

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade de consultar a cotação das criptomoedas no app.
Requisitos que motivaram esse teste:	O usuário deve fazer consulta da cotação das 10 principais criptomoedas(maior market cap) do dia corrente.

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade de consultar a variação da cotação das criptomoedas.
Requisitos que motivaram esse teste:	RF2 - O usuário deve poder consultar a variação da cotação das criptomoedas em comparação com o dia anterior.

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade previsões no app. Testar se os dados de previsão estão sendo recebidos corretamente.
Requisitos que motivaram esse teste:	RF3 - O usuário pode ver as previsões dos próximos 5 anos de 2 criptomoedas (Bitcoin e Ripple)

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar se os dados está sendo recebidos corretamente da API CoinMarketCap.
Requisitos que motivaram esse teste:	RF4 - O aplicativo deverá se conectar a api CoinMarketCap para pegar as cotações das criptomoedas do dia corrente.

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade de previsões no app de todas moedas disponíveis
Requisitos que motivaram esse teste:	RF5 - O aplicativo deverá se conectar a api das previsões

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar se a leitura do arquivo csv está sendo feito corretamente pelo aplicativo
Requisitos que motivaram esse teste:	RF6 - O aplicativo deverá ler os dados de um arquivo com o histórico de preço das criptomoedas.

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar se as informações passadas para os gráficos e os gráficos em si estão sendo exibidos corretamente no aplicativo.
Requisitos que motivaram esse teste:	RF7 - O aplicativo deverá mostrar os dados de histórico das criptomoedas e das previsões em formato de gráfico.

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar se os dados recebidos pela API(passados pelo programa de previsão estão corretos)

Requisitos que motivaram esse teste:	RF8 - A API deverá ser capaz de ler os dados (não necessariamente um arquivo) gerados pelo programa de previsão.
---	--

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade de bucar informações de previsão com o app
Requisitos que motivaram esse teste:	RF9 - A api deverá passar informações para o app com as previsões

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade os classificadores disponíveis no app
Requisitos que motivaram esse teste:	RF10 - Deve ser testados vários classificadores para determinar qual tem melhor acurácia

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade testar as previsões de machine learning no app
Requisitos que motivaram esse teste:	RF11 - A previsão deve ser baseada em series temporais

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos

Objetivo do Teste:	Testar se o programa de machine learning está de fato passando seu resultado no formato que a API espera.
Requisitos que motivaram esse teste:	RF12 - O programa deve ser capaz de passar seus dados para a API (não necessariamente por um arquivo).

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade hospedagem do servidor na api
Requisitos que motivaram esse teste:	RF13 - O servidor deve hospedar a api

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade de hospedar o Programa de Machine Learning no servidor
Requisitos que motivaram esse teste:	RF14 - O servidor deve hospedar o Programa de Machine Learning

Tipo de Teste:	Funcional
Subtipo de Teste:	Requisitos
Objetivo do Teste:	Testar a funcionalidade disponibilidade da api pelo app
Requisitos que motivaram esse teste:	RF15 - O servidor deve deixar pública a api

Tipo de Teste:	Não funcional
Subtipo de Teste:	Disponibilidade
Objetivo do Teste:	Verificar se o app roda com internet
Requisitos que motivaram esse teste:	RNF1 - O app necessitará de internet

Tipo de Teste:	Não funcional
Subtipo de Teste:	Usabilidade
Objetivo do Teste:	Testar se o usuário pode navegar via interface responsiva
Requisitos que motivaram esse teste:	RNF2 - Uso de Design responsivo nas interfaces gráficas

Tipo de Teste:	Não funcional
Subtipo de Teste:	Desempenho
Objetivo do Teste:	Testar se o app demora mais de 30 segundos para se conectar a api
Requisitos que motivaram esse teste:	RNF3 - Tempo limite para processamento e conexão entre api e app não deve exceder 30 segundos

Tipo de Teste:	Não funcional
Subtipo de Teste:	Desempenho
Objetivo do Teste:	Testar o processamento de machine learning não excede 1 dia
Requisitos que motivaram esse teste:	RNF4 - Tempo limite para processamento do programa Machine Learning não deve exceder 1 hora

Tipo de Teste:	Não funcional
-----------------------	---------------

Subtipo de Teste:	Padrão
Objetivo do Teste:	Verificar se o modelo está realmente em MVC
Requisitos que motivaram esse teste:	RNF5 - Divisão arquitetural do sistema em camadas para desacoplamento em modelo MVC

Tipo de Teste:	Não funcional
Subtipo de Teste:	Implementação
Objetivo do Teste:	Verificar se o sistema está sendo codicado em Swift
Requisitos que motivaram esse teste:	RNF6 - O app deverá ser feito em swift

Tipo de Teste:	Não funcional
Subtipo de Teste:	Implementação
Objetivo do Teste:	Verificar se o app roda em iphone 7 e superiores
Requisitos que motivaram esse teste:	RNF7 - O app deverá rodar em iphone 7 ou superior

Tipo de Teste:	Não funcional
Subtipo de Teste:	Implementação
Objetivo do Teste:	Verificar se o app roda em ios 12
Requisitos que motivaram esse teste:	RNF8 - O app deverá rodar em ios12

Tipo de Teste:	Não funcional
Subtipo de Teste:	Implementação
Objetivo do Teste:	Verificar se o sistema está sendo codicado em python com a biblioteca Flask
Requisitos que motivaram esse teste:	RNF9 - A api deverá ser feita em python usando o Flask

Tipo de Teste:	Não funcional
Subtipo de Teste:	Implementação
Objetivo do Teste:	Verificar se o sistema está sendo codicado em Python utilizando as bibliotecas solicitadas.
Requisitos que motivaram esse teste:	RNF10 - O programa de machine learning deve ser feito em python usando o pandas e o numpy

Tipo de Teste:	Não funcional
Subtipo de Teste:	Implementação
Objetivo do Teste:	Verificar se é o JSON o arquivo de comunicação entre api e app
Requisitos que motivaram esse teste:	RNF11 - A transferência de informação entre o app e api deverá ser feita pelo JSON