



Universidade do Minho
Escola de Engenharia

ElderWatch

Deteção e Notificação de Quedas

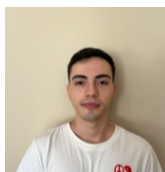
Sensorização e Ambiente
Mestrado em Engenharia Informática

Grupo 12

Gabriela Santos Ferreira da Cunha - pg53829

Millena de Freitas Santos - pg54107

Nuno Guilherme Cruz Varela - pg54117



maio, 2024

Conteúdo

1	Introdução	3
2	Contextualização e Motivação	3
2.1	Objetivos	3
2.2	Abordagem	3
2.3	Modelo de Negócio	4
3	Sistema	5
3.1	Arquitetura	5
3.2	Funcionalidades	7
4	Recolha de Dados	7
4.1	Sensores Utilizados	7
4.2	Método de Recolha	8
4.3	Armazenamento dos Dados	8
5	Deteção de Queda	10
6	Notificações	10
7	Visualização de Dados	11
7.1	Gráficos	12
8	Resultados Obtidos	15
9	Limitações e Trabalho Futuro	17
10	Conclusão	17

1 Introdução

No âmbito da unidade curricular de Sensorização e Ambiente foi-nos proposto o desenvolvimento de um sistema capaz de obter dados e gerar informação útil no contexto do ambiente onde se encontra inserido. O projeto desenvolvido consiste numa aplicação orientada à monitorização de idosos que, através da sensorização móvel, deteta quedas e notifica automaticamente um ou vários contactos definidos pelo utilizador, permitindo também notificá-los manualmente perante um dado contexto de emergência.

2 Contextualização e Motivação

À medida que a população mundial envelhece, surge uma preocupação crescente com o bem-estar e a segurança dos idosos, especialmente aqueles que vivem sozinhos ou que enfrentam desafios de saúde. As quedas são uma das principais causas de lesões e até mesmo de fatalidades entre os idosos e muitas vezes acontecem quando não há ninguém por perto para ajudar. Perante este cenário, a tecnologia pode desempenhar um papel crucial ao oferecer soluções inovadoras para monitorizar e responder rapidamente a emergências. Através da sensorização, é possível criar uma solução que não apenas deteta quedas mas também aciona automaticamente uma resposta adequada, neste caso, notificando contactos pré-definidos. Desta forma, não só é reduzido o tempo de resposta em casos de emergência, como também é promovida uma sensação de segurança tanto para os idosos quanto para os seus familiares ou pessoas responsáveis por cuidar destes.

2.1 Objetivos

- **Segurança do idoso:** Garantir a segurança do idoso, detetando quedas e outras emergências rapidamente e notificando contactos relevantes para prestar assistência;
- **Autonomia e independência:** Promover a autonomia e independência dos idosos, permitindo-lhes viver de forma mais independente sabendo que, quando necessário, a assistência estará disponível;
- **Redução do tempo de resposta:** Minimizar o tempo entre a ocorrência de uma emergência e a chegada da ajuda, aumentando as chances de recuperação e reduzindo potenciais danos.

2.2 Abordagem

- **Deteção de quedas:** Desenvolvimento de um algoritmo que possa identificar quedas com precisão, minimizando falsos positivos e negativos;
- **Obtenção contínua de dados em tempo real:** Captura contínua de dados dos sensores para monitorizar a atividade do utilizador, essencial

para a detecção de queda e para permitir uma resposta rápida em situações de emergência e garantir a segurança do idoso;

- **Comunicação e notificação:** Estabelecimento de um sistema de comunicação robusto que permita notificar automaticamente contactos definidos pelo utilizador em caso de detecção de queda ou emergência manual;
- **Histórico de eventos:** Armazenamento do histórico de quedas e emergências do utilizador, por forma a auxiliar o planeamento de cuidados do idoso, identificar padrões de atividade e avaliar o seu estado de saúde;
- **Armazenamento dos dados recolhidos:** Armazenamento dos dados recolhidos pelos sensores com vista a melhorar o algoritmo de detecção de queda;
- **Interação com o utilizador:** Criação de uma interface intuitiva e acessível, que permita aos idosos interagir facilmente com a aplicação, incluindo a capacidade de definir contactos de emergência e acionar notificações manuais.

2.3 Modelo de Negócio

Reconhecemos a importância de não apenas conceber uma solução eficaz para a monitorização de idosos, mas também de garantir o sucesso e sustentabilidade da sua implementação a longo prazo em ambientes do mundo real. Neste sentido, apresentamos o modelo de negócio proposto pelo grupo que consideramos sólido e viável para a aplicação.

Identificação e Caracterização do Potencial Cliente

O potencial cliente abrange uma ampla gama de indivíduos, com foco principal em idosos que enfrentam desafios relacionados à saúde, bem como aqueles que vivem sozinhos e podem estar mais suscetíveis a situações de emergência. Para este grupo, a monitorização contínua e a resposta rápida em caso de emergência são cruciais para garantir a sua segurança e bem-estar contínuo. Não obstante, o potencial cliente também se estende a outras populações vulneráveis, como pessoas com deficiência física ou mental, estas que podem enfrentar desafios semelhantes em termos de segurança e necessidade de monitorização constante. Da mesma forma, indivíduos que se encontrem a recuperar de cirurgias ou que tenham a sua mobilidade temporariamente limitada constituem outro grupo que pode beneficiar significativamente desta solução.

Eventuais Parcerias

Embora a aplicação tenha sido projetada para que pudesse ser utilizada por famílias comuns como uma solução pessoal para monitorização de idosos, esta pode ser adaptada a uma variedade de contextos e necessidades. Uma das potenciais parcerias seria com hospitais, clínicas médicas ou agências de cuidados

domiciliares, permitindo que a aplicação fosse integrada como parte dos serviços de monitorização remota de pacientes. Outra colaboração interessante seria com lares de idosos, implementando a aplicação como parte integrante dos serviços oferecidos aos residentes. Ao adotar essa solução, os lares poderiam vir a melhorar a segurança e o bem-estar dos idosos, fornecendo uma camada adicional de monitorização e resposta a emergências.

Estratégia de Monetização

Como estratégia de monetização, pensamos na oferta de uma versão básica gratuita da aplicação, juntamente com opções de assinatura que desbloqueiam recursos adicionais e funcionalidades *premium*.

Assim, como versão gratuita, estaria disponível o serviço de deteção e notificação de quedas, bem como o acesso ao histórico de eventos. Uma vez que o objetivo da aplicação se trata de proteger e cuidar dos idosos, consideramos que estas funcionalidades são essenciais para que a monitorização seja efetivamente atingida, eliminando barreiras à adesão de utilizadores.

Como modelo *premium*, pensamos na disponibilização de algumas funcionalidades que permitiriam uma monitorização mais completa e personalizada. Por exemplo, é possível garantir a monitorização de um idoso através de 1 contacto de emergência, algo que seria oferecido na versão gratuita. No entanto, a possibilidade extra de adicionar múltiplos contactos de emergência, proporcionaria uma rede de suporte mais ampla, melhorando a monitorização do idoso em questão. Em muitos casos, com apenas um contacto de emergência, existe o risco de essa pessoa não estar disponível para socorrer o idoso em tempo hábil. Outra das funcionalidades restringidas à versão gratuita seria a disponibilização da localização em tempo real do idoso e a inclusão da mesma em cada queda no histórico de eventos, tendo em conta que, apesar de fornecer uma visão mais abrangente da situação durante uma emergência ou estatísticas mais completas, não se trata de uma funcionalidade essencial, mas sim de uma funcionalidade que fornece uma experiência mais completa.

3 Sistema

3.1 Arquitetura

O sistema que nos propusemos a desenvolver faz uso essencialmente de sensores como o acelerómetro e GPS que recolhem dados acerca da aceleração e localização do utilizador para detetar eventuais situações de queda ou emergência. Os dados provenientes do acelerómetro são usados para detetar as quedas dos utilizadores através de um algoritmo de regras que irá ser abordado posteriormente. Relativamente aos dados do GPS, recorreremos ao Google Maps para mostrar a localização do utilizador, algo útil para os cuidadores dos idosos conseguirem monitorizar os seus comportamentos.

Através da arquitetura do sistema, representada na figura 1, podem ser observados 3 serviços chave que são utilizados pelo sistema:

- **Firestore Authentication:** é um serviço de autenticação seguro e de fácil integração no desenvolvimento da aplicação, que permite aos utilizadores fazerem os registos das respetivas contas e, consequentemente, autenticarem-se na aplicação por forma a fazer uso dela;
- **Firestore Firestore:** é um serviço de base de dados flexível e escalável que se destaca pela sua capacidade de persistir dados em tempo real. Desta forma, os dados recolhidos pelo acelerómetro de cada utilizador juntamente com um *booleano* que indica se foi uma queda são guardados na base de dados para poder ser feita uma visualização dos dados e, futuramente, treinar algoritmos de *machine learning* com estes dados. Para além disso, são persistidos os dados acerca dos utilizadores como o nome, email, número de telemóvel, localização e um histórico de quedas.
- **Firestore Cloud Messaging:** é um serviço que permite enviar *push notifications* e mensagens direcionadas a utilizadores. Este serviço constitui uma peça fundamental no nosso sistema, uma vez que é através dele que situações de queda/emergência são notificadas aos contactos do utilizador. Para a implementação das notificações foi necessário fazer um servidor, desenvolvido em Python Flask, comunicável através de uma API REST, responsável por receber pedidos de notificações e utilizar o serviço do Firestore Cloud Messaging para enviar as notificações aos destinatários.

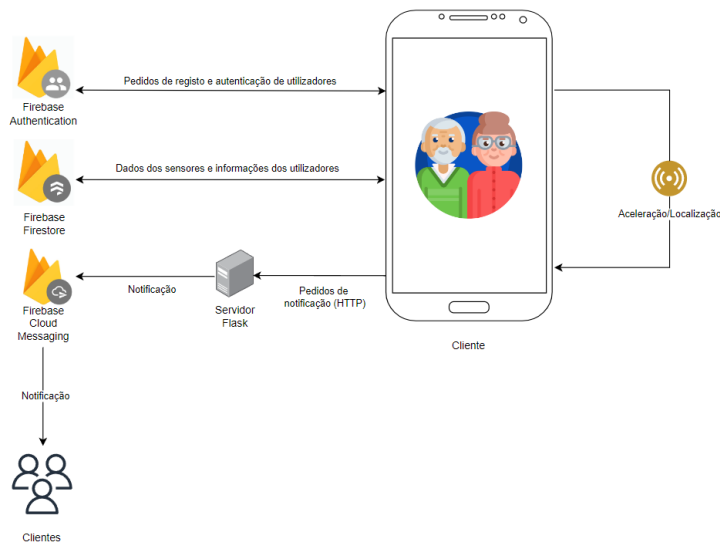


Figura 1: Arquitetura do Sistema.

3.2 Funcionalidades

- Registo e autenticação dos utilizadores;
- Botão de emergência que permite ao utilizador notificar os seus contactos sem que tenha ocorrido uma queda;
- Detecção de quedas em tempo real que notifica os contactos do utilizador em caso de queda;
- Lista do histórico de quedas/emergências do utilizador juntamente com a visualização do mapa e data onde estas ocorreram;
- Lista dos contactos do utilizador juntamente com a visualização do mapa da última localização do contacto selecionado;
- Adição e remoção de contactos;
- Possibilidade de edição dos dados da conta do utilizador (nome, *password* e telemóvel).

4 Recolha de Dados

4.1 Sensores Utilizados

Os dispositivos modernos, como é o caso dos *smartphones*, incluem uma diversidade de sensores disponíveis e são bastante práticos para recolher dados, uma vez que estão desde já presentes na vida quotidiana de grande parte da população. Por forma a detetar quedas e localizar o utilizador, foram utilizados os seguintes sensores:

- **Acelerómetro:** Sensor físico que mede a aceleração total experimentada pelo dispositivo. Na detecção de quedas, este pode identificar padrões de movimento repentinos e abruptos que indicam uma queda, como uma rápida mudança na orientação seguida por uma desaceleração abrupta. Neste caso em específico foi utilizado o sensor de aceleração linear, por forma a extrair a componente linear dos dados recolhidos pelo acelerómetro. O resultado é, portanto, os valores de aceleração que representam apenas as forças de movimento do dispositivo, eliminando a influência constante da gravidade.
- **GPS:** Sistema de navegação por satélite que fornece coordenadas de localização precisas em tempo real. Com este sensor, é possível obter a localização exata do utilizador no momento da queda ou em situações de emergência.

4.2 Método de Recolha

Neste momento, os dados são recolhidos através de um *smartphone*, por uma questão de simplicidade e disponibilidade de recursos. No entanto, uma alternativa seria, por exemplo, a utilização de dispositivos *wearable* projetados especificamente para monitorização de saúde e segurança, como pulseiras ou colares inteligentes.

Assim que é feita a autenticação do utilizador, é iniciado um *foreground service* para a recolha dos dados do acelerómetro. Este tipo de serviço é crucial para aplicações que precisam de manter operações em execução mesmo quando o utilizador não se encontra a interagir diretamente com a aplicação, o que é indicado para a deteção de quedas pois estas podem acontecer a qualquer momento. O serviço regista o ouvinte do sensor com um *delay* normal, adequado para a colheita de dados em tempo real sem consumir recursos excessivos. Assim, o serviço recebe atualizações contínuas das leituras do acelerómetro e sempre que é detetada uma mudança, o método `onSensorChanged()` é chamado para efetuar o tratamento dos dados. Para informar o utilizador sobre a execução do serviço, foi configurada uma notificação persistente que é mostrada enquanto o serviço está ativo. De notar que o serviço só é interrompido até ser explicitamente parado ou até a aplicação ser encerrada.



Figura 2: Notificação persistente.

No que diz respeito à colheita da localização do utilizador, é solicitada a permissão de acesso à localização assim que é realizada a autenticação do utilizador, caso este ainda não tenha concedido essa permissão anteriormente. Após a permissão ser concedida, é recolhida a localização através do serviço de localização disponibilizado pela Google Play Services. O grupo definiu o intervalo de atualização da localização de cada utilizador em 2 minutos.

4.3 Armazenamento dos Dados

Por forma a armazenar os dados recolhidos pelos sensores e os dados essenciais à aplicação, os dados são enviados e armazenados numa base de dados no Firestore. A nossa base de dados é composta pelas coleções **data** e **users**.

A primeira coleção armazena os dados relativos às medições efetuadas pelo acelerómetro, que são enviados de 5 em 5 segundos, isto é, a cada 5 segundos é criado um documento com um conjunto de valores da medição, o **id** do utilizador e se foi ou não detetada uma queda nesse momento. Este armazenamento

permite uma análise detalhada dos padrões de aceleração ao longo do tempo, possibilitando a identificação de comportamentos e eventos relevantes, informações essenciais para o refinamento contínuo do algoritmo de detecção de queda, permitindo ajustes e melhorias com base em dados reais e atualizados.

```

data
  8
    magnitude: 0.008020846515833477
    timestamp: 6 de maio de 2024 às 17:41:37 UTC+1
    x: 0.000026066197936228108
    y: 0.000008549573976779357
    z: 0.00002971820691755056
    1 {magnitude: 0.008071246593...}
    2 {magnitude: 0.008011181213...}
    3 {magnitude: 0.008147980114...}
    4 {magnitude: 0.008133730588...}
    5 {magnitude: 0.008126252228...}
    6 {magnitude: 0.008026619633...}
    7 {magnitude: 0.007988329607...}
    fall: false
    uid: "fFsFLhSijdd2eBuwdFizQoNL0Un2"

```

Figura 3: Exemplo de documento da coleção **data**.

A segunda coleção armazena os dados relativos a cada utilizador, estes que são inseridos após efetuar o registo na aplicação. Para além dos dados vindos diretamente da autenticação, são também aqui guardados a localização do utilizador, os **ids** dos contactos de emergência, bem como a lista de atividades (quedas e emergências), contendo a localização e *timestamp* das mesmas.

```

activities
  8
    fall: false
    location
      latitude: 41.6463561
      longitude: -8.4307535
      timestamp: 7 de maio de 2024 às 20:26:57 UTC+1
    1 {fall: true, location: {lo...}}
  contacts
    8 "4QVfwTd5bPM3eewzNdZiGfoiezm2"
    email: "teste@hotmail.com"
    location
      latitude: 41.6463561
      longitude: -8.4307535
      timestamp: 7 de maio de 2024 às 20:27:07 UTC+1
    name: "Teste"
    phone: "912345678"
    token: "eZbP1WlgQ7i3pog7o2HjaG:APA91bHfEuB0Xxk9TV3qCYctXLxAlrFvEg9tyW-nY6Ql617M8LRgG3mgOhIXQM2IMorAOCqKogW8"

```

Figura 4: Exemplo de documento da coleção **users**.

5 Detecção de Queda

No âmbito do desenvolvimento do algoritmo de detecção de queda, seguimos uma abordagem [1], presente na literatura científica, que utiliza apenas dados provenientes do acelerómetro para detetar quedas. Esta abordagem consiste numa sequência de estágios: aquisição dos dados, tratamento dos dados, detecção da queda e detecção de um período de longa permanência no chão.

Após a recolha dos dados do acelerómetro, inicia-se a fase de tratamento dos mesmos. Neste sentido, os valores da aceleração recolhidos passam por um filtro passa-baixo para remover os ruídos presentes nos dados. Posteriormente, como a aceleração está separada nas suas componentes x , y e z , é calculada a magnitude ou módulo da aceleração, transformando estes 3 valores num único valor que reflete a intensidade total da aceleração em cada instante. O módulo da aceleração pode ser calculado da seguinte forma:

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Na fase da detecção da queda, é utilizado um algoritmo que opera com dois parâmetros principais: o *threshold* baixo de aceleração e o *threshold* alto de aceleração. Estes *thresholds* são estabelecidos com base na observação de que os valores de aceleração durante uma queda são diferentes dos valores durante atividades normais. Durante o período de detecção, correspondente a 10 segundos, o algoritmo recolhe os valores da aceleração. Uma queda é identificada se o valor do módulo da aceleração em algum momento cai abaixo do *threshold* baixo e, subsequente a isso, excede o *threshold* alto dentro do mesmo período de detecção.

Por forma a melhorar a precisão do algoritmo, foi introduzida uma camada adicional de verificação após a detecção da queda chamada *long lie detection*, que verifica se após uma detecção da queda o indivíduo permanece imóvel por um certo período de tempo — um forte indicador de que uma queda realmente ocorreu. Para este refinamento do algoritmo, foi definido um período de tempo de 3 segundos, no qual é verificado se a pessoa está imóvel, ou seja, a sua aceleração é praticamente constante entre valores baixos.

6 Notificações

As notificações desempenham um papel fundamental nesta aplicação de detecção de quedas, especialmente no contexto da monitorização dos idosos. As notificações constituem, portanto, um componente fulcral do nosso sistema, servindo a múltiplas funções essenciais para a eficácia, segurança e resposta rápida em situações de emergência.

Neste sentido, a nossa aplicação emite notificações de emergência para os atuais contactos do utilizador quando ocorre uma das duas seguintes situações:

o idoso aciona o botão de emergência presente na página principal da aplicação ou ocorre uma situação de queda e, automaticamente, notifica os seus contactos.

Por forma a implementar a notificação dos contactos de um utilizador, a aplicação faz uso do serviço de nuvem Firebase Cloud Messaging (FCM) desenvolvido pela Google. O Firebase Cloud Messaging é uma solução que facilita a comunicação entre servidores e dispositivos cliente, incluindo dispositivos Android. Esta plataforma é amplamente utilizada para enviar notificações de *push* e mensagens de dados a aplicações em dispositivos móveis com um grande facilidade de integração.

De forma a utilizar o Firebase Cloud Messaging, é utilizado um servidor hospedado na Google Cloud que trata de receber pedidos de notificação através de uma API REST. O servidor utiliza uma API backend em Flask para atender a estes pedidos e, posteriormente, utilizar os serviços de *messaging* do Firebase para enviar as respetivas notificações. Para além disso, torna-se necessário entender o conceito de *token* no contexto do Firebase Cloud Messaging. O *token* refere-se a um identificador único e é crucial para o processo de envio de mensagens de *push*, uma vez que serve como endereço específico do dispositivo do utilizador no sistema de notificações do FCM. Deste modo, o *token* de cada utilizador é persistido no Firestore. Da perspetiva do cliente, quando é necessário emitir notificações aos seus contactos, faz-se uma consulta na base de dados dos respetivos *tokens* dos contactos e enviam-se pedidos HTTP de notificação ao servidor desenvolvido.

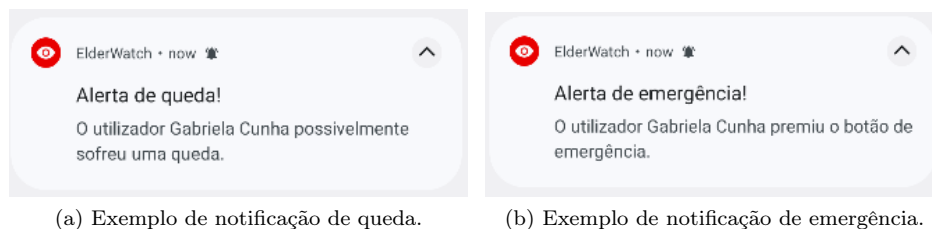


Figura 5: Notificações de queda e emergência.

7 Visualização de Dados

A ferramenta de visualização de dados escolhida para este projeto foi o Looker Studio. Para viabilizar a análise, foi desenvolvido um *script* em Python para extrair os dados necessários da base de dados. Os campos “fall” e “timestamp” foram selecionados a partir do *array* “activities” de cada utilizador, permitindo a análise das quedas registadas por hora.

Após a recolha dos dados, foi realizado um tratamento simples para ajustar os *timestamps* para horas específicas. Por exemplo, um registo de horário entre as

15:00h e 15:59h foi ajustado para 15:00h, facilitando a segmentação das quedas por intervalos de hora na visualização de dados.

Por fim, a nova *spreadsheet* resultante foi armazenada no Google Sheet para acesso pelo Looker Studio. Portanto, seria possível implementar um processo que execute o *script* periodicamente, por exemplo ao utilizar Google Cloud Functions, atualizando os dados na *sheet* que serve como fonte para a ferramenta de visualização de dados. Essa abordagem garante a disponibilidade de informações atualizadas para análise contínua.

É possível observar na figura 6 os campos extraídos da base de dados e os seus respectivos dados após o tratamento mencionado anteriormente. É importante salientar o significado de “TRUE” ou “FALSE” no atributo “fall”. O primeiro refere-se às quedas detetadas enquanto que o último refere-se às emergências acionadas pelo utilizador sem que exista a deteção de queda por parte do algoritmo.

fall	hour_interval
TRUE	2024-05-08 13:00:00
FALSE	2024-05-08 13:00:00
FALSE	2024-05-06 15:00:00
FALSE	2024-05-06 15:00:00
FALSE	2024-05-06 15:00:00
TRUE	2024-05-07 19:00:00
TRUE	2024-05-07 19:00:00
TRUE	2024-05-07 19:00:00
TRUE	2024-05-07 19:00:00
FALSE	2024-05-07 19:00:00

Figura 6: Exemplo da nova planilha criada.

7.1 Gráficos

Os gráficos a seguir foram elaborados com o intuito de analisar a distribuição das quedas detetadas e situações nas quais os utilizadores ativaram o sistema de emergência. Esta análise é de suma importância, uma vez que permite identificar possíveis padrões nos períodos do dia que podem representar maior perigo ou apresentar maior risco aos idosos, isto é, momentos em que há uma maior concentração de ocorrências em que o campo “fall” é verdadeiro para quedas e falso para outras emergências.

A figura 7 representa um gráfico de barras com a distribuição de quedas detetadas por intervalo de hora.

Quedas por hora

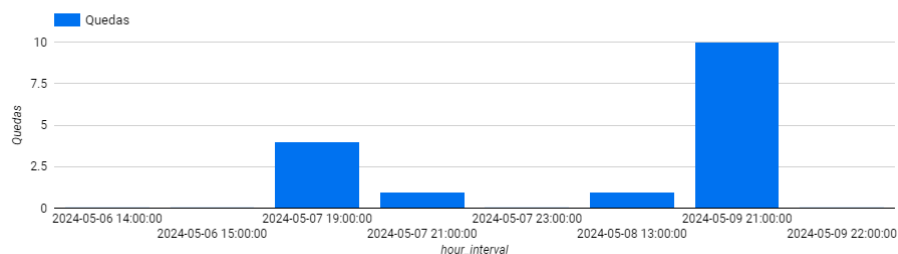


Figura 7: Distribuição de quedas detetadas por intervalo de hora.

A figura 8 representa um gráfico de barras com a distribuição de emergências acionadas pelos utilizadores por intervalo de hora.

Emergências por hora

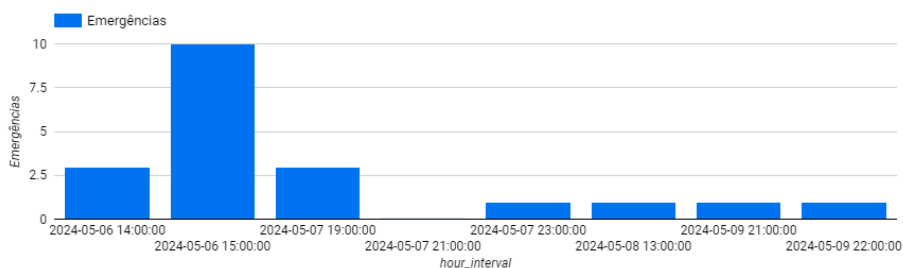


Figura 8: Distribuição de emergências acionadas por intervalo de hora.

A figura 9 representa um gráfico de barras com a distribuição total de quedas e emergências por intervalo de hora. Este gráfico facilita a análise entre a distribuição dos dois tipos de emergência.

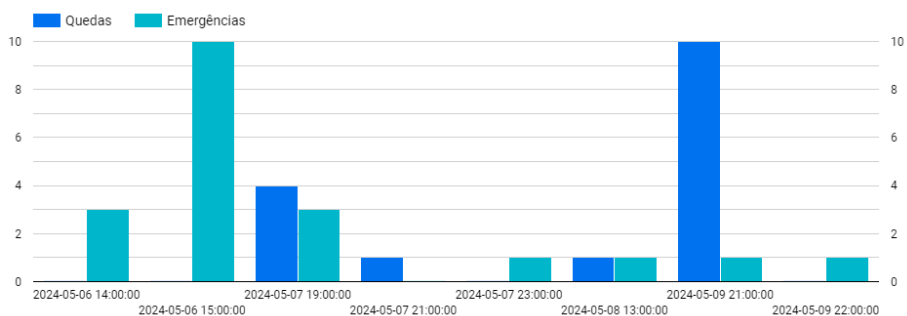


Figura 9: Distribuição de quedas e emergências acionadas por intervalo de hora.

A figura 10 representa um gráfico circular com a distribuição em percentagem de todas as quedas e emergências registadas.

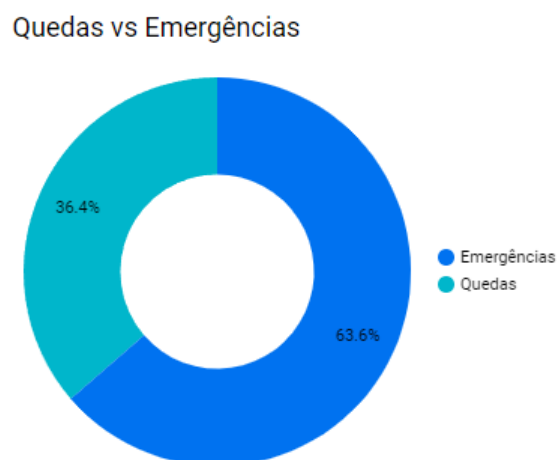


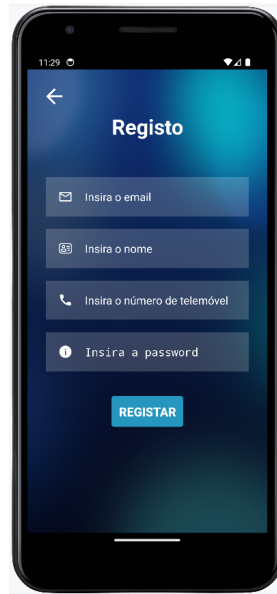
Figura 10: Distribuição em percentagem das quedas e emergências registadas.

Assim, a visualização de dados emerge como uma ferramenta de extrema relevância não apenas para compreender o desempenho da aplicação em si, visando a implementação de aprimoramentos ao longo do tempo, mas também para monitorizar o comportamento dos utilizadores. Esta monitorização pode, no futuro, ser explorada para fornecer aos cuidadores acesso a estatísticas relevantes, contribuindo significativamente para o cuidado e bem-estar do idoso.

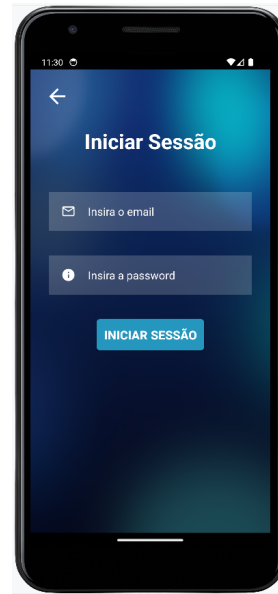
8 Resultados Obtidos



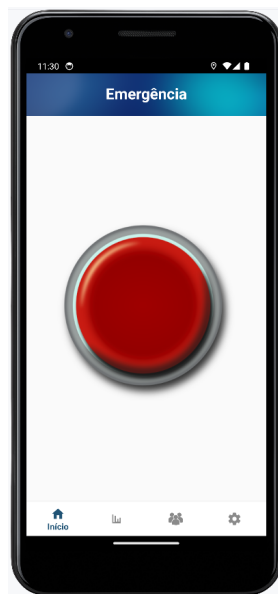
(a) Página inicial



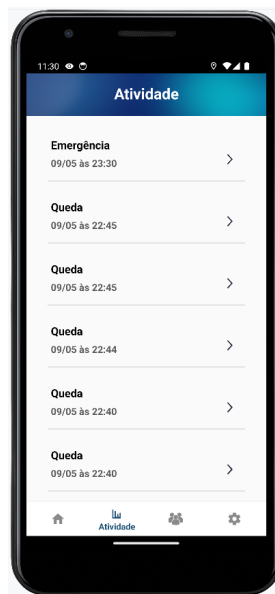
(b) Registo



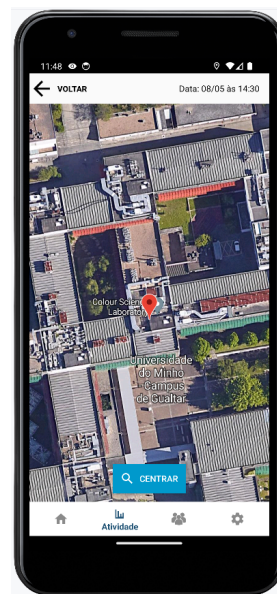
(c) Autenticação



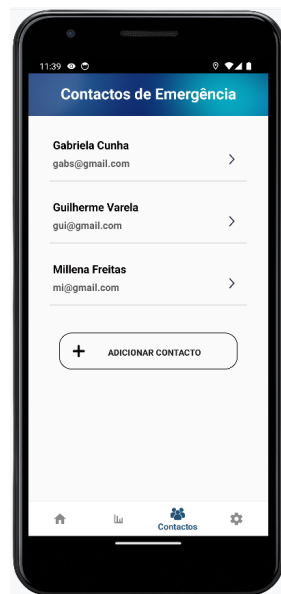
(d) Botão de emergência



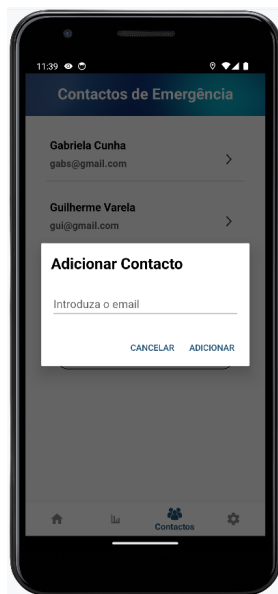
(e) Atividade



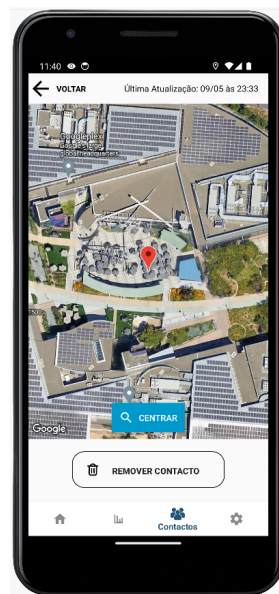
(f) Mapa da Atividade



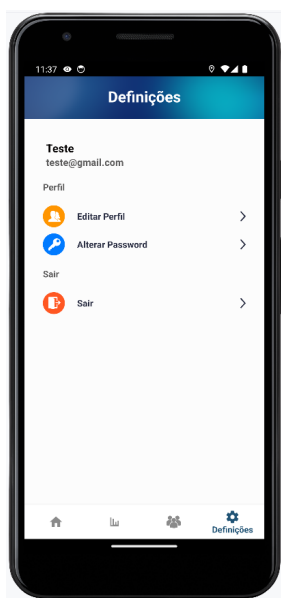
(g) Contactos



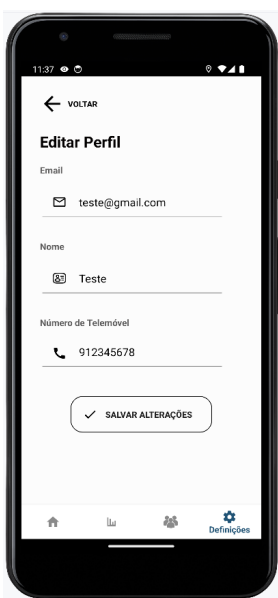
(h) Adicionar Contacto



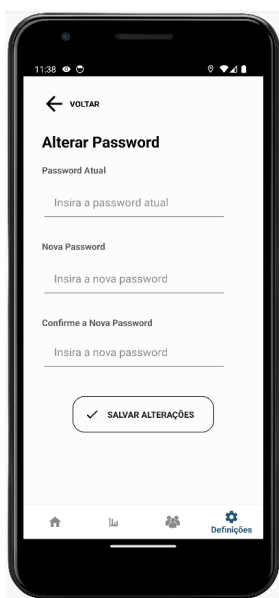
(i) Contacto



(j) Definições



(k) Editar Perfil



(l) Alterar Password

9 Limitações e Trabalho Futuro

Nesta secção, exploraremos as limitações atuais da aplicação, destacando as áreas que exigem aprimoramento contínuo e sugerindo possíveis direções para trabalho futuro.

Um dos desafios significativos encontrados na aplicação é a falta de distinção entre o utilizador principal (idoso) e os contactos de emergência designados. Atualmente, mesmo quando o utilizador é apenas um responsável pelo cuidado de um idoso, os dados do acelerómetro são continuamente recolhidos, o que resulta num consumo desnecessário de bateria e recursos do dispositivo.

Outra lacuna importante trata-se da ausência de um sistema de pedidos de amizade no contexto dos contactos de emergência. Atualmente, os utilizadores têm a capacidade de adicionar contactos de emergência sem que estes tenham a opção de aceitar ou recusar esse papel, ou seja, sem consentimento explícito. Esta abordagem pode resultar em comunicações indesejadas ou até mesmo em violações de privacidade.

Atualmente, a aplicação já disponibiliza o histórico de quedas e emergências juntamente com a localização e *timestamp* das mesmas, mas há espaço para melhorias na apresentação dos dados. A inclusão de estatísticas como o número de quedas por mês ou por zona poderia ser valiosa para a monitorização do idoso, podendo até eventualmente ser oferecidas como um recurso adicional, exclusivo para os utilizadores *premium*.

Por último, como área de aprimoramento contínuo na aplicação temos a melhoria do algoritmo de deteção de quedas, onde a implementação de técnicas de *machine learning* pode oferecer uma deteção mais sensível e específica, em comparação com o algoritmo atual, baseado em regras. Com os dados já a ser recolhidos e enviados para a base de dados, o *machine learning* pode ser aplicado para aprender com padrões históricos e em tempo real, resultando numa deteção mais precisa de quedas e reduzindo falsos positivos.

10 Conclusão

Em conclusão, a realização deste trabalho prático representou uma oportunidade significativa de aprendizagem para o grupo. Primeiramente, tivemos o nosso primeiro contacto prático em grande escala com a sensorização, uma área fundamental para a construção de soluções tecnológicas modernas e eficazes. Além disso, tivemos também uma experiência completamente nova para todos os membros do grupo relativamente ao desenvolvimento de aplicações móveis, onde pudemos aprender e aplicar novas tecnologias.

O grupo encontra-se bastante satisfeito com os resultados alcançados, dado que o algoritmo de deteção de queda se mostrou extremamente robusto. Ao realizar várias experiências a englobar atividades com padrões de movimentos

abruptos e intensos, tais como correr e saltar, verificamos que o algoritmo consegue efetivamente diferenciar estas situações com elevadas oscilações na aceleração das verdadeiras quedas, minimizando falsos positivos e aumentando a confiabilidade da aplicação em cenários do dia a dia. Não obstante, estamos conscientes de algumas lacunas e foram sugeridos ao longo do relatório vários aspectos que poderiam vir a ser melhorados. Assim, acreditamos que este projeto contribuiu significativamente para a nossa formação acadêmica e profissional e motivou-nos a explorar ainda mais estas áreas tecnológicas futuramente.

Referências

- [1] Tri Dang Tran, Hai Truong e Tran Khanh Dang. “Automatic fall detection using smartphone acceleration sensor”. Em: *International Journal of Advanced Computer Science and Applications* 7.12 (2016).