticamente todos os produtos são ofertados no ambiente online, inclusive aqueles de subsistência das pessoas [Silva et al. 2021].

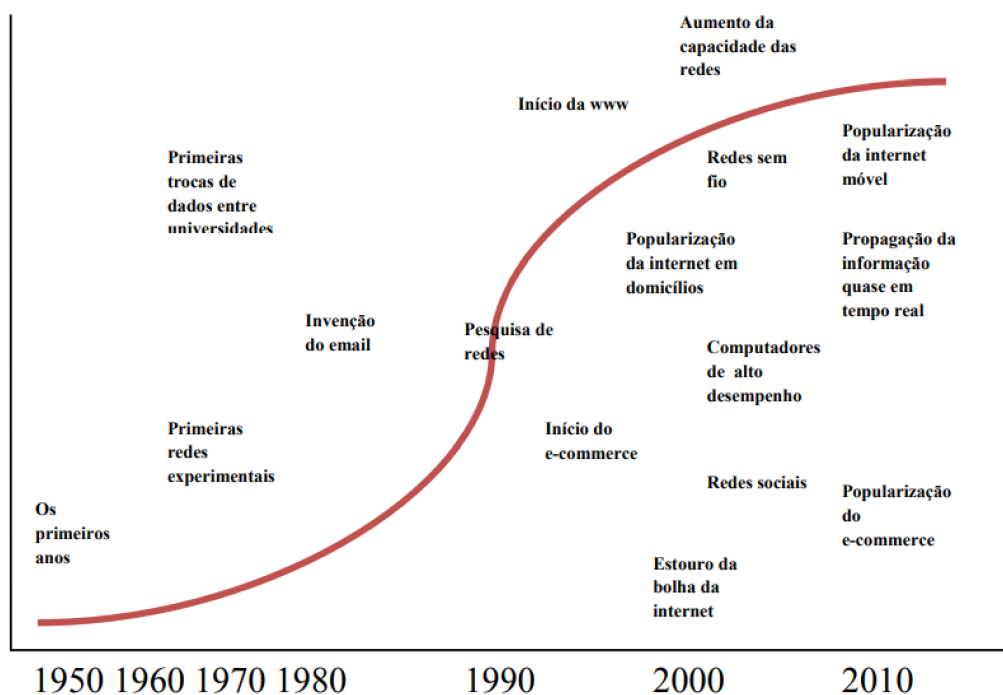


Figura 1. Evolução e Popularização da Internet. [Zimmerman 2013]

2.2. Engenharia de Software

Nos primórdios da Engenharia de Software, ainda não se sabia quais modelos seriam mais eficientes para o desenvolvimento de uma determinada solução, foi assim que surgiram os monólitos, ou então, um grande sistema criado de forma extremamente acoplada e dependente, que funcionava muito bem para pequenos serviços, pela facilidade de manutenção inicial, mas que tinha sua dificuldade aumentada com a expansão do mesmo, porém tinha certa facilidade de ser testado de forma integrada e gerir todas as funcionalidades de segurança e monitoração, por tratar-se de uma única peça, modalidade de desenvolvimento esta que entrou em dicotomia com sua sucessora, os microsserviços [Blinowski, Ojdowska e Przybyłek 2022].

Com o passar do tempo, grandes empresas passaram a encontrar diversos problemas nesse método, caso fosse necessário se adequar a uma nova tecnologia, era praticamente impossível desligar certa parte do sistema e reconectar com um código em outra linguagem, por causa da alta interdependência. Com a rotatividade das equipes e colaboradores, códigos que antes possuíam centenas de linhas, após décadas passaram a ter milhares, e alterar uma determinada parte do sistema, poderia facilmente afetar outras, dentre diversos outros problemas de possuir um simples monólito realizando todas as funções de um serviço.

Os microsserviços buscaram uma solução de desacoplamento, separando pelas mínimas funcionalidades possíveis, para que fosse facilitada qualquer implantação e manutenções no código em diferentes serviços [Blinowski, Ojdowska e Przybyłek 2022]. Dessa forma também, facilita-se a escalabilidade da aplicação, já que é possível focar em apenas uma peça da solução, sem a necessidade de subir a capacidade de todo o software e ter gastos desnecessários, além de permitir que diferentes linguagens de programação sejam adotadas para diferentes partes do programa, já que cada parte tende a ser totalmente agnóstica independente de seu propósito.

Um exemplo real disso é o banco dinamarquês Danske, suas operações de forex, como são majoritariamente na área bancária, era um grande monolito, para realizar a migração desse serviço foi adotada uma estratégia de dentro para fora, visando as maiores e mais importantes partes do software [Bucchiarone et al. 2018], transformando-as em um serviço isolado. Cada um desses microsserviços agora são implantados por meio da containerização (docker-compose), de forma que toda a arquitetura sobe conjuntamente, algo que antes era um trabalho manual. Dessa forma, o time que desenvolve a aplicação se beneficia muito, pois os deploys com falha agora são menos nocivos a toda a estrutura da aplicação, as integrações podem ser feitas de forma mais facilitada, e não estão presos a uma única linguagem de programação, além de ser possível monitorar mais facilmente o estado de cada peça isoladamente e extrair informações importantes de observabilidade sobre o negócio.

Assim funcionavam as infraestruturas on-premise, estas que se referem a todo o servidor, armazenamento e processamento que estão sob controle total da empresa em um local físico dela, além de todas as licenças de softwares e equipe de T.I que cuida da segurança e manutenção do sistema. Por ser uma solução inicial e custosa, esta dificultava a existência de startup's antigamente, já que seria necessário ter um local, comprar/alugar todo o hardware e licenças, para começar a empreender, o que apenas no longo prazo pode ter seu custo inicial recompensado.

Soluções em nuvem vieram com a premissa de oferecer toda essa infraestrutura por uma mensalidade, incluindo toda a gestão da segurança e disponibilidade do serviço, acessada diretamente do navegador web, algo que facilitou muito a entrada de novos players em diversas áreas da tecnologia, já que compensa muito no curto prazo alugar diversos serviços e funcionalidades e não se preocupar com configuração de software, updates, patches e licenças extremamente caros para um novo empresário.

Existem algumas modalidades que o serviço de nuvem é aplicado [Silva 2020], a primeira delas é a nuvem pública, esta é oferecida de forma que empresas possam compartilhar dados entre si por meio da Internet, onde toda a responsabilidade pela disponibilidade, confidencialidade e integridade dos dados é atrelado ao provedor de nuvem. Serviços oferecidos são diversos, incluindo processamento, armazenamento e máquinas virtuais. Diferentemente da Cloud Pública, a solução de nuvem privada visa as empresas que possuem estritas regulamentações e tráfego de dados sensíveis da própria corporação ou de clientes, com o objetivo de ter toda sua infraestrutura aberta apenas para a empresa para garantir uma maior segurança das informações. Já a nuvem híbrida é autoexplicativa, busca ter o melhor das duas anteriores, demarcando quais dados poderão ser trafegados de forma pública, e quais que necessitarão ser exclusivos de uma solução de nuvem privada.

A partir desses diferentes tipos de nuvem, vieram diversas implementações de escalabilidade [Silva 2020], visando reduzir qualquer desperdício que poderia existir no contexto da aplicação. Serviços têm um horário de maior acesso, alguns ramos de mercado possuem sazonalidade, também existem datas comemorativas onde a procura por um determinado serviço é maior, e é justamente essa problemática que fez sistemas de escalonamento em cloud emergirem, o uso de CPU, memória e armazenamento podem ser reduzidos ou expandidos dependendo da demanda deles, e de forma que apenas seja pago pela quantidade que está sendo utilizada no momento, coisa essa que seria irreal sem essa tecnologia.

A AWS possui cinco pilares em seu “Well-Architected Framework”, um deles é a Eficiência de Desempenho, onde um dos tópicos é a de tornar-se global em minutos [Amazon 2022], algo que é importantíssimo para quem trabalha com clientes de diversas partes do mundo, ter seus recursos provisionados em um datacenter do em qualquer lugar do mundo é fundamental caso seu público alvo não esteja localizado próximo.

Com a vinda da computação em nuvem, conjuntamente surgiu o termo “as a service”, produtos que são nomeados com esse pós-fixo referem-se à tecnologia de computação em nuvem que oferece a possibilidade de acessar dados ou funcionalidades que não necessitam estar próximas da rede e/ou máquina do usuário. Esse modelo traz grande vantagem já que as empresas conseguem distribuir soluções que podem atender seu público alvo independente de sua localidade [Silva 2020]. A solução Software as a Service (SaaS) abrange uma aplicação completa, disponibilizada diretamente para o cliente, um exemplo disso seriam os diversos serviços que já estão prontos para uso na AWS, de certa maneira a qual o cliente não necessita de controlar informações de infraestrutura, rede, armazenamento ou demais características não-específicas. Um exemplo característico são os serviços de e-mail, que podem ser acessados na web e não dependem de nenhuma configuração específica por parte do usuário. Já a plataforma como serviço (PaaS), oferece uma infraestrutura bem definida para implementação de aplicações dentro da cloud. O usuário trata apenas do seu próprio software a das informações que serão trafegadas, de forma a trivializar o desenvolvimento, já que as dependências que os desenvolvedores possuem serão reduzidas. Em certas ocasiões, o consumidor do PaaS pode sentir-se limitado, por não possuir controle em determinadas partes da infraestrutura, a infraestrutura como serviço (IaaS) veio para ser o pilar inicial, com o equilíbrio de responsabilidade da aplicação entre a provedora do serviço de nuvem e o desenvolvedor. Um exemplo disso é a própria tecnologia Amazon Elastic Cloud Computing (EC2).

Nuvens PaaS podem utilizar containers para gerir e orquestrar aplicações [Pahl 2015], containers estes que são uma solução nova em contraponto às máquinas virtuais utilizadas previamente, estas que eram extremamente custosas, por conterem todo o sistema em cima de um único host, levando aos mesmos problemas dos monolitos, a queda exponencial da velocidade de processamento em função do tempo e trabalho executado sob o software em questão. Containers buscam prover isolamento e gerenciamento de recursos [Dua, Raja e Kakadia 2014], seguem na mesma linha de desacoplamento referenciada nos microsserviços, de forma que exista independência entre os containers, de forma que segregue cada serviço para que possa ser tratado de forma isolada. Para controlar tudo isso, os containers são organizados em clusters e controlados a partir de uma Plataforma de Orquestração, esta que dispõe funções para gerir a infraestrutura, segurança,

implantação e rede dos containers [Khan 2017].

Web Services são outra pauta popular no ramo tecnológico na atualidade, este que auxilia na integração de sistemas, de forma a interligar aplicações existentes e se conectar com outras novas. Um Web Service provê flexibilidade para estabelecer comunicação entre sistemas separados geograficamente [Bora e Bezboruah 2015]. Essas informações são trocadas por meio de protocolos HTTP, um dos mais difundidos inicialmente foi o SOAP (Simple Object Access Protocol), que teve seu espaço tomado pela arquitetura REST (Representational State Transfer), esta que utiliza o padrão de URI's para ligar dois softwares diferentes, que é considerada muito mais performática em relação a seu antecessor [Hamad, Saad e Abed 2017], devido a essas aplicações, que são denominadas RESTful, trafegarem uma mensagem bem mais enxuta, com um tempo de resposta reduzido.

3. Desenvolvimento da Aplicação

Em vias de aprofundar-se no tema, foi desenvolvida uma aplicação de e-commerce com base na infraestrutura da nuvem. Para a implementação da aplicação, como primeiro passo tivemos a escolha da provedora de serviços, dentre elas: AWS, Google e Azure. Tendo estas como parâmetro, foi escolhida a AWS, plataforma esta que fornece serviços como Fargate, EC2, ECS, Lightsail e Elastic Beanstalk, os quais oferecem as condições necessárias para a implementação da arquitetura apresentada na Figura 2

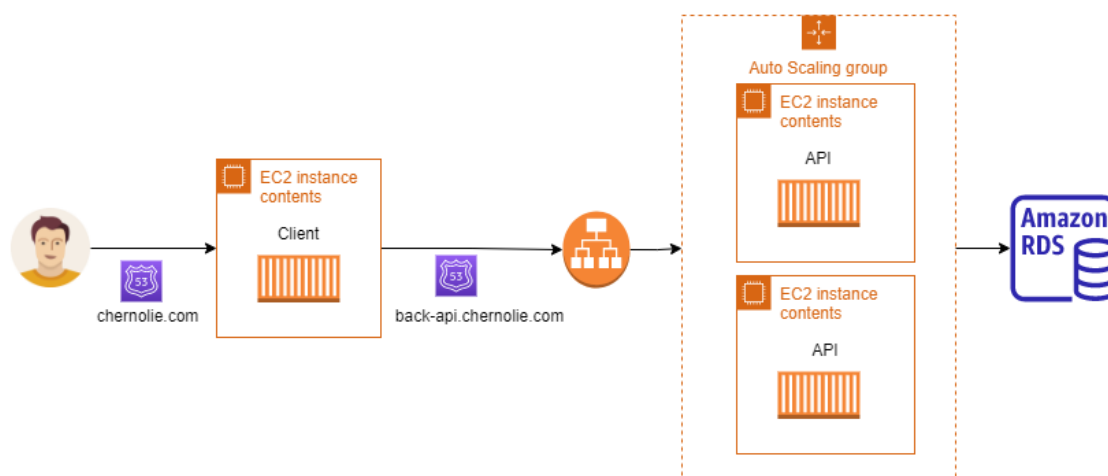


Figura 2. Diagrama de Componentes da Aplicação.

Para a concretização do desenvolvimento do sistema, foi idealizada uma aplicação de e-commerce de produtos customizáveis, como primeira empreitada, disponibilizamos apenas o serviço de personalização e venda de canecas no nosso editor. Ademais, foi desenvolvido um sistema de Explorar, onde os usuários poderão ver os produtos feitos por outros clientes, separados por tags, e adicionar diretamente ao seu carrinho caso os interessarem.

De início foram levantados os requisitos funcionais e não-funcionais da nossa aplicação. Os requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais [Sommerville 2007], funcionam como uma conexão entre os usuários do software proposto, os desenvolvedores e o sistema em si, de

forma que as necessidades do software estejam alinhadas em ambos os lados. Os requisitos funcionais (principais da aplicação destacados na Tabela 1), englobam aqueles em que o sistema deve fornecer alguma reação, dependendo do estímulo que for recebido, tal comportamento que pode ser explícito por fazer ou não fazer determinada ação.

Tabela 1. Requisitos Funcionais da aplicação.

ID	Cena	Usuário deve ser capaz de
RF1	Home	Se registrar
RF2	Home	Logar
RF3	Home	Deslogar da conta
RF4	Personalizador de Produto	Acessar o menu de inserção de texto
RF5	Personalizador de Produto	Fazer upload de uma imagem
RF6	Personalizador de Produto	Ajustar as posições dos itens no canvas
RF7	Personalizador de Produto	Remover os itens do editor
RF8	Personalizador de Produto	Desenhar no editor
RF9	Personalizador de Produto	Girar o modelo 3D ou dar zoom do produto
RF10	Personalizador de Produto	Deslogar da conta
RF11	Menu de Desenho	Escolher a cor da linha do desenho
RF12	Menu Publicar	Criar a descrição do seu desenho, adicionar tags e publicar seu desenho
RF13	Menu Descobrir	Buscar um produto pela descrição ou tags
RF14	Menu Descobrir	Girar o modelo 3D ou dar zoom em determinado produto

Por outro lado, os requisitos não-funcionais (Tabela 2), como já explicitado, não são relacionados diretamente com a funcionalidade do sistema, podem ser relacionados à segurança, tempo de processamento, espaço de armazenamento ou também a fluidez do software.

Tabela 2. Requisitos Não-Funcionais da aplicação.

ID	Descrição da Atividade
RNF1	O canvas deve apresentar tamanhos proporcionais ao produto real
RNF2	A representação do produto 3D deve ser fiel ao produto final
RNF3	A navegação no site deve ser simples e objetiva
RNF4	A Montagem da imagem, camadas de sobreposição e ordem de objetos devem ser claras
RNF5	Os ícones utilizados devem ser objetivos com as funções propostas
RNF6	O sistema deve informar os termos e condições de uso e autorização de imagem
RNF7	Os elementos de personalização de item devem ter posicionamento consistente
RNF8	A fluidez na navegação de telas deve ser suave e rápida
RNF9	O carregamento dos produtos no menu descobrir deve ser fluido

Sobre o desenvolvimento em si do sistema, foi utilizado no front-end o framework Angular, conjuntamente ao BabylonJS (modelagem 3D). Por tratar-se de uma estrutura de portabilidade muito fácil, foi a escolhida para facilitar a implementação do software no mobile.

Foi construída uma API RESTful em .NET Core, para estabelecer conexões HTTPS com o front-end, e realizar persistência de dados na base, esta é modelada dentro do SQL Server no serviço de Relational Database Service (RDS) da AWS, enquanto o client e a API estão isolados em máquinas EC2 distintas, sendo que o client possui um IP elástico fixo e o controle de acesso da API é gerenciado por meio de um Application Load Balancer (ALB) que distribui as cargas entre as instâncias.

Para tornar o processo escalável, foi criado um Auto Scaling Group para as máquinas EC2 da API, além de termos um Target Group dentro do nosso ALB para rotear o tráfego entre as instâncias.

Por tratar-se de um serviço de e-commerce, foi idealizado um sistema de autenticação, pedidos e pagamento dentro da base de dados, modelada de acordo com a Figura 3.

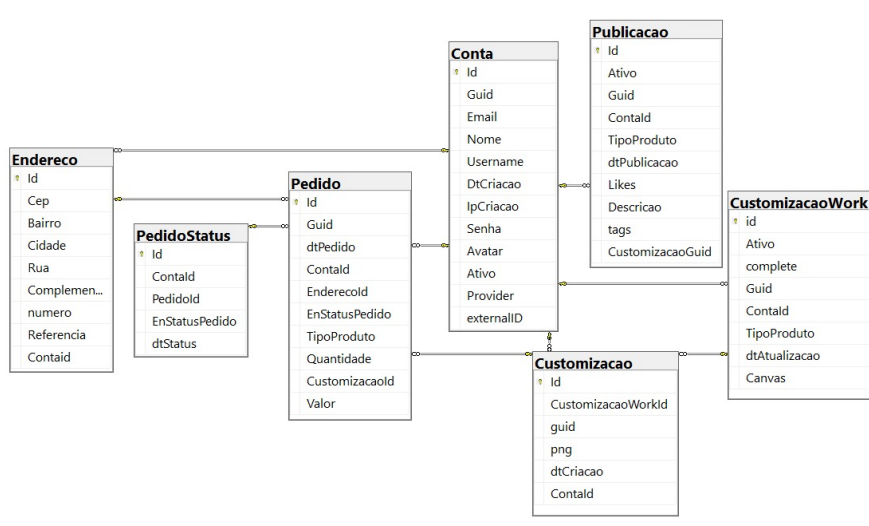


Figura 3. Modelagem de Dados da Aplicação.

4. Resultados

Baseando-se no que foi planejado na metodologia do desenvolvimento da aplicação, incluindo os requisitos funcionais e não-funcionais, diagramas de componente de modelagem de dados, a aplicação foi desenvolvida e hospedada na nuvem pública da Amazon AWS. Como é possível observar na Figura 4a, foi criada uma Home com uma apresentação breve do serviço provido pelo site, além de disponibilizar as opções de ir para os menus de personalização, explorar ou login/registrar.

À partir do Menu Principal, no canto superior esquerdo, é possível acessar o Menu de Explorar (Figura 4b). Nessa visão, apresentamos os produtos feitos pela comunidade, numa visão com o scroll infinito da página, e a opção de adicionar ao carrinho qualquer mercadoria que seja de agrado do usuário.

Dentro do Menu de Personalização de Produto, demonstrado na Figura 4c, é possível que o usuário crie seu produto de forma customizada, tendo um feedback que pode ser visualizado com um modelo 3D atualizado em tempo real com as suas alterações. Algumas das funções disponíveis para o usuário são as de inserir uma imagem e redimensioná-la de acordo com sua preferência, utilizar a função de Pincel, escolhendo uma cor e o tamanho da linha a ser traçada e por fim adicionar um texto, também podendo controlar a fonte e o seu respectivo tamanho. Todas essas inclusões podem ser posicionadas ou excluídas dentro do editor.

Quando o consumidor finalizar a edição de seu objeto, além de poder optar por concluir diretamente a sua compra e seguir para os meios de pagamento, ele possui a opção de salvar o rascunho para mais tarde ou até mesmo publicar o seu trabalho com tags específicas (Figura 4d) para aparecer no Menu Explorar da comunidade.



Figura 4. Telas da aplicação

4.1. Escalabilidade do serviço

Afim de validar a escalabilidade de nossa aplicação, realizamos a criação do ALB com as seguintes configurações: capacidade desejada - 1 instância; capacidade mínima - 1 instância; capacidade máxima - 4 instâncias. Por tratar-se de um teste em escala reduzida, utilizamos as máquinas alvo como t2.micro, instância esta que é a mais básica dentro do EC2, possuindo preços à partir de 0,0058 USD por hora, este tipo t2 é considerado uma das opções de instância do Amazon EC2 de menor custo [Amazon 2022].

Utilizando a ferramenta J-Meter, criada pela Apache, com o objetivo de facilitar a criação e execução de testes funcionais com a medição de performance de aplicações, foi realizado um teste de stress na API para validar o funcionamento do Load Balancer e do Auto Scaling Group criados. À partir disso, foi criado um cenário de teste com 20 threads (usuários) onde cada uma enviava uma requisição a cada segundo. Também foi configurado para que a escalabilidade dinâmica do Auto Scaling suba uma nova instância EC2 quando alguma máquina atinja um fluxo de entrada de rede de 3 megabytes/s.

Na Figura 5 é possível observar um pico no tempo de resposta, que é o exato momento que são instanciadas as novas máquinas, para suprir a quantidade de acessos, e terem suas respectivas cargas balanceadas pelo ALB.

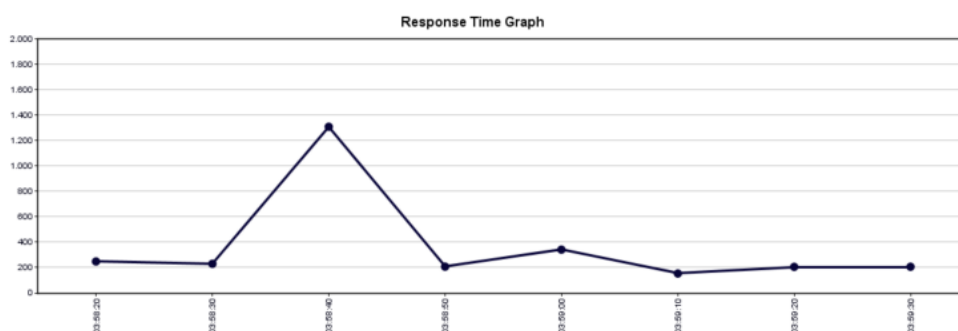


Figura 5. Tempo de Resposta da Aplicação - JMeter.

É exibida uma visão semelhante também nas métricas do Cloudwatch da própria AWS, onde é possível verificar a mesma medida anterior, juntamente do número de solicitações (Figura 6).

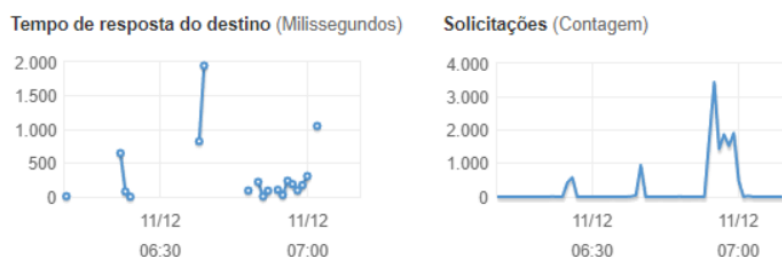


Figura 6. Cloudwatch - Tempo de Resposta e Solicitações.

4.2. Precificação

Mesmo sem toda a infraestrutura montada, é possível realizar cálculos de TCO (Custo total da posse) e gerar projeções financeiras baseado nos recursos a serem utilizados em determinada provedora, mas vale reforçar que isso não reflete o valor exato de montagem dentro da nuvem, já que podem existir outras integrações, impostos e funcionalidades que não são expostos de forma precisa nas calculadoras, por apresentarem certa complexidade de cálculo. Foram utilizadas as calculadoras de preço para estipular os valores médios das duas principais instâncias e banco de dados de nossa aplicação, da Microsoft (Figura 3) e da Google (Figura 4), com serviços com poder computacional semelhante aos utilizados no case atual.

Tabela 3. Custo Mensal Microsoft Azure.

Provedora	Categoria do Serviço	Tipo do Serviço	Descrição	Região	Custo mensal estimado
Azure	Computação	Máquina Virtual	2 A0 (1 vCPU, 0.75 GB RAM) x 30 Days (Pay as you go), Linux. (Pay as you go); 0 managed disks – S4; Inter Region transfer type.	East US	25,92
	Banco de Dados	Azure SQL Database	Single Database, vCore, RA-GRS Backup Storage, General Purpose, Provisioned, Gen 4, 1 - 1 vCore Database(s) x 730 Hours, 32 GB Storage, 0 GB Backup Storage	East US	13,11
				Total	39,03

Tabela 4. Custo Mensal Google Cloud.

Provedora	Categoria do Serviço	Tipo do Serviço	Descrição	Região	Custo mensal estimado
Google	Computação	Máquina Virtual	CP-COMPUTEENGINE-VMIMAGE-F1-MICRO x2 30 Days (Pay as you go) (1 vCPU, 0.6 GB RAM) Linux	Central US	7,76
	Banco de Dados	Cloud SQL (SQL Server)	Single Database, 720 Hours, 32 GB Storage, 0 GB Backup Storage	Central US	54,75
				Total	62,51

Por outro lado, com a arquitetura levantada dentro da AWS, consegue-se utilizar o Cost Explorer (Figura 7) para ter uma previsão de quanto seria o gasto mensal dessa solução, incluindo precificação das instâncias EC2, banco de dados RDS, Load Balancer,

dentre outros valores. Na solução atual, foi evidenciado o valor de 53,76 USD, preço um pouco acima da Azure. Lembrando que todas essas projeções foram mensuradas com base no mês de novembro de 2022.

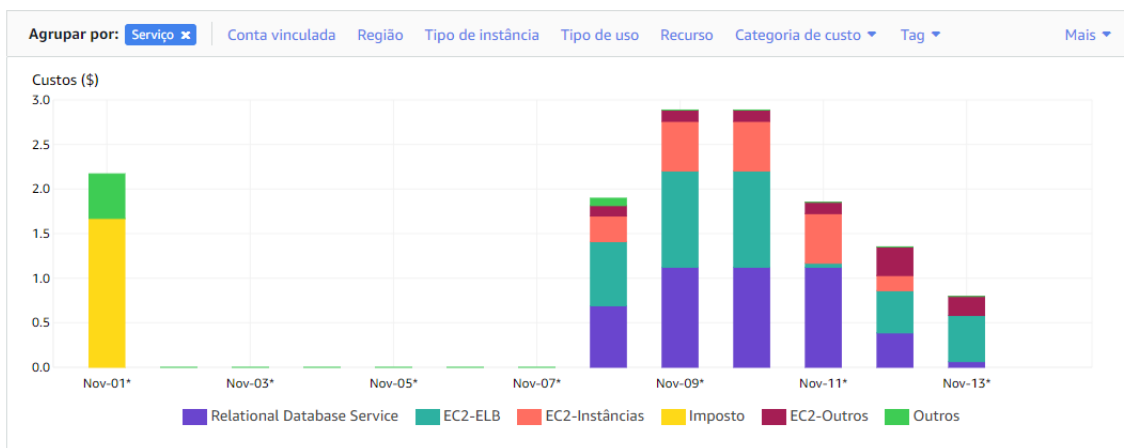


Figura 7. Cost Explorer - Custo mensal aproximado.

Para chegar a uma projeção mais próxima da realidade, é plausível também que haja especialização por parte do time, isso pode ser obtido por meio das provas de certificações, ministradas pelas respectivas provedoras de Cloud, que além de trazer um conhecimento sobre a precificação de cada tipo de produto, também facilita a construção de soluções, por exigirem um amplo conhecimento dos diversos produtos disponibilizados. Para soluções mais complexas, pode ser uma opção também utilizar um serviço de consultoria, como é o exemplo do Amazon Partner Network.

4.3. Considerações Finais

O artigo apresentou alguns cenários envolvendo o e-commerce e o mercado de produtos customizáveis, realizando um contraponto com a evolução simultânea do mercado de Computação em Nuvem, todos esses tópicos que se popularizaram nos últimos anos e que foram condensados por meio deste. Foi criada com sucesso uma aplicação totalmente estruturada dentro da Cloud, com a intenção de oferecer um serviço de customização de produtos pessoais (inicialmente canecas) com uma visualização 3D do objeto, aspecto de rede social com o compartilhamento de criações pela comunidade e estruturação mínima para viabilizar a construção efetiva de uma loja online focada nesse tema.

Foram utilizados recursos mínimos dentro da nuvem por questões de demonstração e teste, mas, como evidenciado, é factível o escalonamento desses processos de acordo com a demanda desejada, trazendo uma facilidade no quesito de expansão do negócio no momento que ele for efetivamente implementado.

Posteriormente também, é possível expandir as funcionalidades da aplicação, como por exemplo incluir novos produtos para serem personalizados, sistemas de recomendação de acordo com os usuários/produtos, maior diversidade de customização no editor, implementar sistema de logística, tópicos para representarem as filas de produção, além de podermos orquestrar uma arquitetura mais complexa dentro da Cloud para atender todas essas necessidades e realizar um plano de negócios para efetivamente viabilizar a entrada do nosso software dentro do mercado.

Referências

- ADOLPHO, C. *Os 8Ps do Marketing Digital*. [S.l.]: Novatec Editora, 1999.
- AMAZON. *The 6 Pillars of the AWS Well-Architected Framework*. 2022. Disponível em: <<https://aws.amazon.com/pt/blogs/apn/the-6-pillars-of-the-aws-well-architected-framework>>. Acesso em: 23 nov. 2022.
- AMAZON. *Gerenciamento financeiro na nuvem com a AWS*. 2022. Disponível em: <<https://aws.amazon.com/pt/aws-cost-management/>>. Acesso em: 11 nov. 2022.
- AMAZON. *Instâncias T2 do Amazon EC2*. 2022. Disponível em: <<https://aws.amazon.com/pt/ec2/instance-types/t2/>>. Acesso em: 14 nov. 2022.
- AUGUSTO, L.; SANTOS, S.; SANTO, P. Manuel do E. *E-Commerce: O Papel Principal Da Confiança*. 2020. 557–570 p. Disponível em: <<https://www.proquest.com/scholarly-journals/e-commerce-o-papel-principal-da-confianca/docview/2452331533/se-2>>. Acesso em: 09 nov. 2022.
- BASSI, W.; PICCHI, M.; GASPAROTTO, A. *Um estudo sobre a customização de produtos*. Revista Interface Tecnológica, 2020. 292–302 p. Disponível em: <<https://revista.fatectq.edu.br/interfacetecnologica/article/view/709>>. Acesso em: 31 mai. 2022.
- BLINOWSKI, G.; OJDOWSKA, A.; PRZYBYŁEK, A. *Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation*. 2022. 20357-20374 p. Acesso em: 29 oct. 2022.
- BORA, A.; BEZBORUAH, T. *A Comparative Investigation on Implementation of RESTful versus SOAP based Web Services*. [S.l.]: Research Gate, 2015. Acesso em: 06 nov. 2022.
- BUCCHIARONE, A. et al. *From Monolithic to Microservices: An Experience Report from the Banking Domain*. 2018. 50-55 p. Acesso em: 06 nov. 2022.
- DUA, R.; RAJA, A. R.; KAKADIA, D. *Virtualization vs containerization to support paas*. In: *2014 IEEE International Conference on Cloud Engineering*. [S.l.: s.n.], 2014. p. 610–614. Acesso em: 06 nov. 2022.
- FISHER, C. *Cloud versus On-Premise Computing*. 2018. Acesso em: 14 jun. 2022.
- HAMAD, H.; SAAD, M.; ABED, R. *Performance Evaluation of RESTful Web Services for Mobile Devices*. [S.l.]: IEEE, 2017. Acesso em: 10 nov. 2022.
- KHAN, A. *Key Characteristics of a Container Orchestration Platform to Enable a Modern Application*. 2017. 42-48 p. Acesso em: 29 out. 2022.
- KIM, W. *Cloud Computing: Today and Tomorrow*. 2009. Acesso em: 10 nov. 2022.
- PAHL, C. *Containerization and the PaaS Cloud*. 2015. 24-31 p. Acesso em: 30 out. 2022.
- SILVA, F. R. *Cloud Computing*. 1st. ed. [S.l.]: Pearson Addison-Wesley, 2020.
- SILVA, W. M. d. et al. *Digital marketing, E-commerce and pandemia: a bibliographic review on the brazilian panorama*. 2021. e45210515054 p. Disponível em: <<https://rsdjournal.org/index.php/rsd/article/view/15054>>. Acesso em: 23 nov. 2022.

SOMMERVILLE, I. *Engenharia de software*. 8th. ed. [S.l.]: Pearson Addison-Wesley, 2007.

STATISTA. *Cloud infrastructure services vendor market share worldwide from 4th quarter 2017 to 1st quarter 2022*. 2022. Disponível em: <<https://www.statista.com/statistics/967365/worldwide-cloud-infrastructure-services-market-share-vendor/>>. Acesso em: 12 nov. 2022.

YU, S. et al. Achieving secure, scalable, and fine-grained data access control in cloud computing. In: *2010 Proceedings IEEE INFOCOM*. [S.l.: s.n.], 2010. p. 1–9. Acesso em: 05 nov. 2022.

ZIMMERMAN, L. *E-commerce: origem, desenvolvimento e perspectivas*. UFRGS, 2013. Acesso em: 6 jun. 2022.