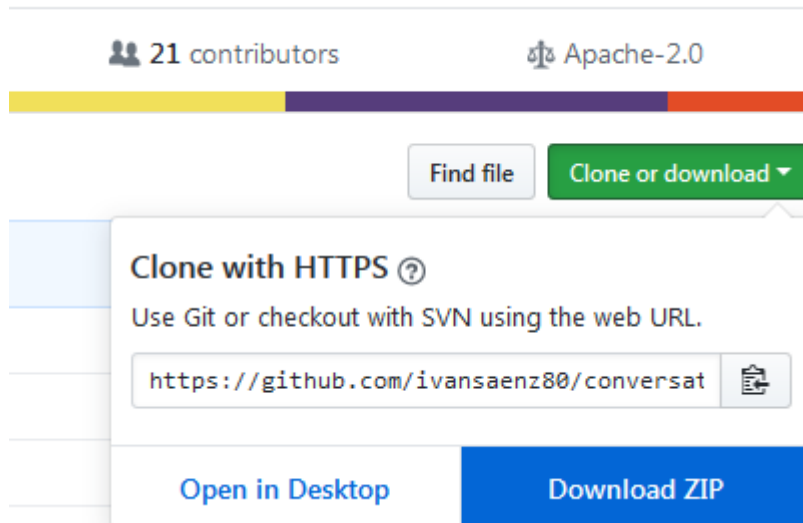
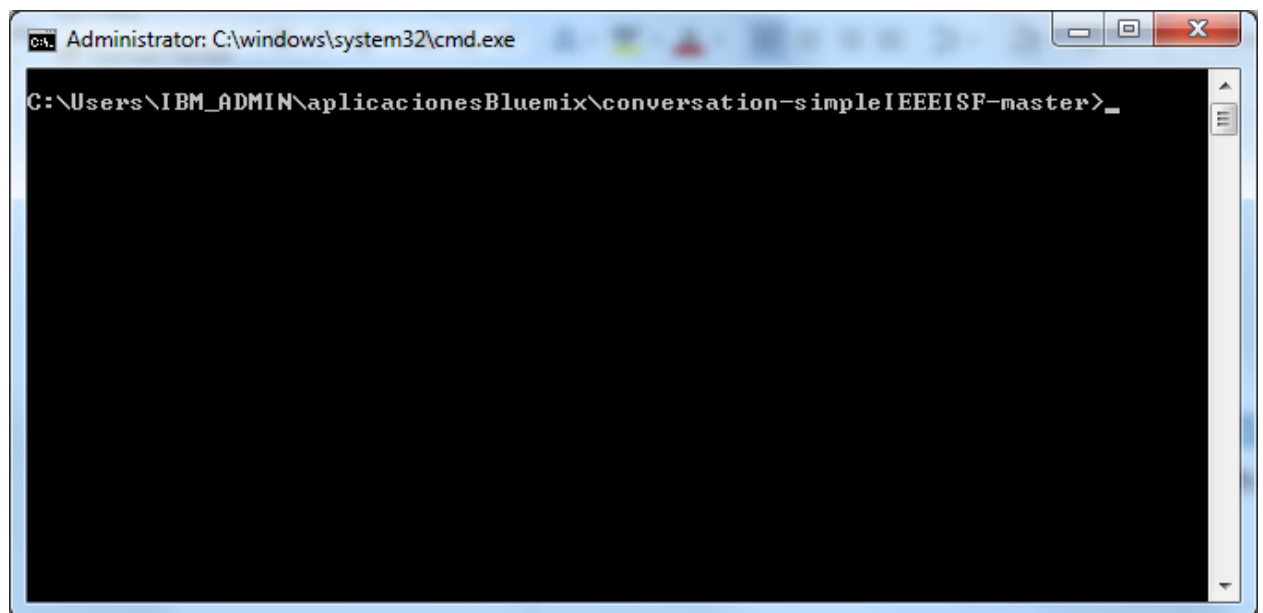


GUIA DE LABORATORIO APLICACIÓN DE PRUEBA WATSON CONVERSATION, CLOUDANT DB Y VISUAL RECOGNITION

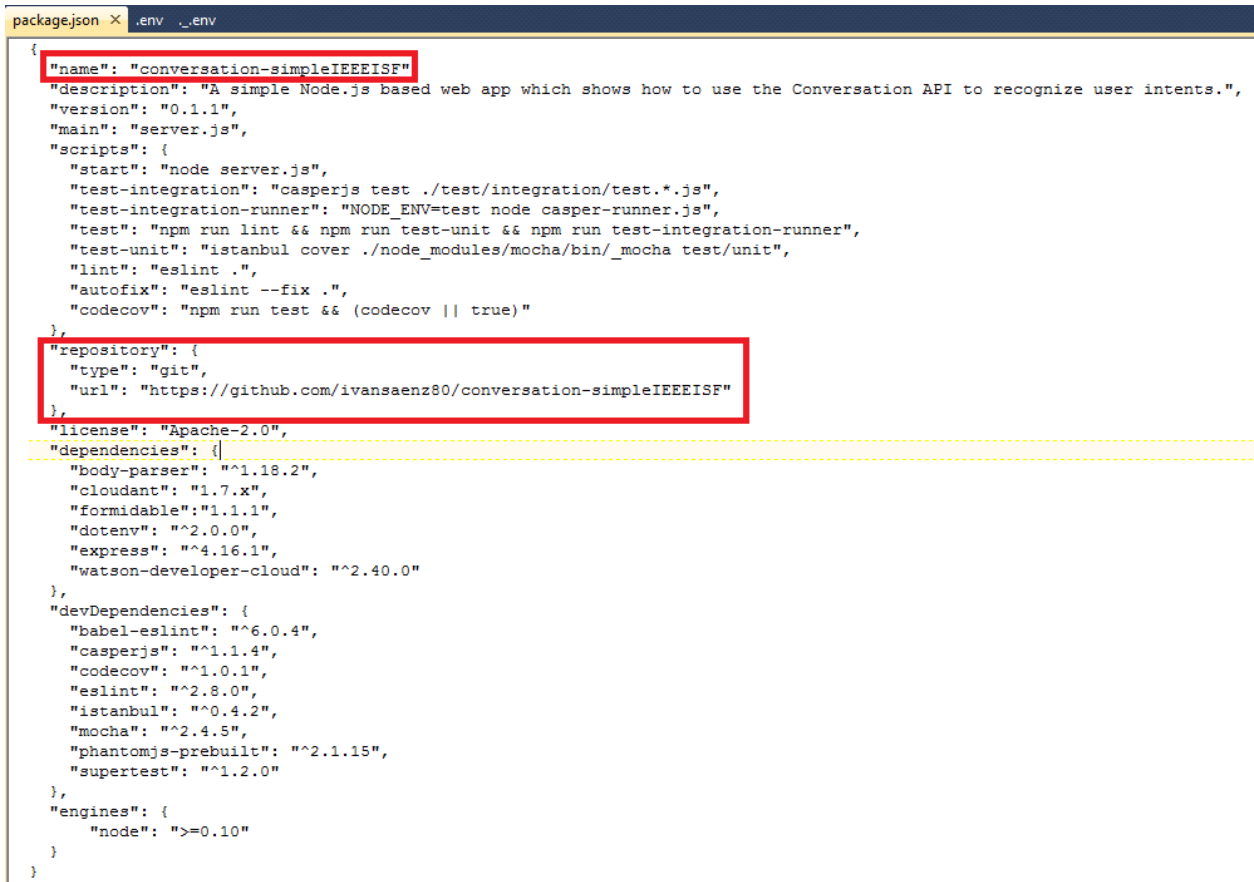
1. Ingresar a : <https://github.com/ivansaenz80/conversation-simpleIEEEISF>
2. Descargamos un ZIP con el código fuente:



3. Extraemos el archivo.
4. Abrimos el terminal y nos ubicamos en la carpeta donde se extrajo el código fuente:

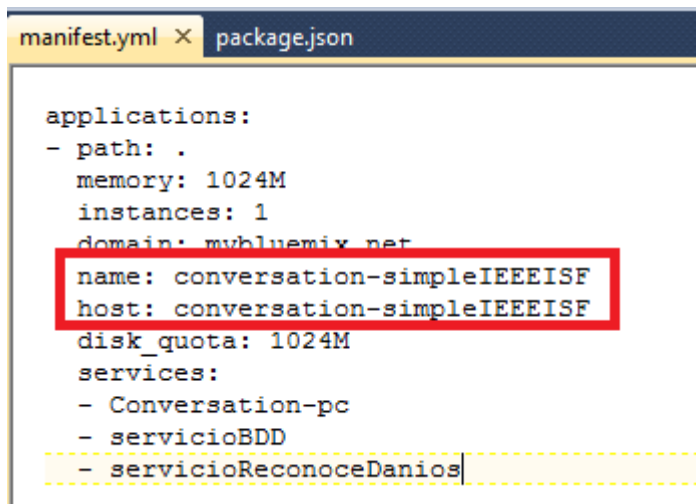


5. Renombramos la carpeta para que la aplicación tenga un nombre único
6. Cambiamos el nombre de la aplicación en el archivo package.json



```
package.json X .env .env
{
  "name": "conversation-simpleIEEEEISF",
  "description": "A simple Node.js based web app which shows how to use the Conversation API to recognize user intents.",
  "version": "0.1.1",
  "main": "server.js",
  "scripts": {
    "start": "node server.js",
    "test-integration": "casperjs test ./test/integration/test/*.js",
    "test-integration-runner": "NODE_ENV=test node casper-runner.js",
    "test": "npm run lint && npm run test-unit && npm run test-integration-runner",
    "test-unit": "istanbul cover ./node_modules/mocha/bin/_mocha test/unit",
    "lint": "eslint .",
    "autofix": "eslint --fix .",
    "codecov": "npm run test && (codecov || true)"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/ivansaenz80/conversation-simpleIEEEEISF"
  },
  "license": "Apache-2.0",
  "dependencies": {
    "body-parser": "^1.18.2",
    "cloudbant": "1.7.x",
    "formidable": "1.1.1",
    "dotenv": "^2.0.0",
    "express": "^4.16.1",
    "watson-developer-cloud": "^2.40.0"
  },
  "devDependencies": {
    "babel-eslint": "^6.0.4",
    "casperjs": "^1.1.4",
    "codecov": "^1.0.1",
    "eslint": "^2.8.0",
    "istanbul": "^0.4.2",
    "mocha": "^2.4.5",
    "phantomjs-prebuilt": "^2.1.15",
    "supertest": "^1.2.0"
  },
  "engines": {
    "node": ">=0.10"
  }
}
```

7. Cambiamos el nombre de la aplicación en el archivo manifest.yml

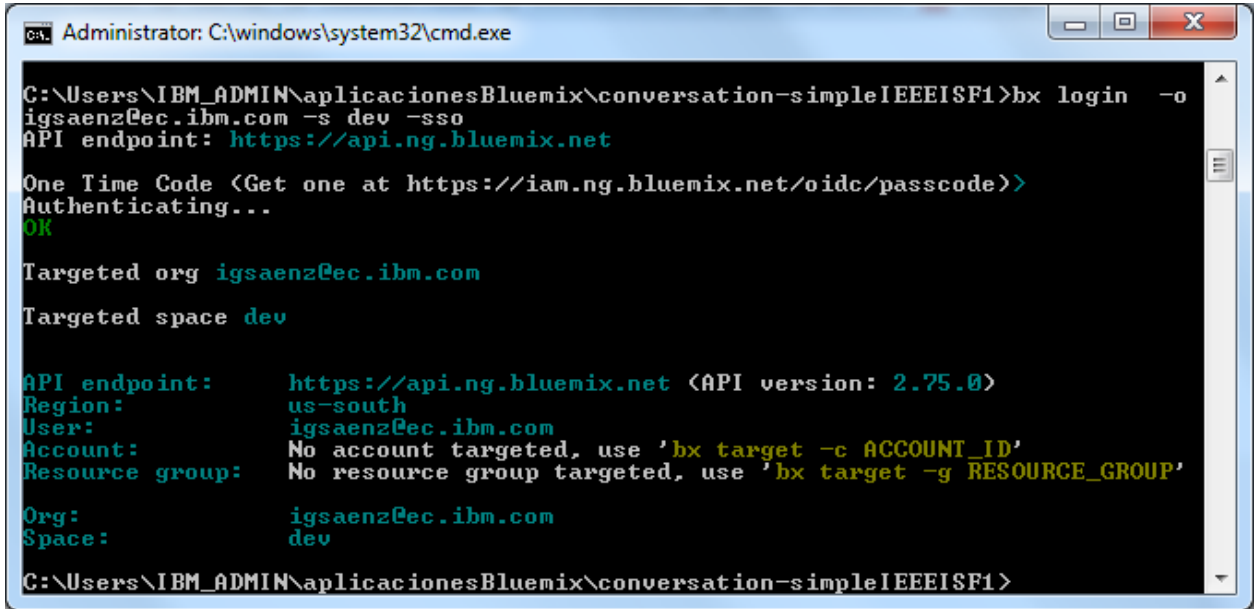


```
manifest.yml X package.json
applications:
- path: .
  memory: 1024M
  instances: 1
  domain: mybluemix.net
  name: conversation-simpleIEEEEISF
  host: conversation-simpleIEEEEISF
  disk_quota: 1024M
  services:
  - Conversation-pc
  - servicioBDD
  - servicioReconoceDanios
```

8. Iniciamos sesión en bluemix con esta serie de comandos:
 - a. `bx api https://api.ng.bluemix.net`

- b. `bx login -u igsaenz@ec.ibm.com -o igsaenz@ec.ibm.com -s dev`, si el procedimiento no funciona por tener una cuenta federada se debe utilizar el comando:

`bx login -o igsaenz@ec.ibm.com -s dev -sso`



```
Administrator: C:\windows\system32\cmd.exe

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>bx login -o
igsaenz@ec.ibm.com -s dev -sso
API endpoint: https://api.ng.bluemix.net

One Time Code <Get one at https://iam.ng.bluemix.net/oidc/passcode>
Authenticating...
OK

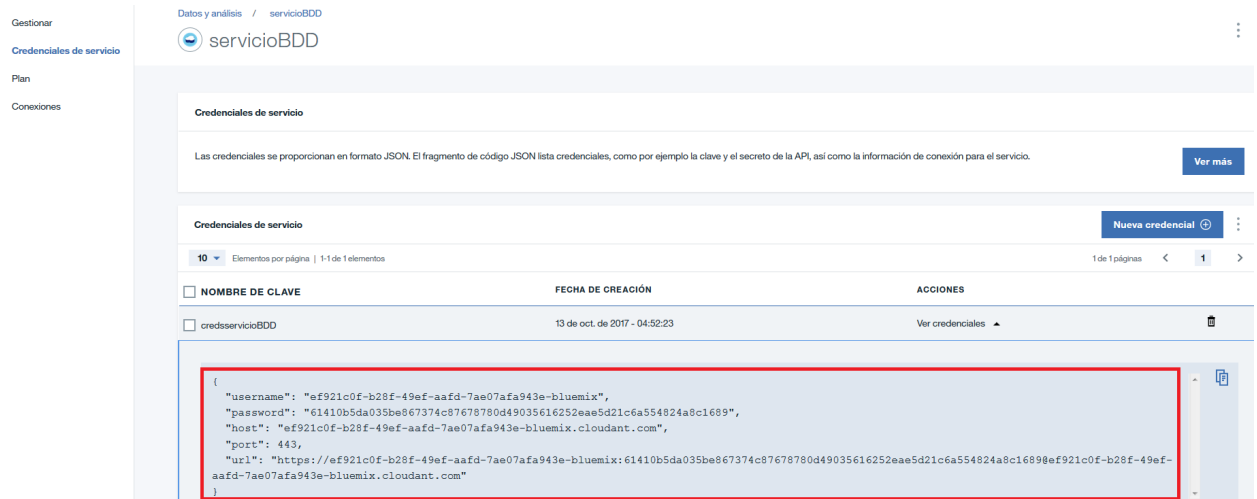
Targeted org igsaenz@ec.ibm.com
Targeted space dev

API endpoint: https://api.ng.bluemix.net <API version: 2.75.0>
Region: us-south
User: igsaenz@ec.ibm.com
Account: No account targeted, use 'bx target -c ACCOUNT_ID'
Resource group: No resource group targeted, use 'bx target -g RESOURCE_GROUP'

Org: igsaenz@ec.ibm.com
Space: dev

C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simple\IEEEISF1>
```

9. Modificamos el archivo `vcap-local.json` con el contenido obtenido de las credenciales del servicio cloudant creado anteriormente desde la consola web de Bluemix:



servicioBDD

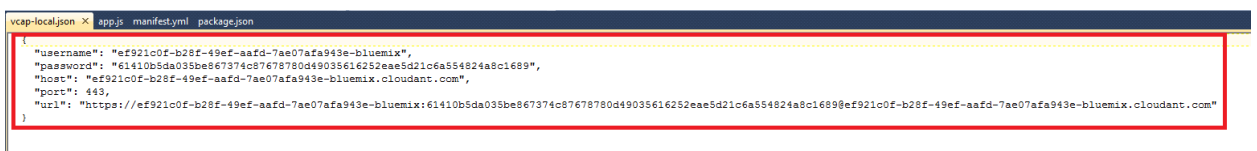
Credenciales de servicio

Las credenciales se proporcionan en formato JSON. El fragmento de código JSON lista credenciales, como por ejemplo la clave y el secreto de la API, así como la información de conexión para el servicio.

Nueva credencial

NOMBRE DE CLAVE	FECHA DE CREACIÓN	ACCIONES
credservicioBDD	13 de oct. de 2017 - 04:52:23	Ver credenciales

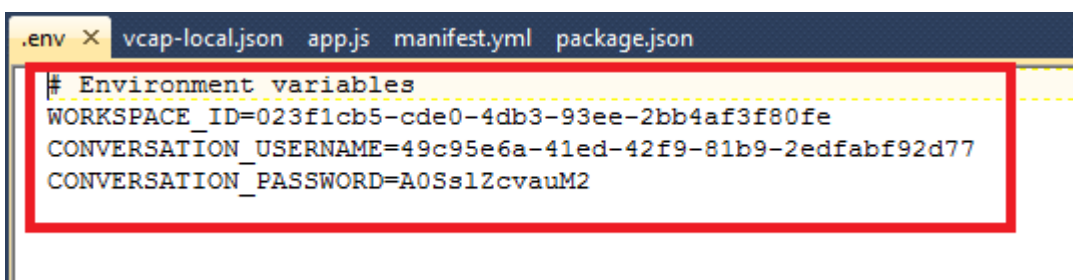
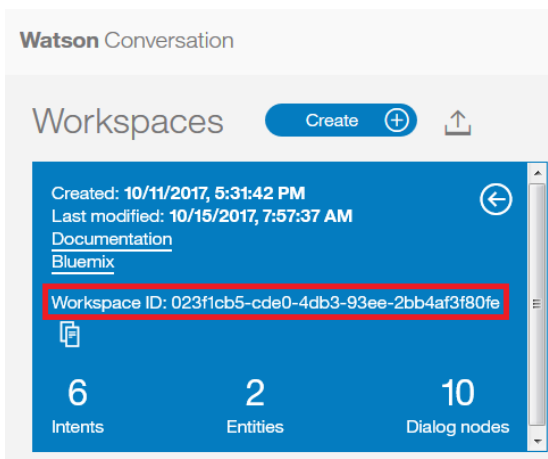
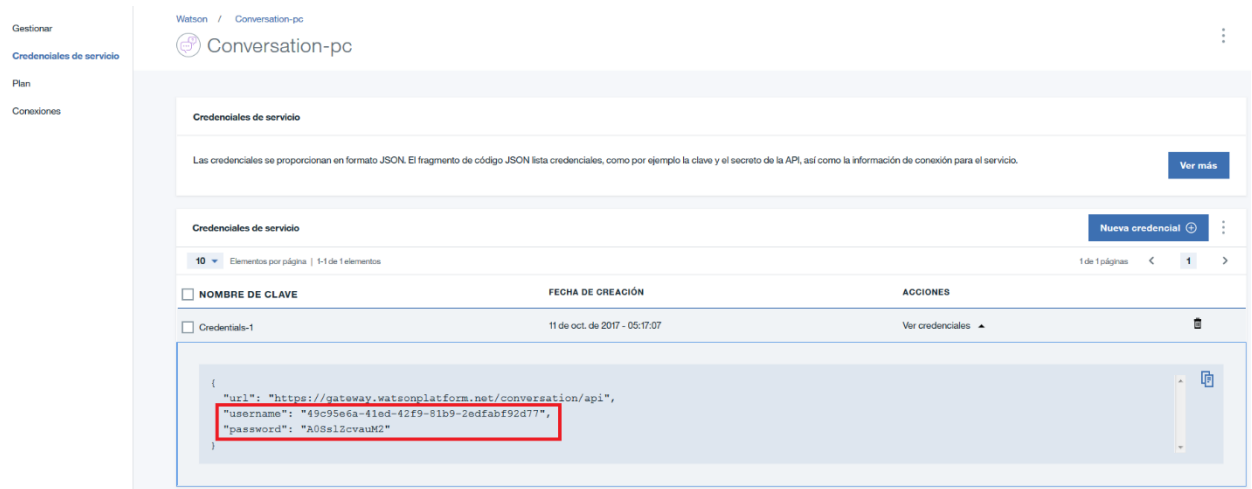
```
{
  "username": "ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix",
  "password": "61410b5da035be867374c87678780d49035616252eae5d21c6a554824a8c1689",
  "host": "ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix.cloudant.com",
  "port": 443,
  "url": "https://ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix:61410b5da035be867374c87678780d49035616252eae5d21c6a554824a8c1689@ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix.cloudant.com"
}
```



```
vcap-local.json X appjs manifest.yml package.json

{
  "username": "ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix",
  "password": "61410b5da035be867374c87678780d49035616252eae5d21c6a554824a8c1689",
  "host": "ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix.cloudant.com",
  "port": 443,
  "url": "https://ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix:61410b5da035be867374c87678780d49035616252eae5d21c6a554824a8c1689@ef921c0f-b28f-49ef-aafd-7ae07afa943e-bluemix.cloudant.com"
}
```

10. Modificamos el archivo .env con el contenido obtenido de las credenciales del servicio Conversation creado anteriormente desde la consola web de Bluemix y con el workspace id que se muestra al lanzar la herramienta de configuración del servicio.



11. Modificar en el archivo app.js línea 35 el valor del campo api_key con el valor obtenido de la consola web de bluemix del servicio Visual Recognition:

Watson / servicioReconoceDanios

servicioReconoceDanios

Gestionar
Credenciales de servicio
Conexiones

Credenciales de servicio

Las credenciales se proporcionan en formato JSON. El fragmento de código JSON lista credenciales, como por ejemplo la clave y el secreto de la API, así como la información de conexión para el servicio. [Ver más](#)

Credenciales de servicio [Nueva credencial](#)

10 Elementos por página | 1-1 de 1 elementos 1 de 1 páginas

NOMBRE DE CLAVE	FECHA DE CREACIÓN	ACCIONES
<input type="checkbox"/> credsservicioVR	14 de oct. de 2017 - 12:13:56	Ver credenciales

```

{
  "url": "https://gateway-a.watsonplatform.net/visual-recognition/api",
  "note": "This is your previous free key. If you want a different one, please wait 24 hours after unbinding the key and try again.",
  "api_key": "295036e67588734c6a3b1c1ee8cd4cc70a571bb1"
}

```

```

.env vcap-local.json app.js manifest.yml package.json
*
* http://www.apache.org/licenses/LICENSE-2.0
*
* Unless required by applicable law or agreed to in writing, software
* distributed under the License is distributed on an "AS IS" BASIS,
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
* See the License for the specific language governing permissions and
* limitations under the License.
*/

'use strict';

var express = require('express'); // app server
var bodyParser = require('body-parser'); // parser for post requests
var Conversation = require('watson-developer-cloud/conversation/v1'); // watson sdk
var fs = require('fs');
var formidable = require('formidable');

var ConversationPanel = require('./public/js/conversation.js');

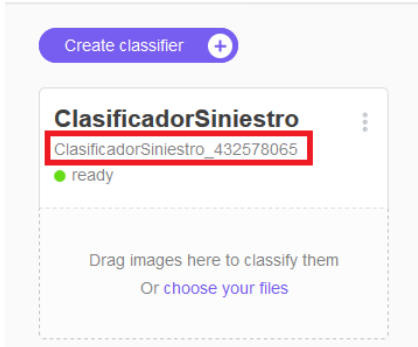
var watson = require('watson-developer-cloud');

var visual_recognition = watson.visual_recognition({
  api_key: '295036e67588734c6a3b1c1ee8cd4cc70a571bb1',
  version: 'v3',
  version_date: '2016-05-20'
});

```

12. Modificar en el archivo app.js línea 78 el valor del campo classifier_ids con el valor obtenido de la consola web de Bluemix del servicio Visual Recognition. En la Visual Recognition Tool:

Visual Recognition Tool

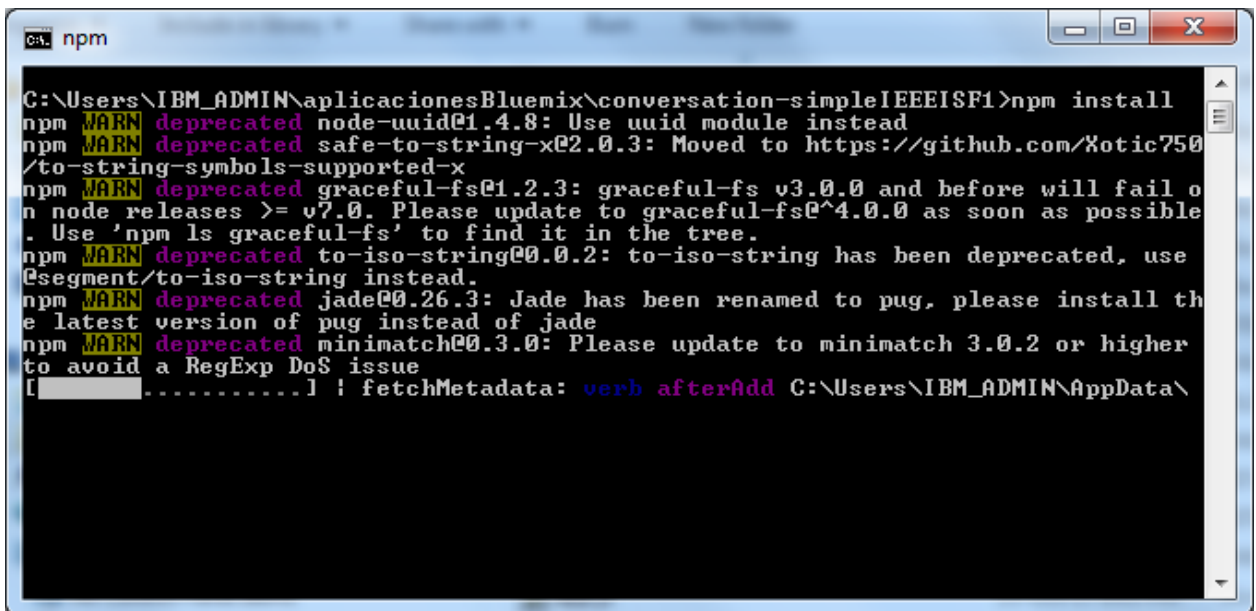


```
datos.json .env vcap-local.json app.js X manifest.yml package.json

});

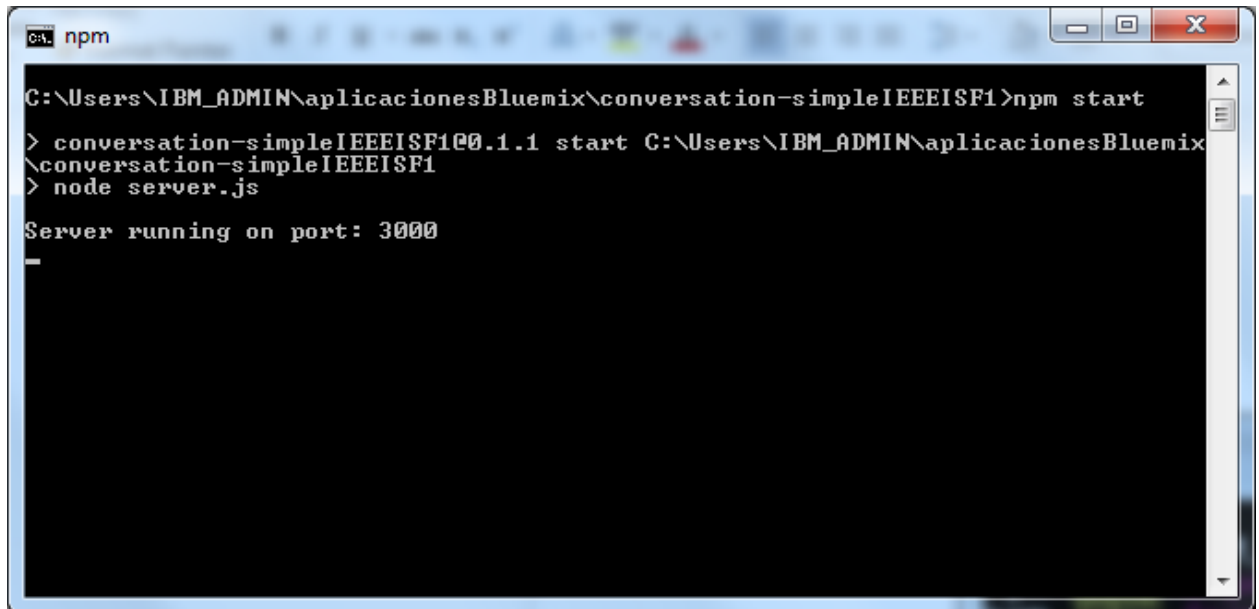
//endpoint para hacer upload a la imagen y clasificarla con Visual Recognition
app.post('/subirFoto', function(req, res) {
  var workspace = process.env.WORKSPACE_ID || '<workspace-id>';
  if (!workspace || workspace === '<workspace-id>') {
    return res.json({
      'output': {
        'text': 'The app has not been configured with a <b>WORKSPACE_ID</b> envi
      }
    });
  }
  var form = new formidable.IncomingForm();
  var params = {
    images_file: "",
    classifier ids: "ClasificadorSiniestro 432578065"
  };
  form.parse(req, function (err, fields, files) {
```

13. Instalamos todos los módulos nodejs necesarios para ejecutar la aplicación con el comando:
npm install



```
cmd npm
C:\Users\IBM_ADMIN\aplicacionesBluenix\conversation-simpleIEEEISF1>npm install
npm WARN deprecated node-uuid@1.4.8: Use uuid module instead
npm WARN deprecated safe-to-string-x@2.0.3: Moved to https://github.com/Xotic750
/to-string-symbols-supported-x
npm WARN deprecated graceful-fs@1.2.3: graceful-fs v3.0.0 and before will fail o
n node releases >= v7.0. Please update to graceful-fs@^4.0.0 as soon as possible
. Use 'npm ls graceful-fs' to find it in the tree.
npm WARN deprecated to-iso-string@0.0.2: to-iso-string has been deprecated, use
@segment/to-iso-string instead.
npm WARN deprecated jade@0.26.3: Jade has been renamed to pug, please install th
e latest version of pug instead of jade
npm WARN deprecated minimatch@0.3.0: Please update to minimatch 3.0.2 or higher
to avoid a RegExp DoS issue
[ ] .....] ! fetchMetadata: verb afterAdd C:\Users\IBM_ADMIN\AppData\
```

14. Ejecutamos la aplicación localmente con el comando: npm start

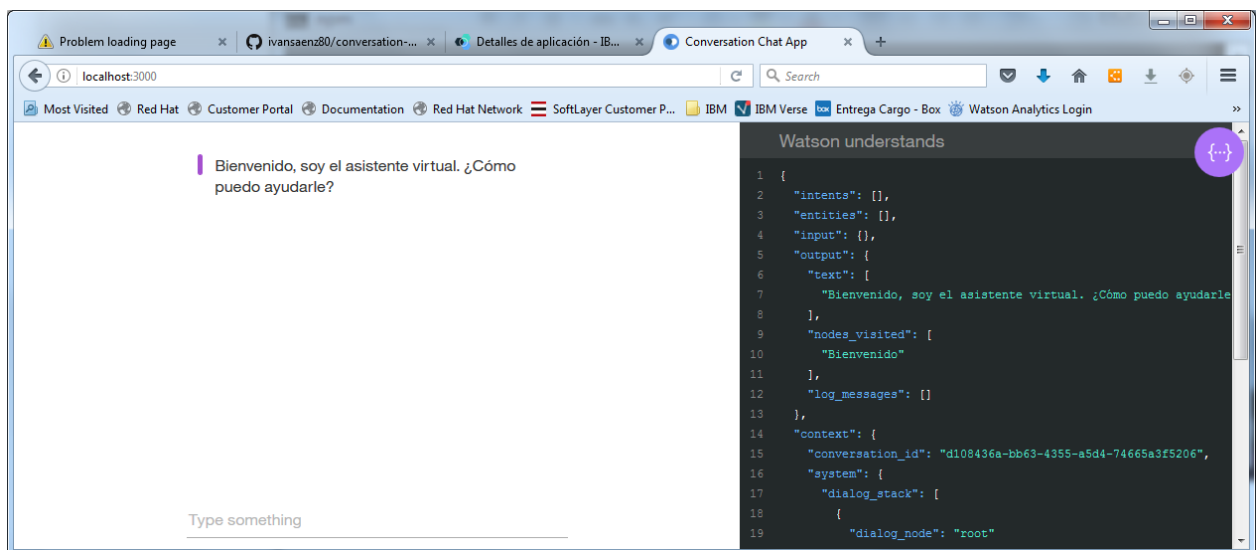


```
C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simpleIEEEISF1>npm start
> conversation-simpleIEEEISF1@0.1.1 start C:\Users\IBM_ADMIN\aplicacionesBluemix\conversation-simpleIEEEISF1
> node server.js

Server running on port: 3000
```

15. Abrimos la aplicación en <http://localhost:3000>

Si se abre la interfaz y se recibe un mensaje de bienvenida la aplicación se ha configurado correctamente:



16. Cualquier cambio sobre el servicio de conversación o reconocimiento visual tendrá un efecto inmediato en la ejecución de la aplicación.