Logo

# **DOCUMENTAÇÃO TÉCNICA**

# **APLICAÇÃO MOBILE**

### **Nome do projeto:**

# My Meals

### **Versão:**

2.0

### **Data de início / Prazo:**

18/08/21 - 8 meses

### **Detalhes técnicos:**

Aplicativo Android em React Native

Backend em Node.JS

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 18/08/2021 | 1.0 | Criação de Requisitos | Lucas Weslen |
| 06/04/2022 | 1.1 | Criação de novos requisitos | Gabriel Tavares,  Lucas Weslen |
| 07/04/2022 | 2.0 | Criação de novas regras de negócio | Gabriel Tavares,  Lucas Weslen |

# **REQUISITOS (\* Obrigatório, % Opcional)**

# **REQUISITOS FUNCIONAIS**:

**TELAS ONEBOARDING**

### **[RF001] Oneboarding 1:**

**Descrição:** Ao abrir o aplicativo pela primeira vez, após a tela de "***splash screen***" será demonstrado uma pequena introdução do produto em si, onde terá as seguintes informações:

* **Título:**.
* **Descrição:**.

### **[RF002] Oneboarding 2: (continuação)**

**Descrição:** Continuação para a segunda página a partir do movimento de swap a esquerda, será exibido as seguintes informações:

* **Título:**.
* **Descrição:**.

### **[RF003] Oneboarding 3: (final da apresentação)**

**Descrição:** Continuação para a terceira página a partir do movimento de swap a esquerda, será exibido as seguintes informações, em seguida, pressiona o botão “***começar agora***”:

* **Título:**.
* **Descrição:**.

**Saída:** Redirecionamento para a tela de “***login***”.

**TELAS DE AUTENTICAÇÃO**

### **[RF004] Cadastro de usuário:**

**Descrição:** O usuário preenche aos campos solicitados pelo sistema, em seguida, pressiona o botão de “**avançar”.**

**Processo:** Os dados do usuário serão resgatados e enviados para o banco de dados, após isso, o usuário terá acesso à sua área pessoal na aplicação mobile.

**Entrada:**

* **Dados pessoais:**
  + Nome\*;
  + E-mail\*;
  + Senha\*;
  + Confirmação de senha\*;

**Saída:** Diálogo de confirmação bem-sucedido do cadastro caso tenha sido efetuado com sucesso, se não, diálogo de erro.

**Pré-condições:** O usuário deve estar na tela de cadastro.

**Pós-condições:** Nenhuma.

### **[RF005] Autenticação social:**

**Descrição:** O usuário clica em botão de autenticação com o google, permite vincular com o google.

**Processo:** Os dados do usuário serão resgatados e enviados para o banco de dados, após isso, o usuário terá acesso à sua área pessoal na aplicação mobile.

**Entrada:**

* **Dados pessoais:**
  + E-mail\*;
  + Senha\*;

**Saída:** Diálogo de confirmação bem-sucedido do cadastro caso tenha sido efetuado com sucesso, se não, diálogo de erro.

**Pré-condições:** O usuário deve estar na tela de cadastro.

**Pré-condições:** O usuário deve ter uma conta do google.

**Pós-condições:** Nenhuma.

### **[RF006] Usuário aceita os termos de uso do sistema:**

**Descrição:** O usuário confirma com checkbox o termo de uso requisitado pelo sistema, em seguida, pressiona o botão de “***confirmar***”.

**Processo:** Nenhum.

**Entrada:** Checkbox\*.

**Saída:** Redirecionamento para tela “***tudo pronto***” da aplicação com uma mensagem de que o cadastro do mesmo foi efetuado com sucesso caso o termo tenha sido aceito, se não, o usuário será forçado a continuar na tela de autenticação, fazendo com que o mesmo saia da aplicação.

**Pré-condições:** O usuário deve preencher todos os campos solicitados pelo cadastro anteriormente e ter pressionado o botão de “***avançar***”.

**Pós-condições:** As informações do usuário ficam habilitadas para serem utilizadas em ações restritas do sistema, como também informações não sensíveis como: nome poderá ser utilizadas em telas.

### **[RF007] Usuário solicita a redefinição de senha:**

**Descrição:** O usuário preenche o campo de entrada solicitado pelo sistema, em seguida, pressiona o botão de “***confimar***”, a partir disso será enviado um código ao email, esse código será utilizado na aplicação para a redefinição de senha.

**Processo:** Os dados do usuário serão enviados para o sistema e substituídos, após isso, o usuário terá acesso à sua área pessoal na aplicação de acordo com as novas credenciais.

**Entrada:** E-mail\*.

**Saída:** Redirecionamento para a tela “***redefinir senha código***” caso o e-mail descrito no campo seja válido, se não, diálogo de erro, sem seguida tela de redefinir com uma nova senha.

**Pré-condições:** O usuário deve estar cadastrado e estar na tela de redefinição de senha e possuir um e-mail válido cadastrado no sistema.

**Pós-condições:** O usuário só poderá ter acesso a sua conta apenas se utilizar as novas credenciais.

### **[RF008] Usuário faz login no sistema:**

**Descrição:** O usuário preenche os campos solicitados pelo sistema, em seguida, pressiona o botão de “***entrar***”.

**Processo:** O sistema verifica o cadastro do usuário a partir dos dados enviados, em seguida, utiliza-se um algoritmo de criptografia para verificar sua autenticidade.

**Entrada:** Email\*, senha\*.

**Saída:** Redirecionamento para tela “***home***” da aplicação caso tenha sido efetuado com sucesso, se não, diálogo de erro.

**Pré-condições:** O usuário deve estar cadastrado no sistema e estar na tela de login.

**Pós-condições:** As informações do usuário ficam habilitadas para serem utilizadas em ações restritas do sistema, como também poderão ser utilizadas em telas.

**TELA HOME**

### **[RF009] Usuário visualiza o painel de refeições diárias:**

**Descrição:** Ao entrar na aplicação o usuário visualiza uma tela com todas as refeições diárias referente ao dia em que está, além de poder navegar nos dias da semana.

**Processo:** Nenhum.

**Entrada:** Nenhuma.

**Saída:** Lista de refeições de acordo com o dia.

**Pré-condições:** O usuário deve estar cadastrado no sistema e estar na tela “***home***”.

**Pós-condições:** Nenhuma.

### **[RF010] Usuário visualiza detalhes de uma refeição:**

**Descrição:** Tela contendo todas as informações de uma refeição.

**Processo:** Quando selecionada uma refeição através da lista de refeições no feed, o usuário será redirecionado para esta tela.

**Entrada:** Nenhuma.

**Saída:**

* Tipo de refeição
* Nome da refeição
* Ingredientes

**Pré-condições:** O usuário deve estar cadastrado no sistema e estar na tela “***detalhes da refeição***”.

**Pós-condições:** Nenhuma.

### **[RF011] Usuário cadastra uma refeição:**

**Descrição:** Inicia-se o processo de cadastro da refeição do dia, de acordo com o tipo de refeição

**Processo:** Quando o usuário clica em uma refeição vazia, ele tem a opção de colocar os dados da refeição.

**Entrada:**

* Nome da refeição
* Ingredientes

**Saída:** Nenhuma.

**Pré-condições:** Estar na tela “***Home***”.

**Pós-condições:** Nenhuma.

### **[RF012] Usuário seleciona o dia da semana para editar a refeição:**

**Descrição:** Inicia-se o processo de montagem ou edição das refeições.

**Processo:** O usuário pode navegar nos dias da semana clicando no dia específico para visualizar suas refeições diárias.

**Entrada:** Nenhuma

**Saída:**

* Refeições de acordo com o dia

**Pré-condições:** Clicar no dia específico da semana.

**Pós-condições:** Nenhuma.

**TELA DE COMPRAS**

### **[RF013] Usuário cadastra uma lista de compras:**

**Descrição:** Inicia-se o processo de cadastro da lista de compras, de acordo com a dieta.

**Processo:** Quando o usuário clica no menu de lista de compras onde tem uma lista vazia e a opção de cadastrar uma lista de compras, em seguida exibe um modal para adicionar uma lista de compras.

**Entrada:**

* Nome da Lista de compras
* Categoria

**Saída:** Um card com a lista de compras.

**Pré-condições:** Estar na tela “***Compras***”.

**Pós-condições:** Nenhuma.

### **[RF014] Usuário cadastra uma categoria de lista de compras:**

**Descrição:** Inicia-se o processo de cadastro de uma nova categoria.

**Processo:** Quando o usuário clica em escolher categoria no momento de cadastrar uma nova lista de compras, ele tem a opção de criar sua própria categoria ou adicionar uma existente.

**Entrada:**

* Nome da categoria
* Ícone da categoria

**Saída:** Nenhuma.

**Pré-condições:** Estar no card de “***Criar Lista de Compras***”.

**Pós-condições:** Nenhuma.

### **[RF015] Usuário adiciona itens à lista de compras:**

**Descrição:** Inicia-se o processo de adicionar um item à lista de compras.

**Processo:** Quando o usuário clica em uma lista de compras ele é redirecionado para a tela de lista de compras, onde tem a opção de adicionar um novo item.

**Entrada:**

* Nome do item

**Saída:** Lista de itens.

**Pré-condições:** Estar na tela “***Lista de Compras***”.

**Pós-condições:** Nenhuma.

**[RF016] Usuário remove itens da lista de compras:**

**Descrição:** Usuário tem a opção de deletar um único item ou limpar a lista de compras, em ambas as opções os itens são removidos.

**Processo:** Usuário clica no botão da lixeira ou no botão de limpar lista.

**Entrada:**

* Nome do item

**Saída:** Nenhuma.

**Pré-condições:** Estar na tela “***Lista de Compras***”.

**Pós-condições:** Nenhuma.

**TELA DE SAÚDE CORPORAL**

### **[RF017] Usuário pode verificar a quantidade de água que deve tomar por dia:**

**Descrição:** O usuário vai poder verificar a quantidade de água que deve tomar por dia e receber uma notificação a cada 1 hora informando que deve tomar água, essas informações são salvas.

**Processo:** Clicar no menu no botão de saúde corporal, e em seguida no botão de verificar quantidade de água.

**Entrada:**

* Sexo
* Peso corporal

**Saída:** Retorna um modal com a quantidade de água que deve tomar por dia aproximadamente, além de um card contendo tais informações detalhadas.

**Pré-condições:** Nenhuma.

**Pós-condições:** Nenhuma.

**TELA DE CONFIGURAÇÕES**

### **[RF018] Usuário visualiza os horários de cada refeição:**

**Descrição:** O usuário pode visualizar o horário de início e de término de uma refeição.

**Processo:** Clicar no menu no botão de configurações

**Entrada:** Nenhuma

**Saída:** Redirecionamento para tela *“****Configurações****”*, caso tenha sido efetuado com sucesso, se não, diálogo de erro.

**Pré-condições:** Nenhuma.

**Pós-condições:** Nenhuma.

### **[RF019] Usuário edita as suas configurações de horário de refeições:**

**Descrição:** O usuário entra com o horário de início e fim de cada refeição, em seguida, pressiona o botão de “***atualizar***”.

**Processo:** Após a edição, assim que as informações forem pro banco de dados, substituirão os dados antigos pelas novas modificações escolhidas por ele.

**Entrada:**

* Horário de início;
* Horário de fim

**Saída:** Nenhuma.

**Pré-condições:** Estar na tela de “***configurações***”.

**Pós-condições:** O usuário só poderá ter acesso a conta apenas se utilizar as novas credenciais.

**SIDEVIEW**

### **[RF020] Usuário edita as suas informações pessoais:**

**Descrição:** O usuário entra com o campo onde ele deseja modificar, em seguida, pressiona o botão de “***confirmar***”.

**Processo:** Após a edição, assim que as informações forem pro banco de dados, substituirão os dados antigos pelas novas modificações escolhidas por ele.

**Entrada:**

* Nome completo%;
* Telefone%;
* Email%;

**Saída:** Nenhuma.

**Pré-condições:** Estar na tela de “***perfil***”.

**Pós-condições:** O usuário só poderá ter acesso a conta apenas se utilizar as novas credenciais.

### **[RF021] Usuário verifica os termos de uso:**

**Descrição:** O cliente pressiona sobre o botão “***termos de serviço***”, em seguida, será redirecionado para a tela de “***termos de serviço***”, contendo descrição dos termos.

**Processo:** Nenhum.

**Entrada:** Nenhuma.

**Saída:** Nenhuma.

**Pré-condições:** Estar na tela de “***termos de serviço***”.

**Pós-condições:** Nenhuma.

### **[RF022] Usuário faz o logout:**

**Descrição:** O cliente pressiona sobre o botão “***sair***”, onde será encerrada a sua sessão no sistema, fazendo com que seja necessária uma nova sessão caso o usuário queira logar novamente no aplicativo.

**Processo:** Nenhum.

**Entrada:** Nenhuma

**Saída:** Nenhuma.

**Pré-condições:** Estar na tela de “***sideview***”.

**Pós-condições:** Nenhuma.

# **REQUISITOS NÃO FUNCIONAIS**:

# **[RNF001] Log de operações do sistema:**

**Descrição:** Nenhuma.

**[RNF002] Implementação:**

**Descrição:** A API deve ser desenvolvida utilizando Node.JS.

**[RNF003] Implantação:**

**Descrição:** A plataforma deverá ser hospedada na Digital Ocean.

**[RNF004] Usabilidade:**

**Descrição:** A aplicação não terá um foco específico em

acessibilidade para usuários com deficiência auditiva e/ou visual, com daltonismo

ou afins.

**[RNF005] Segurança:**

**Descrição:** O sistema deve proporcionar as seguintes seguranças:

* **[RNF005.1] Segurança do usuário:** Todas as funcionalidades que fazem parte de um perfil específico devem ser acessadas por um usuário (cadastrado) de acordo com sua função, onde sua autenticidade é verificada por token. Os dados sigilosos são criptografados.
* **[RNF005.2] Segurança do sistema:** O sistema deve sempre se manter com a proteção do certificado SSL.

**[RNF006] Disponibilidade:**

**Descrição:** O sistema não deverá operar em modo offline, havendo a necessidade

de conexão com a internet. Deve permanecer disponível para os usuários 24/7 de

acordo com o servidor em nuvem contratado. (desconsiderando essa hipótese caso

os recursos do servidor sejam gratuitos).

**[RNF007] Interoperabilidade:**

**Descrição:** A API deve se comunicar apenas com um banco de dados, na qual

seria o PostgreSQL ou MySQL.

**Obs:** Relatórios de alta especificidade contemplam apenas o banco de produção.

**[RNF008] Portabilidade:**

**Descrição:** A aplicação atual é capaz de ser atualizada para as versões mais atuais e com total portabilidade para novos dispositivos em lançamentos, já os dispositivos mais antigos são necessários que tenha no mínimo Android 5+ e iOS 10+ de acordo com a versão mínima do Expo SDK.

**[RNF009] Ético:**

**Descrição:** O sistema não irá (E não deverá) expor dados privativos para qualquer

pessoa ou organização que não seja proprietária do mesmo (exceto por meios

judiciais).

**[RNF010] Legais:**

**Descrição:** O sistema deve proporcionar os seguintes critérios legais:

* **[RNF010.1] Política de privacidade:** O sistema deve respeitar as políticas de privacidade impostas.
* **[RNF010.2] Lei LGPD:** O sistema deve respeitar a nova lei LGPD (Lei Geral de Proteção de Dados) que assegura aos usuários de solicitar a qualquer momento a exclusão de seus dados por definitivo é também possível obter todas as informações de cunho pessoal.

**[RNF011] Entrega:**

**Descrição:** O cronograma deve ser atualizado com as novas entregas a cada

duas semanas dando a visão do deadline do projeto.

# **REGRA DE NEGÓCIOS**

***( ) = Requisitos funcionais, [ ] = Requisitos não funcionais***

**RN001 - (RF000):**.

# **PRODUTOS E PRAZOS**

# **ESPECIFICAÇÕES TÉCNICAS**

# **PROTÓTIPOS:**

**Aplicação mobile - cliente**: [clique aqui](https://github.com/gabrieltav/My-Meals)

* 1. **FRAMEWORKS E BIBLIOTECAS:**

**Aplicação mobile:** React Native.

**Backend**: Adonis.JS.

* 1. **REPOSITÓRIOS:**

**Backend**: Nest.JS: [clique aqui](https://github.com/gabrieltav/My-Meals)

**Aplicação mobile**: React Native: [clique aqui](https://docs.google.com/document/d/10LdnUkAGElsp2TcHH2-r0bsxl5yxZxSYQ8V9sAi0_KQ/edit)

* 1. **DETALHES:**
     1. **TECNOLOGIAS:**

**Backend**: Javascript, Nest.JS.

**Aplicação mobile**: Typescript, React Native.

* + 1. **ARQUITETURA:**

**Backend**: MVC e P.O.O.

* 1. **SUPORTE:**

**Aplicação mobile**: Android e IOS.

* 1. **BASE DE DADOS:**

PostgreSQL ou MySQL.