Caso de Teste:	CT 001 - Login não acessar
Pré - condições :	Estar na tela principal
Procedimentos:	1) o ator não consegue acessar com seu login 2) o sistema não retorna uma mensagem de erro para o usuário.
Resultado esperado:	Mensagem de erro do sistema.
Dado de entrada:	Ausência de mensagem
Prioridade	Alta
Ambiente	Android
Técnica	Manual
Iteração	1º Iteração

Caso de Teste:	CT 002 - Usuário não consegue adicionar veículo
Pré - condições :	Está na tela de veículo
Procedimentos:	1) o ator não consegue adicionar veículo 2) o sistema não retorna mensagem de error para o usuário
Resultado esperado:	Mensagem de erro informando para o Usuário que ele não consegue adicionar veículo.
Dado de entrada:	Ausência de mensagem
Prioridade	Alta
Ambiente	Android
Técnica	Manual
Iteração	1º Iteração

Caso de Teste:	CT 002 - Usuário não consegue cadastrar abastecimento
Pré - condições :	Está na tela abastecimento
Procedimentos:	3) o ator não consegue cadastrar o abastecimento 4) o sistema não retorna mensagem de error para o usuário
Resultado esperado:	Mensagem de erro informando para o Usuário que ele não consegue cadastrar abastecimento.
Dado de entrada:	Ausência de mensagem
Prioridade	Alta
Ambiente	Android
Técnica	Manual
Iteração	1º Iteração

Caso de Teste:	CT 003 - Erro ao acessar relatório
Pré - condições :	O usuário
Procedimentos:	1) o ator não consegue acessar os relatórios de abastecimento 2) o sistema não retorna mensagem de erro para o usuário
Resultado esperado:	Mensagem de erro informando para o Usuário que ele não consegue acessar os relatórios
Dado de entrada:	Ausência de mensagem
Prioridade	Alta
Ambiente	Android
Técnica	Manual
Iteração	1º Iteração

Relatório de teste

1 . Introdução

Android Espresso

Use o Espresso para programar testes concisos, bonitos e confiáveis para a IU do Android. O snippet de código a seguir mostra um exemplo de teste do Espresso:

A API principal é pequena, previsível e fácil de aprender. Mesmo assim, ela também pode ser personalizada. O Espresso testa com clareza expectativas, interações e declarações de estado, sem que a distração causa por conteúdo clichê, infraestrutura personalizada ou detalhes confusos de implementação atrapalhe.

Os testes do Espresso são executados de maneira idealmente rápida! Ele permite ignorar suas esperas, sincronizações, suspensões e pesquisas enquanto faz manipulações e declarações na IU do aplicativo quando ela está em repouso.

Recursos de sincronização

Sempre que seu teste invoca `onView()`, o Espresso aguarda o cumprimento das seguintes condições de sincronização para realizar a ação ou a declaração de IU correspondente:

- A fila de mensagens está vazia.
- Não há instâncias de `AsyncTask` executando uma tarefa no momento.
- Todos os recursos de inatividade definidos pelo desenvolvedor estão inativos.

Realizando essas verificações, o Espresso aumenta significativamente a probabilidade de que apenas uma ação ou declaração de IU ocorra a qualquer momento. Esse recurso oferece resultados de testes mais confiáveis e seguros.

2 . Materiais e métodos

Os teste do espresso são usados no android studio , criando métodos e funções para testar as UI.