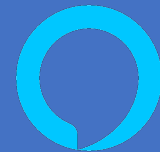


Programación de Skills con Alexa

# Manejo de Diálogos y validación de Slots

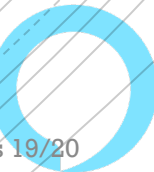
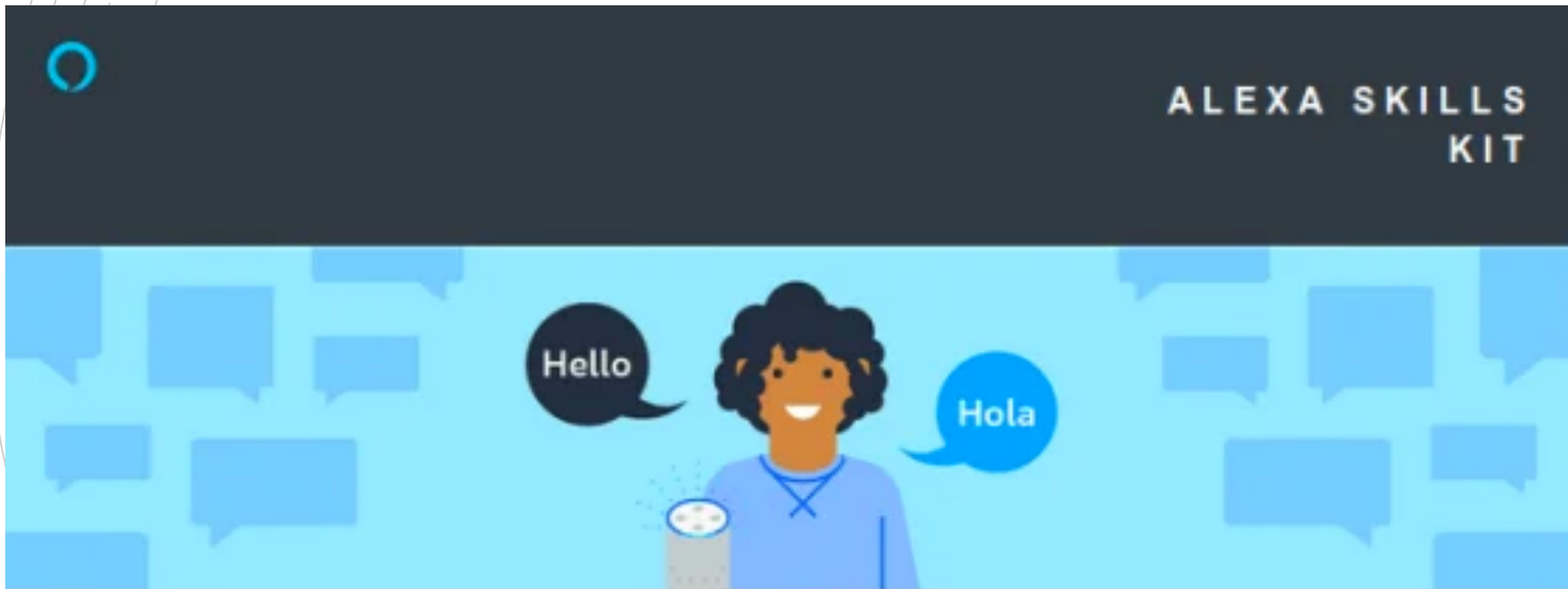
José Luis González Sánchez

@joseluisgonsan



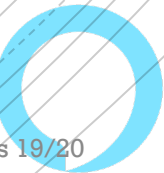
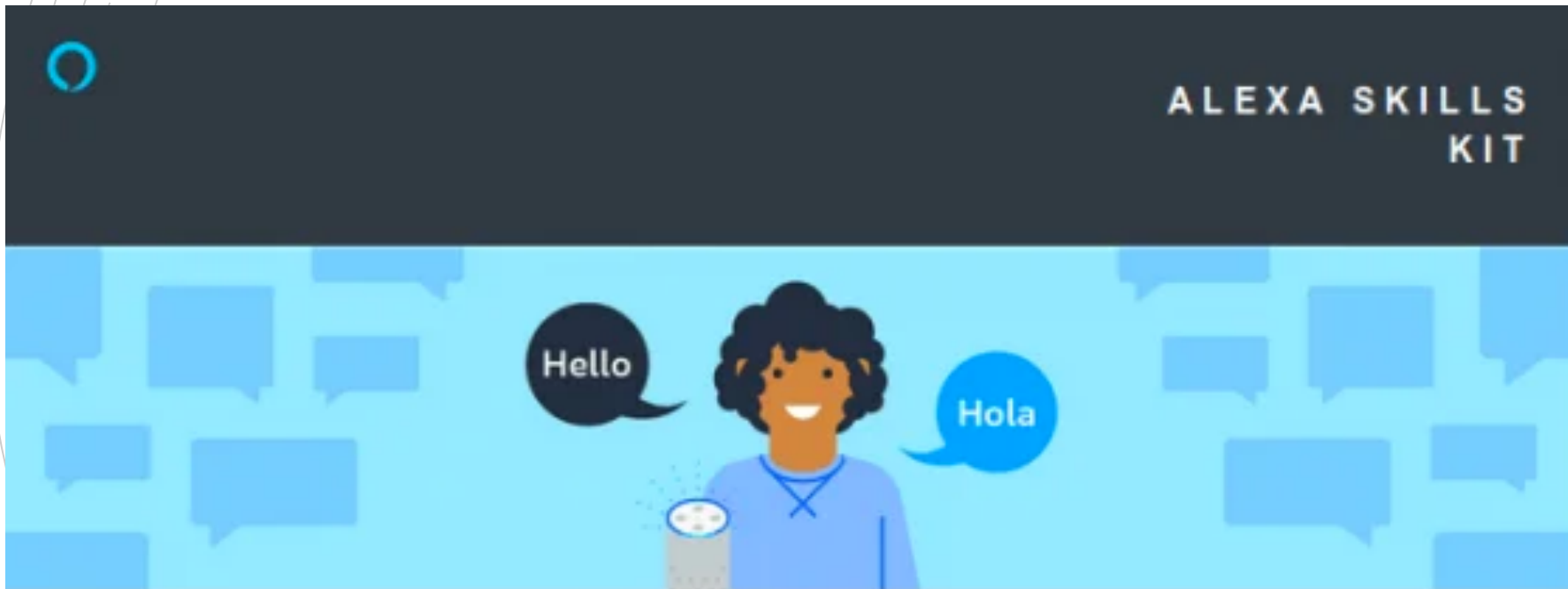
# ¿Qué vamos a aprender?

- Supongamos que queremos listar todos los grados o titulaciones que tenemos.
- O simplemente queremos que para un curso o titulación listar todos sus grados.
- O si le preguntamos por un módulo nos de información del mismo (opcional).



# ¿Qué vamos a aprender?

- Nos basaremos en este tutorial
- <https://developer.amazon.com/es-ES/alexa/alexa-skills-kit/get-deeper/tutorials-code-samples/build-an-engaging-alexa-skill/module-4>



# Comenzamos

- Creamos una Skill en Español basada en HelloWorld y la llamamos Feliz Cumple y la almacenamos en tipo Servidor Node.js
- Cambiamos nuestro Invocation name a feliz cumple
- Borramos el HelloWorldIntent y creamos nuestro propio Intent, llamado RegistrarCumpleIntent.
  - Para ello podemos hacer uso de un Custom Intent personalizado o usar alguno de los preestablecidos por Amazon

**CUSTOM**

- Interaction Model
- Utterance Conflicts (0)
- Invocation
- Intents (4) + Add
  - Built-In Intents (4)
    - AMAZON.CancelIntent
    - AMAZON.HelpIntent
    - AMAZON.StopIntent
    - AMAZON.NavigateHomeIntent
- Slot Types (0) + Add
- JSON Editor

**Interfaces**

- Endpoint
- Intent History
- Annotation Sets
- Display [↗](#)

### Add Intent

An intent represents an action that fulfills a user's spoken request. [Learn more](#) about intents.

☒ Create custom intent <sup>?</sup>

Create custom intent

☐ Use an existing intent from Alexa's built-in library <sup>?</sup>

[Learn more](#) about using built-in intents.

🔍

25/25 built-ins

| Name                            | Description  |
|---------------------------------|--|
| <b>Standard</b><br>25 built-ins | Intents for common actions such as stopping, canceling, and asking for help. |
| AMAZON.CancelIntent             | <span>added</span> <span>view</span>   |
| AMAZON.FallbackIntent           | <span>+ Add Intent</span>  |
| AMAZON.HelpIntent               | <span>added</span> <span>view</span>   |
| AMAZON.LoopOffIntent            | <span>+ Add Intent</span>  |

# Sentencias

- Indicamos una serie de sentencias para esta intención

**CUSTOM**

- Interaction Model
- Utterance Conflicts (0)
- Invocation
- Intents (5) [+ Add](#)
  - RegistrarCumpleIntent** [+ Add](#)
  - Built-In Intents (4)
    - AMAZON.CancelIntent
    - AMAZON.HelpIntent
    - AMAZON.StopIntent
    - AMAZON.NavigateHomeIntent
- Slot Types (0) [+ Add](#)
- JSON Editor
- Interfaces
- Endpoint

Updates to sample utterances qualify for instant live updates. [Learn more](#) about live updates to your skill.

## Intents / RegistrarCumpleIntent

Sample Utterances (5) [?](#) [Bulk Edit](#) [Export](#)

What might a user say to invoke this intent? [+](#)

|                        |                   |
|------------------------|-------------------|
| recuerda mi cumpleaños | <a href="#">+</a> |
| graba mi cumpleaños    | <a href="#">+</a> |
| graba mi cumple        | <a href="#">+</a> |
| registra mi cumple     | <a href="#">+</a> |
| registra mi cumpleaños | <a href="#">+</a> |

1 – 5 of 5 [Show All](#)

# Argumentos

- Un slot no es más que una información que puede variar en la petición de un usuario, en el utterance del intent, y que necesitamos conocer. En la [documentación de Amazon](#) podéis ver un ejemplo detallado. Se puede:
  - Elegir uno de los [tipos predefinidos por Amazon](#). Estos tipos ya saben manejar ciertos valores de entrada, como fechas o palabras comunes en ciertos contextos como nombres de ciudades o tipos de comida. Los valores que aceptan estos tipos se pueden extender en algunos casos. La documentación oficial indica cuáles.
  - Crear un type nuevo. En este tendríamos que darle un nombre al tipo y proveer de una lista de valores que debe manejar.

Dialog Delegation Strategy ?

Dialog management is not enabled f... > Why is this disabled?

Intent Slots (1) ?

| ORDER ? | NAME ?            | SLOT TYPE ?            | ACTIONS              |
|---------|-------------------|------------------------|----------------------|
| 1       | dia               | AMAZON.NUMBER ✓        | Edit Dialog   Delete |
| 2       | Create a new slot | + Select a slot type ✓ | Edit Dialog   Delete |

Intent Confirmation

# Argumentos

- Para establecer un slot personalizado, nos vamos al menú de la izquierda y donde pone slot types le damos a add. Podríamos extender alguno de los existentes, como mes, pero vamos a hacerlo desde cero

## Add Slot Type

Slot types define how data in an intent slot is recognized and handled. All intent slots must be assigned a slot type. [Learn more](#) about using slot types.

- ☒ Create custom slot type <sup>?</sup>

mes

Create custom slot type



- ☐ Use an existing slot type from Alexa's built-in library <sup>?</sup>

[Learn more](#) about using built-in slot types.

Search built-ins



39/39 built-ins

| Name |   | Description  |
|------|---|--|
| >    |  <b>List Types</b><br>31 built-ins               | These slot types each represent a list of items. You can extend these slot types with additional values. |
| >    |  <b>Numbers, Dates, and Times</b><br>8 built-ins | These slot types that convert the user's utterance into data types such as numbers and dates.            |



# Argumentos

- Podemos darle el valor, y ponerle un ID y agregarle un sinónimo.
- Asignamos este tipo al slot mes.

The screenshot displays the Amazon Lexa console interface. On the left, a sidebar shows the navigation menu with 'Intents (5)' and 'Slot Types (2)'. The 'RegistrarCumpleIntent' intent is selected, and its slot type is set to 'TipoMes'. The main area shows the 'Slot Types / TipoMes' configuration page. A light blue box at the top provides information about catalog management. Below this, the 'Slot Values (12)' section is visible, showing a table with columns for 'VALUE', 'ID (OPTIONAL)', and 'SYNONYMS (OPTIONAL)'. The table lists 'ocutubre' with ID '10' and 'diciembre' with ID '12'. A 'Bulk Edit' button and a search bar are also present. A modal window is open, showing the 'Intents / RegistrarCumpleIntent / mes' configuration, where the 'Slot Type' is set to 'TipoMes'.

Invocation

- Intents (5)
  - RegistrarCumpleIntent
    - dia
    - mes
  - Built-In Intents (4)
    - AMAZON.CancelIntent
    - AMAZON.HelpIntent
    - AMAZON.StopIntent
    - AMAZON.NavigateHomeIntent
  - Slot Types (2)
    - AMAZON.NUMBER
    - TipoMes

Slot Types / TipoMes

Use Catalog management for managing slot types with large, constantly changing slot values. Catalog management is currently available only on Alexa Skill Management API (SMAP) and the Alexa Skills Kit Command Line Interface (ASK CLI). [Learn more about Catalog management.](#)

Slot Values (12) ?

Bulk Edit Export Search

Enter a new value for this slot type +

| VALUE ?   | ID (OPTIONAL) ? | SYNONYMS (OPTIONAL) ? |
|-----------|-----------------|-----------------------|
| ocutubre  | 10              | décimo mes de         |
| diciembre | 12              | duodécimo mes         |

Intents / RegistrarCumpleIntent / mes

Slot Type

TipoMes

Auto delegation is disabled for this intent (set in intent setting).

Dialogs Validations



# Argumentos

- Creamos el slot anno (año, no entiende las n 😊)
- Y. empezamos a crear frases que detecten slots

The screenshot displays the Amazon Lex console interface for configuring an intent. On the left sidebar, under 'Intents (5)', the 'RegistrarCumpleIntent' intent is selected. Below it, three slots are listed: 'dia' (blue), 'mes' (orange), and 'anno' (red). Under 'Built-In Intents (4)', several Amazon intents are listed. Under 'Slot Types (3)', 'AMAZON.NUMBER' and 'TipoMes' are visible. The main area shows 'Sample Utterances (13)' with a list of phrases containing slot placeholders. The first sample is '{dia} de {mes} de {anno}'. Other samples include '{dia}', 'cumpló el {mes}', 'mi cumpleaños es el {dia}', and 'mi cumpleaños es el {dia} de {mes} de {anno}'. At the bottom, the 'Dialog Delegation Strategy' section is partially visible.

Intents (5) [+ Add](#)

- RegistrarCumpleIntent
- dia
- mes
- anno

Built-In Intents (4)

- AMAZON.CancelIntent
- AMAZON.HelpIntent
- AMAZON.StopIntent
- AMAZON.NavigateHomeIntent

Slot Types (3) [+ Add](#)

- AMAZON.NUMBER
- TipoMes
- AMAZON.FOUR\_DIGIT\_NUMBER

Sample Utterances (13) [?](#) [Bulk Edit](#) [Export](#)

What might a user say to invoke this intent? [+](#)

- {dia} de {mes} de {anno}
- {dia}
- cumpló el {mes}
- mi cumpleaños es el {dia}
- mi cumpleaños es el {dia} de {mes} de {anno}

1 – 5 of 13 [Show All](#)

Dialog Delegation Strategy [?](#)

# Prompts

- A la hora de usar un slot en nuestro skill podremos editar algunos aspectos del type o el comportamiento del mismo en el intent.
- Con respecto al type, como os dije antes, podemos añadir valores extra en algunos slots ya predefinido por Amazon. Con Amazon.DATE no se puede pero, por ejemplo, si usamos uno de tipo Amazon.FOOD sí sería posible.
- Por otro lado, para el uso del slot por el intent, existen opciones a configurar. Para acceder a ellas solo tenemos que: o clickar en el nombre del slot bajo el intent (desplegable del interaction model), o sobre "Edit Dialog" en la tabla donde se registran los slots para un intent. En ambos casos se llega a la misma pantalla.
- Podemos indicar si un argumento es obligatorio, o obtener frases que ayuden a completarlo.
- Con ello creamos sentencias específicas para un slot (argumento)
- Podemos usar otro slot detectado (mira el ejemplo de mes)

The screenshot displays the Amazon Alexa Developer Console interface for configuring a slot named 'dia'. On the left, a sidebar lists various components: 'dia' (selected), 'mes', 'anno', 'month', 'Built-In intents (4)', 'Slot Types (3)', 'JSON Editor', 'Interfaces', 'Endpoint', 'Intent History', 'Annotation Sets', 'Display', 'IN-SKILL PRODUCTS', and 'ACCOUNT LINKING'. The main area is divided into two tabs: 'Dialogs' and 'Validations'. The 'Dialogs' tab is active, showing the 'Slot Filling' section. A toggle switch for 'Is this slot required to fulfill the intent?' is turned on. Below this, the 'Alexa speech prompts' section contains a table with four prompts: 'vale, me has dicho que cumples en {month} pero qué día?', 'qué día cumples años?', 'en qué día es tu cumpleaños?', and 'dime en qué día del mes es tu cumpleaños'. Each prompt has a trash icon for deletion. The 'User utterances' section below it contains a table with three utterances: 'el día {dia}', 'el {dia}', and '{dia}', each with a trash icon. Navigation arrows at the bottom right indicate '1 - 4 of 4' for prompts and '1 - 3 of 3' for utterances.

# Prompts

- Si salvamos el Modelo, y abrimos el JSON, veremos que tenemos el slot día, sentencias propias (samples).
- Además tenemos un nuevo apartado de dialog (al final) llamado dialog, con intenciones de elicitación de día con el ID del prompt a usar.

```
"slots": [  
  {  
    "name": "día",  
    "type": "AMAZON.NUMBER",  
    "samples": [  
      "el día {dia}",  
      "el {dia}",  
      "{dia}"  
    ],  
  },  
  ....  
]
```

```
"dialog": {  
  "intents": [  
    {  
      "name": "RegistrarCumpleIntent",  
      "confirmationRequired": false,  
      "prompts": {},  
      "slots": [  
        {  
          "name": "dia",  
          "type": "AMAZON.NUMBER",  
          "confirmationRequired": false,  
          "elicitationRequired": true,  
          "prompts": {  
            "elicitation": "Elicit.Slot.801747796238.91610710291"  
          }  
        }  
      ],  
    },  
    ....  
  ],  
  ..  
}
```

```
prompts": [  
  {  
    "id": "Elicit.Slot.801747796238.91610710291",  
    "variations": [  
      {  
        "type": "PlainText",  
        "value": "dime el día"  
      },  
      {  
        "type": "PlainText",  
        "value": "qué día"  
      },  
      {  
        "type": "PlainText",  
        "value": "vale, qué día"  
      }  
    ],  
  },  
  ...  
]
```

- Repetimos para mes. Lo rellenamos como siguiendo la misma idea
- Repetimos con año.



# Validaciones

- Las validaciones nos permiten filtrar los valores de un slot para decir si son correctos o no.
- Se puede indicar la restricción y la frase que dirá Alexa para corregirnos.
- Probamos a poner restricciones a día (entre 1 y 31), meses (1, 12, o enero a diciembre) y años.

The screenshot shows the Amazon Lexa console interface. On the left, a sidebar contains navigation options: 'dia' (selected), 'mes', 'anno', 'Built-In Intents (4)' (with sub-items: AMAZON.CancelIntent, AMAZON.HelpIntent, AMAZON.StopIntent, AMAZON.NavigateHomeIntent), 'Slot Types (3)' (with sub-items: TipoMes, AMAZON.NUMBER, AMAZON.FOUR\_DIGIT\_NUMBER), and 'JSON Editor'. The main area is titled 'Validations' and shows the configuration for a slot named 'dia'. It includes a 'Create new Validation' section with a dropdown menu and a message: 'A valid slot type is needed before you can add a validation'. Below this, the 'Active Validations' section lists two validations. The first validation is 'Greater than or equal to a number' with the value '1' and the description 'ASK\_CIM\_EDIT\_SLOT\_VALIDATION\_DESCRIPTION\_GREATER\_THAN\_OR\_EQUAL\_TO\_NUMBER'. The second validation is 'Less than or equal to a number' with the value '31' and the description 'ASK\_CIM\_EDIT\_SLOT\_VALIDATION\_DESCRIPTION\_LESS\_THAN\_OR\_EQUAL\_TO\_NUMBER'. Both validations have a text input field for the user's value and a plus sign button to add more validations. The interface also shows a 'Dime un día entre uno y treinta y uno' message for both validations.

# Validaciones

- En meses, como es un tipo personalizado podemos hacer uso de sinónimos para su validación
- Recordad que hay que darle a Salvar el modelo y a Construir el modelo para que esté operativo y se pueda entrenar.

The screenshot displays the Amazon Lex console interface for configuring a slot type. On the left, a sidebar shows the navigation menu with 'CUSTOM' selected. Under 'Intents (5)', 'RegistrarCumpleIntent' is expanded, and 'mes' is selected. Below it, 'Slot Types (3)' is also expanded, showing 'TipoMes', 'AMAZON.NUMBER', and 'AMAZON.FOUR\_DIGIT\_NUMBER'. The main panel shows the 'Intents / RegistrarCumpleIntent / mes' path. The 'Slot Type' dropdown is set to 'TipoMes'. A message indicates 'Auto delegation is on for this intent (inherited from skill setting)'. The 'Validations' tab is active, showing 'Create new Validation:' with a 'Choose a validator' dropdown and a note: 'A valid slot type is needed before you can add a validation'. Under 'Active Validations:', there is a dropdown set to 'Value within slot types' slot values' with a note: 'Here is some description about value within slot types' slot values'. Below this, there is a text input field with the placeholder 'What might Alexa say to prompt for a valid slot value?' and a plus icon to add more. At the bottom, there is a text input field with the value 'Por favor, dime un mes válido' and a trash icon to remove it. The bottom right corner shows a pagination indicator '1 - 1 of 1'.

- Renombramos HelloWorldIntent en todos lados por RegistrarCumpleIntent (incluido su handler)
- ¿Cómo procesamos los prompts del código?
- Cada request a un intent tiene acceso a un mapa de slots. Las entradas de los valores de ese mapa son los nombres de slot que se pusieron en el modelo.
- A partir del valor, dependiendo del tipo del slot, habría que hacer la lógica oportuna de interpretarlo.

```
const RegistrarCumpleIntentHandler = {
  canHandle(handlerInput) {
    return Alexa.getRequestType(handlerInput.requestEnvelope) === 'IntentRequest'
      && Alexa.getIntentName(handlerInput.requestEnvelope) === 'RegistrarCumpleIntent';
  },
  handle(handlerInput) {
    // Obtenemos el Intent a través del HandlerInput
    const {requestEnvelope} = handlerInput;
    const {intent} = requestEnvelope.request;

    //Almacenamos el contenido del slot en variables
    const dia = Alexa.getSlotValue(requestEnvelope, 'dia');
    const anno = Alexa.getSlotValue(requestEnvelope, 'anno');
    const mes = Alexa.getSlotValue(requestEnvelope, 'mes');
    // Luego veremos como hacerlo más fácil
    // Construimos la salida
    const speakOutput = "Tu fecha de cumpleaños es el día ' + dia + ' del mes de ' + mes + ' de ' + anno;

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



# CODE

- Es importante que te asegures que tu nuevo controlador (RegistrarCumpleHandler) está incluido en `exports.handler = Alexa.SkillBuilders.custom()` para que se pueda ejecutar en el orden que tú indiques.

```
exports.handler = Alexa.SkillBuilders.custom()
    .addRequestHandlers(
        LaunchRequestHandler,
        RegistrarCumpleIntentHandler,
        HelpIntentHandler,
        CancelAndStopIntentHandler,
        FallbackIntentHandler,
        SessionEndedRequestHandler,
        IntentReflectorHandler)
    .addErrorHandlers(
        ErrorHandler)
    .addRequestInterceptors(
        LocalisationRequestInterceptor,
        LoggingRequestInterceptor)
    .addResponseInterceptors(
        LoggingResponseInterceptor)
    //.withCustomUserAgent('sample/happy-birthday/mod3')
    .lambda();
```



# Probando

- Ya puedes probar tu Skill

[illegible]



# Mejorando el código

- Te aconsejo que mires el código con las mejoras. 2.5
- Vamos a añadir un disparador para ir al evento de indicar el cumpleaños directamente y con ellos mejoramos el diálogo guiando al usuario. Lo hacemos en `LaunchRequestHandler`.

```
addDelegateDirective({  
    name: 'RegistrarCumpleIntent',  
    confirmationStatus: 'NONE',  
    slots: {}  
})
```

- Hemos usado cadenas parametrizadas y el traductor
  - Añadimos la biblioteca en `index.js`: `const i18n = require('i18next');` y en el `package.json`: `"i18next": "^15.0.5"`
- Creamos la matriz de cadenas de mensajes internacionales `languageStrings` con los mensajes globales. Podemos usarla para varios idiomas
- Creamos el interceptor `LocalisationRequestInterceptor`, que nos permite coger de esa matriz de cadenas de localización el mensaje adecuado según el idioma de nuestra aplicación



# Mejorando el código

- Creamos interceptores de entrada y salida de peticiones para analizar logs
- Creamos el IntentReflector. se utiliza para probar y depurar modelos de interacción. Simplemente repetirá la intención que dijo el usuario. Puedes crear manejadores personalizados para tus intentos. Definiéndolos arriba, luego también agregándolos a la cadena de manejador de solicitudes a continuación.
- Añadimos FallbackIntentHandler. se activa cuando un cliente dice algo que no se asigna a ninguna intención en su habilidad. También debe definirse en el modelo de idioma (si la configuración regional lo admite). Este controlador se puede agregar de forma segura, pero se ignorará en las configuraciones regionales que aún no lo admiten
- Añadimos simpleCard para mostrar algo de información en la pantalla.



# Mejorando el código

- Uno de los aspectos fundamentales es aprender a depurar y ver que pasa.
- La primera opción es probar el Frontend. Lo podemos hacer con la herramienta de Evaluate Model de Build. Con ella podemos simular el diálogo y como reacciona antes de pasar al Backend

The screenshot shows the AWS Lex console interface. At the top, there's a navigation bar with tabs: < Your Skills, Informatica Virgen de Gracia, Build, Code, Test, Distribution, Certification, Analytics, and Provide feedback. Below this, there's a toolbar with buttons: Save Model, View Model Versions, Build Model, Update live skill, and Evaluate Model (highlighted with an orange box). The main content area is divided into two sections. On the left, under 'CUSTOM', there's a list of intents under the 'Intents (16)' category. The list includes: CreadorIntent, IniciolIntent, ContactolIntent, DireccionIntent, TelefonolIntent, CorreoIntent, and RegistrarCursoIntent (which is expanded to show slots: curso and ciclo). On the right, the 'Utterance Profiler' tool is active, showing a sample utterance: 'Me has dicho que tu curso es 1.º, ahora necesito saber el ciclo entre ASIR, DAWI, DAW y SMR'. Below this, it shows the resolved slots: 'ciclo: dam' and 'curso: 1.º'.

# Mejorando el código

- Os recomiendo que para depurar la parte de Backen hagáis uso de console.log. Este comando nos permite imprimri en el log. Por ejemplo console.log("Listando módulos de "+ curso);
- Podemos ver los logs etiquetados como INFO desde la pantalla code, en Logs: Amazon CloudWatch (abajo a la izquierda y explorándolos)

loudWatch

aneles

larmas

ALARMA

INSUFICIENTE

OK

Facturación

**registros**

Grupos de registros

Información

létricas

ventos

Reglas

Buses de eventos

erviceLens

Mapa de servicios

Seguimientos

ynthetic

Canaries

ontributor Insights

onfiguración




Logs: Amazon CloudWatch

Media storage: S3 [0/5GB]

Docs: Alexa Hosted Skills

Spanish (Español)

CloudWatch > Grupos de registros > /aws/lambda/c66d8c36-513c-4c9b-89b9-0962ac849d65 > 2020/04/17/[241]b4972bb093674364a9dc5e76dd9ebeca

Expandir todo ☒ Fila ☐ Texto   

Filtrar eventos  todos 16-04-2020 (10:43:20) ▾

| Hora (UTC +00:00)        | Mensaje  |
|--------------------------|--|
| 2020-04-17               |  |
| ▶ 10:42:10               | 2020-04-17T10:42:10.544Z aa6005d7-54ab-4eb1-b7dd-5bc090bf5aa8 INFO Util.s3PreSignedUrl: Media/fondo.jpg URL https://amz      |
| ▶ 10:42:10               | 2020-04-17T10:42:10.545Z aa6005d7-54ab-4eb1-b7dd-5bc090bf5aa8 INFO Outgoing response: {"directives":[{"type":"Alexa.Prese    |
| ▶ 10:42:10               | END RequestId: aa6005d7-54ab-4eb1-b7dd-5bc090bf5aa8  |
| ▶ 10:42:10               | REPORT RequestId: aa6005d7-54ab-4eb1-b7dd-5bc090bf5aa8 Duration: 321.36 ms Billed Duration: 400 ms Memory Size: 512 MB       |
| ▶ 10:42:28               | START RequestId: 0e6cb725-42c5-41d2-97f7-937a371f9fb1 Version: 241   |
| ▶ 10:42:28               | 2020-04-17T10:42:28.007Z 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO Incoming request: {"version":"1.0","session":{"new":false |
| ▼ 10:42:28               | 2020-04-17T10:42:28.008Z 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO Detalle ciclo id: DAM                                     |
| 2020-04-17T10:42:28.008Z | 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO Detalle ciclo id: DAM  |
| ▶ 10:42:28               | 2020-04-17T10:42:28.008Z 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO DAM   |
| ▼ 10:42:28               | 2020-04-17T10:42:28.008Z 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO Desarrollo de Aplicaciones Multiplataforma.               |
| 2020-04-17T10:42:28.008Z | 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO Desarrollo de Aplicaciones Multiplataforma.  |
| ▶ 10:42:28               | 2020-04-17T10:42:28.008Z 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO listando todos modulos de curso 1.º del ciclo DAM         |
| ▶ 10:42:28               | 2020-04-17T10:42:28.008Z 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO DAM   |
| ▶ 10:42:28               | 2020-04-17T10:42:28.008Z 0e6cb725-42c5-41d2-97f7-937a371f9fb1 INFO primero   |

# Conclusiones y Ejercicio

- Ahora puedes hacer un proceso de diálogo más complejo y sofisticado.
- Ejercicio: Listar todos las titulaciones del departamento, listar todos los módulos de un curso, y el número de cursos de una titulación
- Código de ejemplo tutorial Amazon: <https://github.com/germanviscuso/ASKVideoSeries/tree/master/03>
- Código de la Skill Informática: <https://github.com/joseluisgs/informatica-skill>
- Tutoriales seguidos:
  - <https://youtu.be/N9vCxjc5qXI>

