

# Salesforce Platform Developer I

Version	Fecha	Autor	Descripción
1.0	29/04/2024	Adrián Arribas	Documento de estudio para el examen de Salesforce Platform Developer I (Sin organizar)

## To-do

- ☐ Organizar por temas
- ☐ Audit fields ¿q son?
- ☐ Que es un campo de fórmula

## Introducción

El examen de Salesforce Platform Developer I es un examen que evalúa los conocimientos de un desarrollador de Salesforce. En este documento se recogen los conceptos más importantes que se deben conocer para aprobar el examen.

Concepto	Descripción
Puntuación	68%
Tiempo	105 minutos
Preguntas	60

## Topics

Reorganizar por temas

Developer Fundamentals

Process Automation and Logic

User Interface

Testing, Debugging, and Deployment

Interfaz de Usuario

Metodos utiles para Visualforce

- **Apex:** `detail` -> Muestra los detalles de un registro
- **renderAs:** Imprime la página
- `getSelected()` records

Standard controller methods

- **Apex:** `save` -> Guarda el registro

- **Apex:** `cancel` -> Cancela la edición
- **Apex:** `delete` -> Elimina el registro
- **Apex:** `edit` -> Edita el registro
- **Apex:** `quickSave` -> Guarda el registro
- **Apex:** `list` -> Lista los registros

## App Builder

### Componentes

- **Standard:** Componentes que vienen por defecto en Salesforce
- **Custom:** Componentes que se pueden crear
- **Custom Managed:** Componentes que se pueden instalar desde AppExchange
- **Custom Unmanaged:** Componentes que se pueden crear y modificar
- **AppExchange:** Componentes que se pueden instalar desde AppExchange
- **Flow:** Componentes que se pueden crear con Flow
- **Lightning Page:** Componentes que se pueden crear con Lightning Page
- **Visualforce Page:** Componentes que se pueden crear con Visualforce Page
- **Web Tab:** Componentes que se pueden crear con Web Tab
- **Aura Component:** Componentes que se pueden crear con Aura Component
- **Canvas App:** Componentes que se pueden crear con Canvas App
- **Chatter Tab:** Componentes que se pueden crear con Chatter Tab
- **Dashboard:** Componentes que se pueden crear con Dashboard
- **HomePage Component:** Componentes que se pueden crear con HomePage Component
- **HomePage Layout:** Componentes que se pueden crear con HomePage Layout
- **Report:** Componentes que se pueden crear con Report
- **Report Type:** Componentes que se pueden crear con Report Type
- **S-Control:** Componentes que se pueden crear con S-Control
- **Site.com Page:** Componentes que se pueden crear con Site.com Page
- **Visualforce Component:** Componentes que se pueden crear con Visualforce Component
- **Visualforce Tab:** Componentes que se pueden crear con Visualforce Tab

### Lightning Message Service

- **LMS:** Permite la comunicación entre componentes de Lightning Web Components y Aura Components

### Tooling API ApexLog

- **ApexLog:** Permite ver los logs de Apex
  - Partes de un log
    - Header
    - LogLines
    - Execution Units
    - Code Units

### Logs levels

- NONE
- ERROR

- WARN
- INFO
- DEBUG
- FINE
- FINER
- FINEST

**Traced entities:** Permite ver las entidades que se han trazado

- Automated Process
- Platform Integration
- User
- Apex Class
- Apex Trigger
- 

## Relaciones

- **Master-Detail:** Relación que permite que un registro hijo dependa de un registro padre
  - Si se elimina el registro padre, se eliminan los registros hijos
  - Un objeto custom no puede estar en el lado de "Master" con un objeto estándar
  - Un objeto custom en el lado de "Detail" no puede tener colas.
- **Lookup:** Relación que permite que un registro hijo dependa de un registro padre

## Apex

### Tipos

#### Primitives:

Estos tipos no necesitan ser instanciados.

- **Boolean:** Boolean
- **Date:** Date
- **Date and Time:** Datetime
- **Number:** Decimal, Double, Integer, Long
- **String:** String
- **Time:** Time
- **ID:** ID

#### Collections

- **List:** List
- **Set:** Set
- **Map:** Map

### Bucles

- **For:** Bucle que se ejecuta un número determinado de veces

- **While:** Bucle que se ejecuta mientras se cumpla una condición
- **Do-While:** Bucle que se ejecuta al menos una vez y luego se ejecuta mientras se cumpla una condición

## Clases que se pueden usar

- Database
- **Database:** Clase que permite hacer operaciones de DML
- JSON
- Math
- System
  - **System:** Clase que permite hacer operaciones de sistema
- Metadata
- Schema
- Test

## Orden de ejecución salesforce

- Before Trigger
- Validation Rules
- Save to the database (no commit)
- After Trigger
- Assignment Rules
- Auto-Response Rules
- Workflow Field Updates
- before and after triggers one more time
- Escalation Rules
- Parent Rollup Summary Fields
- Criteria Based Sharing Rules
- Commit to the database all DML operations
- Post-commit logic

## Tipos de clases

- **Standard:** Clases que vienen por defecto en Salesforce
- **Custom:** Clases que se pueden crear
- **Abstract:** Clases que no se pueden instanciar
  - No se pueden instanciar
  - Se pueden usar para heredar
  - Se pueden usar para implementar interfaces
- Virtual
  - Se pueden instanciar
  - Se pueden usar para heredar
  - Se pueden usar para implementar interfaces

## Cosas de queries

### Database.query

- **Database.query:** Permite hacer queries dinámicas
- **Database.queryLocator:** Permite hacer queries dinámicas y soporta más de 50,000 registros

## Anotaciones

- **@future:** Permite ejecutar un método de forma asíncrona
- **@isTest:** Indica que el método es un test
- **@TestVisible:** Indica que el método es visible para los tests
- **@AuraEnabled:** Indica que el método es visible para los componentes de Lightning
- **@InvocableMethod:** Indica que el método es visible para los Flows

## Testing

- **System.assert:** Permite hacer aserciones
- **System.assertEquals:** Permite hacer aserciones
- **System.assertNotEquals:** Permite hacer aserciones

## Colecciones

- **List:** Colección que permite almacenar varios elementos
- **Set:** Colección que permite almacenar varios elementos sin duplicados
- **Map:** Colección que permite almacenar varios elementos con clave-valor

## Exception

### Que es una excepción?

- Una excepción es un error que ocurre en tiempo de ejecución
- Las excepciones pueden ser atrapadas y manejadas
- Las excepciones pueden ser lanzadas
- Las excepciones pueden ser lanzadas por el sistema o por el usuario

### Tipos de excepciones

- **DMLException:** Excepción que ocurre cuando hay un error en una operación de DML
- TODO rellenar
- 

## Flows

### Condiciones de Flows

- Only when a record is created
- Only when a record is updated
- Only when a record is updated to meet the condition requirements

### Elementos de Flows

- **Action:** Elemento que permite hacer una acción
- **Assignment:** Elemento que permite asignar un valor a una variable
- **Choice:** Elemento que permite tomar decisiones
- **Collection:** Elemento que permite almacenar varios elementos
- **Data:** Elemento que permite almacenar datos
- **Loop:** Elemento que permite hacer bucles
- **Screen:** Elemento que permite mostrar una pantalla
- **Subflow:** Elemento que permite llamar a otro flow
- **Text Template:** Elemento que permite mostrar texto

## Funciones importantes de Flows

- Update Records
  - Permite actualizar registros de un objeto seleccionando los campos que se quieren actualizar
- Create Records
- Delete Records
- Get Records

## Custom Events

this.eventData = event.detail; this.eventData = event.detail.value; this.eventData = event.detail.recordId;

## Crypt and decrypt

- **Crypto:** Permite encriptar y desencriptar datos

```
Blob cryptoKey = Crypto.generateAesKey(256);
Blob data = Blob.valueOf('Hello, World!');
Blob encryptedData = Crypto.encrypt('AES256', cryptoKey, data);
Blob decryptedData = Crypto.decrypt('AES256', cryptoKey, encryptedData);

System.assertEquals(data, decryptedData);
```

## Schema

- **Schema:** Permite obtener información de los objetos de Salesforce

```
Schema.DescribeSObjectResult objectResult = Account.SObjectType.getDescribe();
System.debug('Is the object createable? ' + objectResult.isCreateable());
```

## DescribeFieldResult

- **DescribeFieldResult:** Permite obtener información de un campo

## DescribeSObjectResult

- **DescribeSObjectResult:** Permite obtener información de un objeto

## IsUpdateable

- **IsUpdateable:** Permite saber si un campo es editable

```
Schema.DescribeFieldResult fieldResult = Account.Name.getDescribe();
System.debug('Is the field updateable? ' + fieldResult.isUpdateable());
```

## Que es una clase de servicio

### Custom Buttons

They can be added to standard page layouts and work with standard and custom objects

## Modelo Vista Controlador

Añadir foto de MVC

### Modelo

- **Modelo:** Representa los datos y las reglas de negocio

### Vista

- **Vista:** Representa la interfaz de usuario

### Controlador

- **Controlador:** Representa la lógica de negocio

## Importación y Exportación de Datos

### Importación

- **Data Import Wizard:** Permite importar datos de forma sencilla
- **Data Loader:** Permite importar datos de forma masiva

### Exportación

- **Data Export:** Permite exportar datos de forma sencilla

todo tabla con numero de registros que se pueden exportar segun el tipo de licencia

## Testing

Cuanta cobertura de código se necesita?

- 75% de cobertura de código para clases Apex
- 1% de cobertura de código para triggers Donde se puede hacer Run All

- Developer Console
- Capabilities:
  - Develop a Visualforce page
  - View and examine debug logs
  - Execute anonymous Apex code
- Setup Apex Classes

## To-do

- getRecord() vs getRecords()
  - getRecord() -> Devuelve un solo registro
  - getRecords() -> Devuelve varios registros ????

## Field deletion

- Cuando se elimina un campo de un objeto No se puede borrar un campo si está en uso en:
- Fórmulas
- Cláusulas de búsqueda
- Workflow field updates

## Restricciones al ser multitenant

- Tiempo de ejecución de SOQL (10 segundos)
- Tiempo de ejecución de CPU (10 segundos)
- Número de registros devueltos por una consulta SOQL (10,000)

## Seguridad

- **OWD: Organization Wide Defaults** Permite definir la visibilidad de los registros

## A nivel de codio

### XSS Threats

- JSINHTMLENCODE()
  - Evita que se ejecute código JavaScript
- HTMLENCODE()
  - Evita que se ejecute código HTML
- JSENCODE()
  - Evita que se ejecute código JavaScript

## Einstein

- **Einstein:** Permite hacer inteligencia artificial en Salesforce Next Best Action
- **\*\* Einstein Activity Capture:\*\*** Permite capturar las actividades de los usuarios
  - ermite capturar las actividades de los usuarios de Gmail
  - Permite capturar las actividades de los usuarios de Office 365
  - Permite capturar las actividades de los usuarios de Exchange
  -



# Ant Migration Tool

**Ant Migration Tool:** Permite migrar metadatos de Salesforce

- Metadata Component Member
- Metadata Component Type

## Code Builder

**Code Builder:** Permite escribir código en Salesforce

- Apex
- Lightning Web Components
- Visualforce

## Salesforce DX

**Salesforce DX:** Permite desarrollar en Salesforce

- Salesforce Extensions Pack
  - Ejecutar consultas SOQL
  - Desplegar componentes de metadata
  - Lanzar Apex tests
- Version Control Systems
- Salesforce CLI

## Sharing

- **With sharing:** Permite que se apliquen las reglas de seguridad de Salesforce Without sharing
- **Without sharing:** No aplica las reglas de seguridad de Salesforce
- **Inherited sharing:** Hereda las reglas de seguridad de Salesforce

## Excepciones

- No podemos obtener debug logs de Time-Based Workflow porque se ejecutan en un contexto de sistema.

## Preguntas

- ¿Qué es un campo de fórmula?
  - Un campo que calcula un valor en función de otros campos

## Variables

- old y new
  - old -> Valores antiguos
  - new -> Valores nuevos
- prior y after
  - prior -> Valores antiguos
  - after -> Valores nuevos

# Salesforce Extension Pack

## Dev hub

que es?

- **Dev Hub:** Permite desarrollar en Salesforce
- **Dev Hub:** Permite crear Scratch Orgs
- **Dev Hub:** Permite crear Sandboxes

## Salesforce CLI

que es?

- **Salesforce CLI:** Permite ejecutar comandos de Salesforce
- **Salesforce CLI:** Permite ejecutar comandos de Salesforce DX
- **Salesforce CLI:** Permite ejecutar comandos de Salesforce Extensions

## WorkBench

que es?

- **WorkBench:** Permite hacer operaciones de Salesforce
- **WorkBench:** Permite hacer operaciones de Salesforce DX
- **WorkBench:** Permite hacer operaciones de Salesforce Extensions
- Insertar registros
- Actualizar registros
- Borrar registros
- Ejecutar SOQL y SOSL
- REST API
- Describe Metadata and data

## Developer Console

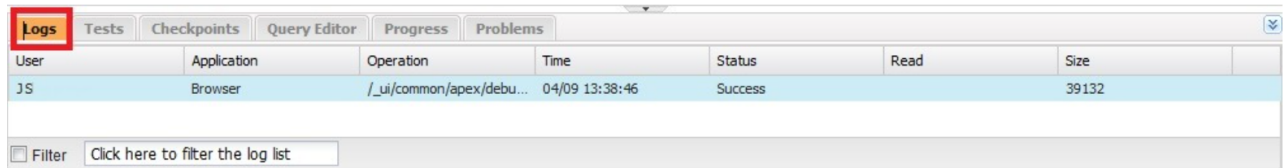
que podemos ver y hacer

- **Developer Console:** Permite hacer operaciones de Salesforce
- **Developer Console:** Permite hacer operaciones de Salesforce DX
- Customizable

Que tabs podemos ver

- Logs
- Tests
- Checkpoints: Contienen lo siguiente:
  - Namespace
  - Class
  - Line
  - Date
- Query Editor

- Progress
- Problems



User	Application	Operation	Time	Status	Read	Size
JS	Browser	/_ui/common/apex/debu...	04/09 13:38:46	Success		39132

## Referencias

Creado por [Adrián Arribas](#) para la preparación del examen de Salesforce Platform Developer I

## Checkpoint Inspector

- **Checkpoint Inspector:** Permite ver los checkpoints de un Flow
- **Checkpoint Inspector:** Permite ver los checkpoints de un Process Builder
- **Checkpoint Inspector:** Permite ver los checkpoints de un Workflow
- **Checkpoint Inspector:** Permite ver los checkpoints de un Flow Builder
- **Checkpoint Inspector:** Permite ver los checkpoints de Apex Code

## Temas

### Developer Fundamentals

- **Developer Fundamentals:** Permite aprender los conceptos básicos de Salesforce

### Process Automation and Logic

### User Interface

### Testing, Debugging, and Deployment

- Dividido en 4 Objetivos
  - Testing
  - Debugging
  - Deployment
  - Performance

### Anonymous Code

- **Anonymous Code:** Permite ejecutar código anónimo en Salesforce
- Se ejecuta como el usuario que lo ejecuta

### Visual Studio Code Salesforce DX

- Apex Unit Tests can be executed from the Code Editor
- Updates from other developers can be tracked and retrieved from a central repository
- Metadata components from the org can be modifies

## Debug logs set flags

## Scheduled Jobs and Apex Jobs

## Performance -> Execution Units debug logs

## Debug level

## User Interface

- **User Interface:** Permite aprender los conceptos de la interfaz de usuario de Salesforce

## Lightning Web Components

Compuestos por los siguientes elementos

- **HTML:** Permite definir la estructura de la página UI
  - No es necesario si se trata de un Service Component
- **CSS:** Permite definir el estilo de la página [Optional]
- **JavaScript:** Permite definir la lógica de la página
- **XML:** Permite definir los metadatos de la página
- **SVG:** Permite definir gráficos vectoriales

Ventajas de usar un Framework Lightning Component:

- Event Driven Architecture
  - Permite que los componentes se comuniquen entre sí
  - Better for decoupling components
  - Device Aware Capabilities and Cross Browser Compatibility

todo colocar esto en algun sitio

Lanzar una query apex corre en system mode por defecto Las DML operations corren en user mode por defecto

## Lightning Message Service

- **LMS:** Permite la comunicación entre componentes de Lightning Web Components y Aura Components

```
import sampleChange from
 '@salesforce/messageChannel/SampleMessageChannel__c';

this.tabId = message.tabId;
```

- Visualforce Overrides: Permite sobrescribir la interfaz de usuario de Salesforce
  - Edit
  - View

## Lightning Style Sheets

- **LSS:** Permite personalizar el estilo de los componentes de Lightning

```
<apex:page standardController="Account" lightningStylesheets="true">
  <body class="slds-vf-scope">
    <apex:form>
      <apex:inputField value="{!Account.Name}"/>
      <apex:commandButton action="{!save}" value="Save"/>
    </apex:form>
  </body>
</apex:page>
```

## Como usar wire en LWC

```
import { LightningElement, wire } from 'lwc';
import { getRecord } from 'lightning/uiRecordApi';
import ACCOUNT_OBJECT from '@salesforce/schema/Account';
import NAME_FIELD from '@salesforce/schema/Account.Name';

export default class AccountRecord extends LightningElement {
  accountId;
  handleAccountIdChange(event) {
    this.accountId = event.target.value;
  }

  @wire(getRecord, { recordId: $accountId, fields: [NAME_FIELD] })
  account;
}
```

## Como evitar el SOQL Injection

- **SOQL Injection:** Ataque que permite que un atacante ejecute código en Salesforce
  - Usar bind variables

```
String name = 'Hello, World!';
List<Account> accounts = [SELECT Id, Name FROM Account WHERE Name = :name];
```

## Usar escapeSingleQuotes

- **escapeSingleQuotes:** Permite escapar las comillas simples
  - Usar bind variables

```
String s = '\Hello World\';
System.debug(s);
<!-- Result -> Hello World -->
String escaped = String.escapeSingleQuotes(s);
System.debug(escaped);
<!-- Result -> \Hello World\ -->
```

## Vulnerabilidades

### SOQL Injection

- **SOQL Injection:** Ataque que permite que un atacante ejecute código en Salesforce
  - Usar bind variables Cross-Site Scripting
- **XSS:** Ataque que permite que un atacante ejecute código JavaScript
  - Usar JSINHTMLENCODE()
  - Usar HTMLENCODE()
  - Usar JSENCODE() Cross-Site Request Forgery
- **CSRF:** Ataque que permite que un atacante realice acciones en nombre de un usuario
  - Usar CSRF Token

## Content Sniffing

### Lightning Data Tables

```
import { updateRecord } from 'lightning/uiRecordApi';

handleSave(event) {
  const fields = {};
  updateRecord(recordInput)
    .then(() => {
      fields['id'] = event.detail.draftValues[0].Id;
      fields['Name'] = event.detail.draftValues[0].Name;
      fields['Industry'] = event.detail.draftValues[0].Industry;
    });
})
}
```

## WITH SECURITY\_ENFORCED

- Permite que se apliquen las reglas de seguridad de Salesforce

```
@AuraEnabled(cacheable=true)
public static List<Account> getAccounts() {
  return [SELECT Id, Name FROM Account WITH SECURITY_ENFORCED];
}
```

## How to use third-party javascript libraries in Aura Components

Use Static Resources to store the library then use ltng:require to load the library

## Aura Components

Permite aprender los conceptos de los componentes de Aura

Orden de firing

- Init
  - Empieza desde el componente innermost y va hacia afuera.

## Eventos

- Application Event
- Component Event
- Bubble Event
  - Configuración de Bubble Event
  - Bubble Event: true
  - Composed: false
    - De esta manera no se propaga a través de Shadow DOM
- Capture Event
- System Event

```
/**
 * @description CSS file
 */
@import "c/style.css";
```

```
<aura:component implements="flexipage:availableForAllPageTypes" access="global">
  <aura:attribute name="message" type="String" default="Hello, World!" />
  <lightning:input type="text" label="Message" value="{!v.message}" />
</aura:component>
```

```
<design:component>
  <design:attribute name="message" label="Message" description="The message to
display" />
</design:component>
```

## Bundles

Quizás aquí hacer una tabla con los tipos de bundles

- **Bundle:** Permite agrupar varios elementos
  - Component
    - Component
    - Controller
    - Design
    - Documentation

- `Renderer`
- `Helper`
- `SVG`
- `JS`
- `Application`
  - `Application`
  - `Controller`
  - `Design`
  - `Documentation`
  - `Renderer`
  - `Helper`
  - `SVG`
- `Lightning Com`

CSRF

- **CSRF:** Cross-Site Request Forgery
  - Que es: Ataque que permite que un atacante realice acciones en nombre de un usuario

Como evitarlo en Salesforce

- **CSRF Token:** Permite evitar que se realicen ataques CSRF
  - mas info [CSRF Token](#)

Visualforce

`Security.stripInaccessible` como usar

- **`Security.stripInaccessible`:** Permite quitar los campos a los que el usuario no tiene acceso
  - mas info [Security.stripInaccessible](#)
    - Devuelve tipo `SObjectAccessDecision`
      - Recuperar las tuplas -> `getRecords()`

todo HERE

- Custom Property Editors
- 

Anexo

Limites

Informacion sobre los límites de Salesforce en la documentación oficial [Salesforce Governor Limits](#)

Límite	Valor
<code>queryLocator</code>	10,000
<code>queryMore</code>	2,000
<code>@future</code>	50



Límite	Valor
@future (con callout)	10
@future (con callout y batch)	5

The Salesforce Administrator at Cosmic Mobile has configured the Next Best Action component to display recommendations to its customer service representatives (CSRs) regarding phone plans. When a CSR accepts a recommendation for a customer who wants to upgrade to a certain plan, a flow is launched to process the upgrade. If the offer is rejected, the flow needs to perform a different action. How can a Salesforce Administrator meet this requirement?

- a Build a single flow to perform an action that is used when a recommendation is accepted or rejected
- b Enable 'Launch Flow on Rejection' for the Next Best Action component in Lightning App Builder

A recommendation can only invoke one and the same flow when it is either accepted or rejected by the user. By default, a recommendation only launches the flow when it is accepted. To also launch the flow when it is rejected, a Decision element is added to the flow that specifically uses an "isRecommendationAccepted" boolean variable for determining whether the recommendation that launched it was approved or rejected, and then perform the necessary logic in the flow accordingly. Also, the 'Launch Flow on Rejection' settings for the Next Best Action component must be enabled in Lightning App Builder.

\$contentAssetGlobal puede contener imagenes, archivos de texto, videos, etc.

- JavaScript
- CSS
- Images

Donde podemos usar Visualforce

- Custom App
- Navigation Bar
- Standard Page Layout