Relatório Projeto

Grupo II - Carolina Jesus, Madalena Duarte, Nuno Tempero, William Rodrigues

Módulo 8 - Segurança e Arquitetura de Sistemas

EM PARCERIA COM

COM O APOIO

ÍNDICE

1. ABSTRACT

Um grupo de livrarias de pequena dimensão, a operar no setor da literatura de autor e especializada, contactou a nossa empresa para desenvolver uma solução para aumentar o baixo número de vendas e a dificuldade de acesso ao mercado global nacional e internacional. Devido à incapacidade das pequenas e locais livrarias de acompanharem o mercado, que se encontra cada vez mais competitivo, estas têm vindo a perder clientes, assim como presença na indústria livreira.

Temos como principais concorrentes a Bertrand, Wook, FNAC, Livraria Lello, que são livrarias de maior dimensão e têm capacidade de assegurar uma forte presença *online*, reconhecimento nacional, uma vasta carteira de clientes e um alto nível de confiabilidade, e apresentam-se por isso como a nossa maior ameaça no mercado. No que toca a livrarias mais pequenas, locais, semelhantes àquelas que pretendemos abranger na nossa aplicação, o mercado é pouco vasto, e a oferta é mínima, já que, como referido anteriormente, as livrarias pequenas não têm uma forte presença *online*, e é exatamente por isso que pretendemos atuar sobre este nicho.

Estas livrarias locais que pretendemos ajudar desempenham um papel fundamental na propagação da cultura, na promoção do hábito de leitura e no acesso ao conhecimento. O encerramento corrente e frequente das mesmas apresenta-se assim como um problema àquilo que é o impacto cultural, social e educacional no nosso país. Para ultrapassar este desafio propusemos uma solução abrangente que combina um *website* promocional e uma aplicação móvel que corre em ambiente Android e iOS.

Tendo em conta o planeamento do desenvolvimento deste projeto, temos que ter em consideração a segurança na *web* e aplicações móveis, que é um tema de crescente importância nesta era digital. À medida que mais aspetos das nossas vidas se tornam dependentes da *internet*, e com o aumento das atividades *online*, desde compras, transferências financeiras, e comunicações pessoais e profissionais, a necessidade de garantir a integridade, confidencialidade e proteção dos dados de cada um tornou-se crucial.

Ao longo deste relatório iremos prever, analisar e corrigir as principais questões relacionadas com a segurança, identificando os desafios, as ameaças e os ataques que moldam a atualidade, e, consequentemente, podem vir a atacar a aplicação que pretendemos desenvolver. Exploraremos como as vulnerabilidades podem afetar a nossa empresa, e os vários indivíduos envolvidos, sejam eles os nossos funcionários, trabalhadores das livrarias clientes ou utilizadores da nossa aplicação, bem como as melhores práticas e tecnologias disponíveis, e soluções, para mitigar os riscos, tentar garantir a segurança, e uma boa recuperação caso um desastre aconteça.

2. INTRODUÇÃO

Para ajudar estas pequenas livrarias e contrariar a vida agitada e tempo insuficiente por parte do público-alvo para se deslocar até elas, pensámos numa solução que permita a qualquer utilizador verificar as livrarias perto de si, pesquisar livros do seu interesse, e comprar produtos com a possibilidade de entrega ao domicílio, proporcionando-lhe flexibilidade e conforto. Portanto, a nossa aplicação irá fornecer aos nossos utilizadores as ferramentas e a facilidade que estão em falta no mercado digital português. utilizando por isso uma aplicação de telemóvel, com o propósito de venda de livros.

As livrarias, por sua vez, beneficiam em ter, pela primeira vez, uma forte presença *online*, o que é essencial para que, enquanto empresa nos dias atuais, atinjam a expansão das suas vendas para além do público que possui morada próxima da sua localização. Isto irá possibilitar o aumento da margem de vendas e, consequentemente, dos lucros ao fim de cada ano, abrindo o leque de oportunidades para o crescimento da empresa e evitando que estas caiam na grande onda de encerramento de livrarias físicas, que tem vindo a afetar pesadamente o país.

A nossa solução integra dois componentes-chave: um *site* promocional de página única e uma aplicação móvel, disponível tanto para iOS como Android. O ***site*** funciona como um portal virtual, convidando os utilizadores a explorar todo o potencial da aplicação móvel. Com ressalte para livros em destaque, e uma visão geral das funcionalidades da aplicação, o *site* oferece um vislumbre do rico mundo literário que espera os utilizadores.

A ***app*** terá as seguintes características e funcionalidades:

- Disponibilização do total do *stock* de livros de todas as livrarias associadas, assim como a possibilidade de efetuar compras através da *app*, com possibilidade de entrega ao domicílio, assim como de levantamento em loja, e venda de produtos relacionados, de *merchandising*.
- Possibilidade de adicionar produtos à sua *wishlist*.
- Comparação de preços entre as várias livrarias, assim como entre os vários livros, para permitir ao *user* optar pelo livro que mais se adequa às suas necessidades.
- Comunicação via notificações, de forma a atualizar o *user* relativamente a novos livros, promoções e produtos.
- Disponibilização de uma rede social privada, dedicada à literatura, para todos os membros com conta na *app* e pelo menos uma compra efetuada, de forma a permitir a interação, e alimentar e promover a sensação de comunidade.
- Divulgação de informação sobre livrarias locais, mostrando ao *user* as livrarias que se encontram perto de si, assim como fotografias, contactos, morada completa, e uma pequena descrição.

3. REQUISITOS DE SEGURANÇA

Este relatório apresenta uma visão abrangente dos requisitos de segurança necessários a ter em consideração para o bom planeamento e desenvolvimento da nossa aplicação, tendo em conta as funcionalidades que delineamos. Os requisitos de segurança desempenham um papel fundamental na proteção dos dados dos utilizadores, na integridade do sistema e na confiança dos utilizadores nas operações da aplicação. Estes foram definidos com base nas funcionalidades previamente delineadas e apresentadas, e nas medidas de mitigação de problemas, assim como nos ataques mais frequentes nos dias atuais, em mente.

3.1. Proteção contra Ataques de Injeção de Código

Ao longo, e em vários momentos no processo de utilização da nossa aplicação, temos campos para o utilizador inserir dados. Logo na fase inicial, de *Login* e/ou de *Sign Up*, existem dois campos de inserção de dados, que permitem o acesso do utilizador a uma conta. Tem de seguida novas opções de inserção de dados pessoais, de forma a personalizar a sua conta. Também temos um *input* de pesquisa, que permite ao utilizador pesquisar por palavras-chave que o levam de encontro àquilo que procura. Adicionalmente, temos a possibilidade de adicionar publicações e fazer comentários. Todos estes *inputs* podem ser vistos como "portas" de entrada para a nossa aplicação, caso sejam utilizados de forma incorreta, e não estejamos preparados para os deter.

É por isso importante implementar medidas de segurança para evitar ataques de injeção de código, tais como: o XSS, que permite injetar *scripts* maliciosos que são executados nos navegadores de outros utilizadores; o CSRF, que permite induzir um utilizador autenticado a executar ações não autorizadas num *site*; e o SQL Injection, onde os atacantes inserem comandos maliciosos em consultas SQL, que lhes permite manipular, alterar ou aceder a informação proibida, indevidamente. Os métodos de proteção contra este tipo de ataques incluem:

- A **validação e filtragem** adequada dos dados inseridos pelos utilizadores, de forma a garantir que os dados de entrada não são interpretados como código.
- O **impedimento da concatenação direta** de dados de entrada em consultas SQL, ou seja, impedir que um atacante introduza comandos de manipulação de SQL diretamente num pedido à base de dados, e para isso utilizaremos *prepared statements* e recorreremos à implementação de listas de permissões de acesso aos dados.
- Manter o ***software***, **bibliotecas** e **frameworks sempre atualizados**, para que possamos garantir a correção de vulnerabilidades conhecidas.
- Tirar partido de **bibliotecas** que nos auxiliam na proteção contra estes ataques, às quais podemos recorrer, como por exemplo o Express-Validator.
- Para nos ajudar a impedir ataques CSRF, iremos incluir **Tokens Anti-CSRF**, e implementar um tempo limite para os mesmos, assim como verificar as origens dos pedidos.

3.2. Proteção de Dados Pessoais

Tendo noção da importância de manter os dados dos nossos utilizadores seguros, e sabendo que a nossa aplicação lida com informações pessoais bastante sensíveis, incluindo nomes, endereços, histórico de compras, informações relativas aos seus cartões de crédito / débito, e NIFs, é da mais alta prioridade manter estes dados devidamente escondidos e inacessíveis a indesejados. O acesso indevido a este tipo de dados seria de uma gravidade elevada, podendo resultar em fraudes, roubos ao efetuar compras com as informações dos cartões de utilizadores, e *identity theft*.

Para garantir a conformidade com regulamentos de privacidade, como o RGPD, implementaremos medidas de proteção de dados, incluindo:

- A **encriptação** de informações pessoais, de forma a que, mesmo que acedidas, estas informações não possam ser compreendidas, ou descodificadas, e tornam-se por isso praticamente "inúteis" para atacantes.
- A obtenção de **consentimento** dos usuários para a coleta e processamento de dados.
- Para garantir a proteção das informações financeiras dos utilizadores, como números de cartão de crédito, iremos ter em conta o cumprimento dos padrões de segurança de dados de pagamento, como o PCI DSS.

3.3. Política de Senhas Seguras

Tendo em conta que a nossa aplicação irá utilizar contas próprias para cada utilizador, iremos necessitar de armazenar informação relativa às *passwords* de cada um. Conhecendo os riscos evidentes que correríamos ao guardar essa informação em *cleartext*, como os ataques de acesso às bases de dados, e o *password sniffing*, que permite a um atacante escutar e obter informações que são transmitidas pela rede, optámos por guardar essa informação em *hash*, através de funções de resumo, devido à sua eficiência, resistência a colisões, e resistência à reversão, sendo que não existe uma lógica para reverter o *hash* de volta para a *password* inserida originalmente.

Mesmo tirando partido destas funções de resumo, continuamos expostos a outros tipos de ataques: os ataques de Dicionário e *Bruteforce*, que consistem em testar *passwords* em massa, rapidamente; e os ataques de *Lookup Table*, que permitem corresponder um *hash* com a sua palavra original, ou seja, partir do *hash* e, utilizando estes dados que podem ser encontrados *online*, chegar à *password* de um utilizador. Para garantir que estamos precavidos contra estes ataques teremos que:

- Garantir que os nossos utilizadores optam por ***passwords* seguras**, e por isso exigir senhas fortes, com um mínimo de 8 caracteres, letras maiúsculas e minúsculas, e que terão também que incluir números e caracteres especiais. Isto dificulta muito os processos de ataque referidos já que, quanto mais complexa e diversa for a *password*, menor a probabilidade de um ataque de *Brute Force* encontrar a *password* correta, no meio das infinitas possibilidades que testam. Irá também diminuir em muito as chances dessa *password* em específico se encontrar nas bases de dados de correspondência com a sua versão resumida.
- Incentivar a **alteração regular de senhas**, mantendo os nossos utilizadores cientes da importância da segurança dos seus dados, assim como guardando o tempo passado desde a última alteração, perto das suas informações pessoais, na sua conta.

3.4. Gestão de *logs* de Segurança e Controlo de Acessos

Para registar e monitorizar, de forma eficaz, as atividades relevantes relacionadas com o nosso sistema, servidor e aplicação, iremos implementar uma gestão de *logs* de segurança, que nos permitirá a deteção e resposta a incidentes de segurança, e nos ajudará a proteger a confidencialidade, integridade e disponibilidade dos objetos, dados e informação, ao:

- **Manter registos** de *logins*, *logouts*, e devidas tentativas de sucesso ou insucesso.
- Implementar um **controlo de acessos** com autenticação segura, baseado em papéis, o que significa que as autorizações e decisões de controlo são tomadas com base nas funções desempenhadas por cada um, o que fará com que cada papel tenha as suas próprias capacidades de acesso. Isto deverá incluir a gestão de direitos de acesso para utilizadores e livrarias.

3.5. Gestão de Sessões Seguras e Sessões Inativas

Para uma boa gestão de sessões seguras, é crucial adotar medidas para proteger a integridade das sessões dos usuários, e diminuir os riscos a que estas se encontram expostas. Iremos implementar:

- ***Timeouts* de inatividade,** onde estabeleceremos *timeouts* para encerrar sessões automaticamente, após um período de inatividade, garantindo a proteção contra acessos não autorizados e minimizando os danos em caso de comprometimento da sessão.
- ***Cookies* seguros**, utilizaremos *cookies* marcados como "*secure*" e "*HttpOnly*" para evitar ataques de *script* entre sites e garantir que os *cookies* sejam acessíveis apenas por conexões seguras.
- **Autenticação baseada em tokens**, recorrendo a autenticação baseada em *tokens*, garantindo assim que eles são gerados e gerenciados com segurança para evitar interseções ou manipulações não autorizadas.
- **Auditoria de sessões**, ao mantermos registos detalhados das atividades das sessões para detetar eventos incomuns ou possíveis violações de segurança.
- **Monitoramento contínuo**, onde implementaremos um sistema de monitoramento ativo para identificar e responder a ameaças à segurança das sessões dos usuários.

Essas práticas reforçam a segurança das sessões dos usuários e contribuem para a proteção geral do sistema contra ameaças cibernéticas, mantendo um equilíbrio entre segurança e experiência do usuário.

3.6.Proteção contra *Phishing*

A consciencialização contra o *phishing* é uma parte crítica da nossa estratégia de segurança. O *phishing* é uma tática comum, de engenharia social, usada por atacantes com o objetivo de enganar os utilizadores, de forma a obter informações confidenciais, como dados de pagamento. Para proteger os nossos clientes, implementaremos as seguintes medidas:

- **Mensagens de alerta** e notificações na nossa plataforma para alertar as pessoas sobre atividades suspeitas como *logins* noutros dispositivos ou pagamentos de grande volume.

- Permitiremos que os **utilizadores verifiquem a autenticidade** dos *logins* ou pagamentos grandes e caso haja algum problema de segurança, estes possam denunciar as possíveis atividades de *phishing*. Quando uma denúncia é feita, tomaremos medidas imediatas para investigar e resolver as situações reportadas.

A consciencialização contra o *phishing* não apenas protege os nossos utilizadores, mas também contribui para a construção de um ambiente *online* mais seguro e confiável. Encorajamos os nossos *users* a relatar qualquer atividade suspeita para que a nossa equipa de segurança possa agir rapidamente e tomar as medidas apropriadas.

3.7. Formação em Segurança, *Backups* e Recuperação de Dados

É de grande importância manter os nossos funcionários cientes dos riscos a que estamos sempre expostos. Sabemos que, mesmo com todas as medidas de segurança bem implementadas, basta um descuido para dar oportunidade a um atacante de "entrar" no nosso sistema. Iremos por isso ter:

- **Formações frequentes** sobre a importância da segurança, cenários e comportamentos a ter em atenção e consideração, e o que devem fazer para minimizar ao máximo os riscos. Serão obrigatórias à equipa de desenvolvimento e restantes funcionários, e irão manter-se atualizadas com base em ataques mais comuns em determinada situação, ou tempo. Terão como fim aumentar a consciencialização sobre as melhores práticas de segurança.
- ***Backups* regulares** de toda a informação relevante, para nos certificar de que os dados são armazenados de forma segura e podem ser recuperados em caso de incidentes de segurança.

3.8. Comunicação de Incidentes de Segurança

Precisamos de estar preparados para receber *feedback* relativamente a quaisquer questões de segurança por parte dos nossos utilizadores, porque, por mais que nos esforcemos para contrariar todos os ataques, podemos não o conseguir fazer. Estabelecer um protocolo claro que permita e incentive a comunicação de incidentes de segurança, irá dar espaço aos utilizadores para relatar problemas que surjam, e irá garantir que temos todos os meios para que os incidentes sejam tratados de acordo com as melhores práticas. A nossa abordagem para a comunicação de incidentes de segurança envolve:

- **Canais de comunicação** como formulários de contato, endereços de email de apoio dedicados, ou sistemas de suporte ao cliente. Fornecer este tipo de suporte aos utilizadores fará com que estes se sintam mais seguros, e percebem que fazem parte, e são peças importantes, do crescimento e melhoria da nossa aplicação e serviço.
- Comprometer-nos a dar **respostas rápidas** a qualquer relatório de incidente de segurança, de forma a demonstrar o nosso compromisso com a segurança dos nossos utilizadores, e consequentemente mostrar-lhes que estamos a agir para resolver os problemas relatados.
- Garantir que os nossos relatórios de incidentes de segurança são tratados com o mais alto grau de **confidencialidade,** encorajando os utilizadores a relatar problemas sem receio de exposição.

- Usar as informações dos relatórios de incidentes para **melhorar continuamente** as nossas medidas de segurança, e prevenir futuros incidentes.

Acreditamos que, ao estabelecer esta abordagem eficaz de comunicação de incidentes de segurança, iremos demonstrar o nosso compromisso para com a segurança e satisfação dos nossos utilizadores.

3.9. Testes de Segurança

A realização de testes regulares de segurança é uma prática fundamental para identificar e mitigar possíveis ameaças e vulnerabilidades na nossa app. Para garantir a robustez da segurança, prevemos a implementação dos seguintes procedimentos:

- Pretendemos realizar testes de segurança ou avaliações de sistema, simulando ataques controlados para avaliar a eficácia dos nossos métodos de defesa. Isto vai ajudar a corrigir as vulnerabilidades antes que possam ser exploradas por hackers;
- A realização de análises periódicas para identificar e avaliar potenciais vulnerabilidades no nosso sistema, app e infraestrutura vai-nos permitir adoptar medidas preventivas mais eficazes e rápidas;
- Quando identificamos vulnerabilidades ou problemas de segurança durante os testes, tomamos medidas imediatas para corrigir esses *bugs*. dando prioridade à resolução rápida dessas questões;
- Usaremos os resultados dos testes de segurança para estar constantemente a melhorar as nossas políticas, procedimentos e práticas de segurança.

A realização de testes regulares de segurança não apenas reforça a proteção da nossa aplicação e serviço, mas também demonstra o nosso compromisso contínuo em manter um ambiente seguro para os nossos utilizadores e seus dados.

A implementação eficaz destes requisitos de segurança é essencial para garantir a proteção dos dados dos utilizadores e a integridade do sistema da aplicação da livraria. A segurança deve ser uma consideração contínua ao longo do ciclo de vida do projeto e da operação da aplicação, para manter a confiança dos utilizadores e a conformidade com regulamentos aplicáveis. É fundamental que a equipa de desenvolvimento e gestão esteja comprometida com a manutenção de padrões rigorosos de segurança.

4. PROPOSTA DE ARQUITECTURA DE SISTEMA

A escolha da arquitetura Model-View-Controller (MVC) em conjunto com microserviços para a arquitetura de software da nossa aplicação *mobile* de venda de livros *online,* é uma decisão estratégica que traz diversos benefícios. Neste contexto, a combinação do MVC com microserviços proporciona uma abordagem escalável, flexível e altamente adaptável, alinhada com as exigências e desafios de uma loja de livros *online*. Os critérios que orientaram nossa escolha foram:

1. Separação de Responsabilidades

O MVC permite separar as responsabilidades da aplicação de forma clara, dividindo-a em três componentes distintos. Isso é particularmente valioso numa aplicação complexa como a nossa, onde é essencial manter isolados os: aspectos de negócios (*Model*), dados e gestão do mesmo; a apresentação de informações (*View*), as interações do utilizador; e o *Controller,* intermediário entre *View* e o *Model*, pois facilita a manutenção e a evolução da aplicação.

2. Escalabilidade

A nossa aplicação de venda de livros *online* pode ter um grande número de utilizadores e operações concorrentes e simultâneas. A arquitetura de microserviços ajuda a dividir a aplicação em serviços menores e independentes, em que cada um cuida de funções específicas, tais como: o *user service*, que gere a conta do utilizador; o *order service*, que gere os pedidos; o *recommendation service*, para as sugestões; e o *product service*, para cada um dos produtos.

Isso permite escalar vertical e horizontalmente os serviços de acordo com os pedidos de mercado, garantindo um desempenho consistente.

3. Reutilização de Código

Com a separação clara das camadas MVC, é mais fácil reutilizar componentes de código. Por exemplo, as regras de negócios (*Model*) da conta do utilizador podem ser reutilizadas em diferentes partes da aplicação, como a aplicação móvel e o *site* da loja.

4. Facilidade de Testes

O MVC facilita a escrita de testes unitários e de integração, uma vez que cada camada pode ser testada independentemente. Isso é crucial para garantir a qualidade do *software*, especialmente numa aplicação de comércio eletrônico onde a precisão e segurança são essenciais.

5. Manutenção Simplificada

À medida que novos recursos são adicionados ou atualizações são feitas, a separação clara das camadas MVC torna mais fácil identificar, isolar e corrigir problemas. Além disso, a manutenção de cada microserviço pode ser realizada sem afetar outros componentes da aplicação.

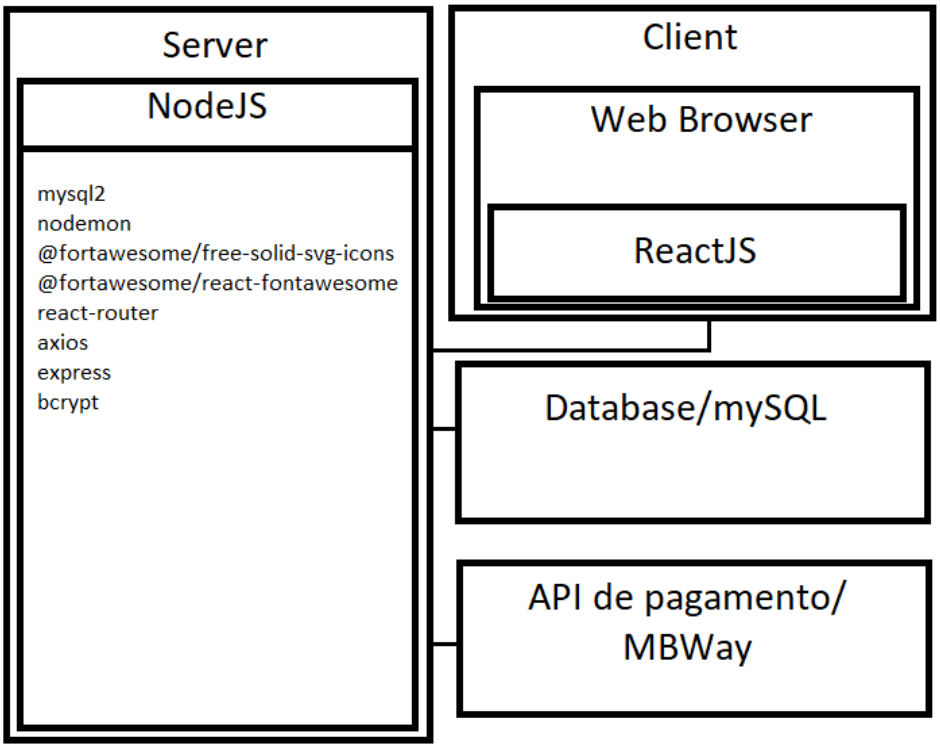
6. Evolução e Expansão

O uso de microserviços torna a arquitetura flexível o suficiente para adicionar novos recursos e serviços sem causar grandes perturbações na aplicação existente. Isso permite uma adaptação mais rápida às mudanças nas necessidades do negócio.

7. Conformidade com Padrões da Indústria

A combinação de MVC e microserviços está alinhada com práticas recomendadas e padrões da indústria, o que facilita a colaboração com *developers* e equipas externas, bem como a integração de tecnologias de terceiros.

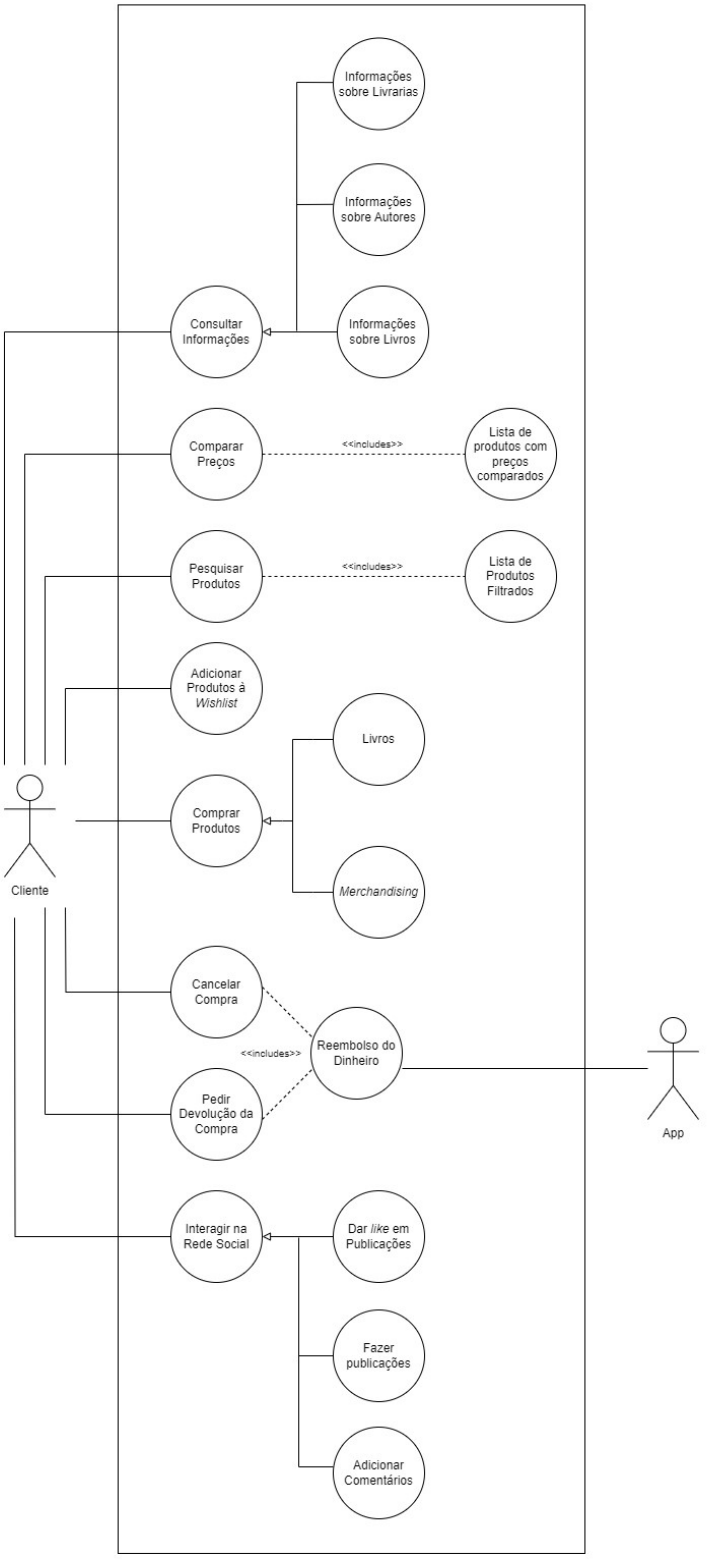
Em resumo, a escolha da arquitetura MVC com microserviços para a nossa aplicação móvel oferece uma estrutura sólida para o desenvolvimento, manutenção e escalabilidade da aplicação. Permite uma gerência mais eficiente de recursos e uma experiência do usuário mais satisfatória, enquanto prepara a aplicação para se adaptar a mudanças e crescimento no futuro.



5. DESENHO DO SOFTWARE

5.1. Desenho Global - Use Case

Delineamos um conjunto de funcionalidades principais das quais os nossos utilizadores podem tirar proveito, assim como os seus intervenientes, e desenhamos o nosso Diagrama de Use Case.

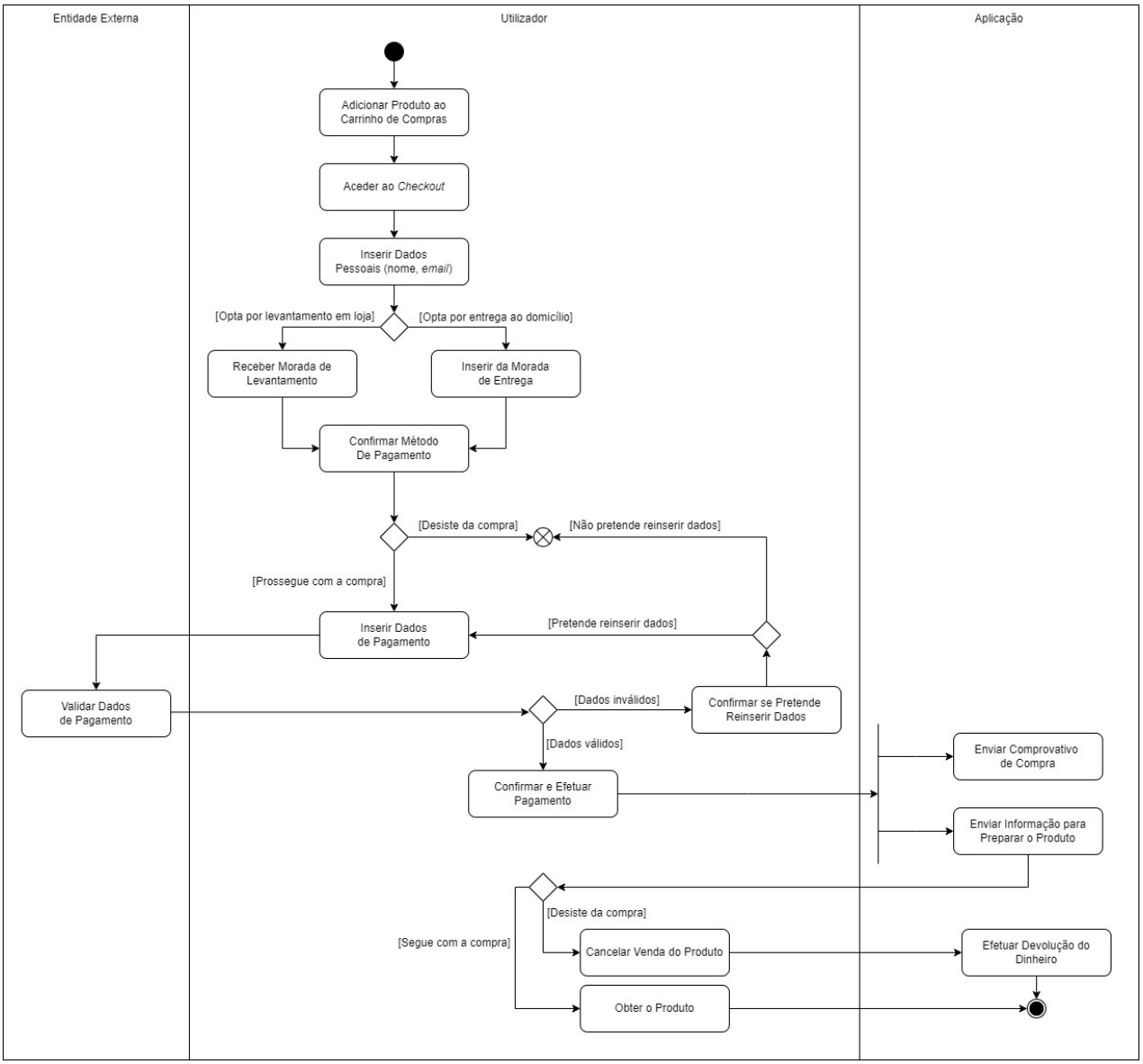


5.2. Fluxos do Sistema - Diagramas de Atividades

Para os nossos Diagramas de Atividades optamos por selecionar 4 fluxos, ou atividades, importantes para o bom aproveitamento da nossa aplicação, e depois dividimos os mesmos em vários passos, tantos quantos os necessários para percorrer a atividade desde o seu início, até à sua finalização.

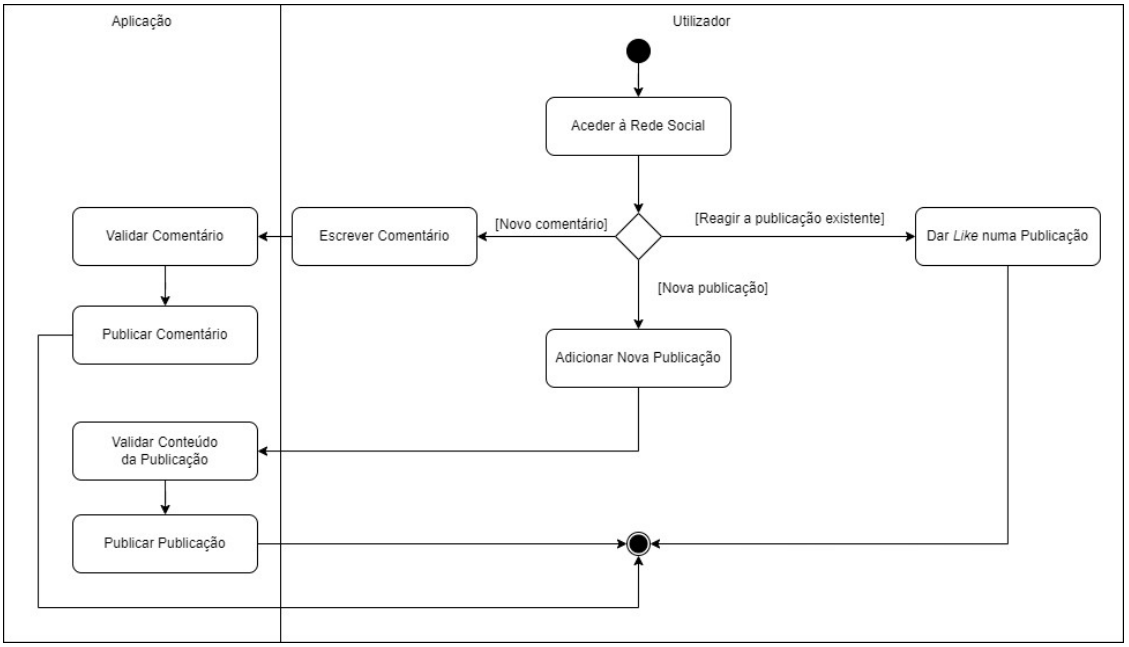
1ª Atividade

Fluxo de compra de um produto, desde o momento em que é adicionado ao carrinho de compras de um utilizador, passando por todas as fases de decisões a serem tomadas, como se prefere levantamento em loja ou entrega ao domicílio, inserção de todos os dados necessários, e até ao momento final de compra, de receção, ou cancelamento, do produto comprado.



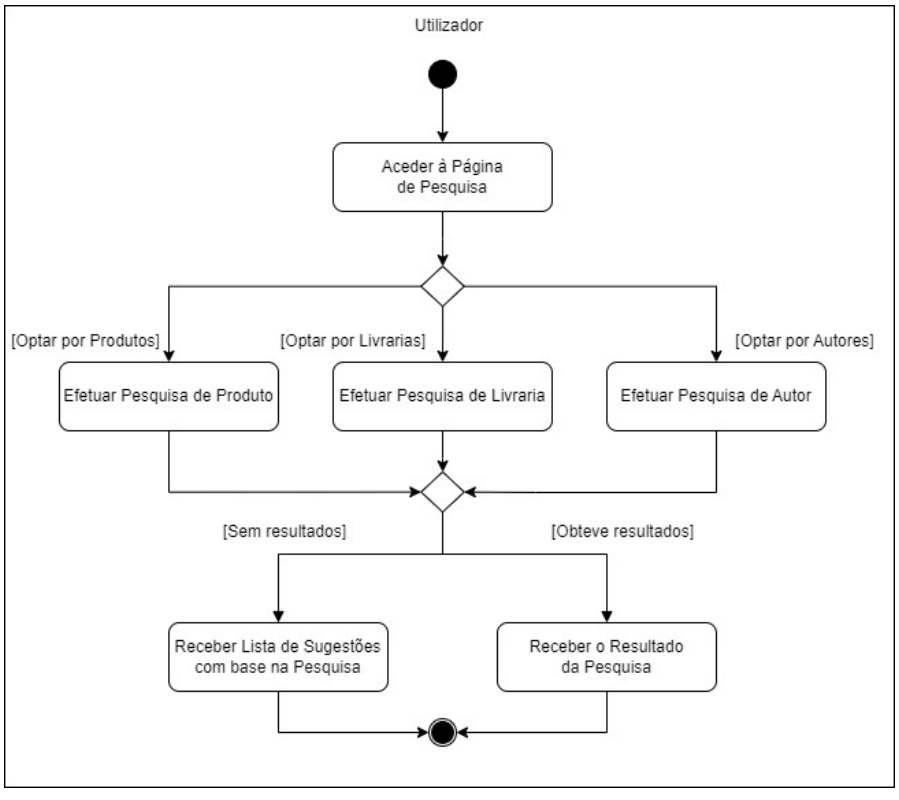
2ª Atividade

Fluxo de interação na rede social integrada na nossa aplicação. O utilizador irá aceder à rede social, e optar por uma das várias ações possíveis como: reagir a novas publicações, adicionar comentários ou até adicionar uma nova publicação.



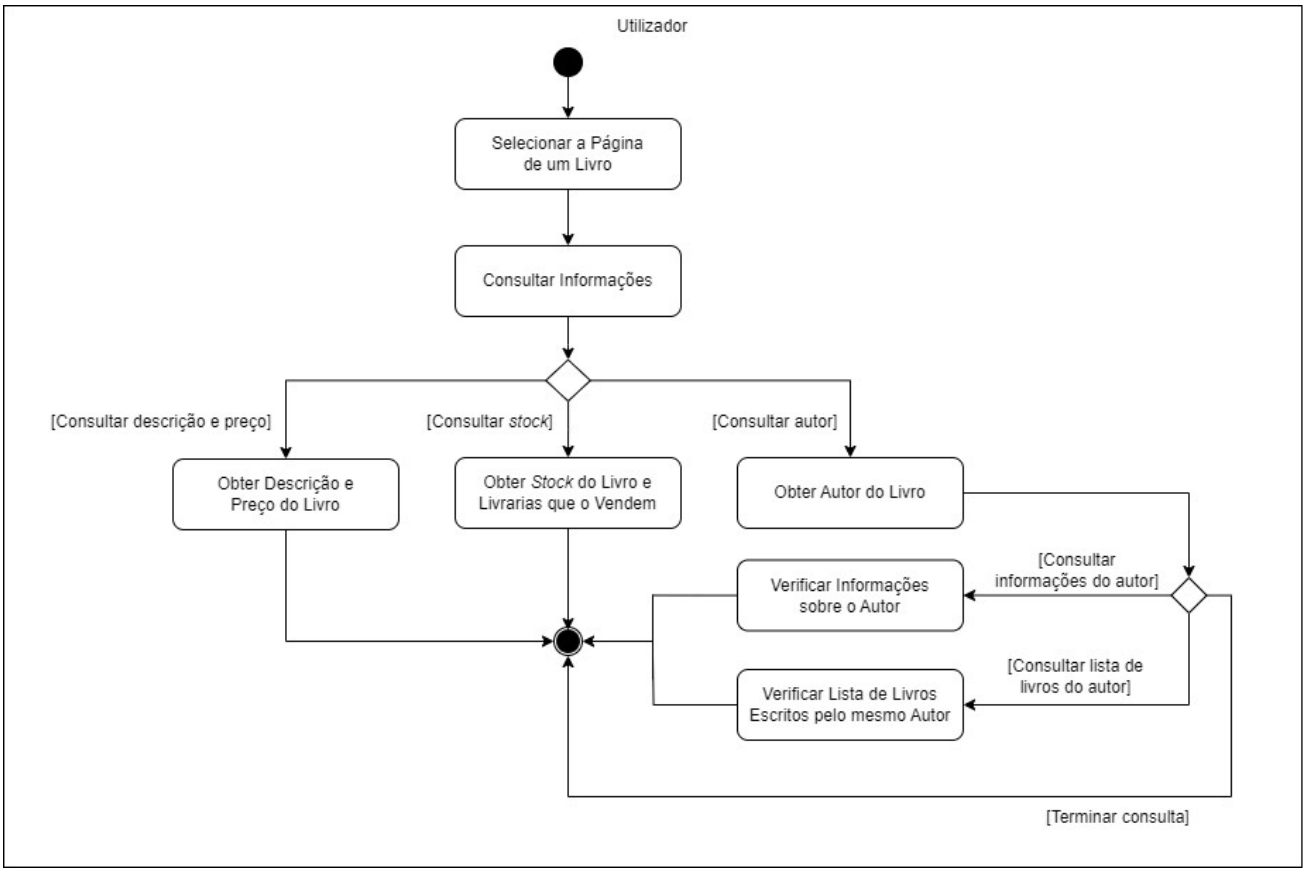
3ª Atividade

Fluxo de pesquisa, tendo como opção a pesquisa por produtos, por livrarias ou por autores. Cada uma destas pesquisas irá devolver uma lista, coincidente com o tema escolhido, e os resultados irão aparecer filtrados com base no que foi pesquisado, e, caso a pesquisa não obtenha resultados, irão aparecer sugestões baseadas na pesquisa, e nos restantes livros marcados como favoritos pelo utilizador.



4ª Atividade

Fluxo de consulta de informação, com início numa página de um Livro, e partindo para as várias opções de informação. A partir da página de informação de um livro, podemos seguir para a consulta relativamente ao seu preço, descrição, disponibilidade em *stock*, locais de venda e autor do livro.

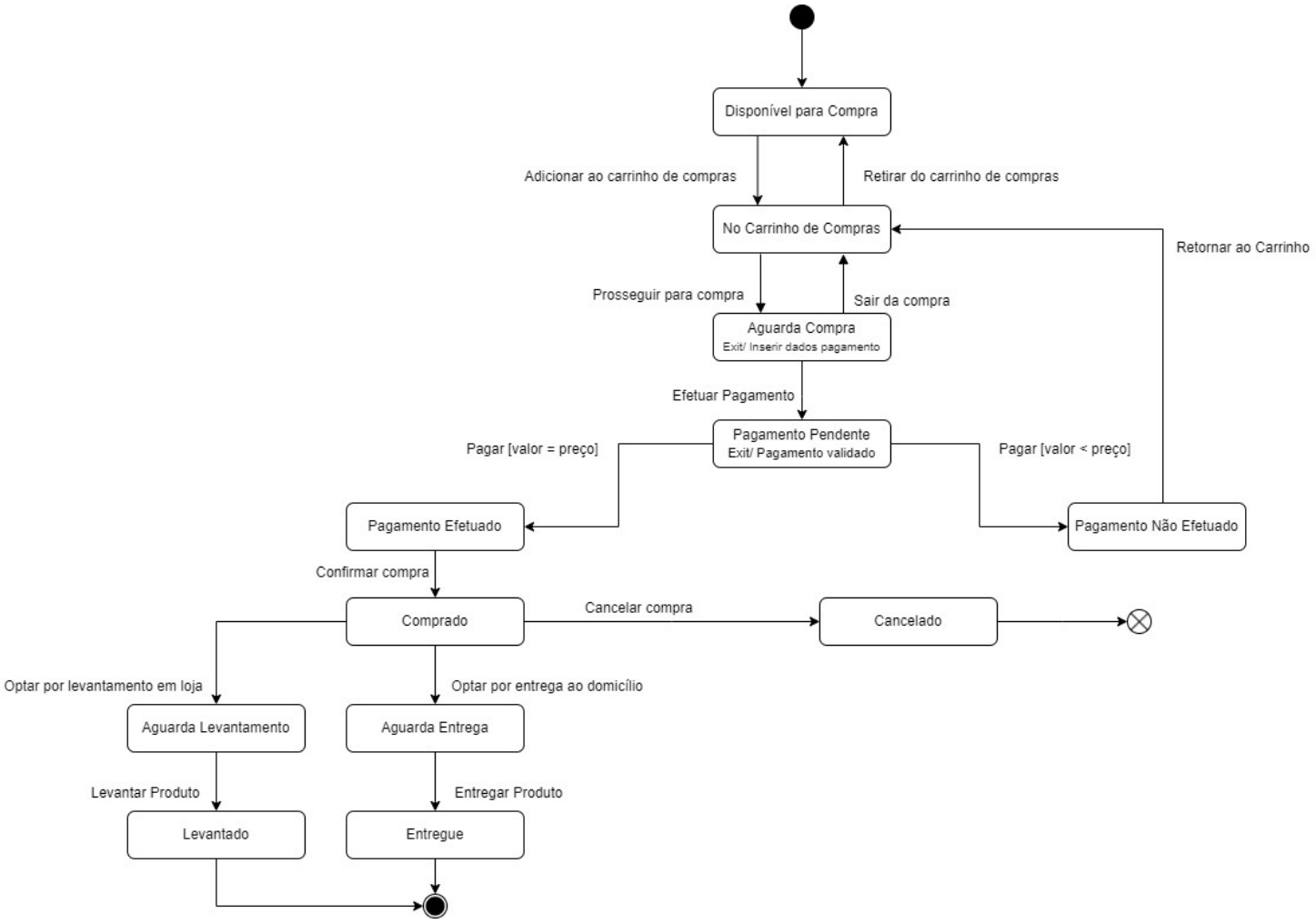


5.3. Estados - Diagramas de Estados

Para os nossos Diagramas de Estado escolhemos 4 objetos, e desenhamos os diferentes estados que eles podem tomar, dependendo das atividades ou ações de um utilizador. Delineamos as várias ações que fazem com que os estados mudem, assim como as condições e "caminhos" que o objeto percorre, enquanto se altera.

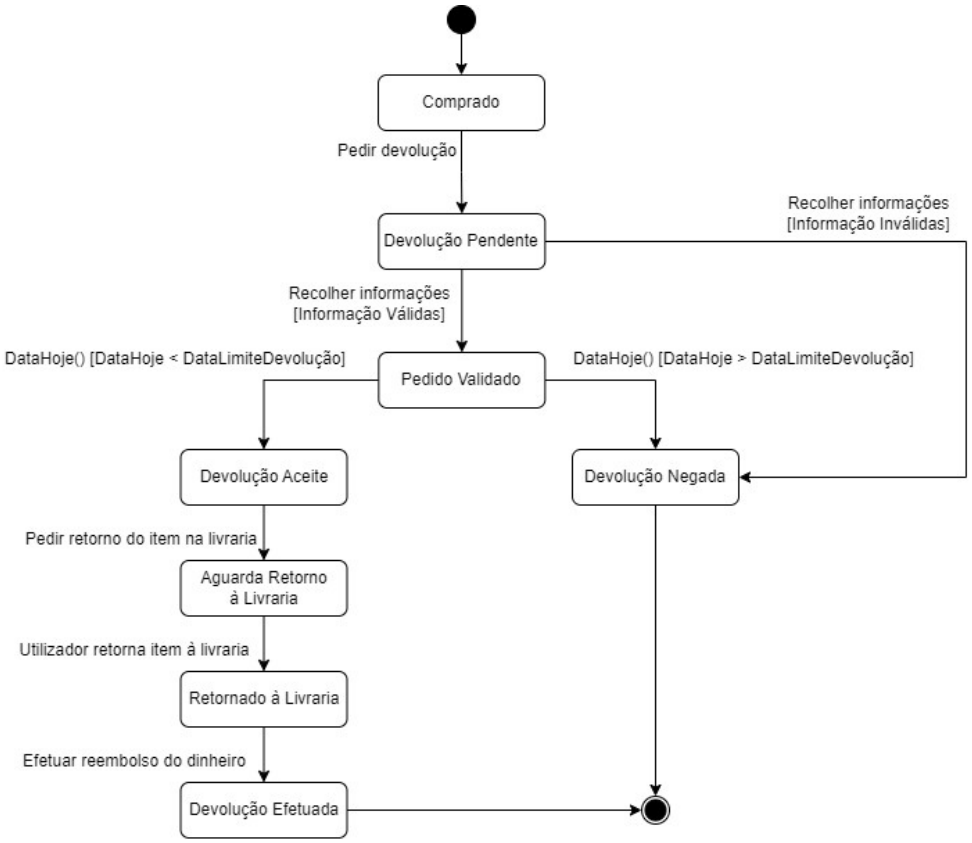
1º Estado

Estado assumido pelo objeto produto, à medida que é colocado no carrinho de compras de um utilizador, comprado, levantado ou entregue, e possivelmente cancelado ou devolvido.



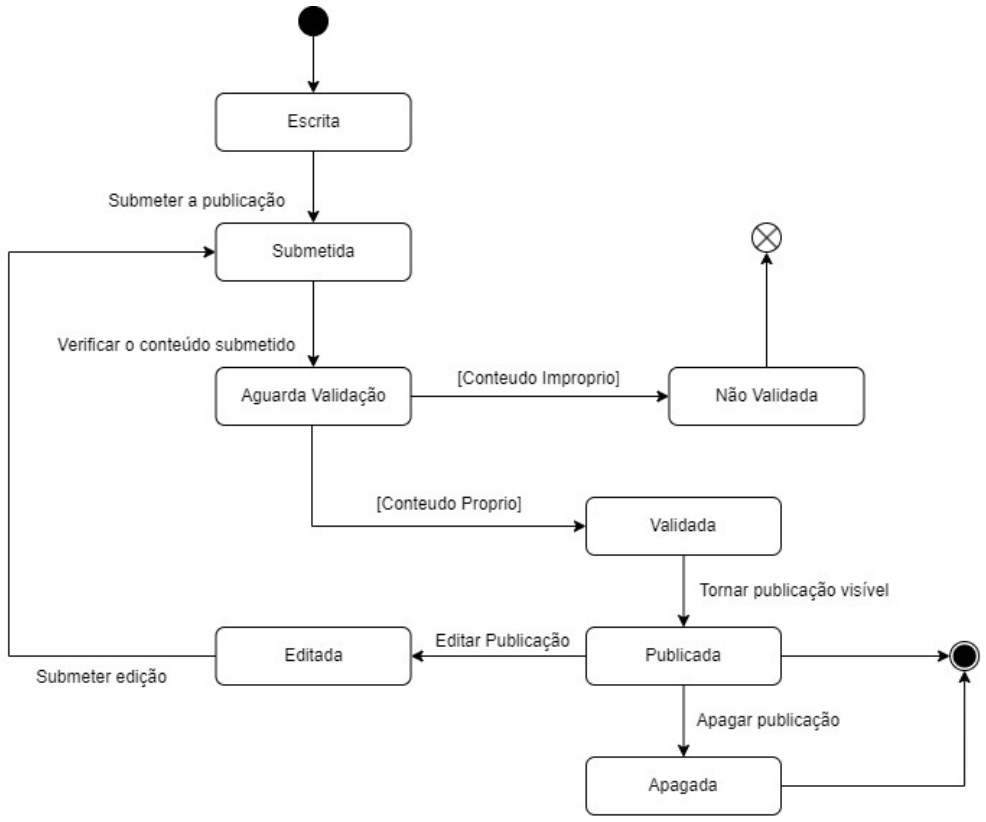
2º Estado

Estado do objeto quando, após uma compra, o utilizador optar por devolvê-lo.



3º Estado

Estado assumido pelo objeto publicação, à medida que é feita a sua submissão, até ao ponto em que está pronta para ficar visível e disponível para consulta na página do utilizador que a publicou.



6. PLANEAMENTO DA CONTINUIDADE DE NEGÓCIOS E RECUPERAÇÃO

Fase 1: Início do Plano de Continuidade de Negócio (BCP)

Objetivos do Projeto

O principal objetivo desta fase do projeto será estabelecer, para a nossa aplicação, um Plano de Continuidade de Negócio robusto, que visa garantir a disponibilidade contínua dos serviços, a integridade dos dados e a minimização dos impactos de possíveis desastres. Para a implementação do plano, será contratada uma equipa externa que ficará responsável pela elaboração do plano, sua manutenção e dará apoio técnico em caso de ataques.

Os objetivos específicos incluem:

- Identificar e avaliar os riscos que podem afetar a continuidade do negócio da aplicação.
- Desenvolver estratégias de mitigação e planos de resposta apropriados.
- Designar e treinar uma equipa de resposta a incidentes.
- Estabelecer procedimentos de *backup*, recuperação e manutenção de sistemas críticos.

Métodos para Organizar e Gerir o Desenvolvimento do BCP

Utilizaremos a abordagem do Ciclo de Vida de Continuidade de Negócio, que inclui as seguintes etapas:

- Iniciar o Projeto de BCP
- Realizar uma avaliação de riscos
- Desenvolver estratégias de mitigação
- Criar planos de resposta a incidentes
- Designar e treinar a equipa de BCP
- Implementar planos de *backup* e recuperação
- Testar os planos do BCP
- Manter e atualizar continuamente o plano

Membros da Equipa Externa de BCP, Tarefas e Responsabilidades

Para garantir a eficácia do BCP, contrataremos os seguintes membros da equipa externa BCP:

- O Coordenador de Continuidade de Negócio, que terá como responsabilidade supervisionar o projeto e tomar as decisões críticas necessárias.

- O Gerente de Projeto do BCP, que será responsável pela gestão diária e pela coordenação de atividades, recursos e relatórios de progresso.
- O Analista de Riscos, encarregado de identificar, avaliar e documentar potenciais riscos.
- A Equipa de TI, que ficará encarregue da implementação de soluções tecnológicas, planos de recuperação de sistemas e *backup* de dados.
- A Equipa de Comunicação, que deverá garantir a comunicação eficaz durante possíveis incidentes, supervisionada pelo nosso departamento de comunicação..

Calendarização das Revisões do Plano e Definição dos *Milestones* do Projeto

Algumas datas importantes a calendarizar, e que irão ajudar a manter o processo organizado, e em dia, serão as datas de:

- Início do Projeto
- Conclusão da Avaliação de Riscos
- Desenvolvimento das Estratégias de Mitigação
- Conclusão dos Planos de Resposta a Incidentes
- Treino da Equipe de BCP
- Implementação dos Planos de *Backup* e Recuperação
- Teste dos Planos de BCP
- Revisões Periódicas do Plano
- Encerramento do Projeto

Fase 2: Análise de Impacto nos Negócios (BIA)

Funções de negócio por ordem de prioridade baseado no *Maximum Tolerable Downtime* (MTD):

1. Processamento de Pagamento

- Seleção da forma de pagamento.
- Entrada de dados de pagamento.
- Validação de pagamento.
- Processamento de reembolso.

Maximum Tolerable Downtime foi definido para 2 horas.

Violações de segurança no processo de pagamentos podem resultar em perdas financeiras devido a fraudes e estornos. Os clientes podem perder a confiança na plataforma, levando a uma diminuição nas vendas e nas receitas.

2. Gerenciamento de conta de usuário

- Registo ou *login* do usuário.
- Gerenciamento de perfil.
- Gerenciamento de lista de desejos.

Maximum Tolerable Downtime foi definido para 4 horas.

Em caso de violação de segurança, as contas dos usuários e as informações pessoais podem ser comprometidas, levando à perda de confiança e credibilidade. Isso pode resultar num declínio no envolvimento do usuário e na diminuição da receita devido ao abandono da plataforma pelos clientes.

3. Pesquisa e Descoberta de Produtos, Livrarias ou Autores

- Acesso à página de pesquisa.
- Realização de pesquisas por produtos, livrarias ou autores.
- Envio de resultados da pesquisa ou sugestões.

Maximum Tolerable Downtime foi definido para 4 horas.

Se os recursos de pesquisa e descoberta não fornecerem resultados precisos ou sejam propensos a manipulação, os usuários poderão ter uma experiência ruim. Isso pode levar à redução do envolvimento do usuário, taxas de conversão mais baixas e diminuição da receita.

4. Gestão de Produtos

- Adicionar produtos à cesta.
- Finalização da compra do produto.
- Comparação de preços e produtos.
- Cancelamento de pedidos e devoluções.

Maximum Tolerable Downtime foi definido para 4 horas.

Falhas ou ataques nesta área podem resultar em cancelamentos de pedidos, atrasos nas entregas ou até mesmo compras não autorizadas. Esses problemas podem levar à insatisfação do cliente, reembolsos e perda de vendas, afetando negativamente a receita.

5. Interação nas redes sociais

- Reagir a publicações.
- Criação e moderação (através de detecção de palavras inadequadas) de novas publicações.
- Criação e moderação de comentários.

Maximum Tolerable Downtime foi definido para 8 horas.

Falhas de segurança ou ataques nesta área podem levar a acesso não autorizado, interações falsas e *spam*. Isso pode prejudicar a reputação da plataforma e desencorajar o envolvimento dos usuários, impactando potencialmente as receitas de publicidade e a retenção de usuários.

Fases 3 e 4: Estratégia de Recuperação e Planeamento do desenho e desenvolvimento

1. Processamento de Pagamento

Usaremos vários serviços de pagamento, em vez de apenas um, para evitar transtornos caso algum deles tenha algum problema. O serviço que se encontrar em baixo aparecerá temporariamente indisponível, e a utilização de um diferente será incentivada. Desta forma, garantimos que os utilizadores podem sempre fazer compras na nossa aplicação, sem interrupções, mesmo se um dos serviços apresentar problemas técnicos, diminuindo assim as chances de indisponibilidade.

2. Gerenciamento de conta de usuário

Em caso de falha, a estratégia de recuperação envolve a restauração das contas dos usuários, através de um *backup* da base de dados, e a restrição temporária de acesso a certas funcionalidades, enquanto o sistema é recuperado.

3. Pesquisa e descoberta

Em situações de falha, a estratégia de recuperação consiste em restaurar dados de produtos a partir de um *backup* da base de dados, e fazer uso temporário de dados em *cache* como solução alternativa.

4. Gestão de Produtos

Quando é necessário realizar uma recuperação do sistema a partir do *backup*, tomamos medidas imediatas para restringir temporariamente as funcionalidades de compra e pesquisa. Além disso, informaremos os utilizadores sobre a manutenção da aplicação na página principal, garantindo transparência e minimizando qualquer inconveniente durante o processo de recuperação.

5. Interação nas redes sociais

Durante o processo de recuperação, as funcionalidades de interação serão gradualmente restauradas à medida que cada uma for recuperada. Compreensivelmente, essa abordagem permite que os usuários retomem as interações à medida que o sistema for recuperando, minimizando a interrupção de serviços essenciais.

Fase 5: Implementação

Implementação de *backup* frequente e estruturação da aplicação de forma a permitir fácil manutenção de partes individuais sem que as demais partes tenham de ser alteradas, além das demais medidas de segurança especificadas pelos requisitos. Para além disso, deve ser aplicado o Plano de Resposta a Ciber-Incidentes, com base nos critérios e medidas de recuperação descritas neste BCP.

Fase 6: Teste

Realização de teste paralelo em sítio alternativo para verificar o nível de qualidade do BCP e de segurança da aplicação. Posterior avaliação a ser feita com base nos resultados do teste para realizar adaptações necessárias a este plano.

Fase 7: Manutenção, disseminação e treino

A atualização do BCP e treino relacionado ocorrerá em três situações:

- Revisão anual.
- Revisão imediata após um ataque, ou *exploit.*
- Revisão não emergencial caso surja uma nova forma de ataque que possa afetar os negócios.

7. CONCLUSÃO

Em conclusão o projeto de arquitetura de software para a nossa aplicação teve como objectivo criar uma plataforma segura, escalável e eficiente para atender às necessidades do negócio. Ao longo deste processo, abordamos quatro elementos fundamentais: requisitos de segurança, proposta de arquitetura do sistema com o modelo Model-View-Controller (MVC) com a utilização de microsserviços, desenho de software com diagramas de Use Cases, diagramas de atividades e diagramas de estado, para além de um plano de continuidade de negócios e recuperação.

Em relação aos requisitos de segurança, identificamos e implementamos medidas robustas para proteger os dados dos nossos clientes e garantir a integridade e confidencialidade das informações. Isso incluiu a criptografia de dados sensíveis, autenticação de utilizadores, controlo de acesso rigoroso e monitorização constante para detectar possíveis ameaças à segurança.

A proposta de arquitetura do sistema com o modelo Model-View-Controller (MVC) e a utilização de microsserviços proporcionou uma base sólida para o desenvolvimento da nossa aplicação. Essa arquitetura modular nos permitirá escalar componentes individualmente, o que é crucial para atender ao crescimento do negócio e às mudanças de procura dos utilizadores. Além disso, a separação de responsabilidades proporcionada pelo MVC ajudará a manter o código limpo e manutenível.

Os diagramas Use Case, atividades e estados fornecem uma representação visual clara do comportamento da aplicação, dos processos de negócios e das interações do utilizador. Isso é essencial para a comunicação eficaz com as partes interessadas e a equipa de desenvolvimento, facilitando a compreensão e a validação dos requisitos do sistema.

Por fim, o plano de continuidade de negócios e recuperação assegura que estamos preparados para enfrentar interrupções inesperadas. Isso inclui a implementação de políticas de backup, procedimentos de recuperação de desastres e a disponibilidade de ambientes de contingência. Garantir a continuidade do nosso negócio é uma responsabilidade crítica, e este plano permite-nos minimizar o impacto de eventos imprevistos.