## Gourmet Management

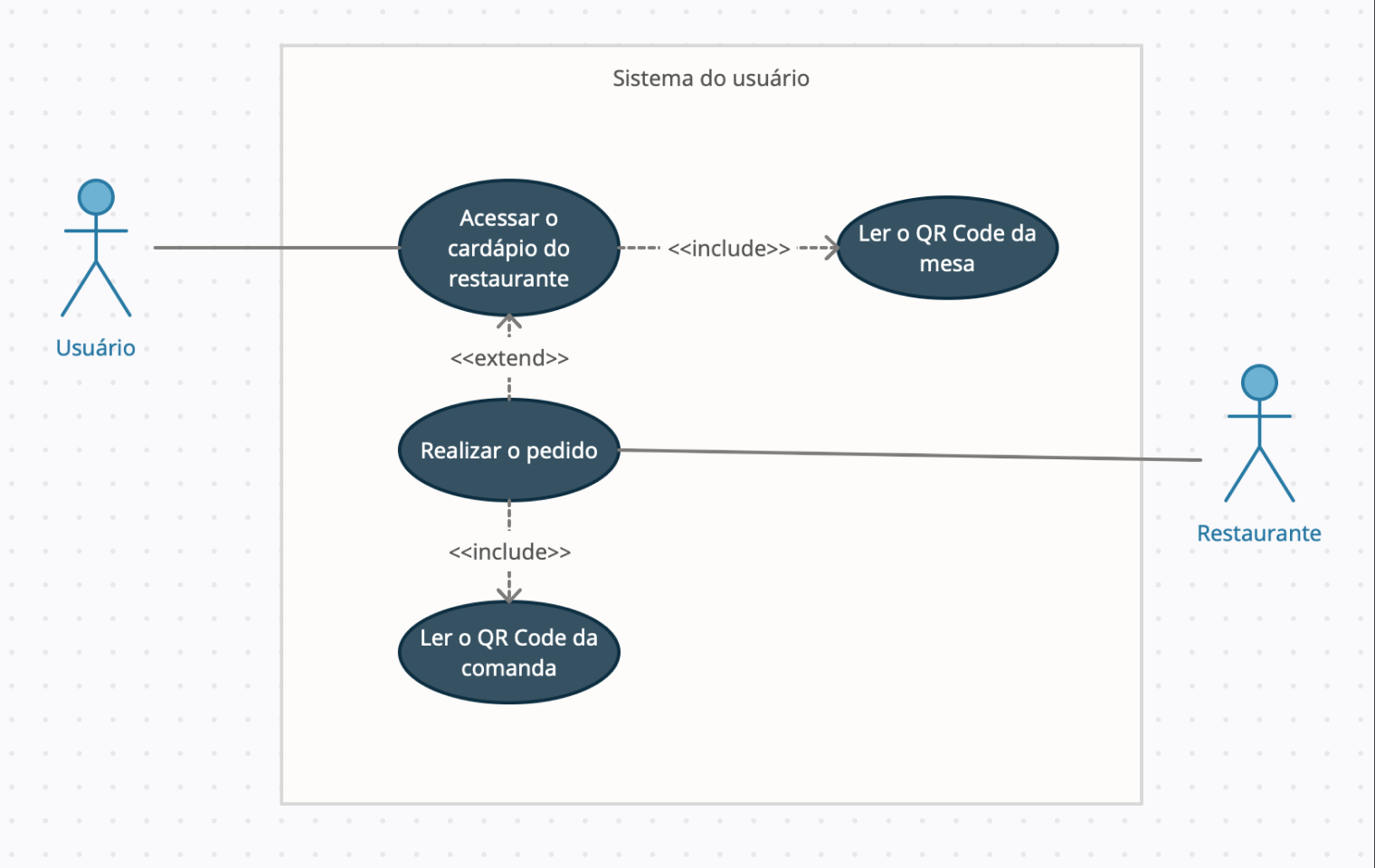
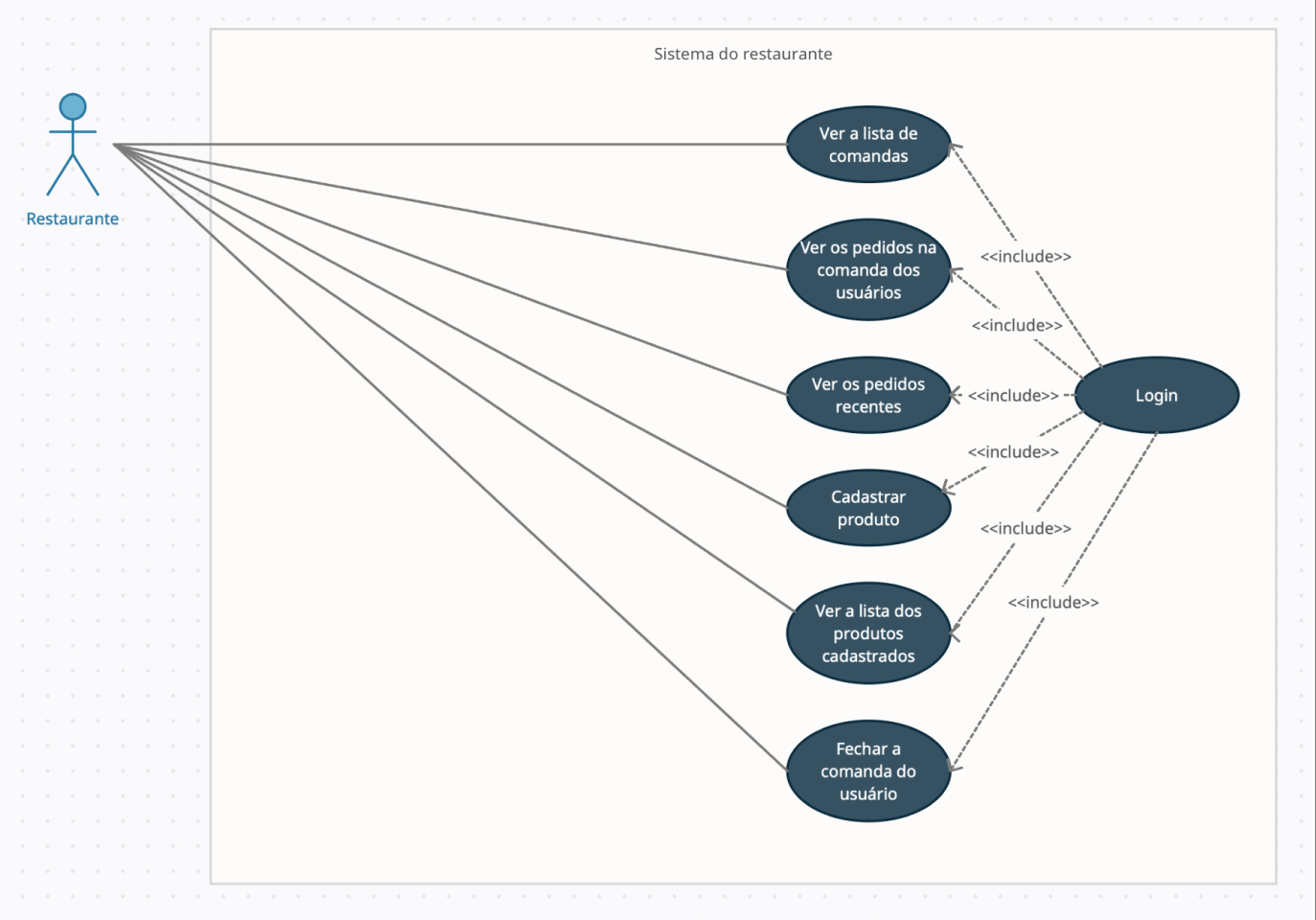
Bernardo Gatto Hansen ([bernardo.gatto@acad.ufsm.br](mailto:bernardo.gatto@acad.ufsm.br)), Darlan Guimarães ([darlan.guimaraes@acad.ufsm.br](mailto:darlan.guimaraes@acad.ufsm.br)), Guilherme Ulbriki ([guilherme.ulbriki@acad.ufsm.br](mailto:guilherme.ulbriki@acad.ufsm.br)), Jacson Ballin ([jacson.balllin@acad.ufsm.br](mailto:jacson.balllin@acad.ufsm.br)), Jean Giordani([jean.giordani@acad.ufsm.br](mailto:jean.giordani@acad.ufsm.br) ), Patricia Lengert ([patricia.lengert@acad.ufsm.br](mailto:patricia.lengert@acad.ufsm.br))

Data da última atualização 11/04/2023

## Introdução

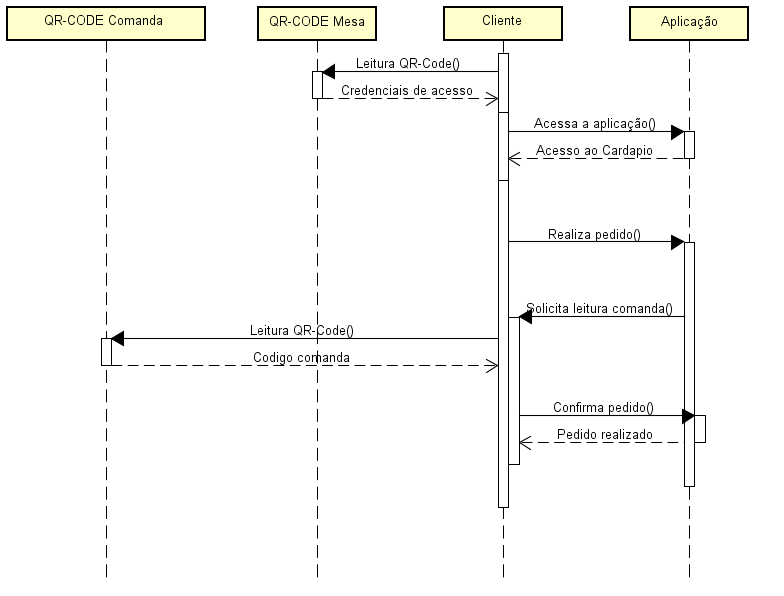
* Esta etapa do projeto tem por finalidade modelar e definir os requisitos que serão necessários para implementar e aplicar o sistema, permitindo que os *stakeholders* possam entender melhor os objetivos e funcionalidades do sistema.
* O sistema tem como objetivo automatizar os processos e fluxos de pontos alimentícios no âmbito da sociedade, tornando o atendimento e tarefas mais rápido e eficiente para os provedores do serviço tanto para os consumidores. O sistema seria para qualquer tipo de local de venda de alimentos tanto quanto restaurantes, lanchonetes, bares e sorveterias.
* Ao utilizar esse sistema, os clientes podem visualizar o catálogo de produtos disponíveis em um dispositivo eletrônico, como um smartphone, e realizar seus pedidos diretamente, sem a necessidade de chamar um garçom ou atendente, após o pedido ser feito no dispositivo eletrônico, ele é enviado automaticamente para a cozinha ou área de preparação, notificando o chefe de cozinha sobre o pedido do cliente. Isso reduz o tempo de espera e minimiza possíveis falhas de comunicação entre o cliente e o estabelecimento.

## 2. Requisitos Funcionais: Casos de Uso

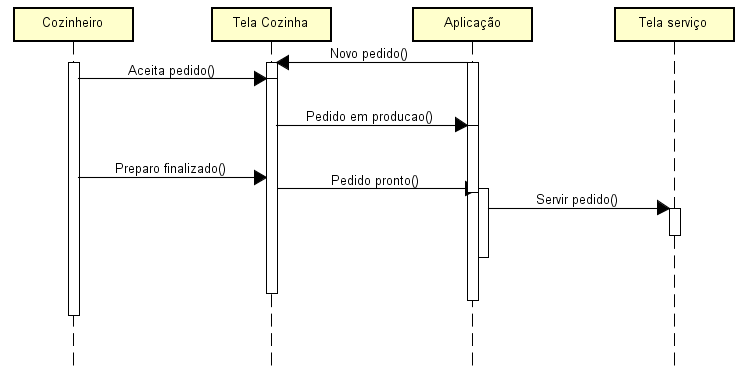
1. Cálculos de processamento: O sistema deve ser capaz de calcular a demora de seus processos.
2. Relatórios mensais sobre vendas: O sistema deve ser capaz de gerar relatórios mensais sobre as vendas dos produtos do estabelecimento.
3. Relatórios mensais sobre desempenho financeiro:: O sistema deve ser capaz de gerar relatórios mensais sobre as finanças do estabelecimento.
4. Controle de estoque: O sistema deve permitir o gerenciamento do estoque de produtos, incluindo alertas de estoque baixo e recomendações de reabastecimento.
5. Cálculo de composição da comanda: Pelo sistema deve ser possível calcular o valor total da comanda com base nos itens selecionados pelo cliente.
6. Integração com outros sistemas: O sistema deve ser capaz de se integrar com outros sistemas e aplicativos, como sistemas de pagamento, sistemas de gerenciamento de estoque e outros.
7. Personalização de pedidos: O sistema deve permitir que os usuários personalizem seus pedidos de acordo com suas preferências e necessidades individuais.
8. Notificação para cozinha: O sistema deve ser capaz de enviar notificações para a cozinha sobre os pedidos feitos pelos clientes, para garantir um serviço rápido e eficiente.
9. Pedidos em grupo - mesa: O sistema deve permitir que os clientes façam pedidos em grupo para compartilhar a mesma mesa, incluindo a divisão dos custos entre os participantes.
10. Dividir valor de um produto em diversas comandas: Pelo sistema deve se poder dividir o valor de um produto ou serviço entre várias comandas, quando necessário.
11. Suporte a diferentes idiomas: O sistema deve suportar diferentes idiomas para atender a uma base de clientes diversa.
12. Gerenciamento de descontos e promoções: Pelo sistema deve ser possível gerenciar descontos e promoções para atrair e reter clientes.
13. Integração com APIs: O sistema deve ser capaz de se integrar com APIs externas para fornecer recursos adicionais para os usuários.
14. Preços e descrições precisos: O sistema deve exibir preços e descrições precisas dos produtos e serviços disponíveis.
15. Notificação em tempo real: O sistema deve enviar notificações em tempo real sobre o status do pedido e o processamento do pagamento.
16. Adição de comentários e avaliações dos produtos: O sistema deve permitir que os usuários adicionem comentários e avaliações sobre os produtos e serviços, para ajudar outros clientes a tomar decisões informadas.
17. Pagamento *Online*: O sistema deve possibilitar que os clientes façam pagamentos online de forma segura e conveniente,
18. Reserva de mesas: Pelo sistema deve ser possível que os clientes façam reservas de mesas para garantir um serviço rápido e eficiente.
19. Comunicação com cliente em tempo real: O sistema deve possibilitar que os usuários se comuniquem com o cliente em tempo real para esclarecer dúvidas e resolver problemas.
20. Relatório e análises de vendas: O sistema deve fornecer relatórios referente as vendas e atividades provenientes do restaurante.
21. Sistema de recomendações com base nas avaliações: O deve reconhecer padrões de pedidos e recomendar com base nos pedidos atuais do usuário.
22. Notificações sobre os pedidos dos clientes: O sistema deve notificar o andamento do pedido do usuário.
23. Geração do *QRCode* para cada mesa: O sistema deve permitir que o administrador gere os *QRCodes* únicos para cada mesa.
24. Criação das comandas *QRCode*: O sistema deve possibilitar que o administrador gere as comandas com os *QRCodes*.
25. Gerenciamento dos pedidos pelo restaurante: O sistema deve ter uma tela de gerenciamento de pedidos, que permita ao restaurante organizar os pedidos recebidos.
26. Possibilidade do cliente realizar seu pedido: O sistema acessado pelo dispositivo do cliente deve permitir que ele realize um pedido e o agregue a sua comanda.
27. Pagamento eletrônico: Pelo sistema deve ser possível visualizar bem como fechar a sua comanda para o pagamento.
28. Gerenciamento do catálogo: O sistema deve permitir que o restaurante realize o gerenciamento do seu catálogo, podendo, criar, editar e excluir produtos.
29. *Login* funcionário: O sistema deve possuir um *Login* para os funcionários.
30. Cadastro funcionário: pelo sistema deve ser possível realizar o cadastro de funcionários.
31. Controle das mesas: Pelo sistema deve ser possível realizar o controle das mesas
32. Controle das comandas: O sistema deve permitir o controle das comandas
33. Gerenciamento de Funcionários: O sistema deverá cadastrar, editar e excluir seus funcionários.
34. Casos de uso:

## 2.1 Diagramas de sequência

Cliente efetua um pedido



Produção do pedido pela cozinha



## 3. Requisitos Não Funcionais

1. Desempenho: Especificar os requisitos de desempenho, como tempo de resposta e taxa de transferência.
2. O Sistema deve ser capaz de lidar com falhas inesperadas: Este requisito se relaciona com a capacidade do sistema de lidar com situações de falhas inesperadas, tais como falhas de energia, falhas de *hardware* ou outros problemas técnicos que possam interromper o funcionamento normal do sistema.
3. Escalabilidade: este requisito se relaciona com a capacidade do sistema de lidar com um aumento no número de usuários ou no volume de dados sem comprometer o desempenho do sistema. É importante que o sistema seja projetado de forma a ser escalável e a permitir que novos recursos sejam adicionados conforme necessário.
4. Tempo de resposta curto durante a navegação no sistema: este requisito se relaciona com o tempo que o sistema leva para responder às solicitações do usuário, tais como a abertura de uma nova página ou a execução de uma ação. É importante que o tempo de resposta seja rápido o suficiente para garantir a usabilidade do sistema e evitar frustrações dos usuários.
5. Agilidade na leitura e geração dos *QRCodes*: este requisito se relaciona com a capacidade do sistema de ler e gerar *QRCodes* rapidamente, sem atrasos significativos que possam afetar a experiência do usuário. É importante que o sistema seja capaz de lidar com *QRCodes* em tempo real e com alta eficiência.
6. Deve ser capaz de lidar com um grande número de requisições: este requisito se relaciona com a capacidade do sistema de lidar com um grande número de requisições simultâneas sem comprometer o desempenho do sistema. É importante que o sistema seja projetado de forma a ser capaz de lidar com picos de tráfego e de requisições sem apresentar lentidão ou falhas.
7. Usabilidade: Especificar os requisitos de usabilidade, como facilidade de uso e acessibilidade.
8. Página de menu: O sistema deve fornecer uma página de menu fácil de usar, onde os usuários possam encontrar rapidamente os produtos e serviços disponíveis e fazer seus pedidos.
9. Suporte a diferentes tipos de dispositivos: este requisito se relaciona com a capacidade do sistema de ser acessível a partir de diferentes tipos de dispositivos, tais como *smartphones*, *tablets*, computadores, entre outros.
10. O sistema deve permitir que os usuários realizem suas tarefas de forma rápida e eficiente: este requisito se relaciona com a capacidade do sistema de permitir que os usuários realizem suas tarefas de forma rápida e eficiente. É importante que o sistema seja projetado de forma a minimizar o tempo necessário para realizar as tarefas, para que os usuários possam realizar suas atividades com rapidez e eficiência.
11. Acessibilidade para todos os tipos de usuários: este requisito se relaciona com a capacidade do sistema de ser acessível a todos os tipos de usuários, independentemente de suas necessidades especiais. É importante que o sistema seja projetado de forma a atender aos padrões de acessibilidade, para que todos os usuários possam utilizar o sistema sem dificuldades.
12. *Software* intuitivo e com facilidade de uso: este requisito se relaciona com a facilidade de uso do software. É importante que o sistema seja projetado de forma a ser intuitivo e fácil de usar, para que os usuários possam realizar suas tarefas de forma rápida e sem dificuldades.
13. O estabelecimento deve disponibilizar acesso a internet *wi-fi* aos clientes: este requisito se relaciona com a disponibilidade de uma conexão de *internet wi-fi* no estabelecimento. É importante que o estabelecimento disponibilize uma conexão de *internet wi-fi* confiável e rápida, para que os usuários possam acessar o sistema sem problemas de conexão. Isso é especialmente importante para sistemas de comanda eletrônica que dependem de uma conexão de *internet* para funcionar adequadamente.
14. Segurança: Especificar os requisitos de segurança, como autenticação e autorização.
15. Criptografia de dados: este requisito se relaciona com a capacidade do sistema de proteger as informações dos usuários contra invasores e outras ameaças. É importante que o sistema utilize algoritmos de criptografia fortes para garantir que as informações dos usuários sejam protegidas durante o armazenamento e a transmissão.
16. *Backup* e recuperação: este requisito se relaciona com a capacidade do sistema de realizar *backups* regulares dos dados do usuário e permitir sua recuperação em caso de falhas ou perda de dados. É importante que o sistema possa recuperar os dados dos usuários rapidamente e sem perda de informações.
17. Possibilidade de Correções e Atualizações: este requisito se relaciona com a capacidade do sistema de receber correções e atualizações para garantir que ele continue funcionando adequadamente e mantendo a segurança dos dados dos usuários.
18. Políticas de privacidade (LGPD): este requisito se relaciona com a necessidade do sistema de seguir as leis e regulamentações de privacidade de dados do país em que está operando. É importante que o sistema siga as políticas de privacidade, como a Lei Geral de Proteção de Dados (LGPD) no Brasil, para garantir que as informações dos usuários sejam tratadas de forma adequada e segura.
19. Proteção física para o *Hardware*: este requisito se relaciona com a capacidade do sistema de proteger fisicamente o *hardware* em que está instalado, por exemplo, por meio de senhas e bloqueios de segurança física. Isso ajuda a proteger o sistema contra danos ou violações físicas.
20. Correção de falhas de segurança: este requisito se relaciona com a capacidade do sistema de detectar e corrigir falhas de segurança. É importante que o sistema tenha uma equipe dedicada à detecção e correção de falhas de segurança para garantir que as informações dos usuários estejam protegidas contra ataques.
21. Confiabilidade nas transações de pagamento: Este requisito se relaciona com a capacidade do sistema de garantir a segurança e a integridade das transações de pagamento. É importante que o sistema utilize protocolos de segurança fortes, como SSL/TLS e criptografia de ponta a ponta, para garantir que as transações sejam seguras e confiáveis.
22. Manutenibilidade: Especificar os requisitos de manutenção, como facilidade de modificação e documentação.
23. Ampla documentação: O sistema deverá ter uma documentação completa.
24. Código limpo e boas práticas e princípios que visam tornar o código fonte mais legível: O sistema deverá seguir as boas práticas e ter um código limpo para manutenção e adição de novas funcionalidades.

## 4. Glossário

* API (Interface de Programação de Aplicativos): D. Woods e D. Jacobson (2011) deixam a entender como sendo um conjunto de regras e protocolos que permite a comunicação entre diferentes sistemas de software. As APIs são usadas para que diferentes aplicativos possam interagir uns com os outros, compartilhar dados e funcionar em conjunto.
* *QRCode* (Código de Resposta Rápida): É um código de barras bidimensional que pode ser escaneado usando um aplicativo de leitura de *QRCode* em dispositivos móveis, como smartphones e tablets. Ele armazena informações, como URLs de websites, informações de contato, textos, entre outros, e é amplamente utilizado para fornecer acesso rápido a informações digitais.
* *Login*: É o processo de autenticação em um sistema de computador ou aplicativo, geralmente por meio de uma combinação de nome de usuário e senha. O *login* permite que os usuários acessem suas contas pessoais e restritas em sistemas *online*, como redes sociais, serviços de *e-mail*, aplicativos de banco, entre outros.
* *Software*: É um conjunto de programas, dados e instruções que controlam o funcionamento de um computador ou dispositivo eletrônico. O software pode ser dividido em duas categorias principais: *software* de sistema, que é responsável pelo funcionamento básico do computador, e software de aplicativo, que é projetado para executar tarefas específicas, como processamento de texto, edição de imagens, navegação na *web*, entre outros define Ruhrmair, U., Devadas, S. e Koushanfar, F (2012).
* Criptografia: KOOPS, Bert-Jaap (1999) coloca como um tipo de técnica de codificar informações para proteger sua confidencialidade e integridade. A criptografia é amplamente utilizada para proteger dados sensíveis, como informações financeiras, senhas, dados pessoais e comunicações *online*. Existem vários algoritmos e métodos de criptografia, e eles são usados para garantir que apenas pessoas autorizadas possam acessar as informações protegidas.
* *Backup*: R. Budi (2011) explica e coloca a entender como uma cópia de segurança de dados importantes armazenados em um dispositivo eletrônico, como um computador, servidor, dispositivo móvel, entre outros. O *backup* é feito para garantir que os dados sejam protegidos contra perda acidental, falhas de *hardware*, ataques de *hackers* ou outros eventos imprevistos. Os *backups* podem ser armazenados em locais diferentes do original, como em nuvem ou em dispositivos de armazenamento externo.
* LGPD (Lei Geral de Proteção de Dados): A. Joice (2018) define como um tipo de uma legislação brasileira que estabelece regras e diretrizes para a proteção de dados pessoais de cidadãos brasileiros. A LGPD tem como objetivo garantir a privacidade e a segurança dos dados pessoais dos indivíduos, regulando a coleta, armazenamento, uso e compartilhamento de informações pessoais por parte de empresas e organizações.
* *Hardware*: É a parte física de um sistema de computador ou dispositivo eletrônico. Isso inclui componentes como processadores, memória, placas-mãe, discos rígidos, telas, teclados, mouses, entre outros. O *hardware* é a base física sobre a qual o software é executado e os dados são armazenados e processados definem Ruhrmair, U., Devadas, S. e Koushanfar, F (2012).
* Código Fonte: É o conjunto de instruções de programação escritas por um programador em uma linguagem de programação específica. É a forma legível e compreensível de um programa, que pode ser modificado e compilado para criar um programa de computador executável define D. Linus (2019).

## 5. Referências

1. D. Woods, D. Jacobson, G. Brail. **APIs: A Strategy Guide: Creating Channels with Application Programming Interfaces**. O’REILLY’. 2011.
2. R. Budi, “**Belajar Otodidak Membuat Database Menggunakan MySQL.” Bandung : Informatika**, 2011.
3. AGOSTINELLI, Joice. **A importância da lei geral de proteção de dados pessoais no ambiente online. Etic-encontro de iniciação científica -ISSN 21-76-8498,** v. 14, n. 14, 2018.
4. KOOPS, Bert-Jaap. **The Crypto Controversy: A Key Conflict in the Information Society. The Haag: Kluwer Law International,** 1999.
5. DIETZ, Linus, HARRER, Johannes, LENHARD, Jörg. **Teaching Clean Code**. 2019.
6. Ruhrmair, U.; Devadas, S.; Koushanfar, F. **Security Based on Physical Unclonability and Disorder. In Introduction to Hardware Security and Trust; Tehranipoor, M., Wang,** C., Eds.; Springer: New York, NY, USA, 2012.