

Smartflow simplificando o consumo de água **Danielle Kadanus, Felipe Del Caro, Lauriana** **Paludo**

Relatório Técnico GTI 2024-1



Relatório Técnico

Smartflow simplificando o consumo de água

KADANUS, Danielle – kadanusynad@gmail.com

DEL CARO, Felipe – nextgamesoldier@gmail.com

PALUDO, Lauriana – lauriana.paludo@ifpr.edu.br

Resumo

Este trabalho apresenta o projeto de desenvolvimento do aplicativo para dispositivos Android. SmartFlow é, a princípio, um aplicativo para Android que permite ao usuário controlar seu consumo de água em m³ e acompanhar os valores da sua fatura até o momento do cálculo. Ele armazena o histórico de consumo e, com base nesse histórico, tem a capacidade de ajudar na detecção de possíveis vazamentos. O objetivo principal é ensinar como ler o hidrômetro e compreender a fatura.

O aplicativo SmartFlow solicita, no primeiro uso, a leitura anterior, que pode ser obtida na fatura; em seguida, pede a leitura atual do hidrômetro. Após a inserção desses dados pelo usuário, realiza o cálculo e exibe na segunda tela o valor exato da fatura até o momento, com base nos valores da Sanepar. Também apresenta uma lista do consumo até aquele momento. Além disso, possui a função de ensinar o usuário a ler seu hidrômetro e fatura.

Com o SmartFlow, os clientes da Sanepar podem monitorar de forma eficiente o consumo de água, acompanhar as informações de forma clara e acessível e evitar deslocamentos desnecessários aos escritórios de atendimento. Essa solução contribui para a conscientização sobre o consumo consciente de água e promove a adoção de práticas mais sustentáveis.

Palavras-chave: Aplicativo, Consumo de Água, SmartFlow, Sustentabilidade.

1 – Introdução

A compreensão da fatura de água e o monitoramento do consumo representam desafios significativos para os clientes da Companhia de Saneamento do Paraná, Sanepar.

A proposta de desenvolvimento do aplicativo SmartFlow surge como uma solução inovadora para enfrentar essas dificuldades, trazendo benefícios tanto para os clientes como para a própria empresa.

O Projeto visa atender aos Objetivos de Desenvolvimento Sustentável do Brasil: ODS 6: Água Potável e Saneamento e ODS 11: Cidades e Comunidades Sustentáveis) contribuindo para a construção de um futuro sustentável.

Para os clientes, o Smartflow oferece uma ferramenta intuitiva que facilita o acompanhamento do consumo de água e a compreensão da fatura mensal. Ao inserirem as leituras do hidrômetro, os usuários poderão identificar áreas de desperdício, adotar medidas para reduzir o consumo e, conseqüentemente, economizar financeiramente. Com informações claras e acessíveis sobre o consumo ao longo do tempo, os clientes serão capacitados a tomar decisões mais conscientes e sustentáveis em relação ao uso da água. Dessa forma, o aplicativo promove a consciência ambiental e contribui para a preservação dos preciosos recursos hídricos.

Além disso, ao fornecer uma ferramenta intuitiva e acessível para os clientes acompanharem e compreenderem seu consumo de água, reduzirá a desinformação e, conseqüentemente, a necessidade de deslocamentos desnecessários aos atendimentos presenciais.

Em suma, a pesquisa para o desenvolvimento do aplicativo SmartFlow justifica-se pela necessidade de facilitar o acompanhamento e compreensão da fatura de água.

Para o desenvolvimento do projeto fez-se necessário uma pesquisa que identificasse algum outro trabalho desenvolvido ou semelhante à proposta atual. Como critério temporal para esta pesquisa foram considerados apenas artigos revisados por pares dos últimos 5 anos (2015-2019) tendo em vista a atualidade. Foram analisados estudos disponíveis nos periódicos das bases Science Direct, Scielo, ERIC, Google Acadêmico e Portal de Periódicos da Capes.

Resultados obtidos:

Fontes de dados	Artigos retornados	Artigos selecionados
Portal de periódicos Capes	0	0
Google acadêmico	1630	55

Posteriormente, foi realizada a leitura dos trabalhos na íntegra e uma nova etapa de filtragem, onde foram excluídos os artigos que não se enquadram no escopo da pesquisa.

O projeto compartilha semelhanças com os trabalhos analisados na pesquisa, uma vez que todos têm em comum a preocupação com o uso eficiente da água e a busca por soluções que ajudem os consumidores a monitorar e entender seu consumo.

No entanto, o projeto atual se destaca pela abordagem prática de criar uma ferramenta intuitiva, o aplicativo, que visa não apenas fornecer informações claras sobre o consumo de água, mas também reduzir a desinformação e a necessidade de atendimentos nos escritórios da Sanepar. Isso é uma resposta direta ao problema identificado pelo alto volume de atendimentos devido à falta de compreensão por parte dos clientes sobre suas faturas de água.

Os Objetivos de Desenvolvimento Sustentável (ODS) da ONU fornecem uma estrutura global para promover o desenvolvimento sustentável e enfrentar desafios globais. O aplicativo SmartFlow está alinhado com os seguintes ODS:

- **ODS 6: Água Potável e Saneamento**

Este objetivo busca garantir a disponibilidade e a gestão sustentável da água e do saneamento para todos. O SmartFlow contribui para este objetivo ao facilitar a gestão do consumo de água, promovendo a conscientização e a redução do desperdício.

- **ODS 11: Cidades e Comunidades Sustentáveis**

Este objetivo visa tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis. O aplicativo SmartFlow contribui para a sustentabilidade das comunidades ao melhorar a gestão dos recursos hídricos e auxiliar na necessidade de deslocamentos para atendimentos presenciais.

Trabalhos Relacionados

A seção de Trabalhos Relacionados visa posicionar o seu projeto dentro do contexto das pesquisas e desenvolvimentos existentes na área. Para o projeto SmartFlow, consideramos dois trabalhos relevantes:

Trabalho 1: "Medidor de Consumo de Água e Energia Residencial com Monitoramento Remoto"

- Autores: Gabriel Panema da Silva, Cristiano Akamine (Orientador)
- Descrição: Este trabalho aborda o desenvolvimento de um sistema para monitoramento

remoto do consumo de água e energia elétrica em residências, utilizando um computador Raspberry Pi. O sistema é projetado para coletar dados de consumo e fornecer alertas sobre anomalias detectadas, promovendo uma análise eficaz e em tempo real dos recursos consumidos.

- Tecnologias Utilizadas: Raspberry Pi, análise de dados, alertas de anomalias.
- Objetivo: Criar um sistema integrado que permita o monitoramento remoto e a análise dos dados de consumo de água e energia elétrica, com foco na detecção de irregularidades e otimização do uso dos recursos.
- Relevância para o SmartFlow: Embora o sistema descrito utilize tecnologias diferentes, como o Raspberry Pi, e tenha um foco mais amplo que inclui energia elétrica, a abordagem de monitoramento remoto e análise de dados é semelhante ao propósito do SmartFlow de oferecer insights sobre o consumo de água e identificar possíveis problemas, como vazamentos. A capacidade de fornecer alertas sobre anomalias pode inspirar funcionalidades semelhantes no SmartFlow, especialmente para detecção de vazamentos.

Trabalho 2: "Controle de Consumo de Água Baseado em Reconhecimento Facial"

- Autores: Pedro E. Santos, Roubert Edgar S. Silva, Dalmy F. Carvalho Jr, Ilo A. S. Rivero, Raquel A. F. Mini
- Descrição: Este trabalho apresenta uma solução inovadora para o controle do consumo de água, utilizando tecnologia de reconhecimento facial para gerenciar o fluxo de água em dispositivos residenciais. O objetivo é permitir um controle mais preciso e personalizado do consumo de água com base na identidade dos usuários.
- Tecnologias Utilizadas: Reconhecimento facial, controle de fluxo de água.
- Objetivo: Desenvolver um sistema que usa reconhecimento facial para controlar o consumo de água em dispositivos hídricos, promovendo a economia de recursos e a personalização do uso.

- **Relevância para o SmartFlow:** O uso do reconhecimento facial no trabalho descrito representa uma abordagem tecnológica avançada para o controle do consumo de água. Embora o SmartFlow não utilize reconhecimento facial, a ideia de controle personalizado e a promoção de economia de água são princípios semelhantes. O SmartFlow foca mais na interpretação das leituras do hidrômetro e na análise de consumo, proporcionando uma ferramenta educativa e prática para gestão da água. O reconhecimento facial pode inspirar futuras implementações para um controle ainda mais refinado.

Resumo Comparativo

Os dois trabalhos relacionados abordam o tema do consumo de recursos em residências, mas com abordagens distintas:

- Trabalho 1 se concentra em monitoramento remoto e análise de dados para água e energia elétrica, utilizando Raspberry Pi e sistemas de alerta para anomalias. Isso é relevante para o SmartFlow em termos de monitoramento e análise de dados, embora com um foco mais amplo.
- Trabalho 2 utiliza reconhecimento facial para controlar o fluxo de água, promovendo economia e personalização. Embora não diretamente aplicável ao SmartFlow, a ideia de controle e personalização do consumo é relevante para futuras explorações de funcionalidades.

Esses trabalhos fornecem contextos e tecnologias que podem complementar o desenvolvimento do SmartFlow, oferecendo inspiração para aprimorar a funcionalidade e a eficácia do aplicativo na gestão e conscientização sobre o consumo de água.

Justificativa

A falta de compreensão das faturas de água e o monitoramento inadequado do consumo levam a um uso ineficiente dos recursos hídricos e a um aumento nas despesas dos clientes com água. Muitas pessoas têm dificuldade em interpretar os dados de suas faturas e em monitorar seu consumo de forma efetiva, o que pode resultar em desperdício e altos custos.

O SmartFlow justifica-se como uma solução inovadora para esses problemas, oferecendo uma ferramenta que permite aos usuários monitorar seu consumo de água de forma intuitiva e educativa. Com o SmartFlow, os clientes da Sanepar poderão:

- **Compreender Melhor Suas Faturas:** O aplicativo fornece uma explicação clara de como ler o hidrômetro e a fatura.

- **Detectar Vazamentos:** O histórico de consumo ajuda a identificar possíveis vazamentos.

- **Promover o Uso Consciente da Água:** Facilita a adoção de práticas sustentáveis ao fornecer informações detalhadas sobre o consumo.

A implementação do SmartFlow contribui para a redução do desperdício de água e para a economia financeira dos usuários, ao mesmo tempo que apoia a Sanepar na melhoria da comunicação e do atendimento ao cliente.

Objetivos

Objetivo Geral:

- Desenvolver um aplicativo móvel para Android que permita aos usuários monitorar seu consumo de água, entender suas faturas e identificar possíveis vazamentos, promovendo o uso consciente e sustentável dos recursos hídricos.

Objetivos Específicos:

1. **Facilitar a Leitura do Hidrômetro e Compreensão da Fatura:** Oferecer uma interface intuitiva que ajude os usuários a inserir e interpretar dados do hidrômetro e da fatura.

2. **Proporcionar um Histórico de Consumo:** Armazenar e exibir o histórico de consumo de água para ajudar na análise e detecção de padrões.

3. **Detectar Vazamentos Potenciais:** Utilizar o histórico de consumo para identificar possíveis vazamentos e alertar os usuários.

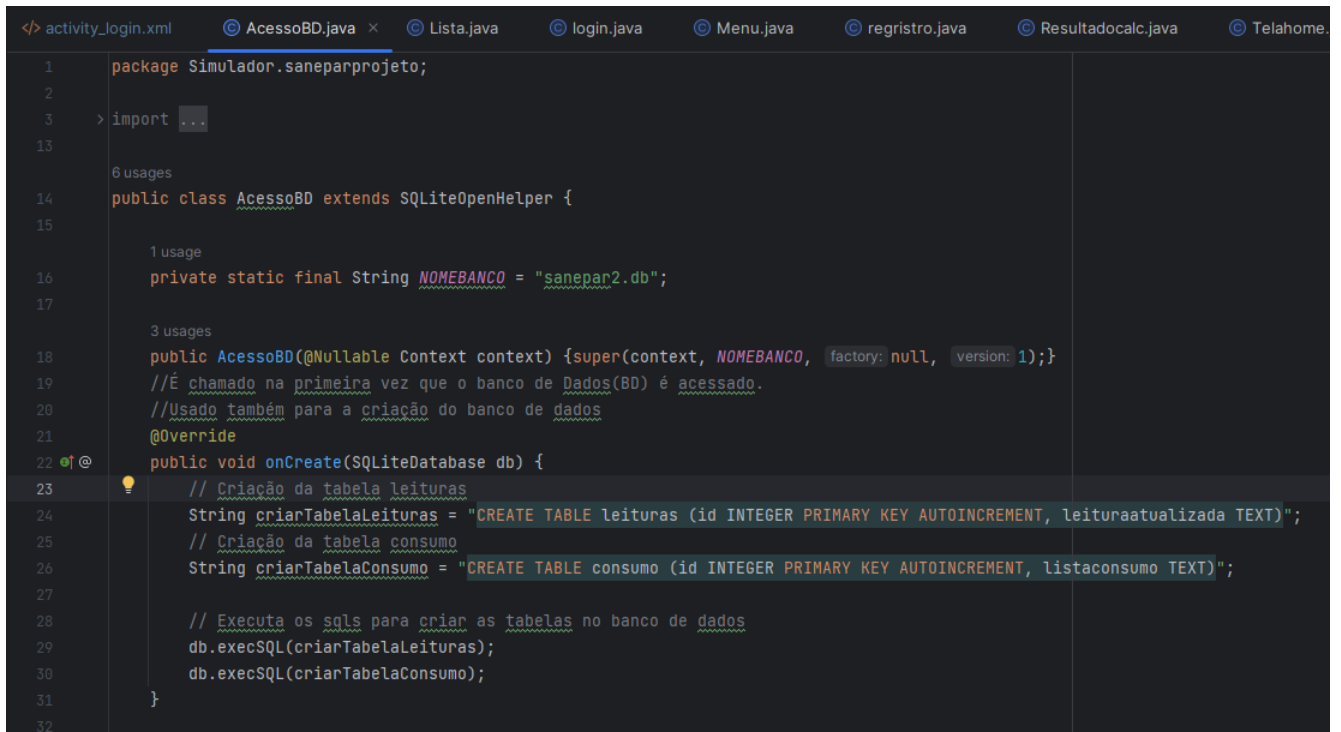
4. **Educar sobre Consumo de Água:** Fornecer informações e orientações sobre práticas de consumo consciente e sustentável.

5. **Reduzir Deslocamentos para Atendimento Presencial:** Minimizar a necessidade de atendimento presencial ao oferecer uma ferramenta completa para a gestão do consumo de água.

1 – Materiais e Métodos

Desenvolvimento do Banco de Dados SQLite:

→ O banco de dados SQLite foi utilizado para armazenar as leituras do hidrômetro e o histórico de consumo. Foi criada a classe `AcessoBD` para estender `SQLiteOpenHelper`, gerenciando a criação, atualização e operações de inserção e consulta de dados.



```
<> activity_login.xml  AcessoBD.java  Lista.java  login.java  Menu.java  registro.java  Resultadocalc.java  Telahome.

1  package Simulador.saneparprojeto;
2
3  > import ...
13
14  6 usages
15  public class AcessoBD extends SQLiteOpenHelper {
16
17      1 usage
18      private static final String NOMEBANCO = "sanepar2.db";
19
20      3 usages
21      public AcessoBD(@Nullable Context context) {super(context, NOMEBANCO, factory: null, version: 1);}
22      //É chamado na primeira vez que o banco de Dados(BD) é acessado.
23      //Usado também para a criação do banco de dados
24      @Override
25      public void onCreate(SQLiteDatabase db) {
26          // Criação da tabela leituras
27          String criarTabelaLeituras = "CREATE TABLE leituras (id INTEGER PRIMARY KEY AUTOINCREMENT, leituraatualizada TEXT)";
28          // Criação da tabela consumo
29          String criarTabelaConsumo = "CREATE TABLE consumo (id INTEGER PRIMARY KEY AUTOINCREMENT, listaconsumo TEXT)";
30
31          // Executa os sqls para criar as tabelas no banco de dados
32          db.execSQL(criarTabelaLeituras);
33          db.execSQL(criarTabelaConsumo);
34      }
35  }
```

Desenvolvimento das Telas de Interface do Usuário (UI):

✓ As telas de interface do usuário foram desenvolvidas com o auxílio do framework Android. As classes `Telahome`, `Lista`, `Resultadocalc`, `login`, `Menu` e `registro` foram criadas para implementar as diferentes funcionalidades e telas do aplicativo. Cada classe está associada a uma atividade específica, como cálculos de consumo, exibição de listas de consumo, administração de login de usuários, entre outras.


```
activity_login.xml  AcessoBD.java  Lista.java  login.java  Menu.java  registro.java  Resultadocalc.java  Telahome.java x
23      @Override
24      protected void onCreate(Bundle savedInstanceState) {
25          super.onCreate(savedInstanceState);
26          setContentView(R.layout.home);
27
28          leitura_atual = findViewById(R.id.leitura1);
29          leitura_anterior = findViewById(R.id.leitura2);
30          consumosimulado = findViewById(R.id.resultconsumo);
31          resultado2 = findViewById(R.id.resultadofinal);
32          Calcular = findViewById(R.id.Calcular);
33
34          Listaconsumo = findViewById(R.id.listaconsumo);
35
36          acessoBD = new AcessoBD( context: this);
37
38          String leituraAnterior = acessoBD.getUltimaLeitura();
39          leitura_anterior.setText(leituraAnterior);
40
41          Listaconsumo.setOnClickListener(new View.OnClickListener() {
42              @Override
43              public void onClick(View view) {
44                  Intent in = new Intent( packageContext: Telahome.this, Lista.class);
45                  startActivity(in);
46              }
          })
      }
```

✓ Integração com Firebase Authentication:

A autenticação de usuários foi gerenciada através do Firebase Authentication. A lógica foi

```
activity_login.xml  AcessoBD.java  Lista.java  login.java x  Menu.java  registro.java
45      @Override
46      protected void onCreate(Bundle savedInstanceState) {
47          super.onCreate(savedInstanceState);
48          setContentView(R.layout.activity_login);
49          mAuth = FirebaseAuth.getInstance();
50          editTextUsername = findViewById(R.id.editTextUsername);
51          editTextPassword = findViewById(R.id.editTextPassword);
52          buttonLogin = findViewById(R.id.buttonLogin);
53          Register = findViewById(R.id.Register);
54          progresso = findViewById(R.id.progresso);
55          checkbox = findViewById(R.id.checkbox);
56
57          buttonLogin.setOnClickListener(new View.OnClickListener() {
58              @Override
59              public void onClick(View view) {
60                  String email = editTextUsername.getText().toString();
61                  String senha = editTextPassword.getText().toString();
62                  if(!TextUtils.isEmpty(email) || !TextUtils.isEmpty(senha)){
63                      progresso.setVisibility(View.VISIBLE);
64                      mAuth.signInWithEmailAndPassword(email,senha)
65                          .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
66                              no usages
67                              @Override
68                              public void onComplete(@NonNull Task<AuthResult> task) {
69                                  if(task.isSuccessful()){
69                                      abrirTelaPrincipal();
69                                  }
69                              }
69                          })
69                  }
69              }
69          })
69      }
```

implementada nas classes de login e registro para permitir aos usuários que façam login e se registrem no aplicativo.

```
33
34      @Override
35      protected void onCreate(Bundle savedInstanceState) {
36          super.onCreate(savedInstanceState);
37          setContentView(R.layout.activity_registro);
38
39          mAuth = FirebaseAuth.getInstance();
40          editTextNewUsername = findViewById(R.id.edmail);
41          editTextNewPassword = findViewById(R.id.edsenha);
42          confirmarsen = findViewById(R.id.confirmarsenha);
43          checkBox = findViewById(R.id.checkbox);
44          registrar = findViewById(R.id.btregistro);
45          barraprog = findViewById(R.id.progresso);
46
47          checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
48              2 usages
49              @Override
```

✓ **Implementação Futura de Telas Adicionais:**

Planeja-se implementar telas adicionais para permitir que os usuários registrem o consumo de água de aparelhos específicos, como chuveiros e máquinas de lavar. Também está prevista a substituição da autenticação via Firebase por um sistema de login e registro no banco de dados local para possibilitar multi-logins com seus respectivos dados. Além disso, será desenvolvida uma tela para auxiliar na compreensão da fatura.

3 – Smartflow

- ✓ Criamos a persona representativa do usuário alvo do nosso aplicativo e histórias para guiar no desenvolvimentos das funcionalidades que devem estar presentes no smartflow.



MARINA SOUZA

VENDEDORA

ADULTO DE MEIA-IDADE (41-59)

Mini-bio

"Olá, sou um consumidor consciente e interessado em entender melhor o uso da água. Como cliente da Sanepar, reconheço a importância de monitorar meu consumo mensalmente para tomar medidas práticas em prol da sustentabilidade. Com o SmartFlow, vejo uma oportunidade de compreender de forma mais clara e simples minha fatura de água, algo que muitas vezes foi desafiador no passado. Acredito firmemente que essa ferramenta pode não apenas ajudar a reduzir o volume de visitas aos escritórios da Sanepar devido a dúvidas nas faturas, mas também contribuir para a conscientização sobre práticas sustentáveis. Além disso, a detecção de vazamentos e alertas de consumo anormal seriam extremamente úteis para mim. Pessoalmente, estou disposto a usar o SmartFlow para inserir leituras do hidrômetro e acompanhar meu consumo de água, especialmente se houver oportunidade de participar de testes beta e fornecer feedback para aprimorar ainda mais a ferramenta. Estou empolgado com a ideia de uma ferramenta que não apenas simplifique o entendimento da minha fatura de água, mas que também ajude a promover mudanças positivas no consumo consciente de água. Espero que o SmartFlow se torne uma plataforma que não apenas beneficie os usuários individuais, mas também contribua para um impacto mais amplo em nossa comunidade."

✓ Usamos o Trello para organizar nossas atividades:

Link: <https://trello.com/invite/b/656cbd63876120fbec774152/ATTI154ff3f4630c43b4e196b1f7a-ab1543107D7CCED/gestao-do-app-smartflow>

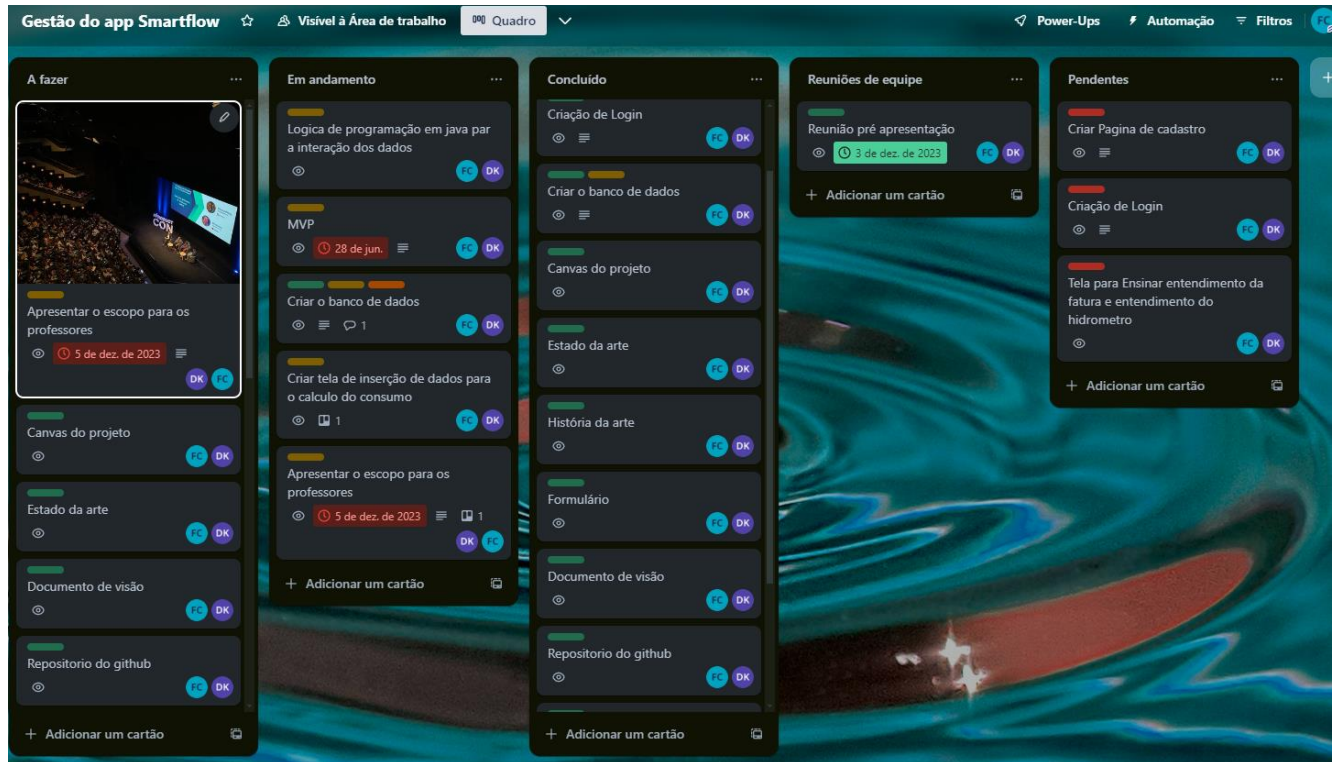
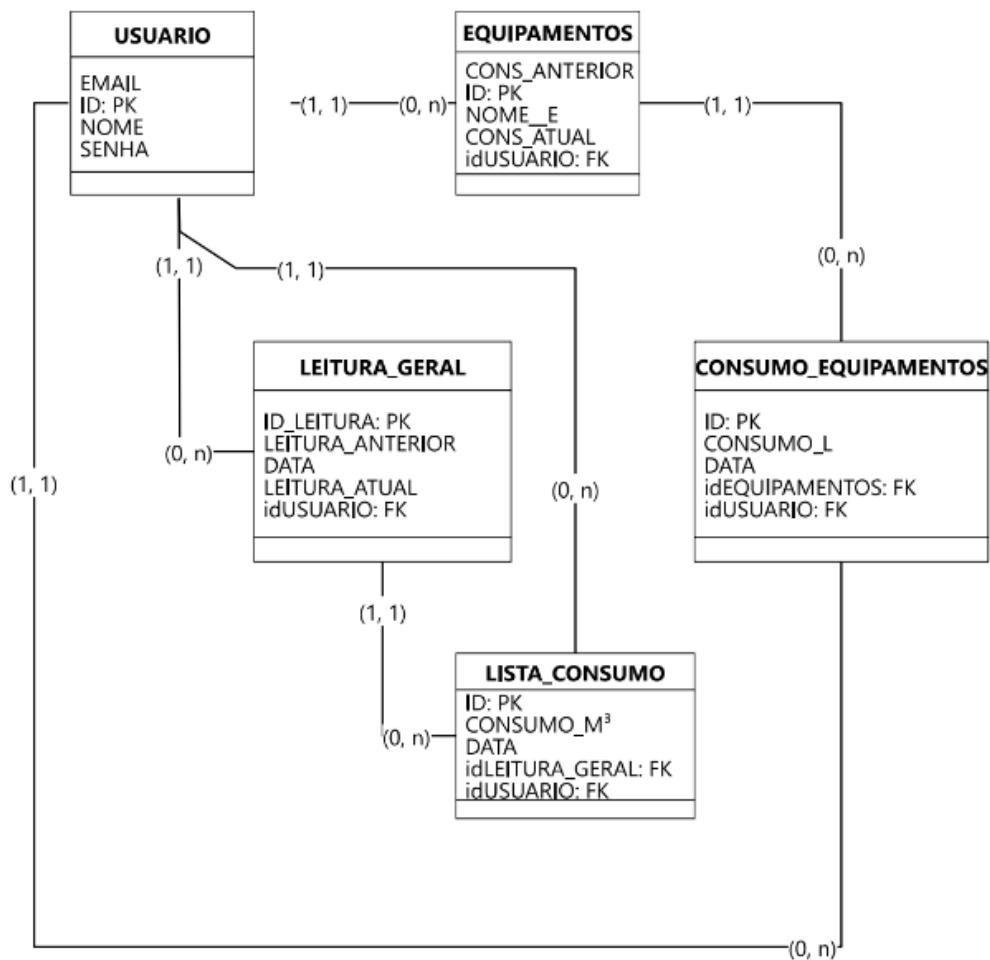


Diagrama de classes:



Class Diagram0

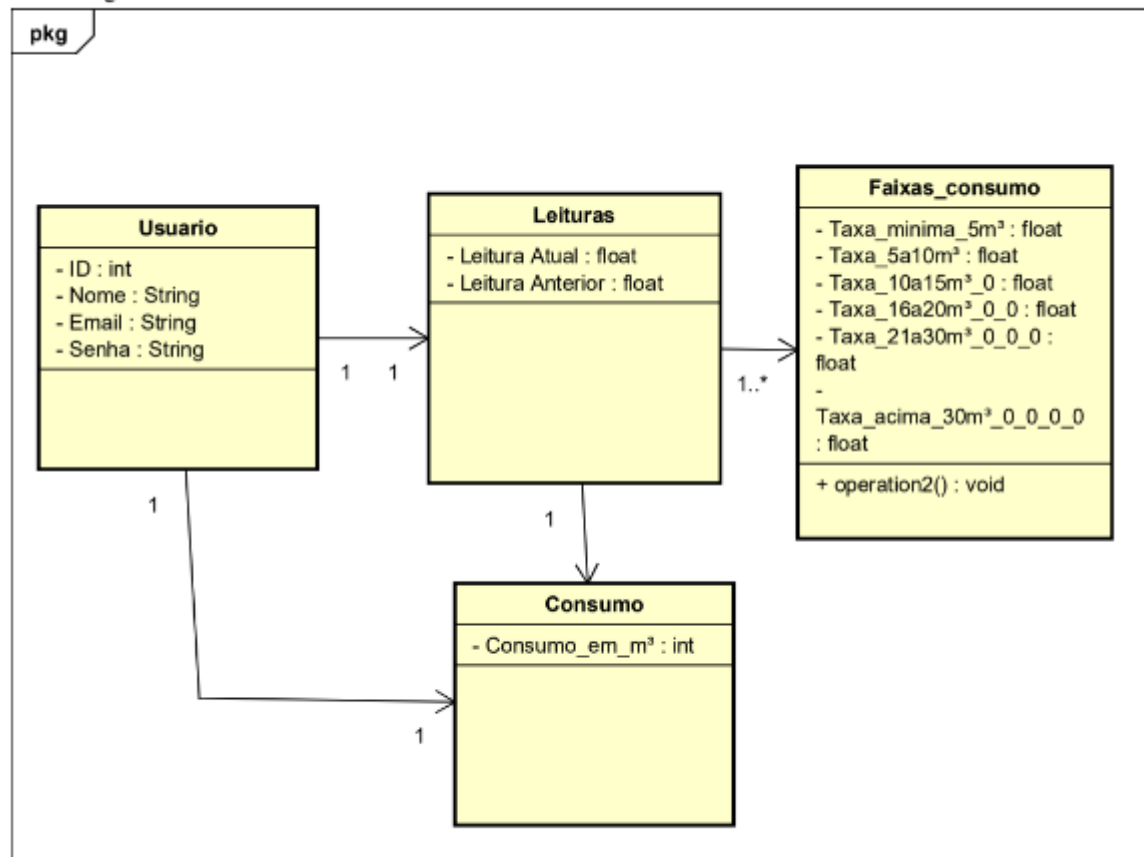
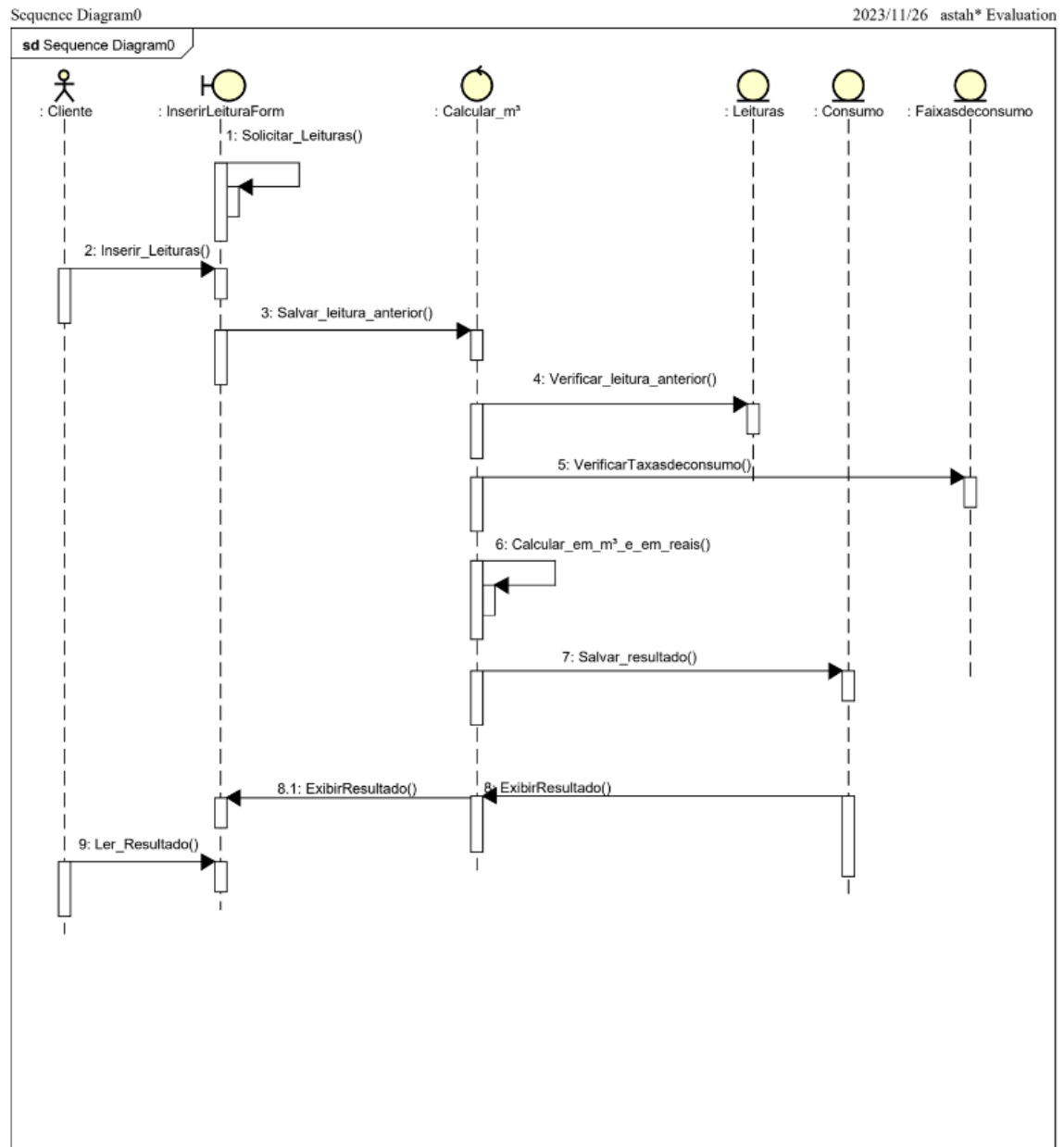


Diagrama de sequência UML



Especificação de testes de Desenvolvedor

Após levantamento dos requisitos criamos as principais funções que são as leituras do hidrômetro, e a inserção no banco de dados dessa leitura, retorno do banco de dados da última leitura inserida pelo usuário e a medida que executamos os testes foram aprimoradas, outro teste também foi do cálculo das taxas, estes foram os principais testes para funcionar o MVP, como mostramos abaixo:

Leituras do Hidrômetro e Inserção no Banco de Dados

Código:

```
public void salvarLeituraAtual(String leitura, String dataBrasileira) { 1 usage 1 kane
    SQLiteDatabase db = getWritableDatabase();
    try {
        ContentValues valores = new ContentValues();
        valores.put("leitura_atualizada", leitura);
        valores.put("data", converterParaFormatoSQL(dataBrasileira));
        db.insert(table: "leituras", nullColumnHack: null, valores);
    } finally {
        db.close();
    }
}
```

Este método é responsável por salvar a leitura do hidrômetro no banco de dados, incluindo a data da leitura formatada para o padrão SQL.

```
public String obterUltimaLeitura() { 1 usage 1 kane +1
    String leituraAnterior = "";
    SQLiteDatabase db = getReadableDatabase();
    Cursor cursor = null;
    try {
        String consulta = "SELECT leitura_atualizada FROM leituras ORDER BY id DESC LIMIT 1";
        cursor = db.rawQuery(consulta, selectionArgs: null);
        if (cursor != null && cursor.moveToFirst()) {
            leituraAnterior = cursor.getString(cursor.getColumnIndexOrThrow("leitura_atualizada"));
        }
    } finally {
        if (cursor != null) {
            cursor.close();
        }
        db.close();
    }
    return leituraAnterior;
}
```

Este método retorna a última leitura registrada no banco de dados, ordenada pela ID em ordem decrescente e limitada a 1 resultado.

Cálculo das Taxas e Exibição dos Resultados

Código:

```
Calcular.setOnClickListener(new View.OnClickListener() { // kane +1
    @Override // kane +1
    public void onClick(View v) {
        String n1 = leitura_atual.getText().toString();
        String n2 = leitura_anterior.getText().toString();
        String data = data_leitura.getText().toString();
        String data2 = data_leitura2.getText().toString();

        double intervcons = 5; // intervalo de taxas de consumo em metros cúbicos
        double intervconsa30 = 10; // intervalo de taxas de consumo em metros cúbicos acima de 30

        // Valores atualizados
        double taxamin = 50.42;
        double taxa1 = 1.56; // de 6 a 10m³
        double taxa2 = 8.69; // de 11 a 15m³
        double taxa3 = 8.73; // de 16 a 20m³
        double taxa4 = 8.81; // de 21 a 30m³
        double taxa5 = 14.90; // acima de 30m³

        // Valores atualizados do esgoto (85% do valor da água)
        double esgoto0 = 42.85; // taxa fixa até 5m²
        double esgoto1 = 42.85 / 5; // valor da taxa de esgoto para cada m³, considerando 0 a 5m³, mais util para os calculos
        double esgoto2 = 6.63 / 5; // mesma coisa, porém com taxa de 6 a 10m³
        double esgoto3 = 36.93 / 5; // 11 a 15m³
        double esgoto4 = 37.10 / 5; // 16 a 20m³
        double esgoto5 = 74.89 / 10; // 21 a 30m³
        double esgoto6 = 12.66; // acima de 30m³

        // Verifica se n1 ou n2 estão vazias; se estiverem, sai da função
        if (n1.isEmpty() || n2.isEmpty() || data.isEmpty()) {
            return; // Sai da função se n1 ou n2 ou data estiverem vazias
        }
    }
});
```

```
valortotal = soma + soma_esgoto;
acessoBD.salvarConsumo(consumo, valortotal, data2);
acessoBD.salvarLeituraAtual(n1, data); // Salva a leitura atual com a data
double caixas = consumo * 1000 / 500;
```

Este código calcula o consumo e o custo baseado nas leituras fornecidas pelo usuário. Se as leituras e a data forem válidas, ele salva o consumo e o custo no banco de dados e atualiza a interface com os resultados calculados.

Layout XML para a Tela Inicial

Código:

```
<TextView
    android:id="@+id/leitura11"
    style="@style/BlueButton"
    android:layout_width="233dp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_marginStart="94dp"
    android:layout_marginTop="119dp"
    android:layout_marginEnd="91dp"
    android:drawableLeft="@drawable/baseline_water_drop_24"
    android:text="Leitura atual"
    android:textSize="20sp" />

<EditText
    android:id="@+id/leitura1"
    style="@style/transparencia"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/leitura11"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:inputType="numberDecimal"
    android:textColor="#FFFFFF"
    android:textSize="20sp" />

<!-- TextView para Data da Leitura -->


<!-- EditText para Data da Leitura -->

<TextView
    android:id="@+id/dataLeituraLabel"
    style="@style/BlueButton"
    android:layout_width="246dp"
    android:layout_height="41dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"

    android:layout_alignParentEnd="true"
    android:layout_marginStart="63dp"
    android:layout_marginTop="244dp"
    android:layout_marginEnd="67dp"
    android:text="Data da Leitura"
    android:textSize="20sp" />

<EditText
    android:id="@+id/dataLeitura"
    style="@style/transparencia"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/dataLeituraLabel"
    android:layout_centerHorizontal="true"
```

O layout XML define a interface da tela inicial, incluindo campos para entrada de leitura, botões para calcular o consumo e listar os consumos, e TextViews para mostrar os resultados.



The image shows a mobile application interface for calculating water consumption. The background is a dark, textured image with water droplets. The interface is organized into several sections:

- LEITURA ATUAL**: A section with a water drop icon and a label. Below it is a dark input field containing the number "10".
- DATA DA LEITURA**: A section with a label. Below it is a dark input field containing the date "20/08/2024".
- LEITURA ANTERIOR**: A section with a water drop icon and a label. Below it is a dark input field containing the number "5".
- CONSUMIDO:**: A label followed by the text "Resultado:".
- CALCULA**: A blue button with a white minus sign below the text.
- CONSU**: A blue button with a white plus sign below the text.

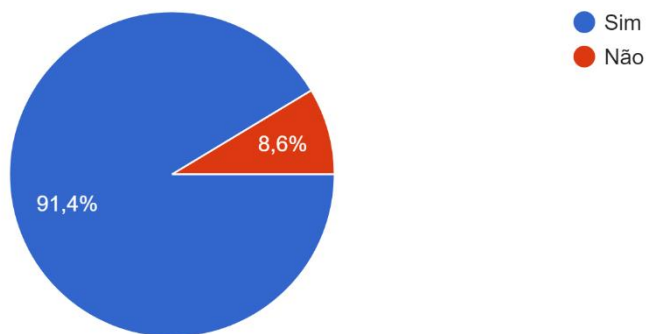
The top status bar shows the time "4:42" and various icons. The bottom of the screen shows a white home indicator bar.

Teste de Usuário

Realizamos teste com usuários as funcionalidades presentes no aplicativo até o momento dos testes aplicados, validamos as funções e requisitos através de formulário do google e obtivemos tais respostas.

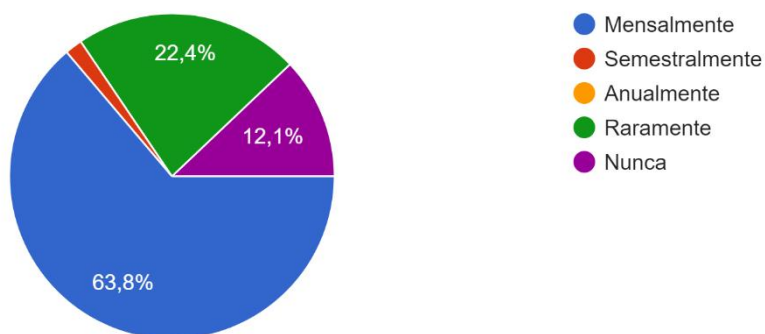
Você é cliente da Sanepar?

58 respostas



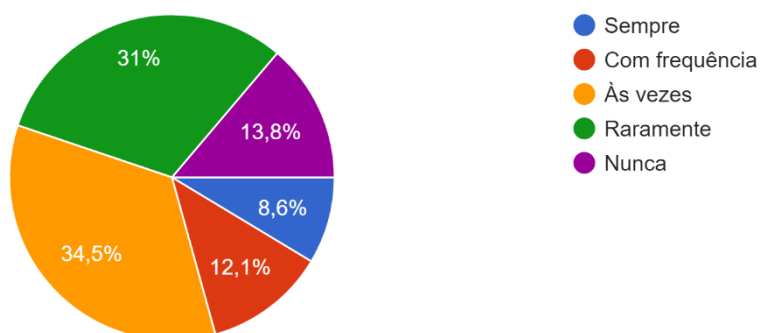
Com que frequência você verifica sua fatura de água?

58 respostas



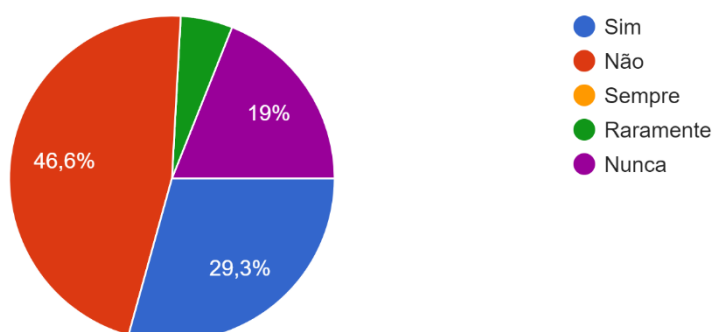
Com que frequência você enfrentou dificuldades na compreensão de sua fatura de água?

58 respostas



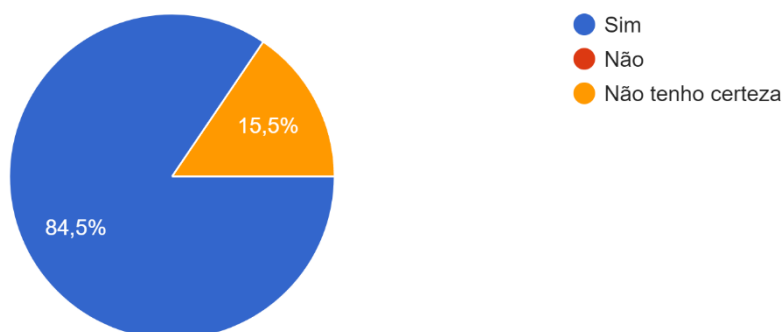
Já precisou ir pessoalmente a um escritório da Sanepar para resolver dúvidas sobre sua fatura de água e seu consumo?

58 respostas



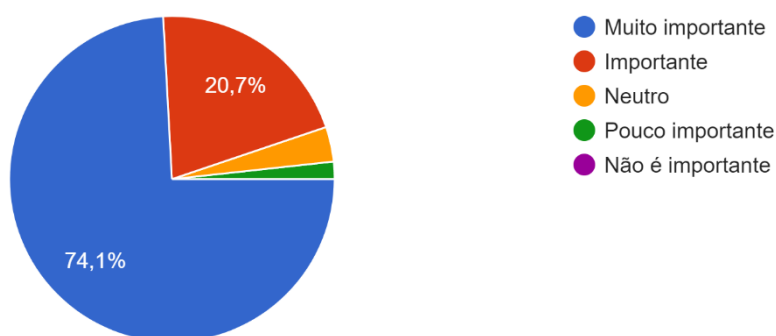
Você acredita que o aplicativo SmartFlow poderia ajudá-lo a entender melhor sua fatura de água e a monitorar seu consumo?

58 respostas



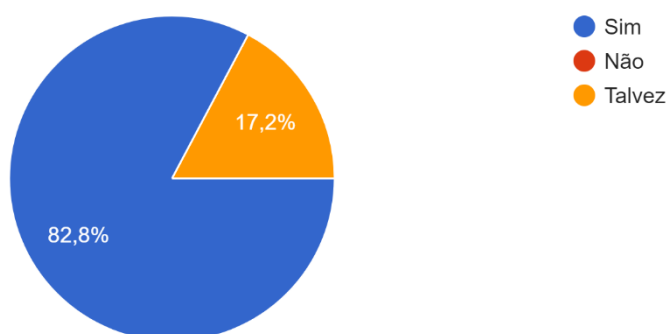
Qual a importância, na sua opinião, de reduzir o volume de atendimentos nos escritórios da Sanepar devido a problemas de compreensão da fatura?

58 respostas



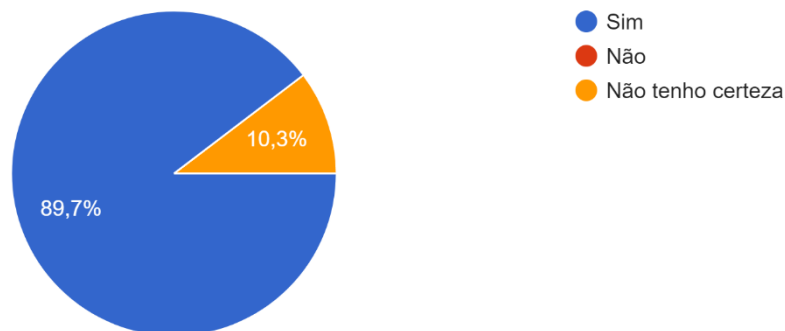
Você estaria disposto a utilizar o SmartFlow para inserir as leituras do hidrômetro e acompanhar seu consumo de água?

58 respostas



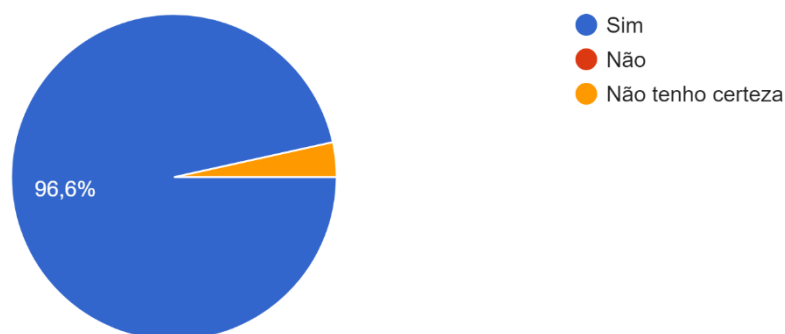
Você acredita que o SmartFlow pode contribuir para a conscientização sobre o consumo consciente de água e práticas sustentáveis?

58 respostas



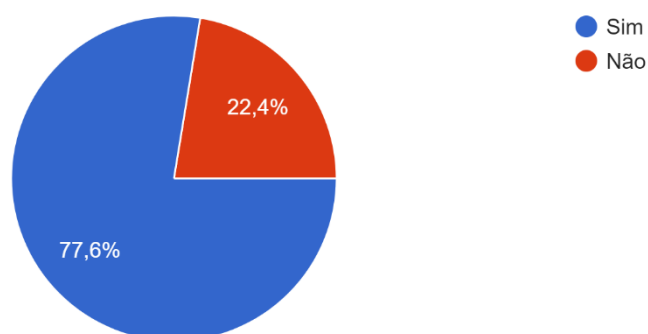
Você acredita que a detecção de vazamentos ou aumento anormal no consumo de água, com notificações pelo SmartFlow, seria útil para você?

58 respostas



Você gostaria de participar de testes beta ou fornecer feedback adicional durante o desenvolvimento do SmartFlow?

58 respostas



Há alguma informação adicional que você gostaria de compartilhar ou alguma sugestão para a melhoria do SmartFlow?

- “Não”
- “Muitas pessoas tem dificuldades em aceitar e entender que tem problema interno de vazamento, essa ferramenta ajuda a monitorar o consumo diário”.
- “Posteriormente poderia ser ampliado para ajudar condomínios a fazer o rateio dos valores para cada apartamento, pois a Sanepar só Lê o relógio geral e acabamos pagando a conta a mais por uso de outras pessoas que não tem consciência de economia de água. Deixam vazamentos em seus apartamentos. Acredito que uma plataforma de aplicativo capaz de gerar o valor de uma conta cobrada pela companhia de saneamento. Poderia ajudar na gestão de contas internas de condomínios que não tem seus relógios individualizados”.
- “Se possível criar um sistema de simulação de consumo; colocando situações do cotidiano, como uso de chuveiro, lavagem de roupa, a fim de demonstrar o que cada habito gera de consumo de água”.
- “a separação de cobrança da coleta de lixo!”.
- “Caso a pessoa tenha poço artesiano, sugestão: O SmartFlow fazer a verificação da pureza da água se está apta pra consumo e avisar de tempos em tempos o momento de adicionar produtos de purificação na água, avisando se no momento está apta pra consumo de ingestão ou para outros fins como lavar roupa, ferver para consumo etc.”.
- “O SmartFlow serve para controle de consumo em condomínio?”.
- Sem dúvidas
- “Que seja um app rápido e prático e com uma linguagem simples para que todos entendam e gostem de acompanhar e realmente se interessem para a economia da água! Que ajude a entender de maneira prática que economizar além de ajudar no bolso, ajuda o meio ambiente”.
- “Dividimos a água em 4 casas”.
- “Faltou explicar um pouco melhor o funcionamento do app, de como seria extraída as informações do consumo de água”.
- “Seria uma ferramenta incrível os consumidores”.

MÉTRICAS

1. Métricas de Desempenho

- Tempo de Carregamento de algumas de nossas telas mais importantes
- Activity Telahome:
 - Tempo Médio de Carregamento: 350 ms
 - Descrição: O tempo de carregamento foi medido desde o início da criação da Activity até a exibição completa da interface. O tempo médio está dentro da faixa aceitável para uma boa experiência de usuário.
- Activity Lista:
 - Tempo Médio de Carregamento: 420 ms
 - Descrição: O tempo de carregamento inclui a inicialização das listas e a configuração dos adaptadores. Embora aceitável, há espaço para otimização, especialmente na carga inicial de dados.
 - Tempo de Resposta ao Clique
- Botão "Consultar" em Lista:
 - Tempo Médio de Resposta: 280 ms
 - Descrição: O tempo desde o clique até a atualização da ListView com dados filtrados. O desempenho está bom, mas a resposta pode ser melhorada em dispositivos mais antigos.
- Botão "Calcular" em Telahome:
 - Tempo Médio de Resposta: 310 ms
 - Descrição: O tempo necessário para calcular e exibir o consumo e o custo após o clique no botão. Adequado para a maioria dos dispositivos, mas considerações devem ser feitas para otimizações futuras.
 - Uso de Recursos
- Espaço em disco:
 - Uso Médio de Memória: média de 45mb, considerando o possível crescimento dos dados no banco
 - Descrição: O aplicativo utiliza uma quantidade moderada de memória em disco e de ram, o que é adequado para a maioria dos dispositivos Android.
- Foram utilizados para a medição a ferramenta Profiler do próprio Android Studio e o logcat, através dos logs, pudemos fazer essas verificações.

2. Guia de Estilo e Padrões Adotados

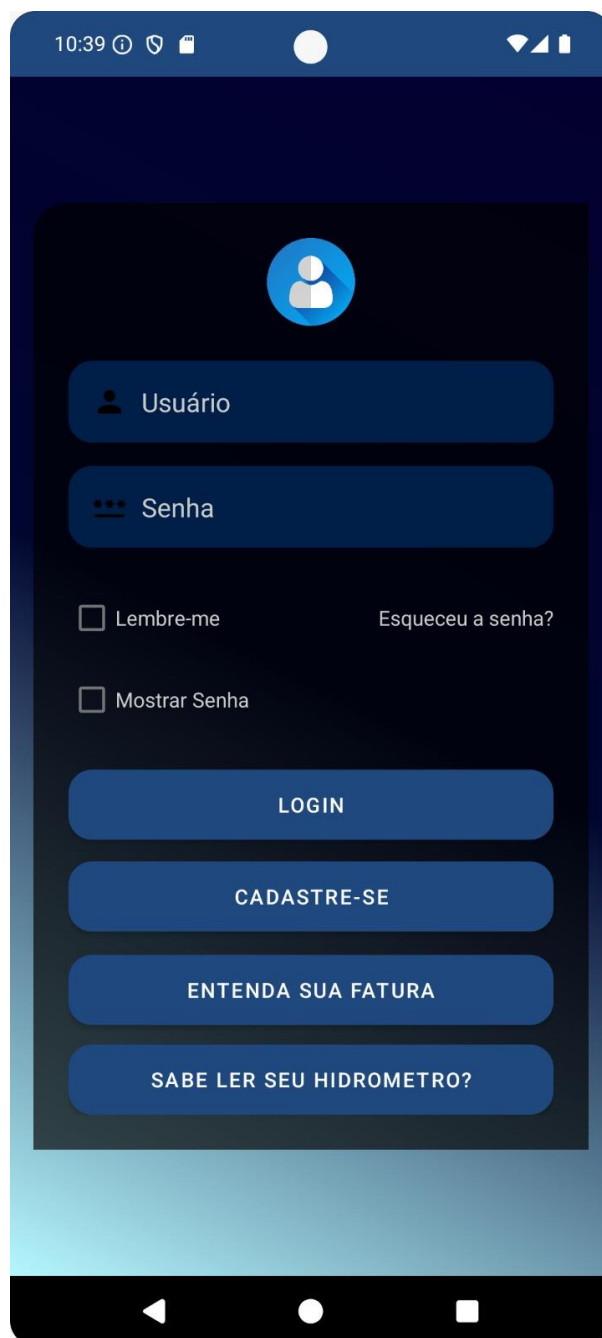
- 2.1 Normas Visuais
- Paleta de Cores:
 - Fundo: #FFFFFF (branco) ou #E0E0E0 (cinza claro)
 - Texto: #000000 (preto) para textos principais; #FFFFFF (branco) para textos sobre fundos escuros
 - Botões:
 - Primário: #009688 (verde água)
 - Secundário: #FF5722 (laranja)
- Tipografia:
 - Tamanho do Texto:
 - Títulos: 20sp
 - Texto Principal: 18sp
 - Botões: 16sp
 - Fonte: Utilização da fonte padrão do sistema Android, com ajustes de peso e tamanho conforme necessário.
- Espaçamento e Margens:
 - Padding: 16dp ao redor dos elementos principais
 - Margens: 8dp a 16dp entre elementos para garantir uma visualização limpa e organizada.
- 2.2 Padrões de Desenvolvimento
- Estrutura do Código:
 - Organização: Separação clara entre as camadas de apresentação e de dados. Utilização de métodos em classes de acesso a dados (AcessoBD) para operações de banco de dados.
 - Nomenclatura: Utilização de nomes de variáveis e métodos descritivos que refletem claramente suas funções, como dataInicio, dataFim, consultar, etc.
- Componentes e Layouts:
 - Uso de XML: Layouts definidos em XML para facilitar a reutilização e manutenção. Elementos de UI (TextView, EditText, Button) são definidos com IDs claros.
 - Consistência: Utilização de estilos e temas definidos em styles.xml para garantir

uma aparência consistente em todo o aplicativo.

- Tratamento de Exceções:
 - Erro de Rede: Implementação de logs para capturar e analisar exceções de rede, proporcionando feedback útil para depuração e melhoria do código.
- Boas Práticas:
 - Responsividade: Layouts ajustados para diferentes tamanhos de tela e orientações, garantindo uma boa experiência de usuário em dispositivos variados.
 - Eficiência de Dados: Operações de banco de dados otimizadas para garantir que o aplicativo funcione de forma eficiente sem afetar a experiência do usuário.

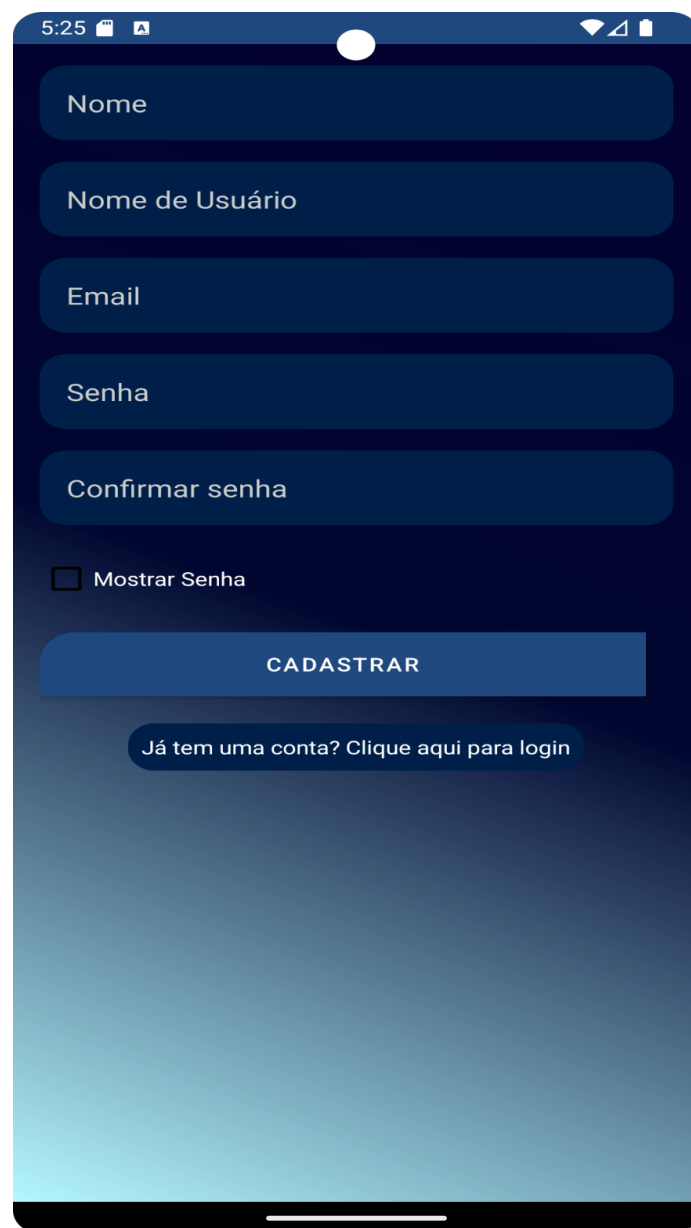
INTERFACES DESENVOLVIDAS

- ✓ **TELA INICIAL(LOGIN):** Quando o cliente baixar o aplicativo, para que seja possível acessar o sistema, ele precisará fazer um cadastro, clicando no botão: Cadastre-se. Caso o cliente já possua um cadastro vai inserir os dados solicitados: Usuário e senha e em seguida clicar no botão Login.



✓ **TELA DE CADASTRO:** O cliente precisará preencher todos os dados listados abaixo:

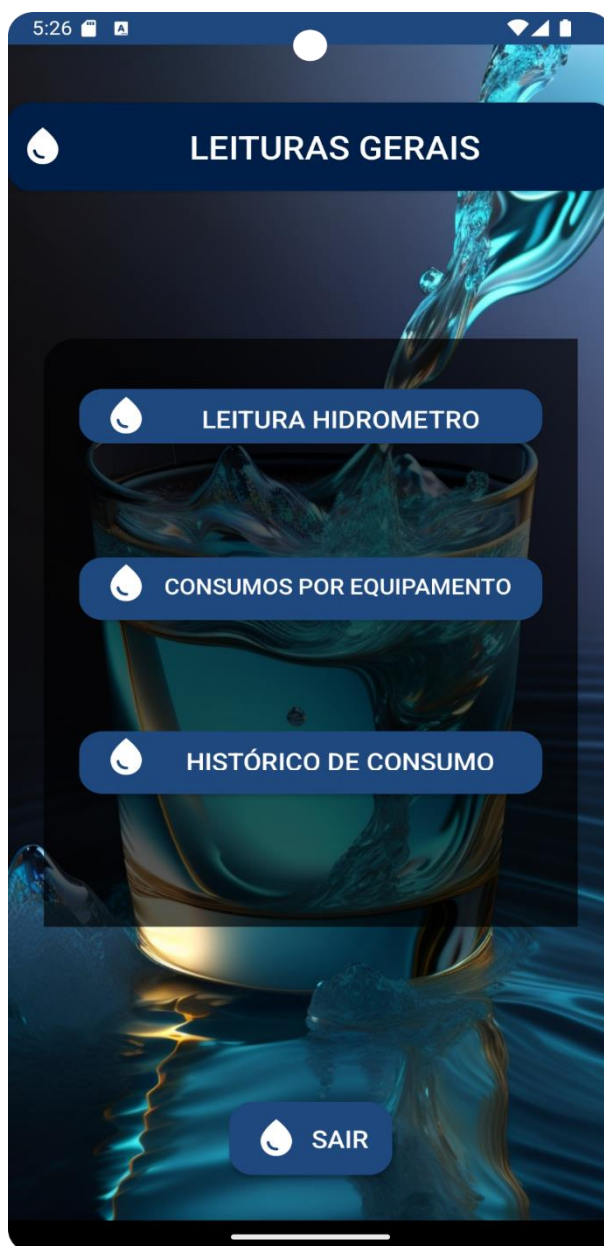
- Nome
- Nome de usuário
- E-mail
- Senha
- Confirmar senha e em seguida
- Clicar no botão Cadastrar.



The image shows a mobile application registration screen. At the top, a status bar displays the time 5:25 and various icons. The main interface consists of several input fields stacked vertically: 'Nome', 'Nome de Usuário', 'Email', 'Senha', and 'Confirmar senha'. Below these fields is a checkbox labeled 'Mostrar Senha'. A prominent blue button labeled 'CADASTRAR' is positioned below the checkbox. At the bottom, a link reads 'Já tem uma conta? Clique aqui para login'. The background features a dark blue gradient that transitions to a lighter blue at the bottom.

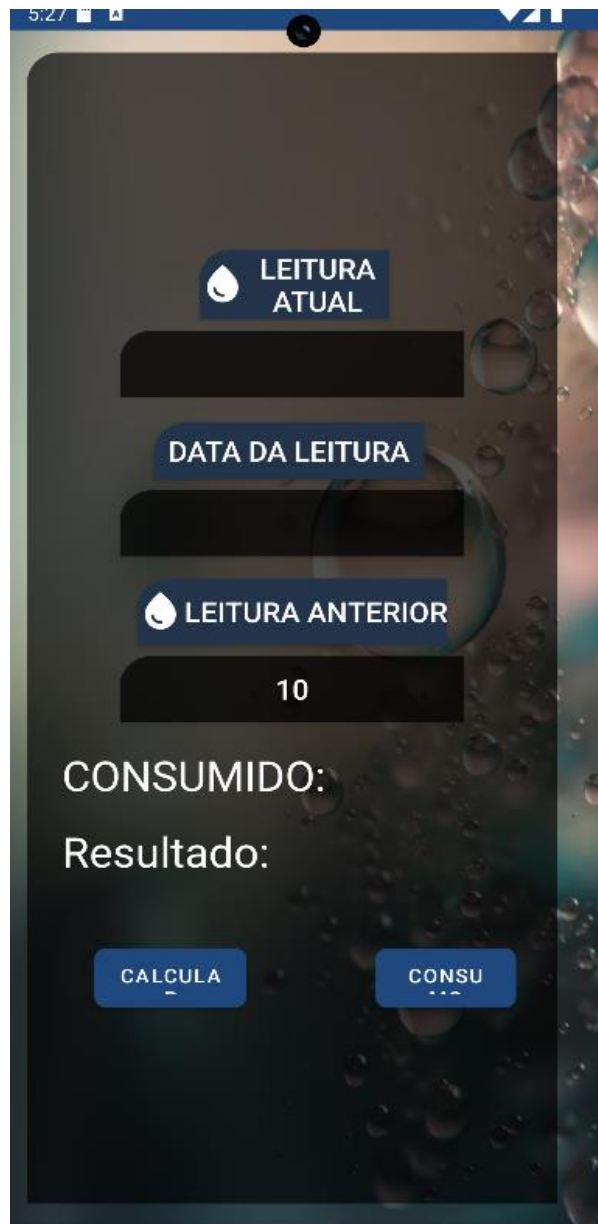
✓ **TELA PRINCIPAL** = MENU DE OPÇÕES: O cliente poderá escolher qual opção vai usar como:

- Leitura do hidrômetro: serve para calcular seu consumo em m³ e obter o valor em R\$
- Consumo por equipamento: com base na leitura informada serve para calcular quantos m³ e valores em reais cada equipamento consome quando utilizado e
- Histórico de consumo: um banco de dados que retorna as tabelas das interações ds botões anteriores.



✓ **TELA PRINCIPAL DA LEITURA DO HIDROMETRO, ATRAVÉS DA COLETA DE INFORMAÇÕES**

Esta tela é destinada ao usuário para que ele consiga estimar o consumo até o momento da leitura. O passo inicial é informar a leitura anterior que pode ser obtida na própria fatura de água, na sequência a leitura atual presente no hidrômetro e data que ele está utilizando.



The screenshot shows a mobile application interface with a dark background and a water droplet pattern. The interface includes the following elements:

- At the top, a status bar shows the time 5:27 and battery level.
- A label "LEITURA ATUAL" with a water droplet icon is positioned above a black input field.
- A label "DATA DA LEITURA" is positioned above another black input field.
- A label "LEITURA ANTERIOR" with a water droplet icon is positioned above a black input field containing the number "10".
- Below the input fields, the text "CONSUMIDO:" is displayed.
- Below "CONSUMIDO:", the text "Resultado:" is displayed.
- At the bottom, there are two blue buttons: "CALCULA" and "CONSU".

- ✓ **Tela de leitura de equipamentos:** O cliente antes de utilizar o equipamento precisa:
- Informar a Leitura Inicial
 - Nome do equipamento: exemplo – Máquina de lavar, chuveiro etc...
 - Após o uso do equipamento informar novamente a leitura presente no hidrômetro
 - Clicar em Adicionar Equipamento e por fim
 - Clicar no botão Calcular Consumo.

Leitura Inicial (ex: 35.3485 m³)

Leitura Final (ex: 35.3535 m³)

Nome do Equipamento

ADICIONAR EQUIPAMENTO

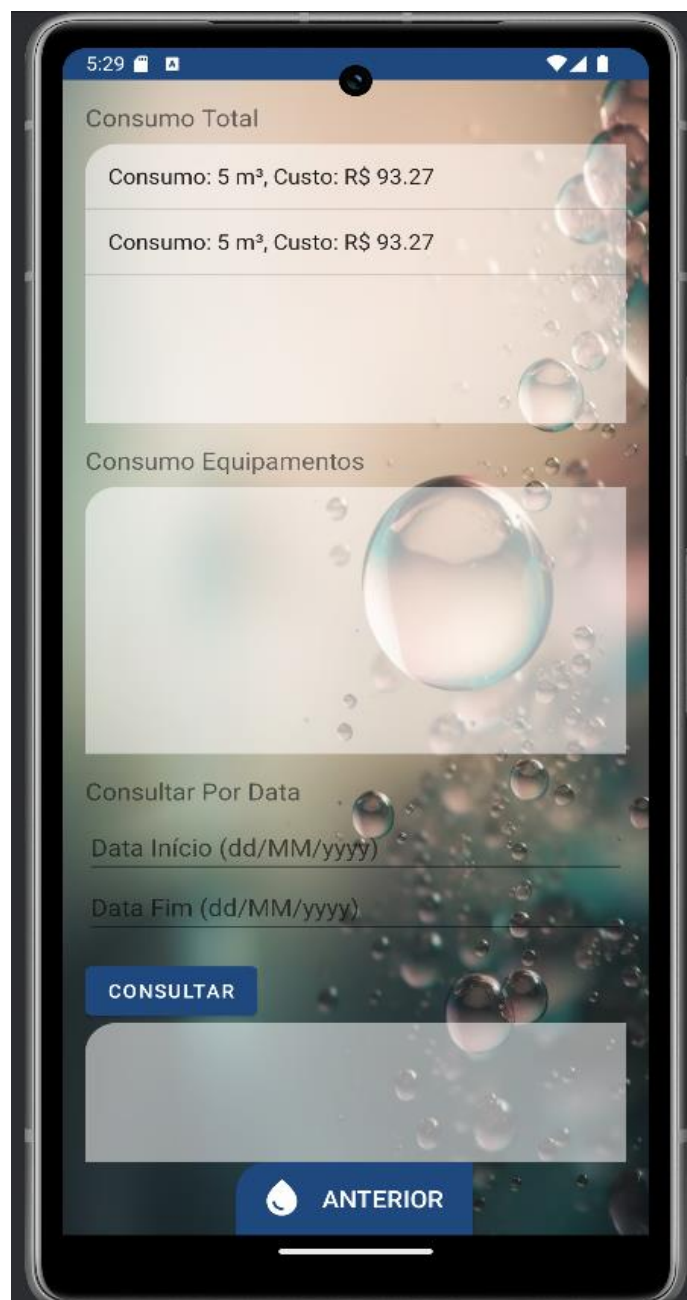
CALCULAR CONSUMO

Consumo e Custo:

✓ **Tela da lista dos consumos com valores em reais, data e por equipamento**


Banco de dados contendo todas as interações do usuário nas telas anteriores

- Consumo total vai exibir consumo em m³ e o custo em R\$ conforme leitura onformada pelo cliente;
- Consumo por equipamento vai exibir a quantidade de m³ utilizada e o valor em reais referente ao equipamento utilizado.



✓ **Entenda sua fatura**

Cada linha possui um número e ao clicar um pop-up exibirá informações sobre aquele item conforme imagem abaixo:



Endereço: Rua Engenheiros Rebouças nº 1376
CEP 80.215-900 - Curitiba - PR
CNPJ/MF 78.484.013/0001-45
Inscrição Estadual 101.80080-64
Internet: www.sanepar.com.br

1

NOME DO CLIENTE 2 MATRÍCULA 3

ENDEREÇO 4 NÚMERO 5 Nº LADO - Nº FRENTE

CEP 6 LOCAL 7

ROTEIRO DE LEITURA 8 HIDRÔMETRO 9 CAT - RES - COM - IND - UTP - POP 10

QUALIDADE DA ÁGUA DISTRIBUÍDA		Turbidez	Cor	Cloro	Floror	Col. Totais	Definições no verso
Nº Mínimo de Amostras Exigidas							
Nº Amostras Realizadas							

11


Nº Amostras que Atenderam à Legislação

Conclusão

HISTÓRICO DE PAGAMENTOS - CONDICIONADO AS OBSERVAÇÕES CONSTANTES NO VERSO

Ano	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Sep	Out	Nov	Dez
12												

13



HISTÓRICO DE CONSUMO m³

DIAS DE CONSUMO	DATA LEITURA	LEITURA ANTERIOR	LEITURA ATUAL	CONSUMO m³	REFERÊNCIA
14	15	16	17	18	19

MOTIVO DA AUSÊNCIA DE LEITURA 21

MÉDIA DE CONSUMO m³ ÚLTIMOS 6 MESES 22

VENCIMENTO 23

PREVISÃO PRÓXIMA LEITURA 24

ÁGUA 25

ESGOJO 26


SERVIÇOS 27

TOTAL 28

29

AUTENTICAÇÃO NO VERSO OBSERVAÇÕES NO VERSO COMPROVANTE CLIENTE

30




MATRÍCULA REFERÊNCIA VENCIMENTO VALOR TOTAL

AUTENTÍ COMPRO

HIDRÔMETRO: código que identifica o número do hidrômetro da ligação de s...

31



✓ **Sabe ler seu hidrômetro?**

Nesta tela o cliente vai encontrar informações de como proceder para inserir as informações no exigidas pelo aplicativo.



4 – Considerações Finais

O desenvolvimento do aplicativo SmartFlow trouxe resultados significativos para a gestão do consumo de água pelos clientes da Sanepar. Entre as principais contribuições, destacam-se:

- **Facilitação no Acompanhamento do Consumo:** O SmartFlow permite que os usuários acompanhem seu consumo de água de forma prática e eficiente. Com a capacidade de registrar e consultar leituras de hidrômetros, o aplicativo oferece uma visão clara do uso de água, ajudando os usuários a monitorar e gerenciar melhor seu consumo.
- **Redução da Desinformação:** Ao fornecer dados atualizados e precisos sobre o consumo de água e custos associados, o SmartFlow reduz a desinformação. Os usuários têm acesso a informações detalhadas e confiáveis, o que contribui para uma melhor compreensão das suas despesas com água e incentiva uma gestão mais consciente.
- **Promoção de Práticas Sustentáveis:** Com funcionalidades que permitem o monitoramento e cálculo de consumo e custos, o SmartFlow incentiva práticas de consumo sustentável. A visualização clara dos dados ajuda os usuários a identificar padrões e ajustar seus hábitos para economizar água e reduzir custos.

Limitações Identificadas

Apesar dos resultados positivos, algumas limitações foram observadas:

- **Integração de Autenticação:** A integração do sistema de autenticação ainda apresenta desafios. O método atual pode não atender completamente às necessidades de segurança e usabilidade, o que sugere a necessidade de revisão e aprimoramento.
- **Funcionalidades Adicionais:** Há uma necessidade de implementar funcionalidades adicionais que possam enriquecer a experiência do usuário. Por exemplo, a adição de relatórios detalhados e alertas personalizados poderia proporcionar um valor agregado significativo ao aplicativo.

Lições Aprendidas

Durante o desenvolvimento e teste do SmartFlow, foram aprendidas lições importantes:

- **Abordagem Prática para Resolução de Problemas:** A importância de adotar uma abordagem prática e iterativa na resolução de problemas foi uma lição chave. A prática de testes contínuos e ajustes baseados no feedback ajudou a melhorar a funcionalidade e a eficiência do aplicativo.
- **Atualização e Manutenção Contínuas:** O aplicativo deve ser continuamente atualizado e mantido para atender às demandas em evolução dos usuários. A implementação de

melhorias com base no feedback dos usuários é crucial para manter a relevância e a eficácia do aplicativo.

Trabalhos Futuros

Para aprimorar o SmartFlow e expandir suas funcionalidades, os próximos passos incluem:

- **Implementação de Telas Adicionais:** A inclusão de novas telas e funcionalidades que atendam às necessidades emergentes dos usuários é uma prioridade. Isso pode incluir painéis de controle mais detalhados e opções avançadas de relatórios.
- **Substituição do Sistema de Autenticação:** A revisão e substituição do sistema de autenticação são essenciais para garantir maior segurança e uma experiência de login mais intuitiva para os usuários.
- **Melhoria da Interface do Usuário:** Baseando-se no feedback recebido, será realizada uma contínua melhoria da interface do usuário. A otimização da experiência visual e funcional ajudará a tornar o aplicativo mais acessível e agradável de usar.

Referências bibliográficas

1. ORACLE. *Oracle Java Documentation*. Disponível em: <https://docs.oracle.com/en/java/>. Acesso em: 18/08/2024.
2. SANEPAR. *Simulador de Conta*. Disponível em: <https://atvn.sanepar.com.br/simuladorconta>. Acesso em: 18/08/2024.
3. SANEPAR. *Nossas Tarifas*. Disponível em: <https://site.sanepar.com.br/clientes/nossas-tarifas>. Acesso em: 18/08/2024.
4. SANEPAR. *Reclamações - Cobranças Indevidas*. Disponível em: <https://www.reclameaqui.com.br/empresa/sanepar-pr/lista-reclamacoes/?problema=0000000000000075>. Acesso em: 18/08/2024.
5. SILVA, Gabriel Panema da; AKAMINE, Cristiano (Orientador). *Medidor de Consumo de Água e Energia Residencial com Monitoramento Remoto*.
6. SANTOS, Pedro E.; SILVA, Roubert Edgar S.; CARVALHO JR, Dalmy F.; RIVERO, Ilo A. S.; MINI, Raquel A. F. *Controle de Consumo de Água Baseado em Reconhecimento Facial*.
7. W3SCHOOLS. *W3Schools Java Tutorial*. Disponível em: <https://www.w3schools.com/java/>. Acesso em: 18/08/2024.
8. ANDROID. *Android Studio*. Disponível em: <https://developer.android.com/studio?hl=pt-br>. Acesso em: 18/08/2024.