

# Find The Bug

## 1. Objetivo do Jogo

Vocês são uma equipe de desenvolvedores. O objetivo é identificar o módulo exato e explicar a causa raiz de um bug antes que o prazo de 5 "dias" (rodadas) se esgote.

### Cenário do Playtest:

- **Relatório do Cliente:** "Alguns usuários relatam que o saldo da conta aparece desatualizado por alguns minutos após realizarem uma compra. O valor correto só aparece depois de um tempo."
- **Arquitetura do Sistema:**  
Frontend (UI) → API Gateway → Serviço de Usuário → Banco de Dados  
(O Serviço de Usuário também se comunica com um Serviço de Cache)

## 2. A Interface do Jogo

1. **Diagrama da Arquitetura (Centro):** O mapa interativo do sistema.
2. **Painel de Recursos e Ações (Esquerda):**
  - **Pontos de Função (PF):** Seu "orçamento" de trabalho. Vocês têm 12 PF por dia.
  - **Dias Restantes:** O prazo. Começa em 5.
  - **Ações de Depuração:** Suas ferramentas de investigação.
3. **Quadro de Evidências (Direita):** Onde as pistas são coletadas e classificadas em Sintoma, Causa ou Ruído.

## 3. Como Depurar: Um "Dia" de Trabalho

### Fase 1: Planejamento e Execução

- A equipe debate e gasta seus Pontos de Função (PF) para executar as Ações de Depuração.

Ação	Custo	Como Executar	O que Revela
Inserir Log()	1 PF	Clique em uma conexão entre dois módulos.	Mostra os dados exatos que passaram por aquela conexão.

<b>Teste Unitário()</b>	2 PF	Clique em um único módulo.	Valida uma função interna daquele módulo, de forma isolada.
<b>Teste de Integração()</b>	3 PF	Selecione dois módulos conectados.	Valida se a comunicação entre os dois está funcionando como esperado.

### Fase 3: Análise

- Para cada pista descoberta, a equipe debate e a classifica no Quadro de Evidências:
  - **Sintoma:** Um efeito visível do problema, mas não a origem (ex: "Tela de erro genérica").
  - **Causa:** Uma pista que aponta diretamente para a raiz do problema (ex: "Valor recebido é nulo").
  - **Ruído:** Informação correta, mas irrelevante para este bug específico (ex: "Uso de memória em 80%").

Ao final do dia, o marcador de Dias Restantes diminui em 1. Um novo dia começa com 12 PF renovados.

## 4. Condições de Vitória e Derrota

### Para Vencer:

A qualquer momento, a equipe pode gastar 1 PF para "Declarar Solução". Para vencer, vocês precisam acertar:

1. **O Módulo Culpado:** Apontar qual módulo contém o bug.
2. **A Causa Raiz:** Explicar verbalmente como as pistas se conectam para causar o problema relatado pelo cliente.


### Para Perder:

1. O contador de Dias Restantes chegar a zero.
2. Declarar uma solução incorreta (Penalidade: perde 2 dias no prazo).

## 5. Exemplo de um Turno (Dia 1)

**Situação:** A equipe lê o relatório: "O botão 'Pagar' dá erro". O prazo é de 5 dias e eles têm 12

PF.

1. **Planejamento:** A equipe debate: "O erro acontece no Frontend, mas a causa provavelmente está mais para frente. Vamos verificar os dados que saem do Frontend para a API."
2. **Execução (Ação 1):**
  - **Ação:** Inserir Log() na conexão Frontend → API Gateway.
  - **Custo:** 1 PF. (Restam 11 PF).
  - **Pista Descoberta:** [LOG] - POST /pagar | body: { amount: 19.99 }.
3. **Análise (Pista 1):** A equipe arrasta a pista para o Quadro de Evidências. "O valor parece correto. Não vemos um problema aqui ainda. Vamos classificar como Ruído por enquanto."
4. **Execução (Ação 2):**
  - **Planejamento:** "Ok, o dado chegou na API. Vamos ver se a comunicação da API com o Serviço de Pagamentos está funcionando."
  - **Ação:** Teste de Integração() entre API Gateway e Serviço de Pagamentos.
  - **Custo:** 3 PF. (Restam 8 PF).
  - **Pista Descoberta:** [TESTE] - API → Pagamentos: FALHOU. Response: 400 Bad Request - Invalid format for 'amount' field.
5. **Análise (Pista 2):** "Opa! Encontramos algo importante. O Serviço de Pagamentos recusou o formato do campo 'amount'. Isso é uma Causa com certeza!" A equipe classifica a pista.
6. **Execução (Ação 3):**
  - **Planejamento:** "Temos quase certeza que o bug está na API Gateway, que está enviando o formato errado. Vamos só confirmar como o Serviço de Pagamentos espera receber esse dado."
  - **Ação:** Teste Unitário() no módulo Serviço de Pagamentos.
  - **Custo:** 2 PF. (Restam 6 PF).
  - **Pista Descoberta:** [TESTE] - Módulo Pagamentos: FUNÇÃO processPayment() | ENTRADA: { amount: 1999 } | RESULTADO:  OK.
7. **Análise (Pista 3):** "Confirmado! A função interna espera um número inteiro (centavos), não um decimal. A API está enviando 19.99 quando deveria enviar 1999. O bug está na API Gateway."

A equipe gastou 6 PF. Eles poderiam continuar investigando, mas já encerram o dia. O prazo muda para 4 dias restantes. No início do Dia 2, eles podem gastar 1 PF e Declarar a Solução:

- **Módulo:** API Gateway.
- **Causa Raiz:** "O Frontend envia o valor 19.99. A API Gateway repassa esse valor diretamente, mas o Serviço de Pagamentos espera receber o valor em centavos, como um inteiro (1999). A API não está convertendo o formato, causando o erro de validação."

**A equipe vence!**