

Nome do Aluno	Gabriel Martins de Almeida
Ra	31010006371
Curso de Ciências da computação	

**CENTRO UNIVERSITÁRIO DE ITAPIRA**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**Desenvolvimento de Projeto Alexa**

**Gabriel Martins de Almeida**

Nome do Aluno	Gabriel Martins de Almeida
Ra	31010006371
Curso de Ciências da computação	

A Skill chamada “Estudos Disciplinares”, se trata de um conto de fábula infantil onde a pessoa pede para a alexa contar uma fábula, e ela conta a fábula do cão e a sombra, logo após a pessoa pode dizer para a mesma dizer a moral da historia com o comando de voz “moral”.

Nome da Intent : FabulaIntent	Chamada: fabula	Resposta: Um cão, que levava um naco de carne na boca, passava numa ponte sobre um rio, quando viu a sua sombra refletida na água lá em baixo. Pensando que era outro cão que levava um segundo naco de carne, o insaciável do cão não resistiu a atirar-se à água para lhe roubar a carne. É claro que, em vez de lhe roubar o segundo naco de carne, perdeu o que tinha, que caiu ao fundo do rio.
Nome da Intent: MoralIntent	Chamada: moral	Resposta: Quem tudo quer, tudo perde. A cobiça acaba por perder aquilo que deseja, e aquele que pretende mais do que lhe é devido merece perder o que tem.

#### Código Programável:

```
// This sample demonstrates handling intents from an Alexa skill using the Alexa Skills
// Kit SDK (v2).
// Please visit https://alexa.design/cookbook for additional examples on implementing
// slots, dialog management,
// session persistence, api calls, and more.
const Alexa = require('ask-sdk-core');

const LaunchRequestHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchRequest';
  },
  handle(handlerInput) {
    const speakOutput = 'olá! Bem vindo a fabulo do cão e a sombra, diga fabula para
    eu lhe contar a historia, depois moral para ouvir a moral da historia!';
    return handlerInput.responseBuilder
      .speak(speakOutput)
```

Nome do Aluno	Gabriel Martins de Almeida
Ra	31010006371
Curso de Ciências da computação	

```
.reprompt(speakOutput)
.getResponse();
}
};
const FabulaIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'FabulaIntent';
  },
  handle(handlerInput) {
    const speakOutput = 'Um cão, que levava um naco de carne na boca, passava
    numa ponte sobre um rio, quando viu a sua sombra refletida na água lá em baixo.
    Pensando que era outro cão que levava um segundo naco de carne, o insaciável do cão
    não resistiu a atirar-se à água para lhe roubar a carne. É claro que, em vez de lhe
    roubar o segundo naco de carne, perdeu o que tinha, que caiu ao fundo do rio.';
    return handlerInput.responseBuilder
      .speak(speakOutput)
      //.reprompt('add a reprompt if you want to keep the session open for the user
to respond')
      .getResponse();
  }
};
const MoralIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'MoralIntent';
  },
  handle(handlerInput) {
    const speakOutput = 'Quem tudo quer, tudo perde. A cobiça acaba por perder
    aquilo que deseja, e aquele que pretende mais do que lhe é devido merece perder o
    que tem.';
    return handlerInput.responseBuilder
      .speak(speakOutput)
      //.reprompt('add a reprompt if you want to keep the session open for the user
to respond')
      .getResponse();
  }
};
const HelpIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) ===
'AMAZON.HelpIntent';
  },
```

Nome do Aluno	Gabriel Martins de Almeida
Ra	31010006371
Curso de Ciências da computação	

```
handle(handlerInput) {
  const speakOutput = 'You can say hello to me! How can I help?';

  return handlerInput.responseBuilder
    .speak(speakOutput)
    .reprompt(speakOutput)
    .getResponse();
}
};

const CancelAndStopIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && (Alexa.getIntentName(handlerInput.requestEnvelope) ===
'AMAZON.CancelIntent'
      || Alexa.getIntentName(handlerInput.requestEnvelope) ===
'AMAZON.StopIntent');
  },
  handle(handlerInput) {
    const speakOutput = 'Goodbye!';
    return handlerInput.responseBuilder
      .speak(speakOutput)
      .getResponse();
  }
};

const SessionEndedRequestHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) ===
'SessionEndedRequest';
  },
  handle(handlerInput) {
    // Any cleanup logic goes here.
    return handlerInput.responseBuilder.getResponse();
  }
};

// The intent reflector is used for interaction model testing and debugging.
// It will simply repeat the intent the user said. You can create custom handlers
// for your intents by defining them above, then also adding them to the request
// handler chain below.
const IntentReflectorHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest';
  },
  handle(handlerInput) {
```

Nome do Aluno	Gabriel Martins de Almeida
Ra	31010006371
Curso de Ciências da computação	

```
const intentName = Alexa.getIntentName(handlerInput.requestEnvelope);
const speakOutput = `You just triggered ${intentName}`;

return handlerInput.responseBuilder
    .speak(speakOutput)
    // .reprompt('add a reprompt if you want to keep the session open for the user
to respond')
    .getResponse();
}
};

// Generic error handling to capture any syntax or routing errors. If you receive an
error
// stating the request handler chain is not found, you have not implemented a handler
for
// the intent being invoked or included it in the skill builder below.
const ErrorHandler = {
    canHandle() {
        return true;
    },
    handle(handlerInput, error) {
        console.log(`~~~~~ Error handled: ${error.stack}`);
        const speakOutput = `Sorry, I had trouble doing what you asked. Please try again.`;

        return handlerInput.responseBuilder
            .speak(speakOutput)
            .reprompt(speakOutput)
            .getResponse();
    }
};

// The SkillBuilder acts as the entry point for your skill, routing all request and
response
// payloads to the handlers above. Make sure any new handlers or interceptors you've
// defined are included below. The order matters - they're processed top to bottom.
exports.handler = Alexa.SkillBuilders.custom()
    .addRequestHandlers(
        LaunchRequestHandler,
        HelpIntentHandler,
        CancelAndStopIntentHandler,
        SessionEndedRequestHandler,
        MoralIntentHandler,
        FabulaIntentHandler,
```

Nome do Aluno	Gabriel Martins de Almeida
Ra	31010006371
Curso de Ciências da computação	

IntentReflectorHandler, // make sure IntentReflectorHandler is last so it doesn't  
override your custom intent handlers

```
)  
.addErrorHandlers(  
    ErrorHandler,  
)  
.lambda();
```