

Treinamento de Agentes LLM com Reinforcement Learning

Germano Andrade
Patrick Saul
Sávio Vinícius



Índice

1. **Objetivo**
2. **Espaço de Ação**
3. **Agentes LLM**
4. **Recompensas**
5. **Treinamento**
6. **Análise de Resultados**
7. **Melhorias**



Objetivos

Objetivo principal

Treinar dois agentes LLM para cooperarem na solução de problemas de programação.

Tarefas específicas

- Escolher o modelo LLM
- Definir o espaço de ação de cada agente
- Definir as recompensas
- Avaliar o resultado



Espaço de Ações



Agente Codificador

Ações que o agente codificador pode tomar

Limpeza e pré-processamento

Limpar, transformar e preparar os dados para análise

Análise estatística


Calcular a média e a variância de um dataset

Deteção de outliers

Criar função para detectar outliers em um dataset

Criar modelos estatísticos

Criar função para realizar uma regressão linear em um dataset





Agente Revisor

Ações que o agente revisor pode tomar

Análise Estática


Analisar o código com
Mypy, Flake8 e Pylint

Executar código

Executar o código para
verificar se ele está
correto

Propor Refatoração

Criar função para
detectar outliers em um
dataset



Agentes LLM

Agentes LLM

Modelos LLM aplicados

- codegen-350M-mono
- Qwen2.5-Coder-1.5B-Instruct
- Llama-3.2-1B
- DeepSeek-Coder-V2-Instruct
- Phi-3-mini-128k-instruct

Recompensas



Recompensas positivas

Reconhecendo a Qualidade e
Boas Práticas no Código



**Pontuação
Pylint**



**Presença de
Estruturas de
Código
Funcionais**



Recompensas negativas

Penalizando Violações e
Aspectos Críticos do
Código



**Violações de
Flake8**



**Violações de
Mypy**



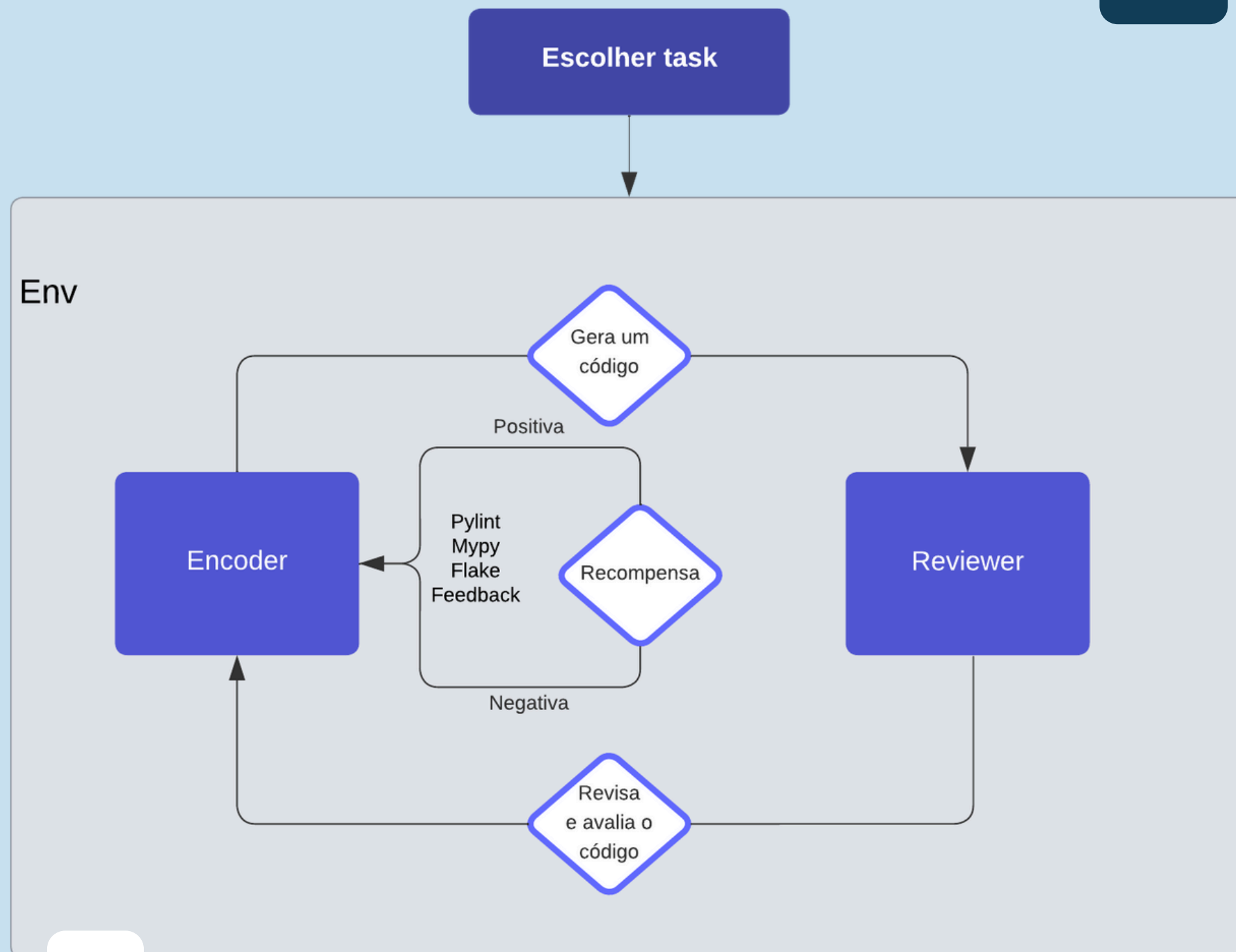
**Feedback
Crítico na
Revisão**



Treinamento

Treinamento

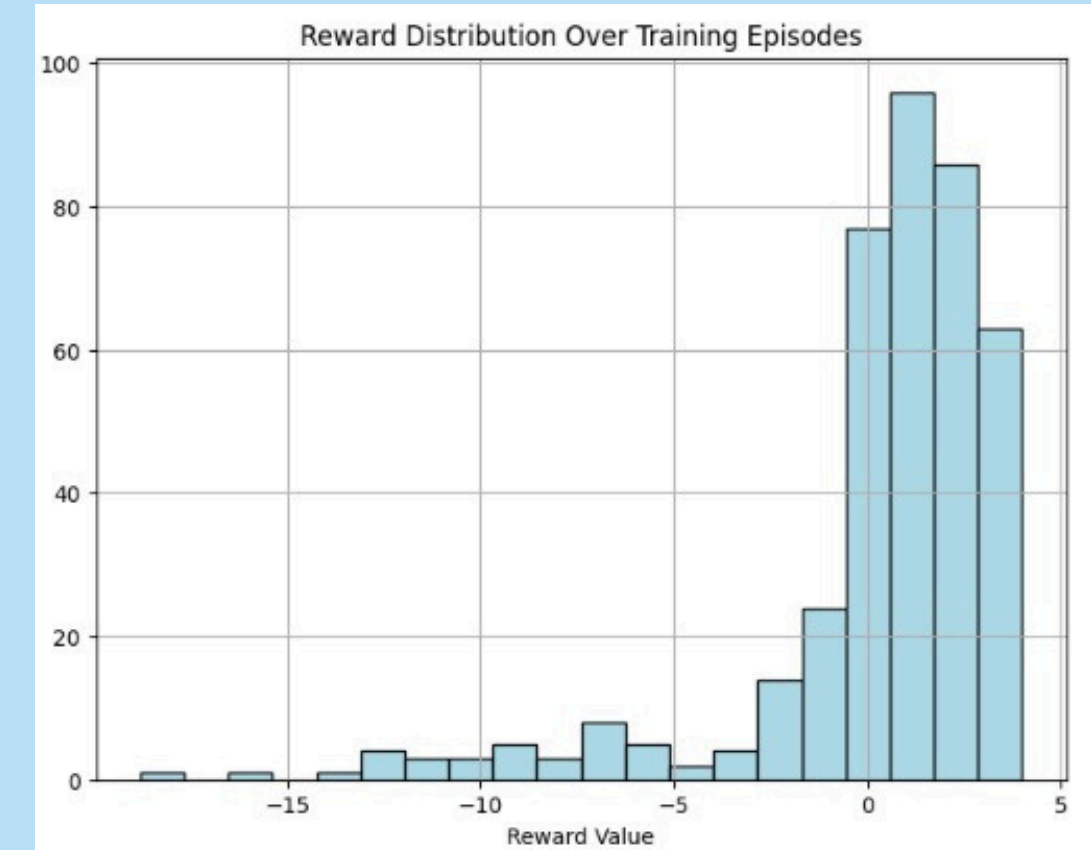
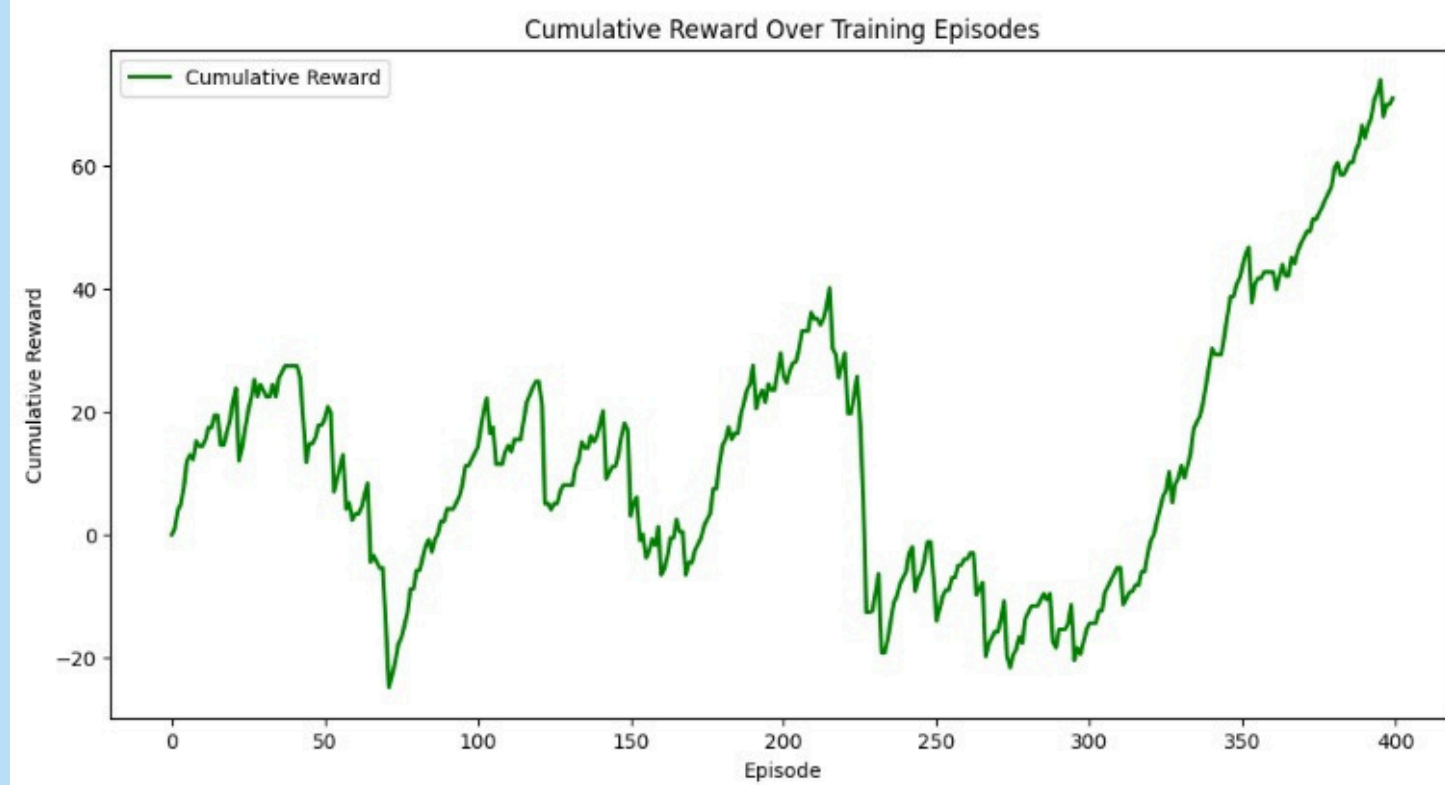
Fluxograma do modelo



Análise de Resultados

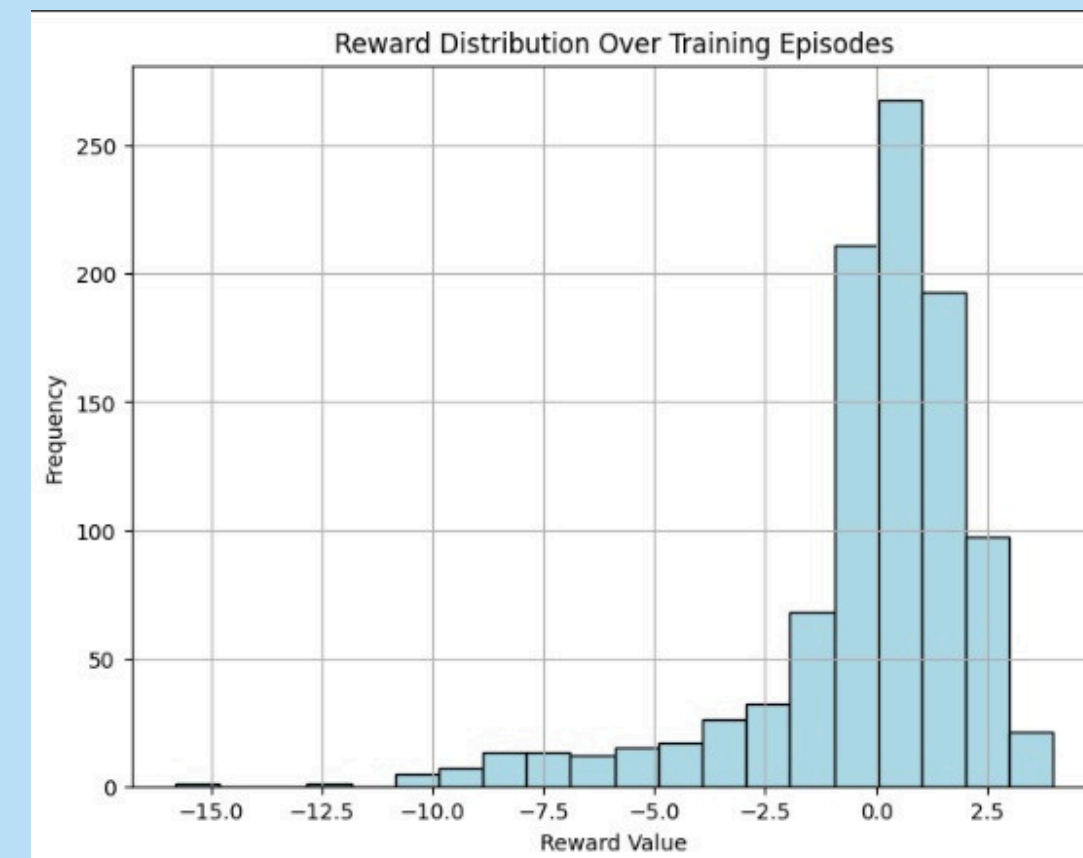
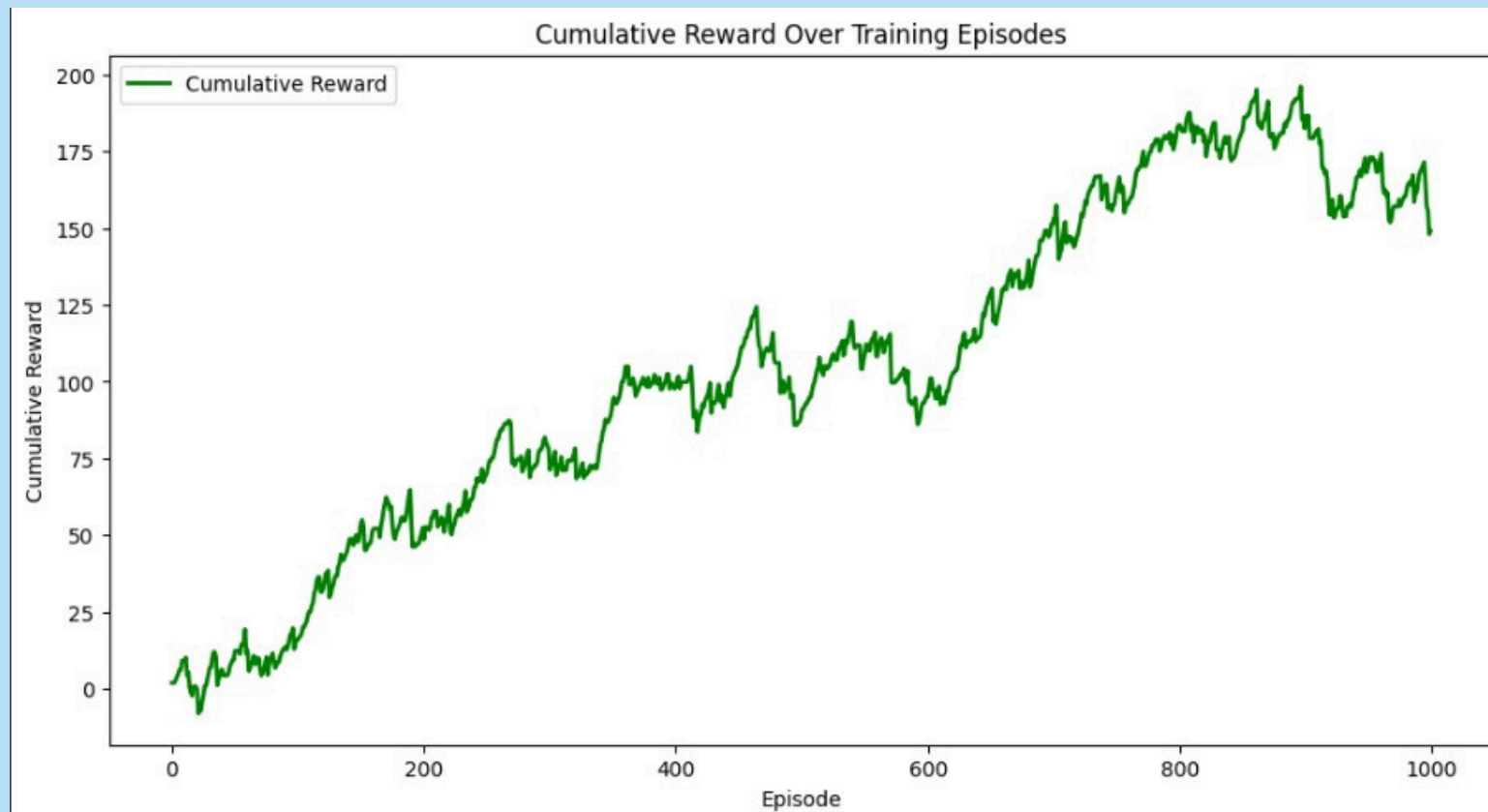
codegen-350M-mono

350 Milhões de parâmetros treinados
400 episódios



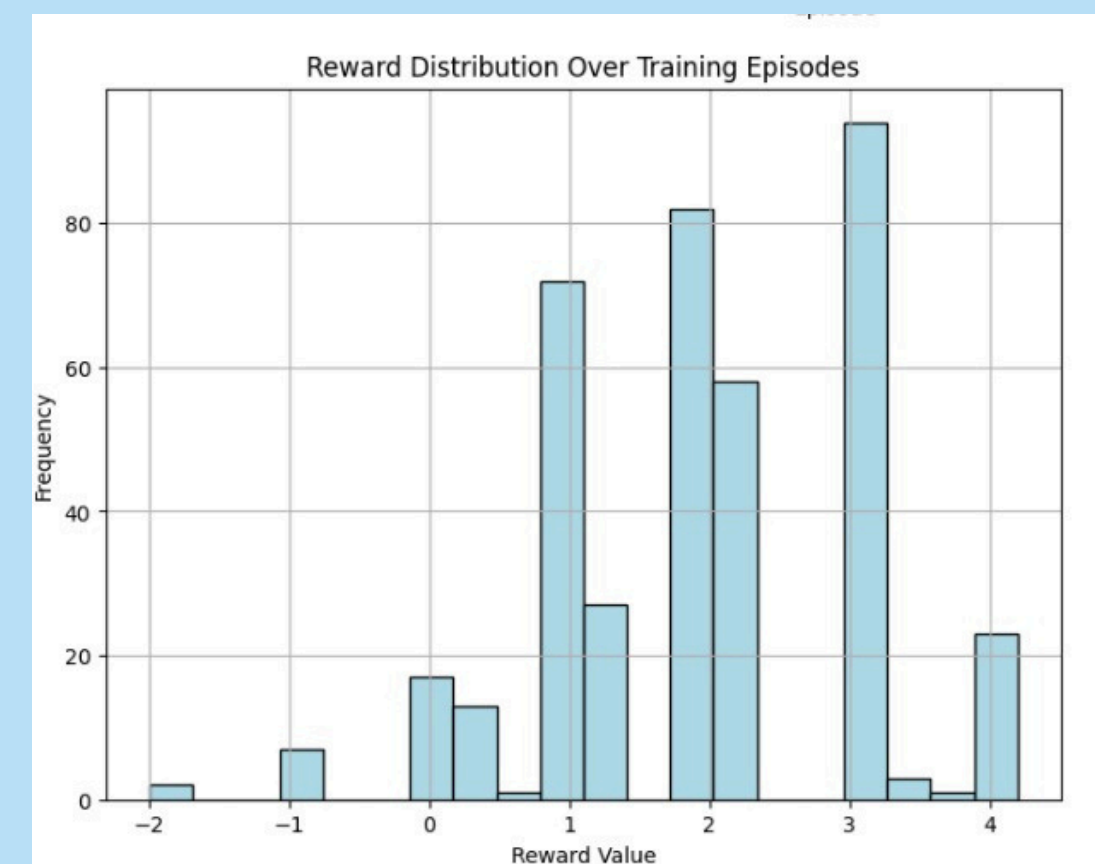
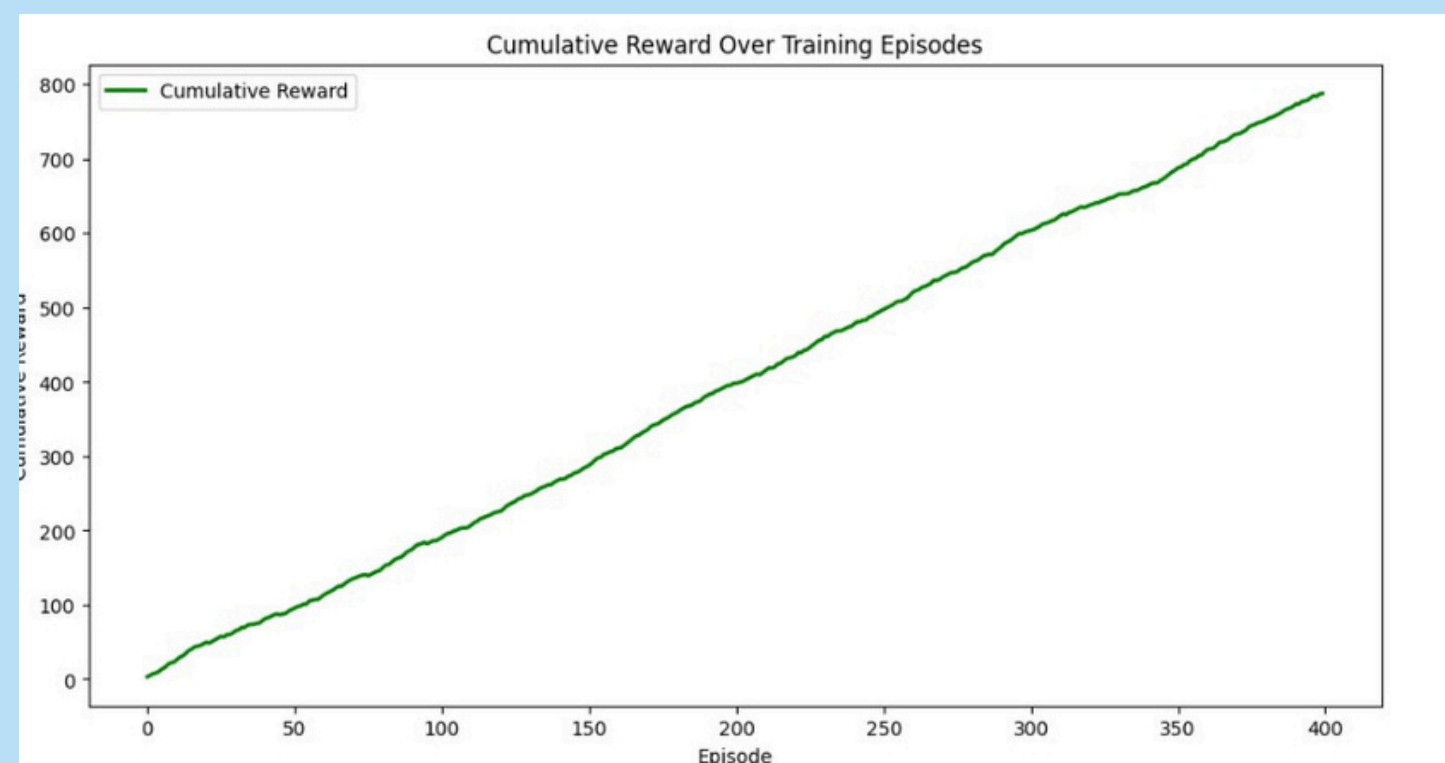
codegen-350M-mono

350 Milhões de parâmetros treinados
1000 episódios



Qwen2.5-Coder-1.5B-Instruct

1.5 Bilhões de parâmetros treinados
400 episódios



Melhorias

Sugestões práticas para aprimorar o desempenho e aplicabilidade

1

Ampliar as tarefas
para incluir casos
mais complexos

2

Avaliar o modelo
em tarefas
inéditas

3

Adicionar
recompensa para a
eficiência do código