

## Desenho detalhado Sprint 4



Engenharia de Software Aplicada
Desenho detalhado – Sprint nº4 - Sprint Final

Turma : 3

Grupo nº1

nº202100299 - Rui Barroso

nº202100296 - Gonçalo Vieira

nº202100984 - Francisco Silva

nº201901953 - André Pauli

1			
2			

- SUMARIO EXECUTIVO
- INTRODUÇÃO
- DESENHO DETALHADO
  - Modulo Planos
    - Requisitos funcionais (implementados)
  - Planos
    - Diagrama de Classes de desenho detalhado do Modulo Planos
    - Diagramas de Processos de negócio referentes ao Modulo Planos
    - Interface com o utilizador referente ao Modulo Planos
  - Mockup Planos
  - Mockup Exercicios e Refeições
  - Modulo Estatísticas
    - Requisitos funcionais (implementados)
  - Estatísticas
    - Diagrama de Classes de desenho detalhado do Modulo Estatísticas
    - Diagramas de Processos de negócio referentes ao Modulo Estatísticas
    - Interface com o utilizador referente ao Modulo Estatísticas
  - Mockup Estatísticas
  - Testes
    - 3.3.1 Testes Unitários
      - Especificação dos Casos de testes:
      - Especificação dos Procedimentos:
      - Especificação dos Casos de testes:
      - Especificação dos Procedimentos:
    - Testes de Automação
    - Testes de Automação
    - Testes de Integração
      - Testes de Regressão
      - Testes de Integração com 3rdparty
    - Testes de Sistema
      - Testes de Funcionalidade
      - Testes:
      - Testes de Eficiência (Carga)
      - Compatibilidade
      - Capacidade de interação

- Testes de Confiabilidade
- Segurança
- Testes de Manutabilidade
- Testes de Flexibilidade
- Testes de Segurança Física (Safety)
- Testes de aceitação
- Testes de Usabilidade
- Resultados dos testes:
- Testes de Segurança
  - Testes de Integridade
  - Testes de Autenticidade
- Testes:
- Resultados:
- Manual de utilização
- Manual técnico

## SUMARIO EXECUTIVO

Este documento delinea o plano detalhado para o Sprint 4 do projeto EasyFitHub. Durante este sprint, nosso foco principal será o desenvolvimento dos módulos de Planos e Estatísticas, a integração do armazenamento de imagens usando Firebase Storage e a correção de funcionalidades nos módulos anteriores. O Sprint 4 é crucial para consolidar a infraestrutura do EasyFitHub, fornecendo aos utilizadores recursos essenciais para gerir planos de exercícios, acompanhar estatísticas e armazenar imagens de perfil. Além disso, abordaremos feedbacks anteriores e aprimoraremos a usabilidade geral do sistema.

## INTRODUÇÃO

O Sprint 4 marca uma etapa significativa no desenvolvimento do EasyFitHub, onde estaremos concentrados na implementação de novos recursos essenciais e na melhoria da funcionalidade existente. O módulo de Planos permitirá que os utilizadores criem, visualizem e gerenciem seus planos de exercícios e alimentação de forma eficiente. Juntamente com isso, o módulo de Estatísticas oferecerá aos utilizadores insights valiosos sobre seu progresso e desempenho, auxiliando-os na definição de metas e no acompanhamento de resultados.

Além disso, integraremos o Firebase Storage para fornecer aos utilizadores a capacidade de armazenar e gerenciar suas imagens de perfil de forma escalável. Isso não só melhorará a experiência do utilizadore, mas também garantirá a integridade e a disponibilidade dos dados.

Por fim, durante este sprint, revisaremos os módulos anteriores para corrigir quaisquer bugs ou problemas de usabilidade identificados, garantindo que o EasyFitHub continue a oferecer uma plataforma robusta e confiável para nossos utilizadores. Com esses esforços combinados, estamos confiantes de que o Sprint 4 será fundamental para impulsionar o EasyFitHub para o próximo nível de excelência e satisfação do cliente.

# DESENHO DETALHADO

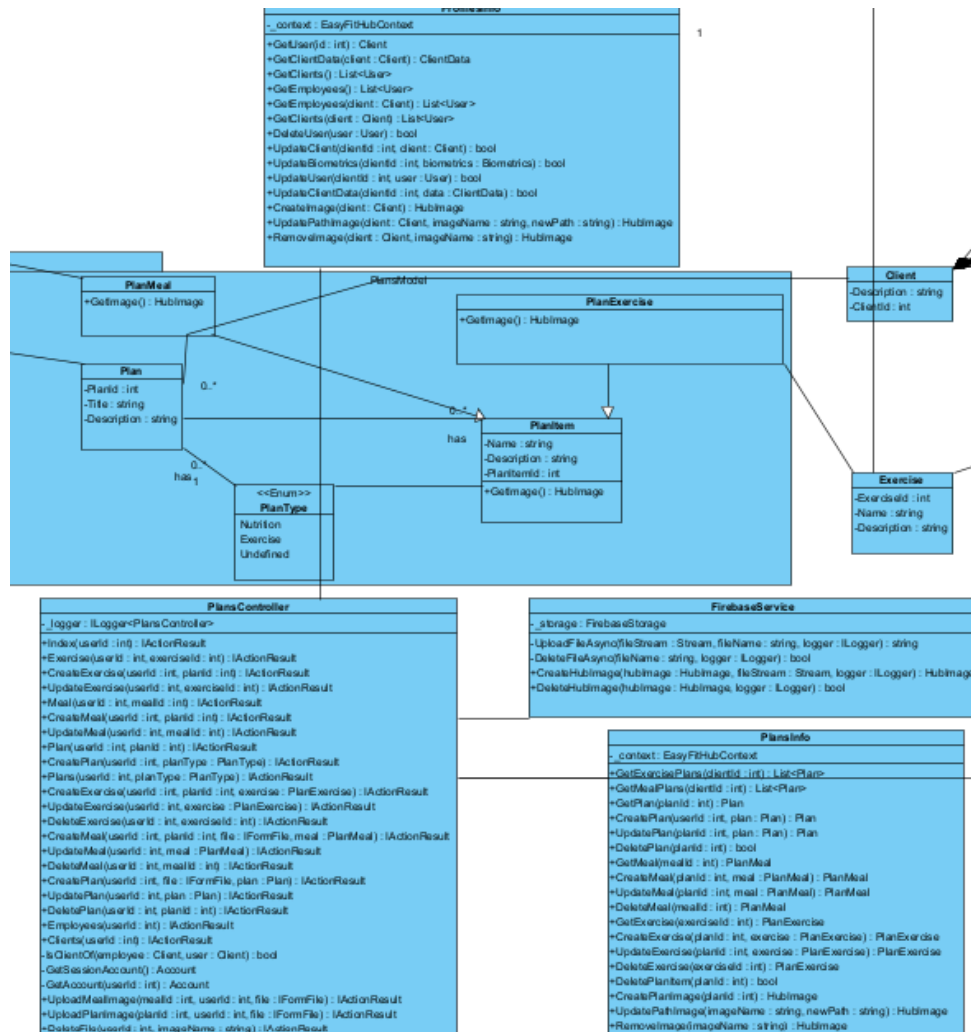
## Modulo Planos

### Requisitos funcionais (implementados)

#### Planos

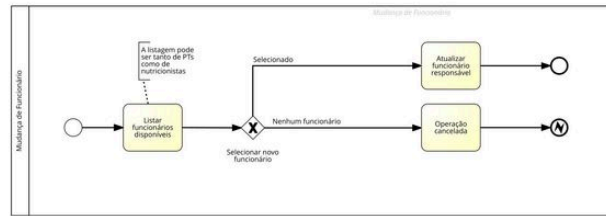
ID	Módulo	Descrição	Prioridade
R1	Planos	O sistema deverá realizar operações CRUD em planos de treino.	Alta
R2	Planos	O sistema deverá realizar operações CRUD em planos nutricionais.	Alta
R3	Planos	O sistema deverá realizar operações CRUD no catálogo de exercícios.	Baixa
R4	Planos	O sistema deverá permitir a mudança de funcionário.	Média
R5	Planos	O sistema deverá permitir a visualização do nutricionista apontado pelo ginásio.	Alta
R6	Planos	O sistema deverá permitir a visualização do PT apontado pelo ginásio.	Alta

### Diagrama de Classes de desenho detalhado do Modulo Planos

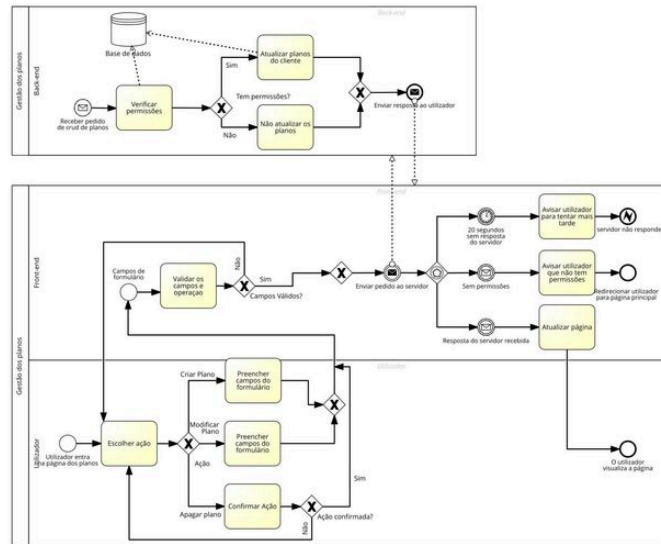


Diagramas de Processos de negócio referentes ao Modulo Planos

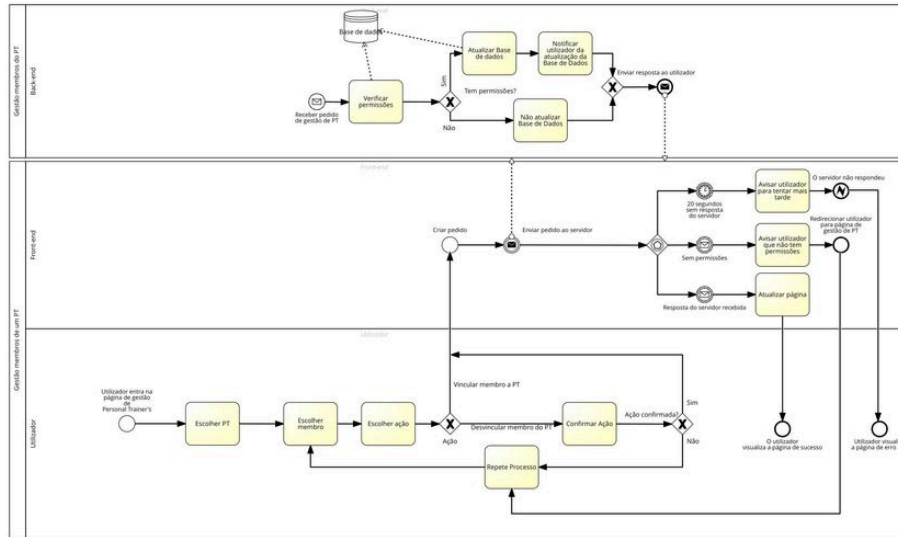
## Mudança de Personal Trainer



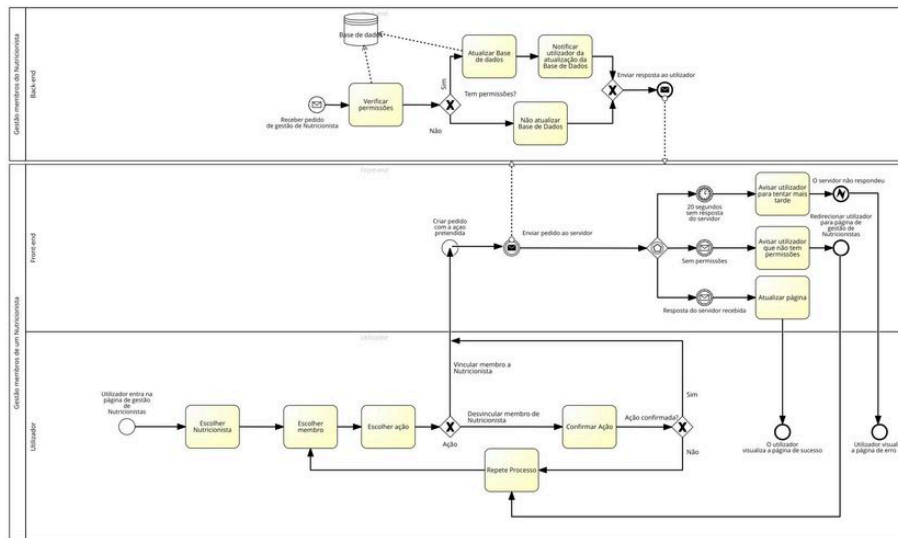
## Gestão dos planos



## CRUD membros de um PT

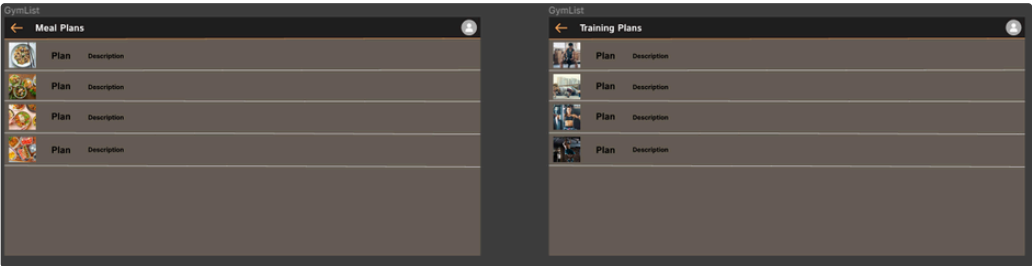
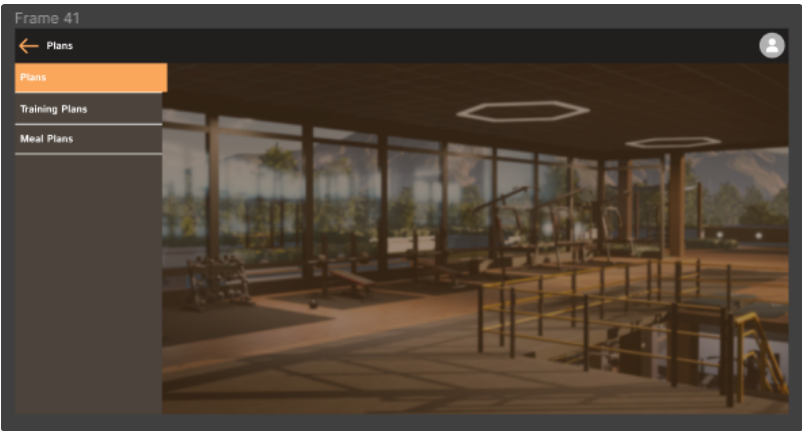


## CRUD membros de um nutricionista

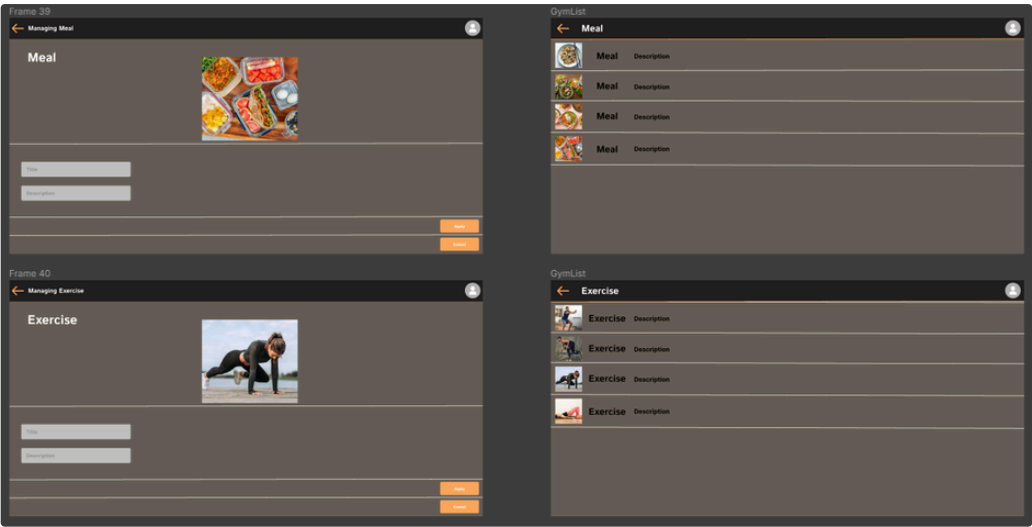


Interface com o utilizador referente ao Modulo Planos

Mockup Planos



Mockup Exercicios e Refeições



Modulo Estatísticas

Requisitos funcionais (implementados)

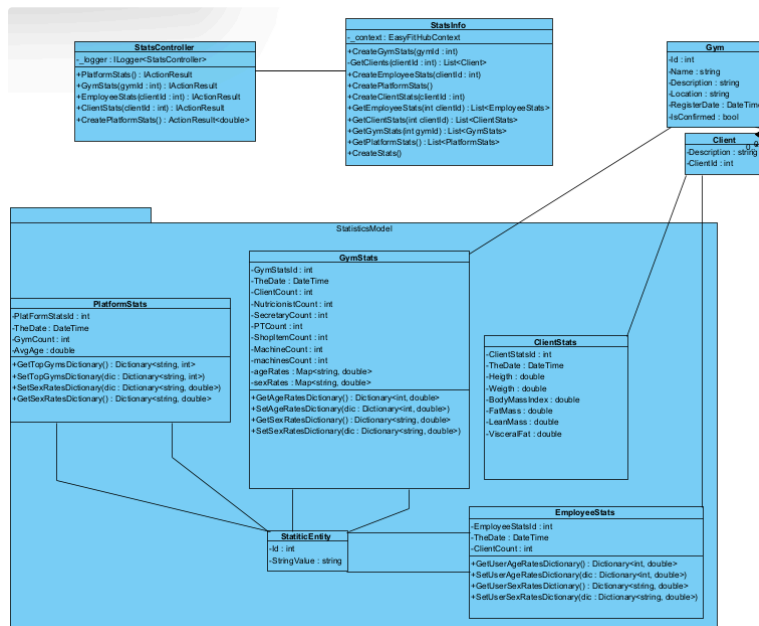
Estatísticas

ID	Módulo	Descrição	Prioridade
R1	Estatísticas	O sistema deverá fornecer as estatísticas da plataforma.	Baixa



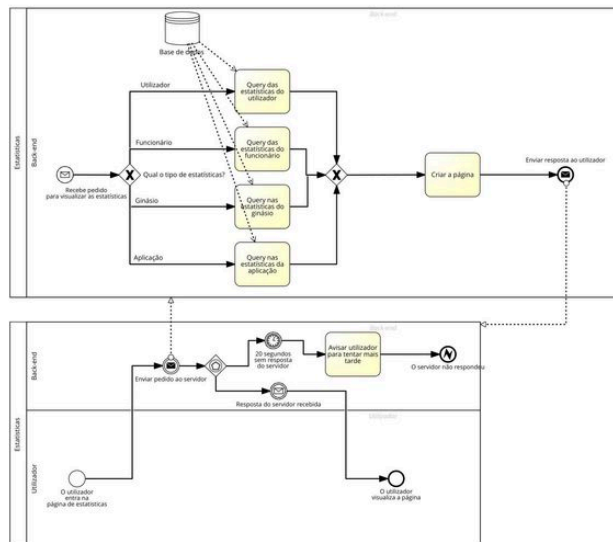
R2	Estatísticas	O sistema deverá fornecer o número de ginásios na plataforma	Baixa
R3	Estatísticas	O sistema deverá fornecer o número de utilizadores na plataforma	Baixa
R4	Estatísticas	O sistema deverá fornecer as idades médias dos utilizadores na plataforma	Baixa
R5	Estatísticas	O sistema deverá fornecer a percentagem dos gêneros na plataforma	Baixa
R6	Estatísticas	O sistema deverá fornecer os locais presentes na plataforma	Média
R7	Estatísticas	O sistema deverá fornecer as estatísticas do ginásio.	Alta
R8	Estatísticas	O sistema deverá fornecer os ginásios mais populares na plataforma	Baixa
R9	Estatísticas	O sistema deverá fornecer o número de clientes no ginásio	Baixa
R10	Estatísticas	O sistema deverá fornecer o número de Personal Trainers no ginásio	Média
R11	Estatísticas	O sistema deverá fornecer o número de nutricionistas no ginásio	Média
R12	Estatísticas	O sistema deverá fornecer as estatísticas do funcionário.	Média
R13	Estatísticas	O sistema deverá fornecer o número de recepcionistas no ginásio	Baixa
R14	Estatísticas	O sistema deverá fornecer o número de itens na loja do ginásio	Baixa
R15	Estatísticas	O sistema deverá fornecer o número de máquinas no ginásio	Média
R16	Estatísticas	O sistema deverá fornecer as estatísticas do cliente.	Média
R17	Estatísticas	O sistema deverá fornecer as percentagens das idades dos utilizadores no ginásio	Baixa
R18	Estatísticas	O sistema deverá fornecer as percentagem dos gêneros dos utilizadores no ginásio	Baixa
R19	Estatísticas	O sistema deverá fornecer as percentagens dos horários mais frequentados em média no ginásio	Alta
R20	Estatísticas	O sistema deverá fornecer as percentagens dos dias da semana mais frequentados em média no ginásio	Alta
R21	Estatísticas	O sistema deverá fornecer as máquinas favoritas do ginásio	Baixa
R22	Estatísticas	O sistema deverá fornecer o rating médio do ginásio	Média
R23	Estatísticas	O sistema deverá fornecer o número de clientes do funcionário	Baixa
R24	Estatísticas	O sistema deverá fornecer o rating médio do funcionário	Média
R25	Estatísticas	O sistema deverá fornecer as percentagens das idades dos utilizadores do funcionário	Baixa
R26	Estatísticas	O sistema deverá fornecer as percentagem dos gêneros dos utilizadores do funcionário	Baixa
R27	Estatísticas	O sistema deverá fornecer as aulas favoritas do cliente	Baixa

## Diagrama de Classes de desenho detalhado do Modulo Estatisticas



## Diagramas de Processos de negócio referentes ao Modulo Estatisticas

### Estatísticas



Mockup Estatisticas



## Testes

### 3.3.1 Testes Unitários

**Especificação dos Casos de testes:**

Nome Caso de teste:	GetPlatformStats
Código:	TC01
Finalidade:	Verificar se a função GetPlatformStats retorna as estatísticas corretas da plataforma.
Entradas:	N/A
Resultados esperados:	A função deve retornar as estatísticas gerais da plataforma, incluindo o número de ginásios, utilizadores, idade média dos utilizadores, percentagem de géneros, entre outros.
Dependências:	Existência de dados na plataforma.

Nome Caso de teste:	GetGymStats
Código:	TC02
Finalidade:	Verificar se a função GetGymStats retorna as estatísticas corretas de um ginásio específico.
Entradas:	ID do ginásio
Resultados esperados:	A função deve retornar as estatísticas específicas do ginásio, como o número de clientes, Personal Trainers, nutricionistas, máquinas, entre outros.
Dependências:	Existência de dados do ginásio na plataforma.

Nome Caso de teste:	GetClientStats
Código:	TC03
Finalidade:	Verificar se a função GetClientStats retorna as estatísticas corretas de um cliente específico.
Entradas:	ID do cliente
Resultados esperados:	A função deve retornar as estatísticas específicas do cliente, como a idade, gênero, aulas frequentadas, entre outros.
Dependências:	Existência de dados do cliente na plataforma.

Nome Caso de teste:	GetEmployeeStats
Código:	TC04
Finalidade:	Verificar se a função GetEmployeeStats retorna as estatísticas corretas de um funcionário específico.
Entradas:	ID do funcionário
Resultados esperados:	A função deve retornar as estatísticas específicas do funcionário, como o número de clientes, rating médio, entre outros.
Dependências:	Existência de dados do funcionário na plataforma.

#### Especificação dos Procedimentos:

Nome Caso de teste:	GetPlatformStats
Código:	TC01
Preparação:	Verificar se há dados suficientes na plataforma para gerar estatísticas significativas.
Inicialização:	Chamar a função GetPlatformStats do sistema.
Recursos específicos:	Base de dados

Nome Caso de teste:	GetGymStats
Código:	TC02
Preparação:	<ol style="list-style-type: none"> <li>1. Certificar-se de que o ginásio alvo está ativo e possui dados suficientes para gerar estatísticas.</li> <li>2. Obter o ID do ginásio para o qual deseja-se obter estatísticas.</li> </ol>
Inicialização:	Chamar a função GetGymStats do sistema, fornecendo o ID do ginásio como entrada.
Recursos específicos:	Base de dados

Nome Caso de teste:	GetClientStats
Código:	TC03
Preparação:	Certificar-se de que o cliente alvo está registrado e possui dados suficientes para gerar estatísticas. Obter o ID do cliente para o qual deseja-se obter estatísticas.
Inicialização:	Chamar a função GetClientStats do sistema, fornecendo o ID do cliente como entrada.
Recursos específicos:	Base de dados

Nome Caso de teste:	GetEmployeeStats
Código:	TC04
Preparação:	Certificar-se de que o funcionário alvo está registrado e possui dados suficientes para gerar estatísticas. Obter o ID do funcionário para o qual deseja-se obter estatísticas.
Inicialização:	Chamar a função GetEmployeeStats do sistema, fornecendo o ID do funcionário como entrada.
Recursos específicos:	Base de dados

#### Especificação dos Casos de testes:

Nome Caso de teste:	GetMealPlans
Código:	TC01
Finalidade:	Verificar se a função GetMealPlans retorna os planos de refeição corretos.
Entradas:	Id do Cliente
Resultados esperados:	A função deve retornar uma lista de todos os planos de refeição de um cliente.
Dependências:	Existência de planos de refeição atribuídos ao cliente.

Nome Caso de teste:	GetExercisePlans
Código:	TC02
Finalidade:	Verificar se a função GetExercisePlans retorna os planos de exercício corretos.
Entradas:	Id do Cliente
Resultados esperados:	A função deve retornar uma lista de todos os planos de exercícios de um cliente.
Dependências:	Existência de planos de exercícios atribuídos ao cliente..

Nome Caso de teste:	GetExercisePlan
---------------------	-----------------

Código:	TC03
Finalidade:	Verificar se a função GetExercisePlan retorna um plano de exercício específico corretamente.
Entradas:	ID do plano de exercício
Resultados esperados:	A função deve retornar os detalhes completos do plano de exercício correspondente ao ID fornecido.
Dependências:	Existência do plano de exercício na plataforma.

Nome Caso de teste:	GetMealPlan
Código:	TC04
Finalidade:	Verificar se a função GetMealPlan retorna um plano de refeição específico corretamente.
Entradas:	ID do plano de refeição
Resultados esperados:	A função deve retornar os detalhes completos do plano de refeição correspondente ao ID fornecido.
Dependências:	Existência do plano de refeição na plataforma.

Nome Caso de teste:	AddPlan
Código:	TC05
Finalidade:	Verificar se a função AddPlan adiciona um novo plano corretamente.
Entradas:	Detalhes do novo plano (exercício ou refeição)
Resultados esperados:	Após a execução da função, o novo plano deve ser adicionado com sucesso à plataforma.
Dependências:	Existência de um cliente para ser adicionado o plano.

Nome Caso de teste:	UpdatePlan
Código:	TC06
Finalidade:	Verificar se a função UpdatePlan atualiza um plano existente corretamente.
Entradas:	ID do plano a ser atualizado Novos detalhes do plano
Resultados esperados:	Após a execução da função, o plano correspondente ao ID fornecido deve ser atualizado com os novos detalhes.
Dependências:	Existência do plano a ser atualizado na plataforma.

Nome Caso de teste:	DeletePlan
Código:	TC07

Finalidade:	Verificar se a função <code>DeletePlan</code> exclui um plano existente corretamente.
Entradas:	ID do plano a ser removido
Resultados esperados:	Após a execução da função, o plano correspondente ao ID fornecido deve ser removido com sucesso da plataforma.
Dependências:	Existência do plano a ser removido da plataforma.

Nome Caso de teste:	GetPlanExercises
Código:	TC08
Finalidade:	Verificar se a função <code>GetPlanExercises</code> retorna os exercícios associados a um plano corretamente.
Entradas:	ID do plano de exercício
Resultados esperados:	A função deve retornar uma lista de todos os exercícios associados ao plano de exercício correspondente ao ID fornecido.
Dependências:	Existência do plano de exercício na plataforma.

Nome Caso de teste:	GetPlanMeals
Código:	TC09
Finalidade:	Verificar se a função <code>GetPlanMeals</code> retorna as refeições associadas a um plano corretamente.
Entradas:	ID do plano de refeições
Resultados esperados:	A função deve retornar uma lista de todas as refeições associadas ao plano de refeição correspondente ao ID fornecido.
Dependências:	Existência do plano de refeição na plataforma.

Nome Caso de teste:	GetPlanExercise
Código:	TC10
Finalidade:	Verificar se a função <code>GetPlanExercise</code> retorna um exercício específico de um plano corretamente.
Entradas:	ID do plano de exercício ID do exercício
Resultados esperados:	A função deve retornar o exercício correspondente ao ID fornecido dentro do plano de exercício especificado.
Dependências:	Existência do plano de exercício na plataforma. Existência do exercício dentro do plano de exercício.

Nome Caso de teste:	GetPlanMeal
---------------------	-------------



Código:	TC11
Finalidade:	Verificar se a função <code>GetPlanMeal</code> retorna uma refeição específica de um plano corretamente.
Entradas:	ID do plano de refeição ID da refeição
Resultados esperados:	A função deve retornar a refeição correspondente ao ID fornecido dentro do plano de refeição especificado.
Dependências:	Existência do plano de refeição na plataforma. Existência da refeição dentro do plano de refeição.

Nome Caso de teste:	AddPlanExercise
Código:	TC12
Finalidade:	Verificar se a função <code>AddPlanExercise</code> adiciona um novo exercício a um plano corretamente.
Entradas:	ID do plano de exercício Detalhes do novo exercício a ser adicionado
Resultados esperados:	Após a execução da função, o novo exercício fornecido deve ser adicionado ao plano de exercício correspondente.
Dependências:	Existência do plano de exercício na plataforma.

Nome Caso de teste:	AddPlanMeal
Código:	TC13
Finalidade:	Verificar se a função <code>AddPlanMeal</code> adiciona uma nova refeição a um plano corretamente.
Entradas:	ID do plano de refeição Detalhes da nova refeição a ser adicionada
Resultados esperados:	Após a execução da função, a nova refeição fornecida deve ser adicionada ao plano de refeição correspondente.
Dependências:	Existência do plano de refeição na plataforma.

Nome Caso de teste:	UpdatePlanExercise
Código:	TC14
Finalidade:	Verificar se a função <code>UpdatePlanExercise</code> atualiza corretamente os detalhes de um exercício em um plano.
Entradas:	ID do plano de exercício ID do exercício a ser atualizado

	Novos detalhes do exercício
Resultados esperados:	Após a execução da função, os detalhes do exercício correspondente no plano de exercício devem ser atualizados conforme fornecido.
Dependências:	Existência do plano de exercício na plataforma.  Existência do exercício dentro do plano de exercício.

Nome Caso de teste:	UpdatePlanMeal
Código:	TC15
Finalidade:	Verificar se a função <code>UpdatePlanMeal</code> atualiza corretamente os detalhes de uma refeição em um plano.
Entradas:	ID do plano de refeição  ID da refeição a ser atualizada  Novos detalhes da refeição
Resultados esperados:	Após a execução da função, os detalhes da refeição correspondente no plano de refeição devem ser atualizados conforme fornecido.
Dependências:	Existência do plano de refeição na plataforma.  Existência da refeição dentro do plano de refeição.

Nome Caso de teste:	DeletePlanItem
Código:	TC16
Finalidade:	Verificar se a função <code>DeletePlanItem</code> exclui corretamente um item (exercício ou refeição) de um plano.
Entradas:	ID do plano (exercício ou refeição)  ID do item a ser excluído
Resultados esperados:	Após a execução da função, o item correspondente (exercício ou refeição) deve ser removido do plano especificado.
Dependências:	Existência do plano na plataforma.  Existência do item dentro do plano.

#### Especificação dos Procedimentos:

Nome Caso de teste:	GetMealPlans
Código:	TC01
Preparação:	Certificar-se de que existem planos de refeição cadastrados na plataforma.
Inicialização:	Chamar a função <code>GetMealPlans</code> do sistema.

Recursos específicos:	Base de Dados
-----------------------	---------------

Nome Caso de teste:	GetExercisePlans
Código:	TC02
Preparação:	Certificar-se de que existem planos de exercício cadastrados na plataforma.
Inicialização:	Chamar a função <code>GetExercisePlans</code> do sistema.
Recursos específicos:	Base de Dados

Nome Caso de teste:	GetExercisePlan
Código:	TC03
Preparação:	Certificar-se de que existe um plano de exercício específico cadastrado na plataforma. Obter o ID do plano de exercício desejado.
Inicialização:	Chamar a função <code>GetExercisePlan</code> do sistema, fornecendo o ID do plano de exercício como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	GetMealPlan
Código:	TC04
Preparação:	Certificar-se de que existe um plano de refeição específico cadastrado na plataforma. Obter o ID do plano de refeição desejado.
Inicialização:	Chamar a função <code>GetMealPlan</code> do sistema, fornecendo o ID do plano de refeição como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	AddPlan
Código:	TC05
Preparação:	Preparar os dados necessários para criar um novo plano.
Inicialização:	Chamar a função <code>AddPlan</code> do sistema, fornecendo os dados do novo plano como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	UpdatePlan
Código:	TC06
Preparação:	Identificar um plano existente que será atualizado.

	Preparar os novos dados para o plano.
Inicialização:	Chamar a função <code>UpdatePlan</code> do sistema, fornecendo o ID do plano a ser atualizado e os novos dados como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	DeletePlan
Código:	TC07
Preparação:	Identificar um plano existente que será excluído.
Inicialização:	Chamar a função <code>DeletePlan</code> do sistema, fornecendo o ID do plano a ser excluído como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	GetPlanExercises
Código:	TC08
Preparação:	Certificar-se de que existe pelo menos um plano de exercício com exercícios cadastrados na plataforma.
Inicialização:	Chamar a função <code>GetPlanExercises</code> do sistema, fornecendo o ID do plano de exercício como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	GetPlanExercise
Código:	TC09
Preparação:	Certificar-se de que existe pelo menos um exercício associado a um plano de exercício na plataforma.
Inicialização:	Chamar a função <code>GetPlanExercise</code> do sistema, fornecendo o ID do exercício e o ID do plano de exercício como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	GetPlanMeals
Código:	TC10
Preparação:	Certificar-se de que existe pelo menos uma refeição associada a um plano de refeição na plataforma.
Inicialização:	Chamar a função <code>GetPlanMeals</code> do sistema, fornecendo o ID do plano de refeição como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	GetPlanMeal
---------------------	-------------

Código:	TC11
Preparação:	Certificar-se de que existe pelo menos uma refeição associada a um plano de refeição na plataforma.
Inicialização:	Chamar a função <code>GetPlanMeal</code> do sistema, fornecendo o ID da refeição e o ID do plano de refeição como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	AddPlanExercise
Código:	TC12
Preparação:	Certificar-se de que o plano de exercício e o exercício que se pretende adicionar estão devidamente cadastrados no sistema.
Inicialização:	Chamar a função <code>AddPlanExercise</code> do sistema, fornecendo o ID do plano de exercício e o ID do exercício a ser adicionado como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	UpdatePlanExercise
Código:	TC13
Preparação:	Certificar-se de que existe pelo menos um exercício associado a um plano de exercício na plataforma.
Inicialização:	Chamar a função <code>UpdatePlanExercise</code> do sistema, fornecendo o ID do plano de exercício, o ID do exercício a ser atualizado e os novos dados do exercício como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	AddPlanMeal
Código:	TC14
Preparação:	Certificar-se de que o plano de refeição e a refeição que se pretende adicionar estão devidamente cadastrados no sistema.
Inicialização:	Chamar a função <code>AddPlanMeal</code> do sistema, fornecendo o ID do plano de refeição e o ID da refeição a ser adicionada como entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	UpdatePlanMeal
Código:	TC15
Preparação:	Certificar-se de que existe pelo menos uma refeição associada a um plano de refeição na plataforma.
Inicialização:	Chamar a função <code>UpdatePlanMeal</code> do sistema, fornecendo o ID do plano de refeição, o ID da refeição a ser atualizada e os novos dados da refeição como

	entrada.
Recursos específicos:	Base de Dados

Nome Caso de teste:	DeletePlanItem
Código:	TC16
Preparação:	Certificar-se de que existe pelo menos um item (exercício ou refeição) associado a um plano (plano de exercício ou plano de refeição) na plataforma.
Inicialização:	Chamar a função <code>DeletePlanItem</code> do sistema, fornecendo o ID do plano (exercício ou refeição), o ID do item a ser removido e o tipo de item como entrada.
Recursos específicos:	Base de Dados

Resultados:

Os testes foram executados com sucesso.

✓ EasyFitHubTests (38)	26,1 s
✓ EasyFitHubTests (38)	26,1 s
✓ AuthenticationTest (2)	1,4 s
✓ FirebaseTest (2)	1,4 s
✓ GymTest (5)	576 ms
✓ InventoryInfoTest (15)	340 ms
✓ PlansTest (3)	390 ms
✓ ProfileInfoTest (2)	454 ms
✓ SearchInfoTest (4)	197 ms
✓ StripeTest (5)	21,3 s

## Testes de Automação

### Testes de Automação

O Katalon é uma ferramenta de automação de testes completa e poderosa, projetada para simplificar e acelerar o processo de testes de software. Oferece uma interface intuitiva e recursos abrangentes para ajudar equipas de desenvolvimento a criar, executar e gerir testes automatizados de forma eficiente.

Com o Katalon, iremos poder gravar facilmente casos de teste, criar scripts de teste automatizados, gerar relatórios detalhados...

Esta ferramenta é especialmente útil para o desenvolvimento de software que deseja garantir a qualidade e a estabilidade das suas aplicações, que é realmente o nosso objetivo. O Katalon ajuda a reduzir o tempo e os esforços necessários para testar aplicativos, que proporciona que a nossa equipa concentrem-se mais na inovação e entrega rápida de software de alta qualidade.



### Testes de Integração

(qual a estratégia adotada [big bag, top down, etc] e demonstrar)

### Testes de Regressão

(atenção: existem autores que colocam este teste como de sistema)

### Testes de Integração com 3rdparty

(integração com sistemas externos como paypal, google maps api, etc.)

### Testes de Sistema

(testar os RQ de acordo com a [ISO25010](#))

### Testes de Funcionalidade

Requisitos de Qualidade:

RQ1	Descrição
<b>Pior caso:</b>	0% de sucesso na execução das operações básicas.
<b>Planeado:</b>	Verificar 100% de sucesso na execução das operações básicas.
<b>Teste:</b>	Execute todas as funcionalidades básicas do sistema, verificando se cada uma opera conforme esperado. Isso inclui criar, editar e excluir elementos fundamentais para o funcionamento do sistema.
<b>Unidades:</b>	Taxa de sucesso na execução das operações básicas.

RQ2	Descrição
<b>Pior caso:</b>	Garantir 95% de sucesso na execução de operações avançadas.
<b>Planeado:</b>	40% de sucesso na execução de operações avançadas.
<b>Teste:</b>	Explore funcionalidades mais avançadas e específicas do sistema, garantindo que todas respondam corretamente a diferentes entradas e cenários.
<b>Unidades:</b>	Taxa de sucesso na execução de operações avançadas.

RQ5	Descrição
<b>Pior caso:</b>	Alcançar uma taxa de sucesso na personalização nos planos de treino de 50%.
<b>Planeado:</b>	Garantir uma taxa de sucesso na personalização nos planos de treino de 85%.
<b>Teste:</b>	Personalize planos de treino e nutrição, verificando se o sistema adapta as recomendações de acordo com as preferências do usuário.
<b>Unidades:</b>	Taxa de sucesso na personalização nos planos de treino.

RQ6	Descrição
<b>Pior caso:</b>	Obter uma taxa de satisfação nas recomendações baseadas no histórico de 60%.

<b>Planeado:</b>	Alcançar uma taxa de satisfação nas recomendações baseadas no histórico de 92%.
<b>Teste:</b>	Explore funcionalidades que se baseiam no histórico do usuário, avaliando se o sistema utiliza essas informações de maneira eficaz.
<b>Unidades:</b>	Taxa de satisfação nas recomendações baseadas no histórico.

#### Testes:

Nome Caso de teste:	Teste das Funcionalidades básicas
Código:	TC01
Preparação:	Preparar testes unitários de CRUD para Planos, exercicios e refeições.
Inicialização:	Executar os testes unitários.
Recursos específicos:	Uma base de dados.

Nome Caso de teste:	Teste das Funcionalidades avançadas
Código:	TC02
Preparação:	Preparar testes unitários de que verifiquem as funcionalidades da integração com o API do Firebase
Inicialização:	Executar os testes unitários.
Recursos específicos:	Ligação com o Api do Firebase

#### Resultados:

Os testes forma executados com sucesso.

✓ EasyFitHubTests (38)	26,1 s
✓ EasyFitHubTests (38)	26,1 s
✓ AuthenticationTest (2)	1,4 s
✓ FirebaseTest (2)	1,4 s
✓ GymTest (5)	576 ms
✓ InventoryInfoTest (15)	340 ms
✓ PlansTest (3)	390 ms
✓ ProfileInfoTest (2)	454 ms
✓ SearchInfoTest (4)	197 ms
✓ StripeTest (5)	21,3 s

#### Testes de Eficiência (Carga)

RQ7	Descrição
<b>Pior caso:</b>	Exceder o tempo máximo de resposta para login (mais de 10 segundos).
<b>Planeado:</b>	Manter um tempo médio de resposta para agendamento entre 1 e 10 segundos.
<b>Teste:</b>	Autentique-se numa conta do sistema e meça o tempo que leva para a operação ser concluída. Verifique se está dentro da faixa especificada de 1 a 10 segundos.



<b>Unidades:</b>	Tempo médio de resposta para login.
------------------	-------------------------------------

RQ8	Descrição
<b>Pior caso:</b>	Exceder uma latência na atualização de estatísticas de 20 segundos.
<b>Planeado:</b>	Alcançar uma latência na atualização de estatísticas inferior a 3 segundos.
<b>Teste:</b>	Atualize estatísticas no sistema e verifique se os dados mais recentes são exibidos imediatamente. Isso garante que o sistema mantenha informações em tempo real.
<b>Unidades:</b>	Latência na atualização de estatísticas.

RQ9	Descrição
<b>Pior caso:</b>	Apresentar uma utilização ineficiente de CPU e memória durante operações intensivas.
<b>Planeado:</b>	Manter o uso eficiente de CPU e memória durante operações intensivas.
<b>Teste:</b>	Execute operações intensivas no sistema, como processamento de grandes conjuntos de dados. Meça a utilização de recursos, garantindo otimização durante essas operações.
<b>Unidades:</b>	Uso eficiente de CPU e memória.

RQ10	Descrição
<b>Pior caso:</b>	Apresentar uma utilização ineficiente de CPU e memória durante consultas frequentes.
<b>Planeado:</b>	Garantir uma utilização eficiente de CPU e memória durante consultas frequentes.
<b>Teste:</b>	Realize uma grande quantidade de consultas aos dados do sistema e meça a utilização de recursos, garantindo otimização durante essas operações.
<b>Unidades:</b>	Uso eficiente de CPU e memória.

RQ11	Descrição
<b>Pior caso:</b>	Verificar uma degradação significativa no desempenho ao cadastrar o número máximo de ginásios.
<b>Planeado:</b>	Avaliar o desempenho ao cadastrar o número máximo de ginásios suportado pelo sistema, buscando manter um desempenho consistente.
<b>Teste:</b>	Inscriva o número máximo de ginásios suportado pelo sistema e avalie seu desempenho.
<b>Unidades:</b>	Desempenho com o número máximo de ginásios.

RQ12	Descrição
<b>Pior caso:</b>	Identificar falhas ou lentidão expressiva nas operações com a carga de 200 ginásios.
<b>Planeado:</b>	Garantir uma experiência consistente mesmo sob carga significativa, realizando operações simultâneas em 200 ginásios.
<b>Teste:</b>	Realize operações simultâneas em 200 ginásios e avalie a fluidez dos serviços.

<b>Unidades:</b>	Desempenho durante carga intensiva.
------------------	-------------------------------------

Nome Caso de teste:	Medir o tempo médio de resposta
Código:	TC01
Preparação:	Certificar-se de que o sistema está configurado com uma carga de trabalho representativa. Garantir que existem contas registradas na plataforma.
Inicialização:	Realizar uma série de solicitações de login simultâneas. Medir o tempo de resposta médio para cada solicitação de login .
Recursos específicos:	Sistema configurado com uma carga de trabalho representativa. Contas registradas na plataforma. Ferramentas de medição de tempo.

Nome Caso de teste:	Avaliar a latência na atualização de estatísticas
Código:	TC02
Preparação:	Certificar-se de que o sistema está configurado com uma carga de trabalho representativa. Garantir que as estatísticas em tempo real estão ativadas e sendo atualizadas regularmente.
Inicialização:	Realizar uma série de solicitações que acionem a atualização das estatísticas em tempo real. Medir o tempo de latência entre a solicitação e a atualização das estatísticas.
Recursos específicos:	Sistema configurado com uma carga de trabalho representativa. Estatísticas em tempo real ativadas e atualizadas regularmente. Ferramentas de medição de latência.

Nome Caso de teste:	Medir o uso eficiente de CPU e memória
Código:	TC03
Preparação:	Certificar-se de que o sistema está configurado com uma carga de trabalho representativa. Garantir que as operações intensivas estejam prontas para execução.
Inicialização:	Iniciar o processo de teste. Executar as operações intensivas. Monitorar e registrar o uso de CPU e memória durante a execução.
Recursos específicos:	Sistema configurado com uma carga de trabalho representativa. Operações intensivas prontas para execução. Ferramentas de monitoramento de CPU e memória.

Nome Caso de teste:	Avaliar a utilização do sistema
Código:	TC04

Preparação:	Certificar-se de que o sistema está configurado com uma carga de trabalho representativa. Garantir que as consultas frequentes estejam prontas para execução.
Inicialização:	Iniciar o processo de teste. Realizar uma série de consultas frequentes. Monitorar e registrar o desempenho do sistema durante as consultas.
Recursos específicos:	Sistema configurado com uma carga de trabalho representativa.  Consultas frequentes prontas para execução.  Ferramentas de monitoramento de desempenho do sistema.

Nome Caso de teste:	Desempenho com o número máximo de ginásios suportados
Código:	TC05
Preparação:	Configurar o sistema para suportar o número máximo de ginásios. Garantir que os ginásios estejam configurados e operacionais
Inicialização:	Iniciar o processo de teste. Realizar uma série de operações em todos os ginásios suportados. Monitorar e registrar o desempenho do sistema durante as operações.
Recursos específicos:	Sistema configurado para suportar o número máximo de ginásios.  Ginásios configurados e operacionais.  Ferramentas de monitoramento de desempenho do sistema.

Nome Caso de teste:	Fluidez dos serviços durante operações simultâneas em 200 ginásios
Código:	TC06
Preparação:	Configurar o sistema para simular operações simultâneas em 200 ginásios. Garantir que os ginásios estejam configurados e operacionais.
Inicialização:	Iniciar o processo de teste com as operações simultâneas em todos os ginásios. Monitorar e registrar a fluidez dos serviços durante as operações.
Recursos específicos:	Sistema configurado para simular operações simultâneas em 200 ginásios.  Ginásios configurados e operacionais.  Ferramentas de monitoramento de desempenho do sistema.

Resultados:

#### 1. Medir o tempo médio de resposta para o login, garantindo intervalo de 1-10 segundos:

- **Resultado:** O tempo médio de resposta para o login foi de 1469 ms (milissegundos), indicando que está dentro do intervalo desejado de 1 a 10 segundos.

```
2024-04-14 10:08:50.626 DEBUG testcase.LoginPost - 5: averageResponseTime = responseTimes.sum() / respo
2024-04-14 10:08:50.640 DEBUG testcase.LoginPost - 6: comment(Tempo Médio de Resposta: $averageResponse
2024-04-14 10:08:50.648 INFO c.k.k.c.keyword.builtin.CommentKeyword - Tempo Médio de Resposta: 1469.2 ms
2024-04-14 10:08:50.668 INFO c.k.katalon.core.main.TestCaseExecutor - END Test Cases/LoginPost
```

#### 2. Avaliar a latência na atualização de estatísticas em tempo real:

- **Resultado:** A latência na atualização das estatísticas em tempo real foi de 891 ms, indicando um tempo de resposta rápido e eficiente.

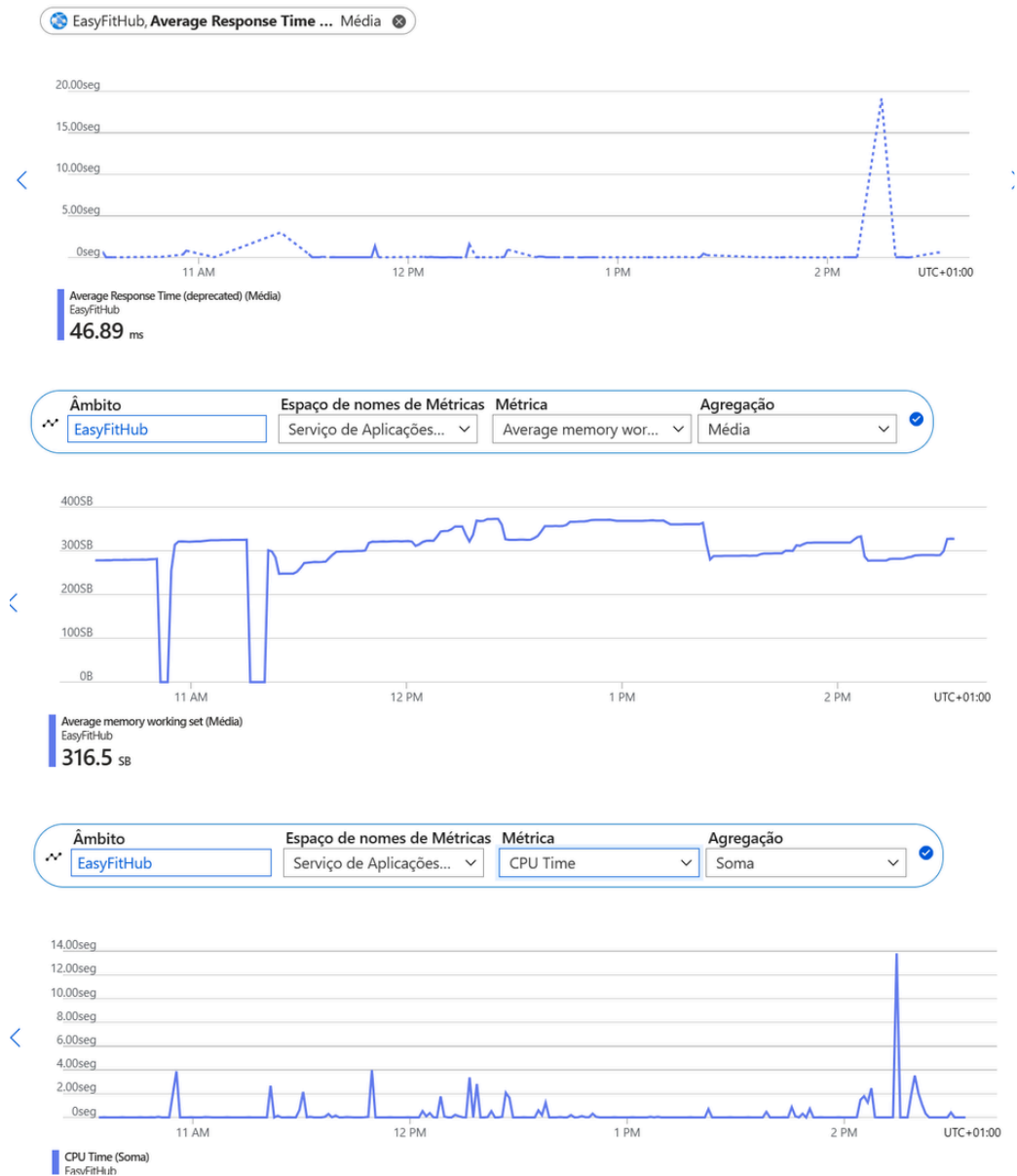
```
2024-04-14 10:54:24.324 DEBUG testcase.AvgLatency - 9: delay(1)
2024-04-14 10:54:25.332 DEBUG testcase.AvgLatency - 4: averageLatency = latencies.sum() / latencies.size
2024-04-14 10:54:25.361 DEBUG testcase.AvgLatency - 5: comment(Latência Média: $averageLatency ms)
2024-04-14 10:54:25.367 INFO c.k.k.c.keyword.builtin.CommentKeyword - Latência Média: 891.8 ms
2024-04-14 10:54:25.377 INFO c.k.katalon.core.main.TestCaseExecutor - END Test Cases/AvgLatency
```

### 3. Medir o uso eficiente de CPU e memória durante operações intensivas:

- **Resultado:** Verifica se houve picos no uso da CPU e da memória durante as operações das estatísticas.

### 4. Avaliar a utilização do sistema durante consultas frequentes:

- **Resultado:** Verifica se houve picos de utilização do sistema durante as operações das estatísticas.



### 5. Avaliar a fluidez dos serviços durante operações simultâneas em 200 ginásios:

- **Resultado:** Não houve alteração significativa no desempenho durante operações simultâneas em 200 ginásios, indicando uma boa capacidade de escalabilidade do sistema.

```
2024-04-14 13:49:40.555 DEBUG testcase.Performance200Gym - 1: sum += future.get()
2024-04-14 13:49:40.557 DEBUG testcase.Performance200Gym - 1: sum += future.get()
2024-04-14 13:49:40.559 DEBUG testcase.Performance200Gym - 1: sum += future.get()
2024-04-14 13:49:40.562 DEBUG testcase.Performance200Gym - 6: executor.shutdown()
2024-04-14 13:49:40.581 DEBUG testcase.Performance200Gym - 7: return (sum / amount)
Resultado 276.92
2024-04-14 13:49:40.602 INFO c.k.katalon.core.main.TestCaseExecutor - END Test Cases/Performance200Gym
```

6. Registrar e avaliar o desempenho com o número máximo de ginásios suportados:

- **Resultado:** O sistema suporta a volta de 1000 ginásios com um desempenho satisfatório, indicando uma boa capacidade de escalabilidade e robustez.

Compatibilidade

Nome Caso de teste:	GetPlatformStats
Código:	TC01
Finalidade:	Verificar se um utilizador autenticado consegue ver as Estatísticas da plataforma sem gastar muito tempo
Entradas:	Token de Sessão do utilizador autenticado + URL
Resultados esperados:	A página aparece em menos de 1 segundo
Dependências:	Utilizadores, Ginásios, e Clientes registados
Resultado obtido:	A página aparece em menos de 1 segundo

Nome Caso de teste:	GetEmployeeStats
Código:	TC02
Finalidade:	Verificar se um utilizador autenticado consegue ver as Estatísticas de Employee's sem gastar muito tempo
Entradas:	Token de Sessão do utilizador autenticado + URL
Resultados esperados:	A página aparece em menos de 1 segundo
Dependências:	Employees registados
Resultado obtido:	A página aparece em menos de 1 segundo

Nome Caso de teste:	GetGymStats
Código:	TC03
Finalidade:	Verificar se um utilizador autenticado consegue ver as Estatísticas de um Ginásio sem gastar muito tempo
Entradas:	Token de Sessão do utilizador autenticado + URL
Resultados esperados:	A página aparece em menos de 1 segundo

Dependências:	Utilizadores registados e autenticados
Resultado obtido:	A página aparece em menos de 1 segundo

Nome Caso de teste:	GetClientStats
Código:	TC04
Finalidade:	Verificar se um utilizador autenticado consegue ver as Estatísticas de Clients sem gastar muito tempos
Entradas:	Token de Sessão do utilizador autenticado + URL
Resultados esperados:	A página aparece em menos de 1 segundo
Dependências:	Utilizadores e Clientes registados
Resultado obtido:	A página aparece em menos de 1 segundo

### Capacidade de interação

(todos as subcaracterísticas: min. 30 questionários a utilizadores diferentes)

### Testes de Confiabilidade

(Tolerância a falhas e Recuperabilidade)

Nome Caso de teste:	Persistência no Create Exercise
Código:	TC01
Finalidade:	Verificar se um utilizador autenticado consegue adicionar um exercicio na Base de dados na ocorrencia de falha no servidor
Entradas	Campos do novo exercicio
Resultado	Utilizador perde acesso temporariamente ao sistema e a informação é persisitida
Dependências:	Ginasio registado

Nome Caso de teste:	Validação de campos na criação de Exercicios
Código:	TC02
Finalidade:	Verificar se um utilizador autenticado consegue adicionar um exercicio na Base de dados quando insere dados inválidos
Entradas	Campos do novo exercicio
Resultado	O Utilizador é notificado que não pode utilizar os valores declarados nos campos
Dependências:	Ginasio registado

Nome Caso de teste:	Persistência no Create Meal
Código:	TC01

Finalidade:	Verificar se um utilizador autenticado consegue adicionar uma Meal na Base de dados na ocorrência de falha no servidor
Entradas	Campos do novo Meal
Resultado	Utilizador perde acesso temporariamente ao sistema e a informação é persistida
Dependências:	Ginasio registado

Nome Caso de teste:	Validação de campos na criação de Meals
Código:	TC02
Finalidade:	Verificar se um utilizador autenticado consegue adicionar uma Meal na Base de dados quando insere dados inválidos
Entradas	Campos do novo Meal
Resultado	O Utilizador é notificado que não pode utilizar os valores declarados nos campos
Dependências:	Ginasio registado

## Segurança

(Integridade, Autenticidade e Resistência)

### Testes de Manutabilidade

Nome Caso de teste:	Resolução de Tarefas no período do Sprint
Código:	TC01
Finalidade:	Verificar se um a Equipa conseguiu concluir as tarefas todas no período do Sprint
Resultado	Houve 3 tarefas em falta
Dependências:	Acesso ao Jira

Nome Caso de teste:	Resolução de Bugs de Sprints anteriores no tempo esperado
Código:	TC02
Finalidade:	Verificar se um a Equipa conseguiu concluir as tarefas todas no tempo previsto
Resultado	A Equipa resolveu os Bugs no tempo previsto
Dependências:	Acesso ao Jira

Nome Caso de teste:	Utilização de Branches no processo de Desenvolvimento
Código:	TC03
Finalidade:	Verificar se um a Equipa geriu os diferentes Branches de Desenvolvimento corretamente.
Resultado	A Equipa geriu 5 Branches diferentes.

(baseado no jira: tempo médio que demoraram a implementar alterações ou correções)

### Testes de Flexibilidade

(testar em laptop e mobile e também com recurso a questionários)

### Testes de Segurança Física (Safety)

(Fail safe, Hazard warning)

### Testes de aceitação

(documento independente a entregar ao cliente)

## Testes de Usabilidade

Serão questionados, no mínimo, 30 indivíduos com níveis de conhecimento tecnológico distintos acerca do funcionamento do Sistema.

No final será medido o tempo gasto em cada etapa de cada tarefa. Também será pedido um feedback para determinar o grau da usabilidade do Sistema.

Os indivíduos cumprirão as seguintes tarefas do Questionário:

#### TC01 Manager

- Criar Conta com Ginásio,
- Fazer o Login na Conta,
- Mudar o Nome do Ginásio,
- Criar uma Máquina no Inventário,
- Editar a quantidade da Máquina,
- Fazer o Logout

#### TC02 Manager

- Fazer o Login na Conta,
- Aceder a lista de subscrições pendentes do Ginásio
- Confirmar uma subscrição de Cliente
- Fazer o Logout

#### TC03 Client

- Criar conta,
- Fazer o Login na conta,
- Pesquisar por um Ginásio,
- Subscrever no Ginásio,
- Fazer o Logout

#### TC04 Client

- Fazer Login numa conta existente,
- Aceder a um Ginásio vinculado,
- Escolher item na Loja,
- Fazer compra,



- Fazer o Logout

#### TC05 Client

- Entrar na página principal do Sistema
- Pesquisar por um Ginásio
- Aceder os detalhes do Ginásio
- Visualizar todas as fotos do Ginásio
- Retroceder, fazer o Login e subscrever no Ginásio

#### TC06 Admin

- Fazer o Login numa conta de Administrador
- Aceder a página de Ginásios pendentes
- Confirmar um Ginásio
- Fazer o Logout

Nome Caso de teste:	Formulário de Usabilidade
Código:	TC01
Preparação:	Antes de iniciar o teste, certifique-se de que o sistema está configurado com dados sensíveis, como informações financeiras ou pessoais, e com diferentes níveis de permissões para os utilizadores.
Inicialização:	Tentar aceder a dados sensíveis sem as permissões adequadas.
Recursos específicos:	Para este teste, serão necessários utilizadores com diferentes níveis de permissões e dados sensíveis, como números de cartão de crédito ou informações de saúde.

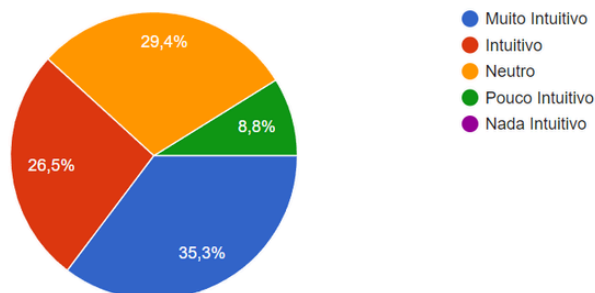
Formulário de usabilidade: [Interação do Utilizador no EasyFitHub](#)

#### Resultados dos testes:

Quão intuitivo foi o processo de criação de conta com o ginásio/utilizador?

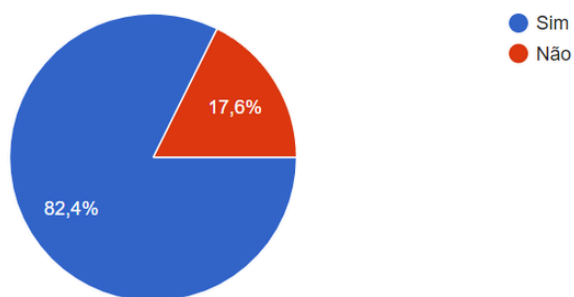
34 respostas

[Copiar](#)



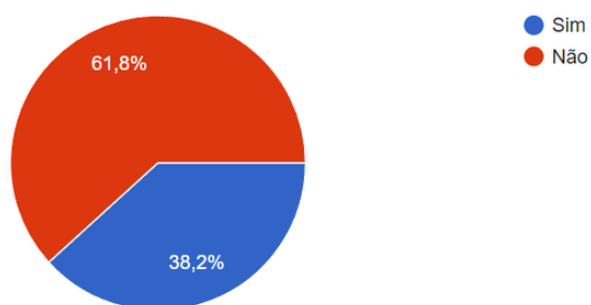
Os campos obrigatórios para criar a conta foram claramente indicados?

34 respostas



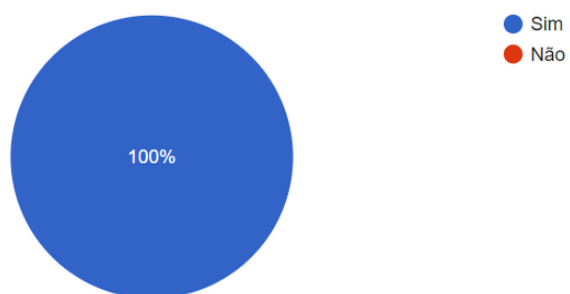
Houve alguma confusão ou dificuldade ao preencher os detalhes da conta com o ginásio/utilizador?

34 respostas



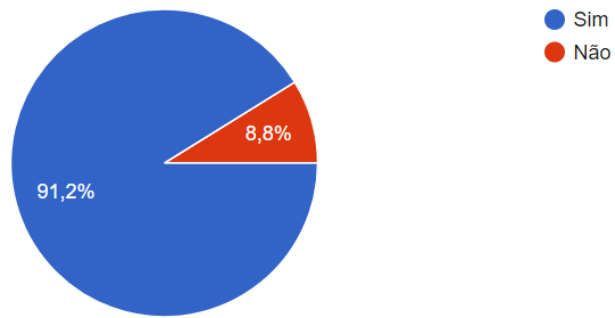
O processo de login foi rápido e sem problemas?

34 respostas



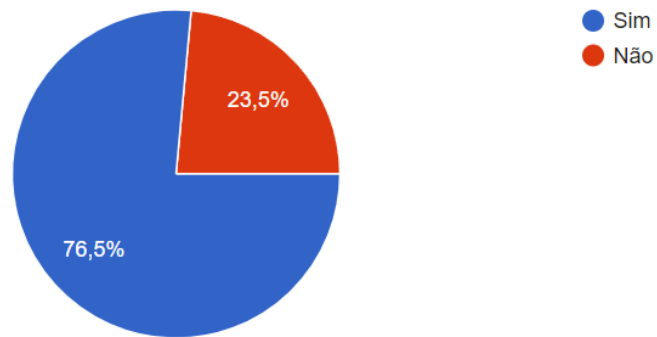
Os campos para inserir as credenciais estavam bem visíveis e de fácil acesso?

34 respostas



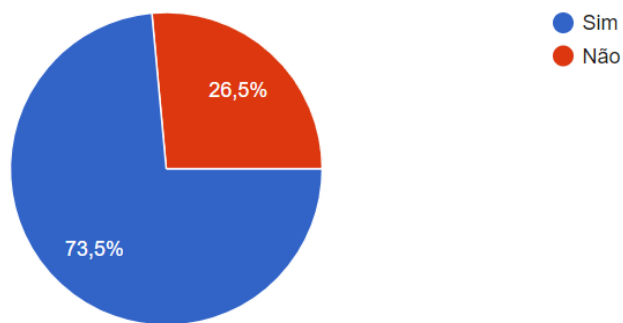
Foi fácil encontrar a opção para mudar dados pessoais/ginásio?

34 respostas



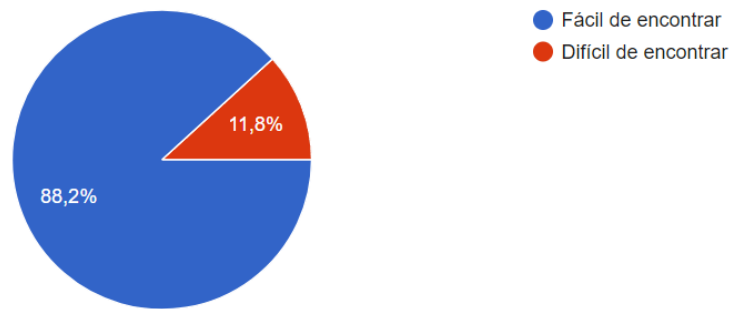
O processo de alteração de dados pessoais/ginásio foi direto e compreensível?

34 respostas



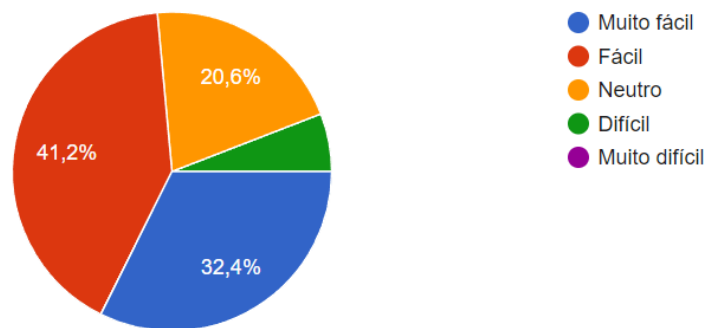
A criação de máquinas/itens para o ginásio foi fácil de encontrar?

34 respostas



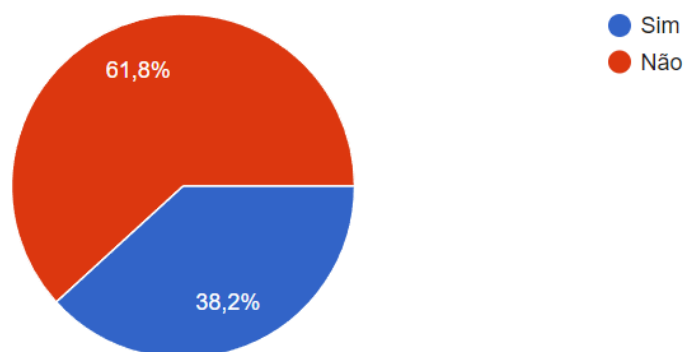
Quão fácil foi adicionar uma nova máquina ao inventário/item na loja?

34 respostas



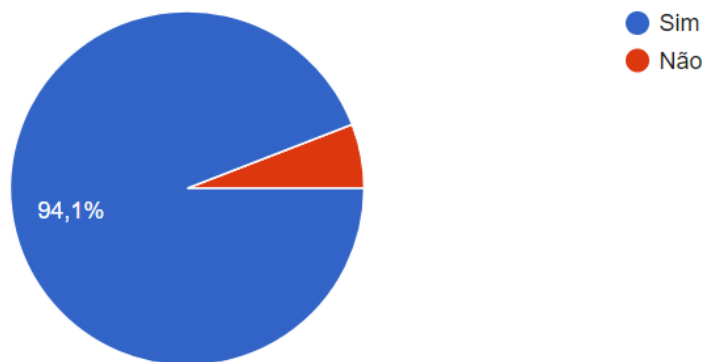
Houve alguma confusão ao preencher os detalhes da máquina/item?

34 respostas



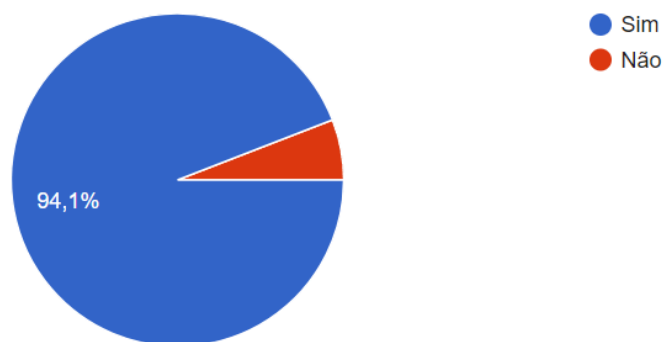
Como utilizador, o acesso ao ginásio que está inscrito foi fácil?

34 respostas



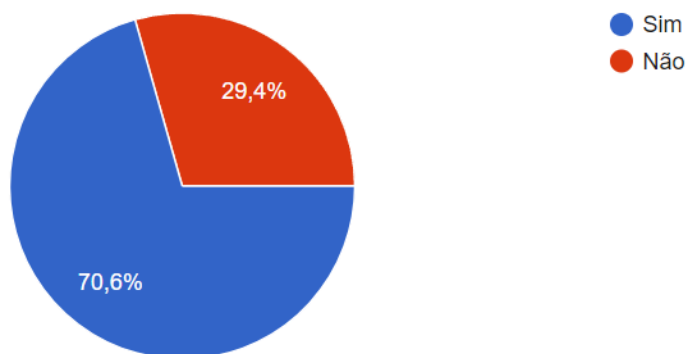
A seleção de um item na loja foi clara e intuitiva?

34 respostas



O processo de compra foi suave e sem problemas?

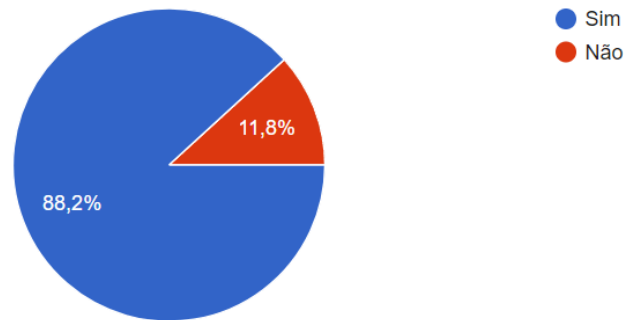
34 respostas



---

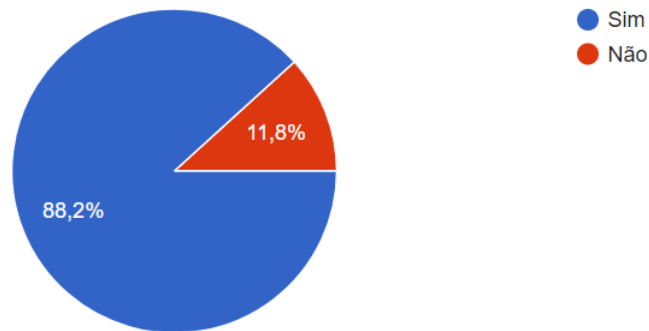
A pesquisa pelo ginásio foi eficaz e os resultados foram relevantes?

34 respostas



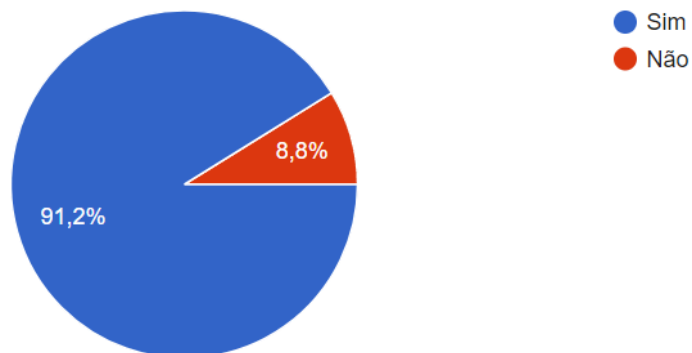
Os detalhes do ginásio estavam apresentados de forma clara e completa?

34 respostas



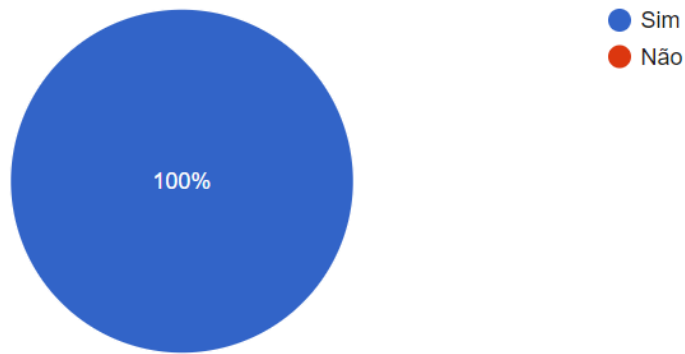
A opção para fazer logout estava claramente visível?

34 respostas



## O processo de logout foi simples e direto?

34 respostas



## Testes de Segurança

### Testes de Integridade

RQ37	Descrição
<b>Pior caso:</b>	Acesso não autorizado a dados sensíveis sem registro ou impedimento.
<b>Planeado:</b>	Tentar acessar dados sensíveis sem as devidas permissões e verificar se o sistema impede o acesso não autorizado e registra tentativas de violação.
<b>Teste:</b>	Tente acessar dados sensíveis sem as devidas permissões.
<b>Unidades:</b>	Capacidade de prevenção de acesso não autorizado.

RQ38	Descrição
<b>Pior caso:</b>	Comprometimento da confidencialidade de dados devido a ataques de injeção bem-sucedidos.
<b>Planeado:</b>	Realizar tentativas de injeção de código ou SQL para comprometer a confidencialidade dos dados, verificando se o sistema valida e filtra entradas de maneira eficaz.
<b>Teste:</b>	Realize tentativas de injeção de código ou SQL.
<b>Unidades:</b>	Eficiência na prevenção de ataques de injeção.

### Testes de Autenticidade

RQ45	Descrição
<b>Pior caso:</b>	Falha na verificação de identidade, comprometendo a autenticidade do acesso.
<b>Planeado:</b>	Verificar se o sistema requer e verifica a identidade dos usuários durante o processo de autenticação.

<b>Teste:</b>	Realize o processo de autenticação e verifique a identidade.
<b>Unidades:</b>	Eficiência na verificação de identidade.

RQ46	Descrição
<b>Pior caso:</b>	Comprometimento da integridade dos certificados digitais, prejudicando a autenticidade.
<b>Planeado:</b>	Avaliar a integridade dos certificados digitais utilizados no processo de autenticação, garantindo que não tenham sido comprometidos.
<b>Teste:</b>	Avalie a integridade dos certificados digitais.
<b>Unidades:</b>	Eficiência na detecção de certificados digitais comprometidos.

## Testes:

Nome Caso de teste:	Acesso não autorizado a dados sensíveis
Código:	TC01
Preparação:	Assegurar que o sistema está configurado com dados sensíveis e utilizadores com diferentes níveis de permissões.
Inicialização:	Tentar acessar dados sensíveis sem as permissões adequadas.
Recursos específicos:	Utilizadores com diferentes níveis de permissões, dados sensíveis.

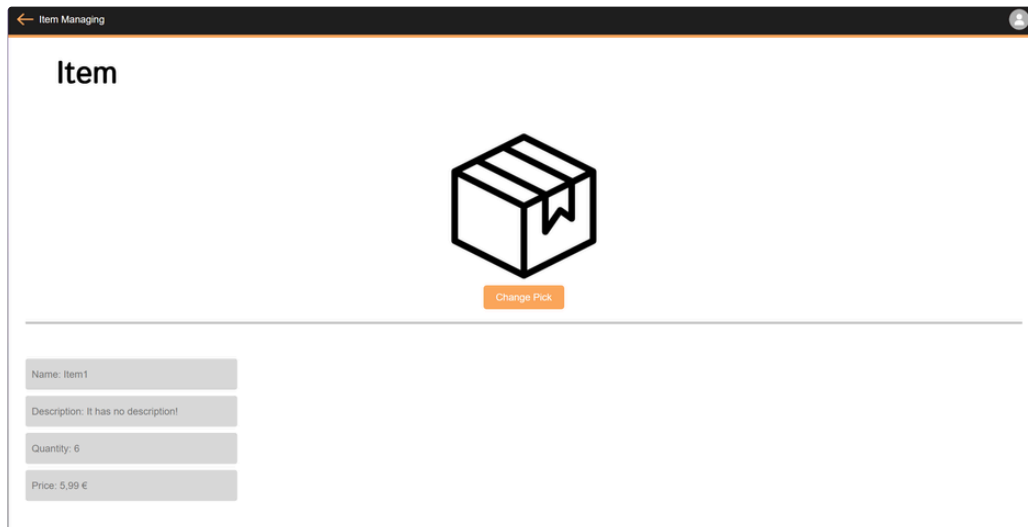
Nome Caso de teste:	Ataques de injeção para comprometer a confidencialidade
Código:	TC02
Preparação:	Assegurar que o sistema esteja configurado com entradas que podem ser suscetíveis a ataques de injeção.
Inicialização:	Realizar tentativas de injeção de código ou SQL para comprometer a confidencialidade dos dados.
Recursos específicos:	Entradas suscetíveis a ataques de injeção.

Nome Caso de teste:	Avaliação da integridade na autenticação
Código:	TC03
Preparação:	Obter o token e verificar se na base de dados tem o token, associado a conta que deu login.
Inicialização:	Realizar o processo de autenticação e verificar se a identidade é corretamente verificada.
Recursos específicos:	Token utilizado no processo de autenticação.



## Resultados:

### TC01:



Esta página serve tanto para ver o item como editá-los caso tenha as permissões para isso, como podemos ver, ao aceder sem permissões não podemos editar este item, só vê-lo.

### TC02:



Tentamos usar um injeção de código na barra de pesquisa dos ginásios, mas o ataque não foi sucedido e a base de dados continuou intacta.

### TC03:

▲	✓	EasyFitHubTests (25)	3 s
▲	✓	EasyFitHubTests (25)	3 s
▲	✓	AuthenticationTest (2)	2,3 s
	✓	AuthenticationsTest	2 s
	✓	CreateAccountTest	245 ms

No teste AuthenticationsTeste foi verificado se conseguimos obter a conta associada ao token e como pode ser visto pela imagem foi bem sucedido.

## Manual de utilização

Diretamente na plataforma e sensível ao contexto

(explicar estratégia)

## Manual técnico

(VSDocMan)