

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CIÊNCIA DA COMPUTAÇÃO

SHARA LIMA

DIEGO MATHIAS

GUSTAVO HENRIQUE

LUAN COSTA

NICOLAS GABRIEL

Workshop - Associações

MEDIANEIRA

2025

Atividade 1: Reconhecendo Associações

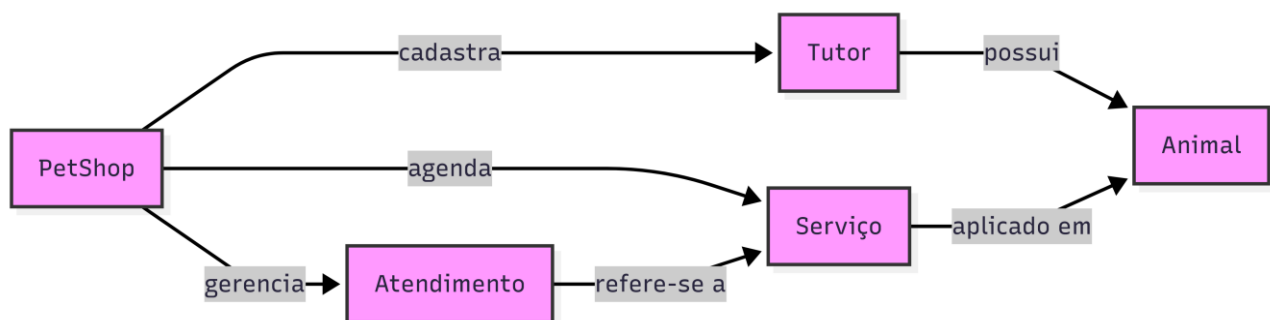
Objetivo Destacar numa narrativa curta candidatos a classes (substantivos) e pistas de vínculo (verbos/preposições), rabiscando associações iniciais.

Mini-app: Detector de Associações

História 1: [E] Pet Shop Um pet shop agenda banhos e tosas para tutores e entrega senhas de atendimento.

História	CLASSE IDENTIFICADA	Associação Identificada
1.PetShop	<p>Petshop: Entidade central do negócio, possui identidade e operações próprias.</p> <p>Banho/Tosa: Serviços oferecidos, com identidade própria (podem ser agendados, cancelados, têm preço).</p> <p>Tutor: Agente que solicita os serviços, possui dados próprios (nome, contato, pets).</p> <p>Senha/Atendimento: Senha é um identificador único de um Atendimento. A princípio, “Senha” é um atributo de “Atendimento”</p>	<p>Agenda: Verbo que indica uma associação entre PetShop e Serviço (Banho/Tosa). Essa associação provavelmente se tornará uma classe de associação "Agendamento", pois terá atributos como data, horário, status.</p> <p>Para: Preposição que liga o Serviço ao Tutor (quem é o beneficiário).</p> <p>Entrega: Ação do PetShop que resulta na criação de uma Senha (atributo) para um Atendimento (classe).</p>

Diagrama



História 2 : [E] Paraciclo Público Um paraciclo público registra entradas e saídas de bicicletas usando QR codes.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
2.Paraciclo Público	<p>Paraciclo: Entidade principal, local físico.</p> <p>Bicicleta: Entidade com identidade única (possui dono, modelo, etc.).</p> <p>QRCode: É um atributo de identificação de uma Bicicleta ou de um Evento (Entrada/Saída), não uma classe.</p>	<p>Registra: Ação do Paraciclo que cria um novo Evento (de entrada ou saída).</p> <p>Usando: Indica que o Evento utiliza o QRCode da Bicicleta para ser executado. O QRCode é um dado (atributo) necessário para a operação.</p>

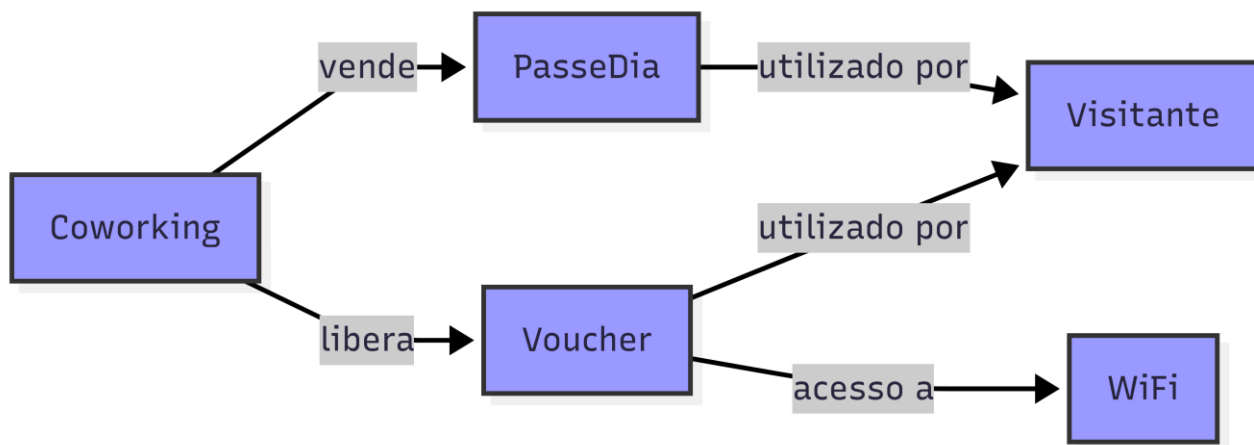
Diagrama



História 3: [E] Coworking Um coworking vende passes diários e libera vouchers de Wi-Fi para visitantes

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
3.Coworking	<p>Coworking: Entidade principal.</p> <p>PasseDiário: Um produto vendido, com identidade (data, preço, validade).</p> <p>Voucher: Um token de acesso. Pode ser uma classe com atributos (código, válido até, utilizado).</p> <p>Wi-Fi: Recurso oferecido. Pode ser inicialmente um atributo do Coworking (ex: nomeRedeWiFi), mas se tem regras complexas (senha, limite de banda), pode virar classe.</p> <p>Visitante: Usuário do serviço, com identidade.</p>	<p>Registra: Ação do Paraciclo que cria um novo Evento (de entrada ou saída).</p> <p>Usando: Indica que o Evento utiliza o QRCode da Bicicleta para ser executado. O QRCode é um dado (atributo) necessário para a operação.</p>

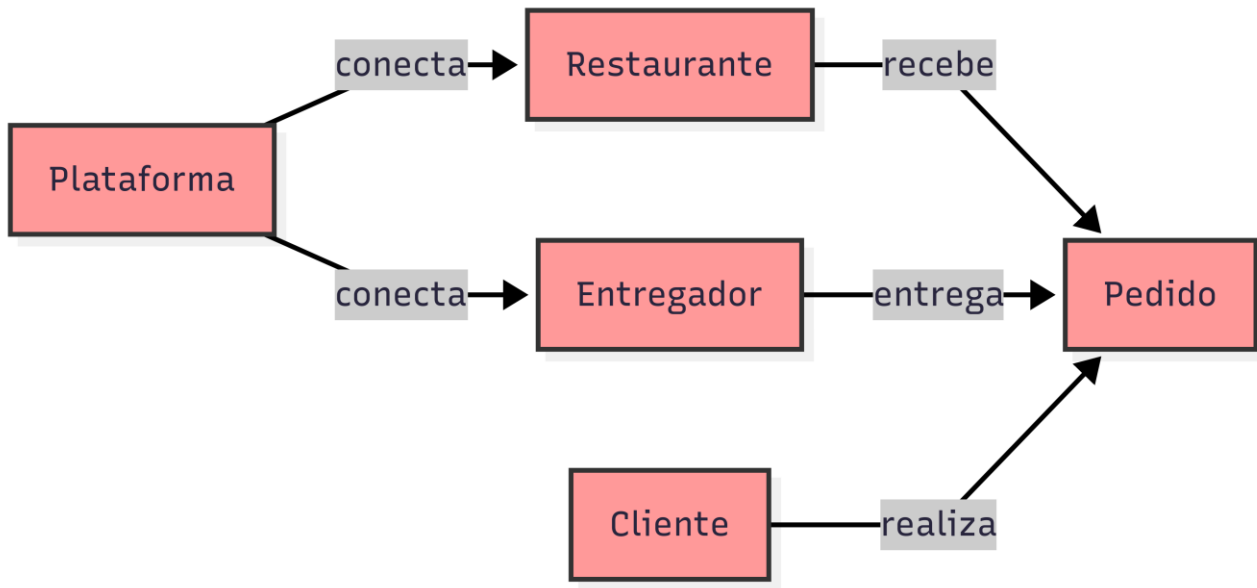
Diagrama



História 4: [I] Plataforma de Entrega Uma plataforma de entrega conecta restaurantes e entregadores por meio de pedidos.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
4.Plataforma de Entrega	<p>PlataformaEntrega: Sistema central.</p> <p>Restaurante: Entidade parceira, com identidade (cardápio, endereço).</p> <p>Entregador: Entidade parceira, com identidade (dados, veículo, disponibilidade).</p> <p>Pedido: O coração do negócio. Classe complexa com ciclo de vida próprio (criado, pago, em preparo, a caminho, entregue).</p>	<p>Conecta:A Plataforma conecta Restaurantes a Entregadores. Esta é uma associação indireta, realizada por meio de pedidos.</p> <p>Por meio de: Esta preposição é crucial. Ela indica que a associação Restaurante <- -> Entregador não é direta, mas sim mediada pela classe Pedido. O Pedido é uma classe de associação complexa entre Restaurante, Entregador e Cliente.</p>

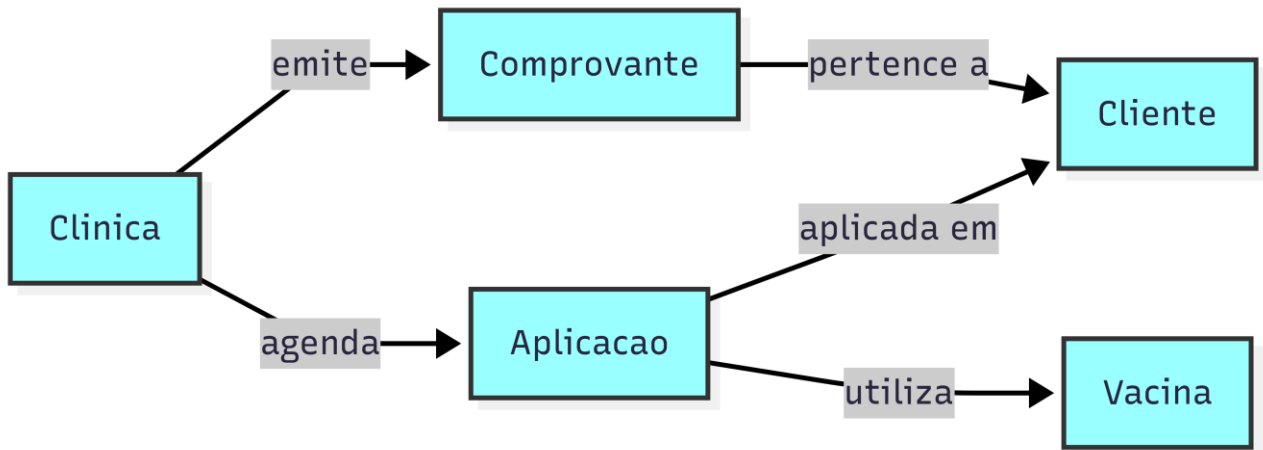
Diagrama



História 5: [I] Clínica de Vacinas Uma clínica de vacinas agenda aplicações e emite comprovantes para clientes.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
5.Clínica de Vacinas	<p>ClínicaVacinas: Entidade principal.</p> <p>Aplicação: O evento principal do negócio. Tem identidade (data, lote da vacina, profissional).</p> <p>Comprovante:O documento gerado pós-aplicação. Tem identidade (código, data emissão) e pode ser uma classe.</p> <p>Cliente: Paciente que recebe a vacina.</p>	<p>Agenda:Associação entre Clínica e Aplicação. Criará uma classe de associação "Agendamento".</p> <p>Emite: Associação entre Clínica e Comprovante. A ação de emitir gera uma instância do comprovante.</p> <p>Para: Liga a Aplicação e o Comprovante ao Cliente (o destinatário). A Aplicação é uma classe de associação entre Clínica e Cliente.</p>

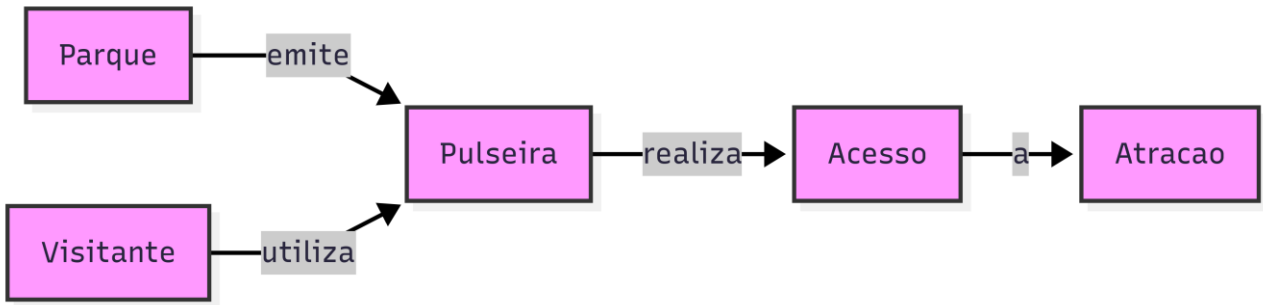
Diagrama



História 6: [I] Parque Temático Um parque temático usa pulseiras RFID para visitantes acessarem atrações.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
6.Parque Temático	<p>ParqueTemático: Entidade principal.</p> <p>PulseiraRFID: Objeto físico que permite acesso. Tem identidade única (ID).</p> <p>Visitante: Portador da pulseira.</p> <p>Atração: Recurso que pode ser acessado.</p>	<p>usa: O Parque usa o sistema de PulseiraRFID como um recurso para controlar acesso. É uma associação de agregação (o parque tem pulseiras).</p> <p>para acessarem: Esta é a finalidade. A associação real é: Visitante acessa Atração. O meio pelo qual isso acontece é a PulseiraRFID. A Pulseira é um atributo do Visitante que habilita a associação acessa.</p>

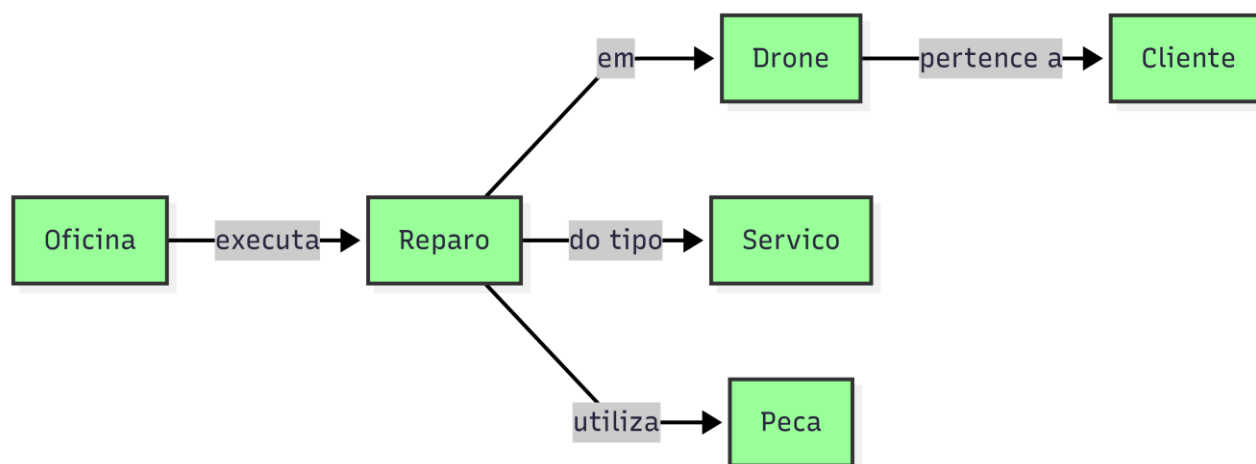
Diagrama



História 7: [I] Oficina de Drones Uma oficina de drones registra serviços e peças usadas em cada reparo.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
7.Oficina de Drones	<p>OficinaDrones: Entidade principal.</p> <p>Serviço: Um tipo de reparo oferecido (ex: "troca de hélice"). Pode ser uma classe catálogo.</p> <p>Peça: Um item em estoque usado em reparos. Tem identidade (nome, modelo, preço).</p> <p>Reparo: O evento real realizado em um drone específico. Tem identidade (data, custo, drone).</p>	<p>registra: A Oficina registra que um Serviço foi realizado. Isso cria uma instância de Reparo.</p> <p>usadas em: Indica que Peças são consumidas em um Reparo. Esta é uma associação crucial que provavelmente se tornará uma classe de associação "ItemUtilizado", com atributos como quantidade e custo naquela ordem. O Reparo em si é uma classe de associação entre Oficina, Drone (não citado) e Serviço.</p>

Diagrama

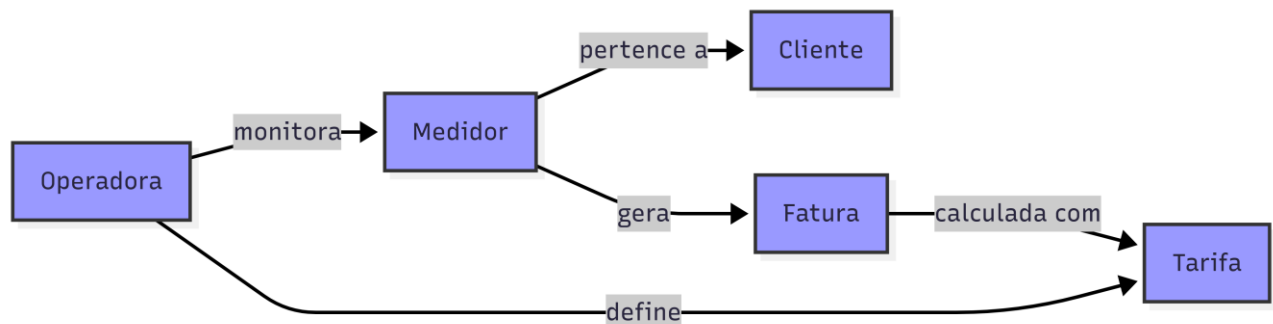


História 8: [A] Operadora de Energia Uma operadora de energia monitora medidores inteligentes e emite faturas com tarifas por faixa.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
8.Operadora de Energia	<p>OperadoraEnergia: Entidade principal.</p> <p>MedidorInteligente: Dispositivo físico com identidade única (número de série, localização).</p>	<p>monitora: A Operadora monitora Medidores. Essa associação gera os dados de consumo.</p> <p>emite: A Operadora emite Faturas com base no consumo dos Medidores.</p>

	<p>Fatura: Documento financeiro com ciclo de vida (emitida, paga, vencida).</p> <p>TarifaFaixa: Uma regra de cobrança. Tem identidade (faixa de consumo, preço por kWh).</p>	<p>com: A Fatura é calculada com base em TarifaFaixa. A fatura possui uma ou mais faixas de consumo aplicadas.</p>
--	---	---

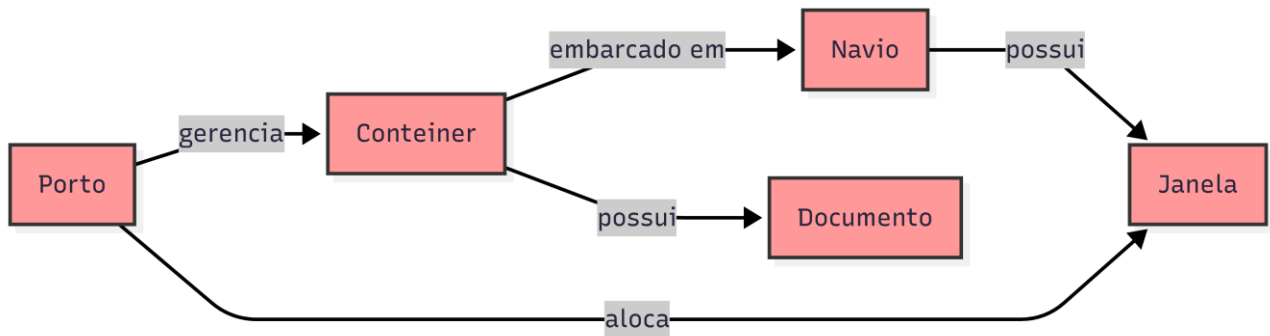
Diagrama



História 9: [A] Porto Um porto organiza contêineres em navios segundo janelas de atracação e documentos alfandegários.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
9.Porto	<p>Porto: Entidade principal.</p> <p>Contêiner: Objeto físico a ser transportado. Identidade única (número, dono, conteúdo).</p> <p>Navio: Veículo transportador.</p> <p>JanelaAtracação: Um intervalo de tempo alocado. É um recurso escasso.</p> <p>DocumentoAlfandegário: Documento associado a um contêiner.</p>	<p>organiza: Ação complexa do Porto para alocar Contêineres em Navios.</p> <p>em: Preposição que indica a associação de composição Navio --(contém)--> Contêiner. Um contêiner está em um navio.</p> <p>segundo: O carregamento é feito segundo uma JanelaAtracação. A janela é um recurso alocado para um Navio.</p>

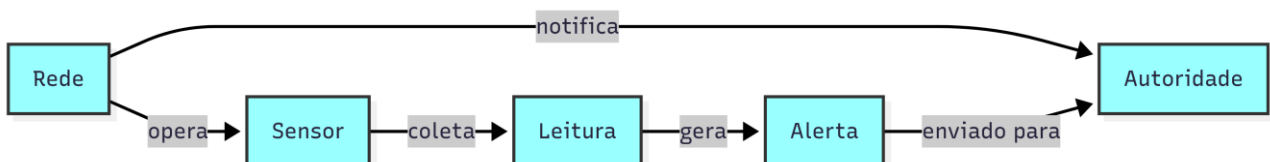
Diagrama



História 10: [A] Rede Ambiental Uma rede ambiental coleta leituras de sensores e notifica autoridades com alertas.

HISTÓRIA	CLASSE IDENTIFICADA	ASSOCIAÇÃO IDENTIFICADA
10. Rede Ambiental	<p>RedeAmbiental: Sistema central.</p> <p>LeituraSensor: Dado capturado. Tem identidade (valor, timestamp, localização).</p> <p>Autoridade: Órgão que deve ser alertado.</p> <p>Alerta: Um evento gerado a partir de leituras anômalas. Tem identidade (tipo, gravidade, status).</p>	<p>coleta: A Rede coleta LeiturasSensor continuamente.</p> <p>notifica: A Rede notifica Autoridades quando necessário.</p> <p>com: A notificação é feita com um Alerta. O Alerta é gerado a partir de uma ou mais LeiturasSensor.</p>

Diagrama



Atividade 2: Associação ou Atributo?

Objetivo: Decidir quando algo é atributo ou deve virar classe associada considerando identidade, regras, histórico e ciclo de vida

Critérios de Decisão Utilizados:

1. Varia no tempo?
2. Tem regras próprias?
3. Possui histórico?
4. Tem idade única?
5. Tem ciclo de vida independente?

Tabela de Decisões do Grupo

Situação	Decisão (A ou C)	Justificativa
1. Cor de Embalagem de um produto	A (Atributo)	É uma propriedade visual estática. Um produto não muda de cor de embalagem sozinho e não tem regras complexas associadas. Pode ser um simples campo cor: String na classe Produto.
2. Estilo de Preparo de um prato	C (Classe)	Envolve uma sequência de instruções, tempos e ingredientes. Pode ser reutilizado em vários pratos (ex: 'Gratinado', 'Ao Pontapé'). Tem regras e uma identidade própria, portanto, é uma classe.
3. Endereço de Entrega com histórico por cliente	C (Classe)	Um cliente pode ter múltiplos endereços (casa, trabalho). Cada endereço tem seu próprio histórico de entregas e pode ter regras específicas (ex: 'não entregar aos sábados'). É uma entidade com vida própria
4. Canal de Aquisição de um Cliente	A (Atributo)	É uma informação categórica simples (ex: 'orgânico', 'indicação'). Não varia para o mesmo cliente, não tem regras complexas. Pode ser um Enum (CANAL_ORGANICO, CANAL_INDICACAO)
5. Nível de Risco em um investimento	C (Classe)	É calculado por fórmulas complexas baseadas no mercado e no perfil do investidor. Muda com o tempo e impacta outras operações. Sua complexidade e variabilidade exigem que seja uma classe.
6. Idioma preferido de um usuário	A (Atributo)	É uma preferência simples e única do usuário. Não possui regras, histórico ou vida independente. É um dado estático, perfeito para ser um atributo idioma: String.
7. Plano de manutenção de um equipamento	C (Classe)	Envolve um calendário de atividades, peças de reposição, custos e prazos. Tem ciclo de vida próprio (datas de vencimento) e é reutilizável. É claramente uma entidade complexa.
8. Tipo de entrega em uma compra	A (Atributo)	É uma opção de catálogo com prazo e custo pré-definidos (ex: 'expresso', 'normal'). Apesar de influenciar o preço, é uma informação estática. Pode ser um atributo do Pedido.
9. Política de devolução de um vendedor	C (Classe)	Tem regras complexas (prazos, condições do produto, quem paga o frete). Um vendedor pode ter várias políticas. Tem texto legal e

		identidade única, portanto, é uma classe.
10.Certificado de calibração de um instrumento	C (Classe)	Tem número único, data de emissão, validade e é emitido por um órgão externo. Precisa ser reinspecionado. Tem todos os critérios de uma classe: identidade, ciclo de vida e regras.

Atividade 3: Multiplicidade e Opcionalidade

Objetivo: Definir as cardinalidades (1..1, 0..1, 0.., 1..) que fazem sentido para cada par de classes, baseando-se nas regras de negócio.

Cardinalidades Disponíveis:

- 1..1 - Obrigatório, exatamente um (ex: um carro tem **um** chassi)
- 0..1 - Opcional, no máximo um (ex: um carro pode ter **zero ou um** reboque)
- 1..* - Obrigatório, um ou mais (ex: um carro tem **pelo menos uma** roda)
- 0..* - Opcional, zero ou mais (ex: um carro pode ter **zero ou mais** adesivos)

1. **Multiplicidade A -> B:** Quantos Bs um A pode ter?
2. **Multiplicidade B -> A:** Quantos As um B pode ter?
3. **Justificativa:** Por que essa escolha faz sentido?

Tabela de Multiplicidades do Grupo

Situação	Multiplicidade (A -> B)	Multiplicidade (B -> A)	Justificativa (Por quê?)
1. Podcast ⇒ Episódios	1..*	1..1	Um podcast precisa ter pelo menos um episódio para existir. Um episódio pertence a apenas um podcast; não faz sentido um episódio estar em dois podcasts ao mesmo tempo.
2. Usuário ⇒ Métodos de Pagamento	1..*	1..1	Um usuário pode ter vários cartões cadastrados (1..*), mas um cartão está vinculado a apenas um usuário titular (1..1).
3. Reserva ⇒ Passageiros	1..*	0..*	Uma reserva de viagem pode ter um ou mais passageiros. Um passageiro pode estar em zero ou mais reservas (ele pode não viajar este mês, ou fazer várias reservas ao longo do ano)."
4. Quadro Kanban ⇒ Cartões	1..*	1..1	Um quadro Kanban precisa ter cartões para ser útil (1..*). Um cartão só pode estar em um quadro por vez (1..1), senão a organização vira uma bagunça.
5. Vaga ⇒ Veículo	0..1	0..1	Uma vaga de estacionamento pode estar vazia (0) ou ocupada por um (1) veículo. Um veículo pode estar estacionado em uma vaga (1) ou não (0) (se estiver em movimento).
6. Curso EAD ⇒ Tutores	1..*	0..*	Um curso EAD precisa de pelo menos um tutor de apoio. Um tutor pode dar apoio a zero ou mais cursos (ele pode ser alocado em vários ou em nenhum, se estiver de férias).
7. Loja ⇒ Vitrines Sazonais	1..*	1..1	Uma loja pode ter várias vitrines ativas (1..*). Uma vitrine sazonal pertence a apenas uma loja (1..1); não faz sentido a mesma vitrine física estar em duas lojas diferentes.
8. Voo ⇒ Bagagens Despachadas	0..*	1..1	Um voo pode ter zero (se for só com bagagem de mão) ou muitas bagagens despachadas. Uma bagagem só pode estar em um voo (1..1); ela não voa sozinha depois!
9. Projeto ⇒ Marcos (Milestones)	1..*	1..1	Um projeto precisa ter pelo menos um marco (milestone) para medir progresso. Um marco pertence a apenas um projeto (1..1); não existe marco 'compartilhado' entre projetos.
10. Máquina ⇒ Sensores	1..*	1..1	Uma máquina precisa de pelo menos um sensor para ser monitorada. Um sensor está

			instalado em apenas uma máquina (1..1); a leitura do sensor não pode ser misturada com a de outra máquina.
--	--	--	--

Conclusão do Grupo:

Shara: A multiplicidade 1..* (um para muitos) foi a mais comum, aparecendo em 7 das 10 situações. Isso mostra que relações 'um-para-muitos' são muito frequentes na modelagem de sistemas.

Diego: Já a 0..* (zero-para-muitos) apareceu para modelar situações onde a existência de uma instância não obriga a existência da outra, como em 'Reserva \Rightarrow Passageiros'.

Gustavo: A 0..1 (zero-ou-um) foi crucial para modelar opcionalidade, como uma 'Vaga' que pode estar vazia ou ocupada.

Luan: Notamos que a multiplicidade 1..1 (um-para-um) sempre aparece do lado 'fraco' da relação, indicando que um objeto 'pertence' exclusivamente a outro."

Nicolas (Moderador): O debate foi focado em encontrar a cardinalidade que melhor representa a realidade do negócio. A regra 'um podcast precisa de episódios' levou a 1..*, enquanto 'uma vaga pode estar vazia' levou a 0..1. A justificativa sempre esteve alinhada com as invariantes do domínio.