Plano de Teste e Registro

Aula 06 — Mostrar o que foi testado

Aula 06 · 12/11/2026 · Indicadores 1, 2, 3

UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais

O mapa de hoje

- **Bloco 1** — os tipos de teste que faltam, e quantos de cada
- **Bloco 2** — o plano de teste e o caso de teste
- **Bloco 3** — registrar, relatar o defeito e tabular o resultado
- **Bloco 4** — mão na massa: o plano e o registro do sistema de vocês

> Tragam para a tela os testes das Aulas 01 a 05 — os arquivos e os papéis com as listas de classes. Hoje eles viram documento.

Retomando a Aula 05

- Por que o teste do `buscar_por_titulo` com dublê não servia?
- Por que um banco `_teste` separado, e não o do projeto?
- O `rollback` no fim da fixture não limpou o banco. Por quê?

Bloco 1

Não há mais nenhum tipo de teste?

"Está testado?"

Vocês mostram ao dono da locadora: *"42 testes, todos verdes."* Ele pergunta:

- "No sábado à tarde, com as três atendentes alugando ao mesmo tempo, o sistema aguenta?"
- "A atendente nova vai achar o botão de devolução sem me ligar?"
- "E se alguém digitar uma aspa no nome do cliente?"

Algum dos 42 testes responde alguma dessas perguntas?

Os níveis que vocês já fizeram — e os que faltam

Nível	O que exercita	Onde vocês viram
unidade	uma regra, sozinha	`RelatorioDoMes`, `multa` — Aulas 01 a 03
unidade com dublê	a regra, com o que é de fora substituído	`CobradorDeAtrasos` — Aula 04
integração	o código conversando com o banco de verdade	`crud_jogo` — Aula 05
sistema	o sistema inteiro, pela tela, como o usuário usa	—
serviço web	o sistema respondendo a outro sistema	—

GLOSSÁRIO

`system test` / `end-to-end` — teste de sistema — de ponta a ponta, pela tela

E os que não são sobre "dá o resultado certo"

Tipo	A pergunta	Na locadora
carga	aguenta muita gente ao mesmo tempo?	sábado à tarde, três atendentes
usabilidade	uma pessoa consegue usar sem ajuda?	a atendente nova e a devolução
segurança	dá para fazer o sistema obedecer quem não devia?	a aspa no nome — o `%s` da Aula 15 da UC2

- Cada um tem ferramenta própria, e nenhum é assunto desta UC.
- Mas a forma de pensar é a mesma de hoje: **cenário, ação, validação**.

Quantos de cada?

Nível	Velocidade	Quebra à toa?	Quantos
unidade	milissegundos	quase nunca	**muitos** — toda regra, todo `if`
integração	décimos de segundo	às vezes (banco fora do ar)	um por operação com o banco
sistema	segundos ou minutos	muito (a tela mudou de lugar)	**poucos** — só o que não pode falhar

- Cem testes de sistema transformam um retorno de segundos num retorno de horas.
- Desenhado, isso vira uma **pirâmide**: base larga de unidade, ponta fina de sistema. O desenho é nosso; o livro fala em equilibrar.

O teste manual não acabou

- Automatizado é o teste que **se repete**: a máquina repete melhor e mais barato.
- Mas tem um teste que só uma pessoa faz: tentar **quebrar** o sistema, sem roteiro, desconfiando de tudo.
- Esse é o **teste exploratório**. Hoje, no Bloco 3, vocês fazem um.

GLOSSÁRIO

`exploratory testing`	teste exploratório — investigar o sistema para achar o que ninguém previu

Checagem do Bloco 1

Em que nível ou tipo está cada teste?

- `multa(3) == 15.0`
- o `CobradorDeAtrasos` com repositório e avisador falsos
- `criar_jogo` gravando no `locadora_teste`
- abrir o sistema, alugar, devolver com 4 dias de atraso e conferir a multa na tela
- 300 aluguéis ao mesmo tempo, cada um em menos de 1 segundo
- a atendente nova acha a devolução sem ajuda?



`forms.gle/SFhwFhFpYXsy6wby5`

Intervalo

10 minutos

Bloco 2

O plano de teste

O que vocês já têm

- Listas de classes no papel — Aula 01
- Fronteiras e o que tem de ser recusado — Aula 02
- A tabela do ciclo do TDD — Aula 03
- Testes com dublê, testes no banco — Aulas 04 e 05

Tudo existe. Mas está espalhado em comentário de código e folha de caderno. Um colega novo no grupo, o professor, ou o dono da locadora: quem consegue ler isso e dizer **o que foi testado**?

O plano de teste responde cinco perguntas

Pergunta	Na locadora
O que será testado — e o que fica de fora?	as regras do balcão e o CRUD; a tela fica de fora
Como — em que níveis?	unidade nas regras, integração no CRUD
Com que dados?	o banco `locadora_teste`, limpo antes de cada teste
O que é sucesso?	todos os casos verdes, duas vezes seguidas
Quem e quando?	o grupo, a cada mudança no código

GLOSSÁRIO

`test plan` plano de teste — o que, como, com o quê e quando testar

O caso de teste

Cada linha do plano vira um caso, escrito para **qualquer pessoa** conseguir executar:

Campo	CT-03
função / regra	`multa` — "a partir do 3º dia, R$ 5,00 por dia"
pré-condição	nenhuma
passos	chamar `multa` com o atraso
entrada	3 dias
resultado esperado	R$ 15,00

GLOSSÁRIO

`test case` caso de teste — uma entrada, os passos e o resultado esperado

⚠ Erro comum — o esperado copiado do código

> "Resultado esperado: o que o código devolver" não é resultado esperado. Se o esperado sai do código, o caso de teste concorda com o defeito — e fica verde.

- O esperado sai da **regra**: o que o dono disse, o que o enunciado pede, o que o bom senso do balcão espera.
- Foi assim no `multa.py` da Aula 02: o comentário no topo tinha a regra; o código tinha o `>`.

Checagem do Bloco 2

Complete o caso de teste:

Campo	CT-07
função	`deletar_jogo`
pré-condição	?
passos	?
entrada	o id do Zelda
resultado esperado	?

Por que a pré-condição deste caso é a parte mais importante dele?



forms.gle/ms8tUAzRfPjoSR649

Intervalo

10 minutos

Bloco 3

Registrar e tabular

Rodou. E agora?

O caso de teste ganha quatro colunas quando é executado:

Caso	Obtido	Situação	Data · quem
CT-01 · maior e menor, crescente	40 e 25	passou	12/11 · grupo 3
CT-03 · multa no 3º dia	R$ 0,00	**falhou**	12/11 · grupo 3
CT-07 · apagar jogo com plataforma	`False`	**falhou**	12/11 · grupo 3
CT-09 · tela de devolução	—	não executado	—

- "Falhou" não é vergonha: é o teste fazendo o trabalho dele.
- "Não executado" também se registra. Plano que esconde o que ficou de fora mente.

O relatório de defeito

Campo	DF-02
título	Busca por título nunca encontra o jogo
caso	CT-05
passos para reproduzir	criar "Zelda"; chamar `buscar_por_titulo(conexao, "Zelda")`
esperado × obtido	o Zelda × `None`, e no painel: *column "titlo" does not exist*
severidade	alta — a busca do balcão não funciona
ação corretiva	`titlo` → `titulo` no `SELECT`; CT-05 verde

GLOSSÁRIO

`bug report`	relatório de defeito — o suficiente para outra pessoa ver o defeito acontecer
`severity`	severidade — o tamanho do estrago para quem usa

⚠ Erro comum — "não funciona"

> "A busca não funciona" obriga quem vai consertar a descobrir sozinho **o que** não funciona, **com que** dado e **como** fazer acontecer de novo. Metade das vezes, a pessoa não consegue — e o defeito fica.

Antes de mandar, a pergunta: alguém que nunca viu o sistema consegue fazer o defeito aparecer **só com o que eu escrevi**?

Tabular — o que o dono lê

Situação	Casos
passou	15
falhou	3
não executado	1
total	**19**

Severidade dos defeitos	Abertos	Corrigidos
alta	0	2
média	1	0

Duas tabelas, e o dono sabe o que foi testado, o que quebrou, e o que já foi consertado — sem abrir um arquivo `.py`.

Exploratório — 15 minutos

Abram o `locadora-crud.ipynb` de exemplo da UC2, no banco `locadora_demo`, e **tentem quebrar**.

1. Uma frente por dupla: cadastro de cliente, aluguel, ou devolução
2. Tentem o que ninguém tentaria: campo vazio, aspas, data no passado, número negativo, o mesmo jogo duas vezes
3. Achou algo estranho? Repita, para ter certeza de que sabe fazer acontecer de novo
4. Registre na aba **Defeitos** da planilha modelo, com os passos

> O sistema é o do professor — achar defeito nele é exatamente o objetivo.

Checagem do Bloco 3

Um colega abriu este defeito:

> *"O sistema não deixa apagar jogo. Arrumem."*

- Reescreva como relatório de defeito, com título, passos para reproduzir, esperado × obtido e severidade
- Que pergunta você faria a ele antes de conseguir escrever os passos?



forms.gle/FG5MJxkHGjpq9G8L9

Intervalo

10 minutos

MÃO NA MASSA

O plano e o registro do sistema de vocês

Na `plano-de-teste-modelo.xlsx` do site:

1. **Plano:** as cinco perguntas, respondidas para o sistema do grupo
2. **Casos de teste:** um por teste que vocês escreveram nas Aulas 01 a 05 — o esperado sai da regra, não do código
3. Rodem a suíte inteira e preencham **obtido**, **situação**, data e quem
4. **Defeitos:** todo vermelho vira relatório, com a ação corretiva de quem já consertou
5. A aba **Resumo** conta sozinha. Confiram se ela bate com o painel Testing

MÃO NA MASSA

O combinado desta entrega

GLOSSÁRIO

`grupoNN-aulaNN-artefato.<ext>` `grupo03-aula06-plano-de-teste.xlsx`

O que eu confiro: O plano diz o que fica **de fora** · Cada caso tem esperado tirado da **regra** · A situação de cada caso bate com o **painel Testing** · Todo vermelho tem relatório de defeito que **outra pessoa** consegue reproduzir · Os defeitos corrigidos têm a **ação corretiva** escrita

> Eu passo de mesa em mesa. Vou escolher um caso da planilha e pedir para vocês o executarem na minha frente, seguindo só o que está escrito.

PARA CASA · OPCIONAL

Antes da próxima aula

Testar em casa: termine o registro e rode a suíte uma última vez. O resumo tem de bater com o painel.

Tarefa opcional: escolha um defeito aberto da planilha e conserte. Antes de mexer, copie o arquivo inteiro para outra pasta — guarde a versão que funcionava.

Para treinar: `plano-de-teste-modelo.xlsx` está no site, na página da aula.

Para ver acontecer: demonstrações da Aula 06 — pôr seis testes no nível certo, e consertar um relatório de defeito até o Bruno conseguir reproduzir — e oito perguntas com resposta na hora.

Para a próxima aula

O teste achou o defeito, vocês consertaram — e quebrou outra coisa. Como voltar à versão que **funcionava**?

UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais