

Identificación de platillos de comida en base a ingredientes empleando Spark

Angel Vidal Larreategui Castro¹

¹Universidad Nacional de Ingeniería. Facultad de Ciencias.
Ciencias de la Computación

Agosto, 2023



**FACULTAD DE
CIENCIAS**

Tabla de contenidos

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados
- 7 Conclusiones
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados
- 7 Conclusiones
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

Introducción

En la gastronomía peruana se puede encontrar una diversidad de platillos entre todas las regiones que conforman a nuestro país.



Figura: Platillos peruanos.



**FACULTAD DE
CIENCIAS**

Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados
- 7 Conclusiones
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

Problema

La elección sobre qué platillo preparar en base a lo que se tenga en la cocina es un problema recurrente en muchos peruanos.



Figura: ¿Qué puedo cocinar hoy?



**FACULTAD DE
CIENCIAS**

Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos**
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados
- 7 Conclusiones
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

■ **Objetivo general**

- Realizar la inferencia de platillos de comida en base a ingredientes para poder contextualizarlo y aplicarlo en el campo de la gastronomía peruana.

■ **Objetivos específicos**

- Revisar técnicas de Big Data.
- Revisar el funcionamiento y procesamiento de datos de Spark.
- Mejorar el diseño de una ontología relacionada a la gastronomía peruana.



Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados
- 7 Conclusiones
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

Comidas de la región costera del Perú

Perú posee una gran diversidad gastronómica.



Figura: Platillos de la costa peruana



**FACULTAD DE
CIENCIAS**

Web scraping

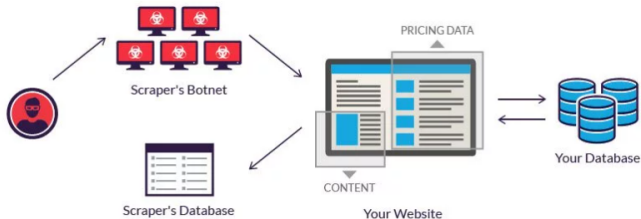


Figura: Proceso de extracción de la información de un sitio web.



**FACULTAD DE
CIENCIAS**



Figura: Apify es una forma más inteligente de extraer información de sitios web.



- Se puede definir una ontología como como una definición formal de propiedades, relaciones entre entidades que van a existir para un determinado dominio en particular.
- Comprende un esquema semántico de clases, propiedades y diversos tipos de relaciones que se encuentren, además de una taxonomía de conceptos, clases y también de ciertos individuos específicos.



Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo**
- 6 Resultados
- 7 Conclusiones
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

El proceso de llegó a dividir en 3 partes.

1. Extracción de la información.
2. Procesamiento de la data en Spark.
3. Elección de conocimientos y construcción de la ontología.



Extracción de la información

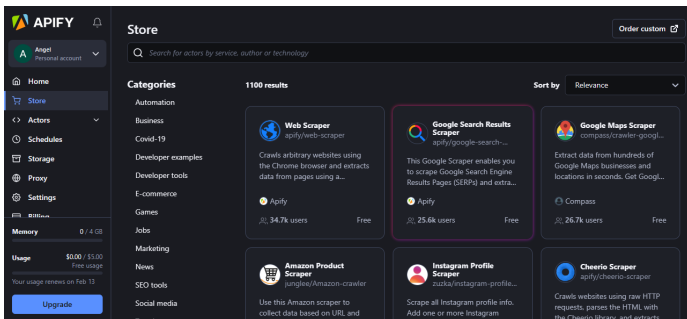


Figura: Interfaz de Apify, donde se muestran los diversos bots que aloja dicha plataforma.



FACULTAD DE
CIENCIAS

Extracción de la información

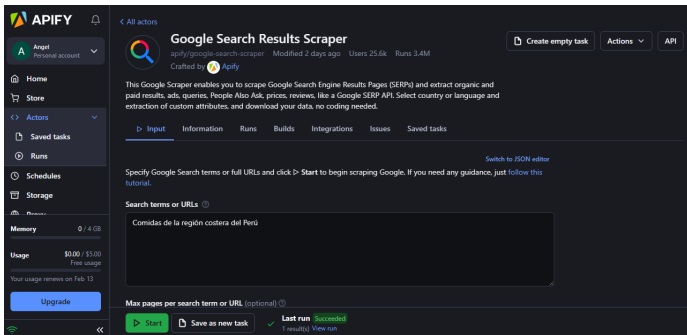


Figura: Interfaz del bot Google Search Results Scraper.



**FACULTAD DE
CIENCIAS**

Extracción de la información

The screenshot displays the APIFY web interface. On the left is a sidebar with navigation options: Home, Store, Actors, Saved tasks, Runs, Schedules, Storage, Proxy, and Settings. The main area shows the 'Google Search Results Scraper - Run' status as 'Succeeded'. A summary bar indicates 1 result, 1 of 1 handled, \$0.004 usage, and a duration of 9s. Below this, the 'Output' tab is active, showing a table of search results. The table has columns for #, Title, URL, Description, Search term, Domain, Page, and Total results. Three results are visible, all from google.com.pe. The first result is '50 PLATOS TÍPICOS DE LA COSTA PERUANA'. The second is '35 platos típicos de la costa peruana más representativos ...'. The third is 'Los 25 Mejores Platos Típicos de ...'. At the bottom of the results table are buttons for 'Export 1 result', 'Go to actor', 'Integrate', and 'Resurrect'.

#	Title	URL	Description	Search term	Domain	Page	Total results
				searchQuery.term	searchQuery.domain	searchQuery.page	results/total
1	50 PLATOS TÍPICOS DE LA COSTA PERUANA	https://comidasperuanas.net/...	Comidas de la región costera del Perú	Comidas de la región costera del Perú	google.com.pe	1	95/2000
2	35 platos típicos de la costa peruana más representativos ...	https://comococinar.pe/platos...	7 oct 2022Comidas de la Costa peruana y los historiadores · Las 5 recetas de la costa peruana más emblemáticas · Ceviche · Tabla informativa: Ceviche ...	Comidas de la región costera del Perú	google.com.pe	1	95/2000
3	Los 25 Mejores Platos Típicos de ...	https://www.todoenperu.net/o...	Comidas de la región	Comidas de la región	google.com.pe	1	95/2000

Figura: Resultados que muestra el bot al haberle solicitado una búsqueda.



**FACULTAD DE
CIENCIAS**

ACTUALIZADO EL OCTUBRE 7, 2022 — CARNE

55 platos típicos de la costa peruana más representativos según la historia (y sus recetas)



Figura: Página elegida para poder extraer la data de las comidas costeras peruanas.



**FACULTAD DE
CIENCIAS**

Extracción de la información

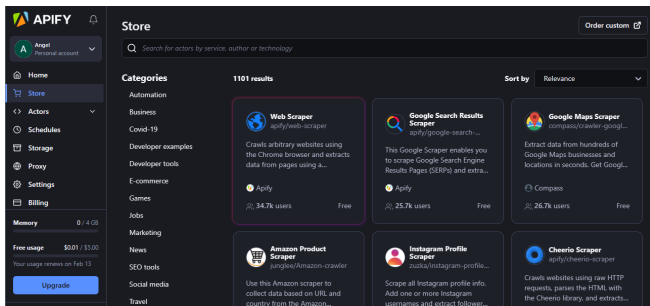


Figura: Bot Web Scraper para poder extraer la data del sitio web elegido.



**FACULTAD DE
CIENCIAS**

Extracción de la información

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	platillos	pescado	cebolla	aji	sal	pimienta	lechuga	culantro	carne_de_res	tomate	arroz	papa	vinagre	aceite	carne_de_vaca	huevo	aceituna	ajo	camote	caldo_de_pescado	aji_amarillo	mostaza	levadura_s
2	ceviche_de_pescado	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	lomo_saltado	0	1	0	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
4	papa_rellena	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	1	1	1	0	0	0	0
5	escabeche_de_pescado	1	1	0	1	0	1	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
6	picarones	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	arroz_con_mariscos	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
8	carapulcra	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0
9	tallarín_saltado_criollo	0	1	0	1	1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	0
10	pavo_al_horno	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
11	aji_de_gallina	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0
12	arroz_con_leche	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
13	tallarines_rojos_peruanos_con_pollo_o_carne	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
14	chilcano_de_pescado	1	1	1	1	1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
15	mazamorra_morada	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	arroz_a_la_jardinera	0	0	0	1	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0
17	empanada_de_carne	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
18	tamales_de_pollo_o_chancho	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
19	turrón_de_dofia_pepa	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	albondigas_de_carne	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
21	arroz_chaufa	0	0	0	1	1	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0
22	chicharrón_de_chancho	0	1	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0
23	leche_de_tigre	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
24	estofado_de_pollo	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0
25	pollo_a_la_brasa	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
26	salchipapa	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1
27	leche_asada	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Figura: Visualización de la data obtenida en Excel.



Procesamiento de la data en Spark

```
val ingredientCount = readLine("Ingrese la cantidad de ingredientes: ").toInt

val ingredientNames = (1 to ingredientCount).map { _ => readLine("Ingrese el nombre de un ingrediente: ") }

val columns = ingredientNames.map(column => s""""$column""").mkString(" AND ")

val query = s"SELECT platillos FROM peruvian_dishes WHERE $columns"

val jdbcDF = spark.read.format("jdbc").
  option("url", "jdbc:postgresql:postgres").
  option("user", "postgres").
  option("password", "021299").
  option("dbtable", s"($query) AS t").
  load()

jdbcDF.select("platillos").show()
```

Figura: Procesamiento de la data obtenida en Spark.



**FACULTAD DE
CIENCIAS**

Construcción de la Ontología

En primer lugar, se hará la elección del conocimiento **Platillo** y realizará su construcción dentro de la presente ontología mostrada en lenguaje OWL.

```
<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#Platillos">  
  <rdfs:comment xml:lang="es">Clase que representa un platillo.</rdfs:comment>  
  <rdfs:label xml:lang="es">Platillo</rdfs:label>  
</owl:Class>
```

Figura: Conocimiento **Platillo** dentro de la ontología en lenguaje OWL.



**FACULTAD DE
CIENCIAS**

Construcción de la Ontología

```
<!-- http://www.semanticweb.org/usuario/smartcooking#Segundo -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#Segundo">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Platillos"/>
  <rdfs:comment xml:lang="es">Clase que representa un platillo de segundo.</rdfs:comment>
  <rdfs:label xml:lang="es">Segundo</rdfs:label>
</owl:Class>

<!-- http://www.semanticweb.org/usuario/smartcooking#SegundoMaritimo -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#SegundoMaritimo">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Segundo"/>
  <rdfs:comment xml:lang="es">Clase que representa un segundo marítimo.</rdfs:comment>
  <rdfs:label xml:lang="es">SegundoMaritimo</rdfs:label>
</owl:Class>

<!-- http://www.semanticweb.org/usuario/smartcooking#SegundoNoMaritimo -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#SegundoNoMaritimo">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Segundo"/>
  <rdfs:comment xml:lang="es">Clase que representa un segundo no marítimo.</rdfs:comment>
  <rdfs:label xml:lang="es">SegundoNoMaritimo</rdfs:label>
</owl:Class>
```

Figura: Clase **Segundo** perteneciente a **Platillo**.



**FACULTAD DE
CIENCIAS**

Construcción de la Ontología

```
<!-- http://www.semanticweb.org/usuario/smartcooking#Sopa -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#Sopa">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Platillos"/>
  <rdfs:comment xml:lang="es">Clase que representa un platillo de sopa.</rdfs:comment>
  <rdfs:label xml:lang="es">Sopa</rdfs:label>
</owl:Class>

<!-- http://www.semanticweb.org/usuario/smartcooking#SopaMaritima -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#SopaMaritima">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Sopa"/>
  <rdfs:comment xml:lang="es">Clase que representa una sopa marítima.</rdfs:comment>
  <rdfs:label xml:lang="es">SopaMaritima</rdfs:label>
</owl:Class>

<!-- http://www.semanticweb.org/usuario/smartcooking#SopaNoMaritima -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#SopaNoMaritima">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Sopa"/>
  <rdfs:comment xml:lang="es">Clase que representa una sopa no marítima.</rdfs:comment>
  <rdfs:label xml:lang="es">SopaNoMaritima</rdfs:label>
</owl:Class>
```

Figura: Clase **Sopa** perteneciente a **Platillo**.



FACULTAD DE
CIENCIAS

Construcción de la Ontología

De manera análoga, también se construyen las clases **Postres** y **Refrescos**. El conocimiento **Ingredientes** también es necesario dentro de la ontología pues a partir de ellos se realizan los platillos.

```
<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#Ingredientes">
  <rdfs:comment xml:lang="es">Clase que representa un ingrediente.</rdfs:comment>
  <rdfs:label xml:lang="es">Ingrediente</rdfs:label>
</owl:Class>
```

Figura: Conocimiento **Ingredientes** dentro de la ontología en lenguaje OWL.



**FACULTAD DE
CIENCIAS**

Construcción de la Ontología

```
<!-- http://www.semanticweb.org/usuario/smartcooking#Carnes -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#Carnes">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Ingredientes" />
  <rdfs:comment xml:lang="es">Clase que representa carnes.</rdfs:comment>
  <rdfs:label xml:lang="es">Carnes</rdfs:label>
</owl:Class>

<!-- http://www.semanticweb.org/usuario/smartcooking#Granos -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#Granos">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Ingredientes" />
  <rdfs:comment xml:lang="es">Clase que representa granos.</rdfs:comment>
  <rdfs:label xml:lang="es">Granos</rdfs:label>
</owl:Class>

<!-- http://www.semanticweb.org/usuario/smartcooking#Aderezos -->

<owl:Class rdf:about="http://www.semanticweb.org/usuario/smartcooking#Aderezos">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Ingredientes" />
  <rdfs:comment xml:lang="es">Clase que representa aderezos.</rdfs:comment>
  <rdfs:label xml:lang="es">Aderezos</rdfs:label>
</owl:Class>
```

Figura: Clases **Carnes**, **Granos**, **Aderezos** pertenecientes al conocimiento **Ingredientes**.



**FACULTAD DE
CIENCIAS**

Construcción de la Ontología

Es importante también definir los individuos pertenecientes a cada clase que se ha definido.

```
<!-- http://www.semanticweb.org/usuario/smartcooking#arroz -->  
  
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/usuario/smartcooking#arroz">  
  <rdf:type rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Integral"/>  
</owl:NamedIndividual>
```

Figura: Ejemplo de como se define un individuo dentro de la ontología, en este caso siendo el individuo **Agua**.



**FACULTAD DE
CIENCIAS**

Construcción de la Ontología

Es necesario también definir una propiedad que pueda relacionar los conocimientos **Platillo** e **Ingredientes**.

```
<!-- http://www.semanticweb.org/usuario/smartcooking#contieneIngrediente -->  
  
<owl:ObjectProperty rdf:about="http://www.semanticweb.org/usuario/smartcooking#contieneIngrediente">  
  <rdfs:domain rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Platillos"/>  
  <rdfs:range rdf:resource="http://www.semanticweb.org/usuario/smartcooking#Ingredientes"/>  
  <rdfs:comment xml:lang="es">Propiedad para relacionar un platillo con los ingredientes que contiene.</rdfs:comment>  
  <rdfs:label xml:lang="es">contieneIngrediente</rdfs:label>  
</owl:ObjectProperty>
```

Figura: Propiedad **contieneIngredientes** que relaciona los conocimientos **Platillos** e **Ingredientes**.



**FACULTAD DE
CIENCIAS**

Construcción de la Ontología

Para apreciar como funciona dicha propiedad, se define un individuo perteneciente a **Platillos**.

```
<!-- http://www.semanticweb.org/usuario/smartcooking#arroz_con_mariscos -->

<owl:NamedIndividual rdf:about="http://www.semanticweb.org/usuario/smartcooking#arroz_con_mariscos">
  <rdf:type rdf:resource="http://www.semanticweb.org/usuario/smartcooking#SegundoMaritimo"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#ajo"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#almeja"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#arroz"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#calamar"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#cebolla"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#choro"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#concha"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#culantro"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#langostino"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#limon"/>
  <smartcooking:contieneIngrediente rdf:resource="http://www.semanticweb.org/usuario/smartcooking#pulpo"/>
</owl:NamedIndividual>
```

Figura: Individuo **arroz_con_mariscos** empleando la propiedad **contieneIngredientes** para relacionarse con sus ingredientes.



**FACULTAD DE
CIENCIAS**

Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados**
- 7 Conclusiones
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

```
scala> 23/07/20 15:55:51 WARN ProcfsMetricsGetter: Exception when trying to compute
pagesize, as a result reporting of ProcessTree metrics is stopped
val ingredientCount = readLine("Ingrese la cantidad de ingredientes: ").toInt
warning: there was one deprecation warning (since 2.11.0); for details, enable `:set
ting -deprecation' or `:replay -deprecation'
Ingrese la cantidad de ingredientes: ingredientCount: Int = 4
```

Figura: Obtención de la cantidad de ingredientes deseados por el usuario.




```
scala> val ingredientNames = (1 to ingredientCount).map { _ => readLine("Ingrese el  
nombre de un ingrediente: ")}  
warning: there was one deprecation warning (since 2.11.0); for details, enable `:set  
ting -deprecation' or `:replay -deprecation'  
Ingrese el nombre de un ingrediente: Ingrese el nombre de un ingrediente: Ingrese el  
nombre de un ingrediente: Ingrese el nombre de un ingrediente: ingredientNames: sca  
la.collection.immutable.IndexedSeq[String] = Vector(cebolla, sal, pimienta, aji)
```

Figura: Obtención de los nombres de los ingredientes ingresados por el usuario.



```
scala> val columns = ingredientNames.map(column => s""""$column""").mkString(" AND  
")  
columns: String = "cebolla" AND "sal" AND "pimienta" AND "aji"
```

Figura: Generación de lista de columnas para la consulta SQL.

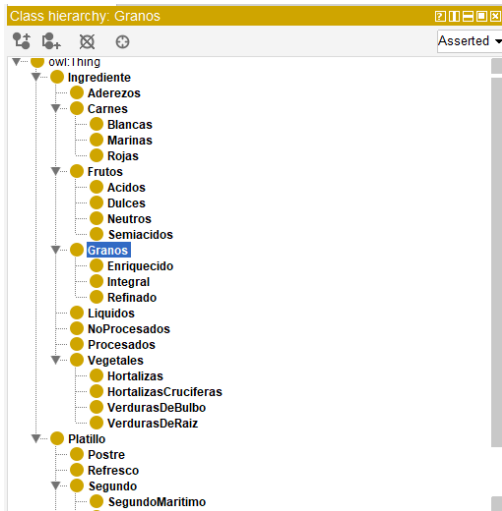


```
scala> val query = s"SELECT platillos FROM peruvian_dishes WHERE $columns"  
query: String = SELECT platillos FROM peruvian_dishes WHERE "cebolla" AND "sal" AND  
"pimienta" AND "aji"
```

Figura: Construcción de la consulta SQL.



Resultados



**FACULTAD DE
CIENCIAS**

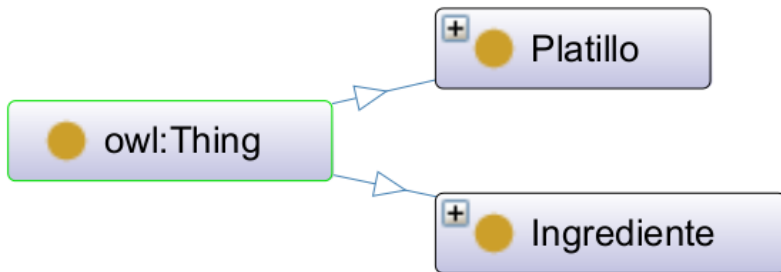


Figura: Clases Platillo e Ingrediente.





Figura: Propiedad *contieneIngrediente* visible entre las clases Platillo e Ingrediente.



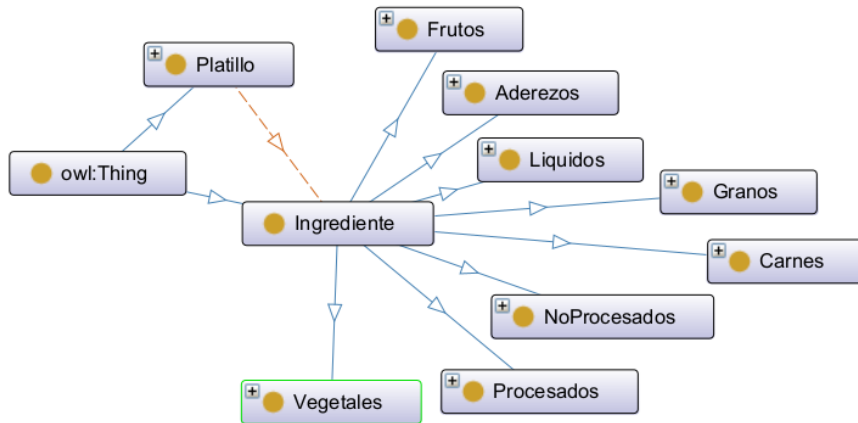


Figura: Clases y subclases de la ontología.



DL query: ⏏ ⏏ ⏏ ⏏

Query (class expression)

Platillo and contieneIngrediente some {sal} and contieneIngrediente some {cebolla} and contieneIngrediente some {pimienta} and contieneIngrediente some {aji}

Execute Add to ontology

Query results

Instances (2 of 2)

◆ ceviche_de_pescado	?
◆ chilcano_de_pescado	?

Query for

- ☐ Direct superclasses
- ☐ Superclasses
- ☐ Equivalent classes
- ☐ Direct subclasses
- ☐ Subclasses
- ☒ Instances

Figura: Inferencia de platillos a partir de ciertos ingredientes.



Resultados

```
C:\Windows\system32\cmd.exe - spark-shell --driver-class-path C:\spark\jars\postgresql-42.5.4.jar --jars C:\spark\jars\p...
warning: there was one deprecation warning (since 2.11.0); for details, enable `:setting -deprecati
on' or `:replay -deprecation'
Ingrese el nombre de un ingrediente: Ingrese el nombre de un ingrediente: Ingrese el nombre de un i
ngrediente: ingredientNames: scala.collection.immutable.IndexedSeq[String] = Vector(cebolla, papa,
pan)

scala> val columns = ingredientNames.map(column => s""""$column""").mkString(" AND ")
columns: String = "cebolla" AND "papa" AND "pan"

scala> val query = s"SELECT platillos FROM peruvian_dishes WHERE $columns"
query: String = SELECT platillos FROM peruvian_dishes WHERE "cebolla" AND "papa" AND "pan"

scala> val jdbcDF = spark.read.format("jdbc").
  |   option("url", "jdbc:postgresql:postgres").
  |   option("user", "postgres").
  |   option("password", "021299").
  |   option("dbtable", s"($query) AS t").
  |   load()
jdbcDF: org.apache.spark.sql.DataFrame = [platillos: string]

scala> jdbcDF.select("platillos").show()
+-----+
| platillos |
+-----+
|aji_de_gallina|
+-----+
```

Figura: Inferencia con 3 ingredientes.



**FACULTAD DE
CIENCIAS**

Resultados

DL query:

Query (class expression)

Platillo **and** contieneIngrediente **some** {pan} **and** contieneIngrediente **some** {papa} **and** contieneIngrediente **some** {cebolla}

Execute

Add to ontology

Query results

Instances (1 of 1)

 aji_de_gallina

?

Figura: Consulta dentro de la ontología realizada en Protege.



**FACULTAD DE
CIENCIAS**

Resultados

The screenshot displays a Semantic Web browser interface for the individual `aji_de_gallina`. The browser's address bar shows the URL `http://www.semanticweb.org/usuario/smartcooking#aji_de_gallina`. The interface is divided into several panels:

- Annotations:** A panel at the top with a search bar and a list of annotations for `aji_de_gallina`.
- Description:** A panel on the left showing the type `SegundoNoMaritimo` and options for "Same Individual As" and "Different Individuals".
- Property assertions:** A panel on the right showing a list of object property assertions for `aji_de_gallina`. The assertions are:

Property	Value
<code>contieneIngrediente</code>	<code>aceite_de_oliva</code>
<code>contieneIngrediente</code>	<code>cebolla</code>
<code>contieneIngrediente</code>	<code>pechuga_de_gallina</code>
<code>contieneIngrediente</code>	<code>aji_amarillo</code>
<code>contieneIngrediente</code>	<code>leche</code>
<code>contieneIngrediente</code>	<code>papa</code>
<code>contieneIngrediente</code>	<code>apio</code>
<code>contieneIngrediente</code>	<code>pan</code>
<code>contieneIngrediente</code>	<code>kion</code>
<code>contieneIngrediente</code>	<code>pimienta</code>
<code>contieneIngrediente</code>	<code>ajo</code>

At the bottom of the interface, there are controls for the Reasoner (active) and Show Inferences (checked).

Figura: Individuo `aji_de_gallina` y la propiedad *contieneIngrediente*.



**FACULTAD DE
CIENCIAS**

Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados
- 7 Conclusiones**
- 8 Trabajos futuros



**FACULTAD DE
CIENCIAS**

- Se realizó la inferencia de platillos de comida costeños peruanos utilizando Spark y la reconstrucción de la ontología de un sistema de cocina inteligente, ampliando su aplicabilidad en la gastronomía peruana.
- Se revisó correctamente el funcionamiento y procesamiento de datos en Spark relacionados con la información de platillos costeños peruanos.
- Se comprendió el funcionamiento de Spark en la inferencia y procesamiento de datos para realizar consultas relacionadas con la gastronomía peruana. logró significativamente mejorar la ontología de un sistema de cocina inteligente, permitiendo consultas exitosas sobre platillos costeños peruanos y su integración con una aplicación móvil mediante la API de OWL.



Contenido

- 1 Introducción
- 2 Problema
- 3 Objetivos
- 4 Conocimientos previos
- 5 Metodología y desarrollo
- 6 Resultados
- 7 Conclusiones
- 8 Trabajos futuros**



**FACULTAD DE
CIENCIAS**

- La ontología mostrada puede también expandirse al campo de la salud, pues se podrían agregar clases relacionadas al valor nutricional de cada alimento para que se pueda tener conocimiento del valor nutricional de cada platillo.
- La revisión de la documentación de la API de owl para que la ontología desarrollada en Protege pueda conectarse de forma correcta con la aplicación Android.

