**Testes de Usuário**

* **TestesViewLogin:** Classe na qual serão focados os testes acerca dos dados dos Templates utilizados.
  + **test\_resposta\_de\_status\_a\_registrar:** Neste teste é avaliado o status da página de registro de usuário em seu acesso. Se o status do código recebido for 200, significa que o teste teve êxito.
  + **test\_status\_pagamento\_em\_espera:** Teste que olha o status do pagamento e avalia se este está como não pago em caso de pagamento ainda não efetuado, retornando True.
  + **test\_calcular\_valor\_correto\_de\_quando\_se\_utilizar\_um\_cupom\_promocional\_individual:** Avalia se ao efetuar um pagamento de dado valor para um determinado evento que tenha um cupom promocional, este pagamento cobrirá o preço do evento descontado o a porcentagem do cupom.
* **TestesViewInicio:** Classe na qual serão focados os testes acerca dos primeiros contatos com o usuário com as páginas iniciais da plataforma.
  + **test\_resposta\_de\_status\_a\_pagina\_inicial:** Neste teste é avaliado o sinal recebido pela página inicial de login, que deverá ter seu código avaliado e caso este seja igual a 302, comprovará que a página recebeu com êxito seu código. Consulte a sessão mais abaixo para compreender melhor sobre os tipos de código de sinal que há nos templates.
* **TestesPerfilUsuario:** Classe na qual serão focados os testes acerca dos dados de cadastro de usuário.
  + **test\_iniciar\_Usuario\_com\_email\_e\_senha:** Teste no qual se é avaliado se o email do usuário criado permanece sendo válido, retornando falso caso algum dado do mesmo tenha sido alterado ou perdido.
  + **test\_criacao\_Lista\_Vazia\_de\_Eventos\_para\_Usuario\_recem\_criado:** Certifica-se que um usuáriorecém criado de forma direta tenha consigo uma lista vazia de eventos em seu histórico, confirmando através do assertEqual que o tamanho da lista é igual a 0.
  + **test\_criacao\_Lista\_Vazia\_de\_Inscricoes\_para\_Usuario\_recem\_criado:** Faz algo semelhante ao caso acima, mas retornando neste caso a lista de inscrições deste usuário que deverá estar sem registros.
  + **test\_nao\_permitir\_criacao\_de\_Evento\_sem\_nome:** Certifica que um Evento criado tenha seu campo nome preenchido.
  + **test\_email\_nao\_nulo:** Certifica que um usuário registrado tenha um email válido registrado.
  + **test\_username\_em\_branco:** Certifica que um usuário cadastrado tenha seu campo de nome preenchido de maneira válida.

* **TesteInscricao:** Classe na qual serão focados os testes acerca dos dados de cadastro de usuário.
  + **test\_calcular\_valor\_correto\_de\_inscricao\_dado\_um\_conjunto\_de\_atividades\_adicionadas:** Testa se o valor da inscrição está sendo atualizado conforme a adição de novas atividades a Inscrição do Usuário, adicionando o valor da atividade ao da inscrição do usuário e depois comparando se este é igual a aquele esperado.
  + **test\_nao\_aceitar\_incluir\_inscricao\_se\_o\_estado\_ja\_for\_ATIVO:** Teste que deve avaliar se o status de uma inscrição varia conforme o status do evento e do pagamento, em outras palavras, varia conforme os status de outros.

**Respostas HTTP as Views**

* **404:** Página não encontrada.
* **500:** Problemas na requisição ao server (problema de renderização de templates).
* **403:** Requisição a um Template cuja permissão está além do Usuário (acesso a páginas confidenciais).
* **400:** Requisição a um Template cujo nome é inexistente.
* **200:** Template estabilizado com sucesso na página.
* **302:** Significa que a página em questão é uma página de transição.
* **301:** Indica que a página foi movida permanentemente.