**~~🚀 Roteiro de Projetos (com a ordem sugerida)~~**

**~~1. Oráculo Simples →~~ *~~primeiro degrau~~***

* ~~Objetivo: treinar~~ **~~arrays, random, scanner~~**~~.~~
* ~~Duração: 1 noite de estudo.~~
* ~~Resultado: você pergunta qualquer coisa, o oráculo responde aleatoriamente.~~
* **~~Expansão~~**~~: adicionar categorias (respostas positivas, negativas, neutras).~~

**2. Jogo do Caçador Lunar**

* Objetivo: treinar **condicionais + loops + arrays para histórico**.
* Duração: 2 a 3 noites de estudo.
* Resultado: jogador enfrenta monstros, programa sorteia e resolve o combate.
* **Expansão**: criar “vida” e “honra” do jogador e decrementar conforme escolhas.

**3. Cadastro de Devotos de Ártemis**

* Objetivo: treinar **arrays + switch + laços de repetição**.
* Duração: 1 semana.
* Resultado: sistema que cadastra pessoas, lista e mostra a categoria (noviço/iniciado).
* **Expansão**: permitir excluir devotos, ou limitar a 10 devotos no array.

**4. Taverna de Bruxas**

* Objetivo: juntar **arrays, switch, loops, cálculo de valores**.
* Duração: 1 a 2 semanas.
* Resultado: sistema de pedidos que mostra conta final no estilo “fantasia medieval”.
* **Expansão**:
  + Criar estoque (se acabar a poção, não pode pedir).
  + Permitir descontos (“se comprar 3 Poções de Cura, ganha 10% de desconto”).

✨ No fim, você pode até juntar tudo em um **Projeto Final: O Reino de Nyxvalis** → onde o jogador entra na Taverna, consulta o Oráculo, enfrenta monstros e se cadastra como devoto. Seria um mini-RPG textual feito só com o nível iniciante.

**🚀 Roteiro de Projetos – Nível Intermediário Java10x**

**1️⃣ Mini RPG de Criaturas Místicas**

* **Objetivo:** treinar **Classes, Objetos e Métodos**.
* **O que você aprende:** criar classes para monstros, magos, itens; métodos para ações (ataque, cura, defesa).
* **Expansão:** adicionar **parâmetros e retorno** nos métodos; criar herança básica (ex: Monstro → Dragão, Troll).

**2️⃣ Sistema de Cadastro de Devotos**

* **Objetivo:** aplicar **Herança, Construtores e Encapsulamento**.
* **O que você aprende:**
  + Criar uma **superclasse Devoto** e subclasses (Noviço, Iniciado, Mestre).
  + Usar construtores para inicializar dados.
  + Aplicar getters e setters para proteger dados sensíveis.
* **Desafio extra:** usar **sobrecarga de construtores** pra permitir diferentes formas de criar devotos.

**3️⃣ Oráculo Avançado**

* **Objetivo:** treinar **Interfaces e Polimorfismo**.
* **O que você aprende:**
  + Criar interface “Resposta” com método gerarResposta().
  + Diferentes classes implementam a interface: RespostaPositiva, RespostaNegativa, RespostaNeutra.
  + Polimorfismo: escolher qualquer objeto de tipo Resposta e chamar gerarResposta().
* **Extra:** adicionar **ENUMS** para categorias de resposta.

**4️⃣ Banco de Konoha (Desafio 5)**

* **Objetivo:** juntar **Classes Abstratas, Herança, Polimorfismo, Encapsulamento, List e Generics**.
* **O que você aprende:**
  + Criar **superclasse Conta** e subclasses (Corrente, Poupança).
  + Aplicar **encapsulamento** com getters/setters.
  + Usar **listas** para armazenar contas e transações.
  + Polimorfismo: método transferir() que funciona para qualquer tipo de conta.
  + Generics: criar lista genérica de objetos (opcional).
* **Expansão:** adicionar **Records** para registrar transações de forma imutável.

**5️⃣ Mini Sistema de Taverna (Projeto Final Intermediário)**

* **Objetivo:** consolidar tudo: **Classes, Herança, Interfaces, Polimorfismo, Lists, Enums, Records e Stack**.
* **O que você aprende:**
  + Classes para Clientes, Itens, Pedidos.
  + Enum para tipos de itens (Poção, Elixir, Pergaminho).
  + Lista para armazenar pedidos ativos.
  + Stack para pedidos recentes (último pedido entra, primeiro sai).
  + Encapsulamento e getters/setters em todas as classes.
* **Extra:** usar **polimorfismo** para que cada item tenha efeito diferente quando consumido.

💡 **Estratégia de estudo:**

* Comece pelo **Mini RPG**, consolidando Classes e Métodos.
* Suba gradualmente até o **Sistema da Taverna**, que junta todos os conceitos intermediários.
* Cada projeto deve ser **prático e divertido**, com outputs no console, pra você ver a “mágica” acontecendo.