

|  |   |
|--|---|
| Descrição.....   | 2 |
| Público-Alvo/Intended User.....  | 2 |
| Funcionalidades/Features.....  | 2 |
| Protótipo de Interfaces do Usuário.....  | 3 |
| Tela 1.....  | 3 |
| Tela 2.....  | 4 |
| Tela 3.....  | 5 |
| Tela 4.....  | 6 |
| Considerações Chave/Key Considerations.....  | 6 |
| Como seu app vai tratar a persistência de dados?.....  | 6 |
| Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX)..... | 6 |
| Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.....           | 7 |
| Descreva como você implementará o Google Play Services.....                                  | 7 |
| Próximos Passos: Tarefas Necessárias.....  | 7 |
| Tarefa 1: Configuração do Projeto/Project Setup.....   | 7 |
| Tarefa 2: Implementar a Interface de Usuário (UI) para cada Activity e Fragment.....         | 8 |
| Tarefa 3: Criar as classes modelos.....  | 8 |
| Tarefa 4: Configurar o acesso ao Firebase.....   | 8 |
| Tarefa 5: Implementar chamadas assíncronas.....  | 8 |
| Tarefa 6: Criar Widget.....  | 8 |
| Tarefa 7: Design Responsivo.....   | 8 |
| Tarefa 8: Implementar testes.....  | 8 |
| Tarefa 9: Últimas conferências.....  | 9 |

**Usuário do GitHub:** arturschaefer

# Saúde 24 Horas

## Descrição

O **Saúde 24 Horas** é uma maneira simples e centralizada de buscar informações sobre atendimentos de saúde. Exibe uma lista de farmácias e hospitais na região desejada, horários de atendimento e canais para contato. Em uma tela especial é possível verificar promoções nas farmácias parceiras.

O que buscamos atingir com o aplicativo é manter um canal simples e direto em que o usuário possa localizar informações relevantes sobre saúde em sua região.

## Público-Alvo/Intended User

O público alvo é constituído por pessoas que necessitam de serviços de saúde em farmácias e hospitais.

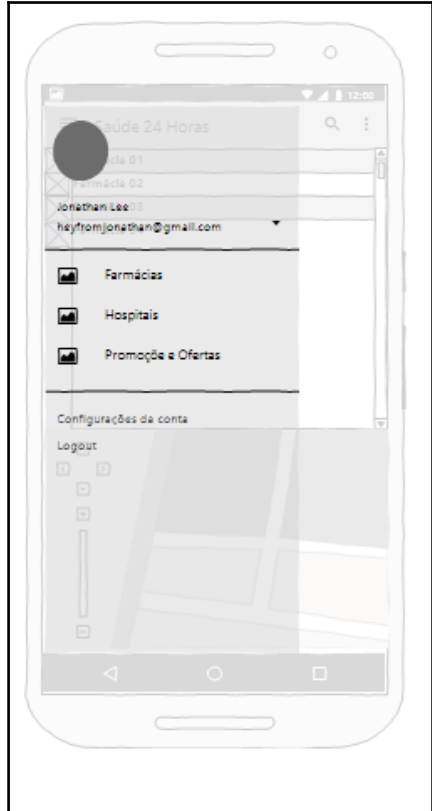
## Funcionalidades/Features

As principais funcionalidades do app são:

- Exibir a lista de farmácias e hospitais, assim como informações de contato e informações adicionais sobre os mesmos
- Listar farmácias e hospitais com atendimento no momento desejado
- Utilizar o Google Maps para localizar farmácias e hospitais, com base na localização do usuário ou alguma outra desejada
- Buscar as informações via Firebase
- Será possível salvar uma lista com as farmácias e hospitais favoritos
- Exibir uma lista de ofertas nas farmácias da região

## Protótipo de Interfaces do Usuário

### Tela 1

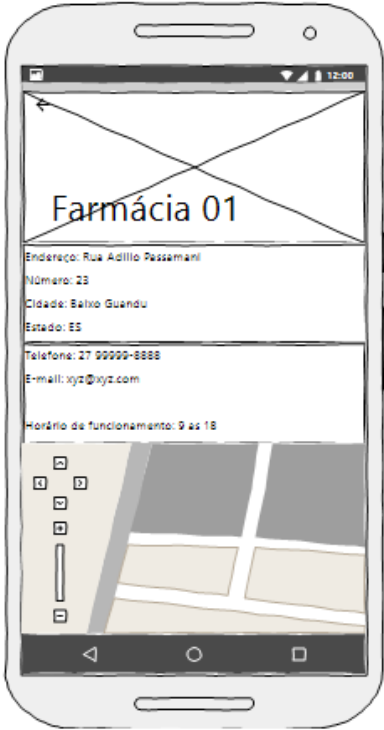


Tela inicial com o menu de Navegação lateral em amostras, sobrepondo a tela inicial.

## Tela 2



### Tela 3

|  |  |
|--|--|
|  | <p>Tela de detalhe da farmácia ou hospital escolhido. Exibindo as informações de contato, endereço e um mapa com a localização da mesma.</p> |
|--|--|

## Tela 4



## Considerações Chave/Key Considerations

### Como seu app vai tratar a persistência de dados?

Usarei o Firebase Database habilitando a persistência de dados Off-line.

Será possível salvar uma lista de estabelecimentos favoritos.

### Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

A persistência de dados será feita através do Firebase Database e os dados salvos online e localmente, para usuários com contas, e somente internamente para usuários anônimos.

O aplicativo lidará com logins e senhas errados de duas formas: a primeira é informar que a troca de senha terá que ser feita no serviço de autenticação escolhido (seja Google ou Facebook, por exemplo); a segunda será a de oferecer uma opção de login anônimo.

Caso seja encontrado algum erro ao tentar obter a localização, será informado ao usuário via AlertDialog, que houve um erro no GPS e que será necessário inserir a localização atual manualmente.

Em caso de erro de conexão de rede a tela principal é substituída por um fragment com uma mensagem de erro de conexão e um botão para tentar conectar novamente.

Ao rotacionar a tela ou ao executar o aplicativo em um aparelho de tela diferente, será feito um layout que responda as necessidades da nova configuração. Para casos de tablets, será utilizado o Master Detail Flow.

### **Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.**

- Picasso: para carregamento de imagens
- Firebase Database: persistência de dados e armazenamento online
- Firebase Auth: para login no Firebase através de contas como Google ou Facebook
- Butterknife: tratar das anotações e realizar a viewbinding
- Retrofit 2: para buscar informações da Web
- Gson: para converter objetos JSON
- Timber: para um propósitos de logging

### **Descreva como você implementará o Google Play Services.**

Utilizarei o Firebase Database, para permitir que o usuário busque dados remotos. Caso o usuário esteja conectado com sua conta no aplicativo, será permitido salvar estabelecimentos favoritos na nuvem, caso contrário será salvo localmente no aparelho.

Para permitir o acesso a contas será utilizado o Firebase Auth, permitindo que ele possa se conectar com contas do Google ou do Facebook, por exemplo.

## **Próximos Passos: Tarefas Necessárias**

Esta é a parte onde você falará sobre as principais funcionalidades do seu app (mencionadas acima) e as dividirá em tarefas técnicas tangíveis que você pode completar de forma incremental até finalizar o app.

### **Tarefa 1: Configuração do Projeto/Project Setup**

1. Configurar as bibliotecas necessárias
2. Popular o Firebase Database com informações relevantes
3. Configurar os serviços do Firebase no projeto do Android Studio

## **Tarefa 2: Implementar a Interface de Usuário (UI) para cada Activity e Fragment**

1. Construir a UI da MainActivity
2. Construir a UI para o Menu de Navegação
3. Construir a UI para a lista de detalhes dos estabelecimentos
4. Construir a UI para o Login

## **Tarefa 3: Criar as classes modelos**

1. Farmácia
2. Hospital
3. Usuário
4. Promoções
5. Item da promoção

## **Tarefa 4: Configurar o acesso ao Firebase**

Implementar a conexão do aplicativo com o Firebase Database e com a autenticação via Firebase Auth.

## **Tarefa 5: Implementar chamadas assíncronas**

Para casos de chamadas assíncronas que dependam de recursos externos, implementar AsyncTask.

## **Tarefa 6: Criar Widget**

Exibir uma lista de estabelecimentos favoritados pelo usuário.

## **Tarefa 7: Design Responsivo**

Configurar o aplicativo para ser responsivo e se adaptar em tablets.

## **Tarefa 8: Implementar testes**

Criar testes relevantes para as interfaces e chamadas assíncronas via requisições de serviços Web.



### **Tarefa 9: Últimas conferências**

Revisar arquivos de layout para eliminar todos os possíveis Hardcoded Strings ou conteúdos sem o `contentDescription` definidos. Buscando atender os requisitos de acessibilidade para o aplicativo.