

SOFTWARE COMO FERRAMENTA PARA GERENCIAMENTO DO RELACIONAMENTO INTERPESSOAL

Maurício Andrade Gomes ¹

RESUMO

A fluidez dos relacionamentos diverge da definição intrínseca de sociedade, descrita como um conjunto de indivíduos que juntos buscam o bem comum. Todos possuem uma necessidade de pertencer, sendo essa uma motivação para estabelecer vínculos, a qual não deve ser frustrada. Além de os relacionamentos serem algo natural para o desenvolvimento humano, os seus benefícios são inúmeros e favorecem tanto a saúde física como também cognitiva. Devido a isso, foi desenvolvido um software que visa facilitar e auxiliar no desenvolvimento de relacionamentos interpessoais.

Palavras-chave: Relacionamentos Interpessoais; Software; Psicologia Social.

ABSTRACT

The fluidity of relationships diverges from the intrinsic definition of society, described as a set of individuals who together seek the common good. Everyone has a need to belong, which is a motivation to establish bonds, which should not be frustrated. In addition to relationships being something natural for human development, their benefits are countless and promote both physical and cognitive health. Because of this, software was developed that aims to facilitate and assist in the development of interpersonal relationships.

Keywords: Interpersonal Relationships; Software; Social Psychology.

1 INTRODUÇÃO

As relações interpessoais de caráter sólido e saudável, são a base de uma sociedade bem estruturada e segundo o conceito que a define. A busca pelo bem comum, torna os desejos e anseios pessoais pertencentes ao todo, onde o objetivo principal é o bem-estar mútuo da vida social. No entanto, o momento atual é caracterizado pelo individualismo que prioriza o bem-estar próprio acima do coletivo, mesmo que isso gere a ruína do próximo. Esse fato torna as relações fluídas e por consequência gera sociedades instáveis.

Com o contexto social atual bem definido, o presente projeto visa entender e descrever como a Computação pode auxiliar no processo do desenvolvimento das relações interpessoais. Logo, a pesquisa trata sobre a importância e as características das relações, como também a relevância que a Computação possui como ferramenta para auxiliar no desenvolvimento dessas interações.

¹ Discente do Curso de Ciências da Computação da Universidade La Salle - Unilasalle, matriculado na disciplina de TCC 2 sob a orientação Prof. Aline Duarte Riva. E-mail: aline.riva@unilasalle.edu.br. Data de entrega:

Em relação ao que tange o conceito de sociedade, a mesma pode ser definida como um conjunto de indivíduos que se relacionam entre si e que juntos buscam alcançar o bem comum (DALLARI, 2009). O bem comum, por sua vez, possibilita e favorece o desenvolvimento integral da personalidade humana, não apenas individualmente, mas coletivamente. Portanto, torna-se evidente a relevância social que bons relacionamentos proporcionam, já que os mesmos visam o bem-estar mútuo entre os participantes. Contudo, quando não ocorre a prática do desenvolvimento desses relacionamentos, produzem-se sociedades afastadas dos objetivos que justificam sua existência.

Tendo em vista o motivo que origina uma sociedade, surge então a questão relacionada àquilo que pode ser proporcionado caso a mesma desenvolva-se. Em relação a esse fato, os relacionamentos interpessoais estão interligados com a saúde física e mental de seus participantes (UMBERSON; MONTEZ, 2010), tornando-se um agente regulador dessas áreas. Logo, bons relacionamentos direcionam a sociedade ao bem comum, favorecendo os objetivos individuais do todo, como também concedendo o seu bem-estar físico e emocional.

O avanço da tecnologia, em conjunto com o processo de globalização, possibilitou que barreiras, tais como a distância, fossem quebradas, permitindo que a comunicação ocorresse independente do local, horário ou circunstância. Dessa forma, torna-se evidente a relevância social que a tecnologia possui, onde a mesma pode ser utilizada como ferramenta para alcançar o desenvolvimento de bons relacionamentos.

Tendo em mente a importância de se relacionar, o presente projeto possui como temática a questão relacionada àquilo que a tecnologia pode proporcionar para auxiliar nas interações interpessoais. Nesse sentido, a escolha da tecnologia como ferramenta se dá visto que a mesma já se mostrou eficiente em aproximar os indivíduos e auxiliar na construção da relação entre eles.

A justificativa do desenvolvimento desse trabalho se dá com base na ideia de que com a contemporaneidade, os relacionamentos tornaram-se fluídos, instáveis e superficiais, possuindo um caráter consumista, onde o prazer se dá na possibilidade de se desfazer daquilo que não satisfaz mais ao indivíduo (RODRIGUES; PORTO, 2021). Isso não está limitado somente ao âmbito amoroso, mas ocorre também nas amizades, no trabalho e em comunidades em geral. Portanto, o contexto atual é caracterizado por relacionamentos instáveis e individualistas, onde o prazer pessoal é o que motiva as relações.

Ainda sobre a fluidez dos relacionamentos, a mesma pode gerar complicações tanto físicas quanto psíquicas. Esse fato é intensificado no relacionamento conjugal, onde a relação não funcional possui influências diretas e indiretas nessas áreas (BOYD; SAMID, 2001). Com isso, pode-se concluir que a falta do desenvolvimento de bons relacionamentos pode levar a problemas de saúde.

Relacionamentos saudáveis e ativos são importantes para o desenvolvimento humano, devido o fato de os laços sociais influenciarem os hábitos relacionados à saúde (UMBERSON; MONTEZ, 2010). Isso ocorre porque o apoio que uma comunidade, amigos ou familiares proporcionam é de extrema relevância para o desenvolvimento pessoal, não somente por questões de saúde, mas também por motivos naturais do ser humano, os quais convergem na necessidade de viver em sociedade.

Com o passar do tempo, surgiram diversas formas diferentes de desenvolver

um relacionamento, fruto de um crescimento tecnológico exponencial, como também do processo de globalização. Com isso, até mesmo a comunicação foi impactada, diminuindo as barreiras físicas antes impostas, devido à longa distância entre os indivíduos. Entretanto, mesmo com o advento da tecnologia e o crescente uso de redes sociais, as relações dificilmente evoluem forma saudável e genuína (CAMPISI et al., 2012). Dessa forma, torna-se necessária a existência de alguma forma de alcançar esse objetivo.

Surge então a questão problema que originou o presente projeto, a qual é caracterizada pela seguinte indagação: **Como a Computação pode auxiliar no processo do desenvolvimento de relacionamentos interpessoais?**

2 DESENVOLVIMENTO

Essa seção é destinada a expor as pesquisas bibliográficas realizadas utilizadas como base para o desenvolvimento do produto. Além da exposição teórica, a seção também apresenta as funcionalidades do software e de que forma o mesmo será validado.

2.1 Relacionamento interpessoal

Segundo a Psicologia Social, uma relação interpessoal trata-se da conexão entre duas ou mais pessoas. Como essa conexão ocorre, é o foco de estudo dessa área, definida como a ciência que estuda o pensamento das pessoas e como elas se influenciam e relacionam-se entre si (MYERS, 2014).

Um conceito importante no estudo das relações sociais é a **necessidade de pertencimento**. Essa necessidade é definida como a motivação para estabelecer vínculos, proporcionando interações positivas e constantes (MYERS, 2014). Ao frustrar essa motivação, como ocorre no ostracismo social, ocorre um impacto negativo sobre a vida emocional dos envolvidos, como também de seus processos cognitivos (BAUMEISTER; LEARY, 1995).

Em uma relação entre duas pessoas quaisquer, a proximidade assume o papel de um excelente preditor de um bom relacionamento entre ambas (MYERS, 2014). Isso ocorre devido à disponibilidade e conveniência em interagir, dada uma distância confortável para tal ação. Portanto, estar próximo fomenta o gostar, visto que se torna facilitada a relação entre os abrangidos (BACK; SCHMUKLE; EGLOFF, 2008). Logo, pessoas que residem proximamente umas as outras, possuem uma probabilidade maior de desenvolverem bons relacionamentos.

A distância funcional, na qual os caminhos das pessoas se cruzam, também possui sua relevância para o entrosamento entre as partes (MYERS, 2014). Essa interação proporciona a exploração das semelhanças entre os envolvidos, além de permitir sentir o gostar mútuo, levando à percepção do pertencimento a uma unidade social (ARKIN; BURGER, 1980).

Além da proximidade, a simples expectativa de interagir com alguém, também estimula o gostar e auxilia no desenvolvimento das relações (MYERS, 2014). Sendo assim, criar a esperança de que uma relação seja agradável, aumenta as chances do desenvolvimento de um relacionamento satisfatório, fornecendo um instrumento para facilitar esse processo (KNIGHT; VALLACHER, 1981).

Estar familiarizado proporciona o afeto referente àquilo que somos repetidamente expostos (MYERS, 2014). Quanto mais as pessoas forem referidas umas as outras, maior será o sentimento agradável entre elas (HARMON-JONES; ALLEN, 2001). Conseqüentemente, quanto maior o tempo investido nos relacionamentos e quanto maior a persistência em desenvolvê-los, maior será o resultado benéfico para essas relações.

Relacionamentos saudáveis e bem desenvolvidos são formados por pessoas que compartilham atitudes, valores, crenças e gostos semelhantes. Quanto mais parecidas forem as características e atitudes entre duas pessoas quaisquer, maior será a consideração em relação ao quão simpáticas apresentam-se ser uma à outra (BYRNE; CLORE; SMEATON, 1986). O fato de pensar parecido não gera apenas apreciação, mas também produz inferências positivas sobre o caráter, elevando as chances do gostar mútuo (MONTROYA; HORTON, 2004).

A semelhança entre as partes, instiga a interação e afeição, gerando contentamento. Isso resulta no fato de que a semelhança leva os indivíduos a gostar (MYERS, 2014). Entretanto, as diferenças reduzem mais o gostar do que as atitudes comuns o aumentam (SINGH; HO, 2000). Nesse sentido, pessoas que sejam mais parecidas, possuem uma maior probabilidade de desenvolverem bons relacionamentos.

2.2 Conhecimento técnico

O produto a ser gerado através do presente projeto terá como base o desenvolvimento orientado ao domínio, buscando construir o software de forma que o mesmo reflita os conceitos da área em que será utilizado. Nesse sentido, o domínio do software pode ser definido como a área ou assunto no qual o software é aplicado (EVANS, 2004), expondo todo o contexto necessário para a sua criação.

Para que o domínio do software possa ser representado de forma fiel e transparente, surge a necessidade de desacoplá-lo de outras partes necessárias do software, tais como o acesso à base de dados ou mesmo o tratamento das requisições HTTP. Isso deve ocorrer para o foco estar totalmente no domínio e não em questões de implementação. Assim sendo, os princípios SOLID podem auxiliar nesse processo, guiando o desenvolvimento para a utilização de uma arquitetura de software que suporte a separação de responsabilidades em camadas (MARTIN, 2017), como ocorre na Arquitetura Limpa.

O produto seguirá algumas das ideias do padrão arquitetural REST, ao tratar os dados de retorno como representações de recursos, tornando-os também navegáveis (FIELDING; TAYLOR, 2000). Para isso, será utilizado como base o trabalho em andamento relativo a um tipo de mídia para representação de recursos e suas relações por meio de hiperlinks (KELLY, 2023).

2.3 Funcionalidades do produto

A primeira versão do produto a ser gerado, visa fornecer auxílio ao contexto inicial de um relacionamento. Suas definições de domínio foram baseadas no estudo exposto na Seção 2.1, a qual define a proximidade como um excelente preditor das relações. Logo, o objetivo do projeto é facilitar o processo inicial do desenvolvimento de relacionamentos e não em mantê-los.

As sugestões fornecidas pelo software, são baseadas nos estudos da Psicologia Social. Por se tratarem de sugestões, não devem ser interpretadas como imposições. As informações fornecidas visam utilizar a Computação como ferramenta para auxiliar e facilitar o desenvolvimento de boas relações e não para criá-las. Logo, o conhecimento fornecido pelo produto é baseado em estudos da área da psicologia e não deve ser entendido como um recurso definitivo para desenvolver relacionamentos, apenas como uma ferramenta para auxiliar nesse processo.

As funcionalidades que o produto proporciona são relativas à proximidade. Essa, por sua vez, é composta pela distância geográfica e a distância funcional. Essas funcionalidades são exteriorizadas mediante uma Interface de Programação de Aplicações, definida como um conjunto de recursos predefinidos que podem ser acessados por outras aplicações de forma simplificada (MENG; STEINHARDT; SCHUBERT, 2018).

O produto fornecerá inicialmente somente a possibilidade de incluir e consultar dados. Isso se dá devido o fato de o sistema tratar-se de um mínimo produto viável, direcionando o foco para a exposição das funcionalidades e não para os detalhes de sua implementação. Sendo assim, a necessidade da aplicação é satisfeita apenas com capacidade de inserir e consultar dados.

A primeira funcionalidade, baseada na proximidade, fornece um conjunto de pessoas candidatas ao desenvolvimento de um relacionamento, dada uma distância geográfica limite entre o local de habitação dos envolvidos. Sendo assim, a proximidade geográfica disponibiliza o conjunto de pessoas próximas fisicamente do local de habitação da pessoa que consumir essa funcionalidade.

A distância geográfica entre duas pessoas pode ser medida através da Fórmula de Haversine (MARIA et al., 2020), como pode ser observado na Figura 1. Essa fórmula pode calcular a distância de dois pontos quaisquer, uma vez que sejam fornecidas as suas coordenadas de latitude e longitude. Dessa forma, possuindo a distância entre dois pontos, é possível determinar um valor limite de distância, definido como o raio máximo em quilômetros, que seja confortável para quem consumir essa funcionalidade.

A segunda funcionalidade, também baseada na proximidade, fornece um conjunto de pessoas candidatas ao desenvolvimento de um relacionamento, dada uma atividade comum que possibilite a interação em um mesmo local, período e em ao menos um dia. Logo, a proximidade funcional disponibiliza o conjunto de pessoas próximas funcionalmente, ou seja, o conjunto de pessoas que tenham alguma atividade em comum, onde seus caminhos possam cruzar-se, gerando a disponibilidade para a interação.

Uma atividade, utilizada como base para a funcionalidade de proximidade funcional, é formada por um endereço, um período e um conjunto de dias na semana. Sendo assim, para que duas pessoas sejam próximas funcionalmente, torna-se necessário que suas atividades coincidam em relação ao ambiente, período e ao menos um dia na semana, como pode ser observado na Figura 2.

2.4 Validação do produto

(PRESSMAN, 2011) define o teste de software como um conjunto de atividades que podem ser projetadas com antecedência e executadas sistematicamente. Isso significa que um teste será baseado em um planejamento anterior, conforme as

Figura 1 – Serviço de proximidade geográfica

```
public class GeographicProximityService : IGeographicProximityService
{
    private const double RadiusOfTheEarth = 6371; // approximately
    private readonly IPersonRepository _personRepository;

    0 referências | mauriciogomes02, há 11 dias | 1 autor, 1 alteração
    public GeographicProximityService(IPersonRepository personRepository)
    {
        _personRepository = personRepository;
    }

    2 referências | mauriciogomes02, há 4 dias | 1 autor, 1 alteração
    public Task<Paginated<Person>> GetAsync(
        CompositeAddress originAddress,
        double radius,
        int pageIndex,
        int pageSize,
        CancellationToken cancellationToken)
    {
        var originLatitude = originAddress.GeographicCoordinate!.Latitude;
        var originLongitude = originAddress.GeographicCoordinate!.Longitude;

        // Haversine Formula
        Expression<Func<Person, bool>> predicate = x => RadiusOfTheEarth * (2 * Math.Asin(
            Math.Sqrt(
                Math.Pow(
                    Math.Sin(Math.PI / 180 * (x.Address.GeographicCoordinate!.Latitude - originLatitude) / 2),
                    2) +
                    Math.Pow(
                        Math.Sin(Math.PI / 180 * (x.Address.GeographicCoordinate!.Longitude - originLongitude) / 2),
                        2) *
                        Math.Cos(Math.PI / 180 * originLatitude) *
                        Math.Cos(Math.PI / 180 * x.Address.GeographicCoordinate!.Latitude)
                )))
            <= radius && x.Id != originAddress.PersonId;

        var criteria = new Criteria<Person, Person>(
            predicate,
            x => x,
            pageIndex,
            pageSize);

        return _personRepository.GetPaginatedAsync(criteria, cancellationToken);
    }
}
```

Fonte: Produzido pelo autor

Figura 2 – Serviço de proximidade funcional

```
public class FunctionalProximityService : IFunctionalProximityService
{
    private readonly IPersonRepository _personRepository;

    0 referências | mauriciogomes02, há 8 dias | 1 autor, 1 alteração
    public FunctionalProximityService(IPersonRepository personRepository)
    {
        _personRepository = personRepository;
    }

    2 referências | mauriciogomes02, há 3 dias | 1 autor, 3 alterações
    public Task<Paginated<Person>> GetAsync(
        Activity activity,
        int pageIndex,
        int pageSize,
        CancellationToken cancellationToken)
    {
        var days = activity.ActivityDay.Select(x => x.Day.Id);

        var criteria = new Criteria<Person, Person>(
            storedPerson => storedPerson.Activities != null && storedPerson.Id != activity.PersonId
            && storedPerson.Activities.Any(
                // Day
                storedActivity => storedActivity.ActivityDay.Any(
                    activityDay => days.Contains(activityDay.Day.Id))
                // Address
                && (storedActivity.Address.PostalCode == activity.Address.PostalCode
                    && storedActivity.Address.Number == activity.Address.Number)
                // Schedules
                && ((storedActivity.Start >= activity.Start && storedActivity.Start < activity.End)
                    || (storedActivity.Start <= activity.Start && storedActivity.End > activity.End)
                )),
            x => x,
            pageIndex,
            pageSize);

        return _personRepository.GetPaginatedAsync(criteria, cancellationToken);
    }
}
```

Fonte: Produzido pelo autor

necessidades do software, e poderá ser executado por um sistema, gerando um conjunto de resultados.

Segundo (PRESSMAN, 2011), o teste de software é um elemento da Verificação e Validação (V&V) que juntas abrangem muitas atividades, garantindo a qualidade do software. A Verificação é responsável por garantir que um software implemente corretamente uma função específica atribuída a ele. A Validação é responsável por

garantir que o software seja criado conforme os requisitos do cliente. Logo, o teste de software possui uma forte relação com a qualidade de software.

(PRESSMAN, 2011) estabelece que no contexto de engenharia de software, o teste é formado por uma série de quatro etapas implementadas sequencialmente. Primeiramente o foco se dá em cada componente individual, garantindo que o mesmo desempenhe sua função adequadamente como uma unidade. Essa primeira etapa é formada pelos testes de unidade ou testes unitários, que serão utilizados para realizar a verificação e validação do produto proposto, visto que são a base do teste de software.

(KHORIKOV, 2020) afirma que o teste de unidade visa permitir o crescimento sustentável do software. Isso significa que com o passar do tempo, um software tende a se tornar complexo, dificultando seu desenvolvimento e aumentando a quantidade de tempo para a criação de uma funcionalidade ou a manutenção de alguma já existente. Esse fato demonstra a dificuldade de manter sustentável o crescimento de uma aplicação. Portanto, os testes de unidade garantem que a funcionalidade existente funcione, mesmo após introduzir novos recursos ou refatorar o código para melhor atender aos novos requisitos (KHORIKOV, 2020).

A terceira etapa do teste de software é representada pelo teste de validação (PRESSMAN, 2011). Essa etapa garante que o software satisfaça todos os requisitos esperados. Dessa forma, a validação do software ocorre por meio de uma série de testes, demonstrando que a aplicação está segundo os requisitos definidos (PRESSMAN, 2011).

Em suma, o produto gerado pelo presente projeto será validado através dos testes de unidade, relativos aos trechos de código da aplicação, como também dos testes de validação. Isso se dá devido o fato de pertencerem à primeira e terceira das quatro etapas de um teste de software, além de serem responsáveis por verificar e validar trechos de código, como também de funções globais do produto.

Os testes de validação serão realizados manualmente, garantindo assim a validação das principais funcionalidades do software. Primeiramente será realizado o cadastro de três pessoas fictícias. Essas pessoas terão seus endereços relativos ao endereço das instituições de ensino superior UniLaSalle, Ulbra e UniRitter, possibilitando a validação da funcionalidade de proximidade geográfica.

O serviço de geocodificação utilizado é gratuito e impreciso na definição das coordenadas geográficas. Isso implica que a latitude e longitude dos endereços cadastrados podem estar divergentes dos valores reais. Entretanto, o foco está em validar se o cálculo da distância entre dois pontos está correto e não se as coordenadas geográficas são equivalentes ao endereço cadastrado. Nesse sentido, a validação ocorrerá considerando a latitude e a longitude que o produto armazenar.

Portanto, a funcionalidade de proximidade geográfica será considerada válida se retornar apenas as pessoas que estejam em um raio de distância máxima em quilômetros, considerando as coordenadas geográficas armazenadas. O cálculo para medir essa distância deve ter como base a Fórmula de Haversine exposta na Seção 2.3.

Para a validação da funcionalidade de proximidade funcional, será necessário realizar o cadastro de três atividades para o aluno da instituição LaSalle, duas atividades para o aluno da instituição Ulbra e duas atividades para o aluno da instituição UniRitter. Os alunos terão uma única atividade comum entre eles. As duas atividades restantes

do aluno da instituição LaSalle deverão ser comum a uma atividade do aluno da instituição Ulbra e a uma atividade do aluno da instituição UniRitter, respectivamente.

Logo, a funcionalidade de proximidade funcional será considerada válida se retornar apenas as pessoas que possuam alguma atividade em comum com uma atividade específica de alguma pessoa. Nesse sentido, a validação ocorrerá realizando o consumo da funcionalidade para cada atividade do aluno da instituição LaSalle.

3 CONCLUSÃO

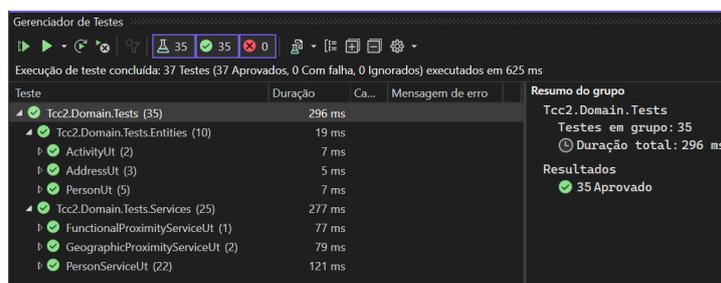
Relacionamentos bons e saudáveis são essenciais para vida em sociedade, a qual é definida de forma intrínseca a esses relacionamentos. Isso se dá porque todas as pessoas precisam suprir a sua necessidade de pertencimento, vivendo em sociedade e desenvolvendo suas relações. Logo, as pessoas foram feitas para se relacionar, gerando benefícios a si mesmas como também umas às outras.

A proximidade é um ótimo preditor desses bons relacionamentos. Estar próximo física ou funcionalmente é de certa forma conveniente e possibilita a interação mútua, permitindo que as relações possam se desenvolver. Nesse sentido, a Computação pode auxiliar esse processo, agindo como uma ferramenta e sugerindo pessoas que sejam candidatas para a criação de vínculos, conforme os parâmetros de proximidade definidos.

Portanto, o produto gerado desempenha corretamente seu papel, auxiliando no desenvolvimento de relações interpessoais ao sugerir candidatos à interação, conforme a proximidade existente entre os envolvidos. Isso se dá visto que as tarefas desempenhadas atualmente consomem grande parte do tempo das pessoas que em sua maioria esquecem a importância de construir e manter bons relacionamentos, deixando até mesmo de procurá-los. Portanto, o produto gerado tem em vista auxiliar e motivar o desenvolvimento desses relacionamentos, facilitando o processo inicial da interação.

Em relação ao que tange o contexto de validação do produto, os testes de unidade desempenharam bem seu papel em abranger os componentes individualmente e garantir a sua verificação e validação conforme o esperado. Por conseguinte, o produto foi validado segundo o seu domínio, mas em um contexto de componentes individuais. Esses componentes exerceram corretamente sua função, como pode ser observado na Figura 3.

Figura 3 – Resultado dos testes de unidade



Fonte: Produzido pelo autor

Os testes de validação comprovaram a eficácia do sistema. Esses testes con-

sideraram a descrição da validação do produto, abordada na Seção 2.4. Para isso, inicialmente foram cadastradas três pessoas fictícias, com endereços e distâncias diferentes. As coordenadas geográficas desses endereços são expostas na Figura 4.

Figura 4 – Coordenadas geográficas do endereço das pessoas fictícias cadastradas

```

SELECT p.Name "Nome Pessoa",
       a.GeographicCoordinate_Latitude Latitude,
       a.GeographicCoordinate_Longitude Longitude
FROM People p
JOIN Addresses a
  ON p.Id = a.PersonId;

```

Nome Pessoa	Latitude	Longitude
Aluno LaSalle	-29.917	-51.181
Aluno Ulbra	-29.899	-51.168
Aluno UniRitter	-29.946	-51.175

Fonte: Produzido pelo autor

Uma calculadora ² baseada na Fórmula de Haversine foi utilizada para garantir o cálculo correto da distância geográfica entre dois pontos, tendo como base as suas coordenadas geográficas. Essas distâncias estão descritas na Tabela 1.

Tabela 1 – Distância geográfica entre as pessoas fictícias cadastradas

Pessoa 1	Pessoa 2	Distância
Aluno LaSalle	Aluno Ulbra	2,36 km
Aluno LaSalle	Aluno UniRitter	3,28 km
Aluno Ulbra	Aluno UniRitter	5,27 km

Fonte: Produzido pelo autor

O retorno obtido através do consumo da funcionalidade de proximidade geográfica está conforme o esperado, considerando os dados obtidos da Tabela 1. Ao pesquisar pelas pessoas próximas geograficamente, tendo como referência o aluno da instituição LaSalle, os resultados obtidos com o raio de distância máxima sendo de 3 km e 5 km podem ser visualizados na Figura 5 e na Figura 6, respectivamente. Como pode ser observado, o retorno é diferente para as duas solicitações, evidenciando que conforme a distância desejada, apenas as pessoas que estejam em uma distância menor ou igual a ela serão sugeridas.

Após validar a funcionalidade de proximidade geográfica, foram realizados os cadastros das atividades que cada pessoa desempenha. Essas atividades são expostas na Figura 7.

O retorno obtido através do consumo da funcionalidade de proximidade funcional também está segundo o esperado, com base nos dados obtidos da Figura 7. Ao pesquisar pelas pessoas próximas funcionalmente, tendo como referência o aluno da instituição LaSalle, os resultados obtidos para a atividade **Trabalhar**, **Nadar** e **Treinar academia** podem ser visualizados na Figura 8, Figura 9 e Figura 10, respectivamente.

² <https://www.vcalc.com/wiki/vcalc/haversine-distance>

Figura 5 – Proximidade geográfica com o raio de distância máxima de 3 km



Fonte: Produzido pelo autor

Figura 6 – Proximidade geográfica com o raio de distância máxima de 5 km



Fonte: Produzido pelo autor

Figura 7 – Atividades das pessoas fictícias cadastradas

```
SELECT p.Name "Nome Pessoa",
a.Address_Street Rua,
a.Address_Number "Número",
a.`Start` "Horário de Início",
a.`End` "Horário de Término",
a.Description "Descrição",
(
    SELECT GROUP_CONCAT(ad.DayId)
    FROM ActivityDay ad
    WHERE ad.ActivityId = a.Id
    GROUP BY ad.ActivityId
) "Dias da semana"
FROM People p
JOIN Activities a
ON p.Id = a.PersonId
```

Nome Pessoa	Rua	Número	Horário de Início	Horário de Término	Descrição	Dias da semana
Aluno LaSalle	Avenida Victor Barreto	2288	08:00:00	17:00:00	Trabalhar	1,2,3,4,5
Aluno LaSalle	Rua Quinze de Janeiro	423	06:00:00	07:00:00	Nadar	1,3,5
Aluno LaSalle	Avenida Dr. Sezefredo Azambuja Vieira	2316	06:00:00	07:00:00	Treinar academia	2,4
Aluno Ulbra	Avenida Victor Barreto	2288	08:00:00	17:00:00	Trabalhar	1,2,3,4,5
Aluno Ulbra	Rua Quinze de Janeiro	423	06:00:00	07:00:00	Nadar	1,3,5
Aluno UniRitter	Avenida Victor Barreto	2288	08:00:00	17:00:00	Trabalhar	1,2,3,4,5
Aluno UniRitter	Avenida Dr. Sezefredo Azambuja Vieira	2316	06:00:00	07:00:00	Treinar academia	2,4

Fonte: Produzido pelo autor

Como pode ser observado, o retorno é diferente para todas as solicitações, evidenciando que conforme a atividade tida como referência, apenas as pessoas que possuam uma atividade em comum a ela serão sugeridas.

Em suma, o produto criado desempenhou sua função conforme o esperado. Logo, a aplicação serviu como uma ferramenta de auxílio ao desenvolvimento de relacionamento interpessoal ao sugerir pessoas candidatas à interação, conforme a proximidade geográfica ou funcional.

Ainda que o produto gerado tenha sido validado e apresentado como sendo fun-

Figura 8 – Proximidade funcional com a atividade **Trabalhar**



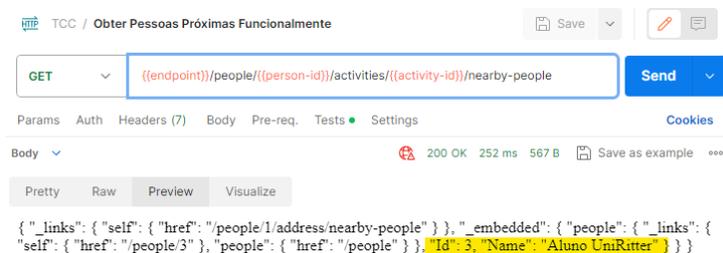
Fonte: Produzido pelo autor

Figura 9 – Proximidade funcional com a atividade **Nadar**



Fonte: Produzido pelo autor

Figura 10 – Proximidade funcional com a atividade **Treinar academia**



Fonte: Produzido pelo autor

cional, existem ainda muitas etapas a serem desenvolvidas até que o software assuma a sua forma completa. Portanto, os próximos passos propostos para desenvolver algumas dessas etapas podem ser enumerados da seguinte forma:

1. Implementar a funcionalidade relacionada às semelhanças e diferenças que as pessoas possuem.
2. Criar uma interface gráfica para se comunicar com a API e utilizá-la como um recurso para a comunicação direta com o usuário final.
3. Implementar o cadastro de usuários, definindo a autenticação e autorização que cada grupo possuirá.
4. Configurar a infraestrutura e o pipeline de CI/CD.
5. Hospedar a aplicação em algum domínio.

REFERÊNCIAS

ARKIN, R. M.; BURGER, J. M. Effects of unit relation tendencies on interpersonal attraction. **Social Psychology Quarterly**, [Sage Publications, Inc., American Sociological Association], v. 43, n. 4, p. 380–391, 1980. ISSN 01902725. Disponível em: <<http://www.jstor.org/stable/3033958>>.

BACK, M. D.; SCHMUKLE, S. C.; EGLOFF, B. Becoming friends by chance. **Psychological Science**, v. 19, n. 5, p. 439–440, 2008. PMID: 18466403. Disponível em: <<https://doi.org/10.1111/j.1467-9280.2008.02106.x>>.

BAUMEISTER, R. F.; LEARY, M. R. The need to belong: Desire for interpersonal attachments as a fundamental human motivation. **Psychological Bulletin**, v. 117, n. 3, p. 497–529, 1995.

BOYD, A. L.; SAMID, D. Marriage and health: his and hers. **Psychological bulletin**, v. 127, n. 4, p. 472–503, 2001.

BYRNE, D.; CLORE, G.; SMEATON, G. The attraction hypothesis. do similar attitudes affect anything? **Journal of Personality and Social Psychology**, v. 51, p. 1167–1170, 12 1986.

CAMPISI, J. et al. Facebook, stress, and incidence of upper respiratory infection in undergraduate college students. **Cyberpsychology, Behavior, and Social Networking**, 2012.

DALLARI, D. de A. **Elementos de teoria geral do Estado**. São Paulo: Saraiva, 2009.

EVANS, E. **Domain-Driven Design: Tackling Complexity in the Heart of Software**. [S.l.]: Addison-Wesley, 2004.

FIELDING, R. T.; TAYLOR, R. N. **Architectural Styles and the Design of Network-Based Software Architectures**. Tese (Doutorado), 2000. AAI9980887.

HARMON-JONES, E.; ALLEN, J. J. B. The role of affect in the mere exposure effect: Evidence from psychophysiological and individual differences approaches. **Personality and Social Psychology Bulletin**, v. 27, n. 7, p. 889–898, 2001. Disponível em: <<https://doi.org/10.1177/0146167201277011>>.

KELLY, M. **JSON Hypertext Application Language**. [S.l.], 2023. Work in Progress. Disponível em: <<https://datatracker.ietf.org/doc/draft-kelly-json-hal/11/>>.

KHORIKOV, V. **Unit Testing Principles, Practices, and Patterns**. [S.l.]: Simon and Schuster, 2020.

KNIGHT, J.; VALLACHER, R. Interpersonal engagement in social perception: The consequences of getting into the action. **Journal of Personality and Social Psychology**, v. 40, p. 990–999, 06 1981.

MARIA, E. et al. Measure distance locating nearest public facilities using haversine and euclidean methods. **Journal of Physics: Conference Series**, IOP Publishing, v. 1450, n. 1, p. 012080, feb 2020. Disponível em: <<https://dx.doi.org/10.1088/1742-6596/1450/1/012080>>.

MARTIN, R. C. **Clean Architecture: A Craftsman's Guide to Software Structure and Design**. Boston, MA: Prentice Hall, 2017. (Robert C. Martin Series). ISBN 978-0-13-449416-6. Disponível em: <<https://www.safaribooksonline.com/library/view/clean-architecture-a/9780134494272/>>.

MENG, M.; STEINHARDT, S.; SCHUBERT, A. Application programming interface documentation: What do software developers want? **Journal of Technical Writing and Communication**, v. 48, p. 295–330, 07 2018.

MONTOYA, R.; HORTON, R. On the importance of cognitive evaluation as a determinant of interpersonal attraction. **Journal of personality and social psychology**, v. 86, p. 696–712, 05 2004.

MYERS, D. G. **Psicologia Social**. Porto Alegre: Grupo A, 2014. ISBN 9788580553390.

PRESSMAN, R. S. **Engenharia De Software: Uma Abordagem Profissional**. 7. ed. [S.l.]: Grupo A - AMGH, 2011.

RODRIGUES, F. R.; PORTO, T. C. B. Modernidade líquida: compreendendo fenomenologicamente a era das relações superficiais. **Brazilian Journal of Development**, v. 7, n. 5, p. 45223–45241, 2021.

SINGH, R.; HO, S. Y. Attitudes and attraction: A new test of the attraction, repulsion and similarity-dissimilarity asymmetry hypotheses. **British Journal of Social Psychology**, v. 39, n. 2, p. 197–211, 2000. Disponível em: <<https://bpspsychub.onlinelibrary.wiley.com/doi/abs/10.1348/014466600164426>>.

UMBERSON, D.; MONTEZ, J. K. Social relationships and health: A flashpoint for health policy. **Journal of Health and Social Behavior**, 2010.