

INSTITUTO FEDERAL DO PARANÁ  
CURSO DE GESTÃO DA TECNOLOGIA DA INFORMAÇÃO

ANSELMO DE ABREU NICOSKI

**APLICATIVO PERSONAL DANCER**

TRABALHO DE CONCLUSÃO DE DISCIPLINA

CURITIBA  
2024

ANSELMO DE ABREU NICOSKI

**APLICATIVO PERSONAL DANCER**

Trabalho de conclusão de disciplina, do curso superior de Gestão da tecnologia da informação, do instituto federal do Paraná, apresentado como requisito parcial a aprovação da disciplina.

Orientador: Alexandre Peres Arias.

CURITIBA

2024

## **FICHA CATALOGRÁFICA**

Espaço destinado a elaboração da ficha catalográfica sob responsabilidade exclusiva do Departamento de Biblioteca do  
INSTITUTO FEDERAL DO PARANÁ

## **TERMO DE APROVAÇÃO**

Anselmo de Abreu Nicoski

### **Aplicativo Personal Dancer**

Este trabalho de conclusão de disciplina foi julgado adequado para a obtenção do título de aprovado na disciplina de dispositivos móveis do terceiro período, pelo professor e pelo departamento de tecnologia do Instituto Federal do Paraná.

Curitiba, \_\_\_\_\_ de \_\_\_\_\_ de 2024.

---

Coordenador do curso  
Instituto Federal do Paraná

#### **Banca Examinadora:**

---

Orientador  
Instituto Federal do Paraná

---

Membro  
Instituto Federal do Paraná

---

Membro  
Instituto Federal do Paraná

Quando finalizado,  
dedico este trabalho à minha família.

## **AGRADECIMENTOS**

Certamente estes parágrafos não irão atender a todas as pessoas que fizeram parte dessa importante fase de minha vida. Portanto, desde já peço desculpas àquelas que não estão presentes entre essas meras palavras, mas elas podem estar certas que fazem parte do meu pensamento e de minha gratidão.

Agradeço aos meus orientadores pela sabedoria com que me guiaram nesta trajetória.

Aos meus colegas de sala.

Gostaria de deixar registrado também, o meu reconhecimento à minha família, pois, acredito que sem o apoio deles seria muito difícil vencer mais esse desafio.

Enfim, a todos que por algum motivo, direta ou indiretamente, contribuíram para a inspiração deste projeto que concluirei em breve.

Melhor morrer tentando.  
(Pulga)

## RESUMO

NICOSKI,A. Anselmo. **APLICATIVO PERSONAL DANCER**. 2022. Trabalho de Conclusão da Disciplina de dispositivos móveis(Gestão de tecnologia da Informação) – Instituto Federal do Paraná. Curitiba - PR, 2024.

O objetivo desse projeto é de favorecer a inclusão de jovens, adultos e idosos em espaços de prática de dança de salão. Para isso, será desenvolvido um aplicativo que favoreça o encontro de pessoas que querem apreender e praticar a dança com pessoas que querem ensinar e dançar.

**Palavras-chave:** aplicativo, dança, cliente, personal, contratação.



## **ABSTRACT**

**NICOSKI**, A. Anselmo. **PERSONAL DANCER APP**. 2022. Completion project for the Discipline of Mobile Devices (Information Technology Management) – Instituto Federal do Paraná. Curitiba – PR, 2024.

The objective of this project is to further the inclusion of young, adults and old people in spaces of practice ballroom dancing.

**Keywords:** app, dance, client, personal, hiring.



## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 TEMA.....	12
1.2 DELIMITAÇÃO DO ESTUDO.....	14
1.3 PROBLEMA.....	14
1.4 JUSTIFICATIVA.....	15
1.5 PÚBLICO ALVO.....	15
1.6 OBJETIVOS.....	16
1.6.1 Geral.....	16
1.6.2 Específicos.....	16
1.7 VIABILIDADE DO PROJETO.....	17
1.7.1 Viabilidade funcional.....	17
1.7.2 Viabilidade Financeira.....	17
1.7.3 Viabilidade Técnica.....	18
1.8 ANÁLISE DO AMBIENTE ORGANIZACIONAL.....	18
1.8.1 Tecnológicas.....	18
1.8.2 Social.....	19
1.8.3 Demográfica.....	19
1.9METODOLOGIAS DE DESENVOLVIMENTO.....	20
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>21</b>
2.1 Aplicativos disponíveis no mercado.....	22
2.2 Mobilidade.....	22
2.3 Dança e aplicativos móveis.....	22
2.4 Dança e saúde.....	23
2.5 Aplicativos móveis.....	23
<b>3 FERRAMENTAS E TECNOLOGIAS UTILIZADAS.....</b>	<b>24</b>
3.1Plataforma.....	24
3.2 Parse SDK.....	24
3.3 Google Maps.....	24
3.4 Firebase.....	25
3.5 Firebase Realtime Database.....	27
3.6 Firebase Cloud Messaging.....	28
3.7 API Cielo eCommerce.....	30
<b>4 DESENVOLVIMENTO.....</b>	<b>30</b>
4.1Descrição do mini mundo detalhado do sistema.....	30
4.2 Requisitos do Sistema Personal Dancer.....	31
<b>4.2.1 Diagramas de casos de uso.....</b>	<b>32</b>
4.2.2 Requisitos Funcionais.....	32
4.2.3 Requisitos não funcionais.....	33
<b>4.3 Caso de uso.....</b>	<b>34</b>
4.3.1 Caso de uso – Cadastrar Cliente.....	35
4.3.2 Caso de uso – Cadastrar Personal.....	36
4.3.3 Caso de uso – Contratar Personal.....	37
4.3.4 Caso de uso – Efetuar pagamento.....	39
4.3.5 Caso de uso – Receber Personal Dancer.....	41
<b>4.4Requisitos não funcionais.....</b>	<b>43</b>
4.4.1 Usabilidade.....	43
4.4.2 NF01 Usabilidade e interface.....	44
4.4.3 NF02 Usabilidade e tipo de usuário.....	44
4.4.4 Confiabilidade.....	45

4.4.5 NF03 Confiabilidade e Segurança da Informação.....	45
4.4.6 NF04 Confiabilidade e disponibilidade.....	46
4.4.7 Desempenho.....	47
4.4.8 NF05 Desempenho e tempo de resposta.....	47
4.4.9 Segurança.....	47
4.4.10 NF06 Segurança – usuário.....	48
4.4.11 Padrões.....	49
4.4.12 NF10 Manutenção de correções.....	49
4.4.13 NF11 Tratamento de exceções.....	49
4.4.14 Hardware e software.....	50
4.4.15 NF12 Flutter.....	50
<b>4.5 Regras de negócio.....</b>	<b>51</b>
<b>4.6 Diagramas de sequência.....</b>	<b>51</b>
4.6.1 Cadastrar Cliente.....	51
4.6.2 Cadastrar Personal Dancer.....	52
4.6.3 Alterar cadastro do Cliente.....	52
4.6.4 Alterar Cadastro do Personal Dancer.....	53
4.6.5 Excluir Cliente.....	53
4.6.6 Excluir Personal Dancer.....	54
4.6.7 Cliente contratar Personal Dancer.....	54
4.6.8 Cliente cadastrar cartão de crédito.....	55
4.6.9 Cliente efetuar pagamento.....	55
4.6.10 Cliente receber Personal Dancer.....	56
<b>4.7 Fluxograma.....</b>	<b>56</b>
4.7.1 Fluxograma completo.....	57
4.8 Diagrama de Atividade.....	58
4.8.1 Personal aceita contratação.....	58
4.8.2 Cancelar contratação.....	59
4.8.3 Iniciar período contratado.....	59
4.8.4 Finalizar do período contratado.....	60
4.8.5 Enviar avaliação da contratação do Personal Dancer.....	60
4.9 Diagrama de classes de objetos e seus relacionamentos.....	61
4.10 Diagrama de entidade relacionamento.....	63
4.11 Projeto de Banco de Dados.....	64
4.12 Projeto de Implementação.....	69
4.13 Interfaces gráficas.....	75
4.13.1 TextView.....	76
4.13.2 Button.....	76
4.13.3 RadioButto.....	77
4.14 Plano de testes.....	79
4.15 Análise de riscos.....	81
4.15.1 Intents e riscos.....	81
4.15.2 Banco de dados e riscos.....	82
4.16 Orçamento.....	83
<b>5 CONCLUSÃO.....</b>	<b>84</b>
5.1 Dificuldades encontradas no trabalho.....	86
5.2 Trabalho e melhoras futuras.....	86
<b>REFERENCIAS.....</b>	<b>88</b>
<b>APENDICES.....</b>	<b>89</b>



# **1 INTRODUÇÃO**

## **1.1 TEMA**

Segundo Nahas (2003) as boas práticas de uma vida ativa, física e mental pode proporcionar uma melhor qualidade de vida as pessoas, retardando o processo de envelhecimento. Segundo o autor, a prática de bons hábitos regulares de atividades físicas, cultivados a partir dos trinta anos são de fundamental importância para a capacidade de movimentação das pessoas com idade acima dos sessenta anos. O autor reforça que as boas práticas devem ser cultivados preferencialmente desde a infância, pois, o segredo para um envelhecimento saudável é ser ativo sempre.

Segundo Maia et al.(2007) a atividade física que melhora a qualidade de vida das pessoas em qualquer idade é a dança de salão. Isso ocorre devido ao fato de a dança de salão ser uma das modalidades de dança mais inclusiva, com acesso a diversos meios, sejam eles sociais, intelectuais e culturais.

Assim, a dança de salão como atividade física é de fundamental importância para a melhora gradativa da qualidade de vida das pessoas de todas as idades, principalmente para as pessoas da terceira idade, cujos cuidados devem ultrapassar o horizonte das atividades físicas isoladas.

O processo de envelhecimento é contínuo e pode ter sua etapa acelerada ou retardada, como qualquer outro processo da vida biológica humana que trabalha em função de perdas e ganhos. Quando desfrutamos da dança nós ganhamos e ganhamos vida ao frear o avanço vicioso da idade, ao frear o sedentarismo, ao frear doenças e principalmente ao melhorar nossa capacidade locomotora. Além disso, na dança ocorre o dialogo corporal e pessoal que chamamos de conexão que é de extrema importância para os ganhos psicológicos e sociais. Por outro lado, nós perdemos quando nos acomodamos a uma vida sedentária, perdemos porque reduzimos nosso tempo de vida, perdemos porque vivemos menos felizes e muitas vezes, como menos dignidade.

Além dos ganhos psicológicos e sociais já citados, a dança proporciona ao indivíduo idoso, segundo Hashizumi et al.(2004) uma maior autonomia e independência ao contribuir com a manutenção da força muscular atreladas a sustentação do corpo, equilíbrio e resistência aeróbica que são indispensáveis para

o bom funcionamento dos pulmões , coração e artérias. Em relação a sustentação do corpo e equilíbrio a dança de salão é de extrema importância para a melhora do equilíbrio, pois, durante a dança ocorre troca de peso e contrapeso, variações de deslocamentos, ora para frente, ora pra traz, para o lado direito, para o lado esquerdo. Essas alterações do eixo do corpo são exercícios moderados de equilíbrio, reconhecida por Almeida et al.(2004) como sendo uma 'atividade físico-recreativa', pois, segundo ele, idosos consideram que a dança de salão é um momento de lazer e divertimento.

Quem pratica a dança de salão sabe que a dança produz dopamina no cérebro, trazendo a tona a sensação de bem estar. Segundo Fortes (2008) a dança reduz a depressão, melhora o humor e melhora a sensação de bem estar. A dança de salão é uma espécie de terapia em grupo, conforme descreveu Okuma(1998), pois ela facilita a integração e fortalece laços de amizades, sendo portanto, a melhor atividade física para pessoas da terceira idade, mas também, uma atividade prazerosa para as demais idade.

Estima-se que em média, mais de 1000 pessoas tem acesso ao nível intermediário de dança no estado do Paraná. Isso mostra que muitas pessoas por vários motivos, buscam alcançar um alto nível de excelência na dança, mesmo sabendo que o mercado ainda não é o ideal e que após atingir um alto nível de excelência na dança fica cada vez mais difícil encontrar um parceiro que possa acompanhar no mesmo nível. Sob essa óptica, muitas pessoas pagam aulas mensais em Studio de dança para manter o nível e obviamente melhorá-lo, sem a intenção de ganhar dinheiro com a dança e muitos menos ensinar a quem sabe menos.

O aplicativo Personal Dancer abrirá portas para muitas pessoas colocarem em prática seu aprendizado melhorando a condução e expandindo repertórios, pois, também é possível apreender com quem sabe menos que a gente e do mesmo modo, se aprende muito com quem sabe mais. Nesse viés, muitas pessoas irão aderirem ao aplicativo Personal Dancer, algumas como hobby outras como profissão.

Ao final desse estudo, será apresentado um aplicativo, baseado na minha experiência com a dança e nos conhecimentos adquiridos ao longo do curso de graduação, utilizando a linguagem de programação Flutter/Dart utilizando a plataforma gratuita do google chamada Android Studio ou Visual Studio da Microsoft.

## 1.2 DELIMITAÇÃO DO ESTUDO

O objetivo desse projeto é de estudar e desenvolver um aplicativo Android em linguagem Flutter/Dart que promova a inclusão de jovens, adultos, idosos e profissionais da dança em espaços públicos de dança de salão.

## 1.3 PROBLEMA

A ideia do aplicativo 'Personal Dancer' surgiu da necessidade de regulamentação da dança como profissão nos salões de baile de Curitiba. Trata-se de dizer que a profissão 'Personal Dancer' existe há séculos, na base da amizade e da indicação. Em Curitiba, muitos pais sustentam suas famílias através do serviço de dança remunerado. Ou seja, a ideia já existe o que não existe é um aplicativo que regulamente e expanda o leque de opções de 'Personais Dancers' a disposição dos apaixonados pela dança. A dança de salão embora fomente uma maior interação e convívio social falado e dançado vem perdendo espaço para o funk e outras modalidades que a pessoa dança sozinha. Com base em minha experiência com a dança, isso acontece justamente por falta de um companheiro de dança que se conecte ao estilo da dança corporal da pessoa. Tanto homens quanto mulheres após alcançar um alto nível de excelência na dança, encontram muitas dificuldades para encontrar um parceiro de dança no mesmo nível, sem falar daqueles que estão no nível intermediário e que portanto, querem alcançar um nível maior de excelência, bem como, aqueles que sabem o básico para acompanhar uma dança e que, por razões óbvias, querem ampliar exponencialmente seu nível de excelência e repertório, através da prática com um 'Personal Dancer' capacitado.

Com o aplicativo, após o fim do período contratado o Personal Dancer estará disponível para ser recontratado ou contratado por outra pessoa. Diferente do modelo atual, cujo Personal Dancer fica o 'baile todo' atrelado a pessoa que o contratou a um preço reduzido para não perder o cliente.

## 1.4 JUSTIFICATIVA



O mercado da dança está em franca expansão, principalmente no Brasil motivados por programas televisivos a exemplo: 'dança dos famosos' apresentado pela emissora Globo e principalmente porque anualmente milhares de pessoas se formam em dança nas universidades, studio de dança e clubes de dança no Brasil e no exterior. A maioria dos dançarinos com formação, não praticam a dança como profissão e nem como 'hobby'. Com o aplicativo, alguns iniciarão a prática como profissão e outros como 'hobby' pelo simples fato de gostar de dançar, de poder ganhar dinheiro com a dança e também porque o aplicativo disponibilizará diversos Personais Dancers capacitados que poderão acompanhar seu ritmo em uma dança agradável e de alto nível. Do mesmo modo, o aplicativo disponibilizará aos Personais Dancers diversos Clientes de todas as idades que pagarão para ter uma dança agradável.

Portanto, com o crescimento do uso de aplicativos Android nos últimos anos (Amador, 2015), acho interessante a ideia de usufruir dessa tendencia tecnológica que sinaliza não ter limites de expansão. Ao fim deste projeto será apresentado um aplicativo que irá satisfazer as expectativas descrita anteriormente.

### 1.5 PÚBLICO ALVO

O público-alvo é diversificado e por isso, deverá ser melhor estudado, pois, o aplicativo atenderá clientes de 18 anos completos a 70 anos. Inicialmente como a maioria das pessoas que praticam a dança de salão nos clubes da cidade de Curitiba possuem idade acima dos 30 anos, vamos dar atenção especial a faixa etária dos 30 aos 60 anos. Obviamente, como o objetivo é de atender e abraçar todas as idades, vamos orientar e treinar Personais Dancers para atender clientes acima dos 50 anos para que a dança seja divertida e ao mesmo tempo, uma terapia prazerosa. Quanto ao publico jovem, vamos desde o início pensar em estratégias para conquistá-los pois, é esse publico que fará o aplicativo crescer e se manter, pois, alguns jovens clientes poderão ser os 'Personais Dancers' do futuro ou um cliente assíduo.

### 1.6 OBJETIVOS

O objetivo é estudar e desenvolver um aplicativo que permita o cadastro e o login de usuários Clientes e usuários Personais Dancers. Dessa forma, um cliente logado e o com o GPS e local do clube devidamente validado poderá contratar um Personal Dancer. E, por conseguinte, um Personal Dancer devidamente logado e

com seu GPS e local validado poderá aceitar a solicitação de contratação do Cliente.

#### 1.6.1 Geral

De modo geral, o objetivo é desenvolver um aplicativo que ofereça o serviço de conhecimento do próprio corpo visando a melhora do equilíbrio, dos limites dos movimentos corporais e de uma atividade física prazerosa por meio da dança, tais como: lateralidade, coordenação motora (ambas muito útil no combate ao mal de Parkinson), bem como, disciplina, cuidados com o corpo e atenção concentrada.

#### 1.6.2 Específicos

- Estudar e desenvolver um aplicativo que oferecerá o serviço de dança a clientes devidamente cadastrados. Para isso, tanto o cliente contratante, quanto o cliente que disponibilizará o serviço de dança, deverão estar cadastrados no aplicativo.
- Estudar e desenvolver aplicativo oferecerá o serviço de aprendizado das principais danças de salão do Brasil, com ênfase nas danças regionais, tais como: na região sul, sertanejo e gauchesco, no nordeste, forró pé de serra e forró eletrônico, no sudeste, sertanejo, forró universitário e samba, bem como outros estilos.
- Estudar e desenvolver um aplicativo que ofereça apenas a modalidade de pagamento via cartão de crédito.
- estudar e desenvolver um aplicativo que, por razões de segurança, permita a contratação de um Personal Dancer se, somente se, ambos estiverem no mesmo clube selecionado no Aplicativo, nos dias e horários de baile.
- Após o desenvolvimento, realizar diversos testes para validar o funcionamento do aplicativo(cadastro, login, contratação, aceitação, cancelamento, encerramento) utilizando ao menos três locais diferentes e três versões de Android diferentes a partir da versão 10 para identificar e corrigir possíveis falhas.

### 1.7 VIABILIDADE DO PROJETO

A viabilidade do projeto depende de muitos fatores e aspectos que devem ser levados em consideração antes da difícil decisão de levar o projeto a diante. Deve

se colocar em pauta a quantidade de recursos financeiros que se está disposto a investir, a disponibilidade de tempo, a disponibilidade de mão de obra, a disponibilidade de oportunidades que poderão surgir e o quanto se poderá ganhar.

#### 1.7.1 Viabilidade funcional

Na viabilidade funcional ou operacional, há dois aspectos que valem apenas ser considerados nesse projeto. Primeiro, se tratarmos o desenvolvimento do projeto como a solução de um problema identificado por acaso, chegaremos a um denominador comum que afere que o problema vale a pena ser resolvido. Segundo, se tratarmos o problema como um problema que existe há séculos e que a solução do problema existente é anterior a disseminação e desenvolvimento da tecnologia móvel atual e que funciona muito bem dentro dos moldes da cultura popular, cuja solução, é a mesma que apresentaremos no aplicativo, chegaremos a conclusão de que a solução proposta para o problema funcionará e que vale a pena investir.

#### 1.7.2 Viabilidade Financeira

Nas fases iniciais do projeto, fazemos uma breve análise da viabilidade financeira, que consiste apenas em fazer algumas inferências e emitir uma conclusão, talvez distante dos dados reais. Assim, trata-se de julgar se o projeto concluído pode ser lucrativo a curto, médio e longo prazo. E a resposta é sim, desde que seja disponibilizado cerca de 5 anos no projeto, não como um projeto acadêmico, mas sim, como um negócio com potencial gigante de crescimento na esfera federal e internacional.

Ainda na análise da viabilidade financeira deve-se considerar o custo benefício do projeto. Nesse sentido, o primeiro custo benefício que vem à cabeça é: Quanto custou ou custará para desenvolver o sistema? Esses dados ainda serão calculados e apresentados no momento oportuno, junto com licença de software, pagamento por utilização, possíveis manutenção, telefone, internet, energia elétrica tempo empregado no projeto e afins.

#### 1.7.3 Viabilidade Técnica

Na fase inicial do projeto, nos deparamos com a análise da viabilidade técnica. Nessa parte, devemos fazer varias perguntas e respondê-las com honestidade, a saber: o aplicativo proposto é prático? Já possuímos as ferramentas tecnológicas necessárias para a construção do aplicativo? Já possuímos o conhecimento técnico necessário para o desenvolvimento do aplicativo?. Por conseguinte, a última pergunta, a respeito do conhecimento técnico de codificação do aplicativo é o maior desafio para a conclusão desse projeto nessa disciplina. Possivelmente, a conclusão do projeto será adiada, uma vez que estamos próximos do encerramento da disciplina.

## 1.8 ANÁLISE DO AMBIENTE ORGANIZACIONAL

Segundo Chiavenato (2003), o ambiente organizacional é constituído de sete variáveis: tecnológicas, políticas, econômicas, legais, sociais, demográficas e ecológicas. Assim, para aumentar a compreensão é necessário reduzir a extensão e, para isso, descreverei apenas três variáveis que considero as mais importantes para o projeto, a saber: tecnológicas, sociais e demográficas.

### 1.8.1 Tecnológicas

É a variável mais complexa e pode causar um forte impacto nas organizações, pois, corresponde a tecnologia desde os rascunhos das ideias iniciais, passando pelo desenvolvimento e aprimoramentos até chegar ao momento da conclusão e entrega do projeto. Por conseguinte, agora dentro do cenário da dança, surgem questões que anseiam por respostas tais como: quem fará parte desse projeto? Como farão parte? A resposta é simples, no inicio será necessário todos os tipos de marketing e divulgação, ou seja, será necessário trabalhar a divulgação tanto nos meios tecnológicos quanto nos meios formais. Porém, as dúvidas não cessarão de imediato surgirão outras como: saberão utilizar essa tecnologia? A resposta é sim, pois, o aplicativo será desenvolvido pensando na praticidade do usuário que com poucos toques, poderá cadastrar, logar, solicitar uma contatação, aceitar/rejeitar e encerrar uma requisição.

### 1.8.2 Social

A variável social, possui grande interferência no ambiente organizacional, a exemplo da opinião pública. Desde o início, faremos o possível no âmbito legal para saber o grau de satisfação dos clientes-usuários do aplicativo e sempre que necessário, irei implantar alterações visando aperfeiçoar cada vez mais a aplicação. Não descartamos a hipótese de que, em alguns determinados clubes, poderão haver uma rejeição dos serviços de Personal Dancer, pois, vivemos numa sociedade extremamente reacionária e nesse sentido, a novidade dinâmica assusta aqueles que preferem a cultura tradicional estática. Pensando nisso, iremos iniciar os trabalhos com Personais Dancer que estão inseridos na cultura de Personal Dancer a moda antiga, para depois, aceitarmos novatos no ramo. Ainda dentro da variável social, tem outras sub-variáveis que tem muita importância e repercussão no ambiente organizacional de um clube, a saber: as tradições culturais, o valor da família e da comunidade. Por conseguinte, o aplicativo irá expandir exponencialmente as tradições culturais regionais da dança de salão, pois, trabalharemos com o foco nas danças de salão raiz e de acordo com cada estado e cultura. Expandiremos a tradição familiar mantendo os bons valores da família, muito presente e marcante nas sociedades de dança da família gaúcha por exemplo, pois, acredito que esses bons valores devem ser mantidos, assim como, o valor de comunidade, de pertencimento a uma boa comunidade, a exemplo dos CTGs gaúchos.

### 1.8.3 Demográfica

Por fim, a terceira variável é a demográfica que nos permite analisar o mercado atual da dança de salão e prever o potencial de crescimento futuro, em função de outras sub-variáveis a exemplo das características dos clientes, tais como: idade, peso, altura, raça e estilo de dança preferido. A princípio, vamos nos ater apenas ao estilo de dança preferido dos clientes para adquirirmos cada vez mais, novos usuários. Conforme o aplicativo for crescendo e se tornando uma plataforma robusta, precisaremos criar nossos próprios clubes de dança com base nas exigências de estilos de dança. Isso já é necessário, ao menos em Curitiba que não tem um clube de dança exclusivo para o estilo de dança nordestino, a exemplo do forró pé de serra.

## 1.9 METODOLOGIAS DE DESENVOLVIMENTO

Inicialmente, selecionei três clubes de dança na região de Curitiba sendo eles, respectivamente: Sociedade Universal, Sociedade Internacional Água Verde e Clube Recreativo Dom Pedro II. Utilizando um trena eletrônica eu coletei as medidas do clube, a medida da pista, a medida do centro da pista até quinze metros para fixar um raio limite de contratação de Personais Dancers.

Realizei algumas entrevistas com alguns Personais Dancers que trabalham nos clubes de Curitiba há mais de uma década para coletar as seguintes informações: quais as principais dificuldades da profissão? O que acham do aplicativo? O que gostariam que o aplicativo tivesse? Essas foram as principais perguntas, e, em relação a primeira pergunta, muitos disseram que com o passar dos anos, eles acabam se tornando amigos dos clientes e isso dificulta muito a negociação e a cobrança dos valores combinados. Segundo eles, isso é diferente quando a pessoa é desconhecida, pois assim, é possível combinar os valores exatos e cobrar antecipadamente. Quanto a segunda pergunta, todos ficaram animados com a ideia pois, acreditam que poderão lucrar mais com a profissão ao obterem mais clientes. Por fim, a última pergunta obteve a mesma resposta, ou seja, querem que o aplicativo receba antecipadamente a contratação e que os valores sejam repassados para suas respectivas contas automaticamente, em tempo real. Outra sugestão interessante é que o aplicativo tenha um formulário de perfil, contendo idade, sexo de nascimento, altura, peso, cor e estilos de dança que a pessoa queira praticar. Quando indagados sobre o porque esse formulário seria útil, muitos responderam que é uma questão de preferência do contratante e não do contratado. Assim o contratante, antes de clicar em contratar saberia que o Personal, tem altura x, peso y e cor tal e só clicaria em contratar caso o perfil fosse compatível. Outros fizeram questão de deixar registrado que seria interessante que o aplicativo tivesse ao menos duas opções de foto de perfil, duas opções de fotos inteira de pé, pois, segundo eles, muitas contratantes escolhem pelo perfil bonito, pelo porte físico e não pela capacidade de conduzir uma bela dança.

Após a análise e coleta do inventário dos clubes e sugestões dos Personais Dancers, foi realizado um levantamento dos requisitos para dar início as implementações necessárias ao ambiente de desenvolvimento, tais como: celulares com Android 10 pra cima, computador com processador i5 pra cima, memória RAM

com no mínimo 12 gigas e conexão com a internet para baixar as atualizações e configurar e conectar ao banco de dados.

Na fase de desenvolvimento foi utilizada boas práticas e método de programação. A princípio, optei por usar o sistema operacional Linux por constatar que a IDE do Android Studio funciona melhor no sistema operacional Linux. O Android Studio é um ambiente de desenvolvimento integrado e foi criado especificamente para o desenvolvimento de aplicativos Android. A linguagem de programação utilizada é o Flutter/Dart e o banco de dados utilizado é o Firebase.

Por fim, após o código estiver pronto, serão realizados diversos testes de todos os tipos para validar o correto funcionamento do aplicativo, principalmente testes relacionados com a usabilidade e disponibilidade. Conforme sejam encontrados erros, serão corrigidos e implementados imediatamente.

## **2 FUNDAMENTAÇÃO TEÓRICA**

O aplicativo será desenvolvido com base em sólidos conhecimentos do mercado de clubes dança de salão de Curitiba. São vários anos de pesquisa e análise de campo, ora como professor, ora como entretenimento de fim de semana. Esse capítulo vai tentar elucidar e aproximar a relação entre dança e tecnologia em particular a utilização de dispositivos móveis na contratação de profissionais da dança.

### **2.1 Aplicativos disponíveis no mercado**

Até o momento, não foi encontrado nenhum aplicativo similar ao qual estou propondo desenvolver. Os poucos aplicativos de dança que estão disponíveis no Play Store se limitam apenas a apresentar vídeos aulas de dança e coreografias, a exemplo: “Just Dance Now”, “ZumbaDance”, “Pocket Salsa Free”, “How To dance Salsa” e etc.

### **2.2 Mobilidade**

Segundo LECHETA(2013) é a conectividade que da vida à palavra mobilidade. Assim, com a crescente expansão da internet sem fio, os dispositivos móveis vem cada vez mais ganhando espaço na sociedade, possibilitando novas formas de interação social. Segundo MARÇAL(2005), a aplicação móvel deve

permitir seu uso a qualquer hora e local, e, sempre que o usuário desejar. Porém, como o aplicativo Personal Dancer em particular trabalhará com horários noturnos preestabelecidos e não com horários diurnos comerciais, o aplicativo ficará disponível de acordo com o horário de início e fim dos respectivos eventos dos clubes de dança de salão. Os dispositivos móveis estreitarão as relações entre os usuários favorecendo a interatividade e o convívio social na dança. Segundo MARÇAL(2005), a 'interatividade é a razão de ser de um mundo virtual' e nesse sentido, se tratando de entretenimento, é a habilidade do usuário poder escolher contratar um profissional de dança dentro do mundo virtual e interagir com ele por meio da dança no mundo real.

### 2.3 Dança e aplicativos moveis

Segundo ABREU(1995), as pessoas procuram diversas formas agradáveis de excitação que as permita ao menos “fugir temporariamente da rotina do dia a dia”. A dança possibilita essa fuga momentânea da realidade, devido ao ambiente repleto de pessoas alegres buscando descontração para aliviar a tensão do trabalho, problemas e etc. Além disso, as pessoas gostam de apreender passos novos na dança, o que estimula cada vez mais a pessoa a dançar e a se reinventar na dança, buscando conhecer cada vez mais clubes, pessoas e estilos de dança diferenciados.

Com o aplicativo o serviço de dança e ao mesmo tempo, a diversão, estarão na ponta dos dedos bastando apenas clicar e aguardar alguns segundo por uma experiência de dança superior ao modelo tradicional de sentar na cadeira e aguardar até que apareça alguém que dance de verdade.

### 2.4 Dança e saúde

A dança ao longo dos anos vem sendo considerada como um ferramenta que melhora e preserva a saúde das pessoas em todas as idades. Assim, segundo (FREIRE et al.,2014) a atividade física é definida como qualquer movimento corporal que requer gastos de energia acima dos níveis de repouso.

Segundo dados da (OMS, 2018) cerca de 20% dos adultos e 80% dos adolescentes não praticam exercícios físicos regulares. Dessa forma, o ministério da saúde adotou a recomendação proferida pela OMS, que sugeriu o mínimo de 150



minutos semanais de atividade física moderada, equivalente a uma hora e meia de dança.

## 2.5 Aplicativos móveis

Segundo (LECHETA, 2013), aplicativos móveis são softwares que utilizam de um dispositivo móvel para realizar uma determinada função para o usuário. São programas simples, desenvolvidos para serem baixados no dispositivo móvel do usuário para fazer uma função que facilite a vida das pessoas, com apenas alguns cliques. Algumas aplicações disponíveis para o usuário são gratuitas, mas outras, não. Atualmente, os dispositivos móveis são o maior meio de acesso a internet conforme previu (ANDERSON & RAINIE, 2008).

Os aplicativos móveis poderão auxiliar as pessoas na prática de exercícios físicos saudáveis, se tornando um instrumento essencial na vida das pessoas ao prover a melhora na saúde e qualidade de vida do usuário (TAWALBEH; BAKHADER, 2016).

## 3 FERRAMENTAS E TECNOLOGIAS UTILIZADAS

### 3.1 Plataforma

A plataforma escolhida para esse projeto é a mais utilizada no mundo e tem o maior mercado em franca expansão. Além disso, é uma plataforma de código aberto que desfruta de contribuições da comunidade open source Linux e mais de 500 hardware, software e demais parcerias. Portanto, o projeto será desenvolvido utilizando a plataforma Android Studio.

Ademais, por possuir código aberto, a plataforma Android Studio possui métodos extremamente flexíveis, pois, permite que os desenvolvedores adaptem e evoluam as funcionalidades das aplicações (PEREIRA et al., 2009). Por conseguinte, produtos de softwares livres possuem muitas vantagens: estabilidade, portabilidade para diversas plataformas, livre acesso ao código-fonte, baixo custo e evolução contínua. A continuidade das evoluções permite que, em curto espaço de tempo, novas versões de softwares sejam lançadas com erros das versões anteriores

corrigidas. Assim, o tempo de correção dos erros são menores se comparados com softwares proprietários(Davis et al.,2000).

### 3.2 Parse SDK

O Parse Server é uma versão de código aberto que será utilizado nesse projeto para administrar APIs e serviços de nuvem de forma rápida, segura e com o mínimo de esforço (Parse, 2019). Assim, a API será usada para controlar a sessão de usuários logados e para salvar as imagens enviadas no cadastro, juntamente com os dados do usuário e a URL correspondente ao envio das imagens. Por conseguinte, tanto a URL de retorno quanto os dados do usuário serão armazenados em segurança no Firebase Realtime Database.

### 3.3 Google Maps

Nesse projeto, sera utilizado o SDK do Google Maps para adicionar mapas marcadores, contendo apenas o espaço internos dos clubes de dança e a posição dos Personal\_dancers disponíveis. Assim, Personais\_dancers que estejam atendendo clientes no momento, serão ocultados do mapa até que fiquem disponíveis novamente, ou seja, a modelagem do mapa será configurada de acordo com a necessidade momentânea da aplicação. Portanto, a API gerenciará automaticamente os servidores do Google Maps, a exibição do mapa, download de dados e as respostas as ações dos usuários (GOOGLE DEVELOPERS, 2019).

Para utilizar o serviço do Google Maps, sera necessário acessarmos a plataforma *Google Clound Plataform Console* e seguirmos alguns passos, tais como:

- 1- Criar uma conta de cobrança: essa conta será usada exclusivamente para rastrear todos os custos associados ao projeto;
- 2- Criar um projeto: a criação do projeto será a base para o gerenciamento de serviços, credenciais, cobrança, APIs e SDKs;
- 3- Habilitar a API Maps SDK para Android: a habilitação da API fornecerá a chave de acesso aos serviços da API, que obrigatoriamente deverá ser adicionada à aplicação para a configuração do mapa;

Por conseguinte, para a configuração do mapa é necessário que, pelo menos uma chave de acesso exclusiva, seja adicionada a aplicação. Assim, a chave autenticará as solicitações associadas a aplicação para fins de uso e cobrança.

Para criar uma chave, siga as etapas abaixo:



### 3.4 Firebase

O Firebase é uma plataforma móvel da Google que fornece as ferramentas necessários para desenvolvimento de aplicativos móveis e Web de alta qualidade, que oferece serviços de autenticação, notificação, banco de dados, armazenamento e outros serviços que possam ser integrados a aplicação móvel, possibilitando ampliar a base de usuários e lucrar mais. O Firebase oferece muitos serviços, alguns gratuitos e uteis para quem utiliza poucos serviços de armazenamento, conexões simultâneas, downloads, banco de dados e etc. Porém, conforme for aumentando o numero de usuários e utilização, será necessário um upgrade para plano pago, seja com tarifa mensal ou cobrança por utilização.

Segue abaixo o plano de preços do Firebase para os produtos utilizados nesse projeto em particular, a saber: Authentication, Cloud Messaging e Realtime Database. O Firebase disponibiliza os planos Spark e Blaze. O primeiro não tem custos, mas tem disponibilidade mínimas dos recursos. Já o segundo, é cobrado de acordo com a utilização.

Figura x: Plano de preços Firebase 2022

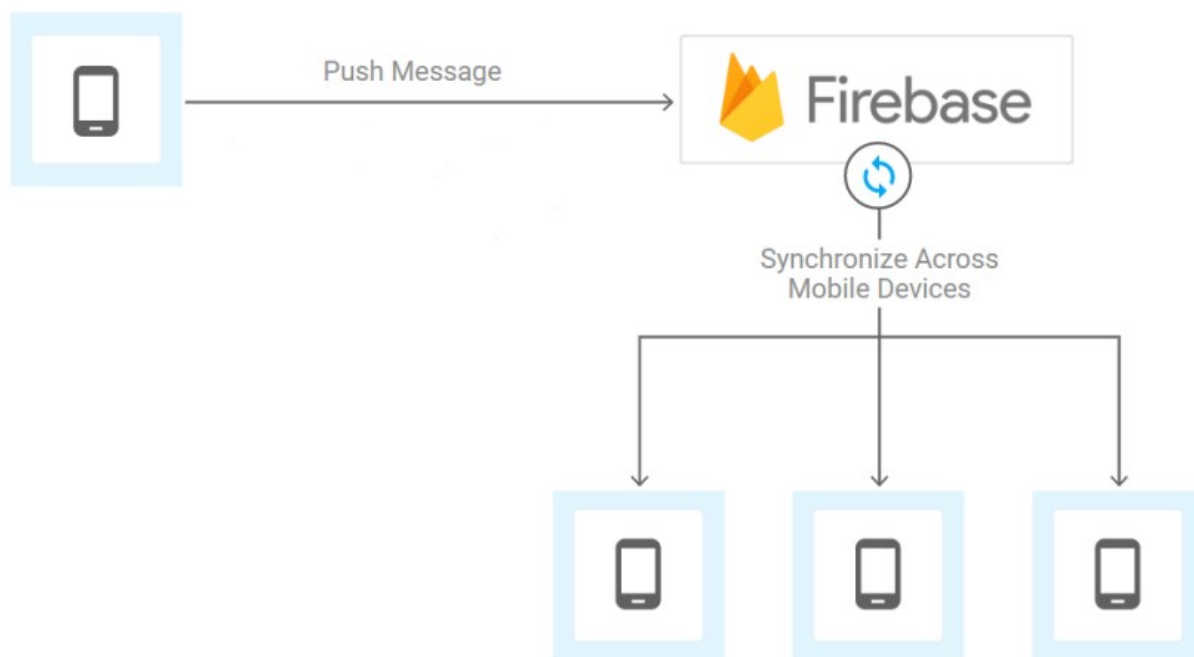
Produtos	Sem custos financeiros Plano Spark Limites generosos para começar	Pagamento por utilização Plano Blaze <a href="#">Calcule o preço de apps em escala</a> ✓ O uso sem custos financeiros do plano Spark está incluído nesta oferta*
<b>Authentication</b>		
Autenticação por telefone - EUA, Canadá e Índia ?	10 mil/mês	US\$ 0,01/verificação
Autenticação por telefone - todos os outros países ?	10 mil/mês	US\$ 0,06/verificação
Outros serviços de autenticação	✓	✓
<b>Cloud Messaging (FCM)</b>	Sem custos financeiros	
<b>Realtime Database</b>		
Conexões simultâneas ?	100	200 mil/banco de dados
GB de armazenamento	1 GB	\$5/GB
GB de download	10 GB/mês	\$1/GB
Vários bancos de dados por projeto	✗	✓

### 3.5 Firebase Realtime Database

A ferramenta Firebase Realtime Database (FRD), é um banco de dados em tempo real hospedado na nuvem, que será utilizado para a persistência de dados da aplicação, ou seja: dados do Personal\_dancer, Cliente\_dancer e Clube (dados pessoais, nota de avaliação, localização do clube) e das contratações de Personal\_dancer em andamento (localização do Clube, valor total, Personal\_dancer, cliente\_dancer, chat, estilo de dança, tempo de contratação) e etc. O Realtime Database é um banco de dados NoSQL que sincroniza todos os usuários em tempo real, até mesmo quando o aplicativo está off-line. Sendo assim, a ferramenta é indicado para estados dinâmicos que exigem processos rápidos o suficiente para que o resultado retorne e seja acionado em tempo real, sincronizado com todos os dispositivos que compartilham da mesma instância. Por isso, mesmo que a conexão fique off-line, os dados locais são recuperados e atualizados instantaneamente após a conexão ser restabelecida, pois, os dados são armazenados no servidor local e mesmo off-line os eventos podem ser acionados a exemplo de contratações e

pagamento com cartão de crédito. Logo que a conexão seja reestabelecida, o Realtime Database sincroniza e atualiza em milissegundos os eventos que ocorreram enquanto o cliente estava off-line.

Figura 7 – Firebase Realtime Database



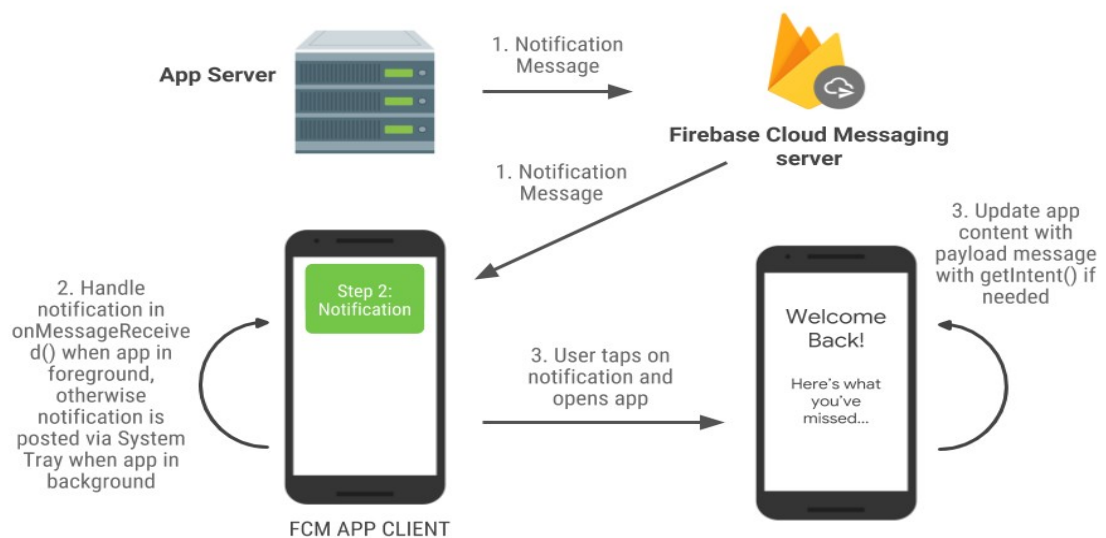
Fonte: Google Cloud (GOOGLE CLOUD, 2019)

### 3.6 Firebase Cloud Messaging

O Firebase Cloud Messaging (FCM), é uma solução de mensagens entre plataformas que permite o envio confiável de notificações sem custo (SHI 2018). Isso facilita as interações com as aplicações conectadas através de envio de notificações aos usuários ou mensagens de dados que determinam em tempo real o andamento da aplicação, ou seja, se um cliente solicita um Personal Dancer o Personal receberá a mensagem, decidirá se aceita ou não, ao escolher aceitar, será enviado uma notificação ao cliente informando o seguinte: 'Personal Dancer Anselmo com especialidade em forró e samba', aceitou a contratação e irá recebê-la em 2 minutos. Outra característica que pode ser melhor aproveitada é o envio de mensagens para grupo de dispositivos. Nesse sentido, após o crescimento do aplicativo, poderemos administrar melhor os eventos em benefício dos clientes e Personais da dança. Por conseguinte, no presente projeto será usado o Firebase

Cloud Messaging para enviar notificações aos Personais Dancer informando-os sobre: status do cadastro, uma nova solicitação de contratação e encerramento da contratação. A notificação sobre uma nova solicitação de contratação de Personal Dancer é realizada através do serviço de segmentação versátil de mensagem que permite o envio obrigatório de mensagens simultâneas para todos os Personais Dancers que estiverem disponíveis e portanto, aptos a receber um novo cliente.

Figura 4: Firebase Cloud Messaging



Fonte: Firebase (SHI, 2018)

### 3.7 API Cielo eCommerce

O API Cielo eCommerce é uma evolução do webservice 1.5 que foi descontinuado. O API pode acessar rotinas de validação de cartão de crédito, endereço postal(CEP) e acesso ao banco de dados. Para poder ser executado pela aplicação Android, o API utiliza modelo Rest, que é um estilo de arquitetura que envolve manipulações de dados, composto por dois papéis: cliente e servidor. O API Cielo eCommerce utiliza quatro formas de pagamento a saber: cartões de crédito, cartões de débito, boletos e transferência eletrônica. Para esse projeto, por questões de segurança, usaremos apenas a opção cartão de crédito, com as seguintes bandeiras: visa, mastercard, america express, elo, aura, dinners, discovery e JCB. A API Cielo eCommerce oferece as seguintes funcionalidade: integração simplificada - Utilizando um API REST, substituindo o envio de XMLs para a autorização de pagamentos. Além disso, aceita varias formas de pagamentos, sem necessidade de armazenar dados do cartão, tudo isso de forma segura e de fácil acesso, por meio da tokenização de cartões. O API possui a função Zero Auth que permite testar o cartão para verificar ou validar se o cartão se encontra apto para realizar uma determinada transação, bem como, consulta de 'Bins' para validar o cartão se é do tipo crédito ou débito. Por fim, a API Cielo eCommerce possui a função antifraude que eleva a segurança do aplicativo ao analisar os risco das transações de cartões de credito.

Não irei entrar em mais detalhes pois o API será implementado no futuro e portanto, até lá, poderá surgir opções melhores e mais acessíveis economicamente. (<https://developercielo.github.io/en/manual/cielo-ecommerce>).

## 4 DESENVOLVIMENTO

### 4.1 Descrição do mini mundo detalhado do sistema

Trata-se de uma descrição formal da realidade a ser representada pelo aplicativo chamado Personal\_dancer. Assim, o objetivo principal do projeto é desenvolver um aplicativo que permita o cadastro de dois tipos de clientes: Personal Dancers e Cliente Dancer. O Personal Dancer prestara serviços da dança e o Cliente Dancer contratará os serviços do Personal. Para isso, ambos deverão estar

devidamente cadastrados. Assim, o Personal Dancer deverá cadastrar o nome completo, CPF, RG, endereço, nº celular, peso, altura, sexo, raça, nacionalidade e estilos de dança que gostaria de ensinar. O Cliente Dancer deverá cadastrar o nome completo, CPF, RG, endereço, nº celular, peso, altura, sexo, raça, nacionalidade, estilo de dança que gostaria de praticar e cadastrar um número de cartão de crédito válido que deverá ser igual ao seu nome de registro. O cliente poderá contratar o Personal Dancer se ambos estiverem cadastrados. O cliente poderá contratar o Personal se ambos estiverem no mesmo local e num raio de 20 metros a contar do centro da pista. O cliente poderá contratar o Personal se ambos estiverem com o dispositivo de geolocalização ativo. Ao fim do período contratado o cliente terá preferência na recontração do Personal e para isso, terá 5 minutos a contar do fim do período contratado para solicitar nova contratação. Se ao fim desse prazo o cliente não renovar a contratação, o Personal Dancer estará automaticamente disponível para outro cliente, podendo ser contratado qualquer momento. O Personal Dancer pode aceitar ou cancelar uma contratação. Quando o Personal cancelar uma solicitação de contratação, o cliente ficará disponível para outro Personal Dancer. Se o cliente não se adaptar ao estilo de dança, a estrutura corporal e aos modos do Personal poderá cancelar a contratação. Se o cliente cancelar a contratação após 5 minutos do início do período contratado, o aplicativo disponibilizará até 5 Personais Dancers caso tenha disponibilidade no momento para o cliente escolher e, em hipótese alguma, reembolsará. O cliente poderá contratar um Personal Dancer se ambos estiverem no mesmo Clube e se o Clube for um dos clubes cadastrados pelo aplicativo. Um cliente só pode cadastrar um CPF que é seu código identificador e, portanto, só poderá ter uma conta. Um cliente só pode contratar um Personal Dancer por vez. Um Personal Dancer só pode aceitar um cliente por vez. No fim do período contratado o cliente pode contratar outro Personal e da mesma forma, um Personal Dancer pode aceitar a contratação de outro cliente. O cliente poderá visualizar seu histórico de contratação. O Personal poderá visualizar seu histórico de contratação recebidas e aceitas, junto com o total ganho no dia subtraído das taxas. O cliente poderá visualizar o recibo da contratação. O cliente poderá avaliar o Personal Dancer. O Personal Dancer poderá avaliar o cliente. O cliente poderá avaliar o Clube de dança. O Personal poderá avaliar o Clube de dança. Personal Dancer bem avaliado receberá mais clientes com

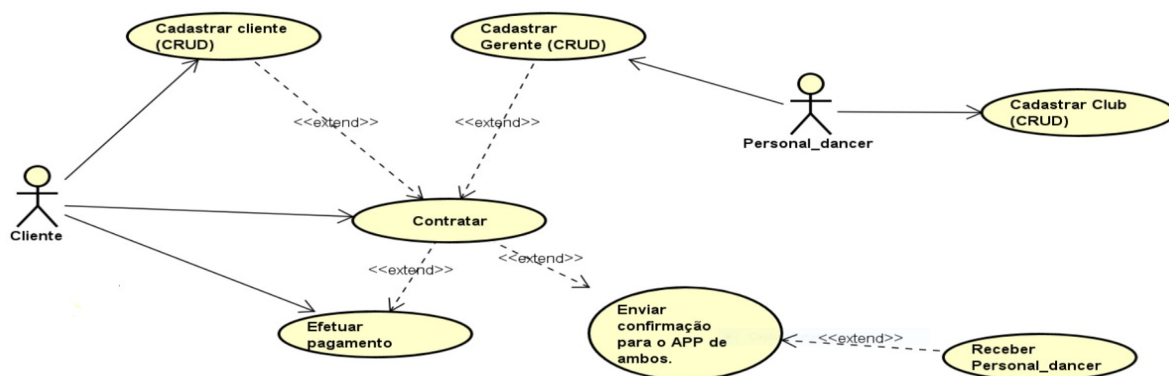


boa avaliação. Clientes bem avaliados terão mais chances da solicitação de contratação ser aceita pelo Personal Dancer.

## 4.2 Requisitos do Sistema Personal Dancer

Nessa parte, falarei sobre os requisitos de software e elaborarei descrições claras sobre o que o App deve ser capaz de fazer, através do diagrama de casos de uso, requisitos funcionais e não funcionais.

### 4.2.1 Diagramas de casos de uso



### 4.2.2 Requisitos Funcionais

Segundo (SOMMERVILLE,2011) “requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir, a entradas específicas e de como o sistema deve se comportar em determinadas situações<sup>1</sup>”. Assim, os requisitos do sistema, variam do requisito geral até o mais específico e descrevem em detalhes as funções do sistema, “suas entradas, saídas e exceções”<sup>2</sup>.

- Um cliente poderá contratar um Personal Dancer por vez, desde que ambos estejam cadastrados, logados, com o GPS ativo e no mesmo clube de dança.
- Não será permitido os usuário ‘Cliente’ pesquisar por usuário ‘Personal’ e da mesma forma, usuário ‘Personal’ pesquisar por usuário ‘Cliente’. Porém, quando o Cliente clicar em contratar Personal Dancer e o Personal aceitar a contratação será

<sup>1</sup> Sommerville. Pag 59.

<sup>2</sup> Sommerville. Pag 59.

mostrado aos dois usuários o nome, idade, altura, sexo, peso, raça, estilos de dança, período contratado e uma foto de perfil e outra inteira. Lembrando que o Cliente pode cancelar a contratação a qualquer momento e que o Personal tem até 3 minutos para se apresentar ao cliente, e se estiverem de acordo, o Personal Dancer poderá clicar em iniciar dança. Após cancelado, aquele Personal Dancer não ficará mais disponível para o Cliente naquele dia.

- O sistema deve fornecer ao cliente o histórico de contratações detalhada, com dia, horário, tempo de contratação, nome do Personal Dancer, nome do Cliente, nome do clube de dança, estilo de dança do Personal, estilo de dança do cliente, valores hora/dança e valores totais do dia.

As especificações dos requisitos funcionais de um sistema deve ser completa e consistente(SOMMERVILLE,2011), ou seja, deve ser definido todos os serviços que o usuário possa requerer por meio de cliques, desde o cadastro, login, alteração, exclusão, saída, contratação, cancelamento, aceitação da contratação, início da contratação, e finalização da contratação. Por conseguinte, os requisitos funcionais são definidos tanto para o cliente quanto para o Personal, e, em alguns momentos serão diferentes e em outros semelhantes. Porém, em nenhuma hipótese poderão ser contraditórios, mas sim consistentes, pois, o 'Contratar Personal Dancer' do Cliente é diferente do 'Aceitar contratação' do Personal e da mesma forma, o 'Cancelar' do Cliente é diferente do 'Finalizar' do Personal.

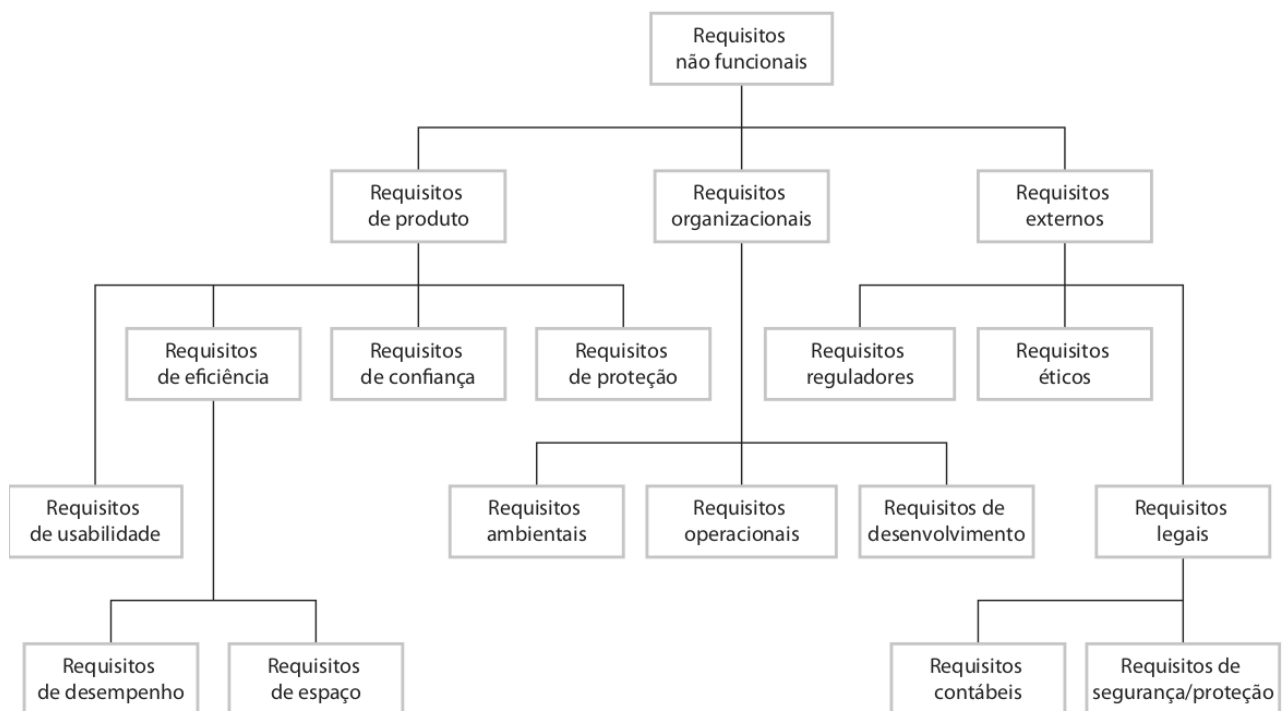
#### 4.2.3 Requisitos não funcionais

Os requisitos não funcionais, conforme o nome sugere, não estão relacionados aos serviços específicos disponibilizados pelo aplicativo a seus usuários (SOMMERVILLE,2011). Eles estão relacionados a outras propriedades importantes para o bom funcionamento do aplicativo, mas que são consideradas secundárias devido a ordem de importância, a exemplo da confiabilidade, disponibilidade, tempo de resposta e ocupação de área.

O requisito disponibilidade define quando o aplicativo deve estar disponível, no caso do aplicativo Personal Dancer, as quarta-feira, quinta-feira e sexta-feira das 18 horas a meia noite, nos sábados das 20 horas até as 4 horas da manhã(ou de acordo com o horário do estabelecimento) e aos domingo, das 18 horas a meia noite.

Os requisitos de desempenho estão relacionadas ao tempo de resposta ou a rapidez que o aplicativo executa determinada ação e quanto de memória é necessário para que uma ação seja executada.

Os requisitos de confiabilidade está associada a um conjunto de métricas de confiabilidade aceitáveis, definidas pela equipe de desenvolvimento que ‘estabelecem uma taxa aceitável de falhas’(SOMMERVILLE,2011).Mais abaixo, mostraremos uma tabela com os tipos de requisitos não funcionais.



*Figura y: Tipos de requisitos não funcionais*

#### 4.3 Caso de uso

Casos de uso, conforme as descrições dos manuais, “é uma técnica de descoberta de requisitos(JACOBSON et al., 1993), muito utilizada na linguagem de modelagem unificada(UML). Resumidamente, um caso de uso “identifica atores envolvidos em uma interação e dá nome ao tipo de interação”(SOMMERVILLE,2011). Atores podem ser tanto pessoas quanto outros sistemas. As informações coletadas no levantamento de requisitos poderão descrever as interações do sistema em forma de textos ou com modelos gráficos, a exemplo de diagramas de sequencia ou de estado.

#### 4.3.1 Caso de uso – Cadastrar cliente

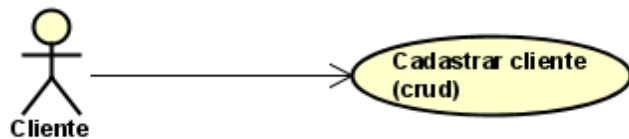


Figura 1: RF01- Cadastrar cliente

ESPECIFICAÇÕES DE CASO DE USO			
<b>CASO DE USO:</b> Cadastrar Cliente			
<b>DESCRIÇÃO:</b> Permitir o cadastro, alteração e exclusão dos dados			
<b>ATOR:</b> Cliente			
<b>FLUXO DE EVENTOS:</b> O caso de uso começa quando o ator seleciona a opção cadastrar cliente			
<b>PRÉ-CONDIÇÃO:</b> Não existe pré-condição			
<b>PÓS-CONDIÇÃO:</b> Cliente cadastrado na base de dados			
<b>PRIORIDADE DE CASO DE USO:</b> ✓ Primário                      Secundário                      Desejável			
<b>FLUXO BÁSICO (INSERIR):</b>			
1 O ator informa os dados do cliente			
2 O ator clica em salvar os dados			
3 O sistema exibe a mensagem MSG01			
<b>FLUXO ALTERNATIVO (ALTERAR):</b>			
1 O ator abre o registro para edição			
2 O ator informa os dados do cliente a serem alterados			
3 O ator clica em salvar os dados			
4 O sistema verifica a RN03			
4.1 Se a RN03 é válida: o sistema exibe a mensagem MSG05			
a- O sistema altera o registro			
4.2 Se a RN03 não é válida: o sistema exibe a mensagem MSG06			
<b>FLUXO ALTERNATIVO (EXCLUIR/INATIVAR)</b>			
1 O ator abre o registro para edição			
2 O ator clica em excluir o registro			
3 O ator clica em salvar os dados			
4 O sistema verifica a RN04			
4.1 Se a RN04 é válida: o sistema exibe a mensagem MSG03			
a- O sistema inativa o registro			
4.2 Se a RN04 não é válida: o sistema exibe a mensagem MSG04			
<b>FLUXO DE EXCEÇÃO:</b> O sistema exibe a mensagem MSG00			

#### REGRAS DE NEGOCIO DO CASO DE USO:

RN01 – Validar tipo de usuário

RN02 – Verificar se cliente já está cadastrado

RN03 – Só poderá ser cadastrado Cliente e Personal maior de 18 anos.

RN04 – Verifica se não foram alterados campos que não podem ser alterados

RN05 – Verifica se não foram excluídos campos que não podem ser excluídos

RN06 – Validar cartão de crédito

RN07 – Cliente tem preferência na recontração do mesmo Personal

#### MENSAGENS DE CASO DE USO:

MSG01 – Registro salvo com sucesso

MSG02 – Registro não encontrado

MSG03 – Registro inativado com sucesso

MSG04 – Não é possível excluir/inativar esse registro

MSG05 – Dados alterados com sucesso

MSG06 – Não é possível alterar esse registro

MSG00 – O sistema apresentou um erro. Contate o administrador

Tabela 01- Cadastrar Cliente

#### 4.3.2 Caso de uso Cadastrar Personal

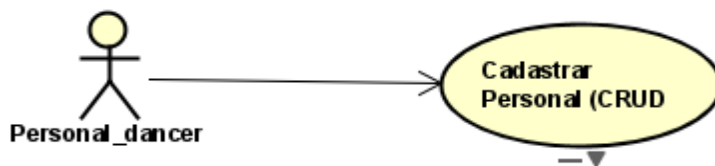


Figura 2: RF02 -Cadastrar Personal

#### ESPECIFICAÇÕES DE CASO DE USO

**CASO DE USO:** Cadastrar Personal

**DESCRIÇÃO:** Permitir o cadastro, alteração e exclusão dos dados

**ATOR:** Personal Dancer

**FLUXO DE EVENTOS:** O caso de uso começa quando o ator seleciona a opção cadastrar Personal Dancer

**PRÉ-CONDIÇÃO:** Não existe pré-condição

**PÓS-CONDIÇÃO:** Personal Dancer cadastrado na base de dados

**PRIORIDADE DE CASO DE USO:** ✓ Primário      Secundário      Desejável

##### FLUXO BÁSICO (INSERIR):

1 O ator informa os dados do Personal Dancer

2 O ator clica em salvar os dados

3 O sistema exibe a mensagem MSG01

##### FLUXO ALTERNATIVO (ALTERAR):

1 O ator abre o registro para edição

2 O ator informa os dados do Personal a serem alterados

3 O ator clica em salvar os dados

4 O sistema verifica a RN03

4.1 Se a RN03 é válida: o sistema exibe a mensagem MSG05

- a- O sistema altera o registro
- 4.2 Se a RN03 não é válida: o sistema exibe a mensagem MSG06

#### **FLUXO ALTERNATIVO (EXCLUIR/INATIVAR)**

- 1 O ator abre o registro para edição
- 2 O ator clica em excluir o registro
- 3 O ator clica em salvar os dados
- 4 O sistema verifica a RN04
  - 4.1 Se a RN04 é válida: o sistema exibe a mensagem MSG03
    - a- O sistema inativa o registro
  - 4.2 Se a RN04 não é válida: o sistema exibe a mensagem MSG04

**FLUXO DE EXCEÇÃO:** O sistema exibe a mensagem MSG00

#### **REGRAS DE NEGOCIO DO CASO DE USO:**

- RN01 – Validar tipo de usuário
- RN02 – Verificar se cliente já está cadastrado
- RN03 – Só poderá ser cadastrado Cliente e Personal maior de 18 anos.
- RN04 – Verifica se não foram alterados campos que não podem ser alterados
- RN05 – Verifica se não foram excluídos campos que não podem ser excluídos
- RN06 – Validar cartão de crédito
- RN07 – Cliente tem preferência na recontratação do mesmo Personal

#### **MENSAGENS DE CASO DE USO:**

- MSG01 – Registro salvo com sucesso
- MSG02 – Registro não encontrado
- MSG03 – Registro inativado com sucesso
- MSG04 – Não é possível excluir/inativar esse registro
- MSG05 – Dados alterados com sucesso
- MSG06 – Não é possível alterar esse registro
- MSG00 – O sistema apresentou um erro. Contate o administrador

Tabela 2- Cadastrar Personal Dancer

#### 4.3.3 Caso de uso Contratar Personal

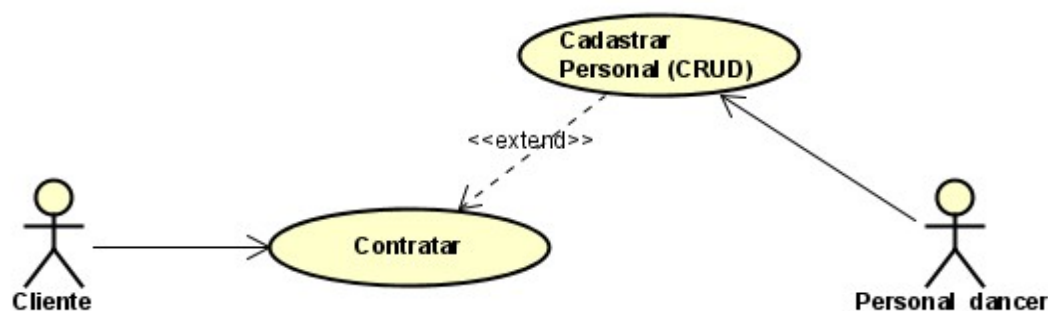


Figura 3: RF03 - Contratar Personal

### **ESPECIFICAÇÕES DE CASO DE USO**

**CASO DE USO:** Contratar Personal

**DESCRIÇÃO:** Permite inserir a contratação de um Personal Dancer no sistema

**ATOR:** Cliente, Personal Dancer

**FLUXO DE EVENTOS:** O caso de uso inicia quando o ator acessa a pagina de contratação Personal Dancer

**PRÉ-CONDIÇÃO:** O cliente precisa estar cadastrado, o Personal Dancer precisa estar cadastrado e disponível

**PÓS-CONDIÇÃO:** Ambos devem estar no mesmo clube.

**PRIORIDADE DE CASO DE USO:** ✓ Primário      Secundário      Desejável

**EXTENSÕES DO CASO DE USO:**

1 Para o ator cliente: Cadastrar cliente, Cadastrar cartão de credito válido, Receber confirmação no APP

2 Para o ator Personal: Cadastrar Personal, Receber confirmação no APP

**INCLUSÕES DO CASO DE USO**

1 Não existe

**FLUXO PRIMÁRIO**

1 O sistema valida a RN01.

1.1 Se ator for o cliente, e a regra RN02 não for válida.

i- Enviar o ator para o caso de uso RN01.

2 O sistema valida a localização do Personal e do Cliente

2.1 Se ambos estiverem dentro do mesmo Clube

i- Será exibida a tela de contratação

2.2 Se não: o sistema exibe mensagem MSG13 ou MSG08

3 O ator cliente clica em contratar Personal

3.1 Uma solicitação de contratação é enviada ao Personal

4 O ator Personal aceita a contratação

4.1 Se o ator Personal aceitar a contratação

i- O sistema exibe a mensagem MSG07

4.2 Se nenhum ator Personal aceitar a contratação

i- O sistema exibe a mensagem MSG08

5 O ator Personal se apresenta ao cliente

5.1 Se os atores estiverem de acordo

i- O ator Personal clica em iniciar dança e o sistema exibe mensagem MSG09

5.2 Se não estiverem de acordo

i- O ator Cliente clica em Cancelar dança e o sistema exibe mensagem MSG10

6 O sistema executa o caso de uso Atualizar Status

7 Encerra o caso de uso.

**FLUXO DE EXCEÇÃO:** O sistema exibe a mensagem MSG00

**REGRAS DE NEGOCIO DO CASO DE USO:**

RN01 – Validar tipo de usuário

RN02 – Verificar se cliente já está cadastrado

RN03 – Só poderá ser cadastrado Cliente e Personal maior de 18 anos.

RN04 – Verifica se não foram alterados campos que não podem ser alterados

RN05 – Verifica se não foram excluídos campos que não podem ser excluídos

RN06 – Validar cartão de crédito

RN07 – Cliente tem preferência na recontração do mesmo Personal

**REGRAS DE NEGOCIO DO SISTEMA:**

**RNS01** – O cliente só poderá contratar o Personal se ambos estiverem dentro do mesmo clube e com o GPS ativo.

**MENSAGENS DE CASO DE USO:**

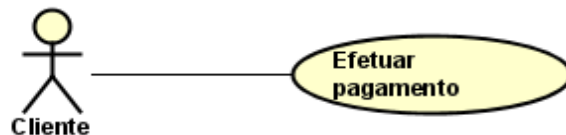
MSG01 – Registro salvo com sucesso

- MSG02 – Registro não encontrado
- MSG03 – Registro inativado com sucesso
- MSG04 – Não é possível excluir/inativar esse registro
- MSG05 – Dados alterados com sucesso
- MSG06 – Não é possível alterar esse registro
- MSG07 – Personal a caminho do cliente
- MSG08 – Não há Personais Disponível
- MSG09 – Dança iniciada, iniciou a contagem do período contratado
- MSG10 – Contratação Cancelada. Procurando novos Personais disponíveis.
- MSG00 – O sistema apresentou um erro. Contate o administrador
- MSG11 – Preenchimento incorreto do numero do cartão ou cartão inválido.
- MSG12 – Sem credito ou saldo suficiente
- MSG13 – Localização inválida- Selecione o clube que está e tente novamente

Tabela 3 – Contratar Personal

---

#### 4.3.4 Caso de uso Efetuar pagamento



*Figura 4: RF04 -Efetuar Pagamento*



## ESPECIFICAÇÕES DE CASO DE USO

**CASO DE USO:** Efetuar Pagamento

**DESCRIÇÃO:** Permite inserir um número de cartão de crédito válido

**ATOR:** Cliente

**FLUXO DE EVENTOS:** O caso de uso inicia quando o ator cliente, devidamente logado e validado, acessa o formulário de dados pessoais para completar o cadastro e inserir um número de cartão de crédito

**PRÉ-CONDIÇÃO:** Estar logado no App

**PÓS-CONDIÇÃO:** Se o ator estiver cadastrando, receberá mensagem MSG14 ou MSG11 ou ainda MSG12. Se o ator estiver contratando, somente receberá confirmação após o encerramento da contratação com mensagem do período contratado e o valor total

**PRIORIDADE DE CASO DE USO:** ✓ Primário      Secundário      Desejável

**EXTENSÕES DO CASO DE USO:**

1 Para o ator cliente: Após o cartão ser debitado, receber confirmação no App

**INCLUSÕES DO CASO DE USO**

1 Não existe

**FLUXO PRIMÁRIO**

1 O ator preenche o campo do formulário com o número do cartão de crédito

2 O ator clica em salvar

3 O sistema envia mensagem de confirmação MSG14

4 O sistema abre a página de contratação

5 O ator poderá contratar um Personal a qualquer momento.

6 Fim do caso de uso.

**FLUXO ALTERNATIVO NÚMERO DE CARTÃO INVÁLIDO**

1 O ator preenche o campo do formulário com o número do cartão de crédito

2 O ator clica em salvar

3 O sistema envia mensagem de confirmação MSG11

4 O sistema retorna para a página anterior.

5 O ator poderá digitar novamente o numero do cartão

6 Fim do caso de uso.

**FLUXO ALTERNATIVO SEM SALDO SUFICIENTE**

1 O ator preenche o campo do formulário com o número do cartão de crédito

2 O ator clica em salvar

3 O sistema envia mensagem de confirmação MSG12

4 O sistema retorna para a página anterior.

5 O ator poderá digitar outro número de cartão de crédito

6 Fim do caso de uso.

**FLUXO DE EXCEÇÃO:** O sistema exibe a mensagem MSG00

**REGRAS DE NEGOCIO DO CASO DE USO:**

RN01 – Validar tipo de usuário

RN02 – Verificar se cliente já está cadastrado

RN03 – Só poderá ser cadastrado Cliente e Personal maior de 18 anos.

RN04 – Verifica se não foram alterados campos que não podem ser alterados

RN05 – Verifica se não foram excluídos campos que não podem ser excluídos

RN06 – Validar cartão de crédito

RN07 – Cliente tem preferência na recontração do mesmo Personal

**REGRAS DE NEGOCIO DO SISTEMA:**

RNS01 – O cliente só poderá contratar o Personal se ambos estiverem dentro do mesmo clube e com o GPS ativo.

**MENSAGENS DE CASO DE USO:**

MSG01 – Registro salvo com sucesso

MSG02 – Registro não encontrado

MSG03 – Registro inativado com sucesso

MSG04 – Não é possível excluir/inativar esse registro

MSG05 – Dados alterados com sucesso

MSG06 – Não é possível alterar esse registro

MSG07 – Personal a caminho do cliente

MSG08 – Não há Personais Disponível

MSG09 – Dança iniciada, iniciou a contagem do período contratado

MSG10 – Contratação Cancelada. Procurando novos Personais disponíveis.

MSG00 – O sistema apresentou um erro. Contate o administrador

MSG11 – Preenchimento incorreto do numero do cartão ou cartão inválido.

MSG12 – Sem credito ou saldo suficiente

MSG14 – Cartão de crédito cadastrado com sucesso

MSG13 – Localização inválida- Selecione o clube que está e tente novamente

Tabela 4 Efetuar pagamento

#### 4.3.5 Caso de Uso Receber Personal Dancer

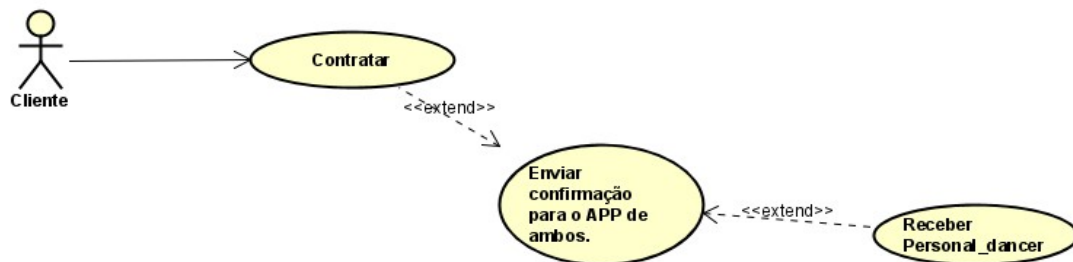


Figura 5: RF05 -Receber Personal Dancer

## ESPECIFICAÇÕES DE CASO DE USO

**CASO DE USO:** Receber Personal Dancer

**DESCRIÇÃO:** Permitir que o cliente receba o Personal

**ATOR:** Cliente

**FLUXO DE EVENTOS:** Fluxo de evento começa quando o Personal Dancer aceita a contratação do Cliente e tem de 3 minutos a cinco minutos para se apresentar ao Cliente que está o aguardando, e que o receberá

**PRÉ-CONDIÇÃO:** Não há

**PÓS-CONDIÇÃO:** Se ambos estiverem em comum acordo, logo após o Personal se apresentar ao cliente e ser recebido, O Personal poderá clicar em Iniciar Dança. Instantaneamente, ambos receberão uma mensagem de confirmação do inicio e termino do período contratado e iniciará a contagem do tempo

**PRIORIDADE DE CASO DE USO:** ✓ Primário      Secundário      Desejável

**EXTENSÕES DO CASO DE USO:**

Para o ator Cliente e ator Personal: O sistema deve enviar confirmação do inicio e termino do período contratado.

**INCLUSÕES DO CASO DE USO**

1 Não existe

**FLUXO PRIMÁRIO**

1 O ator cliente envia uma solicitação de contratação ao Personal

2 O Personal disponível no momento clica em aceitar contratação

3 O sistema envia MSG15

4 O Personal se apresenta e ambos conversam

5 Se ambos estiverem de acordo

5.1 O ator Personal clica em Iniciar Dança e o sistema exibe mensagem MSG09

i- Inicia a contagem do tempo contratado

ii- O ator Personal muda status para indisponível

5.2 Se não estiverem de acordo

i- O ator Cliente clica em Cancelar dança e o sistema exibe mensagem MSG10

6 O sistema executa o caso de uso Atualizar Status

6.1 O ator Personal muda o status para disponível

6.2 O ator cliente muda o status para Aguardando

7 Fim do caso de uso.

**FLUXO DE EXCEÇÃO:** O sistema exibe a mensagem MSG00

**REGRAS DE NEGOCIO DO CASO DE USO:**

RN01 – Validar tipo de usuário

RN02 – Verificar se cliente já está cadastrado

RN03 – Só poderá ser cadastrado Cliente e Personal maior de 18 anos.

RN04 – Verifica se não foram alterados campos que não podem ser alterados

RN05 – Verifica se não foram excluídos campos que não podem ser excluídos

RN06 – Validar cartão de crédito

RN07 – Cliente tem preferência na recontração do mesmo Personal

**REGRAS DE NEGOCIO DO SISTEMA:**

**RNS01** – O cliente só poderá contratar o Personal se ambos estiverem dentro do mesmo clube e com o GPS ativo.

**MENSAGENS DE CASO DE USO:**

MSG01 – Registro salvo com sucesso

MSG02 – Registro não encontrado

MSG03 – Registro inativado com sucesso  
 MSG04 – Não é possível excluir/inativar esse registro  
 MSG05 – Dados alterados com sucesso  
 MSG06 – Não é possível alterar esse registro  
 MSG07 – Personal a caminho do cliente  
 MSG08 – Não há Personais Disponível  
 MSG09 – Dança iniciada, iniciou a contagem do período contratado  
 MSG10 – Contratação Cancelada. Aguardando novos Personais disponíveis.  
 MSG00 – O sistema apresentou um erro. Contate o administrador  
 MSG11 – Preenchimento incorreto do numero do cartão ou cartão inválido.  
 MSG12 – Sem credito ou saldo suficiente  
 MSG13 – Localização inválida- Selecione o clube que está e tente novamente  
 MSG14 – Cartão de crédito cadastrado com sucesso  
 MSG15 – Personal aceitou contratação e está a caminho  
 MSG16–

Tabela 5 Receber Personal

---

#### 4.4 Requisitos não Funcionais

Os requisitos não funcionais fazem referência a todo tipo de restrições de desenvolvimento, relacionados com a interface com usuário, segurança, confiabilidade, disponibilidade, manutenibilidade, portabilidade, tempo de resposta, ocupação de área, desempenho, tecnologias envolvidas, linguagem de programação, banco de dados e plataforma de desenvolvimento.

##### 4.4.1 Usabilidade

A usabilidade descreve os requisitos não funcionais associados à facilidade de uso da interface com o usuário, material de treinamento e documentação do sistema. A interface é de vital importância para qualquer sistema, sejam elas utilizadas diariamente ou não. Pensando nisso, o sistema terá uma interface amigável e intuitiva aos iniciantes, sem ser cansativa aos mais avançados. Nesse sentido, a usabilidade é a qualidade que caracteriza o uso de programas e aplicações. Segundo a ISO 9241, “usabilidade é a capacidade que um sistema interativo oferece a seu usuário em determinado contexto da operação, para a realização de tarefas de maneira eficaz, eficiente e agradável”. (CYBIS;BETIOL;FAUST, 2007).

#### 4.4.2 NF01 Usabilidade e interface

A interface gráfica do sistema deve ser intuitiva, de forma que o cliente não precise pensar muito para se cadastrar no sistema e contratar um Personal. Como nos diz a norma ISO 9241-11, existem três requisitos essenciais para medir o grau de satisfação de um usuário. Assim, quando um sistema se propõe a cadastrar, logar, alterar, excluir, contratar Personal, cancelar contratação, iniciar período contratado, finalizar e etc, podemos dizer que um sistema é eficaz quando o usuário, sem muito esforço, consegue realizar o objetivo proposto pela aplicação. A eficiência está relacionada com a quantidade de recursos necessários para realizar um objetivo. Assim, quanto mais recursos, menos eficiente é a aplicação. Por conseguinte, quanto menos cliques no aplicativo para cadastrar o usuário, mais eficiente será. Por fim, o grau de satisfação do usuário está atrelado ao uso da aplicação e ao modo de desenvolvimento da interface. Pois, uma interface com uma boa performance carrega rápido as páginas e a aplicação. Ao contrário disso, uma interface ruim, demora a carregar e consequentemente, atrapalha o usuário que perde tempo ao esperar o carregamento completo do sistema.

**Prioridade:**    ☒ Essencial        ☐ Importante        ☐ Desejável

#### 4.4.3 NF02 Usabilidade e tipo de usuário

A interface do sistema deve ser o mais adaptável possível ao tipo de cliente logado, pois, teremos dois tipos de usuários: usuário Cliente e usuário Personal. Assim, o usuário Cliente contratará o Personal e o usuário Personal aceitará ser contratado pelo Cliente. Poderemos ter Clientes que acessam a internet esporadicamente e Clientes aposentados com dificuldades em se cadastrar ou até mesmo, dificuldades para contratar o Personal Dancer. Pensando nisso, o sistema será projetado para tolerar e evitar erros de entradas de usuário ou seja, se o usuário estiver com dificuldades para logar no App e errar a senha, receberá uma mensagem dizendo: “senha incorreta, digite novamente” ou “você digitou caracteres a mais, tente novamente” ou ainda, “sua senha tem seis caracteres, digite novamente”. Portanto, “o sistema, na medida do possível, deve detectar os erros e, então, corrigi-los automaticamente ou solicitar ao usuário a reentrada de seus

dados”(SOMMERVILLE, 2011. A curto prazo, pensando em otimizar definitivamente o acesso ao aplicativo, iremos exigir o cadastro das digitais do cliente no aplicativo, para reduzir ainda mais a quantidade de toques na tela.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

#### 4.4.4 Confiabilidade

A confiabilidade descreve os requisitos não funcionais associados à frequência, severidade de falhas do sistema e habilidade de recuperação dos mesmos, bem como à corretude do sistema. De modo geral, a confiabilidade do sistema depende da confiabilidade de seus componentes, mas interações inesperadas, acidental ou não, podem causar novos tipos de falhas e, portanto, afetar a confiabilidade do sistema (SOMMERVILLE, 2011).

#### 4.4.5 NF03 Confiabilidade e Segurança da Informação

Os atributos disponibilidade, confiabilidade, segurança e proteção estão associados ao termo “Confiança”(SOMMERVILLE, 2011). Nesse sentido, a confiança é mais importante que as funcionalidades detalhadas do sistema, pois, por razões óbvias, uma falha no sistema pode afetar inúmeras pessoas por causa de sua indisponibilidade momentânea. Por conseguinte, muitos usuários rejeitam sistemas ao perceber que não são confiáveis e que portanto, podem prejudicar os negócios da empresa. Além disso, sistemas não confiáveis causam perda de informações, e, informações coletadas, armazenadas e mantidas no banco de dados tem um custo altíssimo a empresa. Caso os dados sejam perdidos ou corrompidos, os custo de recuperação serão muito alto, somados a indisponibilidade dos dados e do sistema.

O cadastro no App permite a inclusão, alteração e remoção de dados. Por isso, é essencial que as informações do banco de dados permaneçam lá de forma consistente após uma inserção, atualização e exclusão de dados. No caso de exclusão da conta, ficará no servidor por no máximo 30 dias, após esse prazo, será excluída de vez do servidor e os clientes que queiram utilizar o aplicativo, deverão se cadastrar novamente. Do mesmo modo, cadastro, login e senha ficarão armazenados no servidor por 60 dias e após, excluídas. Se o cliente excluir

acidentalmente alguns dados como, número de cartão de crédito, idade, sexo, peso, altura e estilo de dança, deverá recadastrar novamente na próxima vez que logar no aplicativo. Caso não recadastre o Cliente ficará impedido de contratar um Personal Dancer. E da mesma forma, se um Personal excluir alguns dados do seu cadastro no App acidentalmente, não poderá receber nenhuma solicitação de contratação até que recadastre novamente seus dados. Portanto, uma operação segura de um sistema depende de ele estar disponível e operando de forma confiável(SOMMERVILLE, 2011).

A segurança da informação se apoia em três pilares basilares: confidencialidade, integridade e disponibilidade. A confidencialidade se relaciona com o ideal de privacidade das informações e restrição de acesso. Nesse sentido, a segurança da informação deve ser estruturada de forma que garanta o total sigilo dos dados do usuário, evitando que ações maliciosas roubem e exponham conteúdos visando prejudicar clientes e a empresa, causando danos e prejuízos financeiros.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

#### 4.4.6 NF04 Confiabilidade e disponibilidade

A confiabilidade e disponibilidade estão direta e intimamente relacionadas, pois, o objetivo da disponibilidade é manter dados e informações sempre ativos e funcionando para fornecer os serviços que o usuário solicitar validando sempre a quem os serviços devem estar disponível. Por conseguinte, a confiabilidade tem a função de garantir que os serviços solicitados pelo usuário sejam entregues tal como foi definido nas especificações do sistema (SOMMERVILLE, 2011).

Todas as informações do banco de dados do Cliente e do Personal devem ser atualizadas em tempo real, principalmente aquelas relacionadas com valores ganho do Personal e saldo do cartão de credito do Cliente.

Normalmente, problemas de confiabilidade e disponibilidade são causadas por falhas de sistema, causados por outros sistemas, erros de especificações e comportamentos errados do sistema(SOMMERVILLE, 2011).

Portanto, deve ser analisado diariamente o funcionamento do sistema ou da aplicação com a finalidade de identificar e corrigir possíveis erros que possam causar alguns transtorno e prejuízos ao cliente e a organização.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

#### 4.4.7 Desempenho

O desempenho de um sistema é do interesse do usuário que almeja sempre uma resposta mais rápida do sistema, ou seja, uma melhora na qualidade do serviço. Nesse sentido, os usuários tendem a a rejeitar software mais lento (SOMMERVILLE, 2011).

De modo geral, os requisitos não funcionais relacionados com o desempenho estão diretamente associados à eficiência, uso de recursos e tempo de resposta do sistema.

#### 4.4.8 NF05 Desempenho e tempo de resposta

O desempenho de um software não é um requisito essencial de um sistema, mas, corresponde com o fator qualidade de software. Por isso, o tempo de resposta para uma atualização de uma foto e demais dados como: altura, estilo de dança preferido, saldo do cartão de crédito e etc. feita por qualquer cliente não pode passar de 4 segundos.

Ao planejar e desenvolver um software é necessário saber que é impossível otimizar todos os atributos do sistema, pois, a robustez de um software leva a perda de desempenho. Pensando nisso, para que o software tenha um bom desempenho é necessário definir os atributos de qualidade mais importante para o software em desenvolvimento(SOMMERVILLE, 2011).

**Prioridade:**    ☐ Essencial            ☒ Importante            ☐ Desejável

#### 4.4.9 Segurança



Segundo relatórios da GARTNER<sup>3</sup> o setor de data security teve um aumento de mercado de mais de U\$170 bilhões no fim de 2022.

Os aplicativos e softwares estão presentes em nossas vidas, assim como os celulares e computadores. Por conseguinte, nós usuários precisamos dormir tranquilo na certeza de que nossas informações estão seguras, pois, um sistema bem protegido é aquele que não permite o acesso não autorizado a seus dados(SOMMERVILLE, 2011).

Por isso, é necessário desenvolver aplicações com metodologias seguras que possam blindar o sistema e as informações do usuário, contra qualquer ameaça externa que possa prejudicar o usuário e a organização, garantido à integridade, privacidade e autenticidade dos dados do sistema.

#### 4.4.10 NF07 Segurança - usuário

Informalmente, a segurança de um sistema é imensurável, pois, é a análise da probabilidade de o sistema causar danos a pessoa e a seu ambiente (SOMMERVILLE, 2011). É imensurável, justamente porque existem muitos modos de acesso ainda inexplorados ou desconhecidos e portanto, é difícil bloquear todos. Por conseguinte, só descobrimos que um sistema não é tão confiável quanto imaginávamos quando alguém consegue penetrá-lo.

O aplicativo Personal Dancer segue o manual das boas prática de programação para se certificar de alcançar um alto nível de segurança e proteção, para evitar possíveis inconvenientes e prejuízos. Pois, se um invasor alterar o código e deixar os Personais Dancers ocultos ou com Status dançando, não terá Personal Dancer disponível para ser contratado e o Personal terá prejuízos.

Para desenvolver um software protegido e confiável é preciso evitar erros durante a fase de desenvolvimento, detectar e remover erros enquanto o sistema ainda está na fase de testes e limitar os danos causados por falhas operacionais(SOMMERVILLE, 2011).

Todo usuário cadastrado no sistema deve confirmar seu usuário e senha para incluir, alterar e excluir informações do banco de dados, bem como, para contratar Personal, cujo pagamento é debitado automaticamente do cartão de crédito registrado no aplicativo do cliente.

---

<sup>3</sup> <https://www.gartner.com/en/documents/3889055>

**Prioridade:**   ☐ Essencial      ☒ Importante      ☐ Desejável

#### 4.4.11 Padrões

Padrões são normas que devem ser seguidos pelos usuários ou pelo seu processo de desenvolvimento visando alcançar um padrão melhor de manutenibilidade. Nesse sentido a norma NBR 5462 nos diz que: “a manutenibilidade é a capacidade de um item de executar suas funções requeridas, sob condições de uso especificadas, quando a manutenção é executada sob condições determinadas e mediante procedimentos e meios prescritos”.

#### 4.4.12 NF10 Manutenção de correções

Na elaboração do projeto, definimos todos os requisitos, documentamos problemas e soluções que aparecem durante a fase de desenvolvimento e teste. No entanto, problemas pode aparecer após a aplicação estar em modo de produção. Quando isso acontece, é comum bater o desespero e executar uma correção de emergência sem alterar o projeto e os requisitos deixando o código e o sistema desalinhados, dificultado a manutenção e correções futuras(SOMMERVILLE, 2011). Para evitar que isso aconteça, após passar o susto, deve se debruçar novamente sobre o código com tempo e calma para fazer uma análise aprofundada e reimplementar a correção de emergência, bem como, atualizar a documentação do sistema para facilitar futuras correções reduzindo custos de manutenção.

#### 4.4.13 NF11 Tratamento de exceções

O aplicativo Personal Dancer está sendo desenvolvido na linguagem de programação Flutter/Dart e, o tratamento de exceções em Flutter/Dart é bem simples. Nesse sentido, o programador pode criar e utilizar a exceção que quiser. Pode criar um método e tratá-la no próprio método, o que é mais comum ou criar um método e tratá-la no próximo método, chamando o método de origem.

Como exemplo, vou descrever o tratamento de exceções em java, que utiliza como padrão os comandos `try` e `catch`, conforme o exemplo abaixo:

```

String excecao = "";
try {
    throw task.getException();
}catch ( FirebaseAuthWeakPasswordException e){
    excecao = "Digite uma senha mais forte!";
}catch ( FirebaseAuthInvalidCredentialsException e){
    excecao= "Por favor, digite um e-mail válido";
}catch ( FirebaseAuthUserCollisionException e){
    excecao = "Esta conta já foi cadastrada";
}catch (Exception e){
    excecao = "Erro ao cadastrar usuário: " + e.getMessage();
    e.printStackTrace();
}

```

*Figura 6: Sintaxe de uso try-catch*

#### 4.4.14 Hardware e software

Hardware e software são elementos essenciais para o bom funcionamento e desempenho de celulares, computador, tablets, TVs e etc. O Hardware é a parte física, ou seja, as peças e aparatos eletrônicos que conectadas de forma correta, fazem o equipamento funcionar. O software por sua vez, é um sistema que executa atividades a exemplo do sistema operacional windows.

Nesse trabalho, utilizei um computador i5 da terceira geração com 16 gigas de memória RAM e placa de video comum. Instalei o sistema operacional Linux Ubuntu 22.04, que a meu ver, funciona melhor o software Android Studio, que é um ambiente de desenvolvimento integrado para desenvolver para a plataforma Android. Curiosamente, queimou a placa mãe do meu computador i5 e meu i3 não aguenta executar o código do aplicativo e por isso, estou fazendo essa versão de trabalho como forma alternativa.

#### 4.4.15 NF12 Flutter/Dart

Flutter/Dart é uma linguagem de programação orientada a objetos e foi desenvolvida recentemente pela GOOGLE, Flutter é uma framework, ou seja, uma ferramenta de desenvolvimento com foco em dispositivos móveis. É uma das linguagens mais eficazes para o desenvolvimento de aplicações para desktop e dispositivos móveis, pois é rápida, segura, eficaz, prática e confiável.

**Prioridade:**   ☐ Essencial      ☒ Importante      ☐ Desejável

#### 4.5 Regras de Negócio

Regras de negócio é um conjunto de instruções que a maioria dos usuário seguem sem nenhum sistema, mas que, carece de uma aplicação que garanta, por exemplo: as regras básica de conduta e a arte de fazer bons negócios que reflitam a imagem da política interna da empresa ou aplicação. No âmbito organizacional, que inclui processos de alto nível estratégico, regras de negócio devem ser seguidas ao se usar o sistema, pois, elas afetam a execução do software e são aplicadas a todos os dados do banco de dados(SOMMERVILLE, 2011).

Em resumo, os principais regras de negócio do aplicativo São:

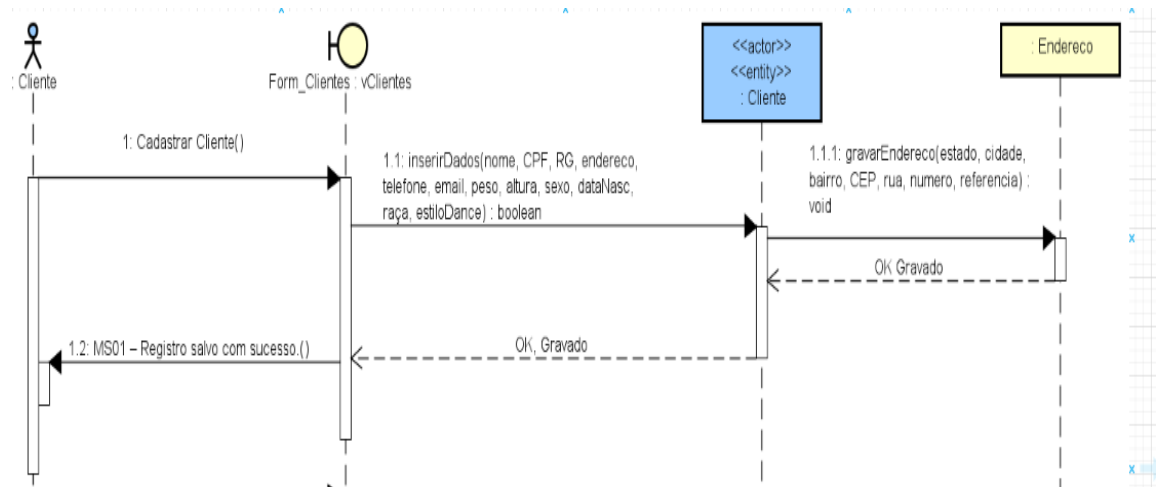
*Quadro x regras de negócio*

RN01	Validar tipo de usuário
RN02	Verificar se usuário já está cadastrado
RN03	Só poderá ser cadastrar usuários maiores de 18 anos
RN04	Verifica se não foram alterados campos que não podem ser alterados
RN05	Verifica se não foram excluídos campos que não podem ser excluídos
RN06	Validar pagamento via cartão de crédito
RN07	O usuário Cliente tem preferência na recontração do mesmo Personal
RN08	O usuário só poderá contratar o Personal se ambos estiverem no mesmo clube e com o GPS ativo

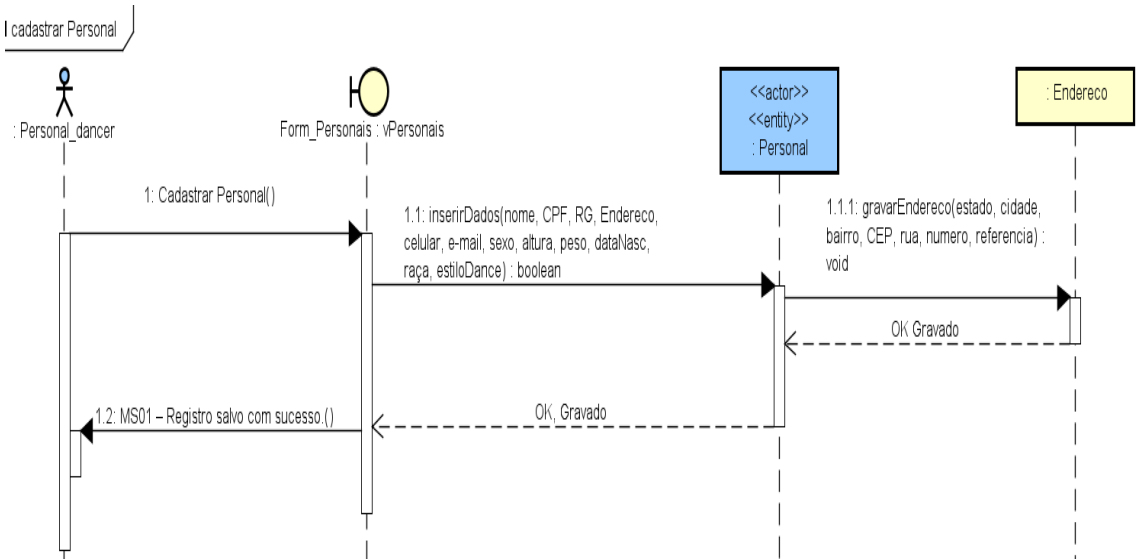
#### 4.6 Diagramas de sequência

Diagrama de sequência mostra a sequência de interações requeridas para completar alguma operação. Em UML, diagramas de sequência podem estar associados com caso de uso (SOMMERVILLE, 2011). Primeiramente, foi definido o papel do sistema através do caso de uso, agora, será definido e descrito como o software realizará seu papel por meio do diagrama de sequência.

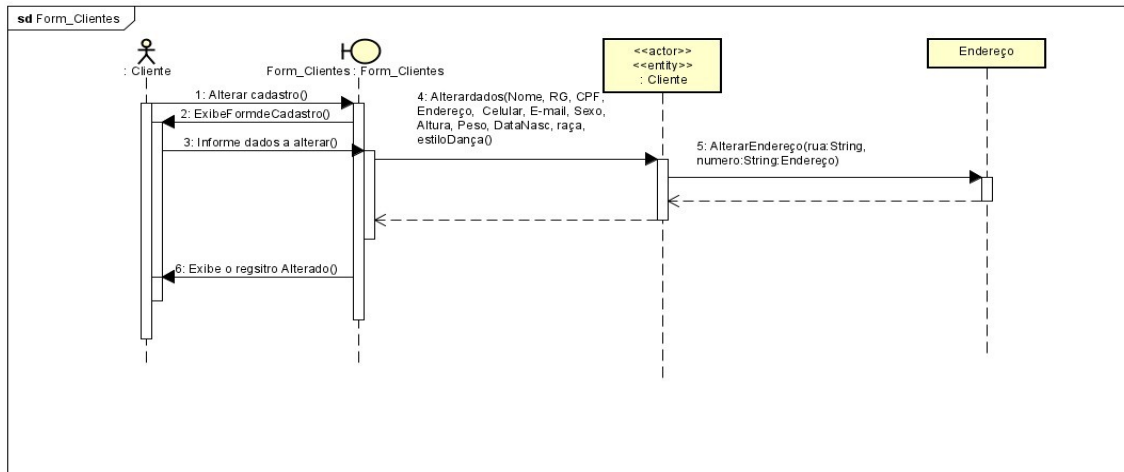
4.6.1 Cadastrar Cliente



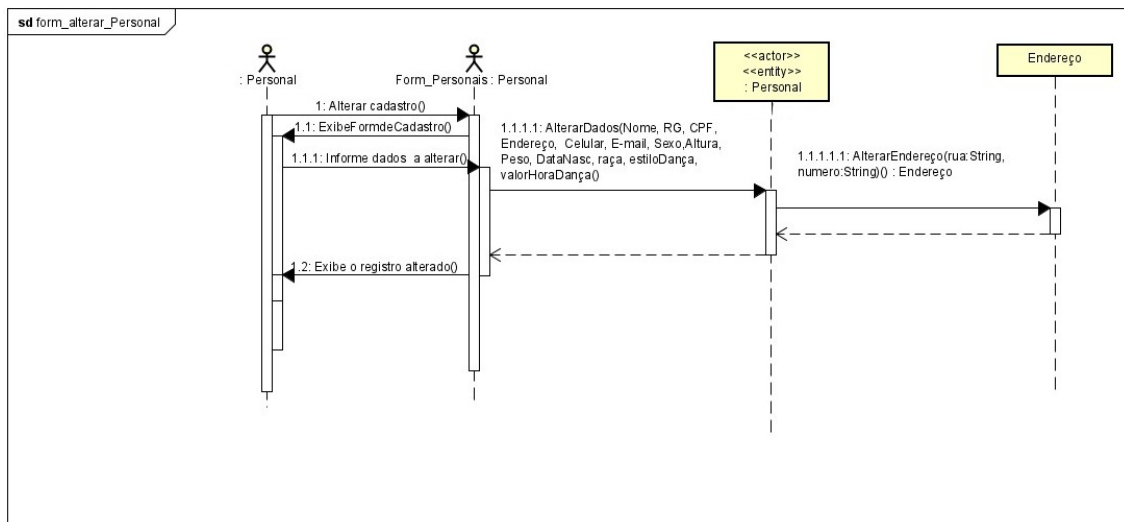
4.6.2 Cadastrar Personal Dancer



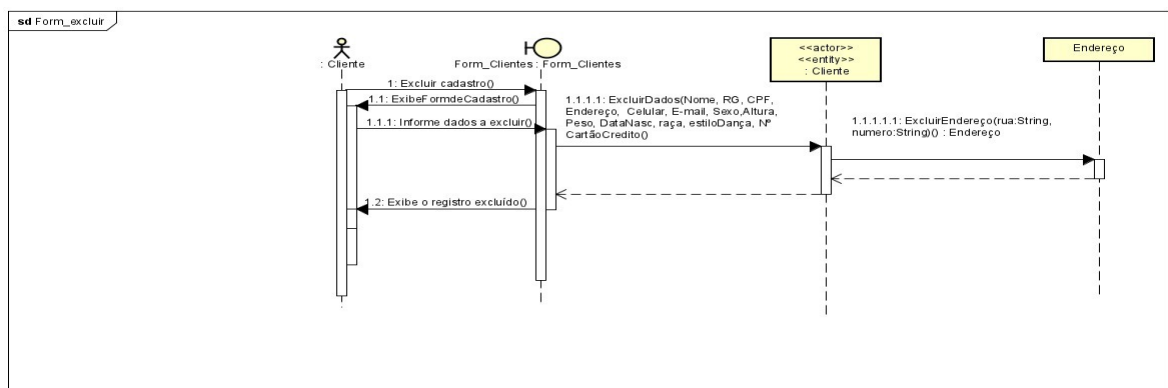
4.6.3 Alterar cadastro Cliente



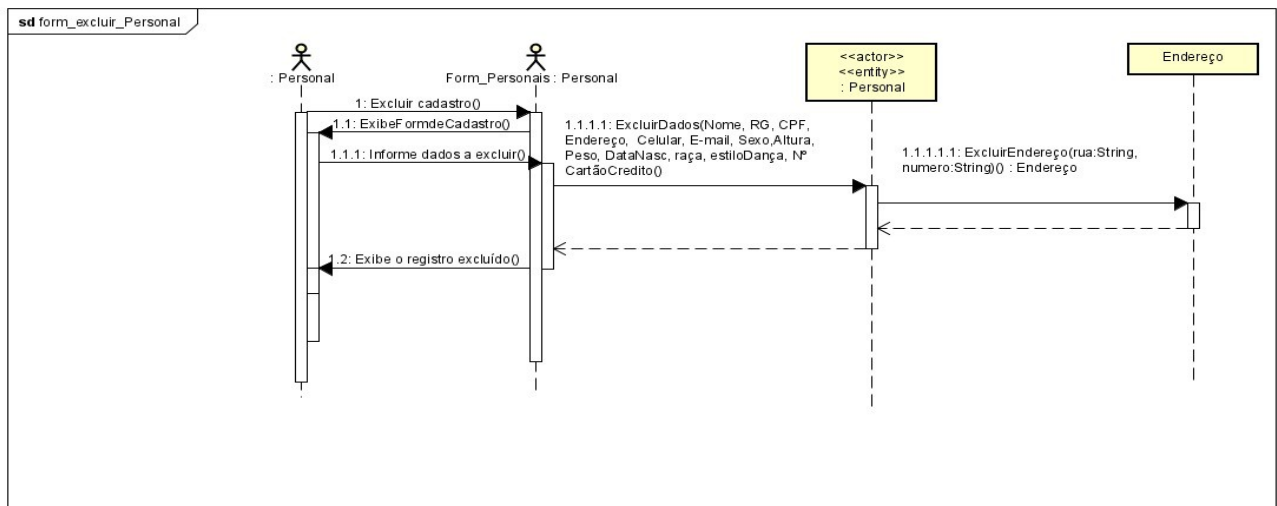
#### 4.6.4 Alterar Cadastro Personal Dancer



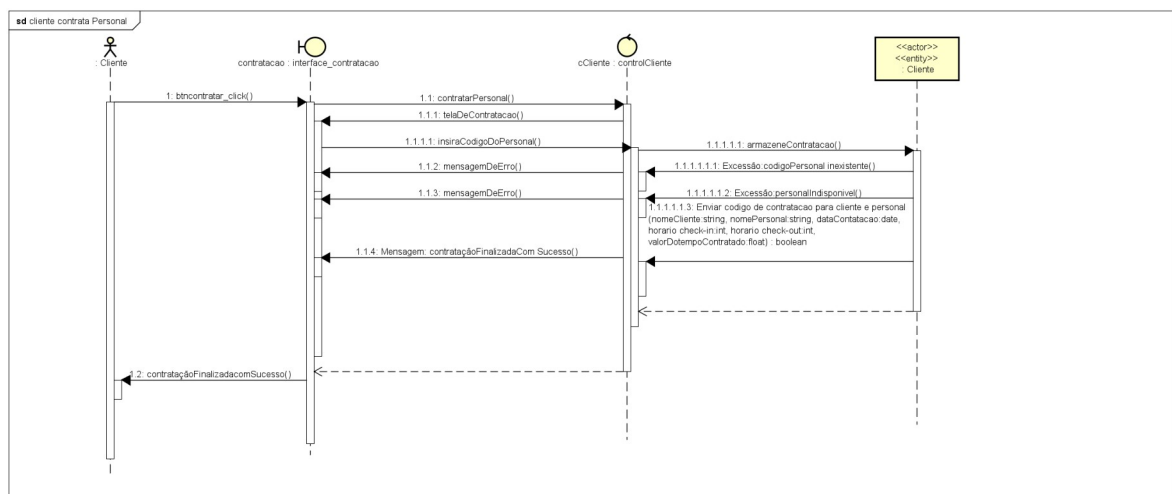
#### 4.6.5 Excluir Cliente



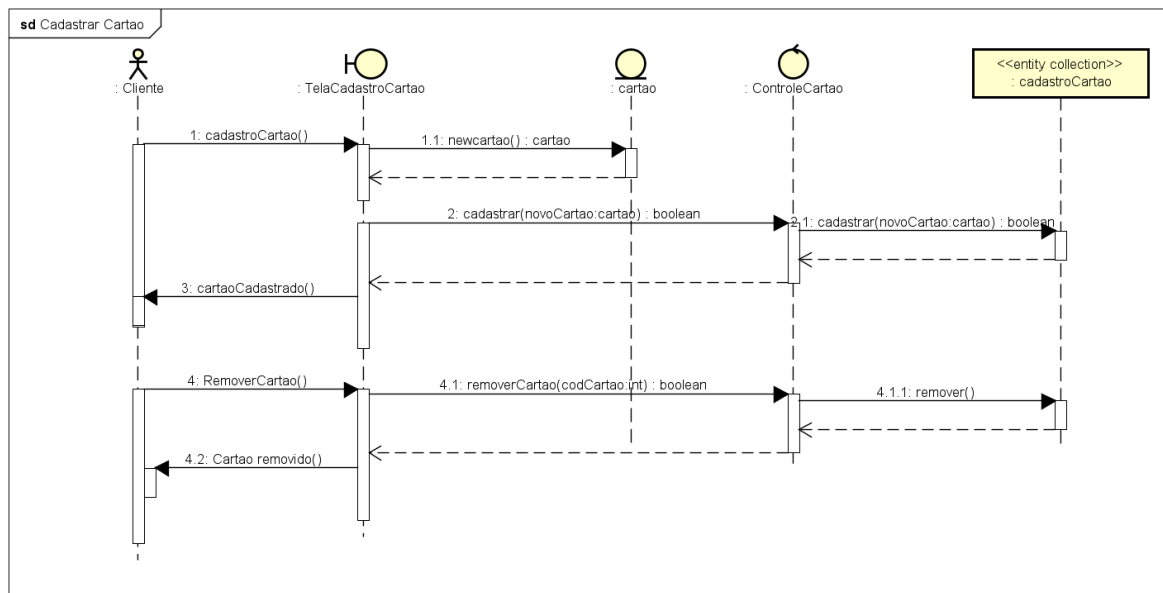
#### 4.6.6 Excluir Personal Dancer



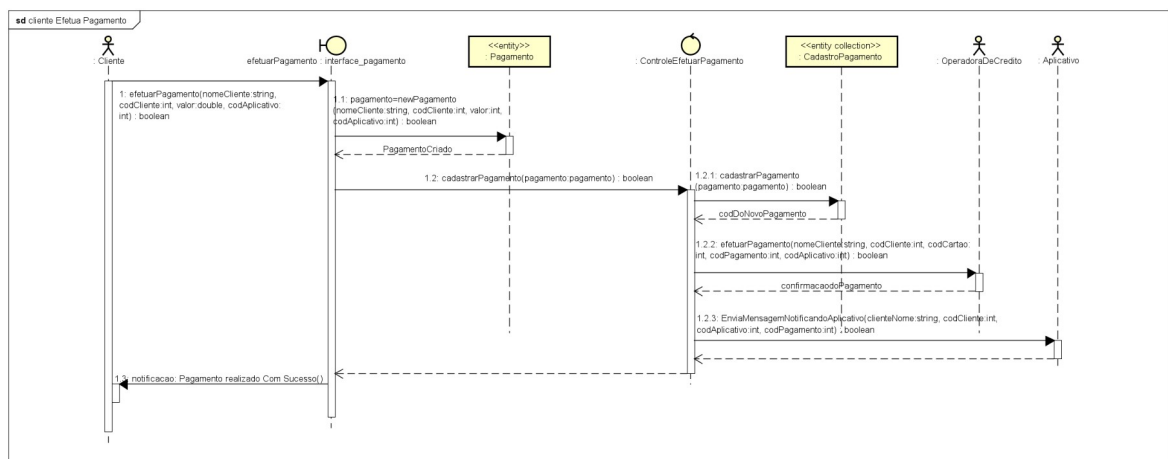
#### 4.6.7 Cliente contrata Personal\_dancer



#### 4.6.8 Cliente cadastrar cartão

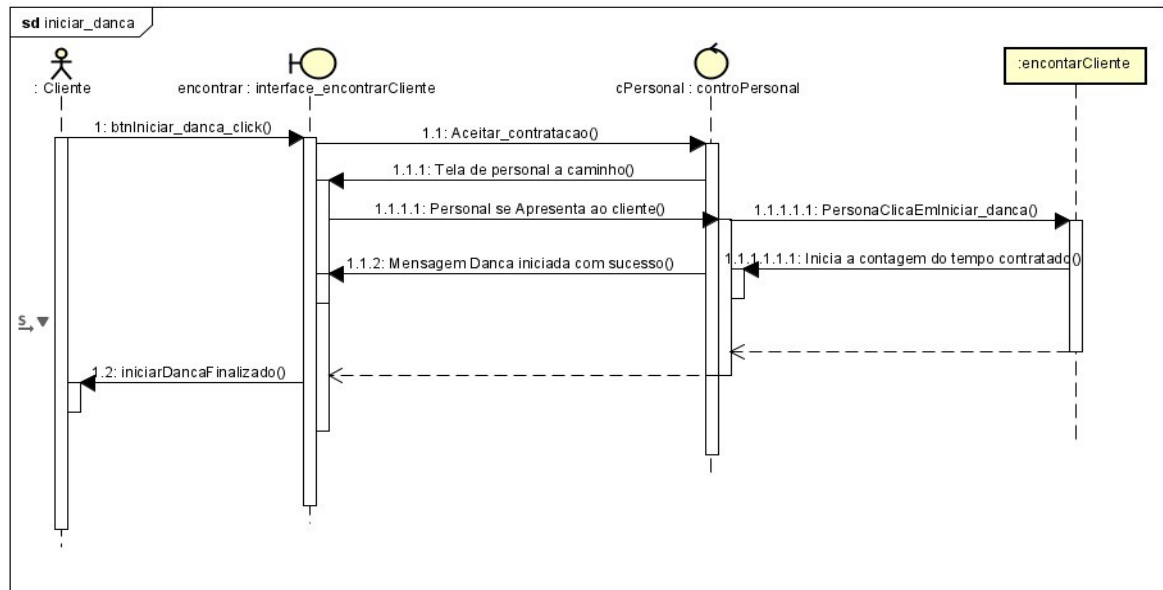


#### 4.6.9 Cliente efetuar pagamento



#### 4.6.10 Cliente receber Personal Dancer

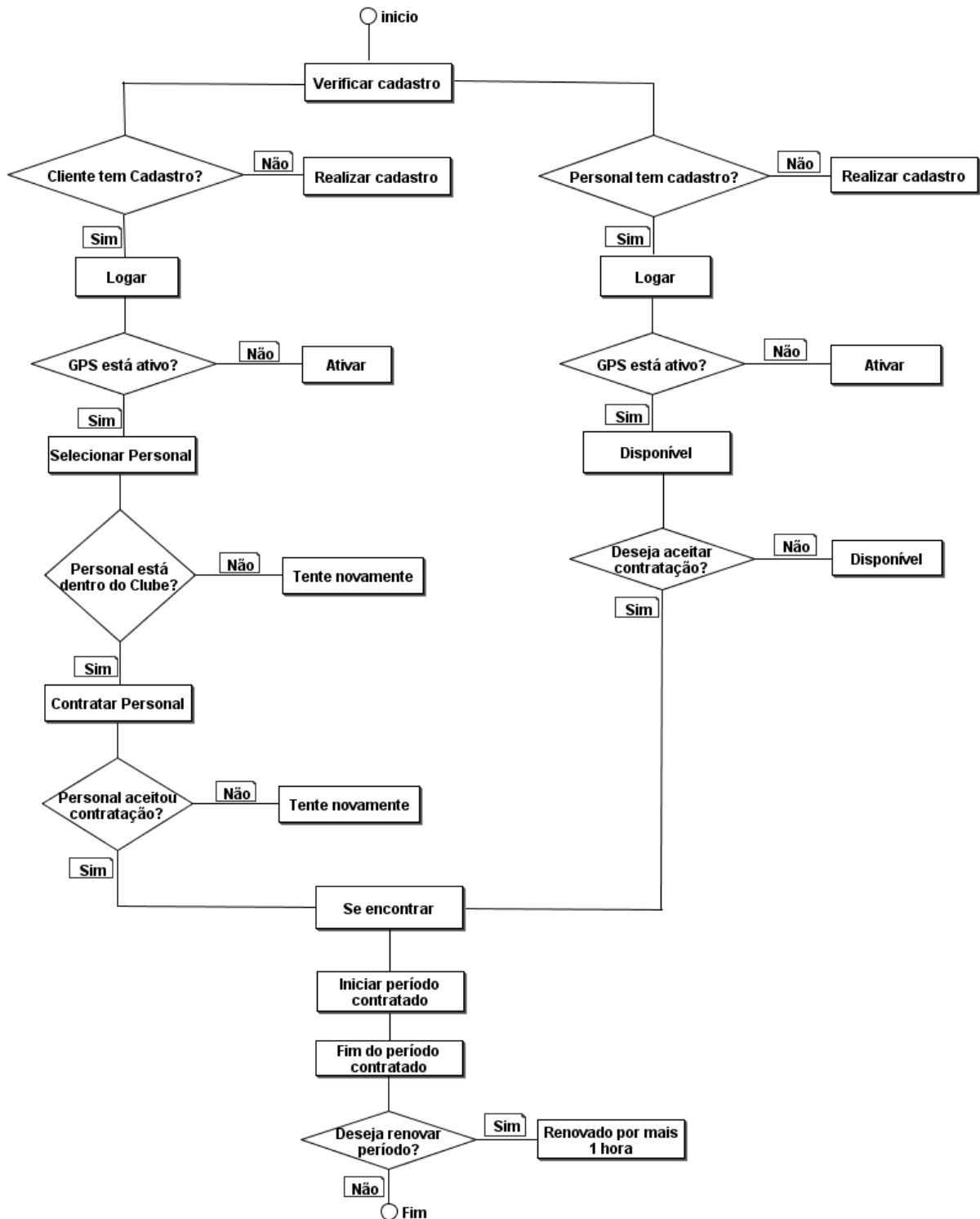




## 4.7 Fluxograma

Fluxograma é um diagrama ou representação esquemática de um processo que tem o objetivo de melhorar os processos e facilitar respectivamente a compreensão de seu funcionamento e comportamentos, bem como, os fluxos de processos e as etapas correlacionadas.

### 4.7.1 Fluxograma completo

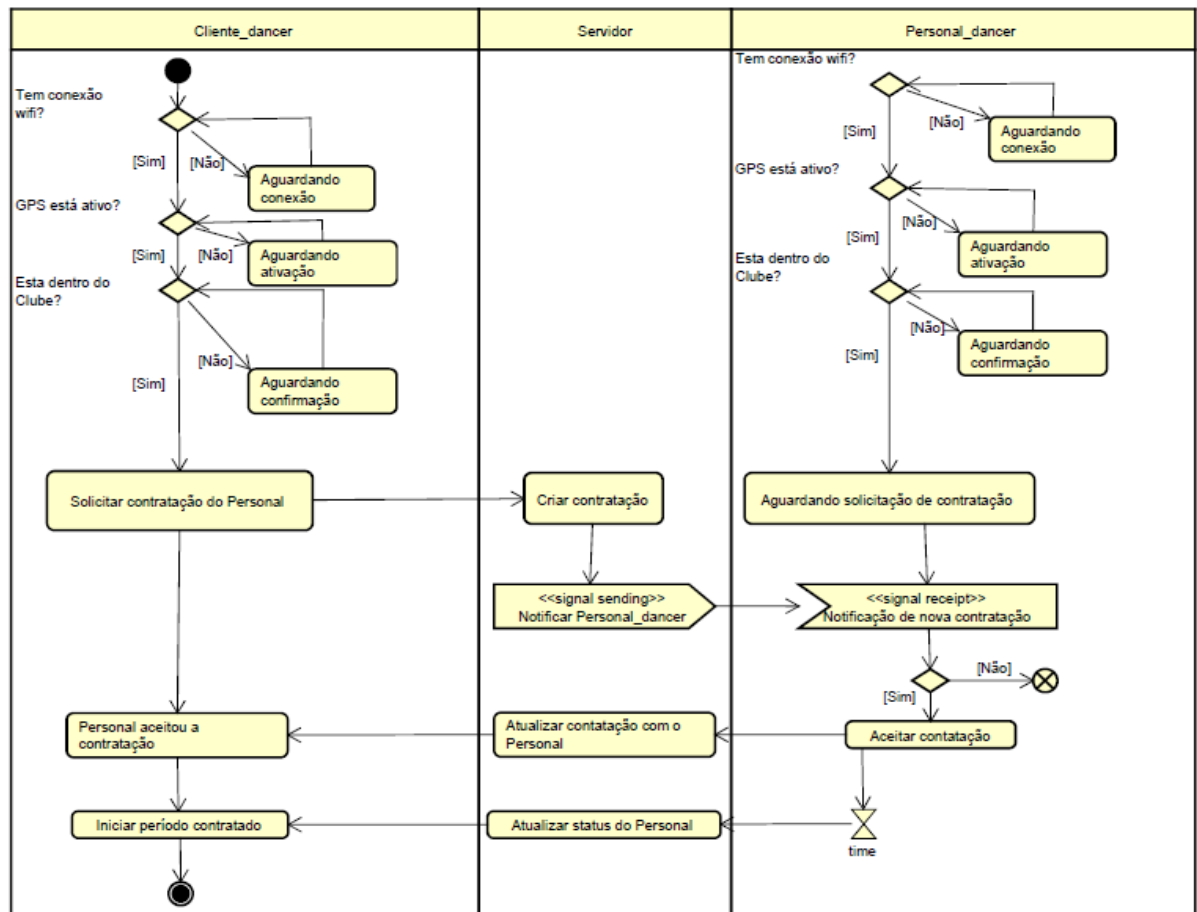


#### 4.8 Diagrama de Atividades

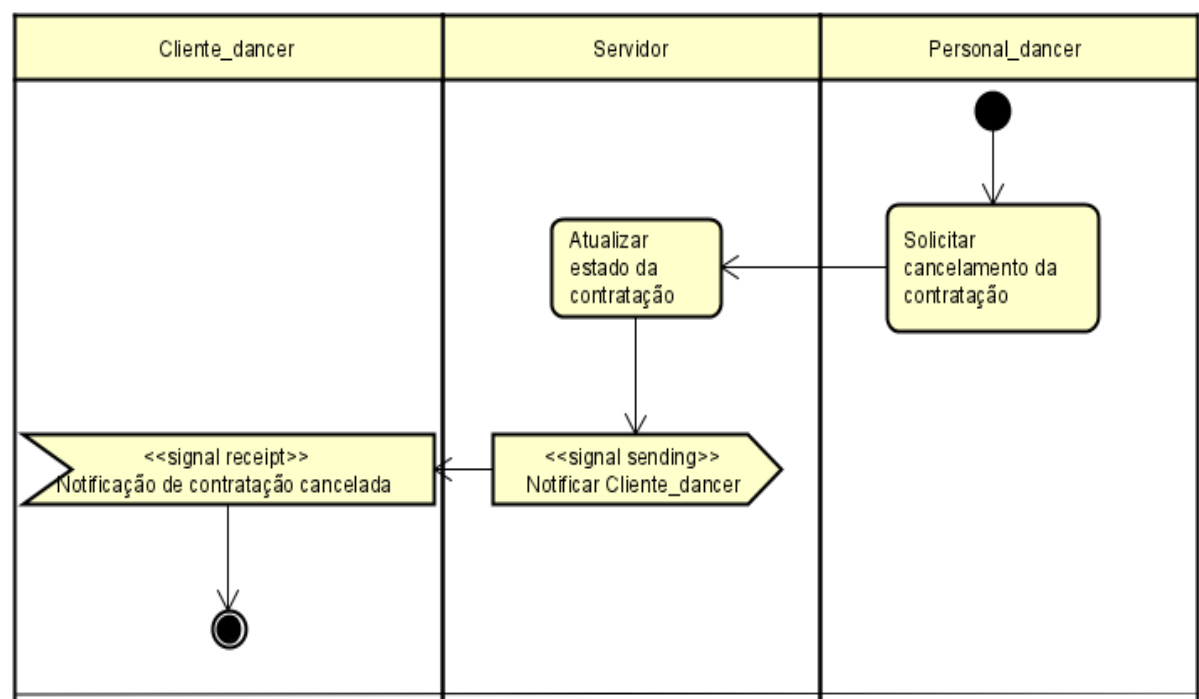
O diagrama de atividades é uma evolução do diagrama de máquinas de estado, pois, descreve de forma detalhada o processo computacional completo de

uma atividade. Assim, o objetivo principal do diagrama de atividades é de especificar o comportamento do software do ponto de vista funcional e das suas funcionalidades, diferenciando do diagrama de sequencia que foca no processo completo.

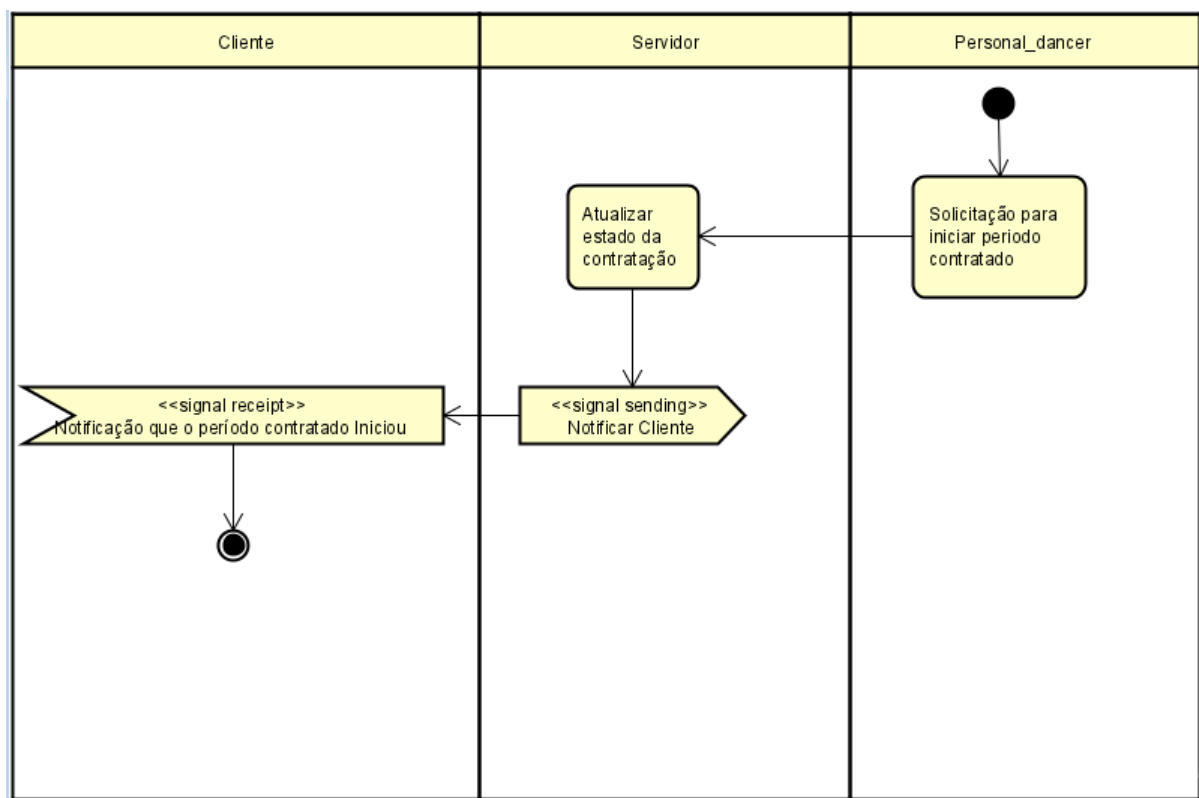
4.8.1 Personal aceita contratação



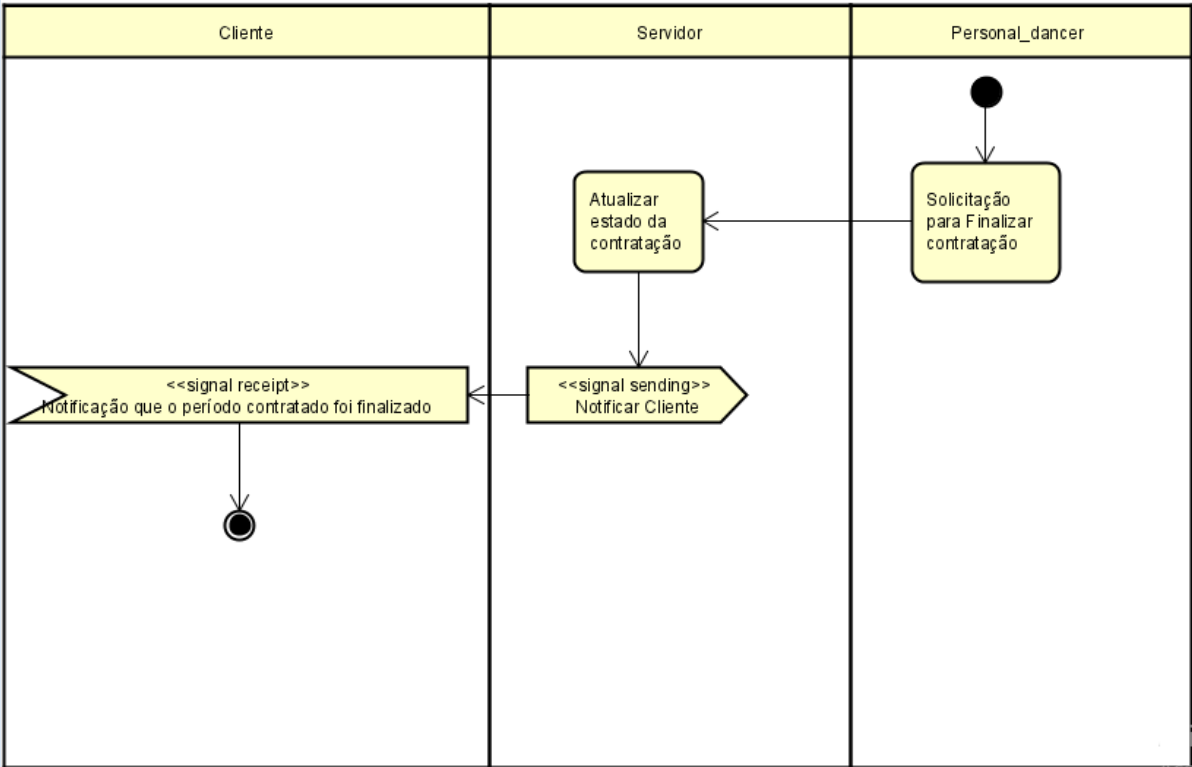
4.8.2 Cancelar contratação



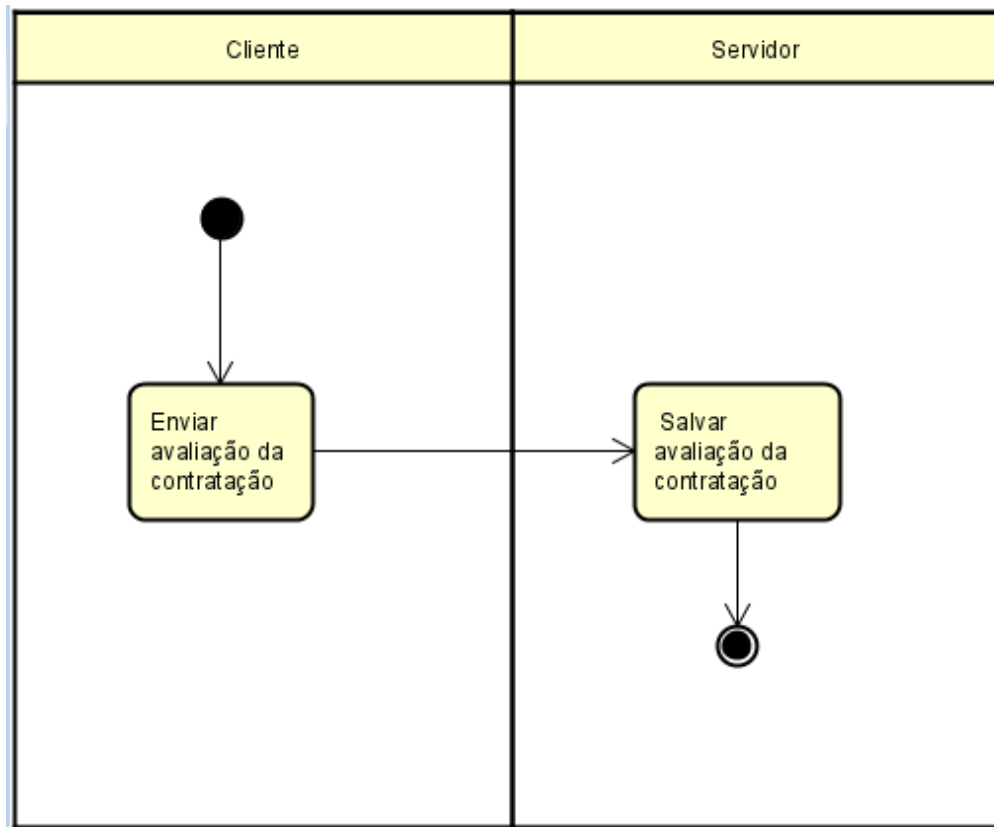
4.8.3 Iniciar período contratado



4.8.4 Finalização do período contratado



4.8.5 Enviar avaliação da contratação do Personal Dancer



#### 4.9 Diagrama de classes

O diagrama de classes tem como objetivo principal, a descrição dos vários tipos de objetos presentes em um sistema, bem como, os relacionamentos entre eles. Além disso, de acordo com o observador, o diagrama de classes pode oferecer três perspectivas diferentes. A primeira é a conceitual básica, destinado ao cliente. Trata-se de conceitos simples que o cliente deve saber a respeito da funcionalidade prática do dia a dia do sistema, de forma que, não precise entender nada de programação. A segunda, é o diagrama específico, que é destinado aos gerentes de projeto. Como o próprio nome diz, é seletivo, foca na arquitetura e métodos e desconsidera a parte de desenvolvimento. Por fim, o diagrama de implementação, destinado a equipe de desenvolvimento, é a mais utilizada das três e foca na implementação em geral. Portanto, diagrama de classes é um tipo de diagrama UML que mostra as classes de objetos e seus relacionamentos (SOMMERVILLE, 2011). Mais abaixo, um exemplo de diagrama de classes.

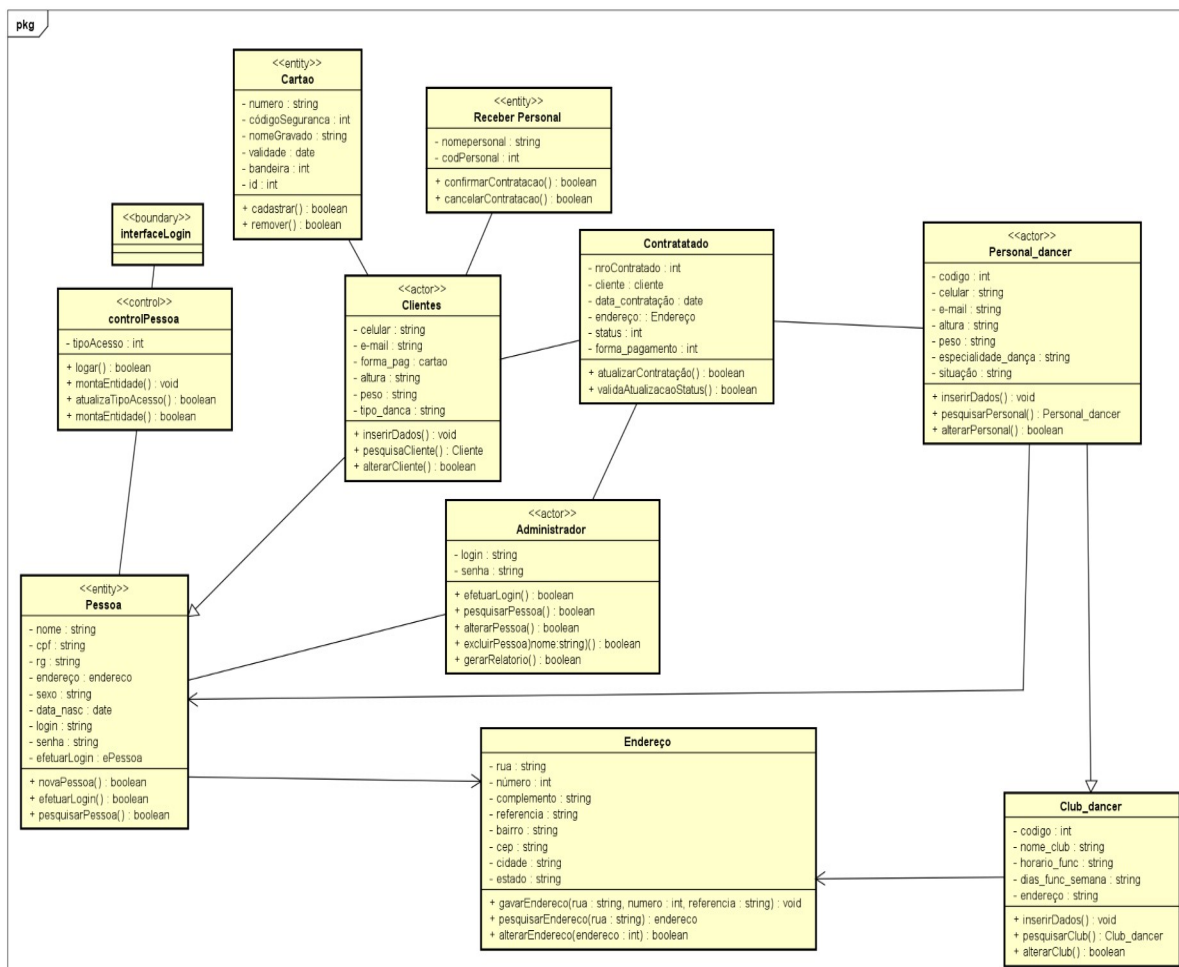


Figura 7: Diagrama de Classes – Autoria própria

#### 4.10 Diagrama de entidade relacionamento

Na engenharia de software, o diagrama entidade relacionamento conhecida pela sigla (DER) é muitas vezes o primeiro passo para a escolha e determinação dos requisitos de um projeto de sistema da informação. Resumidamente, o DER é um tipo de fluxograma que demonstram como as 'entidades' se relacionam entre si dentro do sistema. Quando falamos de 'entidades', nos referimos a pessoas, lugares, objetos ou conceitos. No projeto do trabalho, desenhamos três entidades a saber: cliente, Personal e clube. No exemplo abaixo, existem relacionamentos de um para muitos, pois, um cliente pode contratar um ou mais Personais, e, um Personal pode ser contratado por um ou vários clientes. Do mesmo modo, um clube pode receber vários clientes e vários Personais. O diagrama abaixo é para ilustrar, no projeto do trabalho usei um banco de dados não relacional, chamado de Firebase Realtime Database.



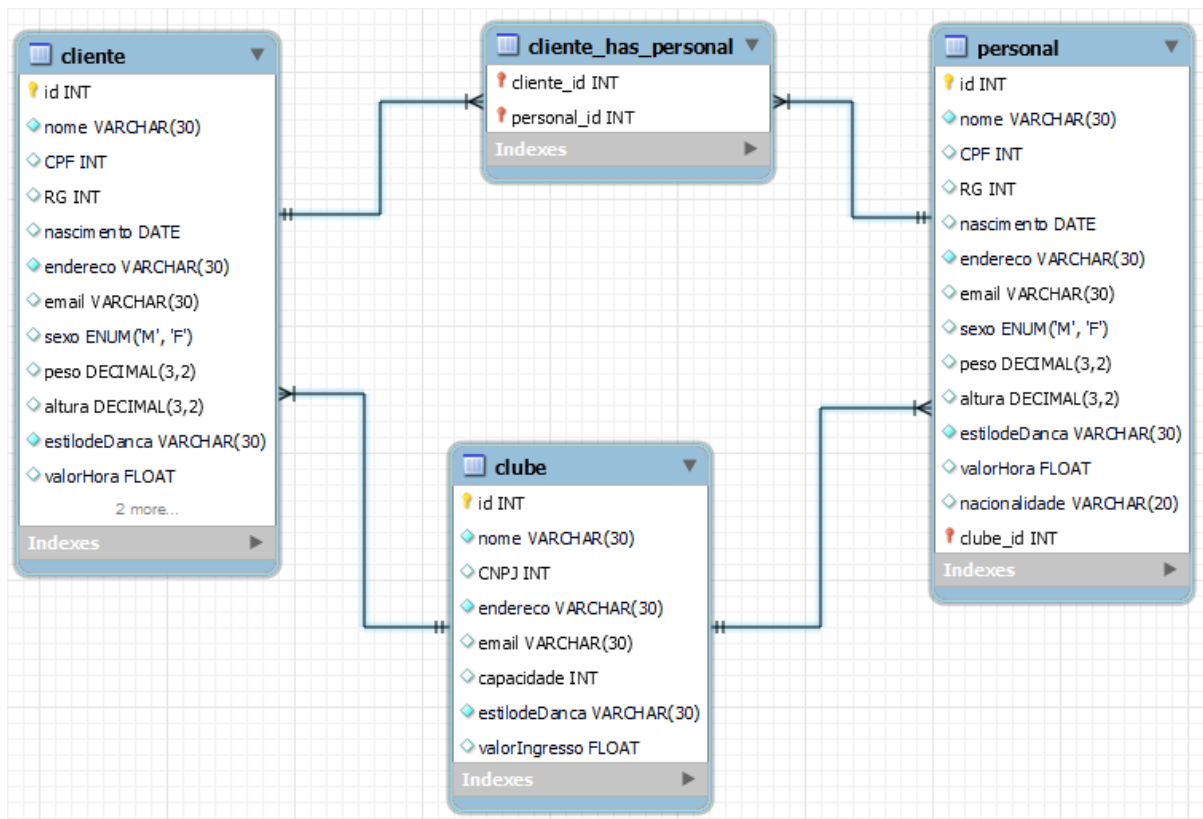


Figura 8: Diagrama entidade relacionamento

#### 4.11 Projeto de Banco de Dados

Como já descrevemos em páginas anteriores, o banco de dados que será utilizado no projeto é o Firebase Realtime Database. Trata-se de um banco de dados NoSQL que difere muito de um banco de dados relacional e que possui muitas funcionalidades e otimizações de forma rápida, prática e segura. Em relação a rapidez e praticidade podemos auferir que o banco de dados pode ser consultado e alterado diretamente no console do Firebase. Diferente do banco de dados relacional, cujos dados são armazenados e estruturados em forma de tabela, o banco de dados NoSQL é estruturado em forma de árvore JSON. Assim, através de um nó inicial chamado de nó raiz, pode ser criado novos nós e cada nó terá uma chave única de identificação. Por conseguinte, para adicionar um novo nó, basta acrescentar um novo dado a árvore.

No console do Firebase clicando em authentication vamos encontrar em users, o e-mail cadastrado, data de cadastro, último login e o UID do usuário, conforme segue a imagem abaixo:

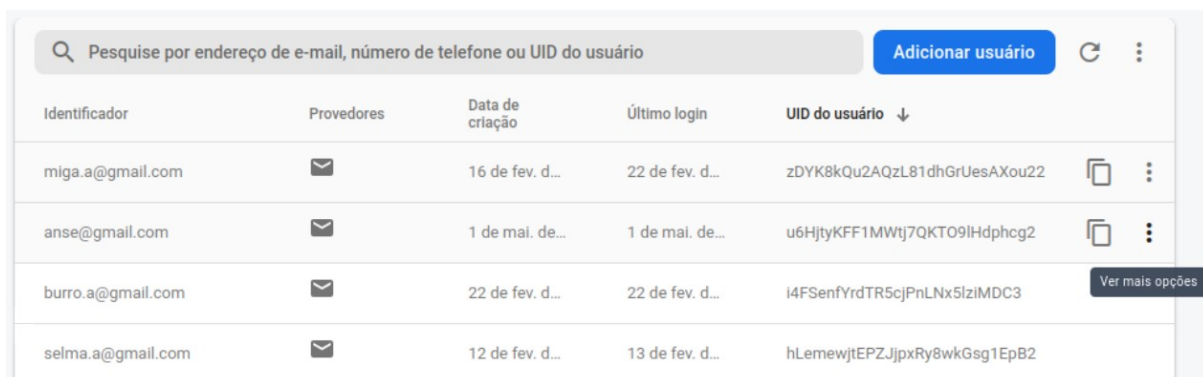


The image shows the Firebase Authentication console interface. At the top, there is a search bar with the placeholder text 'Pesquise por endereço de e-mail, número de telefone ou UID do usuário' and a blue button labeled 'Adicionar usuário'. Below the search bar is a table with five columns: 'Identificador', 'Provedores', 'Data de criação', 'Último login', and 'UID do usuário'. The table contains four rows of user data.

Identificador	Provedores	Data de criação	Último login	UID do usuário ↓
miga.a@gmail.com	✉	16 de fev. d...	22 de fev. d...	zDYK8kQu2AQzL81dhGrUesAXou22
anse@gmail.com	✉	1 de mai. de...	1 de mai. de...	u6HjtyKFF1MWtj7QKTO9lHdphcg2
burro.a@gmail.com	✉	22 de fev. d...	22 de fev. d...	i4FSenfYrdTR5cjPnLNx5IzIMDC3
selma.a@gmail.com	✉	12 de fev. d...	13 de fev. d...	hLemewjtEPZJjpxRy8wkGsg1EpB2

*Figura 9: Authentication*

Clicando em cima da linha cadastrada aparecerá três pontinho na vertical que permitirá a redefinição de senha, desativação da conta e exclusão, conforme as imagens abaixo:



Identificador	Provedores	Data de criação	Último login	UID do usuário ↓	
miga.a@gmail.com	✉	16 de fev. d...	22 de fev. d...	zDYK8kQu2AQzL81dhGrUesAXou22	⋮
anse@gmail.com	✉	1 de mai. de...	1 de mai. de...	u6HjtyKFF1MWtj7QKTO9IHdphcg2	⋮
burro.a@gmail.com	✉	22 de fev. d...	22 de fev. d...	i4FSenfYrdTR5cjPnLNx5lziMDC3	⋮
selma.a@gmail.com	✉	12 de fev. d...	13 de fev. d...	hLemewjtEPZJjpxRy8wkGsg1EpB2	⋮

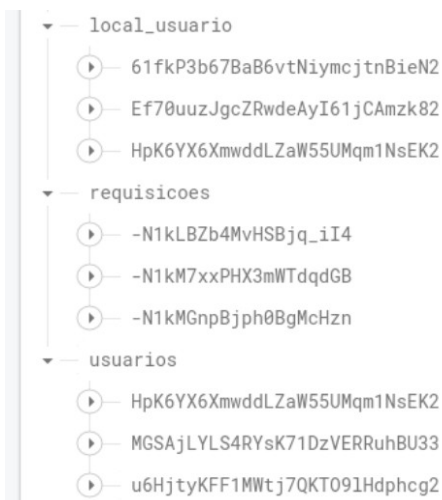
*Figura 10: Authetication - complemento*



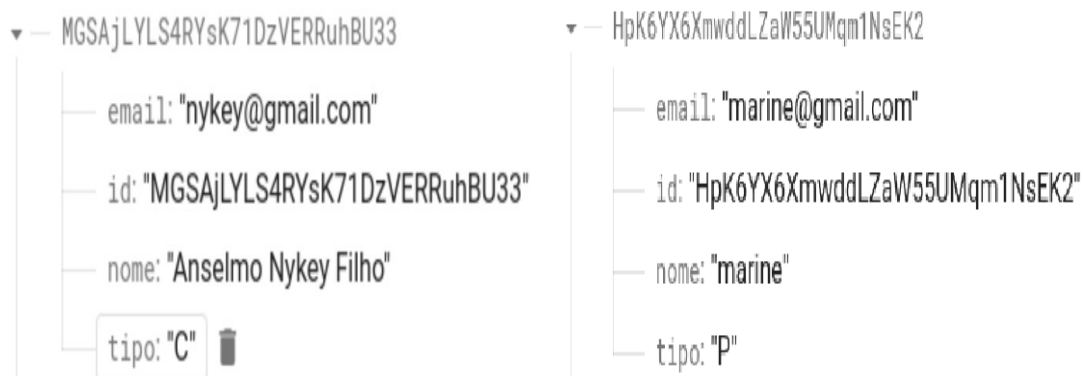
Identificador	Provedores	Data de criação	Último login	UID do usuário ↓	
miga.a@gmail.com	✉	16 de fev. d...	22 de fev. d...	zDYK8kQu2AQzL81dhGrUesAXou22	⋮
anse@gmail.com	✉	1 de mai. de...	1 de mai. de...	u6HjtyKFF1MWtj7QKTO9IHdphcg2	⋮
burro.a@gmail.com	✉	22 de fev. d...	22 de fev. d...	i4FSenfYrdTR5cjPnLNx5lziMDC3	⋮
selma.a@gmail.com	✉	12 de fev. d...	13 de fev. d...	hLemewjtEPZJjpxRy8wkGsg1EpB2	⋮

*Figura 11: Authetication - complemento II*

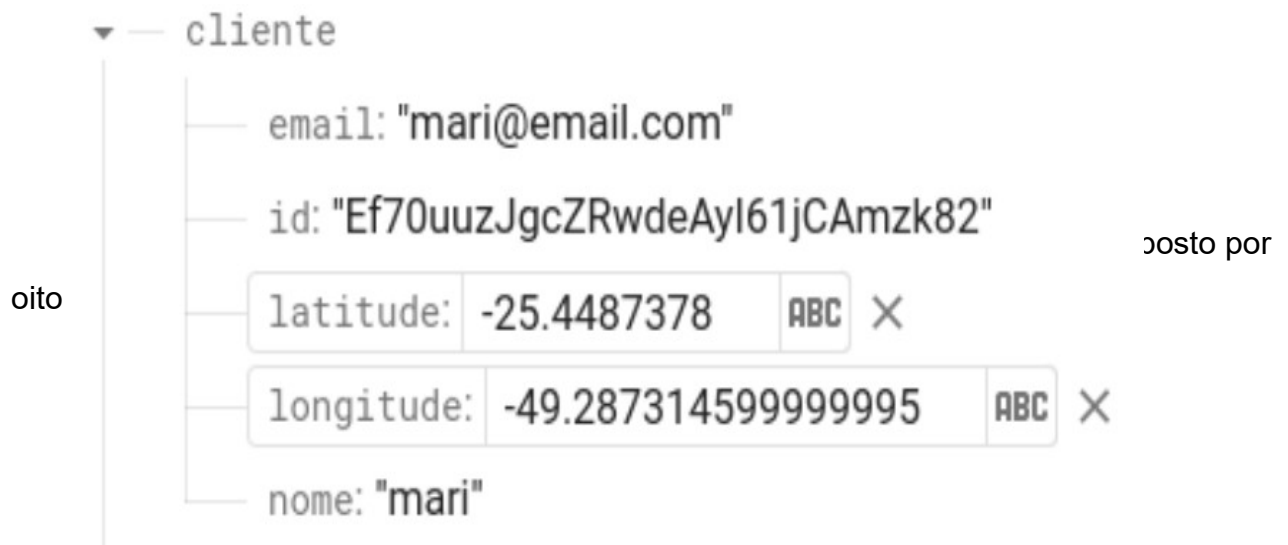
No projeto inicial, está sendo usado 3 nó raízes, sendo eles respectivamente, nó usuários, requisições e local\_usuario, conforme a imagem abaixo:



O nó raiz usuário é composto de usuário cliente e usuário Personal. Cada usuário contém os seguintes dados armazenados no banco, a saber: nome, id, e-mail e tipo. O 'tipo' foi definido com a letra 'P' de Personal e 'C' de Cliente, conforme a imagem abaixo:



O nó raiz requisições é o nó de contratações de Personais Dancers. Esse nó é composto por três nó filhos, a saber: Cliente, clube, id da requisição e Personal. O nó Cliente contém cinco atributos, sendo eles: nome, e-mail, id do cliente, latitude e longitude. Para ilustrar melhor, segue abaixo uma imagem:



atributos, a exemplo: bairro, CEP, cidade, latitude, longitude, número, rua e ID do clube. Segue abaixo, uma imagem ilustrativa:



O nó raiz Personal, a exemplo do nó raiz Cliente, é composto pelos mesmo cinco atributos do nó Cliente mais um atributo definidor, ou seja: é composto por nome, e-mail, ID de identificação do Personal, latitude, longitude e status status da contratação que define a disponibilidade do Personal. O Status pode variar de estado, pois, possui quatro categorias, a saber: aguardando, a caminho, dançando e encerrado. No Status 'aguardando', o Personal vai decidir se aceita a solicitação de contratação realizada pelo Cliente. No Status 'a caminho', o Personal já aceitou a contratação e está prestes a se apresentar ao Cliente. No Status 'dançando' Personal e cliente já estão no salão dançando e o período contratado já está sendo marcado no cronômetro. Por fim, no Status 'encerrado' é apresentado o fim do período

contratado junto com o valor total da dança. Mais abaixo, uma imagem para reforçar as descrições:



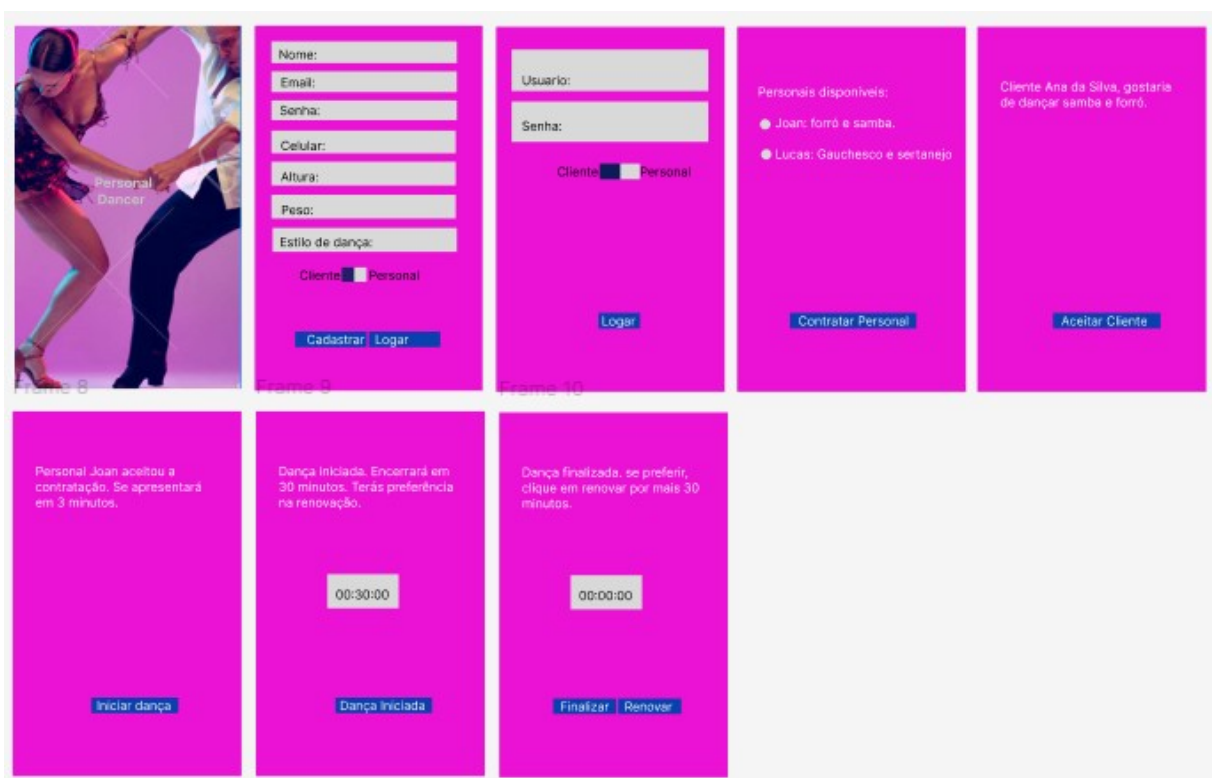
Por fim, o nó raiz 'local\_usuario' que no momento só contém os atributos de localização dos usuários para armazenar e melhorar o direcionamento das contratações. Esse nó, bem como outros, serão melhor aproveitados no futuro, visando uma otimização melhor do espaço para saber se o clube está cheio ou vazio e melhora na segurança dos usuários.

```

local_usuario
├── 61fkP3b67BaB6vtNiyrcjtnBieN2
│   ├── g: "6gkznrnn67"
│   └── 1
│       ├── 0: -25.448736666666665
│       └── 1: -49.287313333333334

```

## 5.1 Interfaces gráficas



## **1- Tela Inicial**

° Apresentará o nome do aplicativo.

## **2-Tela de cadastro e Login**

°Contém campos de texto para inserir Nome, Email, Senha, Celular, Altura, Peso e Estilo de dança.

°Um botão para alternar entre cliente e personal.

°Dois botões na parte inferior: Cadastrar e logar.

## **3 -Tela de Login**

° Campos para inserir usuario e senha

°Dois botões para alternar entre cliente e personal.

°Um botão para Logar.

## **4- Tela de seleção de pessoais disponíveis**

°Exibe uma lista de pessoais disponíveis e seus respectivos estilos de dança.

°Botões de seleção(rádios) para escolher o personal.

°Botão para contratar Personal.

## **5-Tela de solicitação (requisição) enviada ao Personal**

° Mensagem: 'Ana quer dançar forró'

°Botão para Aceitar/Rejeitar

## **6-Tela de aceitação(mensagem) enviada ao cliente**

°Mensagem: 'Personal Marcos aceitou a contratação e se apresentará em 3 minutos.

°Botão para iniciar dança que pode ser apertado por qualquer um dos dois.

## **7-Tela de contagem regressiva.**

°Exibe o início da contagem

°Exibe o fim da contagem e a opção para renovar por mais 30 minutos.

°Exibe o botão Finalizar/Renovar.



#### 5.1.5 Plano de testes

### PLANO DE TESTES

#### **Cadastrar**

O cliente terá duas opções de cadastro. A primeira é cadastrar como cliente e a segunda como Personal. Se o cliente já estiver cadastrado, receberá uma mensagem informando que o cadastro já existe e será encaminhado para a página de login.

#### **Logar**

Para fazer login tanto o cliente quanto o Personal precisam apenas digitar seu e-mail e senha. Se o e-mail estiver incorreto, será apresentado na tela uma mensagem informando que o e-mail não está cadastrado, se a senha estiver incorreta, será informado que a senha está incorreta. Como ainda estão na página de login, será possível cadastrar novo e-mail e senha a qualquer momento, mas nesse caso, será necessário preencher o cadastro completo tanto para poder contratar, quanto para prestar serviços como Personal. Se não lembrar a senha poderá clicar em esqueci senha e cadastrar nova senha.

O sistema permite o login somente se o GPS estiver ligado. Aparecerá uma mensagem perguntando se gostaria de ativar o GPS. Deve marcar sim. Se marcar não, a tela de login fechará.

#### **Contratar Personal**

Após logar o cliente escolhe o clube que está no momento. Se escolher o clube errado aparecerá uma mensagem informando que o clube escolhido difere do clube que o cliente está no momento. Isso ocorre porque, por questões de segurança, foi delimitado um raio 15 metros do centro da pista para contratações de Personais. Assim, se o Cliente estiver acima de 15 metros do centro da pista, não poderá contratar um Personal e do mesmo modo, se um Personal estiver a mais de 15 metros do centro da pista, não poderá aceitar uma contratação.

#### **Aceitar contratação**

Reforçando o que já foi dito, o Personal poderá aceitar uma contratação se ambos estiverem no mesmo clube, com GPS ativo e dentro do raio de 15 metros do centro da pista de dança. Obviamente, essas informações são apenas reforços desnecessários, pois, se não estiver com GPS ativo não será possível logar, se não estiver no mesmo clube, não será possível contratar um Personal e se não estiver dentro do raio de 15 metros do centro da pista, não será possível aceitar a contratação.

Após aceitar a contratação, o Personal irá ao encontro do cliente, se apresentará e

se estiverem em comum acordo, o Personal clicará em iniciar dança. Após iniciar a dança, o cliente terá 5 minutos para desistir da contratação sem nenhuma cobrança de taxa, pois, com a experiência sabemos que 5 minutos é mais que o suficiente para sabermos se adaptaremos ou não, ao estilo de dança e estrutura corporal do Personal, caso não, o cliente clica em cancelar e em 3 minutos, outro Personal se apresentará ao cliente.

### **Iniciar dança**

Após iniciar a dança e passar os 5 minutos de adaptação, inicia a contagem do tempo que varia de 30 minutos a 1 hora. O aplicativo inicialmente trabalhará com apenas 2 opções de tempo.

### **Finalizar dança**

Após iniciar a contagem do tempo e encerrar o período contratado, o aplicativo emitirá uma mensagem informando que o tempo encerrou e soará um alarme discreto que somente o Cliente e o Personal perceberão. Logo após, o fim do período contratado, o cliente terá entre cinco e dez minutos para solicitar uma nova contratação e ganhar a preferência do Personal. Sempre que encerrar uma contratação, o Personal terá 10 minutos de pausa para ir ao banheiro e descansar. Após esse tempo, seu status estará disponível e a qualquer momento poderá receber requisições de contratação. Se o Personal aceitar a nova contratação do Cliente, inicia nova contagem de tempo e automaticamente será debitado do cartão de crédito cadastrado no aplicativo. Se o Personal optar por não aceitar a contratação, bastando apenas ignorar a solicitação do Cliente ou aceitar a requisição de outro Cliente, o Cliente anterior ficará a disposição de outros Personais que poderão aceitar sua solicitação de contratação.

### **Relatório**

Será disponibilizado em tempo real no aplicativo o dia, horário, período contratado, clube, Personal e valores. O relatório ficará armazenado por um período de um ano para eventuais consultas.

### **Testes**

Inicialmente os testes são realizados na plataforma Android Studio validados em varias versões de Android. Lembrando que o aplicativo por questões de padrão do Android Studio, utiliza a API 29 e portanto, somente funcionará em celulares que utilizem a versão de Android 10 para cima. Pensando nisso, realizei vários testes em celulares com versão 10, 11 e 12 e constatei que funcionam da forma que deveria funcionar. Muitas pessoas utilizam celulares com versão abaixo de 10 e para continuar a utilizar aplicações

que recebem atualizações de tempos em tempos, a exemplo de bancos, deverão adquirir celular com versão de Android 10 para cima.

Os testes focaram no cadastro para validar se é possível cadastrar duas vezes a mesma pessoa e validamos que não. No login, validamos que não é possível logar com usuário errado, senha errada e com GPS desativado. Na contratação, fixamos os endereços de clubes para validar que o Cliente só poderá contratar se estiver no clube que está no momento. Essa parte irei desenvolver melhor para melhorar a segurança e mapear melhor os clubes para sabermos quais clubes estão vazios e quais estão cheios para direcionar melhor os Personais no futuro. Quanto ao modelo do celular, foi validado o funcionamento do aplicativo em celulares de vários formatos, inclusive tablets, para confirmar se teria algum erro ou se a tela ficaria diferente do esperado e validamos que funciona perfeitamente. Fizemos outros teste, mas esses são os principais e que estão diretamente relacionados com o funcionamento do aplicativo.

#### Tabela Plano de testes

---

## 6 ANÁLISE DE RISCOS

Na análise de risco, vamos descrever em poucas linhas, alguns possíveis riscos já tratados na codificação do aplicativo. Como se sabe, o sistema operacional Linux é um dos sistema mais seguro que existe até a presente data de escrita desse trabalho. Assim, pensando em segurança, por padrão, a segurança das aplicações Android tem como base a segurança do sistema Linux. No Linux, cada aplicação executa um único processo e cria um único usuário, de modo que, um usuário x não pode acessar arquivos do usuário y. Resumidamente, ‘aplicações não podem interagir diretamente com outras aplicações’(LECHETA, 2015).

### 6.1-Intentes e riscos

Uma intent é uma mensagem da aplicação para o sistema operacional, solicitando que algo seja realizado”(LECHETA, 2015). Algo a ser realizado pode ser: enviar e-mail, enviar uma mensagem pela aplicação informando que o contratado está a caminho, compartilhar um texto com usuários ou clientes cadastrados na aplicação, tirar uma foto, enviar mensagem informando que o GPS esta inativo, informar que o tempo contratado encerrou e etc. Além disso, uma intent é entre muitas coisas, a forma utilizada para fazer com que aplicações em processos

diferentes se comuniquem, utilizando uma mensagem que pode ser interceptada por qualquer aplicação (LECHETA, 2015).

Existem vários tipos de intents que podem ser recebidas e tratadas por uma classe chamada Broadcast Receiver que é uma das mais importantes na arquitetura do Android, pois ela é utilizada para interceptar eventos do sistema, ou seja, mensagens geradas por uma intent (LECHETA, 2015). O recebimento, tratamento e tipo de mensagens são configuradas pelo desenvolvedor que define as regras e filtros no sistema operacional. Assim, o trabalho do Broadcast Receiver é realizado em segundo plano, sem interfaces gráficas e interações com o usuário.

Pode haver riscos de segurança porque as intents carregam conteúdo que pode ser alterado, manipulados ou criado por uma pessoa má intencionada que pode explorar alguma possível vulnerabilidade no código do Broadcast Receiver ou em outros componentes internos da aplicação. Um exemplo de risco corriqueiro que acontece com grande frequência é a famigerada 'Denial of Service'. Esse risco, ocorre quando uma pessoa má intencionada envia propositamente um texto com uma codificação não suportada, com o objetivo de causar erros no código e posteriormente, falha no serviço.

Esse tipo de ataque pode ser tratado com uma atenção especial ao código, tratando melhor as exceções, focando na melhor qualidade de debugging e ofuscamento do código, visando eliminar possíveis brechas que possam ser utilizadas por um atacante, configurando no projeto apenas as permissões necessárias ao funcionamento da aplicação.

## 6.2 Banco de dados e riscos

O Android tem suporte ao SQLite, um leve e poderoso banco de dados. Cada aplicação pode criar um ou mais bancos de dados, um banco de dados é visível somente à aplicação que os criou.

É importante ressaltar que bancos de dados não possuem criptografia e que, porventura, se as regras de permissões do sistema de arquivos da aplicação for comprometida, não será possível garantir a confidencialidade dos dados persistidos. Pensando nisso, foi utilizado no projeto o Firebase Data Base que nos garante uma maior segurança na comunicação e na transferência de dados.

7 ORÇAMENTO

ORÇAMENTO				
DESENVOLVEDOR				
PROGRAMADOR	QTDE DE HORAS	VALOR HORA		TOTAL
Anselmo	190	35,00		R\$ 6.650,00
		Total		R\$ 6.650,00
EQUIPAMENTOS				
EQUIPAMENTOS	VALOR INICIAL	VIDA ÚTIL	TOTAL	VALOR FINAL
Dispositivo Móvel	R\$ 1.200,00	R\$ 3,70	R\$ 317,75	R\$ 882,25
Notebook	R\$ 3.050,00	R\$ 0,65	R\$ 1.339,50	R\$ 1710,50
		Total	R\$ 1.657,25	
SOFTWARES				
SOFTWARE	QTDE LICENÇAS	VALOR LICENÇA		TOTAL
Linux Ubuntu	1	R\$ -		R\$ -
Android Studio	1	R\$ -		R\$ -
Firebase Data Base	1	R\$ -		R\$ -
Astah	1	R\$ -		R\$ -
KolourPaint	1	R\$ -		R\$ -
Libre Office	1	R\$ -		R\$ -
		Total		R\$ -
VALOR TOTAL DO PROJETO				
ITEM		VALOR		TOTAL
MÃO DE OBRA PROGRAMADOR		R\$ 6.650,00		R\$ 6.650,00
MÃO DE OBRA ANALISTA		Não contabilizada		R\$ 0,00
SOFTWARES		R\$ -		R\$ 0,00
EQUIPAMENTOS		R\$ 1657,25		R\$ 1657,25
		Total		R\$ 7.307,25

## 5 CONSIDERAÇÕES FINAIS

Como podemos observar no estudo elaborado neste trabalho, a utilização de aplicativos móveis vem se tornando cada vez mais, uma realidade no dia a dia das pessoas. Nas descrições dos capítulos anteriores foram lançadas a ideia do projeto do aplicativo desde os rascunhos iniciais, passando pelo levantamento de requisitos, banco de dados firebase data base, plataforma do Android Studio e a codificação propriamente dita.

Os aplicativos estão presentes no cotidiano das pessoas ao ajudá-las a realizar diversas tarefas durante o dia, como chamar um UBER, pagar um fatura pelo aplicativo bancario, participar de videoconferência, realizar compras online e etc. Além das atividades corriqueiras do dia a dia, o App se estende ao âmbito empresarial e profissional que é um mercado com grande potencial de crescimento para os próximos anos, motivadas pela facilidade e agilidade na comunicação através do contato direto com o cliente e principalmente pela necessidade de melhora da eficiência dos serviços oferecidos, a exemplo da agilidade da entrega e processos mais ágeis reduzindo operações manuais e pilhas de papeis desnecessários.

O aplicativo Personal Dancer é o objeto de estudo desse trabalho, cuja ideia, está inserida na cultura brasileira de dança de salão, mas que, até o momento, não existem aplicativos que explorem esse nicho em particular. O aplicativo irá facilitar a contratação de Personais da Dança, de forma rápida, prática e segura, tanto para o cliente, quanto para o profissional da dança.

No Brasil, a moda da cultura popular dos clubes de dança, esse serviço é muito requisitado, principalmente por pessoas com idade acima dos 40 anos. Isso ocorre porque a dança é indicada por muitos terapeutas como sendo uma das mais saudáveis terapia em grupo, além de ser também, uma das atividades físicas mais prazerosas que ajuda a perder peso e a combater o stress.

Uma das vantagens do aplicativo é a diversidade de personais da dança disponíveis para um cliente contratar, bem como, a possibilidade de ser contratado a qualquer momento por um cliente. Por conseguinte, um Persona Dancer poderá ganhar mais dinheiro ao ser contratado por mais de um cliente numa noite e o

Cliente poderá aproveitar melhor a noite e a dança na companhia de mais de um Personal da dança devidamente capacitado.

Além das descrições já supracitadas, como praticidade, facilidade e prática saudável, o aplicativo reforçará a segurança dos usuários ao cobrar os honorários contratados do Personal antecipadamente garantindo a descrição que é a alma desse negócio e a segurança do prestador de serviços. Além disso, a segurança do cliente será resguardada, pois, só será permitido a contratação de Personal Dancer se ambos estiverem dentro do clube e do raio de alcance permitido para contratação.

Outra aplicação que faz parte do repertório de implementações a médio prazo é o rastreamento de todos os usuários do clube para saber a capacidade de público do momento para sugerir aos Personais Dancers os clubes que estejam no momento com pouco Personais para que assim, possam lucrar mais.

O projeto de pesquisa em estudo tem muito potencial, pois, os recursos tecnológicos na área de lazer e entretenimento tendem a crescer ao longo dos anos. Nesse viés, existem muitos videos tutoriais no youtube que ensina as pessoas a dançar, aplicativos com coreografias de danças e passos que ajudam no aprendizado, mas que carece de prática com uma pessoa que saiba conduzir de forma correta. Sob essa óptica, o aplicativo Personal Dancer tem um potencial imensurável, pois, em particular, a dança como atividade física terapeutica possui um vasto mercado em ampla expansão e carente de pessoas que possam desfrutar de alguns minutos ou horas em uma dança prazerosa, com um Personal Dancer detentor de um repertório exponencial de passos e coreografias.

No contexto da novidade inserida nos clubes, o projeto de estudo mostra que o aplicativo irá expandir os bons hábitos, bons costumes e os bons valores. Mostra-se que a dança é um hábito saudável milenar que ajuda a perder peso, melhorar a postura, reduzir o estres do dia a dia, evitar a depressão, melhorar o equilíbrio, estimular a memória e etc.

Fica claro, então, que o objetivo desse trabalho foi alcançado, já que, foi demonstrado o funcionamento do aplicativo em primeiro plano, através de recortes de tela de cadastro, login, contratação do cliente, aceitação da contratação pelo Personal, inicio da dança com contagem de tempo, fim do período contratado e encerramento da contratação.

## 5.1 DIFICULDADES ENCONTRADAS NO TRABALHO

Desde o início do curso, principalmente após o início da greve que pendurou por meses, além da dificuldade em si de programar, pois, não são todos que tem facilidade em programar. Diante desse cenário, juntando ao fato de termos necessidade de trabalhar em tempo integral, tive a dificuldade triplicada. Para piorar, meu Computador queimou a placa mãe e estou me virando como posso.

## 5.2 TRABALHOS E MELHORAS FUTURAS

Os próximos passos serão a realização de melhorias no aplicativo, no sentido de deixar mais prático possível ao cliente, reduzindo escrita e toques na tela, melhorar a segurança do Cliente e do Personal, restringindo a contratação ao âmbito interno do clube de forma mais eficaz e de acordo com o mapeamento presencial do clube.

Criar um projeto de divulgação nos diversos meios de comunicação, iniciando nos próprios clubes através de cartõzinhos, propagandas nas telas internas dos clubes, propagandas difundidas pelos artistas musicais mediante comissão, propagandas na internet em particular, youtube, facebook, instagram, twitter e etc, bem como, propagandas em rádios e televisão. Outra forma de divulgação alternativa, seria a realização de apresentações gratuitas de dança realizadas por Personais Dancers antes do início dos bailes e nos intervalos, acompanhados da apresentação do aplicativo no telão principal do clube.

Ademais, vamos dar uma atenção extra ao design do aplicativo, para que seja mais chamativo e convidativo. Por conseguinte, vamos apelar para a ciência cognitiva, fenomenologia da percepção, numerologia das cores e toda gama de informação sobre interface de aplicativos, para que o projeto possa vingar rapidamente.

## REFERÊNCIAS

**AGUIAR.** Tiago. Aprendendo Android do zero a google play. 2ª Edição, 2019.

**ALVES** da Silva. Luciano. Apostila Android -Programação passo a passo Edição FREE, 2015-RJ.

**FIREBASE.** Firebase Pricing.2019. Disponível em:

<https://firebase.google.com/pricing/?hl=pt-br>



**FURLAN**, J. D. Modelagem de Objetos através da UML. São Paulo, Makron Books, 1998.

**GOOGLE CLOUD**. Build an Android App Using Firebase and the App Engine Flexible Environment. 2019. Disponível em:

<https://cloud.google.com/solutions/mobile/mobile-firebase-app-engine-flexible?hl=pt-br>.

**GOOGLE DEVELOPERS**. App Fundamentals. 2018. Disponível em:

<https://developer.android.com/guide/components/fundamentals>

**GOOGLE DEVELOPERS**. Use o Android Jetpack para acelerar o desenvolvimento de aplicativos. 2018. Disponível em:

<https://developers-br.googleblog.com/2018/05/use-o-android-jetpack-para-acelerar-o.html>

**GOOGLE DEVELOPERS**. Maps SDK for Android. 2019. Disponível em:

<<https://developers.google.com/maps/documentation/android-sdk/intro>>

**LECHETA R.** Ricardo. Android Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 4ª Edição, Novatec, 2015, SP.

**LEE, V.; SCHNEIDER, H.; SCHELL, R.** Aplicações móveis: a arquitetura, projeto e desenvolvimento. São Paulo: Pearson, 2005.

**LUIZ TOOLS**. Fundamentos de Aplicações Android. 2011. Disponível em:

<http://www.luiztools.com.br/post/fundamentos-de-aplicacoes-android/>.

**MARÇAL, Edgar; ANDRADE, Rossana; RIOS, Riverson;** Novas Tecnologias na Educação. Aprendizagem utilizando Dispositivos Móveis com Sistemas de Realidade Virtual. V. 3, 2005.

**MONTEIRO B.** João. Google Android Crie aplicações para celulares e tablets. 1ª Edição 2012, SP.

**OGLIARI, Ricardo da Silva; BRITO, Robison Cris.** Android - do básico ao avançado. Rio de Janeiro: Editora Ciência Moderna Ltda., 2014.

**OLIVEIRA, Kalinka, BOSNIC, Ivan.** Software Engineering, 9th edition, publicada pela Pearson Education, Inc., sob o selo Prentice Hall. Traduzida.

**PARSE**. *Parse Server Guide*. 2019. Disponível em:

<https://docs.parseplatform.org/parse-server/guide/#getting-started>

**PRESSMAN, Roger S.** Engenharia de Software, 6 ed. São Paulo: McGraw Hill, 2006.

**PRESSMAN**, Roger S. **Engenharia de Software: Uma abordagem profissional**. 7 ed. São Paulo:ARTMED, 2011.

**RIBEIRO**, R.; **FREIRE**, P. Frameworks de Desenvolvimento Móvel Multiplataforma. In: Conferência da associação Portuguesa de sistemas de informação, 2013.

**SANTOS**, JÚNIOR, **PAULA**, Gesmar de. . Development geolocation system in augmented reality for mobile multiplatform. 2015. Dissertação (Mestrado em Engenharias) - Universidade Federal de Uberlândia, Uberlândia, 2015.

**SHI**, J. Handle FCM messages on Android. 2018. Disponível em:

<https://firebase.googleblog.com/2018/09/handle-fcm-messages-on-android.html>

**SMYTH**, Neil. Firebase Essentials – Android Edition First Edition 2017

## **APÊNDICES**

### **APÊNDICE A – Instruções de instalação do Android Studio**

Para instalar o Android Studio no Ubuntu e derivados, faça o seguinte:

**Passo 1.** Abra um terminal pressionando as teclas Ctrl+Alt+T.

**Passo 2.** Adicione o repositório com o seguinte comando:

```
sudo apt-add-repository ppa:paolorotolo/android-studio
```

**Passo 3.** Atualize o APT com o comando:

```
sudo apt-get update
```

**Passo 4.** Agora instale o programa com o comando:

```
sudo apt-get install android-studio
```

### **APÊNDICE B - Instruções de instalação do aplicativo para quando estiver disponível no play store.**

**Passo 1.** Acesse o play store → digite Personal dancer → Clique em instalar.

11:11 74% 2 1 Wi-Fi



Personal dancer



Personal Dancer - para  
clientes e pessoais

Technologies, Inc.

4,4 ★

2 mi avaliações ⓘ



65 MB

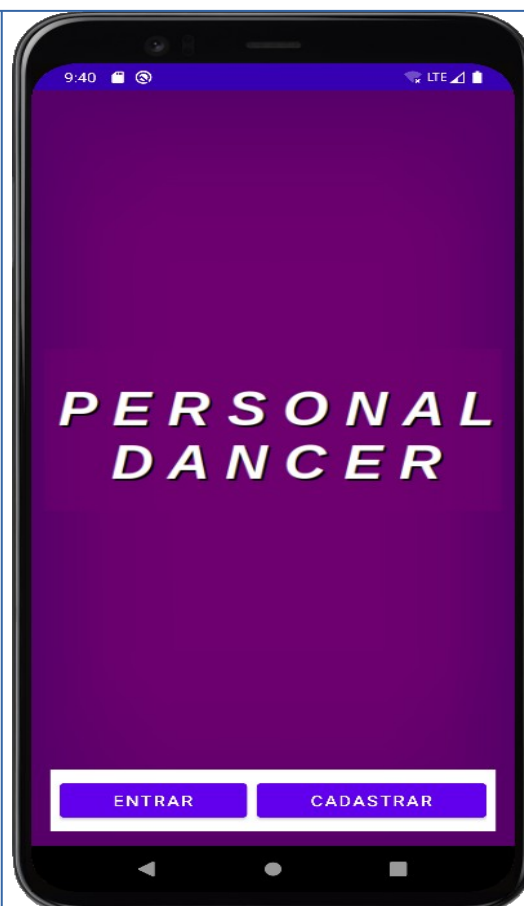
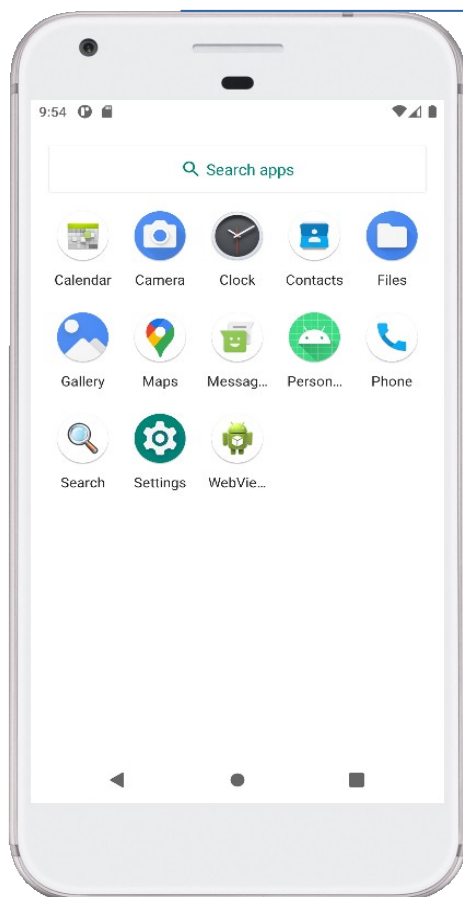


Classificação Livr

Instalar

## APÊNDICE C – Instruções de utilização do aplicativo

**Passo 1.** Abra o Aplicativo→ Clique em cadastrar.

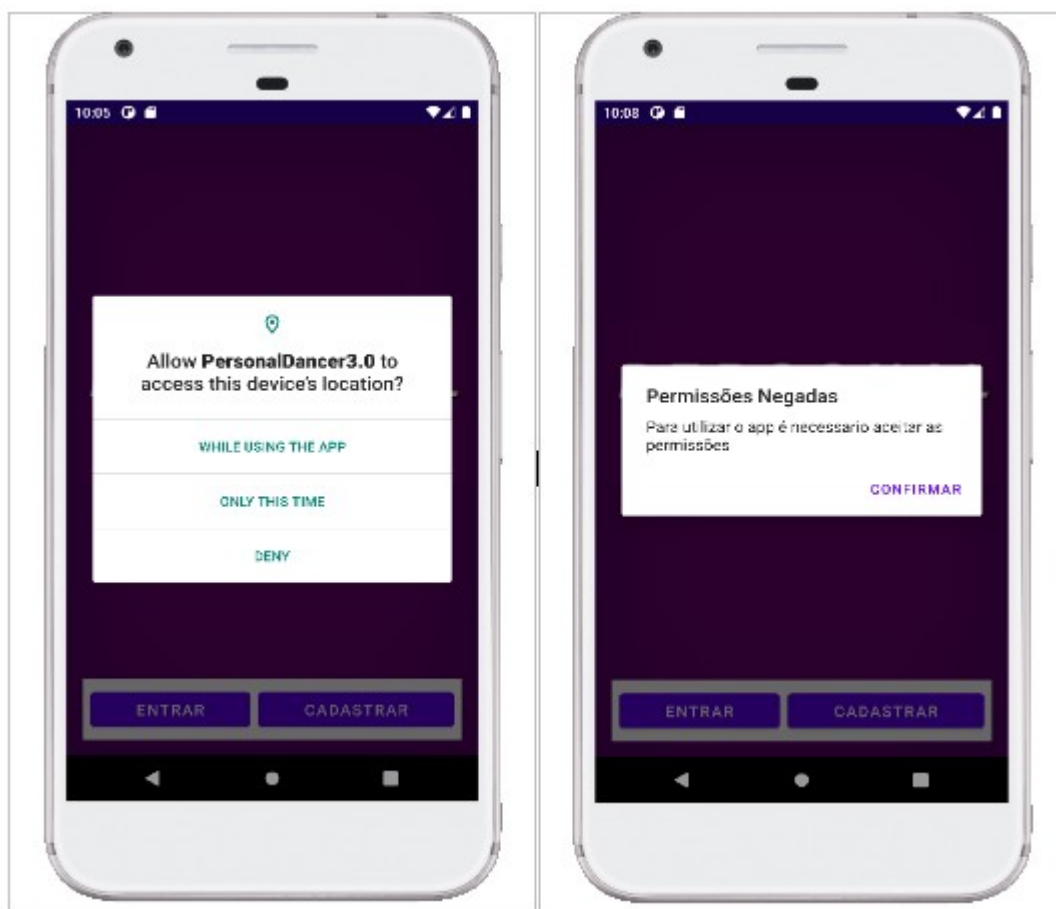


**Passo 2. Ativar GPS**

Para ter permissão para finalizar o cadastro é necessário ativar o GPS. Clique em 'While using the App'.

Se não ativar o GPS irá parecer na tela uma mensagem dizendo 'Permissões negadas'.

Figura 15: Ativar GPS



*Autor: autoria própria*

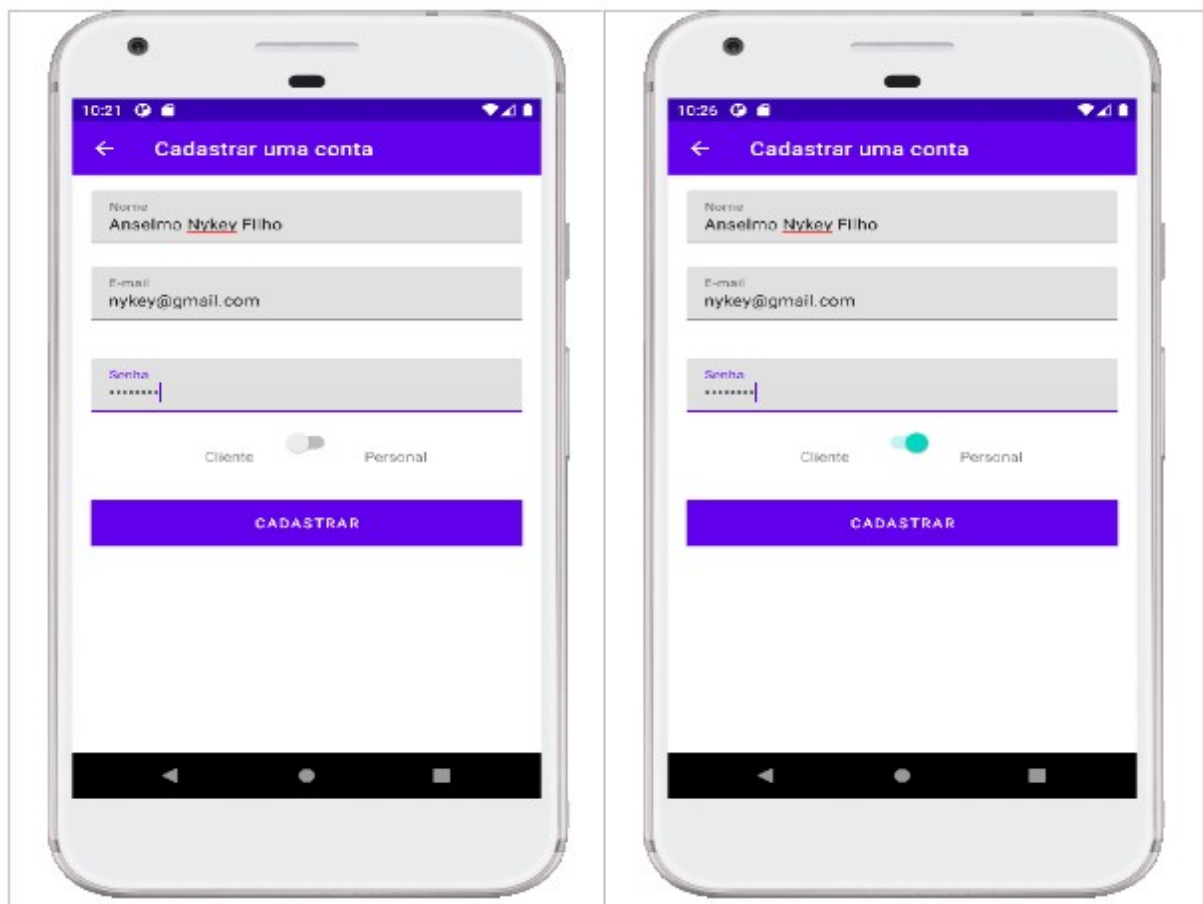
### **Passo 3. Cadastrar Usuário**

Para cadastrar Cliente puxa a bolinha de seleção para a esquerda e da mesma forma, o contrário, puxa a bolinha para direita para cadastrar Personal.

Se cadastrar o Personal a bolinha ficará na cor azul claro.

Se cadastrar o Cliente a bolinha ficará branca.

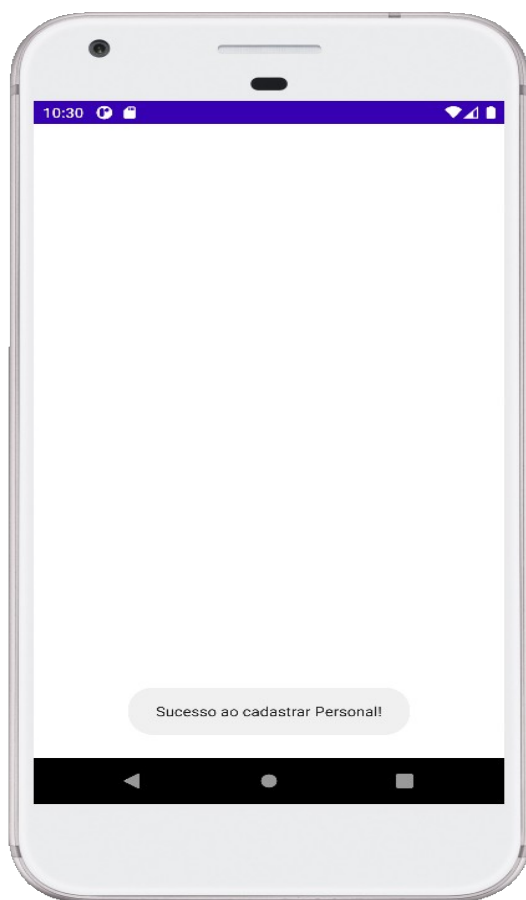
Figura 16: Cadastrar Usuário



*Fonte: autoria própria*

#### **Passo 4. Confirmação do Cadastro**

Se o cadastro for bem sucedido, aparecerá por 5 segundos uma mensagem dizendo: “ Sucesso ao cadastrar Cliente”, caso seja o cliente e “Sucesso ao cadastrar Personal, caso seja o Personal”. Logo abaixo, uma imagem para validar melhor.



## **Passo 5. Logar no Aplicativo**

**Passo 6. Tela para complementar o cadastro**



O usuário pode cadastrar, alterar e excluir a conta. Essa parte está para ser melhorada, pois pretendo fazer alguns ajustes.

Ainda vou criar um seletor para o usuário pode selecionar a altura, o peso e a data de nascimento. Isso tudo para facilitar o preenchimento final do cadastro e ajustes, evitando a digitação do usuário.

O ajuste principal que vou elaborar diz respeito as novas leis da LGBT. É importante ressaltar que a identidade de gênero refere-se ao gênero que reconhecemos pertencer (masculino, feminino ou não binário), independente do sexo designado ao nascer (FREIRE, COSTA, MARQUES, 2016).

O aplicativo vai focar apenas na identidade de gênero, pois, o objetivo do aplicativo é facilitar o acesso ao maior número de pessoas possíveis na dança de salão. Nesse contexto, o aplicativo não levará em conta a orientação sexual que está relacionada com a atração sexual e que define os tipos de orientação sexual como sendo: heterossexual, homossexual, bissexual e pansexual.

Portanto, dentro do contexto da identidade de gênero, vou pensar e estudar a hipótese de criar um seletor com as seguintes opções: masculino, feminino, mulher trans e homem trans.



Autor: autoria própria

### **Passo 7. Interface para selecionar o clube**

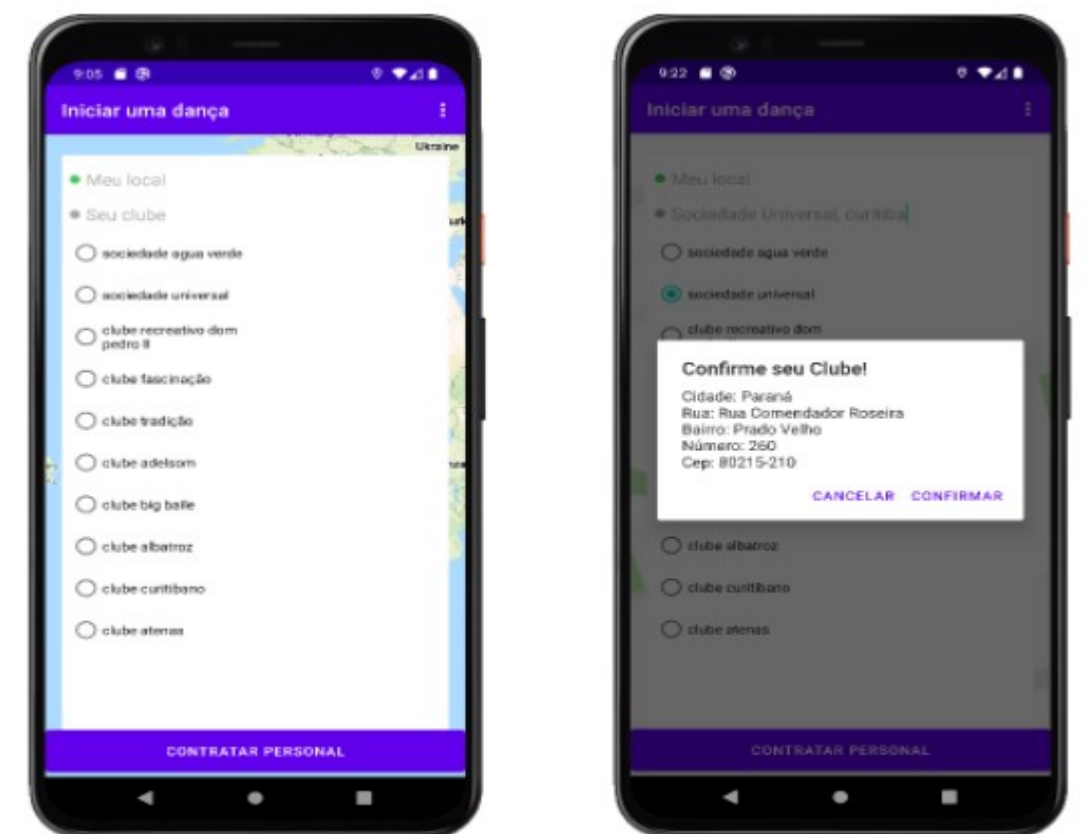
Somente o cliente deve escolher o clube que está no momento, e após isso, clicar em contratar Personal.

Logo após, aparecerá uma mensagem pedindo para confirmar o clube selecionado e se estiver de acordo, clicar em confirmar.

Se clicar em cancelar retornará para tela anterior de escolha do clube.

O Personal Dancer não precisa escolher o clube, basta apenas que faça o login no aplicativo e aguarde o recebimento de alguma solicitação de contratação.

Figura 20: Interface para selecionar Clube



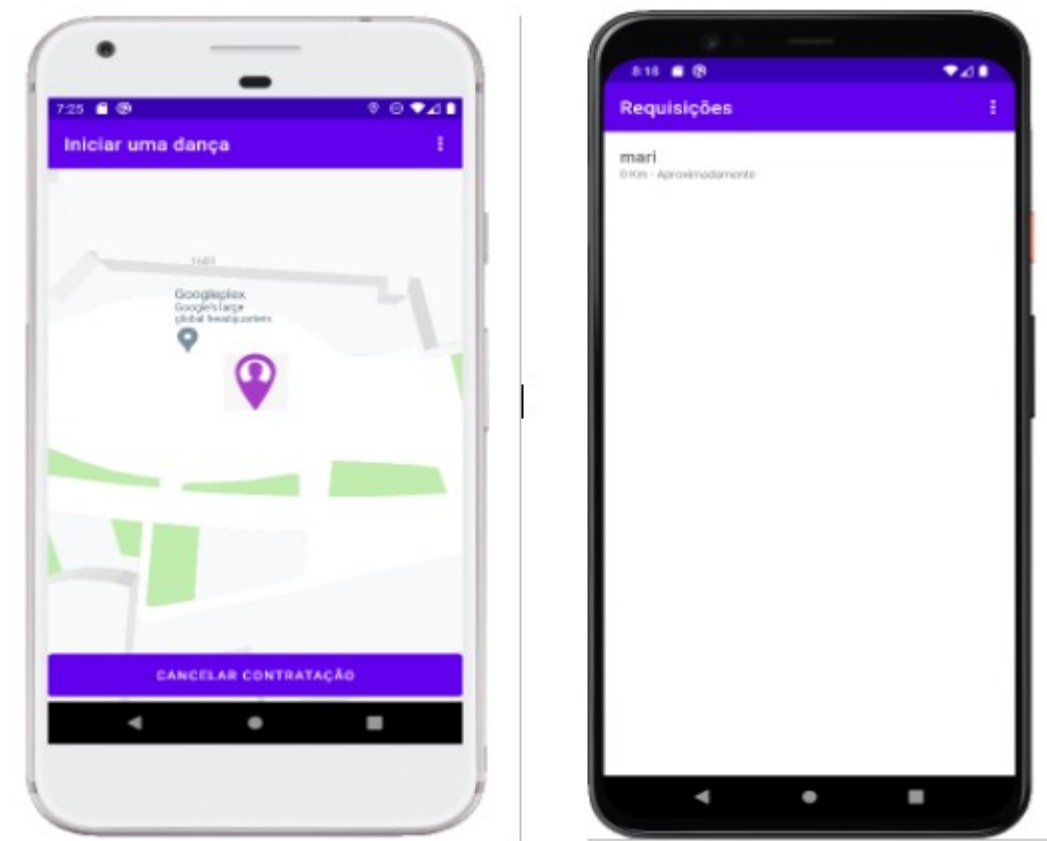
*Autor: autoria própria*

### **Passo 8. Interface para aceitar ou rejeitar contratação**

A interface de requisições de contratação aparecerá apenas para o Personal Dancer.

O Personal deve clicar em cima do nome do solicitante para decidir se aceita a contratação. Após clicar em cima do nome do solicitante, uma mensagem será enviada ao Cliente informando que o Personal está à caminho. Após o Personal se apresentar ao cliente poderá clicar em aceitar contratação e do mesmo modo, o cliente poderá clicar em cancelar a contratação.

Figura 21: Interface para aceitar ou rejeitar contratação

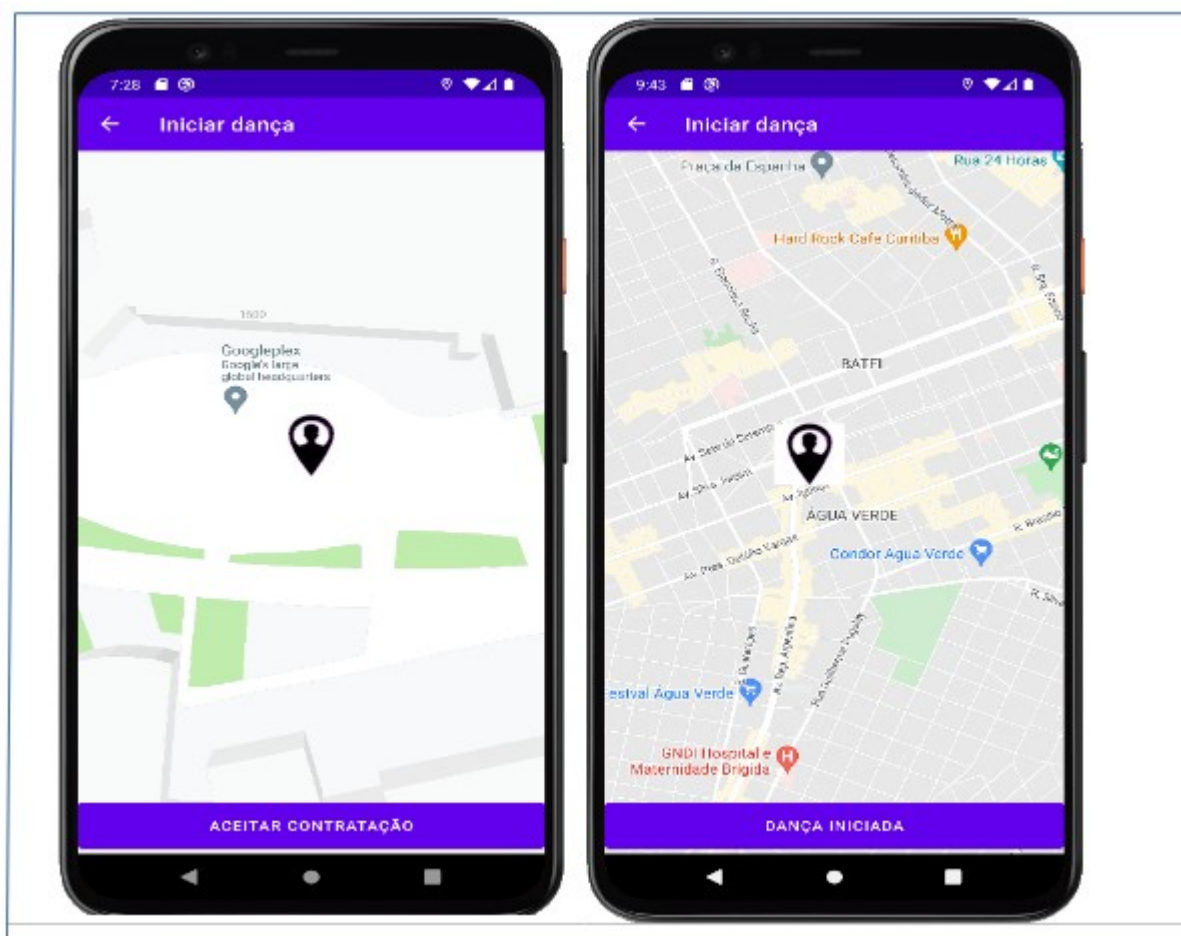


*Autor: autoria própria*

### **Passo 9. Personal aceita contratação e inicia a dança**

Após o cliente clicar em aceitar a contratação, o status da contratação altera para dança iniciada.

Figura 22: Interface dança iniciada

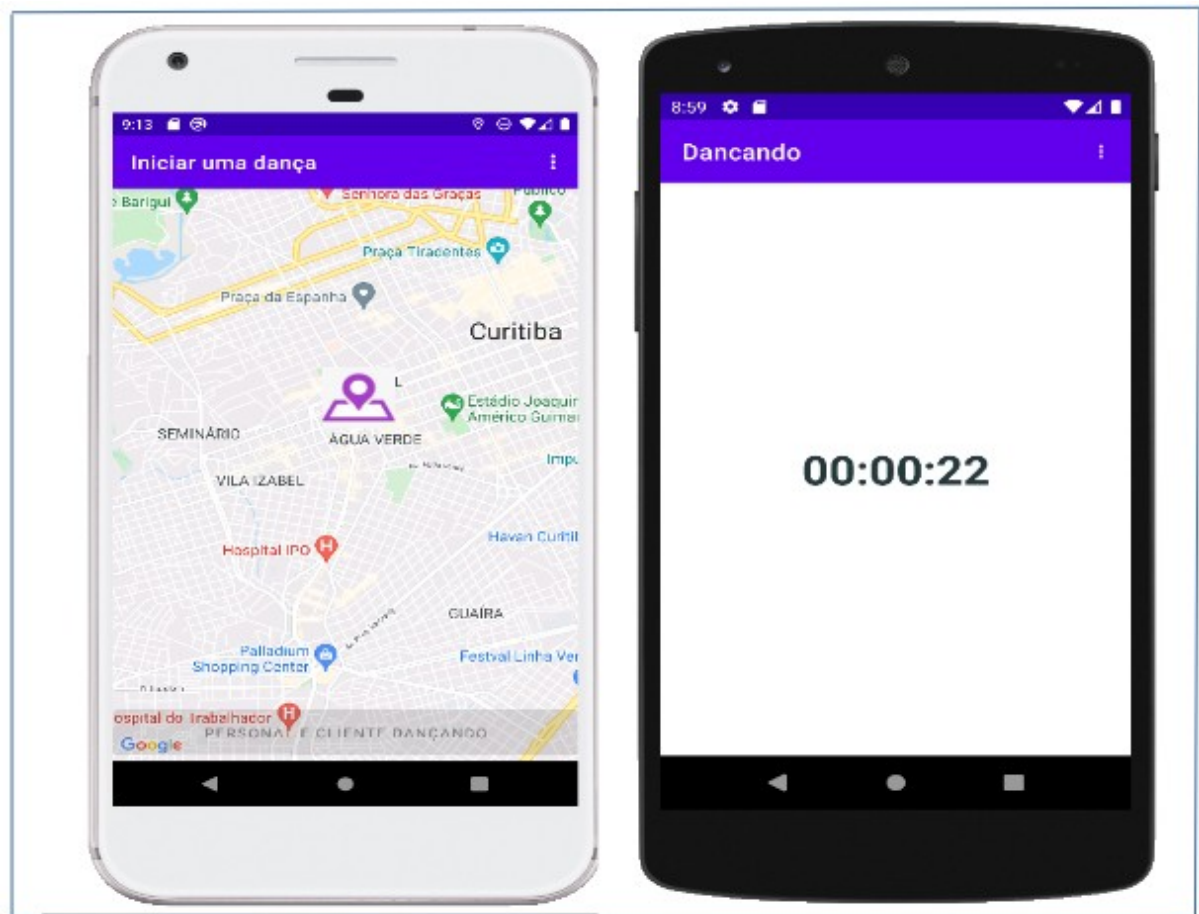


*Autor: autoria própria*

## **Passo 10. Personal e cliente dançando**

Após o cliente clicar em iniciar dança, o status do Personal é alterado para dançando e inicia a contagem do período de tempo contratado.

Figura 23: Início da contagem do tempo

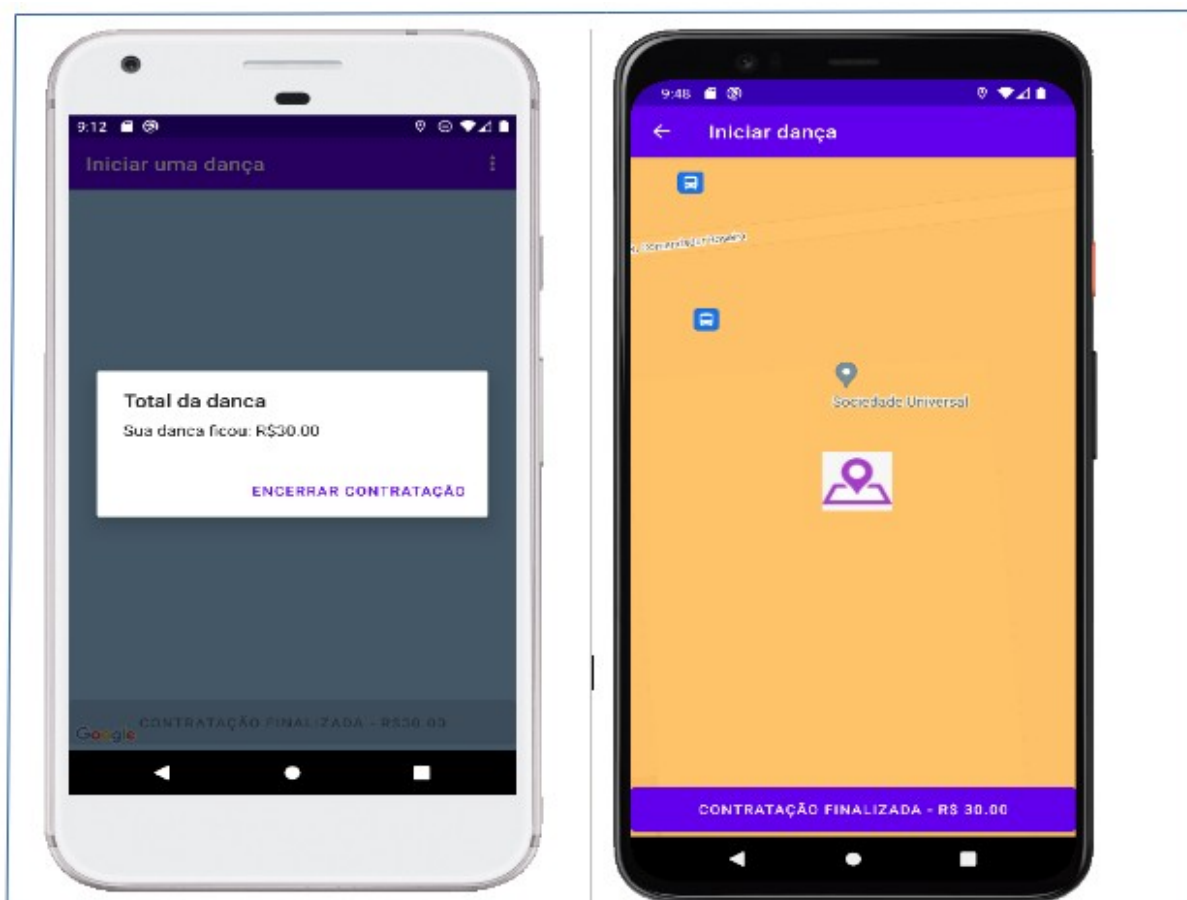


*Autor: autoria própria*

**Passo 11. Contratação finalizada**

Após encerrar o período contratado, o status é alterado para contratação finalizada e uma mensagem é emitida ao Cliente informando que a dança encerrou junto com o valor total da dança que automaticamente é debitada do cartão de crédito do cliente. O Cliente deve clicar em encerrar contratação para recontratar o Personal ou contratar outro Personal disponível. Por conseguinte, o Personal deve clicar em contratação finalizada para ficar disponível a outros clientes.

Figura 24: Contratação finalizada

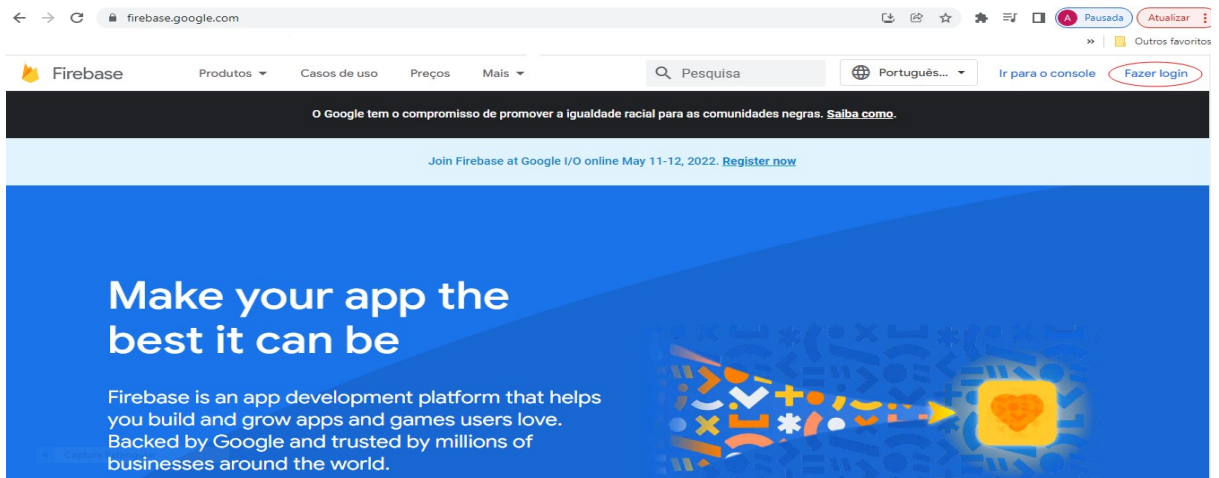


tor: autoria própria

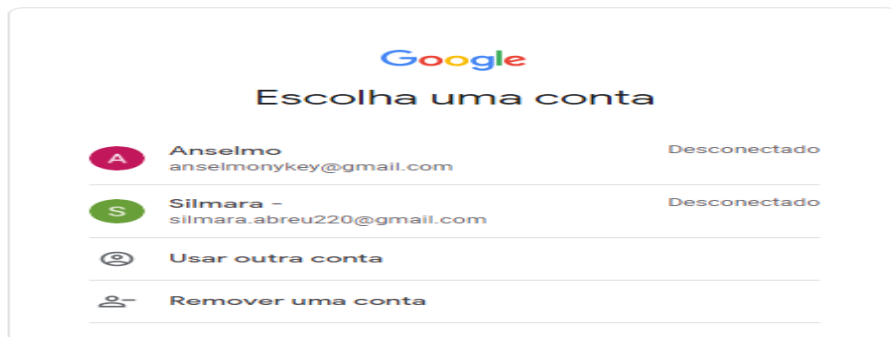
Au

## APÊNDICE D – Configuração do banco de dados Firebase

1- Acesse o site do Firebase→ [firebase.google.com](https://firebase.google.com)→ clique em fazer login

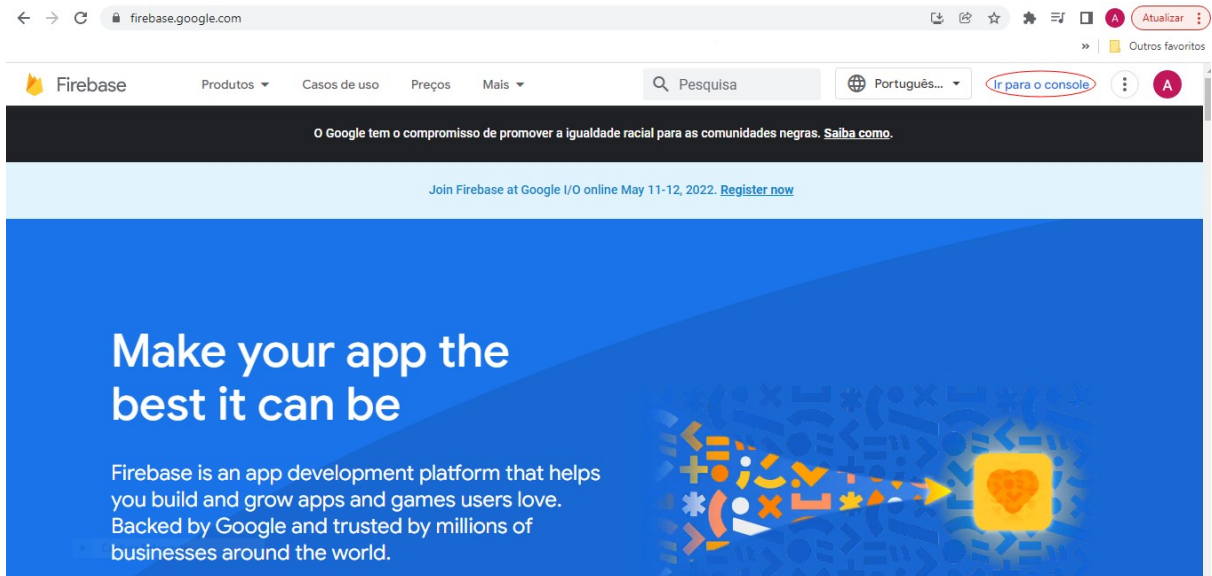


2-Faça o Login→ entre com sua conta do google→ se não tiver, crie uma conta

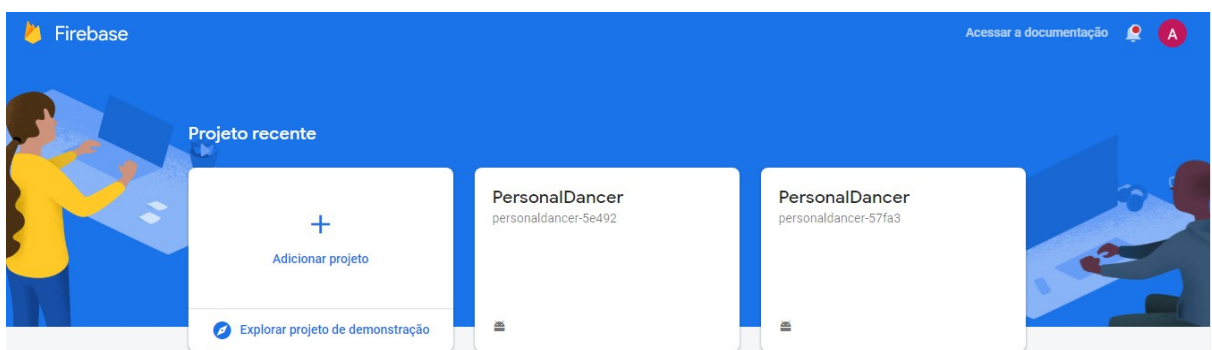


3-Acesse o Console→ clique em ir para o console





4-Crie um Projeto→ clique em adicionar projeto



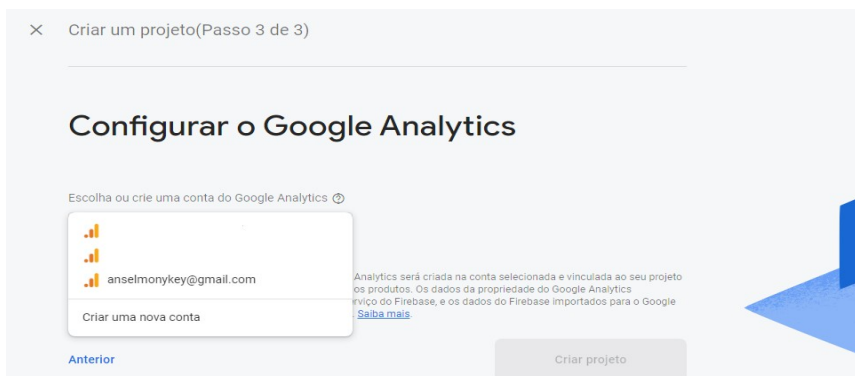
5-Dê um nome ao projeto→ escolha um nome, pode ser qualquer nome.



## 6-Clique em continuar



## 7- Escolha a mesma conta que usou no Login




## 8- Criar projeto→ clique em criar projeto.

×

Criar um projeto(Passo 3 de 3)


## Configurar o Google Analytics

Escolha ou crie uma conta do Google Analytics ?



anselmonykey@gmail.com

▼

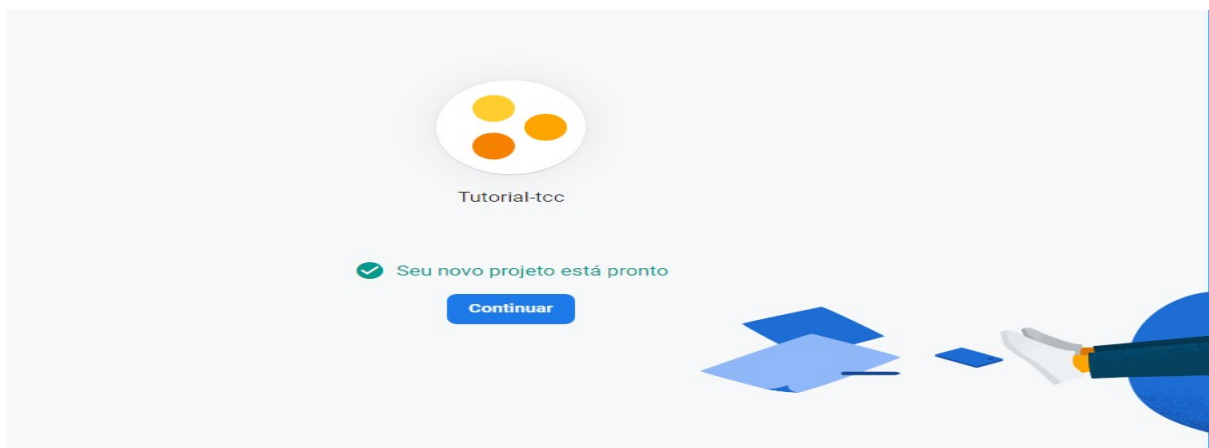
Criar automaticamente uma nova propriedade nesta conta 

Após a criação do projeto, uma nova propriedade do Google Analytics será criada na conta selecionada e vinculada ao seu projeto do Firebase. Esse processo permitirá o fluxo de dados entre os produtos. Os dados da propriedade do Google Analytics exportados para o Firebase ficam sujeitos aos Termos de Serviço do Firebase, e os dados do Firebase importados para o Google Analytics ficam sujeitos aos Termos de Serviço do Analytics. [Saiba mais](#)

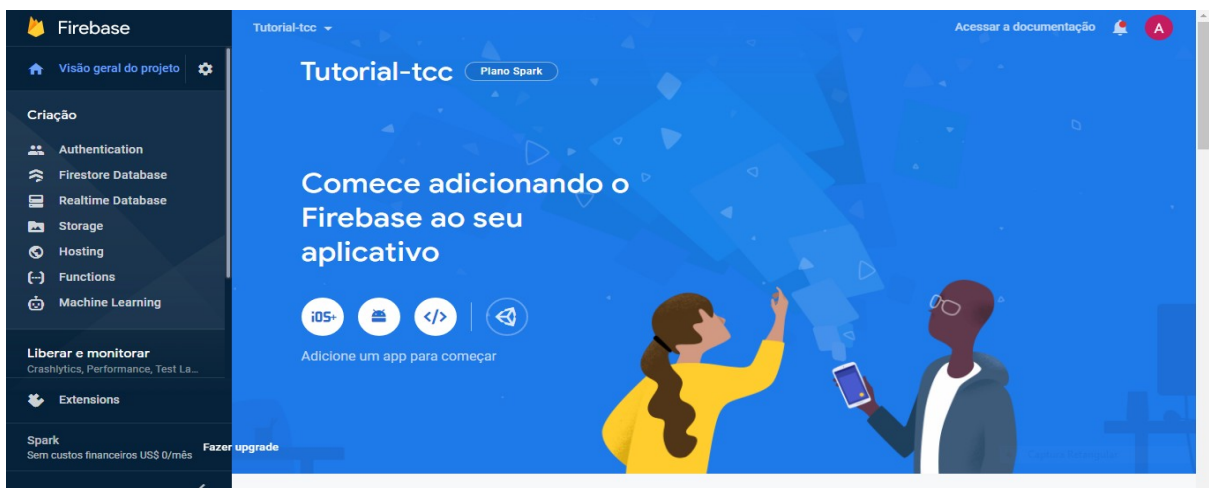
Anterior

Criar projeto

## 9- Projeto pronto



## 10- Configurar projeto - Clique em Android




11- Registrar App→ clique em registrar app

[illegible]

12- Fazer o download de google-services.json e salve no diretório raiz do módulo do app Android e clique em próxima.


✓ **Registrar app**  
Nome do pacote Android: com.example.tutorial, apelido do app: Tutorial TCC

2 **Fazer o download do arquivo de configuração**      Instruções para o Android Studio abaixo | [Unity](#) [C++](#)

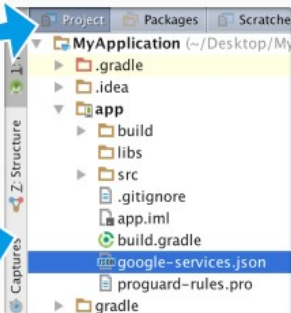
 **Fazer o download de google-services.json**

Mude para a visualização do **Projeto** no Android Studio para ver o diretório raiz.

Mova o arquivo google-services.json que você acabou de salvar para o diretório raiz do módulo do app Android.



google-services.json



13- Em build.gradle no nível do projeto adicione o class path em dependências

build.gradle no nível do projeto (<project>/build.gradle):

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.10'
    }
}

allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}
```

☒ Java ☐ Kotlin

14- Em build.gradle no nível do app adicione implementation platform em dependências.

build.gradle no nível do app (<project>/<app-module>/build.gradle):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:29.3.1')

    // Add the dependency for the Firebase SDK for Google Analytics
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebase:firebase-analytics'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

Ao usar uma lista de materiais (BoM, na sigla em inglês) do Firebase para Android, o app sempre usará versões da biblioteca do Firebase compatíveis. [Saiba mais](#)

15-Clique em continuar no console

✓

Registrar app

Nome do pacote Android: com.example.tutorial, apelido do app: Tutorial TCC

✓

Fazer o download do arquivo de configuração

✓

Adicionar o SDK do Firebase

4

Próximas etapas

Está tudo pronto!

Certifique-se de verificar a [documentação](#) para saber como começar a usar os produtos Firebase que quiser usar em seu aplicativo.

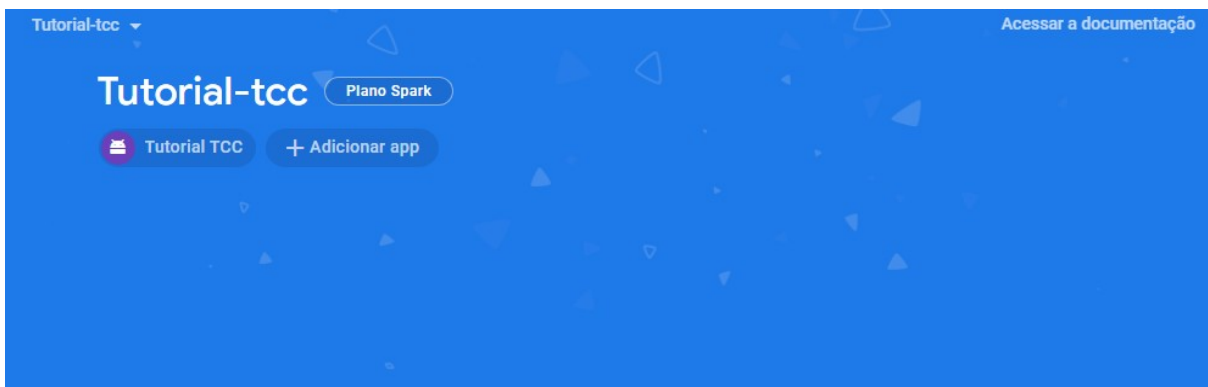
Também é possível explorar [apps de amostra do Firebase](#).

Como opção, continue no console para explorar o Firebase.

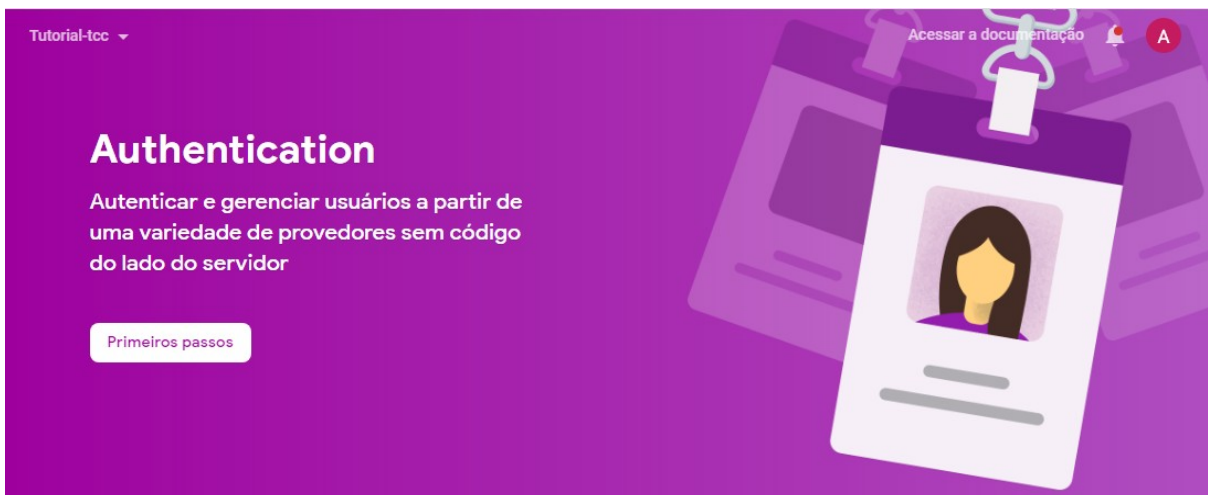
Anterior

Continuar no console

## 16-Projeto configurado-Clique a esquerda em Authentication



## 17-Clique em primeiros passos



## 18- Primeiros passos para Android

Codelabs

## 21-Lista de dependências necessárias do gradle


Realtime Database	com.google.firebase:firebase-database	20.0.4
Configuração remota	com.google.firebase:firebase-config	21.0.2
Plug-in do Google Play Services <a href="#">🔗</a>	com.google.gms:google-services	4.3.10
Bibliotecas suspensas		
Indexação de apps	com.google.firebase:firebase-appindexing	20.0.0
APIs do modelo personalizado do Firebase ML	com.google.firebase:firebase-ml-model-interpret	22.0.4
APIs do Firebase ML Vision	com.google.firebase:firebase-ml-vision	24.1.0
Firebase ML: modelo de rotulagem de imagens	com.google.firebase:firebase-ml-vision-image-label-model	20.0.2
Firebase ML: modelo de rastreamento e detecção de objetos	com.google.firebase:firebase-ml-vision-object-detection-model	19.0.6
Firebase ML: modelo de detecção facial	com.google.firebase:firebase-ml-vision-face-model	20.0.2
Firebase ML: modelo de leitura de código de barras	com.google.firebase:firebase-ml-vision-barcode-model	16.1.2

## 22-Configurar método de Login

Tutorial-tcc
Acessar a documentação
A
?

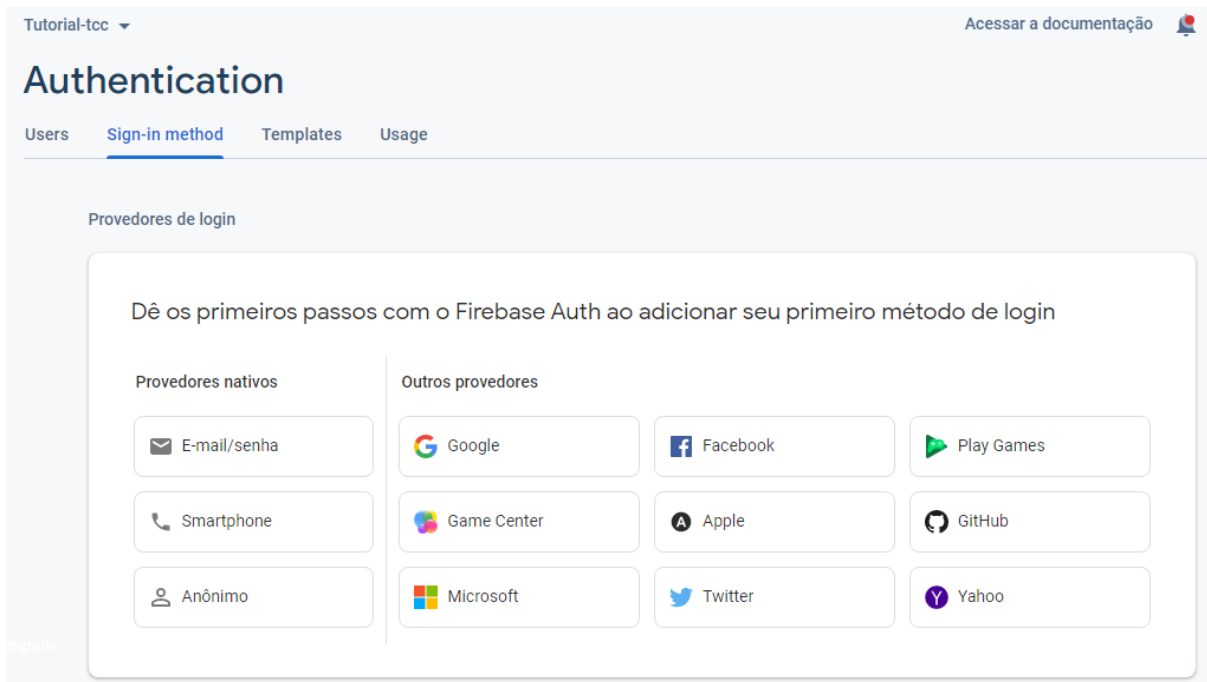
Authentication
Users
Sign-in method
Templates
Usage

Adicionar usuário
↻
⋮

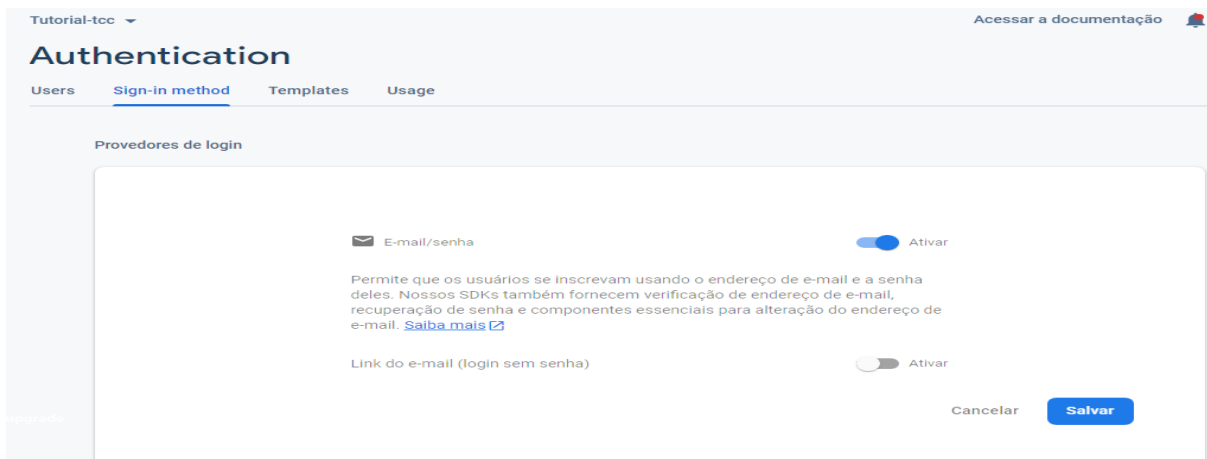
Identificador	Provedores	Data de criação	↓	Último login	UID do usuário
<div>  <div> Autenticar e gerenciar usuários a partir de vários provedores sem um código do servidor <a href="#">Saiba mais</a> <a href="#">Ver os documentos</a> </div> <div>Configurar método de login</div> </div>					

## 23- Em Sign-in method clique E-mail/senha





24- Marque ativar E-mail/senha e clique em salvar.



25-Status ativado

