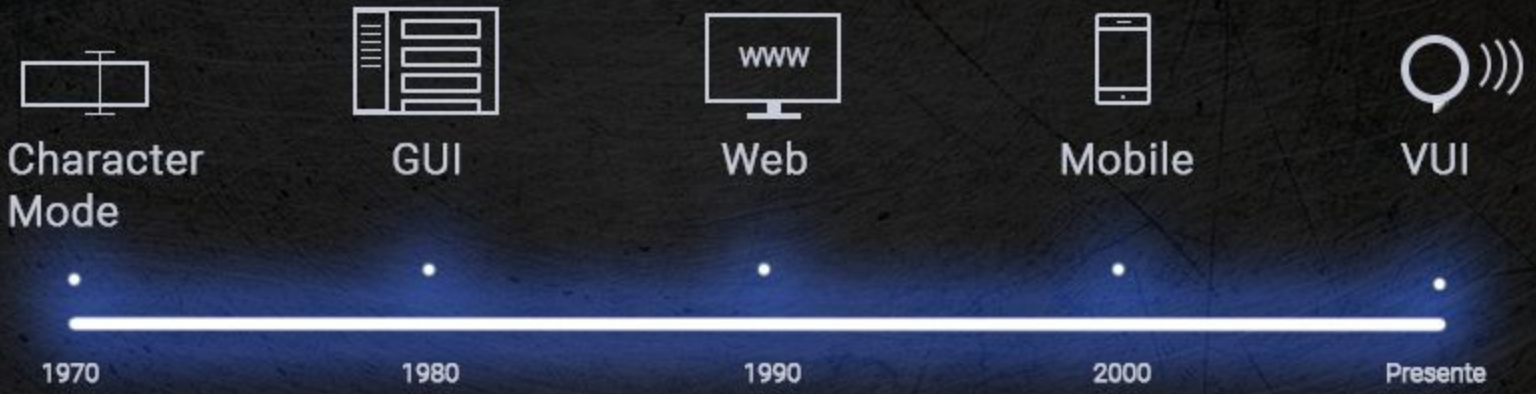

ALEXA FROM ZERO TO
HERO
lab


Character Mode
GUI
www
Web
Mobile
VUI
1970
1980
1990
2000
Presente

Assistentes de voz

Amazon Echo

Apple Homepod

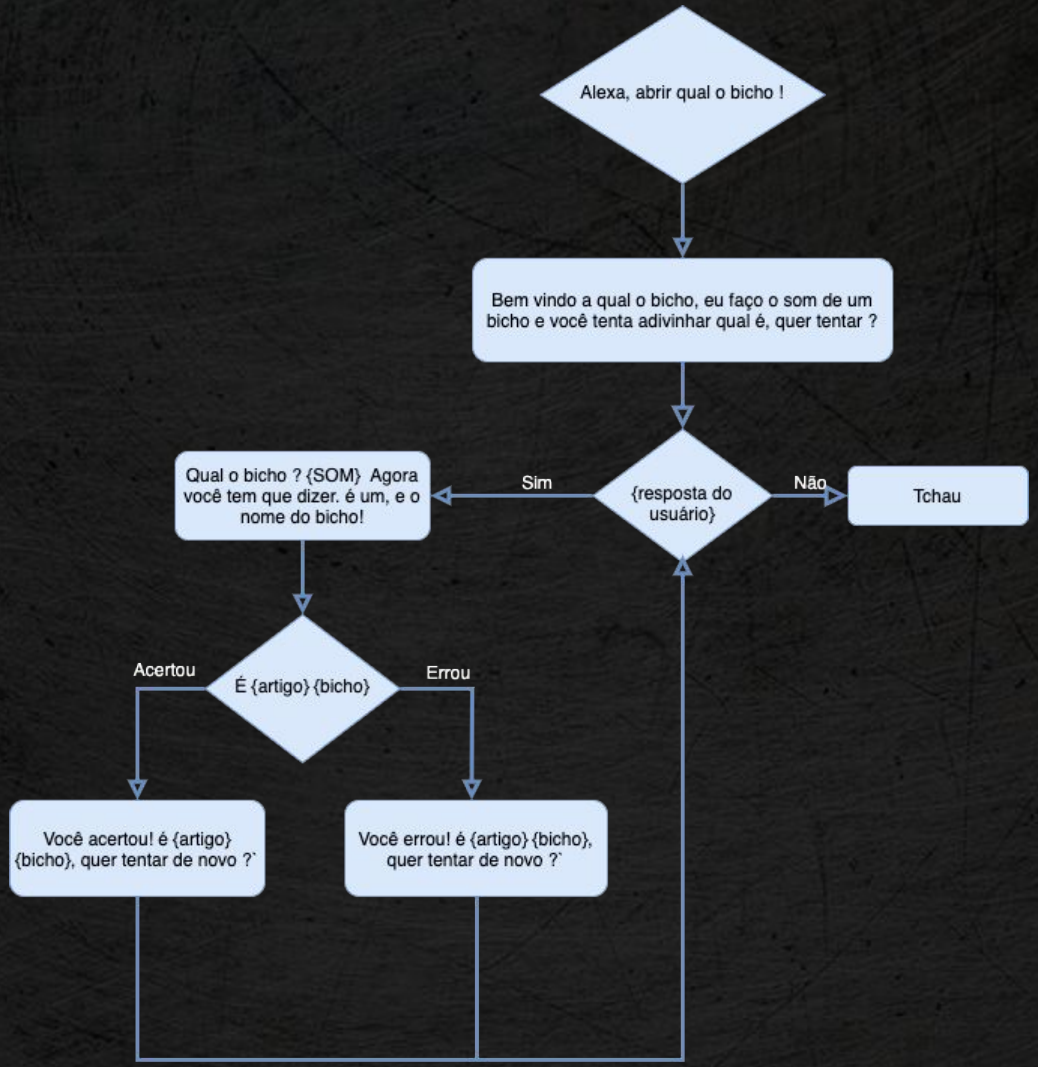
Google Home

Cortana - Microsoft

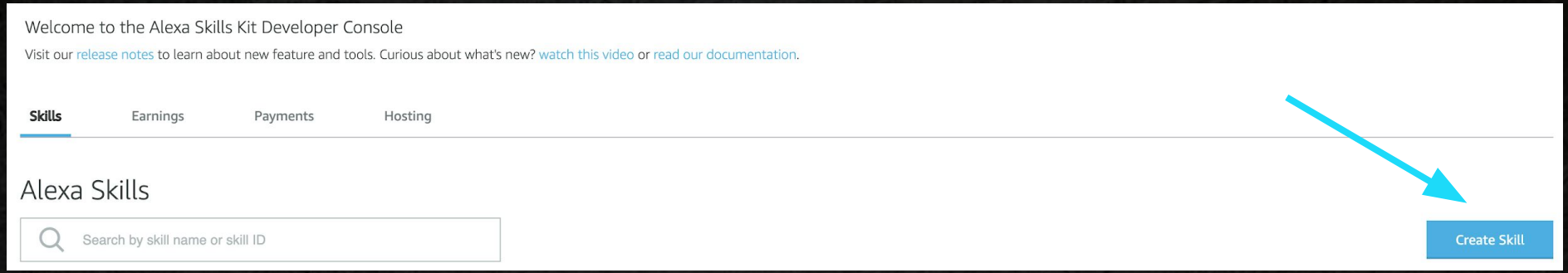

50,000+ SKILLS
and growing every day


Processamento de Linguagem
IOT
Alexa
Machine Learning
IA

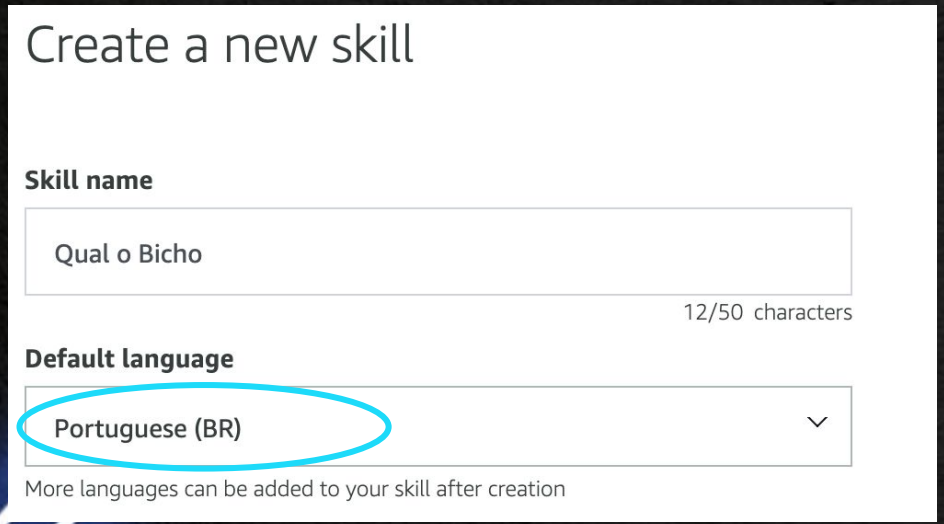

Bora codar!
lab


Alexa, abrir qual o bicho !
Bem vindo a qual o bicho, eu faço o som de um bicho e você tenta adivinhar qual é, quer tentar ?
{resposta do usuário}
Sim
Não
Tchau
Qual o bicho ? {SOM} Agora você tem que dizer. é um, e o nome do bicho!
É {artigo} {bicho}
Acertou
Errou
Você acertou! é {artigo} {bicho}, quer tentar de novo ?`
Você errou! é {artigo} {bicho}, quer tentar de novo ?`

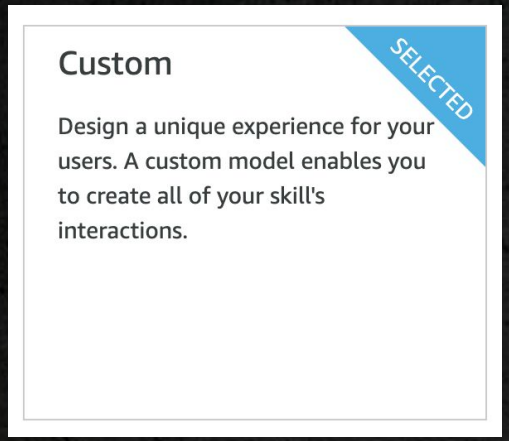
1. Criar uma nova skill

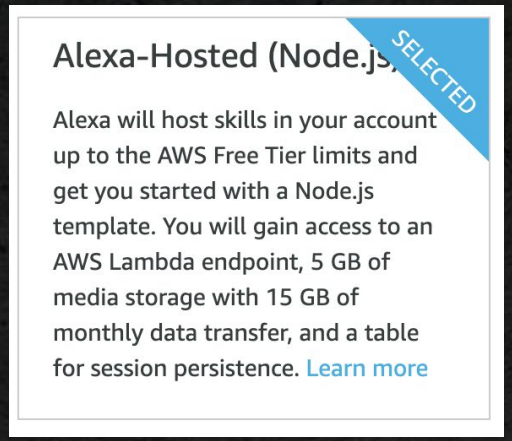


2. Nome e língua da skill

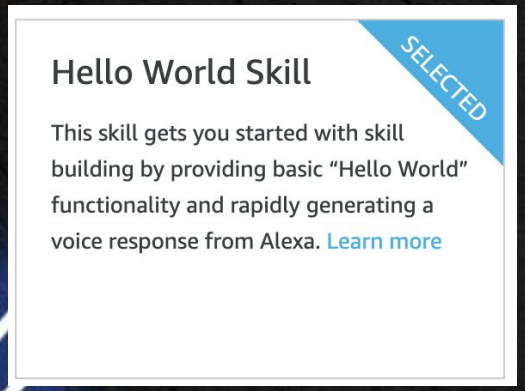


3. Ambiente Dev

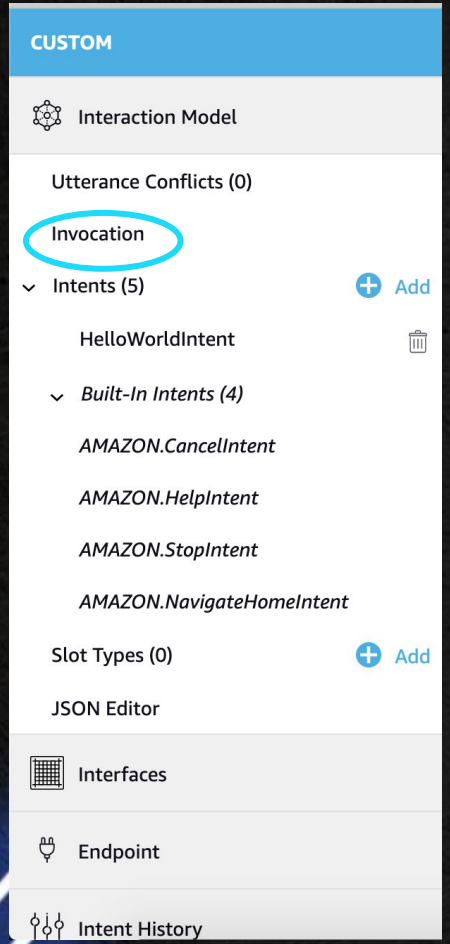




4. Template para a skill

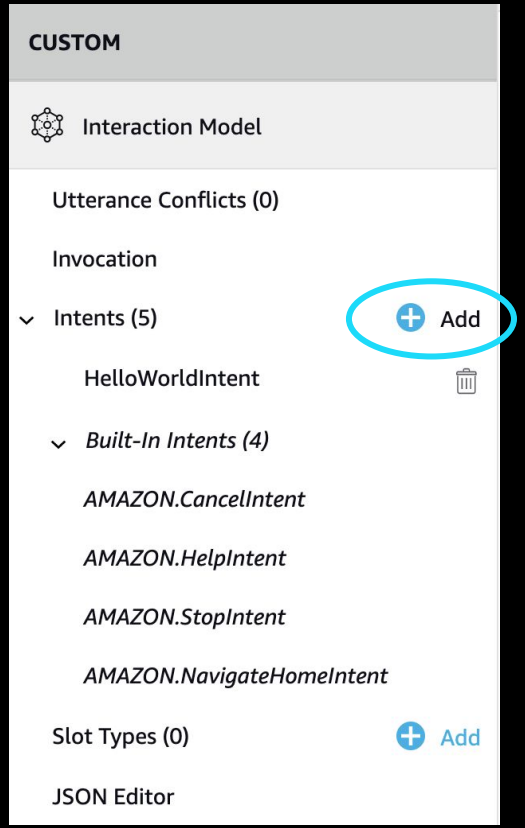


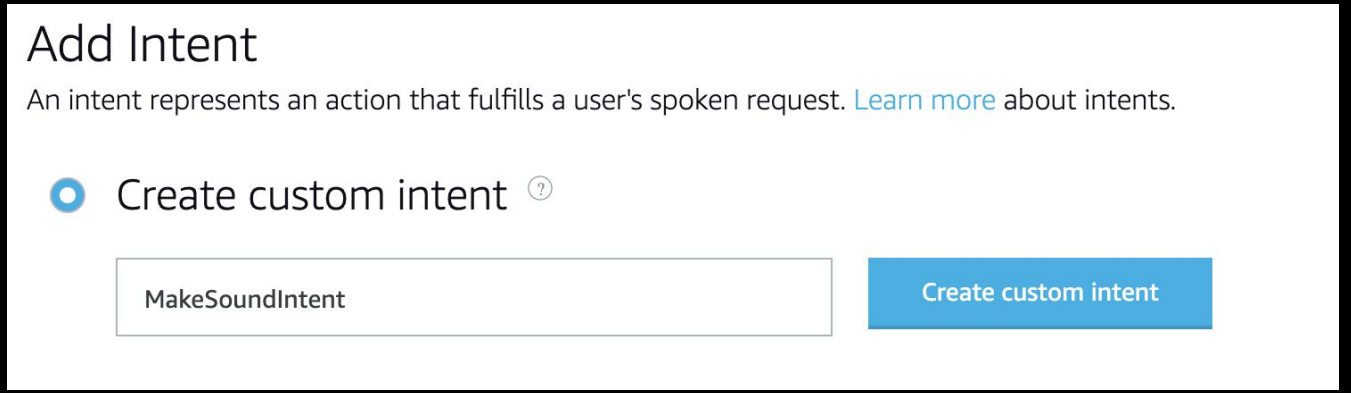
5. Template pra skill

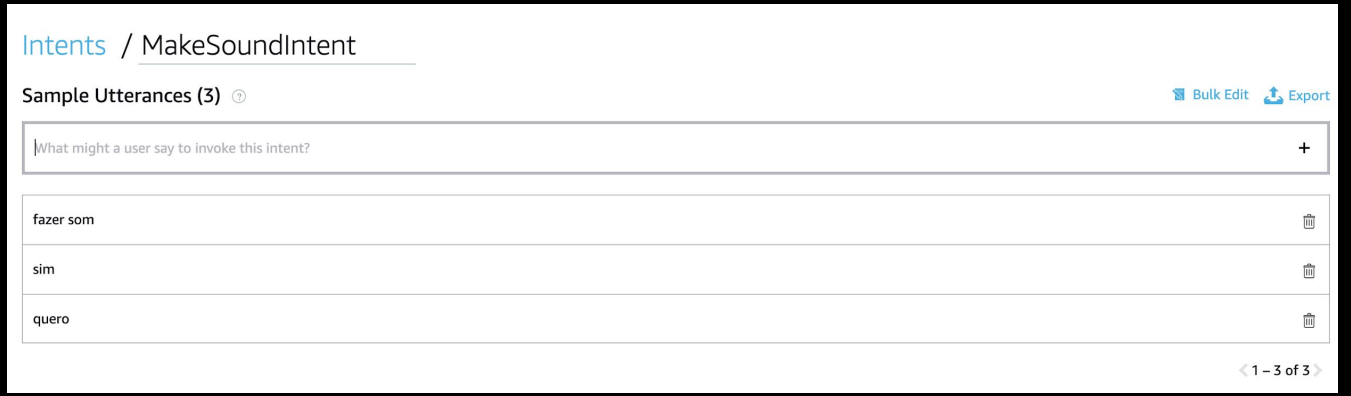


6. Nome de Invocação

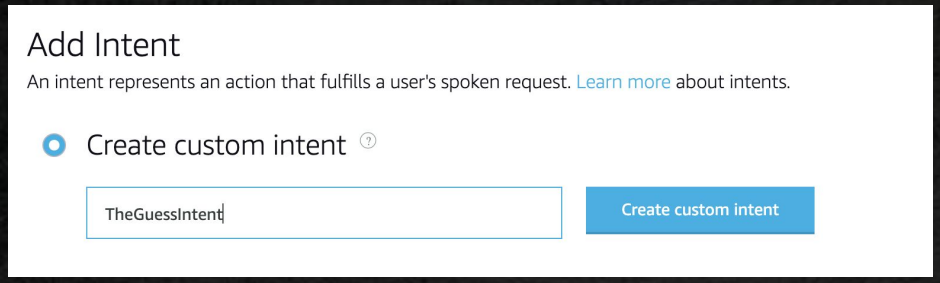
7. Criar Intent (intenção) para reproduzir o som do bicho

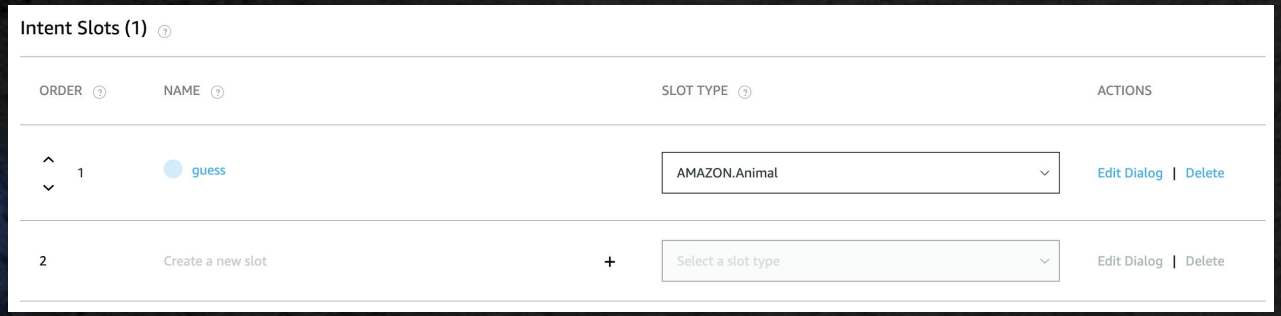




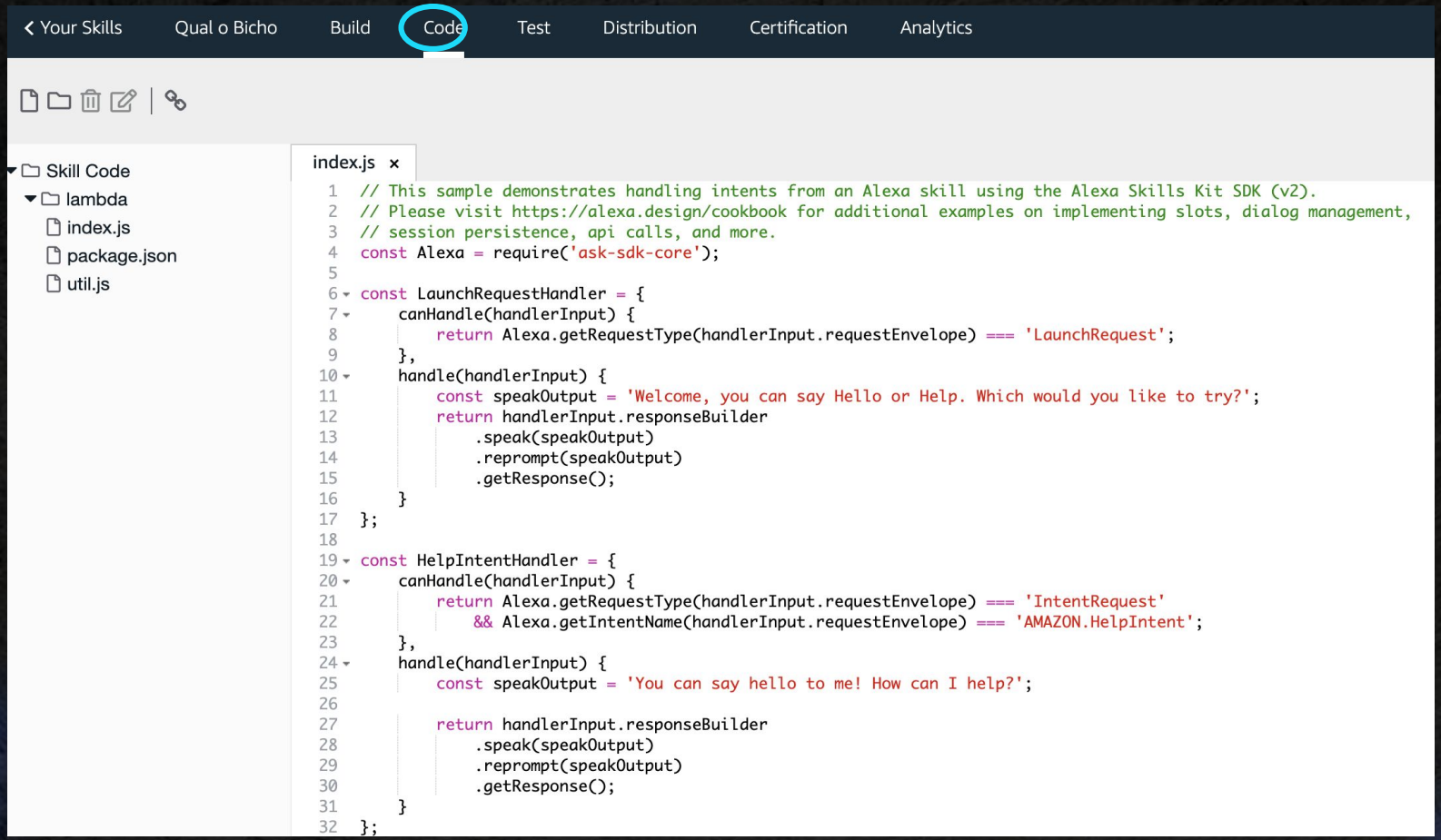


8. Criar Intent (intenção) para adivinhar qual o bicho





9. Ufa! Vamos codar



10. Mudar handler de boas vindas

```
const LaunchRequestHandler = {
    canHandle(handlerInput) {
        return Alexa.getRequestType(handlerInput.requestEnvelope) === 'LaunchRequest';
    },
    handle(handlerInput) {
        const speakOutput = 'Bem vindo a qual o bicho, eu faço o som de um bicho e você tenta adivinhar qual é,
quer tentar ?';
        return handlerInput.responseBuilder
            .speak(speakOutput)
            .reprompt(speakOutput)
            .getResponse();
    }
};
```

11. Traduzir mensagens padrões

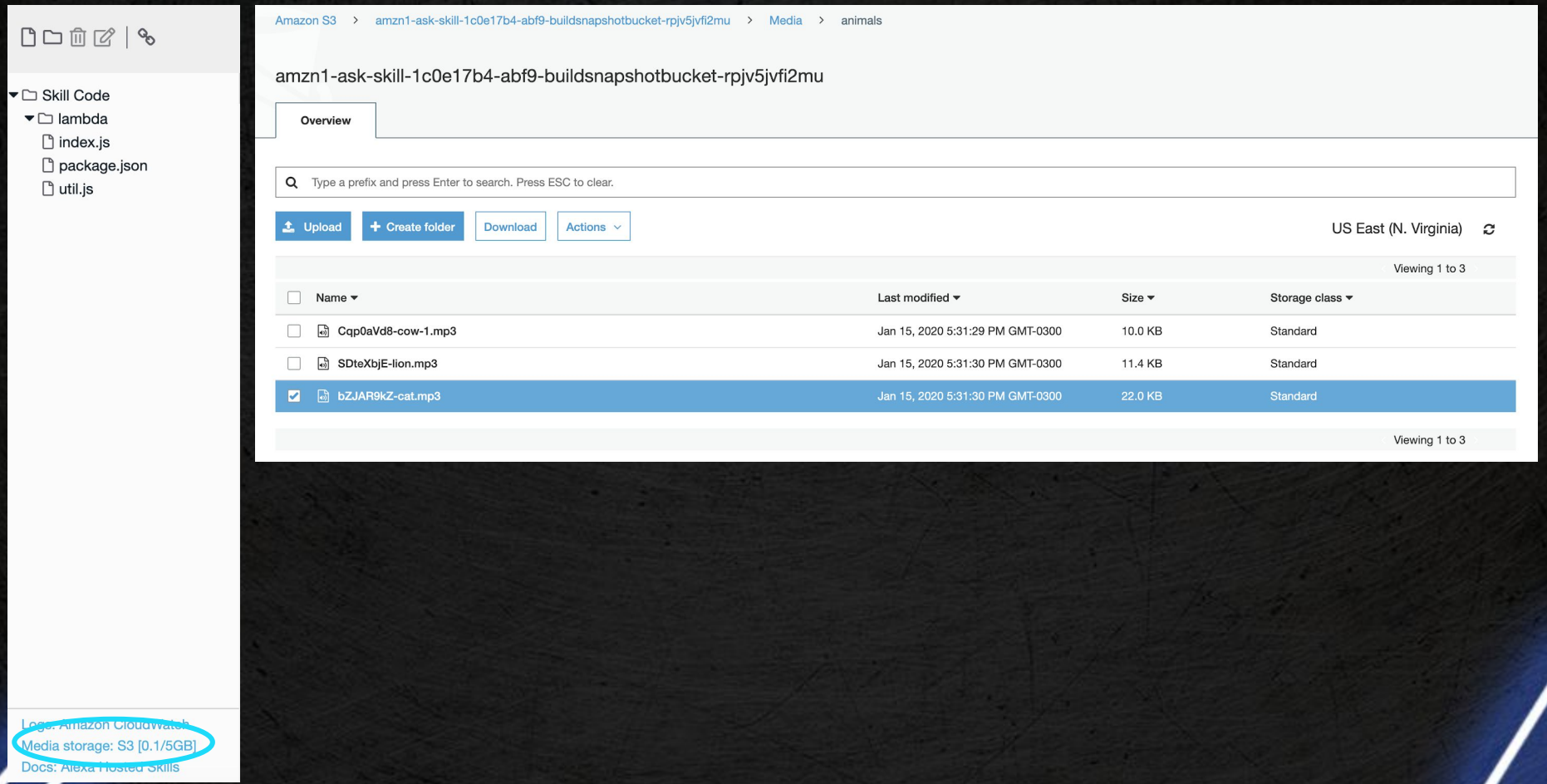
```
const HelpIntentHandler = {
      const speakOutput = 'Posso fazer um som de bicho pra você. quer tentar?';
};

const CancelAndStopIntentHandler = {
      const speakOutput = 'Até logo!';
};

const IntentReflectorHandler = {
      const speakOutput = `Você tentou acionar ${intentName}`;
};

const ErrorHandler = {
      const speakOutput = `Desculpa, Eu não consegui entender o que você disse. Por favor tente outra vez.`;
};
```

12. Subir som dos bichos



13. Array de bichos

```
const animals = [
    {
        'name':"Leão",
        'article':'um',
        'url':"Media/animals/SDteXbjE-lion.mp3"
    },
    {
        'name':'gato',
        'article':'um',
        'url':'Media/animals/bZJAR9kZ-cat.mp3'
    },
    {
        'name':'vaca',
        'article':'uma',
        'url':'Media/animals/Cqp0aVd8-cow-1.mp3'
    }
];
```

14. Método para obter bicho do array

```
function getAvailableIndex(){
      return Math.floor(Math.random() * 2);
}
```

15. imports

```
const MakeSoundIntentHandler = {
    canHandle(handlerInput) {
        return handlerInput.requestEnvelope.request.type === 'IntentRequest'
            && handlerInput.requestEnvelope.request.intent.name === 'MakeSoundIntent';
    },
    handle(handlerInput) {

            const sessionAttributes = handlerInput.attributesManager.getSessionAttributes();
            //me da o indice de um dos bichos no array
            let index = getAvailableIndex();
            //me da o bicho
            let currentAnimal = animals[index];
            //magica pra assinar a url que vai chamar o som armazenado no s3
            let currentAnimalURL = Util.getS3PreSignedUrl(currentAnimal.url);

            sessionAttributes.currentAnimal = currentAnimal;

            let speechText = `Qual o bicho ? <audio src="${Escape(currentAnimalURL)}"/>  Agora você tem que dizer. é um, e o nome do bicho!`;

        return handlerInput.responseBuilder
            .speak(speechText)
            .reprompt(speechText)
            .getResponse();
    }
};
```

OBS : Incluir método no exports.handler

16. imports

```
const Util = require('./util.js');
const Escape = require('lodash/escape');
const _LOG = 'MEU-LOG: ';
```

17. incluir biblioteca Escape

```
{
  "name": "qual o bicho",
  "version": "1.0.0",
  "description": "alexa utility for quickly building skills",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Amazon Alexa",
  "license": "ISC",
  "dependencies": {
    "ask-sdk-core": "^2.6.0",
    "ask-sdk-model": "^1.18.0",
    "aws-sdk": "^2.326.0",
    "lodash": "^4.17.11"
  }
}
```

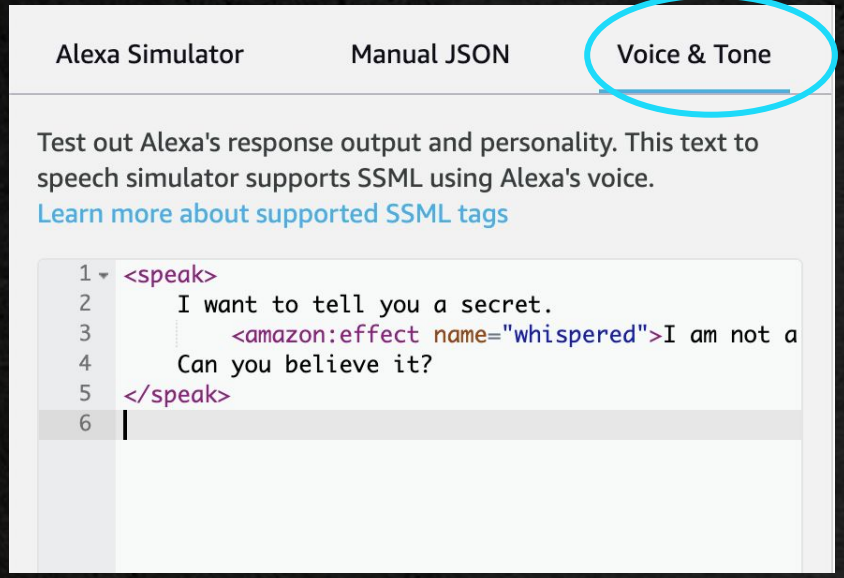
18. imports

```
const TheGuessIntentHandler = {
    canHandle(handlerInput) {
        return handlerInput.requestEnvelope.request.type === 'IntentRequest'
            && handlerInput.requestEnvelope.request.intent.name === 'TheGuessIntent';
    },
    handle(handlerInput) {
        const sessionAttributes = handlerInput.attributesManager.getSessionAttributes();
        const slots = handlerInput.requestEnvelope.request.intent.slots;
        const guess = slots['guess'].value;
        let speechText = `Você disse ${guess}`;
        let currentAnimal = sessionAttributes.currentAnimal;
        if(currentAnimal.name.toLowerCase() === guess){
            speechText = `Você acertou! é ${currentAnimal.article} ${currentAnimal.name}, quer tentar de novo ?`;
        }else{
            speechText = `Você errou! é ${currentAnimal.article} ${currentAnimal.name}, quer tentar de novo ?`;
        }
        return handlerInput.responseBuilder
            .speak(speechText)
            .reprompt(speechText)
            .getResponse();
    }
};
```

OBS : Incluir método no exports.handler


Bom trabalho
lab

Bônus


Alexa Simulator
Manual JSON
Voice & Tone
Test out Alexa's response output and personality. This text to speech simulator supports SSML using Alexa's voice.
Learn more about supported SSML tags
1 <speak>
2 I want to tell you a secret.
3 <amazon:effect name="whispered">I am not a
4 Can you believe it?
5 </speak>
6


Intents / AMAZON.CancelIntent
Sample Utterances (1)
What might a user say to invoke this intent?
não

DESAFIO : MUDAR O JOGO PARA SOMENTE FAZER UMA VEZ O SOM DE CADA BICHO !

```
function getAvailableIndex(availableGuesses){
      return Math.floor(Math.random() * availableGuesses.length);
}

//LaunchRequestHandler :

const sessionAttributes = handlerInput.attributesManager.getSessionAttributes();
sessionAttributes.availableGuesses = animals;
//MakeSoundIntentHandler:

let availableGuesses = sessionAttributes.availableGuesses;
let speechText;

if(availableGuesses.length === 0){
speechText = 'Desculpa, eu não tenho mais sons de bichos. Quer começar de novo? Ou quer sair do jogo?'
}else{
      let index = getAvailableIndex(availableGuesses);
      let currentAnimal = availableGuesses[index];

      availableGuesses.splice(index,1);
      sessionAttributes.availableGuesses = availableGuesses;
```