

**CARRERA PROFESIONAL DE ANÁLISIS DE  
SISTEMAS**

**MÓDULO SOPORTE TÉCNICO Y OFIMÁTICO**



**VALLE GRANDE**

INSTITUTO DE EDUCACIÓN SUPERIOR TECNOLÓGICO PRIVADO

**Servicio Cognitivo de Azure - Qna Maker**

**Estudiante**

**JHIANPOL MAXIMILIANO RAMOS GIL**

**Responsable**

**Giancarlo Valencia Sotomayor**

**Cañete - Perú**

**Julio de 2022**

# **TABLA DE CONTENIDO**

1.	Activar Sandbox Azure e ingresar al portal de Azure .....	3
2.	Creación de un QnA Maker.....	5
3.	Conociendo el portal de QnA Maker.....	6
4.	Guardar y entrenar nuestro QnA Maker.....	8
5.	Creando nuestro Bot en AZURE.....	10
6.	Probar en Postman nuestro QnA Marker.....	13
7.	Recibiendo respuesta en POSTMAN.....	15
8.	Consumir servicio De QnA en Java usando el Apache NetBeans IDE 12.3	17
9.	Creación de Proyecto.....	17
10.	Añadir dependencias.....	18
11.	Creación de Paquetes.....	20
12.	Creación de clases dentro de los paquetes.....	20
13.	Para el modelo.....	20
14.	Para el Services.....	21
15.	Creación del controller.....	25
16.	Configuración del web.xml para correr proyecto .....	
17.	Run al proyecto.....	28
18.	Conclusiones.....	29
19.	Consideraciones.....	29
20.	Anexos.....	30
21.	Referencias.....	30

# Activar Sandbox Azure y ingresar al portal de Azure

[Link para activar Sandbox](#)

Ejercicio:  
**Configuración de la base de datos y el contenedor de Azure Cosmos DB**

10 minutos

Espacio aislado activado Tiempo restante:  
**3 h 59 min.**

Ha usado 1 de los 10 espacios aislados de hoy. Mañana habrá disponibles más espacios aislados.

Luego nos tenemos que dirigir al Portal de Azure  
[Link de la página del Portal de Azure](#)

Ingresamos con la cuenta hotmail que tenemos creada  
Click en el 2do (cambiar de directorio)

Microsoft

Sign outdesconectar

diretorio predeterminado

axelitho\_ramosaxeli...

axelitho\_beba@hotmail.com...  
mi cuenta microsoftmi cuenta...

Cuentas de almacenamiento API de cara

Switch directory Cambiar dir ...

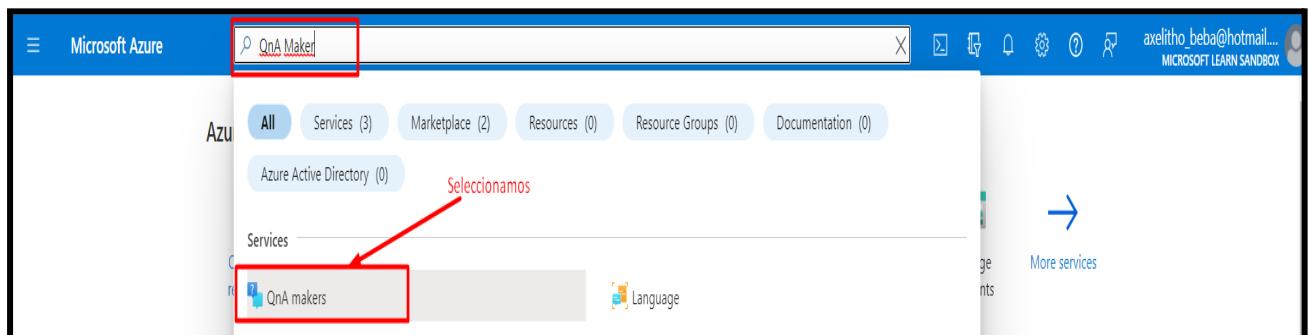
## Seleccionamos en cambiar, para usar el servidor de Sandbox

The screenshot shows the Azure portal's configuration page for directories and subscriptions. On the left, there's a sidebar with options like 'Menú de búsqueda', 'Directorios + suscripciones' (which is selected), 'Apariencia + vistas de inicio', 'Idioma + región', 'Mi información', and 'Cerrar sesión + notificaciones'. The main area displays subscription information: 'Azure para estudiantes - ¿No ves una suscripción? Cambiar a otro directorio.' Below this, it shows the current directory: 'Directorio actual': 'Directorio por defecto' and 'directorio de inicio': 'Última visita (cambiar)'. There are tabs for 'Favoritos' and 'Todos los directorios', with 'Todos los directorios' currently selected. A search bar is at the top. At the bottom, a table lists directories: 'directorio por defecto' (marked as 'Actual') and 'Zona de pruebas de aprendizaje de Microsoft'. A red arrow points to the 'Cambiar' button next to the 'Actual' dropdown.

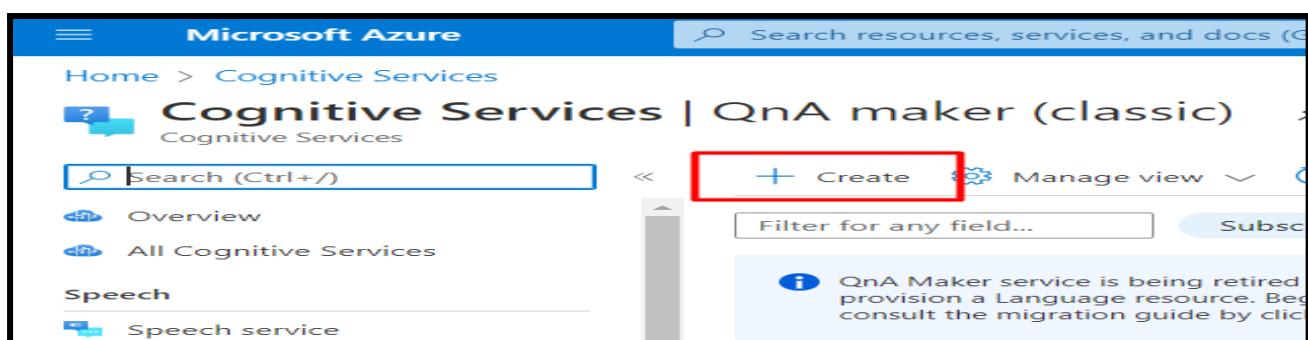
Una vez que se cambió podremos corroborar viendo la parte derecha de arriba en la cuenta

The screenshot shows the Azure portal's homepage. At the top, there's a navigation bar with icons for search, notifications, and settings, followed by the user account information: 'axelitho\_beba@hotmail....' and 'MICROSOFT LEARN SANDBOX', which are both enclosed in a red box. Below the navigation bar, there are three service tiles: 'App Services' (represented by a globe icon), 'Storage accounts' (represented by a teal bar icon), and 'More services' (represented by a blue arrow icon).

En la parte de arriba escribimos el nombre del servicio = QnA Maker y click en el servicio encontrado



## Creación de un QnA Maker



Rellenamos los siguientes datos y click en crear

A screenshot of the QnA Maker creation form. The form is divided into several sections: "Project details", "Azure Search details - for data", and "App Service details - for runtime". The "Project details" section includes fields for "Subscription" (Concierge Subscription), "Resource group" (learn-c9426f2c-ceb4-4ab2-af4e-83290de671cb), "Name" (PreguntasDemo1), and "Pricing tier" (Standard S0). The "Azure Search details" section includes "Azure Search location" (US East US) and "Azure Search pricing tier" (Basic B (15 Indexes)). The "App Service details" section has a note: "When you create a QnAMaker resource, you host the runtime in your own Azure subscription. App Service is the". At the bottom, there are two buttons: "Review + create" (highlighted with a red box) and "Next : Tags >".

Nos dirigimos al inicio del portal de azure y veremos los recursos creados.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar labeled "Buscar recursos, servicios y documentos (G+/" followed by several icons for account management. Below the header, the "Servicios de Azure" section is visible, featuring a "Crear un recurso" button and various service icons like QnA Maker, Cognitive Services, and API Management. The "Resources" section shows a table of recent resources, with one entry highlighted: "PreguntasDemo1" (QnA Maker) was last updated "hace 1 minutos".

## Conociendo el portal de QnA Maker

Enlace : [Link del portal de QnA Maker](#)

Si no tenemos cuenta de QnA Maker, nos registramos con la misma cuenta de hotmail.

Ahora le daremos click en Crear una base de conocimiento

The screenshot shows the "Creador de QnA" (QnA Maker Creator) interface. At the top, there are tabs for "Servicios cognitivos", "Creador de QnA", and "mis bases de conocimiento". A red box highlights the "Crear una base de conocimiento" button. To the right, a sidebar titled "Perfil" shows the user's email (axelitho\_beba@hotmail.com), name (axelitho ramos), service configuration, and disconnect options. Another red arrow points to the "Sigo usando Sandbox" button at the bottom right of the main panel. The main panel features a large title "De datos a bot en minutos" and a description about creating and publishing a bot using frequent questions, support sites, product manuals, SharePoint documents, or editorial content via a user-friendly interface or API REST. A "Empezar" button is also visible.

## Crear una base de conocimiento

Cree un servicio de Azure para su base de conocimiento de QnA y agregue fuentes que contengan los pares de preguntas y respuestas que le genera. [Obtenga más información sobre cómo crear una base de conocimientos.](#)

PASO 1

### Cree un servicio QnA en Microsoft Azure.

Cree un servicio Azure QnA para su KB. Si ya tiene un servicio Azure QnA para esta KB, omita este paso. Seleccione "Vista previa" para probar la función de respuesta a preguntas personalizadas (versión preliminar). [Obtenga más información sobre las suscripciones, los niveles de precios y las claves de Azure.](#)

[Crear un servicio QnA](#)

Si no lo creaste en azure, le das click y te llevará a Azure y te hará crearlo. De lo contrario si lo tenemos omitimos este paso.

- ⓘ Le recomendamos encarecidamente que use la función [Respuesta](#) a preguntas (ahora disponible de forma general) en Azure Cognitive Service for Language. Puede visitar [Language Studio](#) para crear un recurso de idioma.

PASO 2

### Conecte su servicio QnA a su KB.

Después de crear un servicio Azure QnA, actualice esta página y luego seleccione su servicio Azure usando opciones a continuación

[Actualizar](#)

## Le damos click en Actualizar

PASO 2

### Conecte su servicio QnA a su KB.

Después de crear un servicio Azure QnA, actualice esta página y luego seleccione su servicio Azure usando opciones a continuación

[Actualizar](#)

\* Id. de directorio de Microsoft Azure

Microsoft Learn Sandbox

\* Nombre de suscripción de Azure

Concierge Subscription

\* Servicio Azure QnA

Seleccionar servicio

\* Idioma

español

CLICK

Seleccionamos el servidor de Sandbox y el nombre de la suscripción, el servicio que hemos creado en azure de QnA Maker y el idioma.

Servicios cognitivos | Creador de QnA mis bases de conocimiento Crear una base de conocimiento

PASO 3 Asigne un nombre a su KB.  
El nombre de la base de conocimiento es para su referencia y puede cambiarlo en cualquier momento.

\* Nombre  
 Agregamos un nombre

---

PASO 4 Rellene su KB.  
Extraiga pares de preguntas y respuestas de preguntas frecuentes en línea, manuales de productos u otros archivos. Los formatos admitidos son .tsv, .pdf, .doc, .docx, .xlsx, que contienen preguntas y respuestas en secuencia. [Obtenga más información sobre las fuentes de la base de conocimientos.](#) Omita este paso para agregar preguntas y respuestas manualmente después de la creación. La cantidad de fuentes y el tamaño del archivo que puede agregar depende de la SKU del servicio QnA que elija. [Obtenga más información sobre las SKU de QnA Maker.](#)

Habilitamos  Habilita la extracción de varios turnos desde URL, archivos .pdf o .docx. [Aprende más.](#)

\* Texto predeterminado de varias vueltas ?

URL

si gustas agregas url y si no solo localmente  Agregamos un archivo de preguntas y respuestas dentro de nuestra pc, con extensión .pdf o .docx.

Servicios cognitivos | Creador de QnA mis bases de conocimiento Crear una base de conocimiento

+ Agregar archivo

Charla  
Dale a tu bot la capacidad de responder miles de preguntas triviales con una voz que se adapte a tu marca. Cuando agrega una charla a su base de conocimientos seleccionando una personalidad a continuación, las preguntas y respuestas se agregarán automáticamente a su base de conocimientos y podrá editarlas en cualquier momento que desee. [Obtenga más información sobre charlar.](#)

Ninguna  
 Profesional  
 Amigable  
 Ingenioso  
 Cuidando  
 Entusiasta

---

PASO 5 Crea tu KB  
La herramienta revisará sus documentos y creará una base de conocimiento para su servicio. Si no está utilizando un documento existente, la herramienta creará una tabla de base de conocimiento vacía que puede editar.

Procedemos a seleccionar ninguna y dar click en crear tu KB

## Guardar y entrenar nuestro QnA Maker



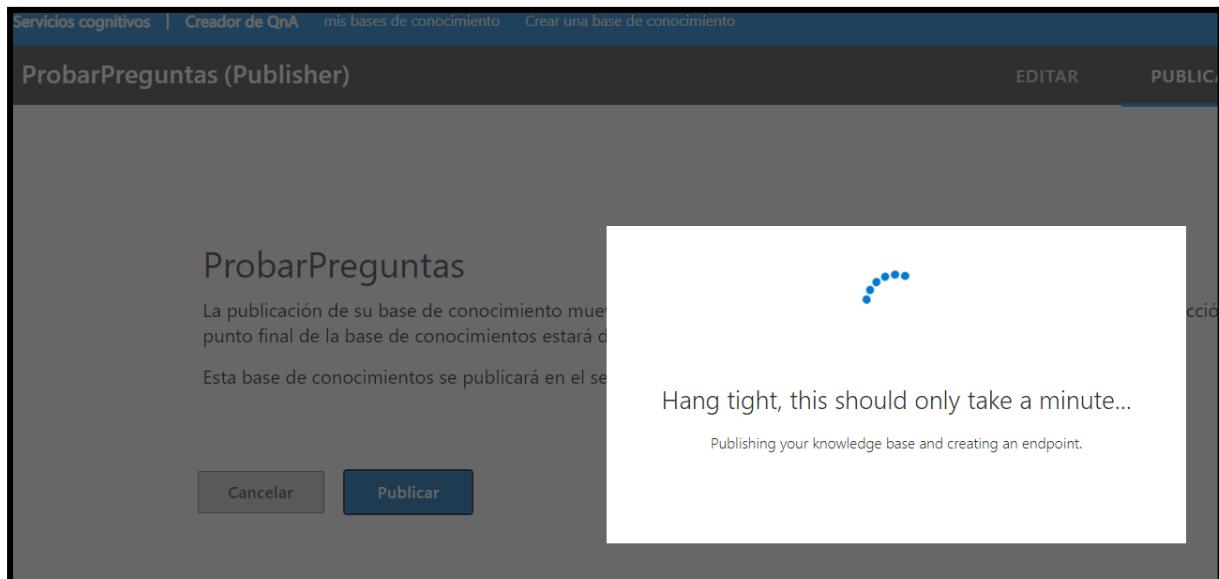
Esperamos :



Una vez que entrenamos procedemos a publicar nuestro QnA



A screenshot of the 'ProbarPreguntas' (Publisher) page. The top navigation bar has links for 'Servicios cognitivos', 'Creador de QnA', 'mis bases de conocimiento', and 'Crear una base de conocimiento'. Below the bar, the title 'ProbarPreguntas (Publisher)' is shown, along with buttons for 'EDITAR', 'PUBLICAR' (highlighted with a red box), 'AJUSTES', and 'Ahorra y entrena'. The main content area contains the heading 'ProbarPreguntas' and a paragraph explaining the publication process. At the bottom, there are two buttons: 'Cancelar' and 'Publicar' (highlighted with a red box).



Utilice la siguiente solicitud HTTP para llamar a su base de conocimientos.

The screenshot shows a success message: "Success! Your service has been deployed. What's next?". It includes a "Create Bot" button and a link to "View all your bots on the Azure Portal". Below that, it says "Use the below HTTP request to call your Knowledgebase." followed by a "Learn more" link. There are two tabs: "Postman" (which is selected) and "Curl". A red box highlights the "Postman" tab and the API endpoint details. The API endpoint details are as follows:

```
POST /knowledgebases/aba04c1e-f09a-4b99-bfd9-5c2ff1a28b76/generateAnswer
Host: https://preguntasademo1.azurewebsites.net/qnamaker
Authorization: EndpointKey 63a6cb0-c586-4b7a-8991-5fbe704b64ec
Content-Type: application/json
{"question":<Your question>"}
```

At the bottom, there is a link "Need to fine-tune and refine? Go back and keep editing your service." and a "Edit Service" button.

# Creando nuestro Bot en AZURE

Success! Your service has been deployed. What's next?

You can always find the deployment details in your service's settings.

[Create Bot](#)

[View](#) all your bots on the Azure Portal.

Use the below HTTP request to call your Knowledgebase. [Learn more.](#)

Postman    Curl

```
POST /knowledgebases/aba04c1e-f09a-4b99-bfd9-5c2ff1a28b76/generateAnswer
Host: https://preguntasdemo1.azurewebsites.net/qnamaker
Authorization: EndpointKey 63a6cbe0-c586-4b7a-8991-5fbe704b64ec
Content-Type: application/json
{"question":"<Your question>"}
```

Le damos click en Crear bot, para así poder correrlo en Azure y poder interactuar.

Luego nos llevará al portal de azure donde crearemos el bot y rellenamos los siguientes datos y click en crear.

Ubicación \*

Centro de EE. UU.

Nivel de precios \*

Estándar

Nombre de la aplicación \*

preguntasdemo1-bot-81f6

lenguaje SDK \*

C#

Clave de autenticación QnA \*

63a6cbe0-c586-4b7a-8991-5fbe704b64ec

Plan de servicio de la aplicación/Ubicación \*

PreguntasDemo1/Este de EE.UU.

Perspectivas de la aplicación

Crear

## Una vez creado el bot

**BOT CREADO**

**Click en prueba en Web Chat**

Escribimos una pregunta y a continuación probaremos si podemos interactuar de acuerdo a los datos que le hemos cargado.

**Test**

Hello and Welcome

Just now

¿Quién te creó?

The screenshot shows the Microsoft Azure Bot Service interface. On the left, there's a sidebar with various options like Visión general, Registro de actividades, Control de acceso (IAM), Etiquetas, Ajustes, Perfil de bot, Configuración, Canales, Precios, Prueba en Web Chat (which is selected), Cifrado, Redes, Propiedades, Cerraduras, and Vigilancia. The main area is titled 'preguntasdem01-bot| Prueba en Web Chat' and has a 'Test' tab selected. The conversation log shows the following messages:

- Hello and Welcome (3 minutes ago)
- Envie el mensaje (red arrow pointing to this message)
- Mi creador es Jhianpol Ramos (red arrow pointing to this message)
- ¿Quién te creó? (2 minutes ago)

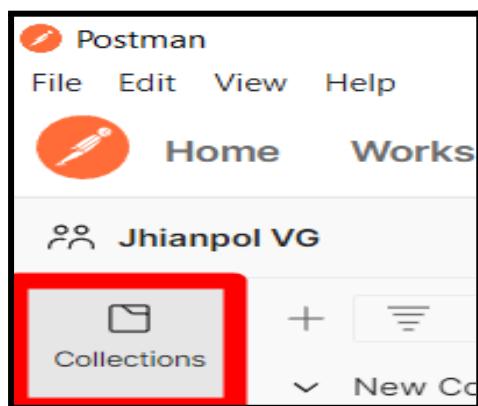
Below the conversation log is a text input field with the placeholder 'Type your message'.

Podemos ver como nos llega la respuesta y sabemos que todo funciona correctamente.

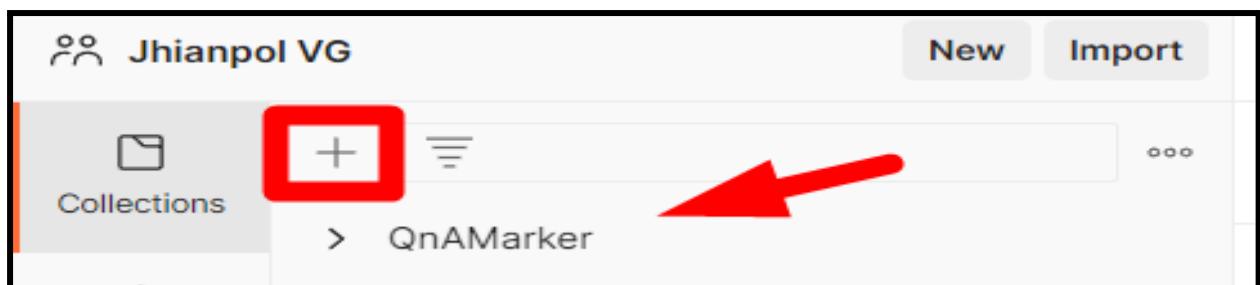
## Probar en Postman nuestro QnA Marker

Abrimos Postman

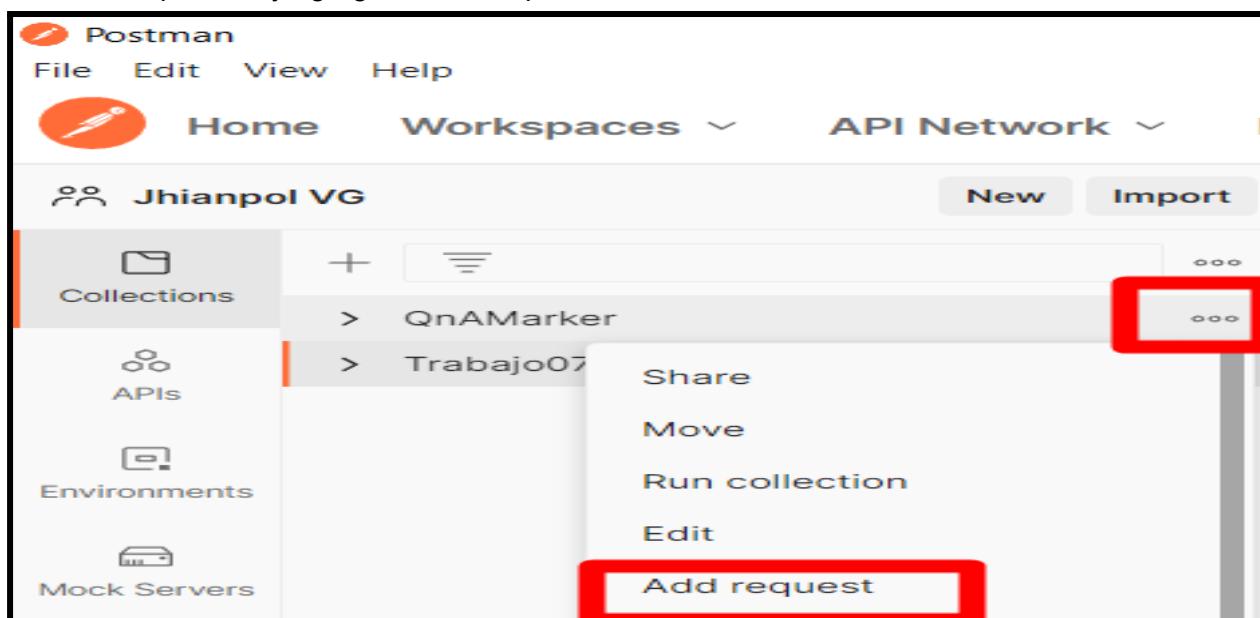
Click en Collections



Procedemos a crear una nueva colección



Click en los puntitos y agregamos un request



Colocaremos todos estos datos en el postman

The screenshot shows the Azure portal page for a QnAMaker knowledgebase. It features a "Create Bot" button and a link to "View all your bots on the Azure Portal". Below that, it says "Use the below HTTP request to call your Knowledgebase." and provides "Postman" and "Curl" examples. A red box highlights the API endpoint URL: "POST /knowledgebases/aba04c1e-f09a-4b99-bfd9-5c2ff1a28b76/generateAnswer".

```
POST /knowledgebases/aba04c1e-f09a-4b99-bfd9-5c2ff1a28b76/generateAnswer
Host: https://preguntasdemo1.azurewebsites.net/qnamaker
Authorization: EndpointKey 63a6cbe0-c586-4b7a-8991-5fbe704b64ec
Content-Type: application/json
{"question":"<Your question>"}
```

The screenshot shows the Postman interface for a new request named "Trabajo07". The method is set to POST, and the URL is https://preguntasdemo1.azurewebsites.net/qnamaker/knowledgebases/aba04c1e-f09a-4b99-bfd9-5c2ff1a28b76/generateAnswer. The Headers tab is selected, displaying two checked headers: Authorization (EndpointKey 63a6cbe0-c586-4b7a-8991-5fbe704b64ec) and Content-Type (application/json). The Body tab is also visible.

The screenshot shows the Postman interface for a new request named "Trabajo07". The method is set to POST, and the URL is https://preguntasdemo1.azurewebsites.net/qnamaker/knowledgebases/aba04c1e-f09a-4b99-bfd9-5c2ff1a28b76/generateAnswer. The Headers tab is selected, showing 12 items. The Body tab is selected, showing the raw JSON input field containing the question: {"question": "¿Cuál es tu nombre?"} and the JSON dropdown menu. A red arrow points from the JSON input field towards the blue "Send" button.

## Recibiendo respuesta en POSTMAN

The screenshot shows a POST request to <https://preguntasdemo1.azurewebsites.net/qnamaker/knowledgebases/aba04c1e-f09a-4b99-bfd9-5cff1a28b76/generateAnswer>. The request body contains the JSON: {"question": "¿Cuál es tu nombre?". The response status is 200 OK, time: 391 ms, size: 1.04 KB. The response body is highlighted with a red box and shows the following JSON structure:

```
1 {
2   "answers": [
3     {
4       "questions": [
5         "¿Cuál es tu nombre?"
6       ],
7       "answer": "Mi nombre es Jhianbot",
8       "score": 94.68,
9       "id": 8,
10      "source": "PreguntasJhianpol1.pdf",
11      "isDocumentText": false,
12      "metadata": [],
13      "context": {
14        "isContextOnly": false,
15        "prompts": []
16      }
17    }
18  ],
19  "activeLearningEnabled": false
20 }
```

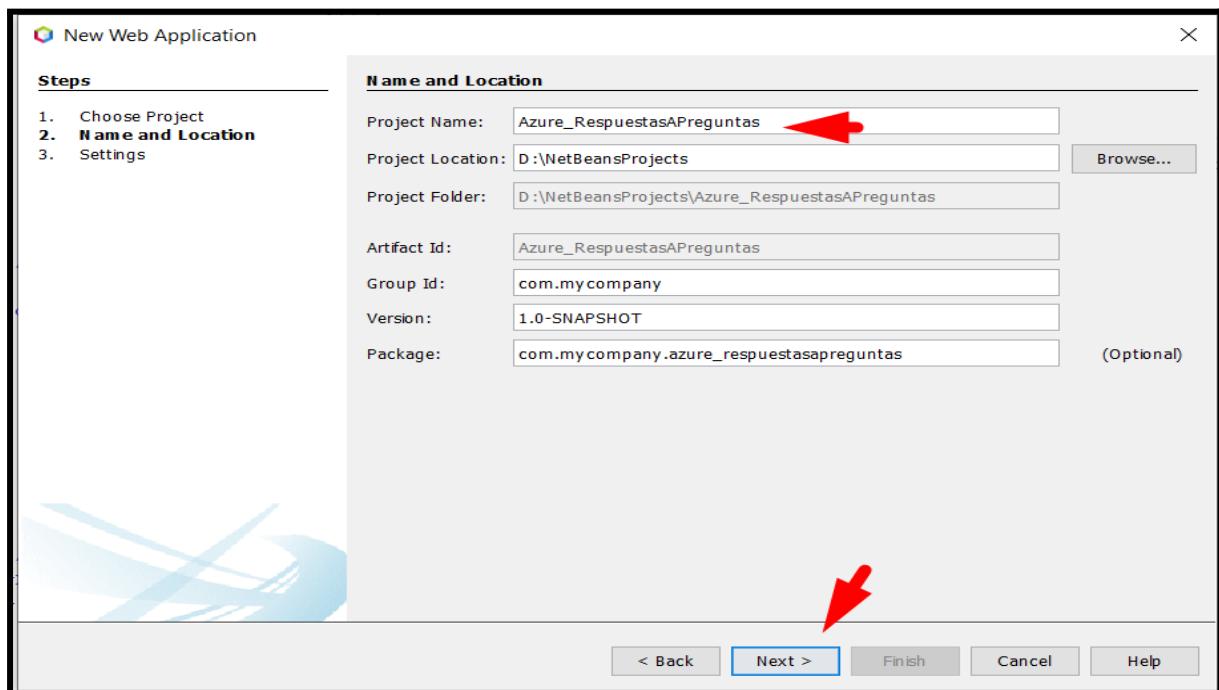
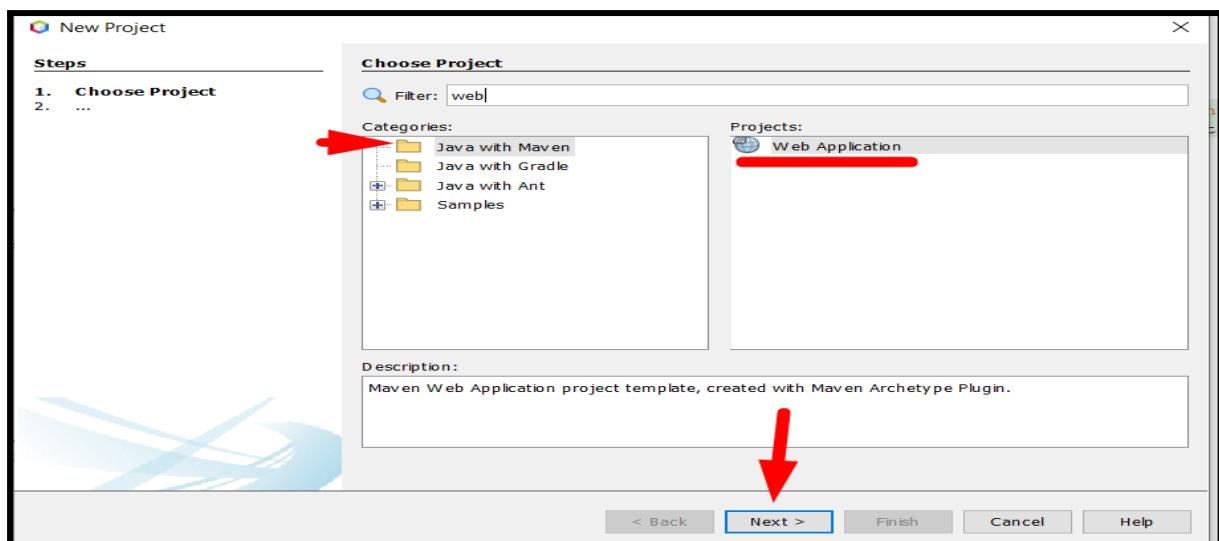
**Con eso comprobamos que nuestra Api está funcionando para luego consumirlo en Java.**

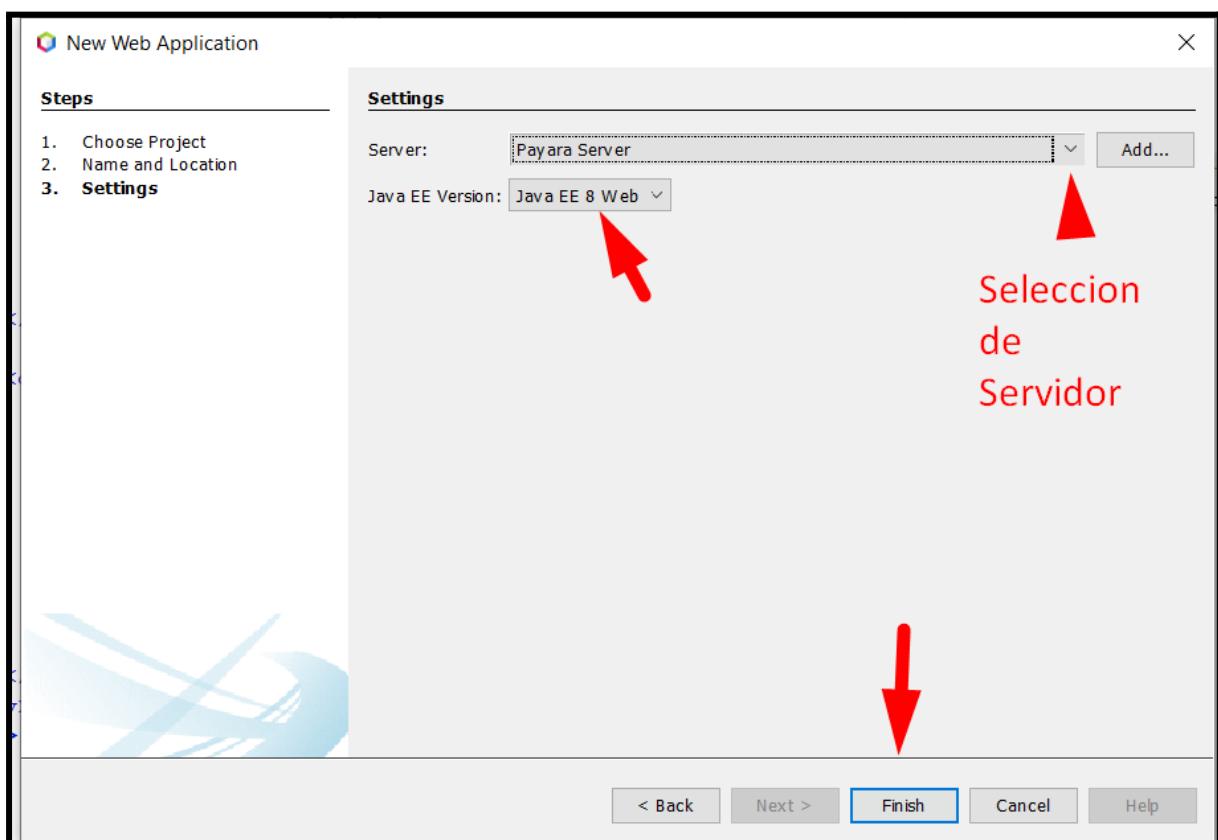
**¡¡Felicidades!!**  
**Ya estamos por terminar.**

# Consumir servicio De QnA en Java usando el Apache NetBeans IDE 12.3

## Creación de Proyecto

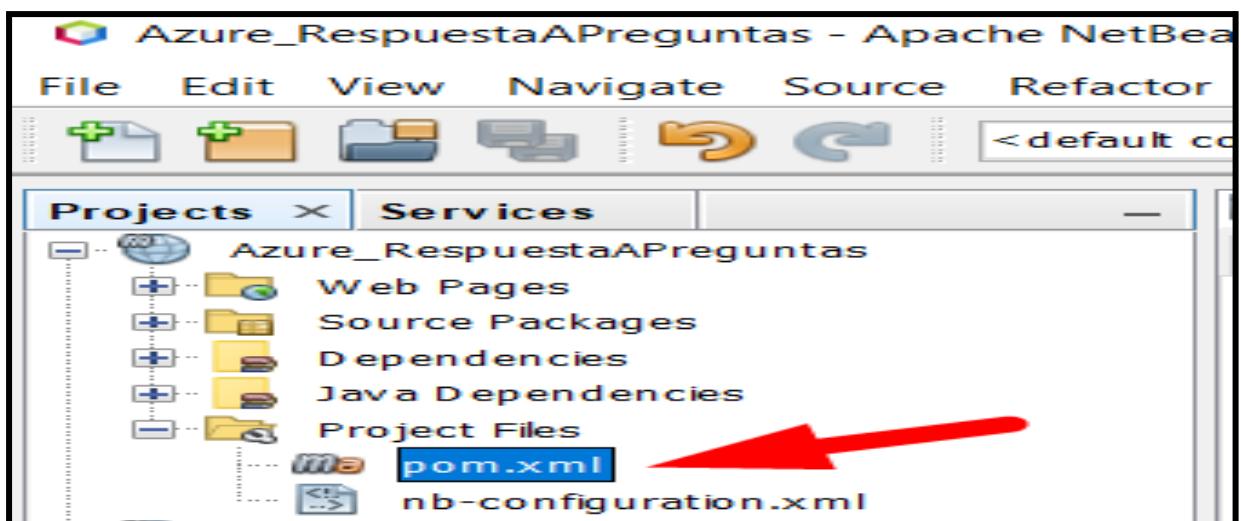
Para ello, empezaremos creando un proyecto Maven  
\_ Web Application y seleccionamos en next





Ahora nos dirigimos a colocar las dependencias en el archivo pom.

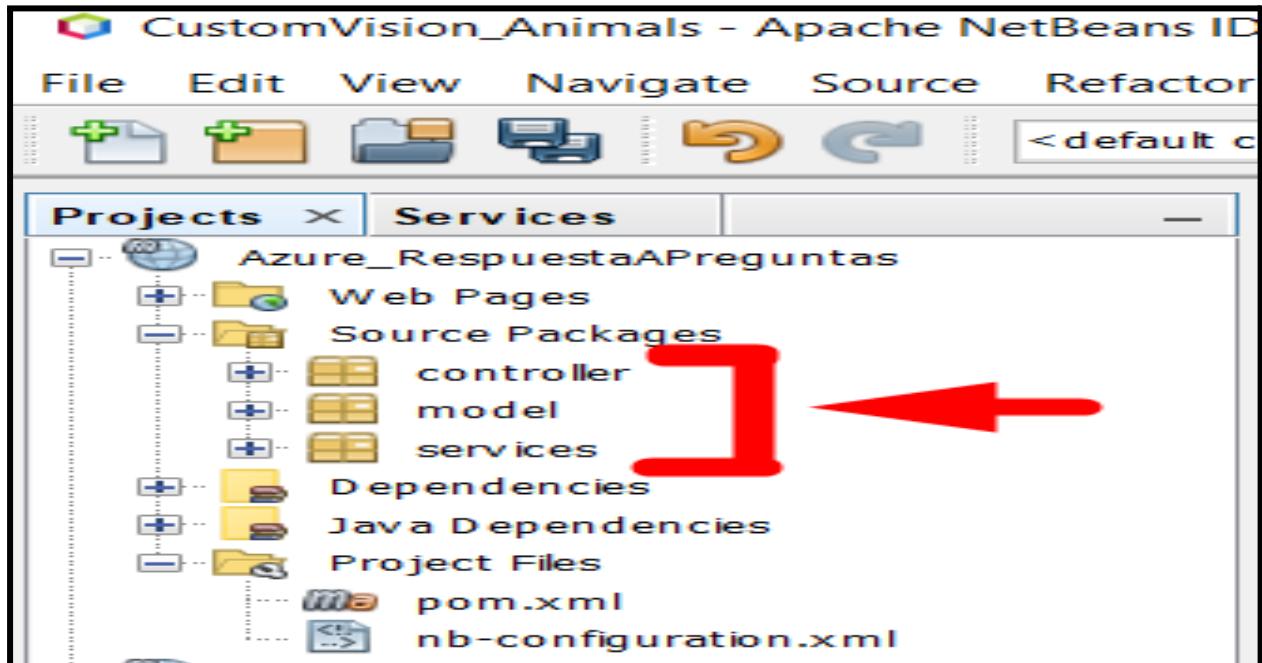
## Añadir dependencias



```
22 <dependency>
23     </dependency>
24     <dependency>
25         <groupId>org.primefaces</groupId>
26         <artifactId>primefaces</artifactId>
27         <version>11.0.0</version>
28     </dependency>
29     <dependency>
30         <groupId>org.projectlombok</groupId>
31         <artifactId>lombok</artifactId>
32         <version>1.18.20</version>
33         <scope>provided</scope>
34     </dependency>
35     <dependency>
36         <groupId>com.google.code.gson</groupId>
37         <artifactId>gson</artifactId>
38         <version>2.9.0</version>
39         <type>jar</type>
40     </dependency>
41     <dependency>
42         <groupId>org.apache.httpcomponents</groupId>
43         <artifactId>httpcore</artifactId>
44         <version>4.4.13</version>
45         <type>jar</type>
46     </dependency>
47     <dependency>
48         <groupId>com.squareup.okhttp3</groupId>
49         <artifactId>okhttp</artifactId>
50         <version>4.9.3</version>
51     </dependency>
52     <dependency>
53         <groupId>org.apache.httpcomponents</groupId>
54         <artifactId>httpclient</artifactId>
55         <version>4.5.13</version>
56         <type>jar</type>
57     </dependency>
58     <dependency>
59         <groupId>org.json</groupId>
60         <artifactId>json</artifactId>
61         <version>20160810</version>
62     </dependency>
63     <!-- https://mvnrepository.com/artifact/org.json/json -->
64     <dependency>
65         <groupId>org.json</groupId>
66         <artifactId>json</artifactId>
67         <version>20160810</version>
68     </dependency>
69     </dependencies>
```

## Creación de Paquetes

Luego creamos nuestros paquetes que serían model, service, controller

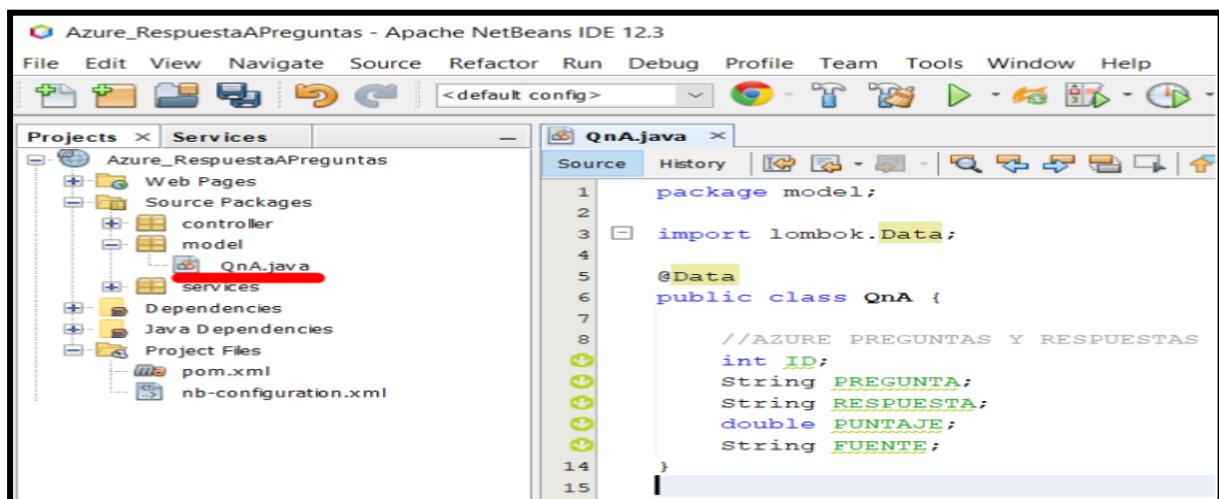


## Creación de clases dentro de los paquetes

### Para el modelo

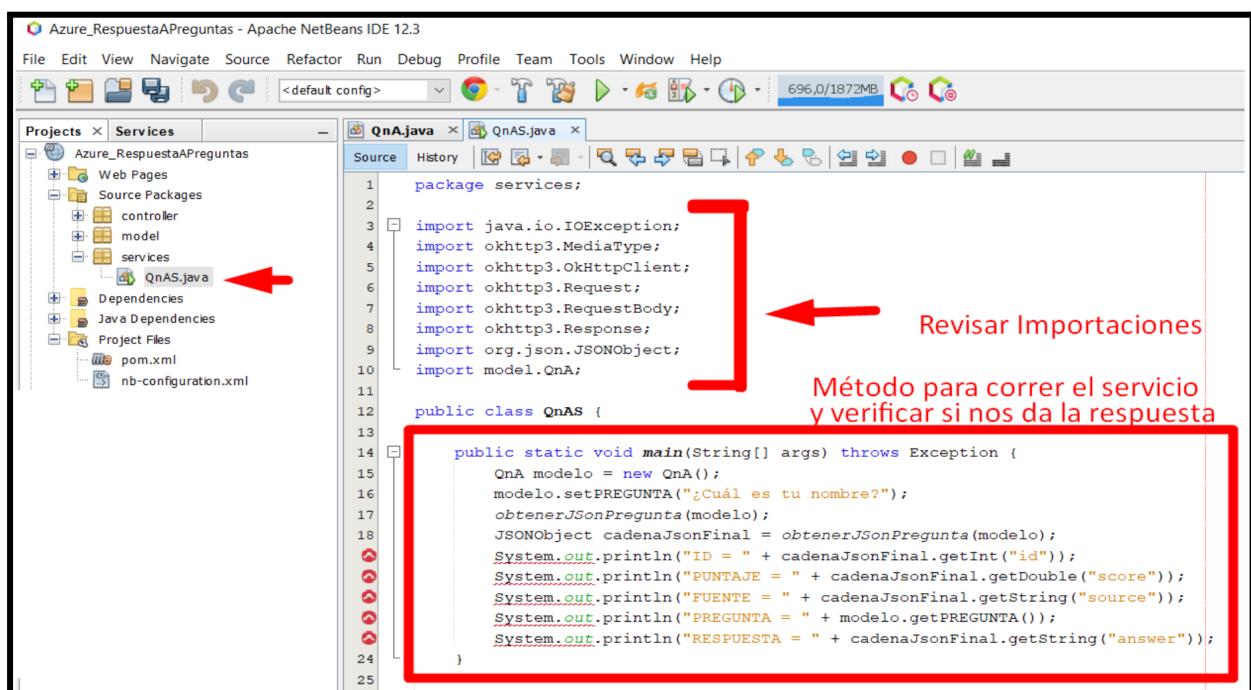
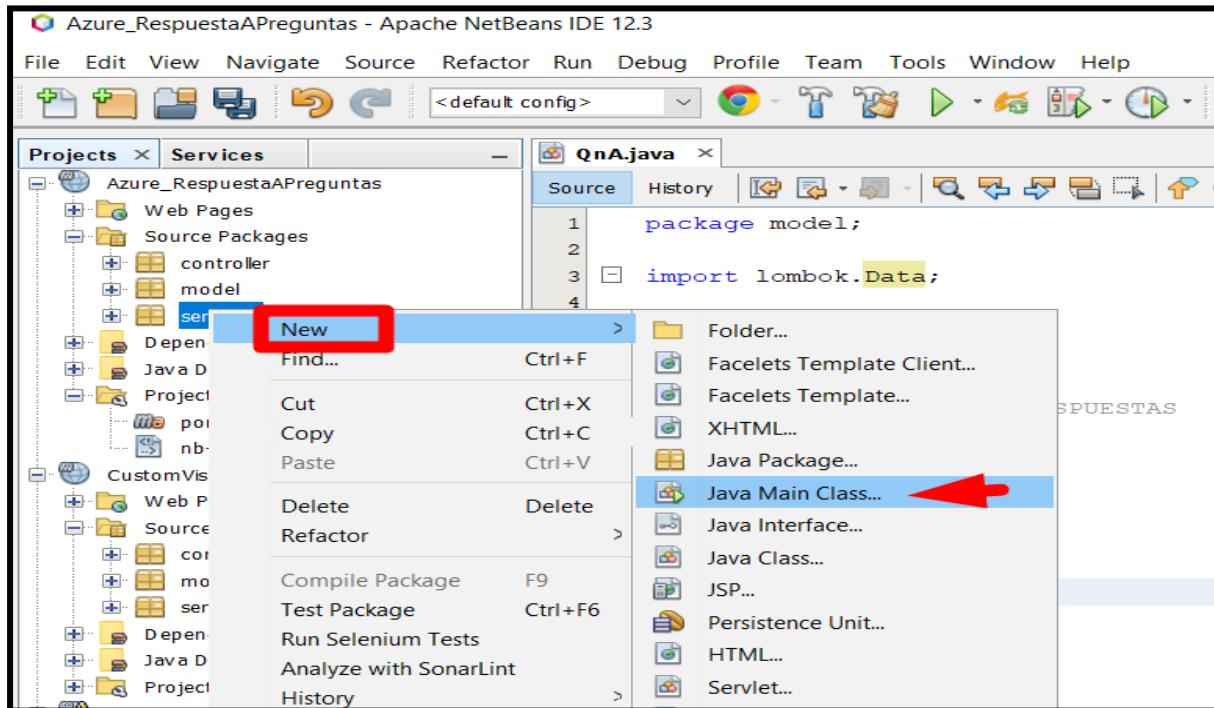
– Creamos una clase con nombre QnA, importamos lombok para así no tener la necesidad de generar los getter y setters, y escribimos las siguientes variables, ID, PREGUNTA, RESPUESTA, PUNTAJE, FUENTE.

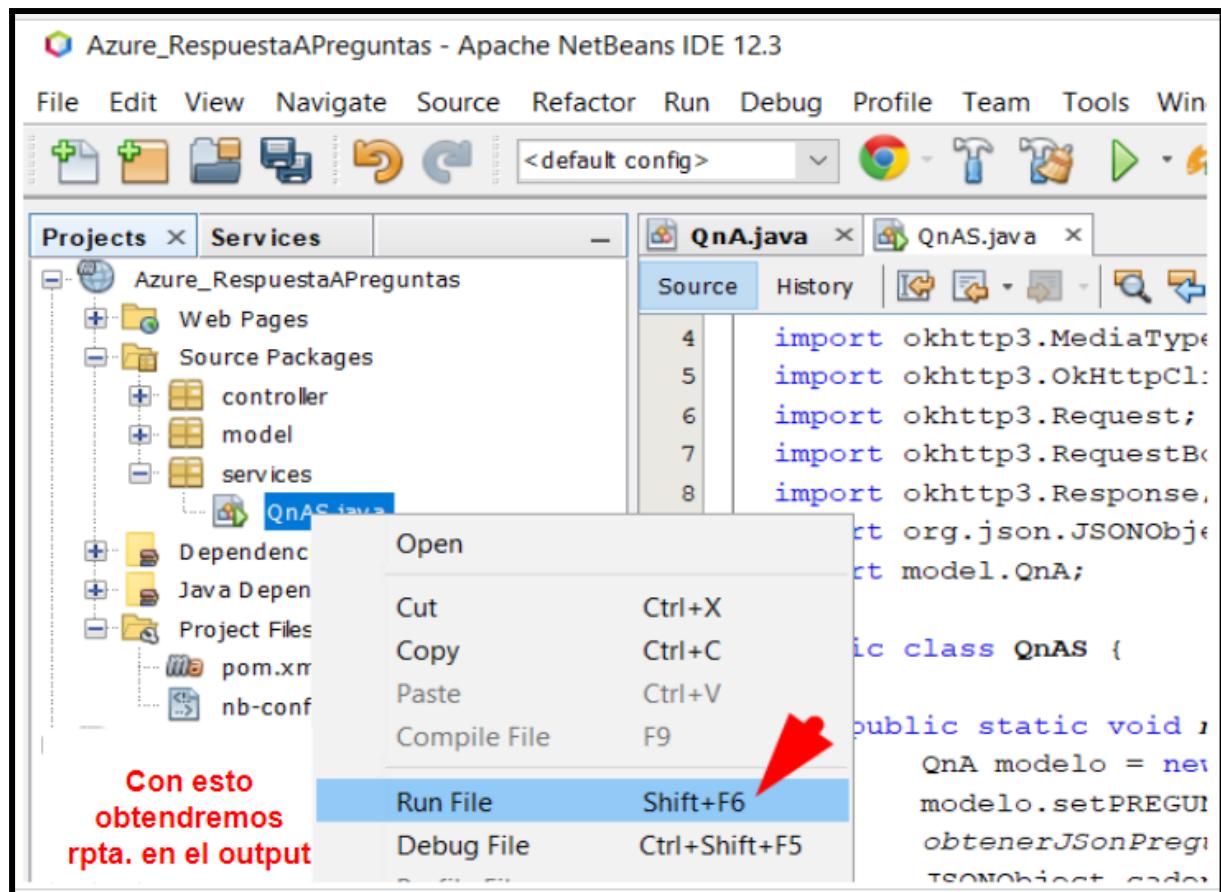
– Guardamos y cerramos, ahora nos iremos al paquete services



# Para el Services

Click derecho, new, creamos un archivo Java Main Class con nombre QnAS





```

public static JSONObject obtenerJSONObject(QnA modelo) throws IOException {
    OkHttpClient client = new OkHttpClient().newBuilder().build();
    MediaType mediaType = MediaType.parse("application/json");
    RequestBody body = RequestBody.create(mediaType, "\r\n" + "question:" + modelo.getPREGUNTA() + "\r\n");
    Request request = new Request.Builder()
        .url("https://preguntasdemol.azurewebsites.net/qnamaker/knowledgebases/aba04c1e-f09a-4b99-bfd9-5cff1a28b76/generateAnswer")
        .method("POST", body)
        .addHeader("Authorization", "EndpointKey 63a6cbe0-c586-4b7a-8991-5fbe704b64ec")
        .addHeader("Content-Type", "application/json")
        .addHeader("Cookie", "ARRAffinity=135f536e0d078a815c3b83fda3468ab8a3a2abc604a35eefb28648d3961cfa9b; "
            + "ARRAffinitySameSite=135f536e0d078a815c3b83fda3468ab8a3a2abc604a35eefb28648d3961cfa9b")
        .build();
    Response response = client.newCall(request).execute();
    if (!response.isSuccessful()) {
        throw new IOException("Unexpected code " + response);
    }
    JSONObject cadenaJson = new JSONObject(response.body().string());
    JSONObject cadenaJsonFinal = cadenaJson.getJSONArray("answers").getJSONObject(0);
    return cadenaJsonFinal;
}

```

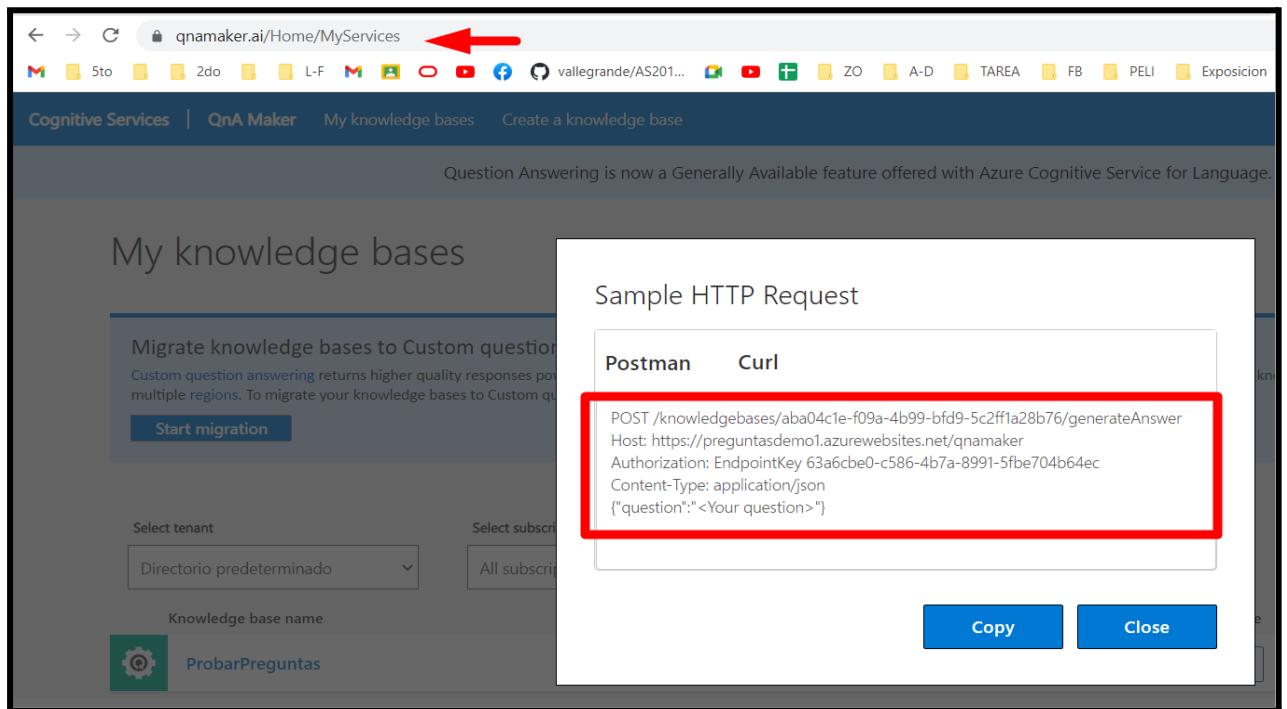
Para saber que colocar en la línea 31, 32, 33, nos dirigimos al portal de **QnA Marker**,

donde nos dió el código de la solicitud HTTP.

\_ En la línea 31, escribimos el Host y seguido el POST con eso estamos consumiendo el servicio de Azure de QnA Marker.

\_ En la 33 y 34, colocamos los siguientes datos de authorization y el content type

\_ Si no recuerdan aquí les dejo el link del portal, y entran en mi base de conocimiento y al lado derecho en code y se les mostrará de nuevo el código <https://www.qnamaker.ai/Home/MyServices>



En la línea 35 colocamos los cookies.

¿De donde sacas eso?

Pues cuando ejecutamos en el Postman, nos da la respuesta y se nos crea unos cookies

The screenshot shows the Postman interface with a successful POST request to `https://preguntasdemo1.azurewebsites.net/qnamaker/knowledgebases/aba04c1e-f09a-4b99-bfd9-5c2ff1a28b76/generateAnswer`. The response body is displayed in JSON format:

```

1
2   "answers": [
3     {
4       "questions": [
5         "¿Cuál es tu nombre?"
6       ],
7       "answer": "Mi nombre es Jhianbot",
8       "score": 100.0,
9       "id": 8,
10      "source": "PreguntasJhianpol1.pdf",
11      "isDocumentText": false,
12      "metadata": []
13    }
14  ]

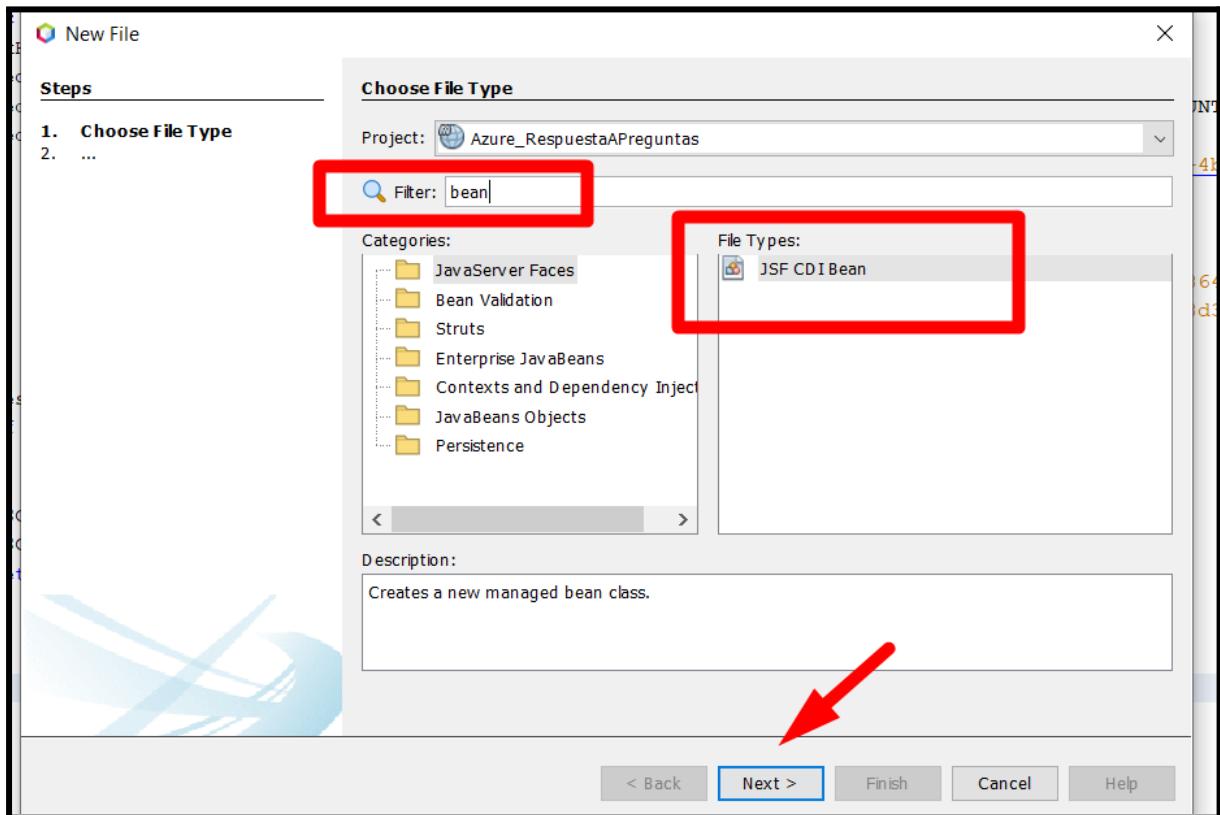
```

Una vez que ya entendemos dónde está, colocamos los cookie en la línea 35

The screenshot shows the Postman interface with the 'Cookies' tab selected. Two cookies are listed:

Name	Value	Domain	Path	Expires
ARRAffinity	135f536e0d0...	preguntasder...	/	Session
ARRAffinitySe...	135f536e0d0...	preguntasder...	/	Session

# Creación del controller



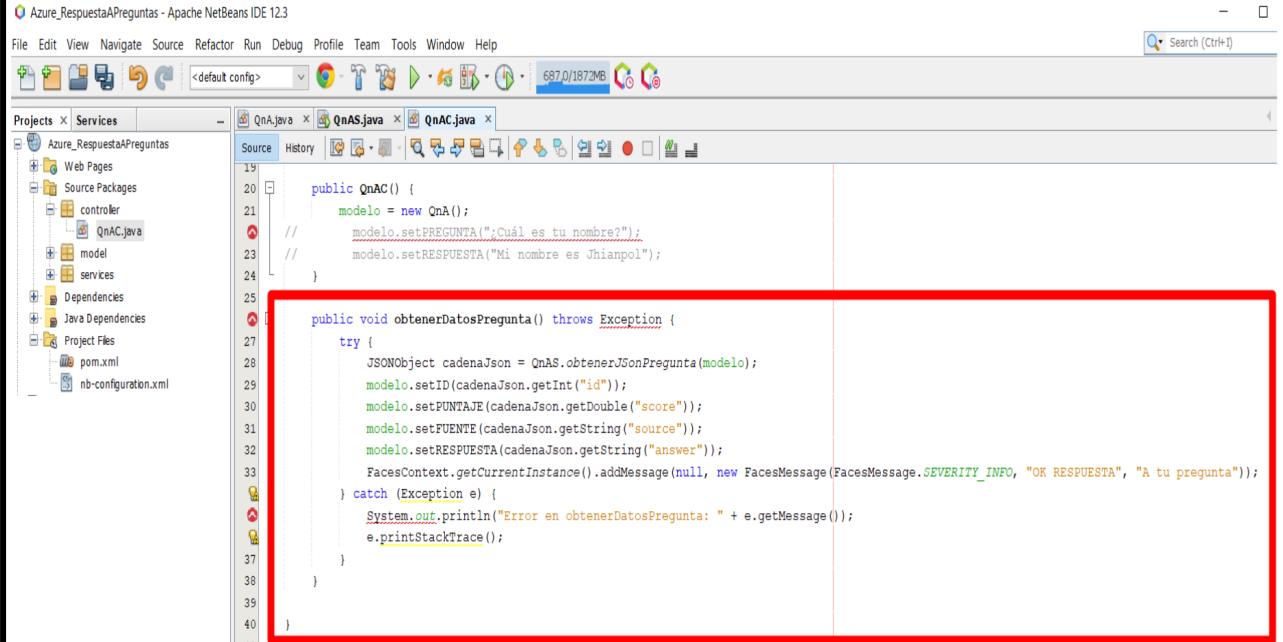
Revisamos la importaciones

Agregamos las anotaciones

```
1 package controller;
2
3 import javax.inject.Named;
4 import javax.enterprise.context.SessionScoped;
5 import java.io.Serializable;
6 import javax.faces.application.FacesMessage;
7 import javax.faces.context.FacesContext;
8 import model.QnA;
9 import org.json.JSONObject;
10 import services.QnAS;
11 import lombok.Data;
12
13 @Data
14 @Named(value = "qnaC")
15 @SessionScoped
16 public class QnAC implements Serializable {
17
18     QnA modelo;
19
20     public QnAC() {
21         // ...
22         modelo = new QnA();
23         // ...
24         modelo.setPREGUNTA("¿Cuál es tu nombre?");
25         modelo.setRESPUESTA("Mi nombre es Jhianpol");
26     }
27 }
```

The screenshot shows the Apache NetBeans IDE interface with the 'Azure\_RespuestaAPreguntas' project open. The 'Projects' panel on the left shows the project structure with a red arrow pointing to the 'QnAC.java' file in the 'controller' package. The 'Source' tab of the code editor is active, displaying the Java code for the 'QnAC' class. A red box highlights the import statements at the top of the file. The code defines a controller class 'QnAC' that implements 'Serializable'. It uses annotations like @Data, @Named, and @SessionScoped. The class has a private attribute 'modelo' of type 'QnA' and a constructor that initializes it with a JSON object and sets some properties.

Creamos el siguiente método con nombre **ObtenerDatosPregunta**.

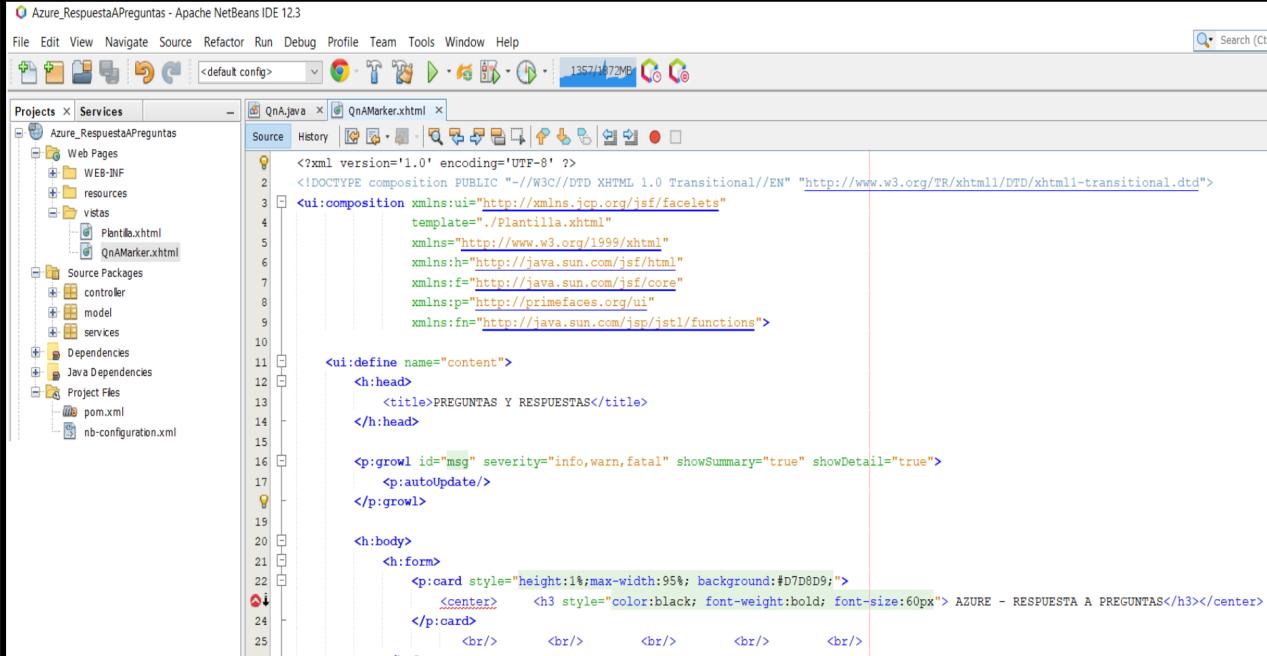


The screenshot shows the Apache NetBeans IDE interface with the project "Azure\_RespuestaAPreguntas" open. The code editor displays the file `QnAC.java`. A red box highlights the following method:

```
public void obtenerDatosPregunta() throws Exception {
    try {
        JSONObject cadenaJson = QnAS.obtenerJsonPregunta(modelo);
        modelo.setID(cadenaJson.getInt("id"));
        modelo.setPUNTAJE(cadenaJson.getDouble("score"));
        modelo.setFUENTE(cadenaJson.getString("source"));
        modelo.setRESPUESTA(cadenaJson.getString("answer"));
        FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_INFO, "OK RESPUESTA", "A tu pregunta"));
    } catch (Exception e) {
        System.out.println("Error en obtenerDatosPregunta: " + e.getMessage());
        e.printStackTrace();
    }
}
```

## Creación de la vista

Para ello crearemos un archivo **xhtml**



The screenshot shows the Apache NetBeans IDE interface with the project "Azure\_RespuestaAPreguntas" open. The code editor displays the file `QnAMarker.xhtml`. A red box highlights the following code:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    template="/Plantilla.xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:p="http://primefaces.org/ui"
    xmlns:fn="http://java.sun.com/jsp/jstl/functions">

    <ui:define name="content">
        <h:head>
            <title>PREGUNTAS Y RESPUESTAS</title>
        </h:head>

        <p:growl id="msg" severity="info, warn, fatal" showSummary="true" showDetail="true">
            <p:autoUpdate/>
        </p:growl>

        <h:body>
            <h:form>
                <p:card style="height:1%;max-width:95%; background:#D7D8D9;">
                    <center>      <h3 style="color:black; font-weight:bold; font-size:60px"> AZURE - RESPUESTA A PREGUNTAS</h3></center>
                </p:card>
                <br/>          <br/>          <br/>          <br/>
            </h:form>
        </h:body>
    </ui:define>
</ui:composition>
```

```

27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

```

27     <p:panelGrid columns="3" >  
 28         <h:form>  
 29             <p:card style="width:25em; background:#FFCDD2;">  
 30                 <p:linkButton value="QnA MARKER" style="font-size:25px; font-weight:bold"/>  
 31                 <f:facet name="footer">  
 32                     <p:graphicImage name="/images/MAKERQnA.jpg" style="width:320px;height:230px;"/>  
 33                 </f:facet>  
 34             </p:card>  
 35         </h:form>  
 36         <br/>  
 37     <p>  
 38         <h:form id="form">  
 39             <center>  
 40                 <div class="container">  
 41                     <br/>  
 42                 <p:outputLabel style="color:black; font-weight:bold; font-size:40px" value="HAS TU PREGUNTA"/>  
 43                     <br/><br/>  
 44                 <p:inputText value="#{qnaC.modelo.PREGUNTA}" required="true" requiredMessage="Ingresa tu Pregunta"  
 45                     style="width: 500px">  
 46                 <p:ajax listener="#{qnaC.obtenerDatosPregunta()}" update="form msg"/>  
 47             </p:inputText>  
 48             <br/><br/>  
 49             <br/><br/>  
 50             <p:panelGrid columns="2" >  
 51                 <p:outputLabel value="RESPUESTA " style="font-size:25px; font-weight:bold"/>  
 52                 <p:outputLabel id="RESPUESTA" value="#{qnaC.modelo.RESPUESTA}"/>  
 53                 <p:outputLabel value="ID " style="font-size:25px; font-weight:bold"/>  
 54                 <p:outputLabel id="ID" value="#{qnaC.modelo.ID}"/>  
 55                 <p:outputLabel value="PUNTAJE " style="font-size:25px; font-weight:bold"/>  
 56                 <p:outputLabel id="PUNTAJE" value="#{qnaC.modelo.PUNTAJE}"/>  
 57                 <p:outputLabel value="FUENTE " style="font-size:25px; font-weight:bold"/>  
 58                 <p:outputLabel id="FUENTE" value="#{qnaC.modelo.FUENTE}"/>  
 59     </p:panelGrid>

```

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67

```

50     <p:panelGrid columns="2" >  
 51         <p:outputLabel value="RESPUESTA " style="font-size:25px; font-weight:bold"/>  
 52         <p:outputLabel id="RESPUESTA" value="#{qnaC.modelo.RESPUESTA}"/>  
 53         <p:outputLabel value="ID " style="font-size:25px; font-weight:bold"/>  
 54         <p:outputLabel id="ID" value="#{qnaC.modelo.ID}"/>  
 55         <p:outputLabel value="PUNTAJE " style="font-size:25px; font-weight:bold"/>  
 56         <p:outputLabel id="PUNTAJE" value="#{qnaC.modelo.PUNTAJE}"/>  
 57         <p:outputLabel value="FUENTE " style="font-size:25px; font-weight:bold"/>  
 58         <p:outputLabel id="FUENTE" value="#{qnaC.modelo.FUENTE}"/>  
 59     </p:panelGrid>  
 60     </div>  
 61     </center>  
 62     </h:form>  
 63     </p:panelGrid>  
 64     </h:body>  
 65     </ui:define>  
 66   </ui:composition>

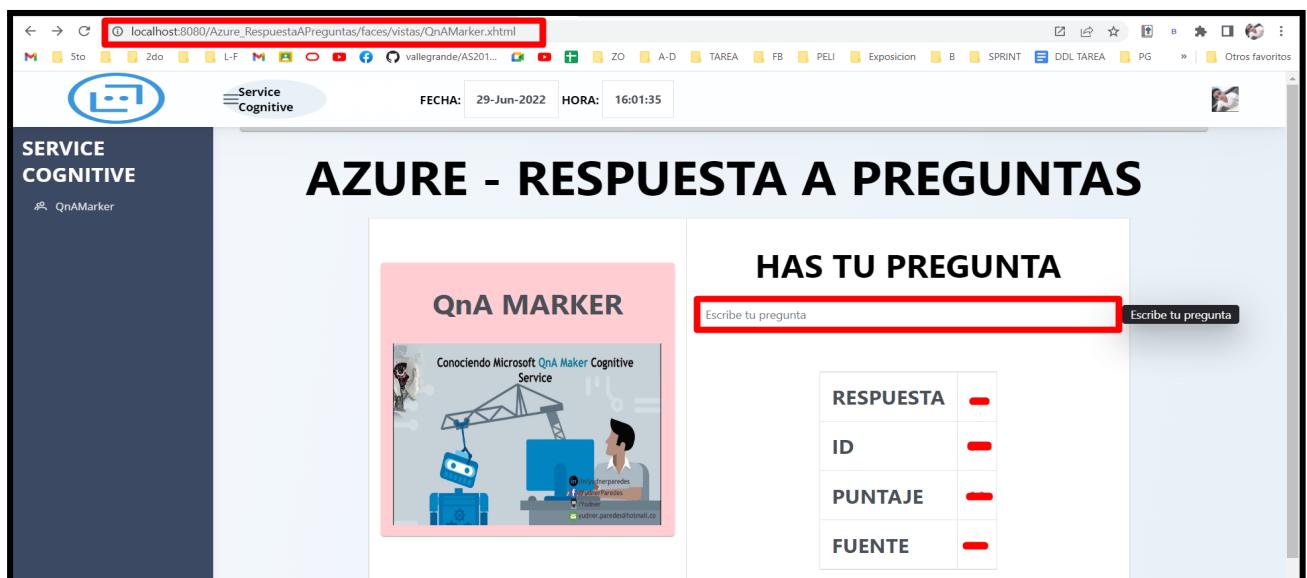
# Configuración del web.xml para correr proyecto

The screenshot shows the Apache NetBeans IDE interface. The left pane displays the project structure for 'Azure\_RespuestaAPreguntas'. A red arrow points to the 'vistas' folder, which contains 'Plantilla.xhtml' and 'QnAMarker.xhtml'. The right pane shows the 'Source' tab of the 'web.xml' file. The code is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="4.0" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_4_0.xsd">
    <context-param>
        <param-name>javax.faces.PROJECT_STAGE</param-name>
        <param-value>Development</param-value>
    </context-param>
    <servlet>
        <servlet-name>Faces Servlet</servlet-name>
        <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Faces Servlet</servlet-name>
        <url-pattern>/faces/*</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>/faces/vistas/QnAMarker.xhtml</welcome-file>
    </welcome-file-list>
</web-app>
```

## Run al proyecto

Una vez que corremos el proyecto, escribimos la pregunta y le damos click en cualquier parte y esperamos a que el ajax busque entre los datos que hemos puesto y encuentre una respuesta para plasmarla entre las respuestas



RESPUESTA	Mi creador es Jhianpol Ramos
ID	15
PUNTAJE	71.97
FUENTE	PreguntasJhianpol1.pdf

Y así concluimos con nuestro Servicio Cognitivo de Azure.

## Conclusiones

- ❖ QnA Marker es rápido y fácil de implementar modelos de Machine Learning.
- ❖ QnA nos da la oportunidad de usar datos que tengamos localmente o en la web, de acuerdo a preguntas y respuestas para luego ser evaluada y usada consumiendo la Api.
- ❖ Debemos tener un pdf con preguntas o url, link donde se encuentre una página para extraer o usar las preguntas y responder de acuerdo a sus respuestas en el documento, más su extensión de documento debe ser pdf o docx .
- ❖ Si queremos agregar más datos o pdf con más preguntas y respuestas para el bot, debemos ir al portal de QnA Marker, en ajustes y poder subir los datos.
- ❖ Para implementar la Api, no es necesario tener una cantidad exacta de documentos de preguntas y respuestas, bastaría con uno, o de lo contrario, plasmamos las preguntas y respuestas en el portal de QnA Marker ahí mismo.
- ❖ Cuando subamos nuestros documentos localmente, no debemos olvidar que debemos guardar, entrenar y publicar el servicio para que se nos actualice los datos subidos.

## Consideraciones

- ❖ El Api de QnA Maker soporta archivos con extensión pdf y docx, subidos desde su portal.
- ❖ En Primefaces sólo se implemento por url o solicitud http de código que se generó desde el portal de QnA Marker
- ❖ El modelo ha sido entrenado con dos documentos de extensión pdf.
- ❖ El servicio devuelve las respuestas a preguntas que se le hace en el frontend, busca entre los documentos la pregunta y responde con alguna respuesta que encuentre similar a su pregunta.
- ❖ El modelo tardó 2 minutos en entrenarse.

## Anexo

- ❖ [Repositorio de GitHub](#)

## Referencias

- ❖ Tutorial QnA Maker TI Capacitación. (2018). Hello Bots: [YouTube Video]. Retrieved from <https://www.youtube.com/watch?v=BiX8LX9oCKA>
- ❖ Portal Microsoft Azure. (2022). Retrieved June 29, 2022, from Azure.com website: <https://portal.azure.com/#home>
- ❖ Portal QnA Maker. (2022). Retrieved June 29, 2022, from Qnamaker.ai website: <https://www.qnamaker.ai/>
- ❖ Servicio de QnA Maker. (2022). Retrieved June 29, 2022, from Qnamaker.ai website: <https://www.qnamaker.ai/Home/MyServices>
- ❖ Creación de QnA Maker en Microsoft Azure. (2022). Retrieved June 29, 2022, from Azure.com website: [https://portal.azure.com/#view/Microsoft\\_Azure\\_ProjectOxford/CognitiveServicesHub/~/QnAMaker](https://portal.azure.com/#view/Microsoft_Azure_ProjectOxford/CognitiveServicesHub/~/QnAMaker)

