

# Mango Project

## Salesforce Technical Interview

### SUMMARY:

**Company:** Mango (victor.melo@mango.com)

**Agency:** Scalian (laura.escalante)

**Candidate:** Sosthene Grosset-Janin

**Position:** Desarrollador Apex (Remoto)

### Vacancy Description:

¿Qué tareas tendrás?

- Trabajarás en estrecha colaboración con el Salesforce Service Cloud Lead.
- Realizarás el diseño técnico de componentes e integraciones en colaboración con el Salesforce Service Cloud Lead.
- Desarrollarás componentes de front (Lightning Web Components) y componentes de back (Apex).
- Desarrollarás componentes de integración con otras plataformas (Mulesoft, Google Dialogflow, Google Vertex Auto ML, Talkdesk, Qualtrics, Kafka, SAP). Y componentes de integración con IA Generativa, Co-Pilots para los agentes de Atención al Cliente.
- Supervisarás el desarrollo y la funcionalidad de los componentes programados por otros miembros del equipo.
- Supervisarás el desarrollo y la funcionalidad de los componentes programados por otros miembros del equipo.
- Liderarás los test de integración, coordinándote con los equipos implicados.
- Junto al Salesforce Service Cloud Lead investigarás, evaluarás y recomendarás tecnologías y herramientas para mejorar continuamente el proceso de desarrollo y la calidad del software.

¿Qué buscamos en ti?

- Formación en ingeniería informática o similar.
- Al menos 5 años de experiencia trabajando con Apex o Java.
- Experiencia demostrable trabajando con Lightning Web Components o HTML, CSS, Javascript y React o Angular.
- Al menos 2 años realizando diseños técnicos, refinamientos y supervisión de tareas técnicas.
- Valorable experiencia en Salesforce Service Cloud, Salesforce Sales u otro CRM.

**Date:** May 2024

**Version:** 1.0.0

**OBJECTIVE:**

This technical assessment aims to evaluate the competencies and abilities of the candidate related to the vacancy advertised (Salesforce Developer). In this regard, the client company will assess the candidate in the following areas:

- Knowledge of the Salesforce technology stack
- Knowledge of Salesforce Apex programming
- Knowledge of Webservices, and REST APIs
- Knowledge of Salesforce API integration and relevant authentication flows
- Knowledge of TDD methodology (unit test for Apex classes)
- Knowledge of software release lifecycle (deployment)

**REQUIREMENTS:**

Haz una clase en Apex o Java que haga lo siguiente:

Obtener productos de una REST API. Aquí tienes una API pública:

<https://dummyjson.com/docs/products>. Usa autenticación.

Guardar los productos. En Apex puedes guardarlos en el objeto Product. En Java puedes guardarlos en un fichero plano. Con guardar un par de campos del producto es suficiente.

Controla que no se guarden productos con el mismo nombre (title en la API).

Test unitarios. Haz uno o más tests que comprueben la funcionalidad solicitada.

Tómate el tiempo que necesites para hacer esta prueba y envíala por correo electrónico en un fichero adjunto a [victor.melo@mango.com](mailto:victor.melo@mango.com). También puedes subir el código a github si te resulta más sencillo y enviar el enlace al repositorio.

Se valora especialmente el código limpio, una nomenclatura coherente, una modularización adecuada y unos tests que comprueben el trabajo realizado.

**SUCCESS CRITERIA:**

Design, Implementation, Testing, Deployment of Apex code:

Salesforce Apex callout class

Salesforce Apex callout test class

Salesforce Apex callout Mock class

Planning, Implementation, Testing of API calls to 3rd party provider

(DummyJSON - <https://dummyjson.com/docs/products>)

Planning, Implementation of Salesforce package.xml for deployment to target org

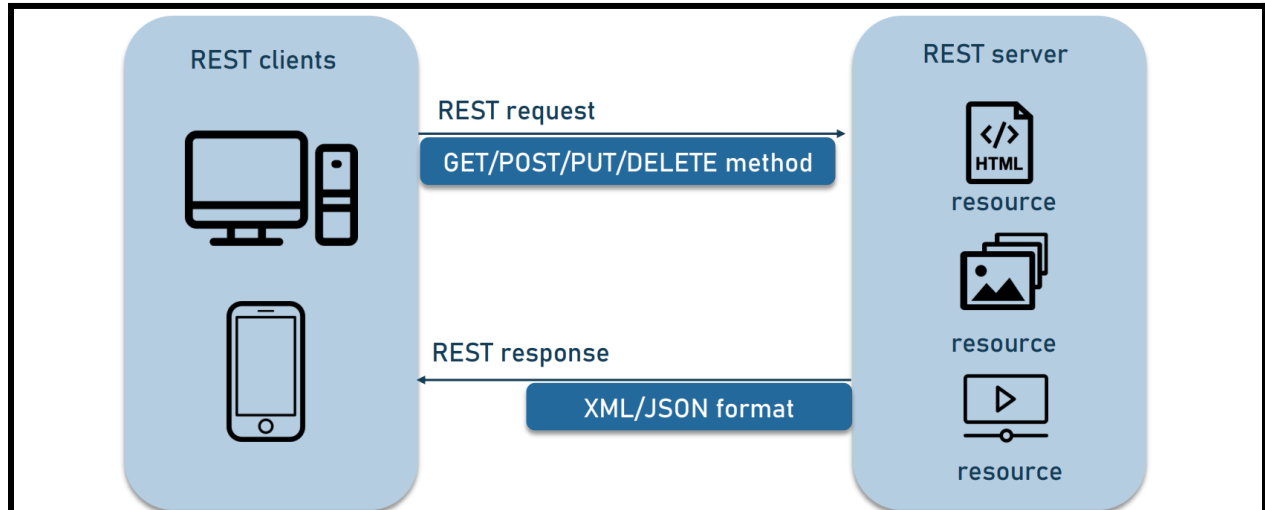
Planning, Implementation of software project to online repository (github)

Documentation of project, end user manual, installation instructions

## DESIGN:

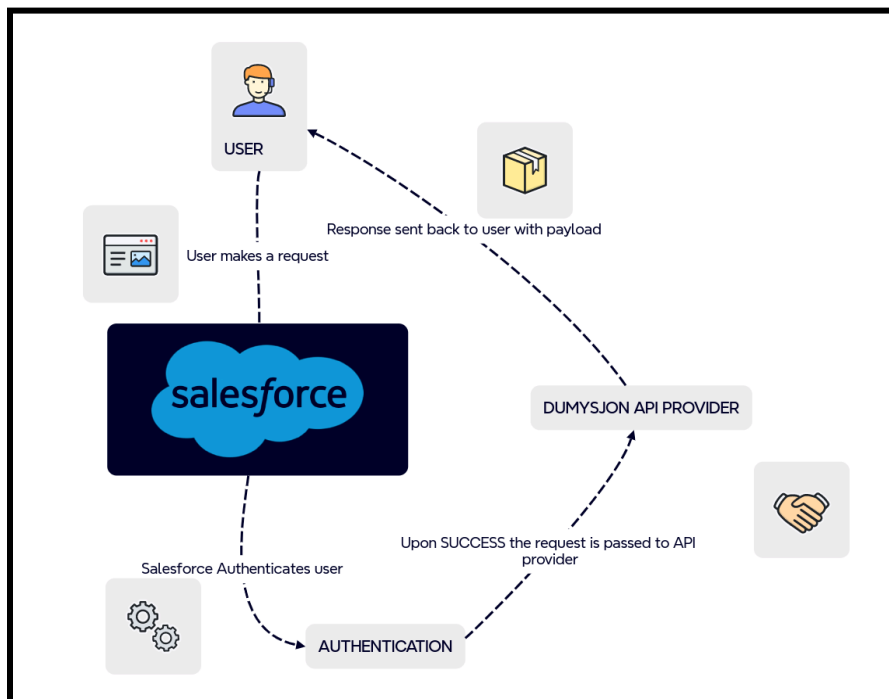
Following the C4 model for software architecture (C1-Context / C2-Container / C3-Component / C4-Code), here is the outline for the design of this project.

### C1-Context



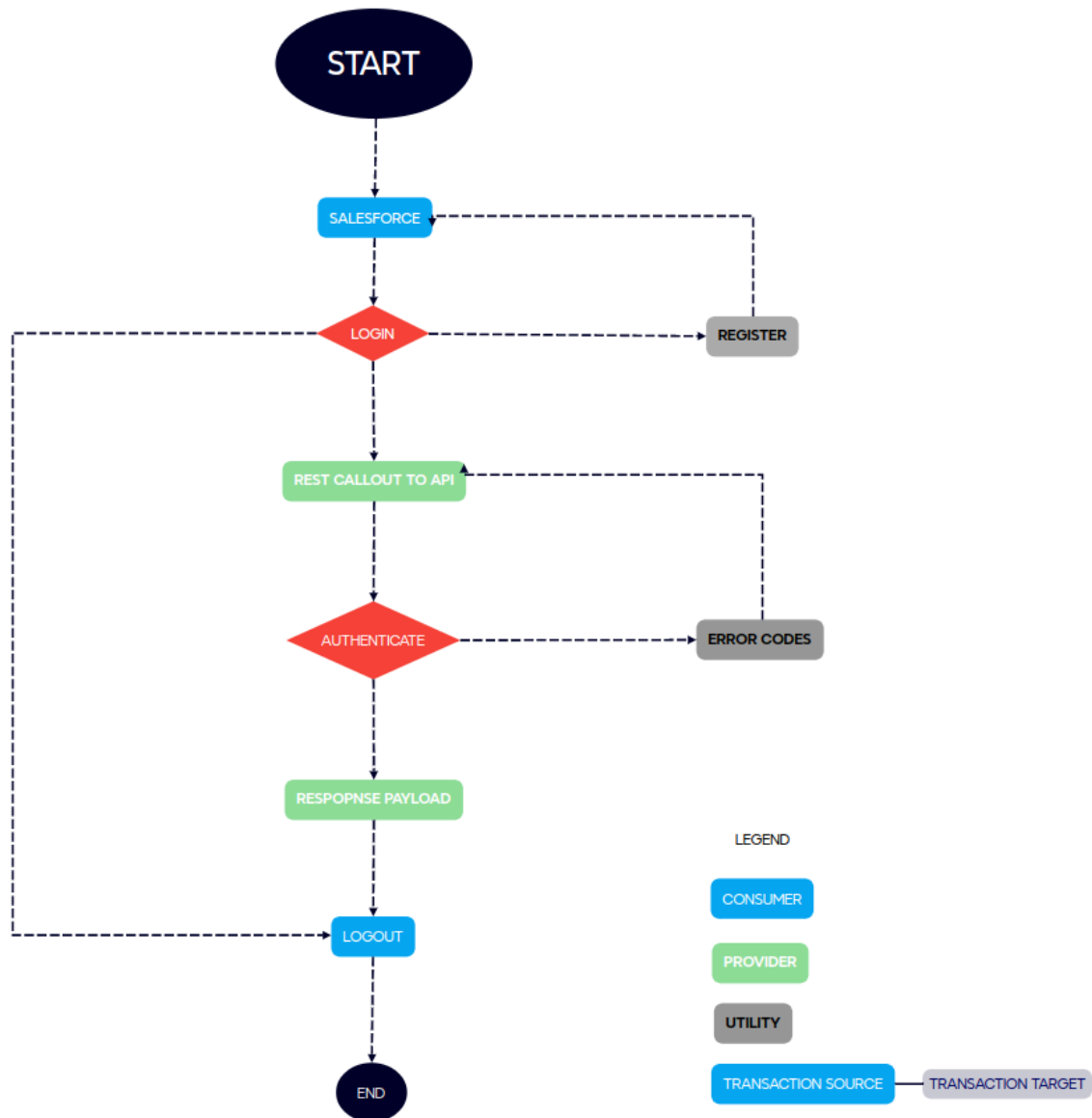
We need to implement a webservice between a Salesforce org and a 3rd party API (DummyJSON).

### C2-Container



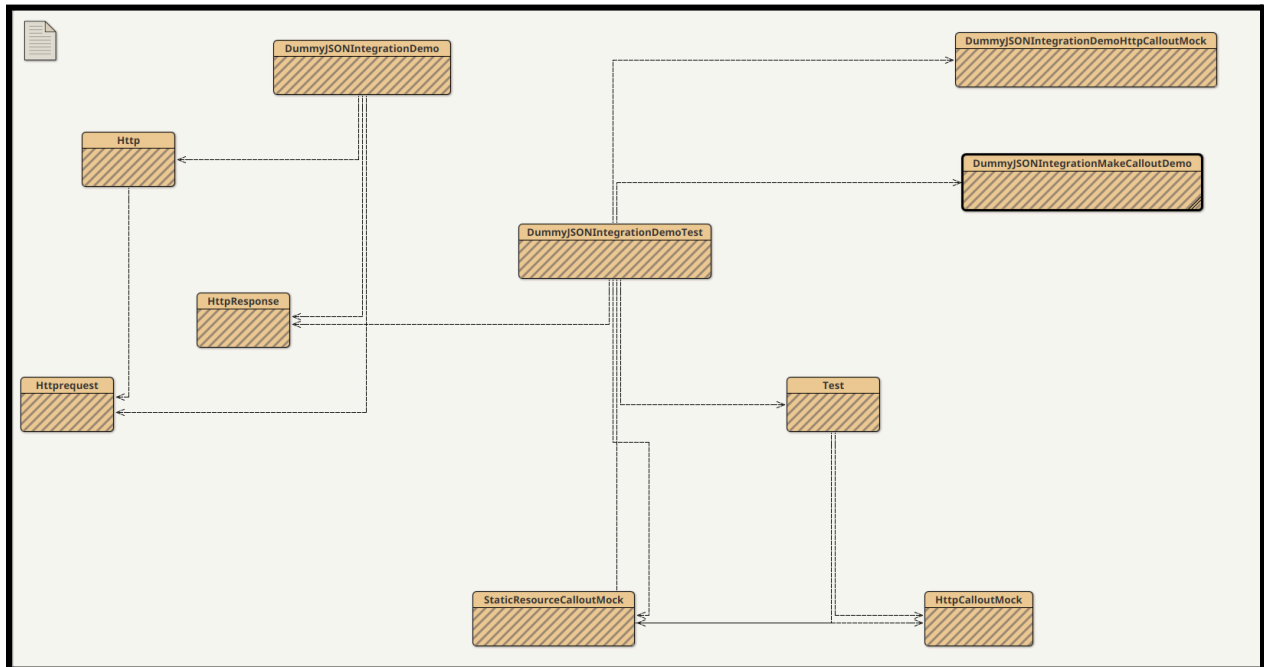
We need to implement a system that takes a request from a user in Salesforce. Salesforce then connects to the 3rd party API using “bearer” authentication. Finally a response is returned to the user with a payload (the information requested).

### C3-Component (flowchart)



We need to implement a workflow which starts from the user's request from within Salesforce, connects to the 3rd party API provider through authentication and returns a payload response back to the user

## C4-Code (ERD)



We need to build the data model, object model, scope model, functional map, data dictionary in order to implement the foundations of this application architecture

**DEVELOPMENT:**

## 1. Provision a Salesforce Sandbox (Playground)

a.

Parameter	Value
URL	https://creative-shark-wbpeem-dev-ed.trailblaze.lightning.force.com/
username	eoexconsulting@creative-shark-wbpeem.com
email	eoexconsulting@gmail.com

b.

## 2. Prepare the Salesforce org for 3rd API callouts using “Remote Site Settings”

a.

Parameter	Value
Remote Site Name	DummyJSONIntegrationDemo
Remote Site URL	https://dummyjson.com
Description	This is the endpoint for the Salesforce webservice demo using REST API callouts from this salesforce org into the DummyJSON API  endpoint: https://dummyjson.com/products/
Active	yes

3. We need to engage/register with the 3rd party API provider
  - a. Some API providers require a registration through a Google account or Github account.
  - b. DummyJSON setup

Parameter	Value
URL	https://dummyjson.com
Endpoint	https://dummyjson.com/docs/products
Access Token	"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTUsInVzZXJuYXW1IjoiY21pbmNoZWxsZSIsImVtYWIsIjoia21pbmNoZWxsZUBxcS5jb20iLCJmaXJzdE5hbWUiOiJKZWZubmUiLCJsYXN0TmFtZSI6IkhkbHZvcnNvbilzImdlbmRlcil6ImZlbWFsZSIsImItYWdlIjoiaHR0cHM6Ly9yb2JvaGFzaC5vcmcvSmVhbW5lLnBuZz9zZXQ9c2V0NCIsImIhdCI6MTcxMTIwOTAwMSwiZXhwIjoxNzExMjE5NjAxQ.F_ZCpi2qdv97grmWiT3h7HcT1prRJasQXjUR4Nk1yo8"
Authentication	
CRUD method	

4. Class Library (**written by the candidate**)
  - a. DummyJSONIntegrationDemo
    - i. public static void testIntegration()
  - b. DummyJSONIntegrationMakeCalloutDemo
    - i. public static HttpResponse makeGetCallout()
  - c. DummyJSONIntegrationDemoTest
    - i. @isTest static void testGetCallout()
    - ii. @isTest static void otherTestGetCallout()
  - d. @isTest global class DummyJSONIntegrationDemoHttpCalloutMock
    - i. global HTTPResponse respond(HTTPRequest request)
5. Utility Class Library (**provided by Salesforce**)
  - a. Httprequest
  - b. Http
  - c. HttpResponse
  - d. StaticResourceCalloutMock
  - e. Test
  - f. HttpCalloutMock
6. Static Resource
  - a. Resource: GetProductResource
  - b. Contents: {"products": ["Coca-Cola Bottle", "Milk Carton", "Champagne Bottle"]}

## TESTING:

### 1. 3rd party API provider testing (<https://dummyjson.com/docs>)

#### a. Get Login and Access Token

```
fetch('https://dummyjson.com/auth/login', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({

    username: 'emilys',
    password: 'emilypass',
    expiresInMins: 30, // optional, defaults to 60
  })
})
.then(res => res.json())
.then(console.log);
```

Response:

```
{
  "id": 1,
  "username": "emilys",
  "email": "emily.johnson@x.dummyjson.com",
  "firstName": "Emily",
  "lastName": "Johnson",
  "gender": "female",
  "image": "https://dummyjson.com/icon/emilys/128",
  "token":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTUsInVzZXJuY
  WllIjoia21pbmNoZWxsZSIsImVtYWlsIjoia21pbmNoZWxsZUBxcS5jb20i
  LCJmaXJzdE5hbWUiOiJKZWZubmUiLCJsYXN0TmFtZSI6IkhkbHZvcnNvbiI
  sImdlbmRlciI6ImZlbWFsZSIsImltYWdlIjoiaHR0cHM6Ly9yb2JvaGFzaC
  5vcmcvSmVhbm5lLnBuZz9zZXQ9c2V0NCIsImVhdCI6MTcxMTIwOTAwMSwiZ
  XhwIjojoxNzExMjEyNjAxfQ.F_ZCpi2qdv97grmWiT3h7HcT1prRJasQXjUR4
  Nk1yo8"
}
```



### b. Get all Products

```
fetch('https://dummyjson.com/products')
  .then(res => res.json())
  .then(console.log);
```

Response:

```
{
  "products": [
    {
      "id": 1,
      "title": "Essence Mascara Lash Princess",
      "description": "The Essence Mascara Lash Princess is
a popular mascara known for its volumizing and lengthening
effects. Achieve dramatic lashes with this long-lasting and
cruelty-free formula.",
      "category": "beauty",
      "price": 9.99,
      "discountPercentage": 7.17,
      "rating": 4.94,
      "stock": 5,
      "tags": [
        "beauty",
        "mascara"
      ],
      "brand": "Essence",
      "sku": "RCH45Q1A",
      "weight": 2,
      ...
    },
    ...
  ],
  "total": 194,
  "skip": 0,
  "limit": 30
}
```

### c. Apex Unit Test

	Result	Coverage
i. Class tested: DummyJSONIntegrationDemo	pass	100%
ii. Class: DummyJSONIntegrationDemoTest	pass	
1. @isTest static void testGetCallout()	pass	
2. @isTest static void otherTestGetCallout()	pass	

## DELIVERY:

1. Salesforce package.xml deployment

### Apex Classes:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://soap.sforce.com/2006/04/metadata">
  <types>
    <members>DummyJSONIntegrationDemo</members>
    <members>DummyJSONIntegrationMakeCalloutDemo</members>
    <members>DummyJSONIntegrationDemoTest</members>
    <members>DummyJSONIntegrationDemoHttpCalloutMock</members>
    <name>ApexClass</name>
  </types>
  <version>51.0</version>
</Package>
```

### Remote Site Settings:

```
<?xml version="1.0" encoding="UTF-8"?>
<RemoteSiteSetting xmlns="http://soap.sforce.com/2006/04/metadata">
  <description>This is the endpoint for the Salesforce webservices demo
  using REST API callouts from this salesforce org into the DummyJSON API

  endpoint: https://dummyjson.com/products/
</description>
  <disableProtocolSecurity>>false</disableProtocolSecurity>
  <isActive>true</isActive>
  <url>https://dummyjson.com</url>
</RemoteSiteSetting>
```

### Static Resource:

```
<?xml version="1.0" encoding="UTF-8"?>
<StaticResource xmlns="http://soap.sforce.com/2006/04/metadata">
  <contentType>text/plain</contentType>
  <description>Test Resource</description>
</StaticResource>
```

### Project Repository (Github):

<https://github.com/sgrosset/Mgo-Project>

### Output:

Execution Log		
Timestamp	Event	Details
06:26:16.978	USER_DEBUG	[14]DEBUG(response status --> 200
06:26:16.981	USER_DEBUG	[15]DEBUG(response check --> [{"products":[{"id":1,"title":"Essence Mascara Lash Princess","description":"The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening effects.
06:26:17.058	USER_DEBUG	[29]DEBUG(Received the following products:
06:26:17.058	USER_DEBUG	[30]DEBUG([{"availabilityStatus":Low Stock, brand=Essence, category=beauty, description=The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening effects. Achieve drama