Carlos Maurício Borges de Melo[[1]](#footnote-2)\*

Lucélia Vieira[[2]](#footnote-3)\*\*

Miguel Francisco da Silva Rodrigues[[3]](#footnote-4)\*\*\*

Zulmira Monteiro Ximenes [[4]](#footnote-5)\*\*\*\*

**Projeto integrador II (TCC)**

**RESUMO**

Este trabalho tem por objetivo demonstrar as vantagens do uso de aplicativos móveis no dia-a-dia das pessoas, podendo tornar ações complexas em ações simples e eficientes. São diversas as maneiras como os chamados “Apps” vem influenciando os usuários nas suas atividades diárias, desde os afazeres domésticos, acadêmicos ou, até mesmo no trabalho. O presente trabalho abrangerá o ambiente dos esportes, mais voltado para o lado da prática do futebol amador. Considerando a grande utilização de campos society’s, e a forma como é realizada a reserva destes, bem o funcionamento das partidas e montagens das equipes em partidas informais. O aplicativo a ser desenvolvido irá atender o público praticante do futebol amador, onde contemplará funcionalidades que vão desde a reserva do campo até mesmo na escalação do time em uma partida de futebol.

**Palavras-Chave:** [Aplicativos Móveis, Futebol Amador, Reservas]

**ABSTRACT**

This work was intended to show how the use of mobile applications in people's daily life can make everything simpler. There are a number of ways in which so-called Apps have been influencing users in their daily activities, from home chores, to academic tasks, or even at work. This work will cover the sports environment, more towards the practice side of amateur football. Considering the great use of society's fields, and the way in which they are held in reserve, as well as the functioning of teams' matches and assemblies in informal matches. The application to be developed will cater to the amateur soccer practitioners, where it will include features ranging from the reservation of the field to the team's formation in a game.

**Keywords:** [Mobile Applications, Amateur football, Reservations]

# INTRODUÇÃO

A tecnologia revolucionou o modo como as pessoas vivem e principalmente como realizam suas atividades diárias. E cada vez mais, a tecnologia vêm contribuindo para ser um facilitador. Com o surgimento da internet na década de 90, onde surgiram os browsers, ou navegadores (Internet Explorer, Netscape, Mozilla Firefox...), o dinamismo de transmissão da informação se tornou algo extraordinário, conforme Castells (2003, p.13) “A história da criação e do desenvolvimento da Internet é a história de uma aventura humana extraordinária”. A utilização de aplicativos móveis trouxe para o cotidiano a facilidade de realizar uma transação bancária, chamar um serviço de táxi ou ainda pedir uma refeição dentre outras atividades, através de aparelhos modernos e elegantes e de fácil navegação. O *android* que é um sistema operacional da Google que é uma das empresas líderes neste segmento, sendo vista como uma plataforma ágil para o desenvolvimento de aplicativos pelas grandes fabricantes LG Motorola, Sony Intel

Os benefícios de se utilizar aplicativos móveis são inúmeros, entre eles destaca-se: aproveitamento de tempo, comodidade, entre outros. Além, de ser uma boa fonte para tornar tudo mais prático e também encontrar opções de diversão e lazer com facilidade utilizando a tecnologia para monitorar e ou promover maior adesão às ofertas de lazer como forma de bem-estar e convívio social.

Sobre os problemas cotidianos que podem gerar um problema ao já comentado bem-estar, pode-se destacar que pessoas que praticam ou frequentam partidas de futebol amador, a famosa “pelada”, percebem as dificuldades para organizar um jogo. É bastante comum integrantes dos times não irem ao jogo por diversos motivos. Existem casos também de times que não têm atletas o suficiente na hora do jogo, resultando, às vezes, no atraso do início da partida ou até mesmo no cancelamento do jogo.

Em se tratando dos jogos amadores, é bastante comum equipes buscarem por um campo para a realização de uma partida e não encontram de imediato. Destaca-se, também, a dificuldade de ratear o valor cobrado da reserva entre os atletas. Para a reserva do espaço, normalmente é realizado um contato por meio de ligação telefônica ou até mesmo visita ao local. O que ocasiona um desgaste para quem busca se divertir.

# REFERENCIAL TEÓRICO

## Spring Framework:

Spring Framework é um *framework* voltado para o desenvolvimento de aplicações corporativas para a plataforma Java, baseado nos conceitos de inversão de controle e injeção de dependências.

Com o aumento da produção de software os problemas são frequentes, a framework Spring possibilita solucionar inúmeros problemas comuns presentes no desenvolvimento J2EE (*Java 2 Enterprise Edition*) e Java JEE (*Java Enterprise Edition*). O objetivo dessa ferramenta é lembrar de fazer a coisa certa na programação orientada a objetos: projetar aplicações usando interfaces. [Chhatpar 2006].

## Ionic Framework

Ionic é um framework de desenvolvimento de aplicativos móveis em HTML5 voltado para o desenvolvimento híbrido. Ele utiliza o Angular, que é um framework JavaScript desenvolvido pela Google. Os aplicativos desenvolvidos com Ionic são como pequenos sites que funcionam em um uma camada embutida do navegador em um aplicativo, o qual possui acesso à camada da plataforma nativa através de bibliotecas fornecidas pelo próprio Ionic.

Ao contrário de uma estrutura responsiva qualquer, o Ionic vem com elementos e layouts de UI móveis de estilo nativo, que só seria possível obter com um SDK nativo no iOS ou no Android, mas que não seria possível nos elementos e layouts da Web. (IONIC FRAMEWORK, 2016).

Por se tratar de um framework HTML5, o Ionic precisa de um invólucro nativo como o Cordova ou PhoneGap para ser executado como um aplicativo nativo, assim como acessar algumas funcionalidades nativas do equipamento.

## Angular

Angular é uma plataforma e *framework* para construção da interface de aplicações usando HTML, CSS e, principalmente, *JavaScript*, criada pelos desenvolvedores da *Google*.

O angular é composto basicamente por componentes, templates, diretivas, roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas, como a de executar os testes unitários de uma aplicação.

Foi escolhido *framework* citado acima pela facilidade de criação das chamadas SPA´s “*Single Page Applications*”, em sua tradução significa Aplicação de Página Ùnica, modelo que vem conquistando o mercado por gerar resultados com qualidade e produtividade.

## Java

Java é uma linguagem poderosa em ambientes distribuídos complexos como a rede Internet. Mas sua versatilidade permite ao programador ir além, oferecendo uma poderosa linguagem de programação de uso geral, com recursos suficientes para a construção de uma variedade de aplicativos que podem ou não depender do uso de recursos de conectividade.

Com base nas pesquisas realizadas, optou-se pela linguagem de programação JAVA, pelo fato de nos oferecer uma maior diversidade em relação a liberdade de desenvolvimento.

## Hibernate

O Hibernate consiste em um poderoso framework utilizado para realizar o ORM (Mapeamento Objeto-Relacional), e como a maior parte de um uma aplicação envolve na criação e manutenção de camadas de persistência, se o modelo de banco de dados sofrer alterações, pode ser muito oneroso realizar as alterações para todo o resto da aplicação, fazendo com que o Hibernate elimine esta lacuna [KRAEMER, VOGT 2004].

Como o framework *Hibernate* utiliza sua própria linguagem SQL, o HQL (Hibernate Query Language), o programador não necessita de escrever muitos códigos SQL para acesso ao banco de dados, pois o HQL possui um considerável aumento na velocidade de desenvolvimento, podendo mudar a qualquer momento o SGDB (Sistema de Gerenciamento de Banco de Dados) utilizado [FRAGOSO 2008].

## Maven

Utilizado para gerenciamento de bibliotecas, o *Maven* é ajudar os desenvolvedores que antes tiravam bastante de seu tempo buscando as dependência de tecnologias a serem utilizadas em seus projetos. Com a utilização do Maven, os desenvolvedores não perdem tempo procurando bibliotecas para o software, o gerenciador já traz consigo um arquivo que possuirá as dependências do projeto, este arquivo é chamado de pom (*Project Object Model*), por padrão, possuindo o tipo de arquivo “.xml”, onde são declarados os tipo de empacotamento e versões e repositórios das dependências.

Os repositórios remotos ficam em um servidor *web* e centralizam todas as bibliotecas Java, bibliotecas *open source*, utilizadas como dependências ou artefatos dos próprios projetos da empresa, que por sua vez podem ser utilizados como dependências de outros projetos [Mieko 2013].

## Git

O Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, de projetos pequenos a muito grandes, com velocidade e eficiência.

Considerando em resumo o que Pressman (2011) aponta, uma equipe de desenvolvimento deve ter uma gestão eficaz das versões do produto para permitir que os desenvolvedores possam retroceder versões anteriores durante o teste e depuração. (DART, 1991, apud PRESSMAN, 2011, p.523) aponta que o controle de versão “captura todas as alterações a todos os arquivos na configuração juntamente com a razão para aquelas alterações e aos detalhes de quem as fez e quando”, permite assim à uma grande equipe de desenvolvedores trabalhar de forma cooperativa.

Controle de versão é um sistema que registra alterações em um arquivo ou conjunto de arquivos ao longo do tempo para que você possa lembrar versões específicas mais tarde. Com isso, os desenvolvedores terão a facilidade de analisar o código desenvolvido por ele mesmo e por possíveis colaboradores, já que os repositórios são compartilhados na internet.

## PostgreSQL

É um Sistema de Gerenciamento de Banco de Dados (SGBD) Relacional, utilizado para armazenar informações de soluções de informáticas em todas as áreas de negócios existentes, bem como administrar o acesso a estas informações.

## Intellij IDEA - IDE (Ambiente de Desenvolvimento Integrado)

IDE, ou Ambiente Integral de Desenvolvimento em tradução livre, é um software criado com a finalidade de facilitar a vida dos programadores. Neste tipo de aplicação estão todas as funções necessárias para o desenvolvimento desde programas de computador a aplicativos mobile, assim como alguns recursos que diminuem a ocorrência de erros nas linhas de código.

Se no passado os desenvolvedores precisavam apenas de um editor de texto e de um navegador para criar um software, agora, com os IDEs, eles possuem mais opções para otimizar o tempo gasto com os códigos. Imagine os IDEs como as calculadoras. Logicamente você aprende a fazer as operações matemáticas na escola, mas raramente as faz manualmente quando precisa.

Uma das principais vantagens dos IDEs está na capacidade de compilar bibliotecas completas de linguagem. Outra função bastante comum neste tipo de programa são os *debuggers*, que apontam os erros que ocasionalmente podem ocorrer ao escrever o código. Alguns IDEs também possuem o autocompletar.

# DESENVOLVIMENTO

## APLICATIVOS MÓVEIS

A utilização de aplicativos móveis é sempre uma boa alternativa para automatizar processos repetitivos e gerir melhor as ações realizadas por pessoas, seja para envio de e-mails mensagens, compartilhamento de imagens gerando assim um processo mais performático e obtendo vantagens e economia de tempo. O mercado corporativo tem investido cada vez mais introduzir as suas melhorias as tecnologias Mobile. Segundo pesquisas, mais de 3 bilhões de pessoas têm um telefone celular, e o mercado de aplicativos virou uma febre. (Ricardo R. Lecheta; Android 4ª ed. Novatec).

De acordo companhia *App Annie,* em estudo realizado no ano de 2017, o Brasil é campeão mundial em uso de aplicativos de celular por dia.

A pesquisa aponta ainda que o brasileiro usa em média 12 aplicativos a cada 24 horas em seu *smartphone*.

No cenário de reservas de campos e as partidas ocorridas nestes ambientes considerando ainda os processos de escalação de times, é de extrema importância que aquele que disponibiliza seu estabelecimento (campo) para jogos, tenha um melhor gerenciamento do seu espaço, seja com relação ao tempo disponibilizado, ou no controle de reservas e também na divulgação de seu espaço.

## PRINCÍPIOS E FUNDAMENTOS DO JOGA FÁCIL

O aplicativo Joga Fácil traz alguns princípios fundamentais objetivando levar a facilidade e simplicidades a seus usuários, como usabilidade, interação clara e objetiva, e principalmente sua compatibilidade com todos os sistemas operacionais disponíveis no mercado. A exemplo do *Android* sistema operacional da Google, e sistema IOS disponibilizado pela Apple.

Para aqueles que buscam reservar espaço esportivos (campos, quadras) para realizar partidas de futebol, é importante que os jogadores tenham uma maior facilidade no processo de reserva e na localização do espaço. O aplicativo contará com uma forma clara e objetiva de realizar a reserva, além de disponibilizar a localização do estabelecimento.

Destaca-se também o papel de um líder (técnico) para a distribuição das equipes, considerando que processo este é realizado manualmente, dificultando todo processo de organização, que gera na maioria das vezes consumo do tempo, tempo este pago para praticar a atividade propriamente dita.

## LÓGICA DE NEGÓCIO DA APLICAÇÃO

O aplicativo Joga Fácil contará com regras de negócio necessárias para que seu funcionamento seja adequado e fiel a todas as suas funcionalidades, para que dessa forma os usuários tenham êxito em suas requisições para com o sistema.

Tendo como escolha o framework Ionic que é um *framework* para desenvolvimento de aplicações para dispositivos móveis que possibilita o desenvolvimento de apps híbridas e de rápido e fácil desenvolvimento. (GRILLO, 2015).

Por possibilitar um desenvolvimento de aplicações híbridas, ou seja, compatíveis com os sistemas operacionais mais utilizados hoje no mercado, o aplicativo a ser desenvolvido terá uma maior chance de utilização por parte dos usuários.

## Regras para Inserção de Dados

O Joga Fácil possuirá oito entidades no seu total, são elas: Usuário, Tipo de usuário, Time, Reserva, Posição, Estabelecimento, Endereço e, Campo. Estas entidades possuem campo específicos que servirão como referência para o sistema e relacionamentos entre si (vide [Figura 01](#_m0xv0cmo9vgf)).

### Usuário

A entidade Usuário será responsável por guardar todos os usuários do aplicativo, garantindo a segurança e integridade dos dados, esta entidade possui os atributos, id, email, senha, id\_posicao, id\_tipo\_usuario , onde o id(*PK*[[5]](#footnote-6)), e e-mail(*UQ[[6]](#footnote-7)*) serão únicos, o sistema não deverá permitir usuários com o mesmo id e e-mail. O campo senha deverá conter no mínimo 6 caracter, caso contrário o sistema invalidará o cadastro. O campo id\_tipo\_usuario (FK) irá identificar o perfil do usuário a ser cadastrado no aplicativo de acordo com a entidade [tipo de usuário](#_ofazfifjwaop).

O campo id\_posicao (FK) identificará a posic

### Tipo de Usuário

A entidade Tipo de Usuário é responsável pode identificar os perfis de usuário do aplicativo. São os tipos de usuários pré-definidos: Jogador, Treinador, Proprietário. No ato do cadastro de usuário, o sistema exigirá a seleção de um perfil para que possa trabalhar melhor as regras de negócio.

### Time

A entidade Time possuirá todos os times cadastrados no aplicativo. Esta entidade está modelada com os seguintes atributos, id(*PK*), descrição, id\_usuario(*FK)*, onde o atributo id\_usuario será a chave estrangeira da tabela usuário, que por si identifica o usuário logado no aplicativo.

### Reserva

A entidade Reserva possuirá todas a reservas de um determinado campo e contém os seguintes campos: data\_da\_reserva, valor\_reserva, id\_campo(FK), este último será a identificação da reserva através da identificação do campo reservado.

### Posição

A entidade Posição terá a função de guardar as posições dos jogadores do time, esta entidade possui os seguintes campos: id, descrição. Todas as posições serão inseridas na tabela de imediato, possibilitando ao usuário a visualização destas.

### Estabelecimento

A entidade Estabelecimento representa o estabelecimento físico composto dos campos a serem reservados. Esta tabela possui os seguintes campos: id, descrição, email, id\_endereco, id\_usuario. Este último será o responsável pelo estabelecendo, pois o sistema atribui o seu perfil como proprietário(vide [Tipo de Usuário](#_ofazfifjwaop)).

### Endereço

A entidade Endereço representa o endereço do estabelecimento para facilitar o acesso pelos usuários. O modelo representativo desta entidade possui os seguintes campos: id, bairro, cidade.

### Campo

A entidade campo descreve o espaço a ser reservado, esta contém os campos na tabela, como: id, descricao, status, id\_estabelecimento(*FK*). O campo status irá informa aos usuários se o campo está disponível ou não.

# ARTEFATOS DESENVOLVIDOS

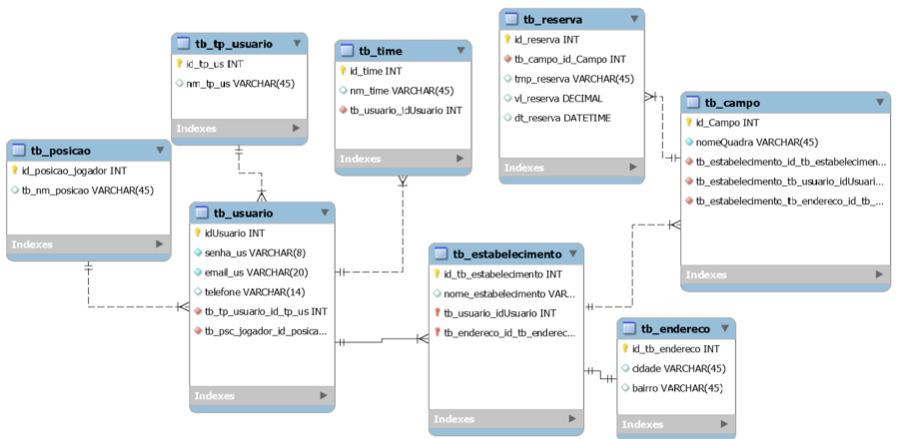
São considerados artefatos todo e qualquer documento criado para compor um *software* em seu desenvolvimento, como por exemplo: diagramas, casos de testes, protótipos de tela, especificação de caso de uso, entre outros documentos desenvolvidos no decorrer do processo de criação do projeto.

## Diagrama de Entidade e Relacionamento – DER

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).

Em geral, este modelo representa de forma abstrata a estrutura que possuirá o banco de dados da aplicação. Obviamente, o banco de dados poderá conter várias outras entidades, tais como chaves e tabelas intermediárias, que podem só fazer sentido no contexto de bases de dados relacionais. (Disponível em <https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>).

Figura 1 - DER do Banco de Dados da Aplicação JOGA FÁCIL



## Diagramas de Caso de Uso

FALTAM COLOCAR UMA EXPLICAÇÃO AQUI, DO MESMO JEITO QUE FIZERAM COM O DER/MER

Figura 2 - Funcionalidades realizadas pelo ator Técnico

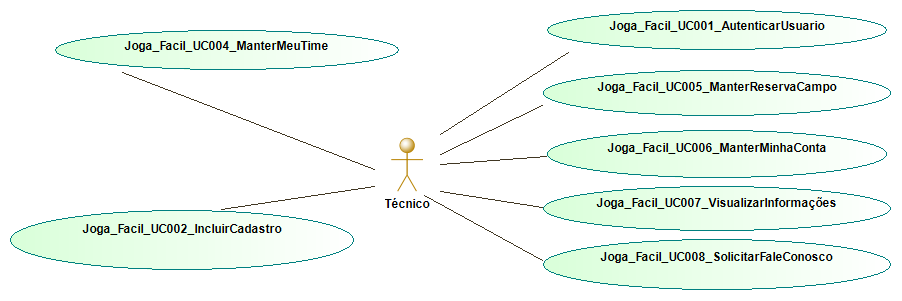
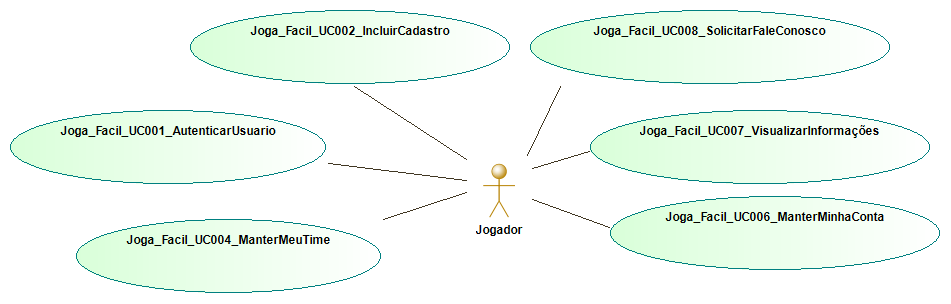


Figura 3 - Funcionalidades realizadas pelo ator Jogador



# CONSIDERAÇÕES FINAIS

Este trabalho teve como propósito oferecer, de forma sintética e objetiva, uma alternativa para resolver eventuais problemas quando se busca reservar campos de futebol.

Verificou-se que com o uso da tecnologia, em sua modalidade móvel, as pessoas tendem a se beneficiarem em suas tarefas diárias. Benefícios estes que vão desde a economia de tempo, até mesmo à economia econômica. O avanço da tecnologia em dispositivos móveis permite alcançar a maioria da população e, segundo a [FGV (Fundação Getúlio Vargas)](https://exame.abril.com.br/tecnologia/brasil-tera-um-smartphone-por-habitante-ate-outubro-diz-fgv/) calcula-se que o Brasil atingiu a taxa de um (01) *smartphone* por habitante em 2017. Esse número não indica que 100% da população possui um aparelho do tipo, visto que em alguns casos uma mesma pessoa pode ter mais de um aparelho celular ou mais de um chip habilitado, mas ainda assim é um dado que chama à atenção. Portanto conclui-se que a área de desenvolvimento de App para *smartphones* é bastante promissora para investir na área de Tecnologia.

**REFERÊNCIAS**

ANGULAR, Angular – Style Guide. Disponível em <https://angular.io/guide/styleguide>. Acesso em: 26 nov. 2018.

ASCENCIO, Ana Fernanda Gomes; DE CAMPOS, Edilene Aparecida Veneruchi. Fundamentos da programação de computadores. Pearson Educación, 2008.

CONTROLE DE VERSÃO, <https://git-scm.com/book/pt-br/v2/Come%C3%A7ando-Sobre-Controle-de-Vers%C3%A3o>.

IONIC FRAMEWORK, Ionic – Componentes. Disponível em <https://ionicframework.com/docs/components/>. Acesso em: 26 nov. 2018.

FRAGOSO, R. R.,(2008)" O que é Hibernate.

INDRUSIAK, Leandro Soares. Linguagem java. Grupo JavaRS JUG Rio Grande do Sul, 1996.

J2EE - <https://pt.wikibooks.org/wiki/J2EE/Introdu%C3%A7%C3%A3o>. Acesso em: 20 out. 2018.

MRACK, M. (2006) "Hibernate, Uma visão geral sobre o framework padrão de fato para mapeamento objeto-relacional", <http://www.slideshare.net/mmrack/workshop-hibernate-com-comentarios>, acessado em 29 de Julho de 2013.

PARA QUE SERVE UMA IDE?.Disponível em <https://www.psafe.com/blog/o-que-serve-ide/>. Acesso em: 26 nov. 2018.

SILVA, Bruno Costa; RUFINO, Ricardo. Frameworks funcionais para JSF que proporciona o desenvolvimento de aplicaç oes computacionais WEB.

SPRING FRAMEWORK - <<https://spring.io/>>. Acesso em 03 de novembro de 2018.

PACHECO, Diego. 10 motivos para utilizar maven. Disponível em: <http://imasters.com.br/artigo/12336/java/dez-motivos-para-voce-usar-maven/>. Acesso em 03 de novembro de 2018.

MILANI, André. PostgreSQL-Guia do Programador. Novatec Editora, 2008.

TAVARES, Henrique Leal. Introdução a Desenvolvimento de Aplicações Híbridas. Revista Eletrônica eF@ tec, v. 6, n. 1, p. 11-11, 2016.

WEISSMANN, Henrique Lobo. Vire o jogo com Spring Framework. Editora Casa do Código, 2014.

1. \*\*\* Mestre em Computação Aplicada Pela Universidade de Brasília, Coordenador de Infraestrutura de Redes do Ministério da Defesa e Professor Acadêmico da Universidade de Brasília e das Faculdades JK. E-Mail: [carlos.mello@jk.edu.br](mailto:carlos.mello@jk.edu.br) [↑](#footnote-ref-2)
2. \*\*\* Mestranda em Ciências da Computação - UNB. Especialização em Gestão Pública - UNB. Especialização em Segurança da Informação - UNB. Bacharel em Análise de Sistemas - Universo. [↑](#footnote-ref-3)
3. \*\* Graduado em Análise e Desenvolvimento de Tecnologia pela Faculdade JK Santa Maria. Graduando em Bacharel em Sistema da Informação pela Faculdade JK Gama. [↑](#footnote-ref-4)
4. Graduando Análise e Desenvolvimento de Tecnologia pela Faculdade JK Gama. [↑](#footnote-ref-5)
5. Primary Key, ou chave primária, atributo responsável por identificar um campo da tabela como único e inviolável, utilizado para facilitar sua identificação em meio às demais. [↑](#footnote-ref-6)
6. Unique, ou único, define um campo na tabela como único, não permitido a existência de um valor igual na base. [↑](#footnote-ref-7)