Projeto Integrador

Aula 01 — O problema que o sistema de vocês resolve

Aula 01 · 18/11/2026 · Indicadores 1, 3

UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais

A UC4 numa tela

Indicador	O que vocês vão mostrar
1	Cumprir as atividades do **plano de ação** que o próprio grupo fez
2	Apresentar uma **solução** para o problema que o grupo escolheu
3	Trabalhar com as **Marcas Formativas**: colaboração, visão crítica, autonomia digital...

- **Cinco aulas:** hoje o problema e o plano; 02 e 03 organizar e integrar; 04 validar e fechar a versão; 05 a demonstração
- Não há sistema novo: é o de vocês — o CRUD da UC2, os testes da UC3 e uma interface — **num projeto só**

O mapa de hoje

- **Bloco 1** — o problema do negócio que o módulo de vocês vai resolver
- **Bloco 2** — o problema virando histórias, e as histórias virando issues
- **Bloco 3** — o plano de ação: quem faz o quê, até quando
- **Bloco 4** — mão na massa: o plano do grupo, no GitHub

> Tragam para a tela o repositório da UC3, o banco do projeto no pgAdmin e o notebook do CRUD da UC2. Hoje a gente olha tudo junto.

Retomando a UC3

- O que o repositório de vocês tem hoje — e o que ainda está fora dele?
- Uma coisa que o dono do negócio pediria na próxima versão
- Se uma pessoa nova entrasse no grupo amanhã, por qual arquivo ela começaria?

Bloco 1

O problema, antes do sistema

Três UCs, três lugares

O que vocês têm	Onde está
o CRUD, com o menu de `input()`	um notebook, da UC2
os testes, o `conftest.py`, o plano de teste	o repositório da UC3
o banco e o backup	o PostgreSQL e um arquivo `.backup`
uma interface	a UC1 — se ainda existir em algum lugar

- Cada parte funciona sozinha. Juntas, ainda não são um **sistema**.
- E nenhuma delas começou por uma pergunta do negócio: começaram pelo conteúdo de cada UC.

O que tira o sono do dono?

Na locadora, numa sexta à noite, o dono reclama:

- *"Não sei quem está com jogo atrasado sem abrir o caderno."*
- *"Comprei dez cópias do FIFA e não sei se alugam."*
- *"Cliente novo pergunta o preço e eu tenho de olhar três telas."*

Qual dessas o sistema de hoje já resolve? E qual delas um **módulo** novo resolveria em quatro aulas?

O Projeto Integrador

O plano do Senac chama de **situação-problema**: uma demanda de **organização e controle** de um negócio, resolvida por um módulo do sistema.

1. **Problematização** — hoje: o problema, as perguntas, o plano
2. **Desenvolvimento** — Aulas 02 a 04: organizar, integrar, validar
3. **Síntese** — Aula 05: o que ficou pronto, e o que o processo ensinou

> O problema é do **negócio do grupo**, o mesmo minimundo desde a UC2. A locadora é o exemplo que a gente faz junto.

Um problema bem escrito

Pergunta	Na locadora
qual é o problema?	o dono não sabe quem está com jogo atrasado
quem sofre com ele?	o dono e a atendente, no balcão
como é hoje?	um caderno, folheado toda sexta
como saber que resolveu?	a atendente vê na tela os atrasados e a multa de cada um, em segundos

A última linha é a mais importante: é ela que vira o teste de aceitação no fim.

Checagem do Bloco 1

Para o negócio do **seu** grupo:

- Qual problema de organização ou controle o módulo vai resolver?
- Quem sofre com ele, e como é feito hoje?
- Como vocês vão saber, no dia 25/11, que ele foi resolvido?



forms.gle/JPTfM2mf5MzWiJnt5

Intervalo

10 minutos

Bloco 2

O problema virando trabalho

A história do usuário

Um jeito curto de escrever o que o sistema deve fazer, do ponto de vista de quem usa:

> **Como** atendente, **quero** ver os empréstimos atrasados e a multa de cada um, **para** cobrar no balcão sem abrir o caderno.

- Cabe num cartão. Se não cabe, é grande demais — divide.
- Vem com os **critérios de aceitação**: o que tem de acontecer para a história estar pronta.

GLOSSÁRIO

`user story`	história do usuário — uma necessidade, na voz de quem usa
`acceptance criteria`	critérios de aceitação — como se sabe que ficou pronto

O critério de aceitação já é um teste

História: ver os atrasados e a multa.

Pronta quando:

- empréstimo devolvido **não** aparece
- empréstimo com 2 dias de atraso aparece com multa **R$ 0,00**
- empréstimo com 3 dias aparece com **R$ 15,00** — a fronteira da UC3
- sem nenhum atrasado, a tela diz **"nenhum atraso"**, e não fica em branco

Cada linha dessas vira um teste no `pytest`. Quem escreve o critério está escrevendo a suíte.

Uma história, três camadas

Camada	A tarefa da história "ver os atrasados"
regra	calcular a multa — a `multa()` da UC3 já existe
dados	a consulta dos empréstimos abertos com prazo vencido
interface	a tela com a lista e o total

- Uma tarefa por camada vira uma **issue** no GitHub, com o rótulo da camada.
- E já dá para dividir: três pessoas, três tarefas, a mesma história.

⚠ Erro comum — a história do tamanho do sistema

Assim, não

- *"Fazer o sistema da locadora"*
- *"Melhorar o banco"*
- *"Interface bonita"*

Assim, sim

- *"Como atendente, quero cadastrar um jogo na tela, para não depender do notebook"*
- *"Como dono, quero ver quantas vezes cada jogo foi alugado no mês"*

"Fazer o sistema" não tem critério de aceitação e não cabe em quatro aulas. Uma história boa termina numa aula — duas, no máximo.

Checagem do Bloco 2

- Escreva **uma** história do problema do seu grupo, no formato *como… quero… para…*
- Escreva três critérios de aceitação para ela — um deles, um caso especial
- Divida a história em tarefas, uma por camada

forms.gle/Ly4SZ6HyXhQvTZgA7

Intervalo

10 minutos

Bloco 3

O plano de ação

O plano cabe em quatro aulas

Aula	O que fica pronto	Quem
02 · 19/11	o projeto organizado em camadas, a suíte verde	todos
03 · 23/11	a interface ligada ao banco, na primeira história	Ana · Bruno
04 · 24/11	as outras histórias, a validação, a versão `v2.0.0`	Carla · Ana
05 · 25/11	a demonstração	todos

- Cada aula é um **ciclo curto**: começa com o que ficou da anterior, termina com algo que roda.
- Não deu tempo? **Corta a história, não a data.** O que entrar tem de funcionar inteiro.

Cinco minutos no começo de cada aula

Em pé, na mesa do grupo, cada pessoa responde:

1. O que eu terminei desde a última aula?
2. O que está me travando?
3. O que eu vou fazer hoje?

- Não é prestação de contas para o professor: é o grupo sabendo onde está.
- Quem trava fala **logo** — a ajuda chega na mesma aula, não na véspera da entrega.

O plano mora no GitHub

- Cada tarefa é uma **issue**: título, a história a que pertence, o critério de aceitação
- **Responsável** — o *Assignee* da issue
- **Prazo** — um *Milestone* por aula: `Aula 02`, `Aula 03`, `Aula 04`
- **Camada** — um rótulo: `regra`, `dados`, `interface`
- O conserto fecha a issue sozinho: `fixes #N` no commit, como na UC3

> O Indicador 1 é "cumprir o plano de ação". O que eu vou olhar é este quadro: o que foi planejado, o que fechou, e o que o grupo decidiu cortar — e por quê.

Checagem do Bloco 3

O plano de um grupo:

Tarefa	Quem	Prazo
fazer a interface	—	Aula 05
organizar os arquivos	Ana	Aula 05
testes	Ana	Aula 05
banco	Ana	—

- Aponte **três** problemas deste plano
- Reescreva duas linhas do jeito certo

forms.gle/HDKWcNvc9KqHckTD9

Intervalo

10 minutos

MÃO NA MASSA

O plano do grupo, no GitHub

1. No `README.md` do repositório, a seção **O problema**: as quatro perguntas do Bloco 1
2. De três a cinco **histórias**, cada uma com critérios de aceitação
3. Cada história dividida em **issues**, com rótulo de camada
4. Os milestones `Aula 02` a `Aula 04`, e cada issue com dono e milestone
5. Ninguém com tudo, ninguém sem nada: confiram o quadro antes de me chamar

MÃO NA MASSA

O combinado desta entrega

GLOSSÁRIO

o link do repositório `github.com/<conta>/grupo03-uc3`, com as issues e os milestones

O que eu confiro: O problema escrito com o critério de **como saber que resolveu** · Histórias que cabem numa aula, com critério de aceitação · Toda issue com **dono**, **milestone** e **rótulo de camada** · O trabalho dividido entre todo o grupo · Pelo menos um critério que é **caso especial**

> Eu passo de mesa em mesa e leio o quadro com vocês. É normal o plano mudar depois de hoje — o que conta é ele existir, e o grupo dizer por que mudou.

PARA CASA · OPCIONAL

Antes da próxima aula

Testar em casa: leia as histórias do grupo como se fosse o dono do negócio. Elas resolvem o problema que vocês escreveram?

Tarefa opcional: liste, numa folha, todos os arquivos do projeto que existem hoje — notebook, `.py`, `.sql`, planilha — e onde cada um está.

Para treinar: o `locadora-integrada.zip` no site é o projeto da locadora já organizado. Abram e olhem as pastas — amanhã a gente faz o mesmo com o de vocês.

Para a próxima aula

O plano está no GitHub. Mas o projeto ainda está espalhado em três lugares. Onde cada coisa nova vai morar — e como a interface conversa com o banco sem virar uma bagunça?

UniSenac Centro Universitário RS
Senac
Fecomércio
IFEP
Sesc
Sindicatos Empresariais