

Universidade Federal Fluminense
Instituto de Computação

ATM - ES2

Engenharia de Software II

Prof. Leonardo Gresta Paulino Murta

Grupo 2

Integrantes

- Augusto Cezar Souza Sales
- Gustavo Dauer
- Igor Gonçalves
- Jorge da Silva
- Matheus Froes

Ferramentas e técnicas de teste usadas

Para fazer os teste unitários, utilizamos JUnit com Mockito. Criamos poucos testes unitários porque o grupo teve dificuldade a início com o uso de stubs, e, além disso, não sabíamos como encaixar os testes unitários à forma de programação que já tínhamos feito no projeto. Adicionalmente, fizemos alguns cenários com o Selenium IDE, testando as operações mais básicas e limites da aplicação.

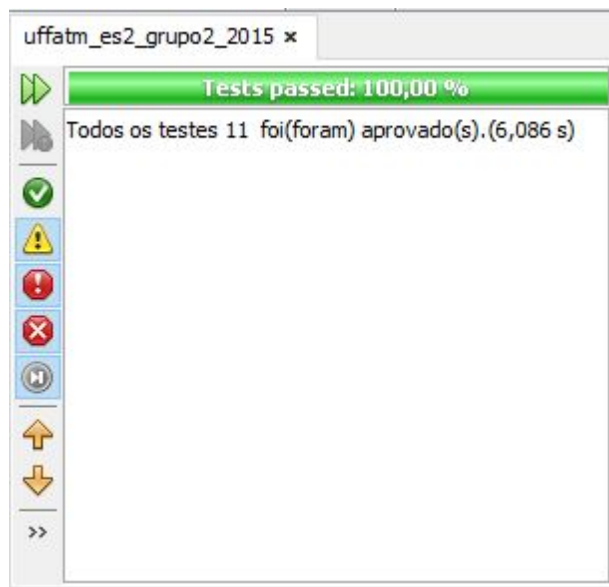
Foi feito uso do Profiler do Netbeans também.

Casos de teste e resultados da execução

Testes unitários:

- Inserção de Cliente
- Depósito Válido
- Depósito Inválido
- Pagamento com Saldo Suficiente
- Pagamento com Saldo Insuficiente
- Saque sem Limite Insuficiente
- Saque com Limite Insuficiente
- Saque sem Notas Disponíveis
- Saque com Notas Indisponíveis
- Transferência com Saldo Insuficiente
- Transferência com Saldo Suficiente

Resultados:



Testes de Sistema com Selenium IDE:

- Login Inválido
- Depósito
- Depósito feito por Não Correntista
- Transferência
- Pagamento
- Aplicação na Poupança
- Saque
- Teste de Limites do Cheque Especial

Resultados:

The screenshot shows the Selenium IDE 2.9.1 interface. The title bar is 'deposito (TestesCenarios) - Selenium IDE 2.9.1'. The menu bar includes File, Edit, Ações, Opções, and Help. The URL Base is 'http://localhost:8080/'. The Test Case list on the left includes 'deposito', 'pagamento', 'transferencia', 'depositoNaoCorren...', 'aplicacaoPoupanca', 'saque', 'testaLimite', and 'loginInvalido'. The main area displays a table of commands and their targets/values.

Comando	Alvo	Valor
open	/uffatm_es2_grupo2_2015/	
select	name=id	label=Caixa Eletrônico 1
type	name=numeroCartao	1
type	name=senha	123
clickAndWait	css=input.botao	
clickAndWait	link=Consulta saldo	
storeText	css=label	saldoString
storeEval	storedVars['saldoString'].match(/^\d+/);	saldo
clickAndWait	link=Depósito	
store	200	depositoString
type	name=valor	\${depositoString}
storeEval	storedVars['depositoString'].match(/^\d+/);	deposito
clickAndWait	css=input.botao	
storeText	css=label	saldoString
storeEval	storedVars['saldoString'].match(/^\d+/);	saldoFinal
echo	\${saldo},\${deposito},\${saldoFinal}	
verifyEval	\${saldoFinal} == \${saldo}+\${deposito}	
clickAndWait	link=Log Out	true

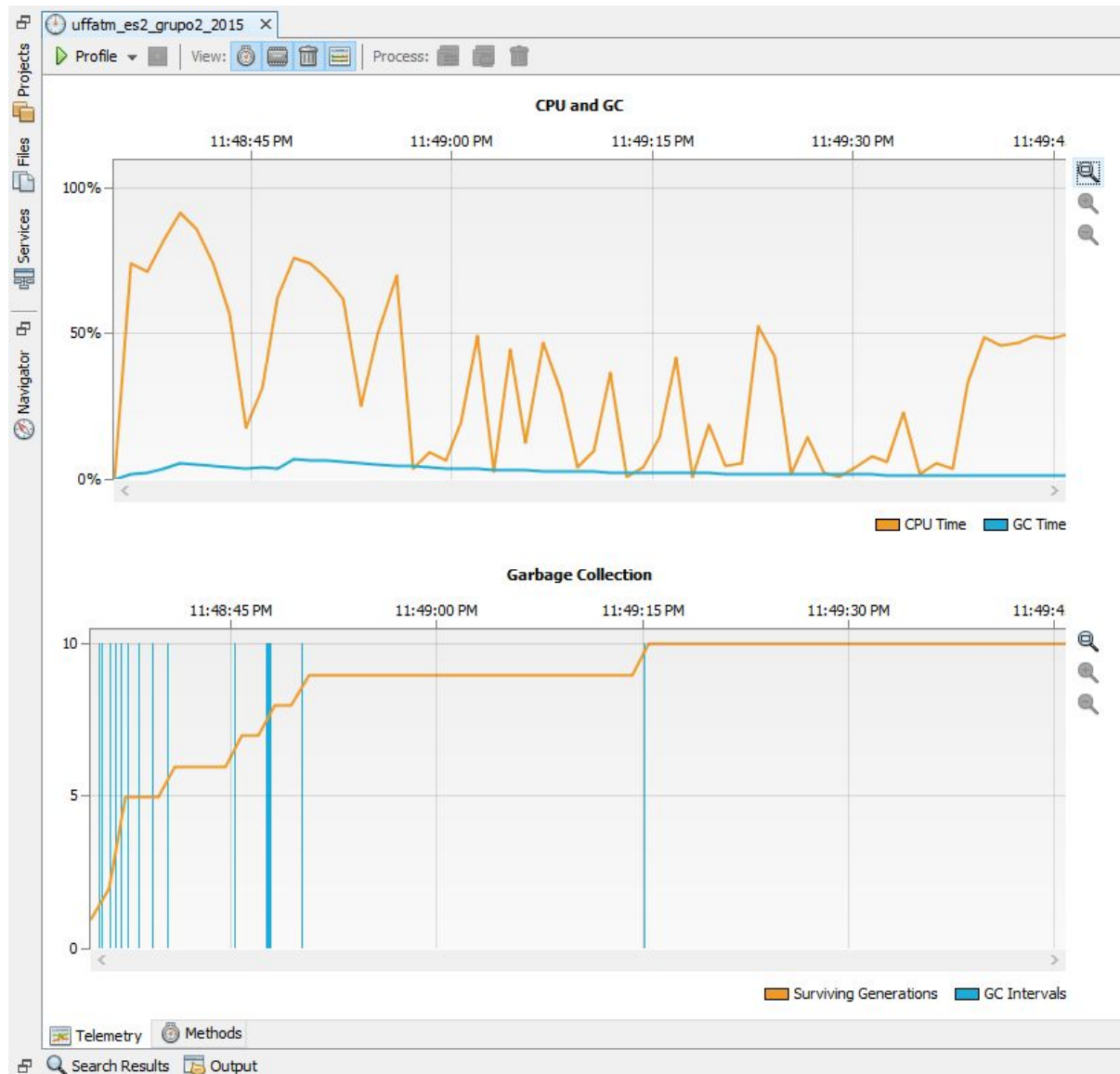
Below the table, there are input fields for 'Comando', 'Alvo', and 'Valor', along with 'Select' and 'Procurar' buttons.

The bottom section of the interface shows the 'Reference' tab for the 'clickAndWait(locator)' command. It states: 'Generated from click(locator)'. Arguments: 'locator - an element locator'. Description: 'Clicks on a link, button, checkbox or radio button. If the click action causes a new page to load (like a link usually does), call waitForPageToLoad.'

Os casos de teste do Selenium podem ser encontrados na pasta "test" no repositório.

Foi feito um teste de utilização de recursos com Profiler do Netbeans. Após iniciado, foi feita uma sequência de todas as ações disponíveis no projeto.

Resultado do Profiler:





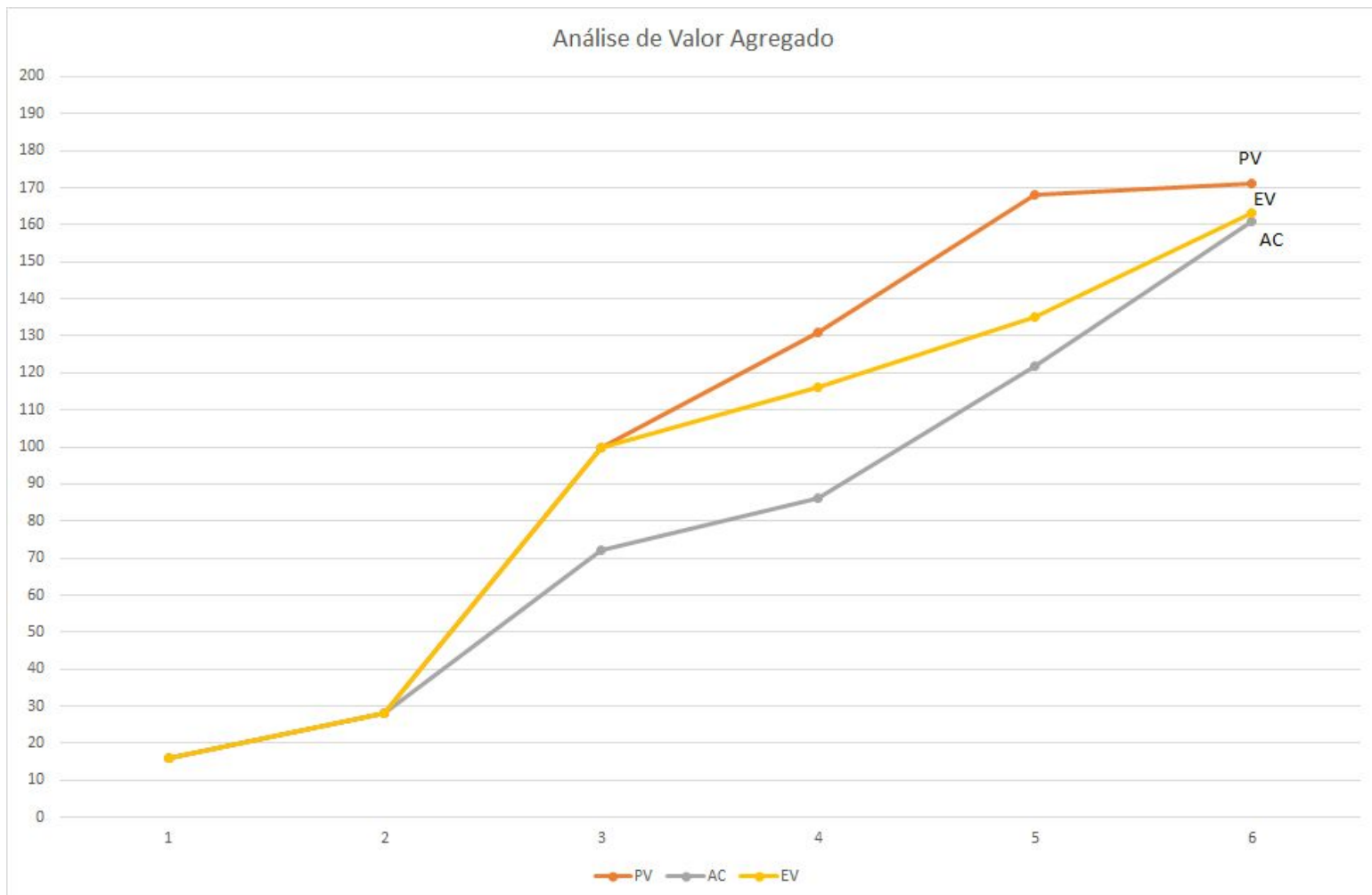
Dados de monitoramento e controle do projeto

Nossa quinta sprint terminou uma semana depois da segunda apresentação, porém os resultados não melhoraram, já que os integrantes do grupo também focaram em outros trabalhos e provas.



A sexta e última sprint mostrou um melhor desempenho em relação às anteriores, porém também não foi o ideal, já que cortamos algumas funcionalidades que consideramos menos importantes e/ou que consideramos mais difíceis para conseguirmos entregar no prazo estabelecido.

O que nos leva ao nosso gráfico de valor agregado, de onde tiramos as seguintes informações: SPI - 0,9532 & CPI - 1,0124. Significando que, similarmente a sprint anterior, continuamos atrasados, porém abaixo do custo previsto. O valor agregado do projeto não alcançou o valor planejado porque, por motivos de gerenciamento de tempo, decidimos remover as funcionalidades de validação de cheque e validação de depósito.



Dificuldades encontradas e decisões tomadas

Uma decisão que tivemos foi a de retirar as funcionalidades “Validar Depósito” e “Validar Cheque”, pois estávamos em desacordo de como fazê-las funcionar e achamos que demandaria mais tempo do que o previsto.

Percebemos também que nossas atividades descritas no WBS estavam com granularidade alta e não consideramos a intenção de teste unitário para cada funcionalidade na época de sua criação.

Testes unitários foram uma grande dificuldade também, pois o estilo de programação que estávamos fazendo dificultava a criação de testes unitários com JUnit. E, além disso, o grupo teve dificuldades com o uso de stubs, porém conseguimos criar alguns depois de algum esforço e melhoramos o entendimento sobre essas ferramentas.

Também planejamos o uso das issues disponíveis no Bitbucket, porém infelizmente não conseguimos seguir adiante.

Repositório contendo artefatos e como acessar

O nosso repositório no Bitbucket pode ser acessado por aqui:

https://bitbucket.org/asales/uffatm_es2_grupo2_2015.2

Foi criada uma máquina virtual do nosso sistema, que pode ser baixada pelos links:

Pasta dropbox: <https://goo.gl/19cEGD>

Arquivo da VM: <https://goo.gl/aXDCgV>