

Desenho detalhado



Engenharia de Software Aplicada
Desenho detalhado – Sprint nº3 - Pagamentos

Turma : 3

Grupo nº1

nº202100299 - Rui Barroso

nº202100296 - Gonçalo Vieira

nº202100984 - Francisco Silva

nº201901953 - André Pauli

1			
2			

- SUMARIO EXECUTIVO
- INTRODUÇÃO
- DESENHO DETALHADO
 - Introdução
 - Modulo Pagamentos
 - Requisitos
 - Diagrama de Classes
 - Diagramas de Processos de negócio
 - Interface com o utilizador
- Mockup Payment Method
- Mockup Confirm Payment
 - Testes
 - 3.3.1 Testes Unitários
 - Especificação dos Casos de testes:
 - Especificação dos Procedimentos:
 - Testes de Automação
 - Testes de Integração
 - Testes de Integração
 - Testes de Regressão
 - Testes de Integração com 3rdparty
 - Testes de Sistema
 - Testes de Funcionalidade
 - Testes:
 - Resultados:
 - Resultados:
 - Testes de Confiabilidade
 - Testes de Eficiência
 - Testes de Manutabilidade
 - Testes de Portabilidade
 - Testes de Usabilidade
- Testes de Segurança
 - Testes de Integridade
 - Testes de Autenticidade
- Testes:
- Resultados:

SUMARIO EXECUTIVO

O projeto EasyFitHub visa criar um ecossistema digital que integre eficientemente os aspectos cruciais da gestão de ginásios. Este sumário executivo destaca o módulos principal que vai ser implementado no sprint3 o de Pagamentos. Este documento possui requisitos específicos que serão detalhadamente abordados neste módulo.

INTRODUÇÃO

O EasyFitHub surge como resposta à necessidade crescente de gerenciamento eficiente e personalizado em ambientes de ginásio. Este documento é uma exploração detalhada dos requisitos funcionais, diagramas de classes, processos de negócio, mockups, testes unitários, testes de aceitação, testes de sistema e testes de integração relacionados ao desenvolvimento do sistema EasyFitHub. Ao compreender e delinear esses elementos, buscamos criar uma solução tecnológica completa e eficaz para a gestão de ginásios, proporcionando aos usuários uma experiência aprimorada.

DESENHO DETALHADO

Introdução

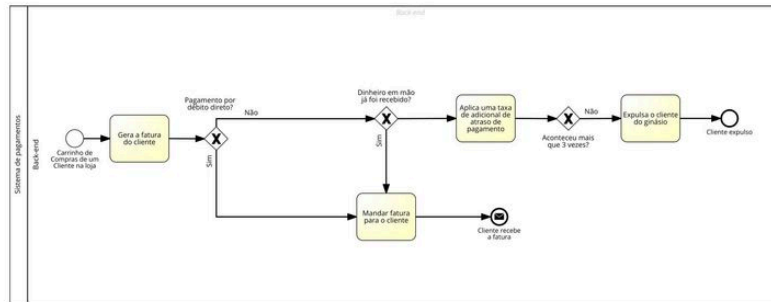
Modulo Pagamentos

Requisitos

ID	Módulo	Descrição	Prioridade
R1	Pagamentos	O sistema deverá lidar com operações de faturação.	Baixa
R2	Pagamentos	O sistema deverá emitir referências para pagamento.	Baixa
R3	Pagamentos	O sistema deverá permitir a confirmação do pagamento.	Baixa
R4	Pagamentos	O sistema deverá realizar operações CRUD no pagamento mensal.	Baixa

Diagrama de Classes

Sistema de Pagamento na Loja



Interface com o utilizador

Mockup Payment Method

Frame 37

Home icon

Profile Settings

Privacy Settings

Payment Method

Payment Method

Visa

Nº of card

Expire Date

Security Code

First Name

Last Name

City

Payment Address

Postal Code

Country

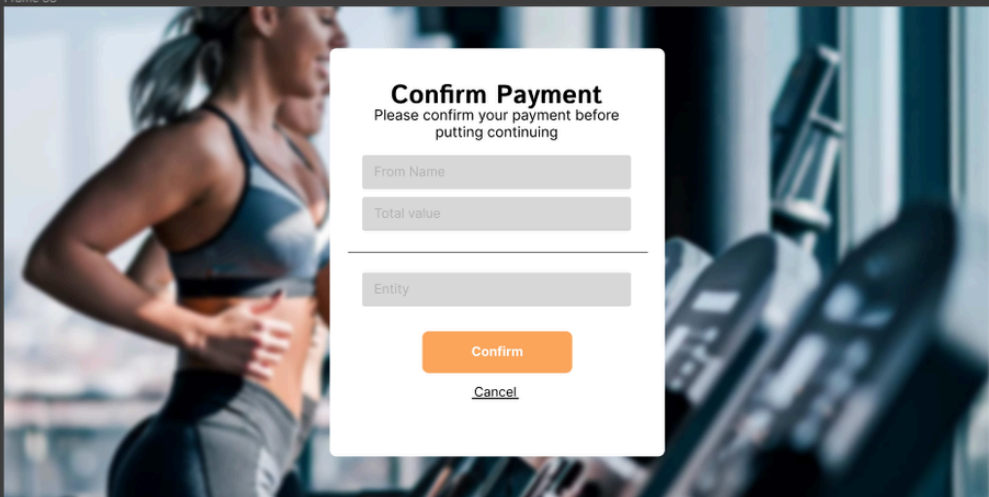
Phone Number

Continue

Cancel

Mockup Confirm Payment

Frame 38



Confirm Payment

Please confirm your payment before putting continuing

[Cancel](#)

Testes

3.3.1 Testes Unitários

Especificação dos Casos de testes:

Nome Caso de teste:	Adicionar Assinatura
Código:	TC01
Finalidade:	Verificar se a função de adição de assinatura adiciona corretamente uma nova assinatura ao sistema.
Entradas:	DebitCard
Resultados esperados:	A função deve adicionar a assinatura ao sistema e retornar um valor booleano com o resultado da operação.
Dependências:	A função de adição de assinatura deve estar implementada.

Nome Caso de teste:	Remover Assinatura
Código:	TC02
Finalidade:	Verificar se a função de remoção de assinatura remove corretamente uma assinatura do sistema.
Entradas:	ID válido da assinatura a ser removida.
Resultados esperados:	A função deve remover a assinatura correspondente ao ID fornecido do sistema.

Dependências:	A função de remoção de assinatura deve estar implementada.
---------------	--

Nome Caso de teste:	Funcionalidade Periódica
Código:	TC03
Finalidade:	Verificar se a função periódica é executada corretamente em intervalos específicos de tempo.
Entradas:	Configurações de tempo para a função periódica (por exemplo, intervalo de tempo, ação a ser realizada, etc.).
Resultados esperados:	A função periódica deve ser executada no intervalo de tempo especificado e realizar a ação designada de forma precisa e consistente.
Dependências:	A função periódica deve estar implementada.

Nome Caso de teste:	Criar Carrinho de Compras
Código:	TC04
Finalidade:	Verificar se a função de criação de carrinho de compras cria corretamente um novo carrinho de compras para um cliente.
Entradas:	ID do cliente para o qual o carrinho de compras está sendo criado, ID go ginásio e os itens.
Resultados esperados:	A função deve criar um novo carrinho de compras para o cliente especificado no sistema.
Dependências:	A função de criação de carrinho de compras deve estar implementada

Nome Caso de teste:	Remover Carrinho de Compras
Código:	TC05
Finalidade:	Verificar se a função de remoção de carrinho de compras remove corretamente um carrinho de compras do sistema.
Entradas:	ID válido do carrinho de compras a ser removido.
Resultados esperados:	A função deve remover o carrinho de compras correspondente ao ID fornecido do sistema.
Dependências:	A função de remoção de carrinho de compras deve estar implementada.

Nome Caso de teste:	Atualizar Detalhes de Débito
Código:	TC06
Finalidade:	Verificar se a função de atualização de detalhes de débito atualiza corretamente os detalhes de débito de um cliente.
Entradas:	ID do cliente e o novo DebitCard.

Resultados esperados:	A função deve atualizar os detalhes de débito do cliente correspondente ao ID fornecido no sistema.
Dependências:	A função de atualização de detalhes de débito deve estar implementada.

Nome Caso de teste:	Realizar Pagamento
Código:	TC07
Finalidade:	Verificar se a função de realização de pagamento processa corretamente o pagamento de uma assinatura.
Entradas:	ID da assinatura e dados de pagamento.
Resultados esperados:	A função deve processar o pagamento da assinatura correspondente ao ID fornecido e atualizar o status do pagamento no sistema.
Dependências:	A função de realização de pagamento deve estar implementada. O API de pagamentos.

Nome Caso de teste:	Criar Recibo
Código:	TC08
Finalidade:	Verificar se a função de criação de recibo cria corretamente um recibo para um pagamento.
Entradas:	ID do pagamento para o qual o recibo está sendo criado
Resultados esperados:	A função deve criar um recibo para o pagamento correspondente ao ID fornecido no sistema.
Dependências:	A função de criação de recibo deve estar implementada.

Especificação dos Procedimentos:

Nome Caso de teste:	Adicionar Assinatura
Código:	TC01
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam clientes e ginásios no sistema.
Inicialização:	Chamar a função de adicionar assinatura, fornecendo dados válidos para uma nova assinatura.
Recursos específicos:	Dados de clientes existentes no sistema.

Nome Caso de teste:	Remover Assinatura
Código:	TC02

Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam assinaturas válidas no sistema.
Inicialização:	Chamar a função de remover assinatura, fornecendo o ID de uma assinatura existente no sistema.
Recursos específicos:	Dados de assinaturas existentes no sistema.

Nome Caso de teste:	Funcionalidade Periódica
Código:	TC03
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que a função periódica esteja devidamente implementada.
Inicialização:	Configurar a função periódica para realizar uma determinada ação em intervalos de tempo específicos.
Recursos específicos:	Mecanismo de simulação do tempo para garantir que a função seja testada de forma controlada e previsível.

Nome Caso de teste:	Criar Carrinho de Compras
Código:	TC04
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam clientes e ginásios no sistema.
Inicialização:	Chamar a função de criar carrinho de compras, fornecendo o ID de um cliente existente no sistema.
Recursos específicos:	Dados de clientes e Ginásios existentes no sistema.

Nome Caso de teste:	Remover Carrinho de Compras
Código:	TC05
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam carrinhos de compras válidos no sistema.
Inicialização:	Chamar a função de remover carrinho de compras, fornecendo o ID de um carrinho de compras existente no sistema.
Recursos específicos:	Dados de carrinhos de compras existentes no sistema.

Nome Caso de teste:	Atualizar Detalhes de Débito
Código:	TC06
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam clientes e ginásios no sistema.
Inicialização:	Chamar a função de atualizar detalhes de débito, fornecendo o ID de um cliente existente e dados válidos para os detalhes de débito.

Recursos específicos:	Dados de clientes e ginásios existentes no sistema.
-----------------------	---

Nome Caso de teste:	Realizar Pagamento
Código:	TC07
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam clientes e ginásios com informações de pagamento configuradas.
Inicialização:	Chamar a função de realizar pagamento, fornecendo o ID de um pagamento existente.
Recursos específicos:	Dados de pagamentos existentes no sistema.

Nome Caso de teste:	Criar Recibo
Código:	TC08
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam pagamentos válidos no sistema.
Inicialização:	Chamar a função de criar recibo, fornecendo o ID de um pagamento existente.
Recursos específicos:	Dados de pagamentos existentes no sistema.

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.



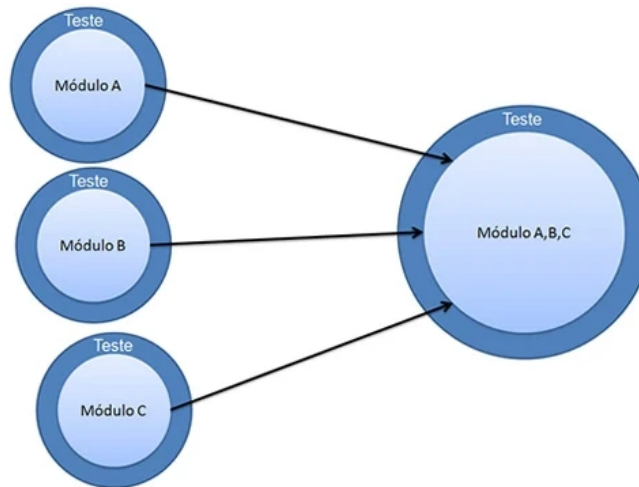
Testes de Integração

Testes de Integração

O teste de integração que iremos utilizar será o BigBang.

O teste de integração de big bang é uma abordagem de teste de software em que todos os componentes ou módulos do sistema são integrados simultaneamente para testar a sua funcionalidade como um todo.

Ao realizar o teste, todos os módulos serão combinados e testados em conjunto para verificar se o sistema como um todo funciona corretamente.



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

Testes de Funcionalidade

Requisitos de Qualidade:

RQ1	Descrição
Pior caso:	0% de sucesso na execução das operações básicas.
Planeado:	Verificar 100% de sucesso na execução das operações básicas.
Teste:	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.
Unidades:	Taxa de sucesso na execução das operações básicas.

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

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

RQ6	Descrição
Pior caso:	Obter uma taxa de satisfação nas recomendações baseadas no histórico de 60%.
Planeado:	Alcançar uma taxa de satisfação nas recomendações baseadas no histórico de 92%.
Teste:	Explore funcionalidades que se baseiam no histórico do usuário, avaliando se o sistema utiliza essas informações de maneira eficaz.
Unidades:	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 Account, Gym, Item, Machine e verificar se a atualização do Profile funciona.
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 Stripe
Inicialização:	Executar os testes unitários.
Recursos específicos:	Ligação com o Api do Stripe

Resultados:

▲ ✓ EasyFitHubTests (33)	24,5 s
▲ ✓ EasyFitHubTests (33)	24,5 s
▷ ✓ AuthenticationTest (2)	1,6 s
▷ ✓ GymTest (5)	500 ms
▷ ✓ InventoryInfoTest (15)	165 ms
▷ ✓ ProfileInfoTest (2)	184 ms
▷ ✓ SearchInfoTest (4)	87 ms
▷ ✓ StripeTest (5)	22 s

No testes unitários foi verificado se o Sucesso das operações básicas e das mais complexas como pode ser visto na imagem acima

Testes de Compatibilidade

Requisitos de Qualidade:

RQ13	Descrição
Pior caso:	Identificar conflitos entre o sistema em teste e o sistema simultâneo, sem resolução adequada.
Planeado:	Verificar se o sistema pode ser executado simultaneamente com outro sistema sem conflitos, resolvendo possíveis problemas de coexistência.
Teste:	Execute o sistema simultaneamente com outro sistema que compartilhe recursos.
Unidades:	Conflitos identificados e resolvidos.

RQ14	Descrição
Pior caso:	Identificar ações externas que afetam adversamente o sistema e não podem ser resolvidas.
Planeado:	Avaliar se o sistema continua operacional e não é afetado pela presença de outras aplicações durante a interação.
Teste:	Interaja com outras aplicações concorrentes enquanto o sistema está em execução.
Unidades:	Resiliência a interferências externas.

Testes:

Nome Caso de teste:	Compatibilidade e coexistência do sistema
Código:	TC03
Preparação:	Preparar os servers e a base de dados e conectar os server a base de dados
Inicialização:	Executar os sistemas simultaneamente na cloud e no localhost e verificar se existem conflitos na base de dados partilhada
Recursos específicos:	Server na cloud, Server no localhost e uma base de dados na cloud

Resultados:

Os 2 sistemas foram lançados em simultâneo e varias funcionalidades foram testadas, como:

Autenticação, Pesquisa de ginásios, Visualização do gym e Modificação das relações.

Não houve nenhum conflito encontrado entre os sistemas durante a execução simultânea.


Testes de Confiabilidade

RQ29	Descrição
Pior caso:	Falhas significativas ou comportamento instável durante a execução de funcionalidades básicas.
Planeado:	Realizar testes extensivos nas funcionalidades básicas do sistema, avaliando a estabilidade e o comportamento esperado em condições normais de uso.
Teste:	Realize testes extensivos nas funcionalidades básicas do sistema.
Unidades:	Percentual de execução bem-sucedida de funcionalidades básicas.

Nome Caso de teste:	Falhas e instabilidade no Sistema na execução de funcionalidades básicas
Código:	TC29
Preparação:	Certificar-se de que o sistema está configurado e operacional. Garantir que existam funcionalidades básicas implementadas no sistema.
Inicialização:	Chamar funções relacionadas ao módulo inventory
Recursos específicos:	Dados de gym existentes no sistema.
Resultado:	O Sistema conseguiu executar as funcionalidades básicas <div><div><div></div></div>GetGyms_GetStore_GetIndex171 ms</div>

RQ30	Descrição
Pior caso:	Falha do sistema ou comportamento inadequado em situações de sobrecarga.
Planeado:	Submeter o sistema a situações de sobrecarga simuladas, como picos de tráfego, e avaliar se o sistema mantém a estabilidade e continua a operar sem falhas graves.
Teste:	Submeta o sistema a situações de sobrecarga simuladas.
Unidades:	Estabilidade do sistema em condições de sobrecarga.

Nome Caso de teste:	Falha do sistema ou comportamento inadequado em situações de sobrecarga.
Código:	TC30
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Submeter o sistema a situações de sobrecarga de utilizadores, e avaliar se o sistema mantém a estabilidade e continua a operar sem falhas graves.
Recursos específicos:	Acesso a métricas de funcionamento do Sistema

Resultado:	O Sistema apresentou comportamento adequado na autenticação de 1000 Users. <div>  MaxSimultaneousPostClients 280 ms </div>
------------	--

RQ31	Descrição
Pior caso:	Tempo de inatividade excessivo que não atende aos requisitos estabelecidos.
Planeado:	Monitorar o tempo de inatividade do sistema ao longo de um período específico, verificando se atende aos requisitos de disponibilidade estabelecidos.
Teste:	Monitore o tempo de inatividade do sistema ao longo de um período específico.
Unidades:	Percentual de tempo de inatividade em relação ao tempo total.

Nome Caso de teste:	Tempo de inatividade excessivo que não atende aos requisitos estabelecidos.
Código:	TC31
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Monitore o tempo de inatividade do sistema ao longo de uma semana.
Recursos específicos:	Acesso ao Sistema
Resultado:	O Sistema encontrou-se ativo no período de tempo expectado

RQ32	Descrição
Pior caso:	Incapacidade do sistema de se recuperar ou manter disponibilidade sob condições adversas simuladas.
Planeado:	Simular condições adversas, como falhas de servidores, e avaliar como o sistema responde, verificando se consegue manter um nível adequado de disponibilidade.
Teste:	Simule condições adversas, como falhas de servidores.
Unidades:	Tempo de recuperação após condições adversas.

Nome Caso de teste:	Incapacidade do sistema de se recuperar ou manter disponibilidade sob condições adversas.
Código:	TC32
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Simular sobrecarga de utilizadores e avaliar como o sistema responde, verificando se consegue manter um nível adequado de disponibilidade.
Recursos específicos:	Acesso ao Servidor e Base de Dados na Cloud
Resultado:	O Sistema conseguiu manter-se disponível.

RQ33	Descrição
Pior caso:	Impacto significativo no funcionamento do sistema após a falha controlada.

Planeado:	Intencionalmente desativar um componente crítico do sistema e avaliar como o restante do sistema responde, verificando se continua a funcionar sem impacto significativo.
Teste:	Desative intencionalmente um componente crítico do sistema.
Unidades:	Capacidade de isolamento e continuidade do serviço.

Nome Caso de teste:	Impacto no funcionamento do sistema após falha.
Código:	TC33
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Intencionalmente desativar o ProfileInfo do sistema e avaliar como o restante do sistema responde.
Recursos específicos:	Acesso ao Sistema
Resultado:	O Sistema respondeu adequadamente, falhando a persistência de Dados de Perfil

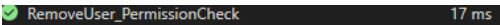
RQ34	Descrição
Pior caso:	Falha na manutenção da funcionalidade básica durante falhas de rede simuladas.
Planeado:	Simular falhas de rede e avaliar como o sistema lida com a perda temporária de conectividade, verificando se mantém a funcionalidade básica mesmo durante interrupções de rede.
Teste:	Simule falhas de rede e avalie a resposta do sistema.
Unidades:	Tempo de recuperação após falhas de rede simuladas.

Nome Caso de teste:	Falha na manutenção da funcionalidade básica durante falhas de rede
Código:	TC34
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Simular falhas de rede e avaliar como o sistema lida com a perda temporária de conectividade, verificando se mantém a funcionalidade básica
Recursos específicos:	Acesso ao Sistema na Cloud
Resultado:	O sistema conseguiu persistir a informação após a conexão ser reestabelecida

RQ35	Descrição
Pior caso:	Incapacidade do sistema de se recuperar adequadamente a partir de backups.
Planeado:	Avaliar a capacidade do sistema de se recuperar a partir de backups, simulando a perda de dados e verificando se pode ser restaurado a um estado funcional.
Teste:	Simule a perda de dados e avalie a recuperação a partir de backups.
Unidades:	Eficácia na recuperação a partir de backups.

Nome Caso de teste:	Recuperação adequada a partir de backups.
Código:	TC35
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Avaliar a capacidade do sistema de se recuperar a partir de backups, simulando a perda de dados e verificando se pode ser restaurado a um estado funcional.
Recursos específicos:	Acesso ao serviço de backup do Sistema
Resultado:	O sistema conseguiu recompor a informação do Backup

RQ36	Descrição
Pior caso:	Incapacidade do sistema de se recuperar após uma falha crítica, resultando em perda significativa de dados.
Planeado:	Simular uma falha crítica no sistema e avaliar como ele se recupera, verificando se pode retomar as operações normais com o mínimo de perda de dados.
Teste:	Simule uma falha crítica no sistema e avalie a recuperação.
Unidades:	Tempo de recuperação após falha crítica.


Nome Caso de teste:	Recuperação após uma falha crítica, resultando em perda de dados.
Código:	TC36
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Remover dados de Utilizador durante uma operação com autorizações específicas e ver se foram efetivadas na Base de dados depois de quanto tempo.
Recursos específicos:	Acesso à Base de Dados do Sistema na Cloud
Resultado:	O sistema lidou bem com o lapso nos dados, não persistindo a informação após a remoção dos Dados. 


Testes de Eficiência

RQ7	Descrição
Pior caso:	Exceder o tempo máximo de resposta para agendamento (mais de 10 segundos).
Planeado:	Manter um tempo médio de resposta para agendamento entre 5 e 10 segundos.
Teste:	Agende uma aula no sistema e meça o tempo que leva para a operação ser concluída. Verifique se está dentro da faixa especificada de 5 a 10 segundos.
Unidades:	Tempo médio de resposta para agendamento.


RQ8	Descrição
Pior caso:	Exceder uma latência na atualização de estatísticas de 6 segundos.

Planeado:	Alcançar uma latência na atualização de estatísticas inferior a 3 segundos.
Teste:	Atualize estatísticas no sistema e verifique se os dados mais recentes são exibidos imediatamente. Isso garante que o sistema mantenha informações em tempo real.
Unidades:	Latência na atualização de estatísticas.
Resultado:	Verificou-se assíncronidade, e o tempo de espera foi negligente.

RQ9	Descrição
Pior caso:	Apresentar uma utilização ineficiente de CPU e memória durante operações intensivas.
Planeado:	Manter o uso eficiente de CPU e memória durante operações intensivas.
Teste:	Execute operações intensivas no sistema, como processamento de grandes conjuntos de dados. Meça a utilização de recursos, garantindo otimização durante essas operações.
Unidades:	Uso eficiente de CPU e memória.
Resultado:	 <p>Este tempo de resposta foi atribuído à criação de 1000 Users</p>

RQ10	Descrição
Pior caso:	Apresentar uma utilização ineficiente de CPU e memória durante consultas frequentes.
Planeado:	Garantir uma utilização eficiente de CPU e memória durante consultas frequentes.
Teste:	Realize uma grande quantidade de consultas aos dados do sistema e meça a utilização de recursos, garantindo otimização durante essas operações.
Unidades:	Uso eficiente de CPU e memória.
Resultado:	 <p>Tempo de execução de 200 chamadas GetGyms</p>

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

	<div>  MaxCreateGymTest 7.4 sec </div> <p>Tempo de resposta de criação de 1500 ginásios em simultâneo</p>
--	---

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

Nome Caso de teste:	Avaliar a capacidade do Sistema de lidar com um Stress Test
Código:	TC05
Preparação:	Certificar-se que a Base de Dados encontra-se funcional.
Inicialização:	Verificar o tempo de resposta do Servidor quando encontra-se sobrecarregado de utilizadores autenticados em simultâneo
Recursos específicos:	Acesso ao Sistema

Testes de Manutabilidade

RQ47	Descrição
Pior caso:	Impacto negativo nos módulos existentes devido à adição de um novo módulo.
Planeado:	Adicionar um novo módulo ao sistema e verificar se a integração é feita sem impacto negativo nos módulos existentes.
Teste:	Adicione um novo módulo e avalie o tempo de integração no Jira.
Unidades:	Facilidade de adição de novos módulos sem causar rupturas.

Nome Caso de teste:	Adicione um novo módulo e avalie o tempo de integração no Jira.
Código:	TC47
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Adicionar o módulo Pagamentos ao sistema e verificar se a integração é feita sem impacto negativo nos módulos existentes.
Recursos específicos:	Acesso ao Sistema e ao Jira e testes Unitários
Resultado:	<p>O Jira indica que o tempo de integração reflete o esperado.</p> <p>As Sprint Reviews confirmam o resultado</p> <p>Os testes Unitários indicam que o funcionamento dos outros Módulos mantém-se na adição do módulo Payments</p>

RQ48	Descrição
Pior caso:	Modificações em um módulo afetam negativamente outros módulos.
Planeado:	Fazer modificações em um módulo e verificar se essas modificações permanecem isoladas, sem afetar outros módulos.
Teste:	Faça modificações em um módulo e avalie o isolamento.
Unidades:	Capacidade de manter o isolamento entre os módulos.

Nome Caso de teste:	Modificar um Módulo e avaliá-lo em isolamento
Código:	TC48
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Fazer modificações no módulo GymExplorer e verificar se essas modificações permanecem isoladas
Recursos específicos:	Acesso ao Sistema e a testes Unitários
Resultado:	As mudanças no módulo não afetaram módulos adjacentes segundo os Testes Unitários.

RQ49	Descrição
Pior caso:	Problemas de eficiência ou conflitos ao reutilizar componentes.
Planeado:	Utilizar um componente em diferentes partes do sistema e verificar se a reutilização é eficiente e não causa conflitos.
Teste:	Utilize um componente em diferentes partes do sistema.
Unidades:	Facilidade e eficiência na reutilização de componentes.

Nome Caso de teste:	Conflitos na reutilização de componentes
Código:	TC49
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Utilizar o AuthenticationInfo no módulo Payment do sistema e verificar se a reutilização é eficiente e não causa conflitos.
Recursos específicos:	Acesso ao Sistema
Resultado:	A reutilização do AuthenticationInfo é eficiente e não causou conflitos no módulo Payment

RQ50	Descrição
Pior caso:	Falta de adaptabilidade dos componentes em diferentes contextos.
Planeado:	Tentar utilizar um componente em um contexto diferente daquele para o qual foi

	originalmente projetado, avaliando a adaptabilidade.
Teste:	Tente utilizar um componente em um contexto diferente.
Unidades:	Adaptabilidade dos componentes em diferentes contextos.

Nome Caso de teste:	Adaptabilidade dos componentes em diferentes contextos
Código:	TC50
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Utilizar o AuthenticationInfo no módulo inventory.
Recursos específicos:	Acesso ao Sistema
Resultado:	O authenticationInfo apresentou comportamentos espectados no módulo Inventory

RQ51	Descrição
Pior caso:	Dificuldade em analisar o código-fonte devido à falta de estrutura para análise estática.
Planeado:	Utilizar ferramentas de análise estática para avaliar a qualidade do código-fonte, verificando se o sistema é construído de forma a facilitar análises estáticas.
Teste:	Utilize ferramentas de análise estática no código-fonte.
Unidades:	Facilidade de análise estática do código.

Nome Caso de teste:	Utilize ferramentas de análise estática no código-fonte.
Código:	TC51
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Utilizar Visual Code pra analisar o código fonte.
Recursos específicos:	Acesso ao Sistema e a ferramentas de análise estática
Resultado:	O Visual Code não apresenta 'Warnings' nem erros no processo 'Build Solution'

RQ52	Descrição
Pior caso:	Falta de rastreabilidade eficaz, comprometendo a compreensão dos requisitos atendidos por cada componente.
Planeado:	Verificar se é possível rastrear cada componente até os requisitos específicos que ele atende, avaliando a clareza na associação entre requisitos e componentes.
Teste:	Tente rastrear componentes até requisitos específicos.
Unidades:	Rastreabilidade eficaz de requisitos até componentes correspondentes.

Nome Caso de teste:	Rastrear componentes até requisitos específicos.
---------------------	--

Código:	TC52
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Verificar se é possível rastrear cada componente do módulo Autenticação até os requisitos específicos que ele atende.
Recursos específicos:	Acesso ao Sistema e à Análise e Especificação de Requisitos
Resultado:	O módulo Autenticação foi implementado conforme os requisitos.

RQ53	Descrição
Pior caso:	Dificuldade em incorporar mudanças nos requisitos sem comprometer a funcionalidade existente.
Planeado:	Introduzir mudanças nos requisitos do sistema e avaliar a facilidade com que as alterações podem ser incorporadas sem causar impactos significativos.
Teste:	Introduza mudanças nos requisitos e avalie a adaptação do sistema.
Unidades:	Facilidade de adaptação a mudanças nos requisitos.

Nome Caso de teste:	Avaliar a adaptação do sistema a mudanças
Código:	TC53
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Introduzir mudanças nos requisitos e avaliar a adaptação do sistema.
Recursos específicos:	Acesso ao Sistema e à Análise e Especificação de Requisitos
Resultado:	O sistema adaptou-se espectavelmente bem

RQ54	Descrição
Pior caso:	Problemas na atualização de bibliotecas externas, causando incompatibilidades.
Planeado:	Atualizar bibliotecas externas utilizadas no sistema e avaliar se o sistema se adapta sem problemas às novas versões.
Teste:	Atualize bibliotecas externas e avalie a adaptação do sistema.
Unidades:	Facilidade na atualização de bibliotecas externas.

Nome Caso de teste:	Avaliar a adaptação do sistema em relação a Bibliotecas externas
Código:	TC54
Preparação:	Certificar-se de que o sistema está configurado e operacional e implementa corretamente uma determinada biblioteca externa
Inicialização:	Atualizar StripeService e avaliar a adaptação do sistema.
Recursos específicos:	Acesso ao Sistema

Resultado:	O sistema adaptou as novas atualizações com êxito
------------	---

RQ55	Descrição
Pior caso:	Falta de cobertura abrangente, comprometendo a detecção de falhas.
Planeado:	Executar testes abrangentes e verificar se todas as partes do sistema são cobertas pelos testes, avaliando a qualidade da cobertura de testes.
Teste:	Execute testes abrangentes e avalie a cobertura.
Unidades:	Cobertura abrangente de testes.

Nome Caso de teste:	Avaliar cobertura de testes
Código:	TC55
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Executar testes nas componentes e verificar se todas as partes do sistema são cobertas pelos testes, avaliando a qualidade da cobertura de testes.
Recursos específicos:	Acesso ao Sistema e a Ferramenta de Teste Unitários
Resultado:	Os testes abrangiram todas as componentes essenciais. E os testes passaram todos.

RQ56	Descrição
Pior caso:	Dificuldade na criação e execução de cenários de teste complexos.
Planeado:	Criar cenários de teste complexos e avaliar a facilidade com que esses cenários podem ser desenvolvidos e executados.
Teste:	Crie cenários de teste complexos e avalie a facilidade de desenvolvimento e execução.
Unidades:	Facilidade na criação de cenários de teste complexos.

Nome Caso de teste:	Facilidade de desenvolvimento e execução de testes
Código:	TC56
Preparação:	Certificar-se de que o sistema está configurado e operacional.
Inicialização:	Crie testes unitários complexos e avalie a facilidade de desenvolvimento e execução dos mesmos
Recursos específicos:	Acesso a ferramentas de Testes Unitários
Resultado:	O desenvolvimento e execução dos testes Unitários complexos não apresentou dificuldades.

Testes de Portabilidade

RQ60	Descrição
------	-----------

Pior caso:	Instabilidades inesperadas devido a alterações simultâneas.
Planeado:	Realizar alterações simultâneas em diferentes partes do sistema e avaliar se essas mudanças podem causar instabilidades inesperadas.
Teste:	Realize alterações simultâneas e avalie a estabilidade.
Unidades:	Estabilidade durante alterações simultâneas.

RQ61	Descrição
Pior caso:	Problemas na substituição de componentes, causando incompatibilidades.
Planeado:	Substituir um componente do sistema por uma versão atualizada ou alternativa e avaliar se a substituição é realizada de maneira transparente.
Teste:	Substitua um componente e avalie a transparência na substituição.
Unidades:	Transparência na substituição de componentes.

RQ62	Descrição
Pior caso:	Quebras nas interfaces durante a substituição de componentes.
Planeado:	Realizar uma substituição de componente e verificar se as interfaces com outros componentes são mantidas, evitando quebras na comunicação.
Teste:	Substitua um componente e avalie a manutenção das interfaces.
Unidades:	Manutenção de interfaces durante a substituição.

Nome Caso de teste:	Funcionabilidade do Sistema em diferentes Dispositivos
Código:	TC07
Preparação:	Certificar-se que o Sistema encontra-se Online
Inicialização:	Aceder o Sistema através de SmartPhones e verificar se as páginas encontram-se formatadas devidamente
Recursos específicos:	Acesso ao Sistema

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](#)

Testes de Segurança

Testes de Integridade

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

RQ38	Descrição
Pior caso:	Comprometimento da confidencialidade de dados devido a ataques de injeção bem-sucedidos.
Planeado:	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.
Teste:	Realize tentativas de injeção de código ou SQL.
Unidades:	Eficiência na prevenção de ataques de injeção.

Testes de Autenticidade

RQ45	Descrição
Pior caso:	Falha na verificação de identidade, comprometendo a autenticidade do acesso.
Planeado:	Verificar se o sistema requer e verifica a identidade dos usuários durante o processo de autenticação.
Teste:	Realize o processo de autenticação e verifique a identidade.
Unidades:	Eficiência na verificação de identidade.

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

Testes:

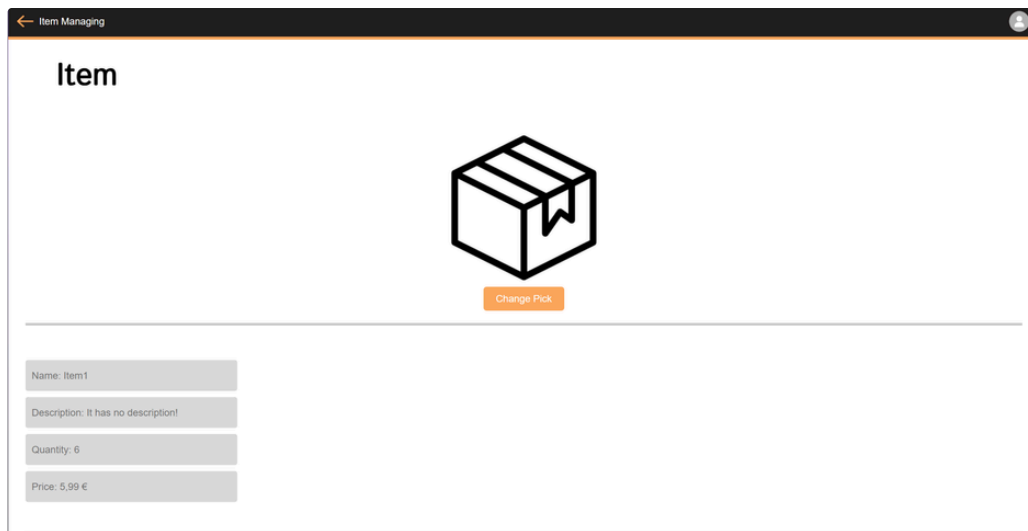
Nome Caso de teste:	Acesso não autorizado a dados sensíveis
Código:	TC01
Preparação:	Assegurar que o sistema está configurado com dados sensíveis e utilizadores com diferentes níveis de permissões.
Inicialização:	Tentar acessar dados sensíveis sem as permissões adequadas.
Recursos específicos:	Utilizadores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