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)
1.1	29/04/2024	Adrián Arribas	Organizado y añadido cosas de Examenes FoF
1.2	16/09/2024	Adrián Arribas	Añadida la guía de estudio.
1.2	23/09/2024	Adrián Arribas	Terminado Dev Fundamentals Topic I.

Status estudio

Tema	Porcentaje
User Interface	53%
Developer Fundamentals	64.29%
Testing, Debugging, and Deployment	76.92%
Process Automation and Logic	88.89%

To-do

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.

Por una parte los topics son resumenes o transcripciones de la guia de estudio de focus on force y mas adelante hay respuestas concretas a las preguntas test ademas de un banco de preguntas que he hecho yo.

Concepto	Descripción
Puntuación	68%
Tiempo	105 minutos
Preguntas	60

Syllabus	Weightage
Developer Fundamentals	23%
Process Automation and Logic	30%

Syllabus	Weightage
User Interface	25%
Testing, Debugging, and Deployment	22%

Indice

Developer Fundamentals

Study Guide

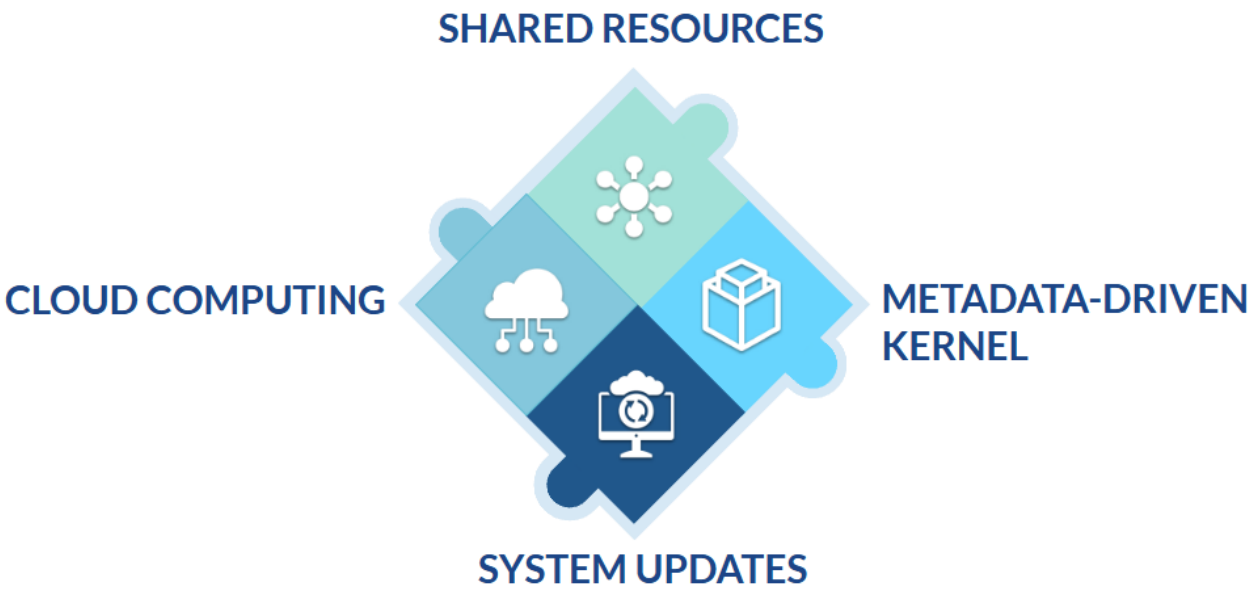
Introduction

Salesforce is built upon a multi-tenant architecture and consists of features and characteristics such as cloud computing, shared system resources,automatic system updates, and metadata-driven kernel. In effect, development in this environment impose different considerations compared to traditional development.

Salesforce generally follows the Model-View-Controller (MVC) architecture where each layer presents specific aspects pertaining to application development.

Solutions can be developed in the platform using several built-in declarative tools. For more complex requirements, programmatic tools are available such as Apex and Visualforce. The Lightning Component framework, which is a UI framework, can be used for building custom Lightning components to deliver responsive and efficient event-driven applications.

Salesforce Multitenant Architecture consist of the following key elements and features:



Cloud Computing

Lighthning platforms is a PaaS that is bult for cloud computing and based on the multitenant architecture.

1. Web Based Platform

Resources are accesed via Internet using a browser

2. Focused Development

Underlying IT mechanism do not need to b

3. Developer Console

An Integrated DevelopmentEnvironment (IDE) may be used

4. No software installation

No client software is required for access and development

Shared Resources

1. All Customers share the same computing power, custom-designed database, data storage and core features.
2. Salesforce monitors code execution and has various governor and resource limits associated with code execution
3. Resources such as CPU Usage, queries and records returned are limited per customer to ensure optimal performance.

System Updates

- System Updates

Automatic Seamless Upgrades are rolled out three times a Year Spring,Summer and Winter

- Shared Environment

All customers automatically get the same updates at the same time throughout the year

- Release Preview

Sandboxes are upgraded before production so that the changes can be previewed and tested.

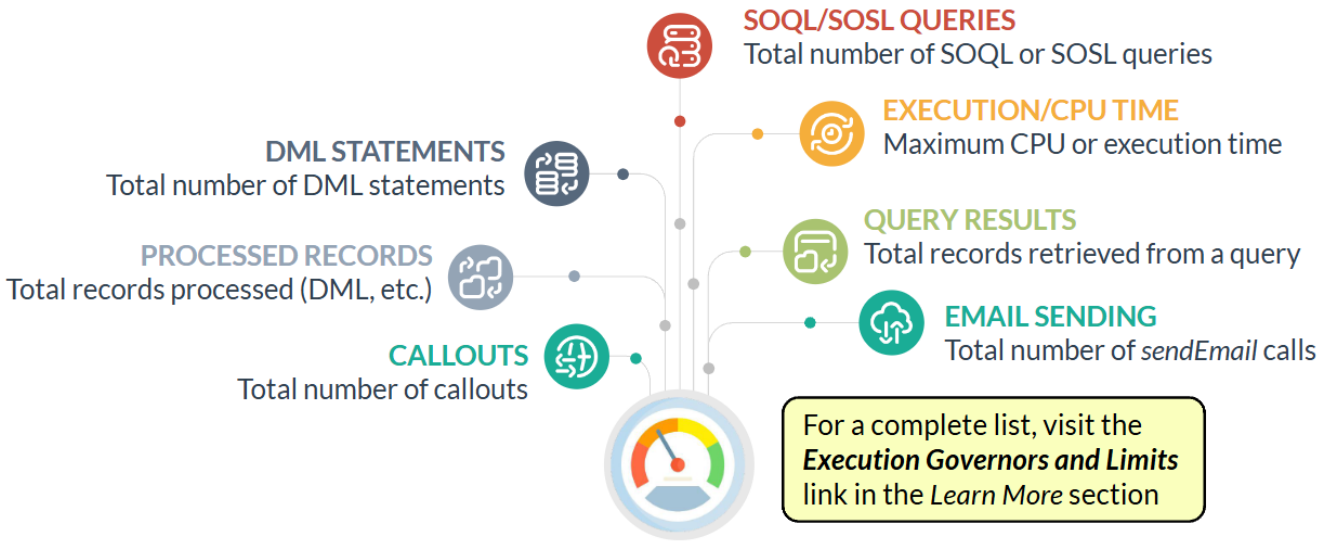
Metadata Driven Kernel

Arch: There is a clear separation between the runtime database engine(kernel), tenant data and metadata, which allows easy upgrades.

Deployment: Deployment of new or updated metadata components to production is strictly managed. Unit tests must cover 75% of the application's source code for deployment to production.

Governor Limits

Governor limits ensures tenants do not monopolize shared resources. If exceeded, these limits throw exceptions that cannot be handled and should always be taken in account during developing.



MVC Architecture

In application development, the MVC is an architectural pattern that separates the data layer from thebusiness logic and how the data is presented in the user interface.

❖CONTROLLER represents the business logic, either declarative or programmatic. Custom controllers and controller extensions are written as Apex classes.

It includes the following Components:

- Standard Objects
- Custom Objects
- Object Fields
- Object Relationships

- Apex Classes (Data)

❖VIEW represents the presentation layer which consists of pages and components.

It includes the following Components:

- Standard Pages
- Visualforce Pages
- Visualforce Components
- Custom Tabs
- Page Layouts

❖MODEL represents the structure of the data through sObjects, fields, and Apex classes.

It includes the following Components:

- Standard Controllers
- Custom Controllers (Apex)
- Extensions (Apex)
- Declarative Rules & Tools (Apex Triggers, Validation Rules, Flows, etc.).

Lightning Component Framework

A UI framework that allows building SPAs with dynamic and responsive User Interfaces in Salesforce.

- JavaScript on client side
- Apex in server side

There are 2 programming language models: Aura components and LWC. They both can coexist and interoperate on a page.

Benefits:

- Device awareness
- Cross Browser Compatibility
- Out of the box components
- Customizing Lightning Experience

About Performance it utilizes Stateful Client using JS and stateless server (Apex) The client call the server only when absolutely necessary, which results in fewer calls to the server and more responsive and efficient apps.

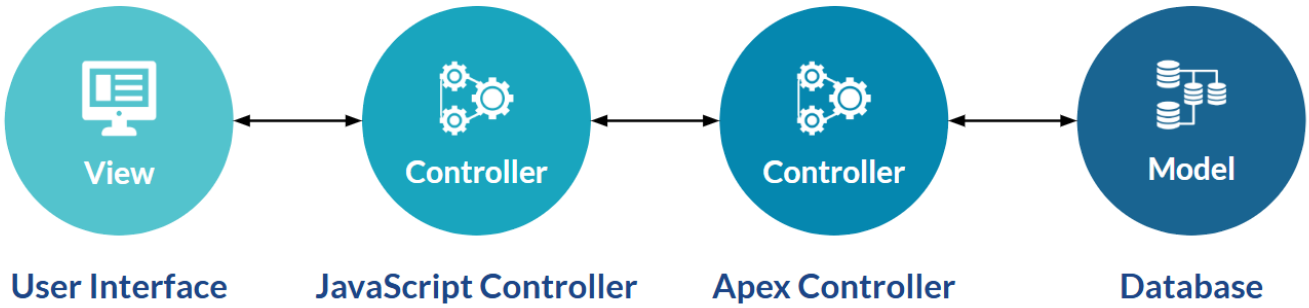
Lightning components utilize Event-driven arch. Components are capable of listening to events and responding accordingly.

The USAGE may vary for different Contexts. For example they can be added as custom tabs in lightning Experience and the Salesforce Mobile App.

The Lightning Component Framework is not strictly based on the MVC architecture. Aura and Lightning web components follow the MVCC (Model-View-Controller-Controller) pattern.

- MODEL represents the database (sObjects,fields, and Apex classes).

- VIEW represents the Lightning component. Ithas the .cmp suffix and contains markup. The markup contains text and/or references to other components.
- CONTROLLER (SERVER-SIDE) The Apex controller on the server side is used to perform database operations. The JavaScript controller is used to call methods in the Apex controller.
- CONTROLLER (CLIENT-SIDE) The JavaScript controller on the client side is used to perform client-side operations and also acts as the intermediary between the server and the UI.



App Manager Page 24

An app is a group of tabs that work as a unit. In Salesforce both Lightning and Classic apps can be created and customized to serve specific functions in Salesforce.

- App Contents

Classic Apps contain standard and custom tabs, which can include standard and custom object tabs, Visualforce tabs, Lightning component tabs, Canvas apps and web apps. On the other hand, Lightning apps contain everything from the Classic apps list, plus they can contain Lightning Page tabs and utilities like Sales Dialer.

- Creating Apps

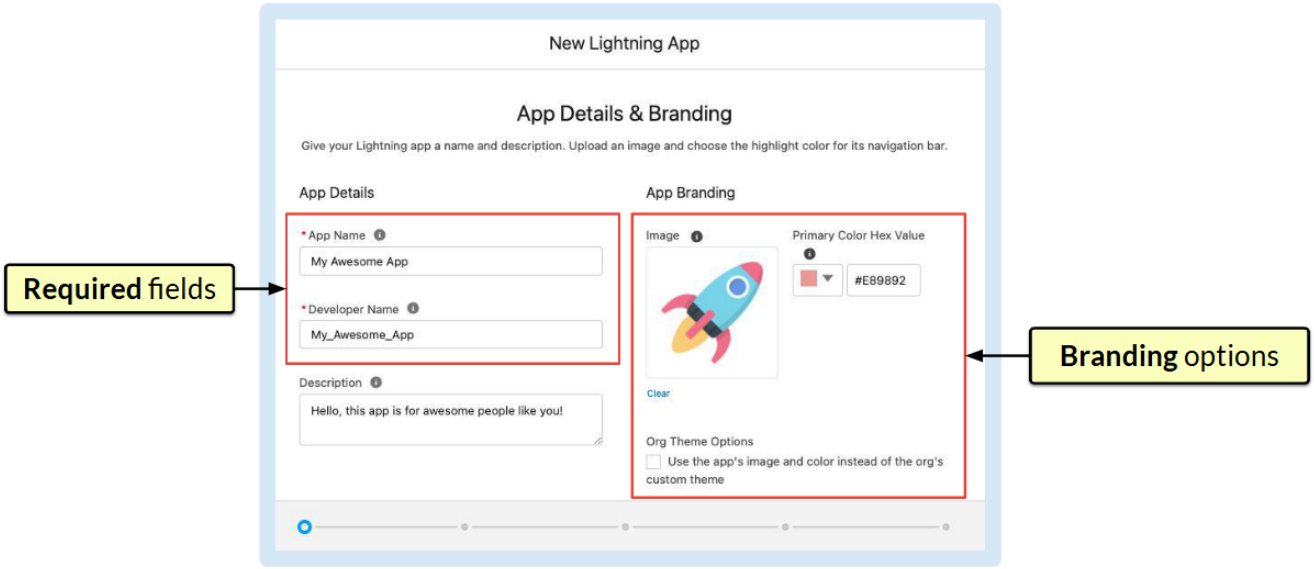
A Lightning App can be created by navigating to App Manager in Setup. Once an app is created, it can be extended by adding objects and fields, using automation, defining Access Settings, adding Users, etc... Lightning App Builder can be used to update the app branding, navigation, and other options, and manage the Lightning pages assigned to the app.

- New Lightning App Wizard

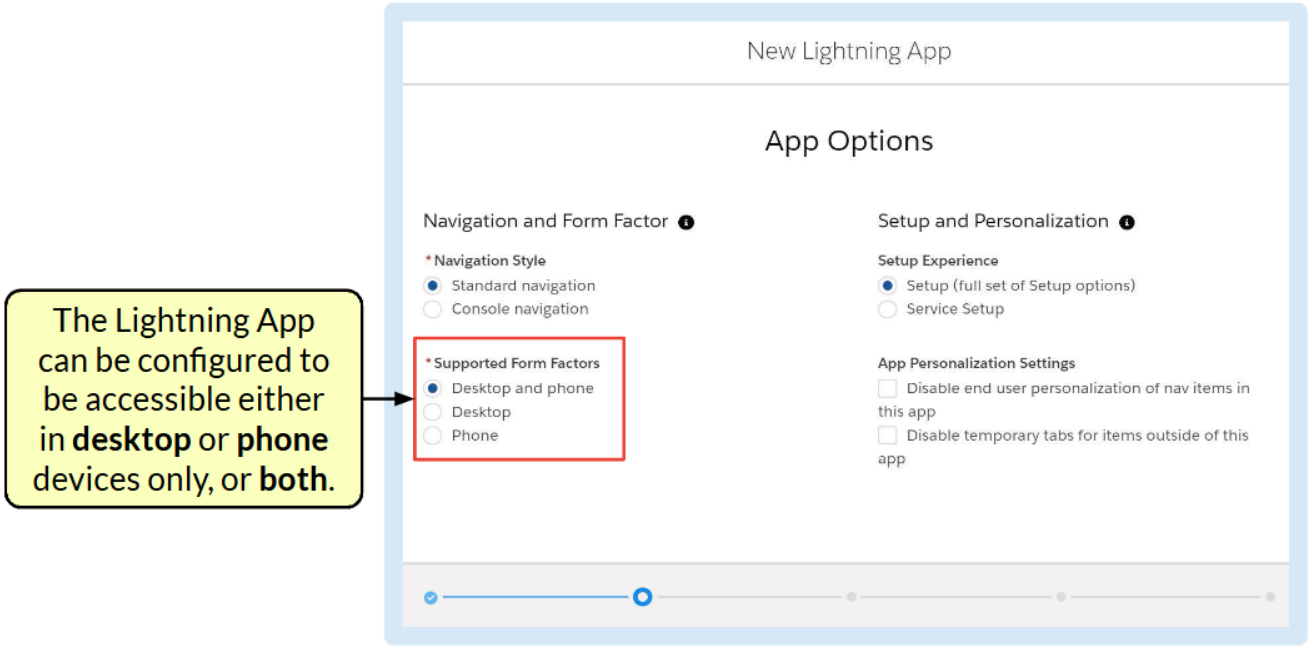
When creating a new Lightning app using the App Manager, the app can be given a name, its primary color and logo can be set, app description can be added, and the type of navigation (standard or console) can be set. Form factors available for the app can be selected. The items that appear in the navigation bar can be selected. Utility items, like Recent Items, Notes, Dialer, To do list, and Open CTI, can be added to the utility bar. The app can be assigned to user profiles.

Creating an App

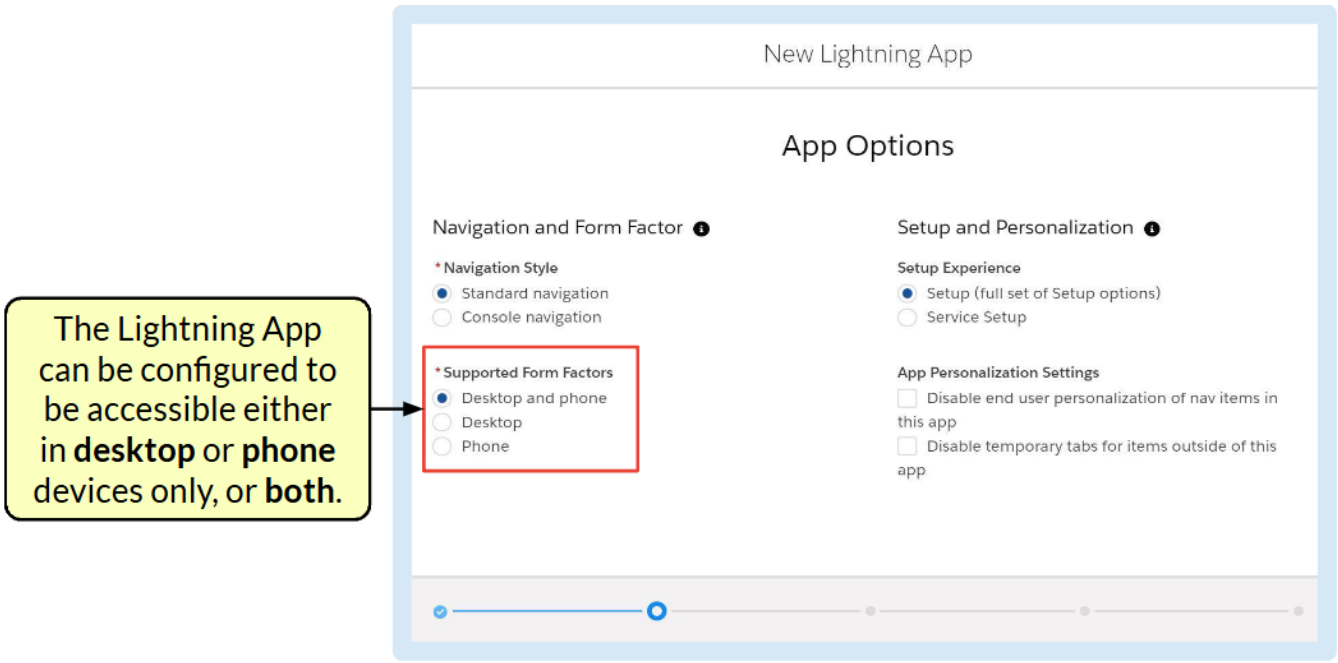
When creating a new Lightning App, the App Name and Developer Name are required. Other details such as its description and branding can also be specified.



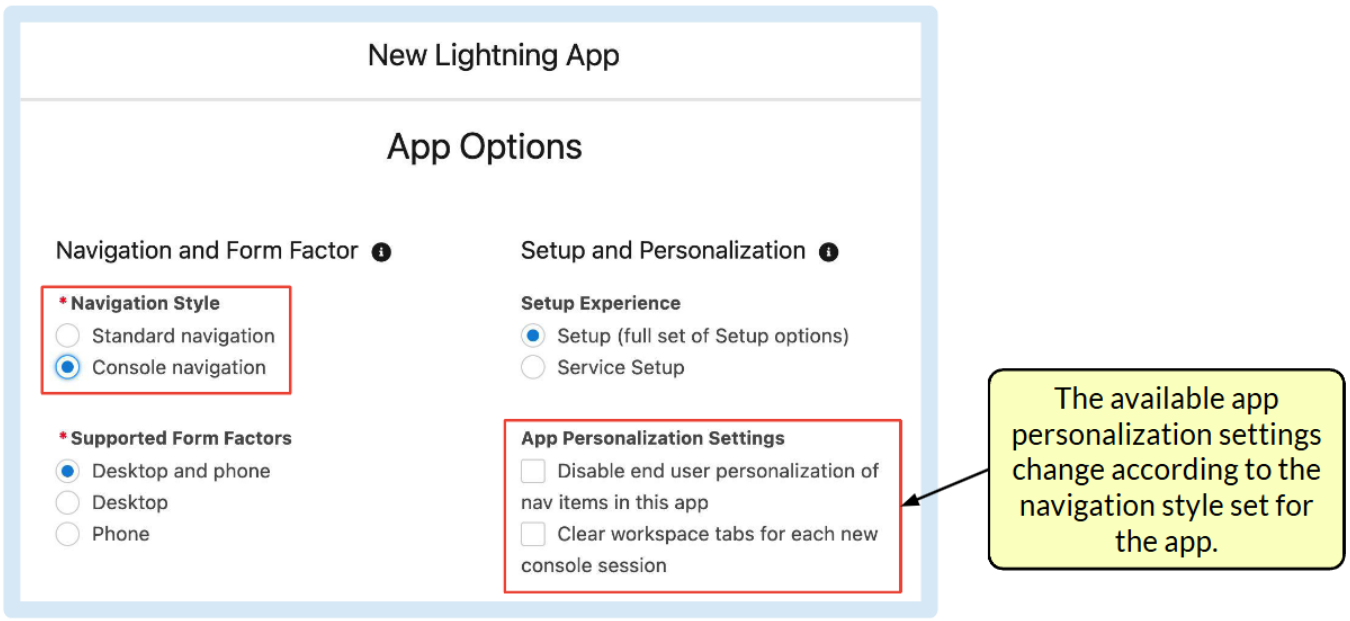
In the App Options configuration page, the navigation style and form factors that the app will support are specified. Other options are related to setup and personalization settings.



In the App Options configuration page, the navigation styleand form factorsthat the app will support are specified.

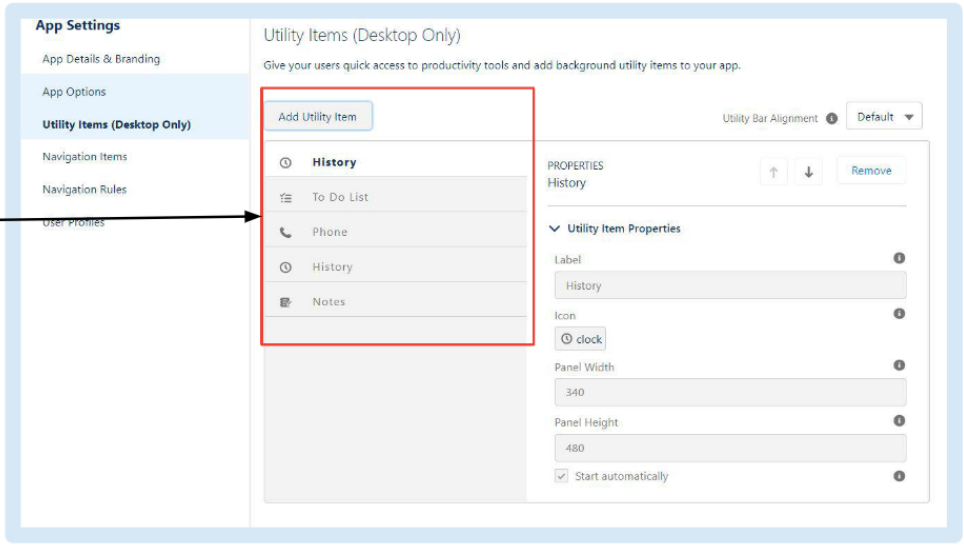


The navigation style determines the personalization settings available for the App

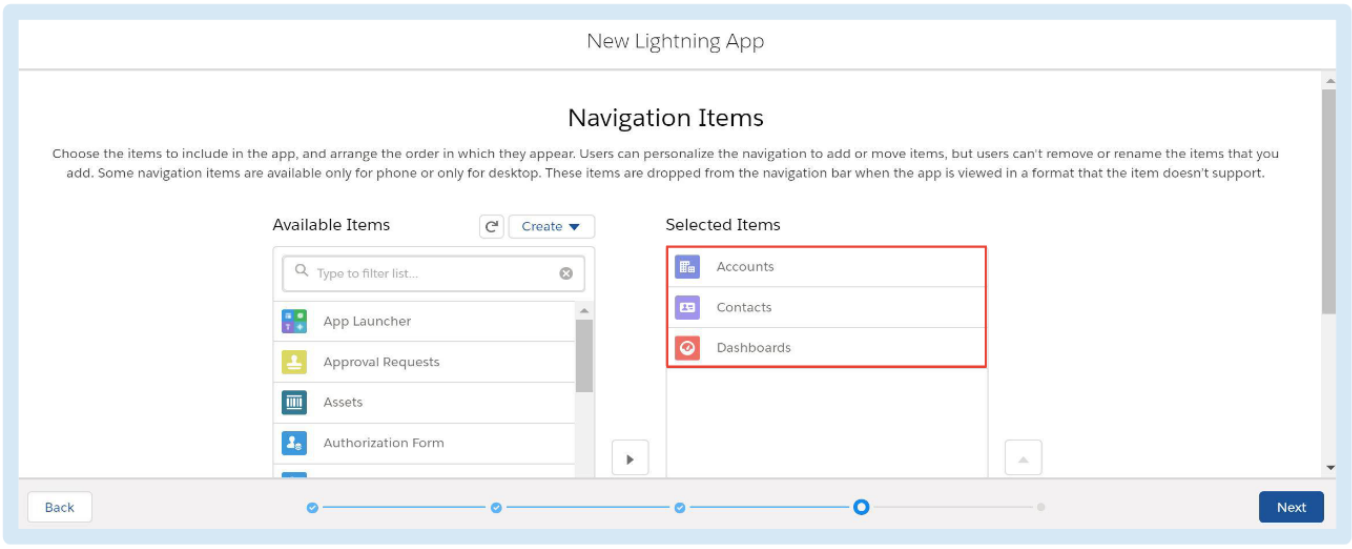


Utility Items can be added to a utility bar, which is a feature that is only available to desktop devices.

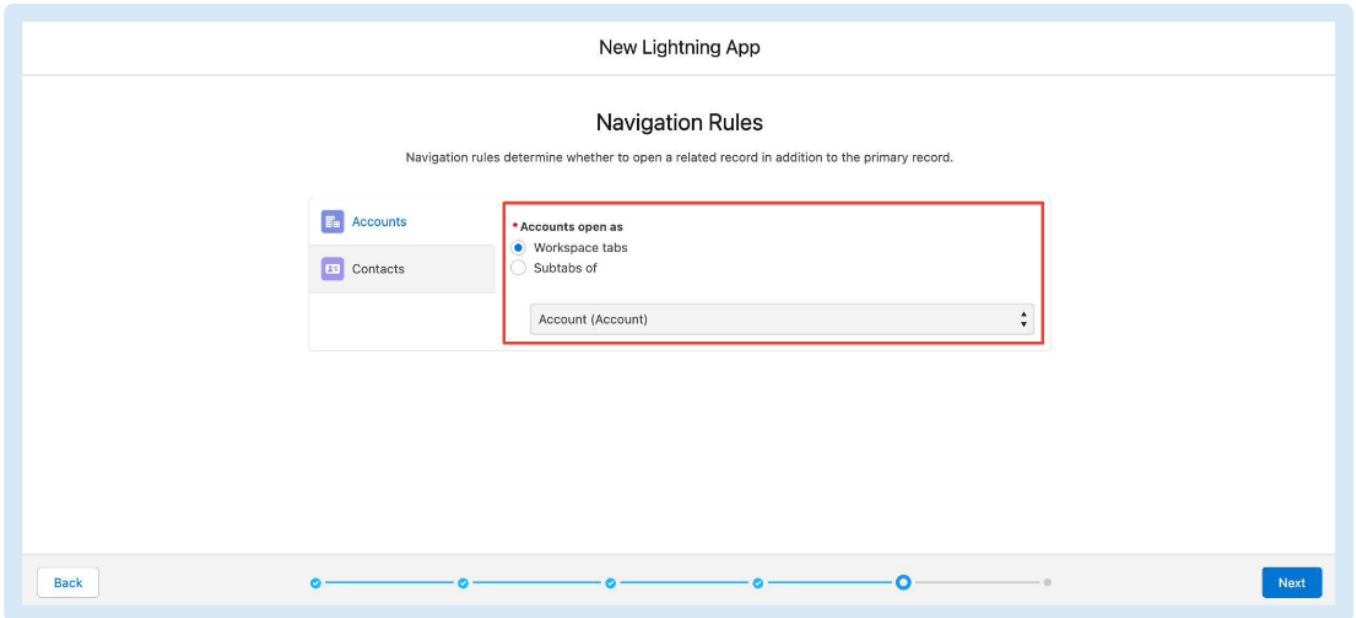
To Do List, History, Notes, and several other utility items can be added. It is important to note that the availability of the utility items depends on the product and the edition that a company uses.



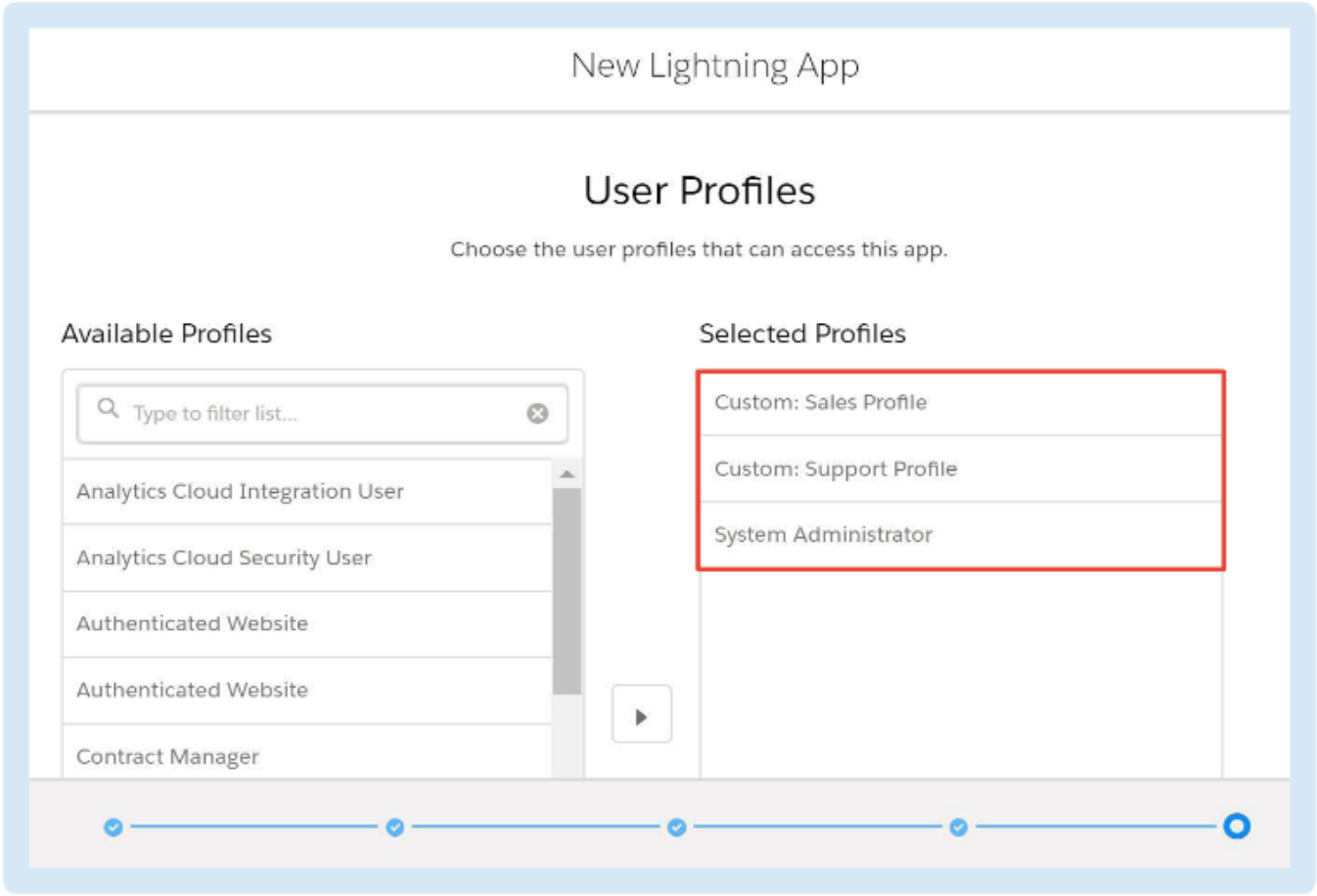
The items that are added to the Selected Itemscolumn will appear in the navigation bar of the Lightning App.



The Navigation Rules option, which is only available when using the Console navigation style, determines whether records are opened as a workspace tab or subtab of another record.



Profiles are assigned to the Lightning App by adding them to the Selected Profilescolumn. Only users assigned to these profiles will be able to access the Lightning App.



❖NAVIGATION BAR

The item at the top of the list in the navigation bar becomes the app's landing page on both desktop and mobile. The order of the items determines the default objects shown in the Top Results page on the search results page.

LIGHTNING APP BUILDER

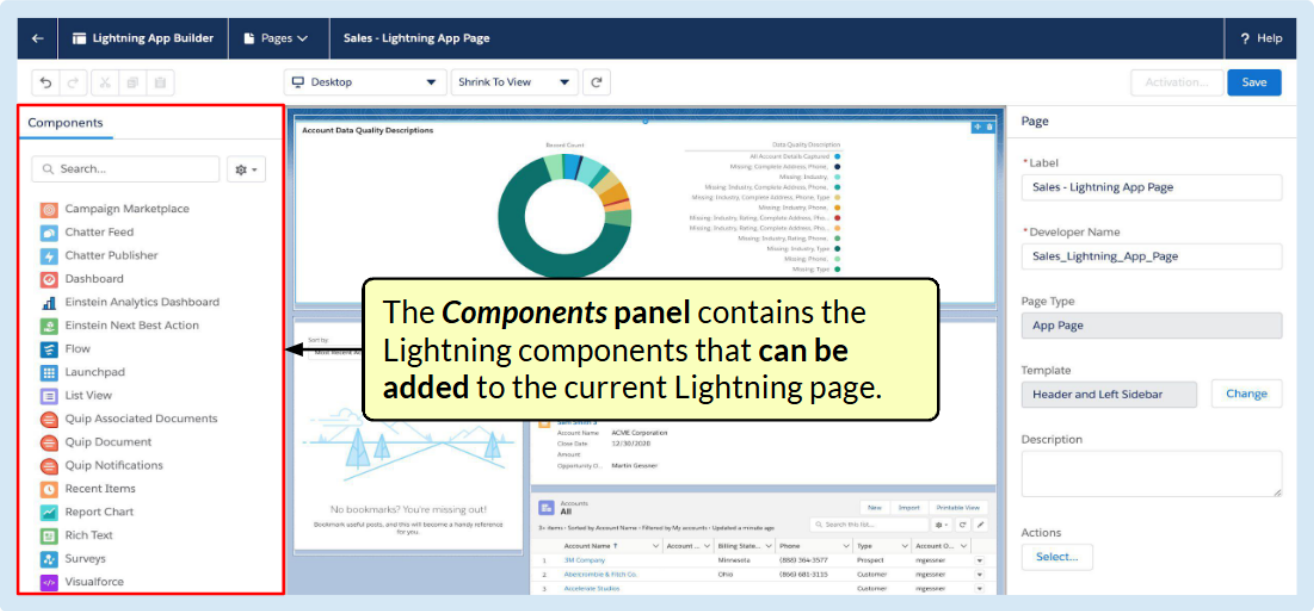
Lightning App Builder is a declarative tool that can be utilized to create and configure custom pages for Lightning Experience and the Salesforce mobile app.

- CAPABILITIES Lightning App Builder allows building and customizing single-page apps, dashboard-style apps, 'point' apps that solve a particular task, customrecord pages, custom Home pages, and custom email application panes.
- PAGE TYPES The following Lightning page types are available: apppage, homepage, recordpage, emailapplicationpane, forecasts page, and Omni Supervisorpage. Note that some of these page types require additional configuration in order to appear as an option in Lightning App Builder. After specifying the page type to use, a template is selected that determines the layout and number of regions in the Lightning page.
- CREATING A LIGHTNING PAGE To define the content of a Lightning page, standard, customand third-party Lightning components from the AppExchange can be added to the Lightning page.
- STANDARD LIGHTNING COMPONENTS everal standard Lightning components built by Salesforce are available for creating Lightning pages. Some of them have required properties that must be configured.

Examples include Accordion, Chatter, RecordDetail, RelatedRecord, etc.

- CUSTOM LIGHTNING COMPONENTSCustom Lightning components can be built and added to a Lightning page. However, to make a custom Lightning component usable in the Lightning App Builder and Lightning pages, the component and its component bundlemust be configured so that they are compatible. My Domain must also be deployed in the org.
- DYNAMIC LIGHTNING PAGESIt is possible to configure when a component appears on a Lightning page by adding filter conditions and logic to its properties in the Lightning App Builder. For example, a Lightning component can be set to display exclusively when its page is viewed on a phone.

Lightning App Builder can be used to build Lightning pages for using Standard, Custom and Third-party Lightning components.



Declarative Tools

Salesforce provides various declarative tools that can be used to build and extend apps.

AUTOMATION

Included tools for automating business processes:

- Flow Builder
- Process Builder
- Approval Process
- Workflow Rule

AUTOMATION

Declarative automation tools can implement simple or complex processes.

PROCESS BUILDER

Process Builder performs immediate or time-based actionswhen a record is created, a record is updated, or a platformevent occurs. It is capable of performing actions conditionallybased on a single or multiple if/then statements.

Process Builder supports the following operations:

1. Creating new records
2. Updating any related record
3. Invoking Apex code
4. Creating a Chatter post
5. Sending an email alert
6. Submitting a record for approval
7. Launching a flow or another process
8. Sending a custom notification

FLOW BUILDER

Flow Builder can be used to automate a guided visual experience or start a business process from a User Interaction, process, schedule, platform Event or record changes. Flows can also be invoked trough Apex and vice-versa.

Flows can be used, for example, in the following:

1. Creating new records
2. Deleting records
3. Launching another flow
4. Updating any record
5. Sending an email alert
6. Sending a custom notification
7. Invoking Apex code
8. Submitting a record for approval
9. Creating a Chatter post
10. Sending an outbound message

APPROVAL PROCESS

An **approval process** can be used to submit a record for approval when a user clicks a button or link or when a flow or process submits a record for approval.

1. Creating new tasks
2. Sending an email alert
3. Updating the original record or its parent
4. Sending an outbound message

WORKFLOW RULE

A **Workflow rule** can be used to implement a business process that consists of a single if/then statement. It can be triggered when a record is created or updated, and supports both immediate and time-based actions.

1. Creating new tasks
2. Sending an email alert
3. Updating the original record or its parent
4. Sending an outbound message

TABLA COMPARATIVA

Acción	Process Builder	Flow Builder	Approval Process	Workflow Rule
Crear nuevos registros	✓	✓	✗	✗
Eliminar registros	✗	✓	✗	✗
Actualizar registros	✓ (relacionados)	✓ (cualquier)	✓ (original o padre)	✓ (original o padre)
Invocar código Apex	✓	✓	✗	✗
Crear publicación en Chatter	✓	✓	✗	✗
Enviar alerta de correo	✓	✓	✓	✓
Enviar una notificación personalizada	✓	✓	✗	✗
Enviar mensaje saliente	✗	✓	✓	✓
Lanzar flujo u otro proceso	✓	✓	✗	✗
Enviar registro para aprobación	✓	✓	✗ (inicia proceso)	✗
Crear nuevas tareas	✗	✗	✓	✓

SHARING AND SECURITY

Features such as profiles, permission sets and org-wide defaults can be used to control sharing and security.

Sharing and security features can be utilized to expose different data sets to different sets of users in Salesforce.

❖PROFILES & PERMISSION SETS Profiles and permission sets can be used to specify the objects and fieldsthat users can access as well as user permissions that control what tasks users can perform and what features they can access.

❖RECORD ACCESS Access to records can be controlled by specifying organization-wide default sharing settings. Additional access to records can be provided using features such as sharing rules, role hierarchy, manual sharing, and teams.

BUSINESS LOGIC AND VALIDATION

Formula fields and **Roll-up summary fields** can be used to calculate field values. **Validation rules** can be used to enforce data quality

Declarative features can be used to automate the calculation and validation of field values.

FORMULA FIELDS

A formula field can be used to automatically calculate the value of a field based on other fields, expressions, or values. A data type can be selected while building a formula field. The formula specified in the field can

contain references to the values of fields, operators, functions, literal values, or other formulas.

ROLL-UP SUMMARY FIELDS

A roll-up summary field is used to automatically calculate the value of a field on a parent record based on the values of a field on child records. For example, the sum of the values of a custom field named 'Amount__c' on child records can be displayed on the parent record.

VALIDATION RULES

A validation rule can be used to verify that the data entered by a user on a record meets the standards before the record is saved.

USER INTERFACE

There are various tools available to create UI such as **App Manager**, **Lightning App Builder** and **Quick Actions**.

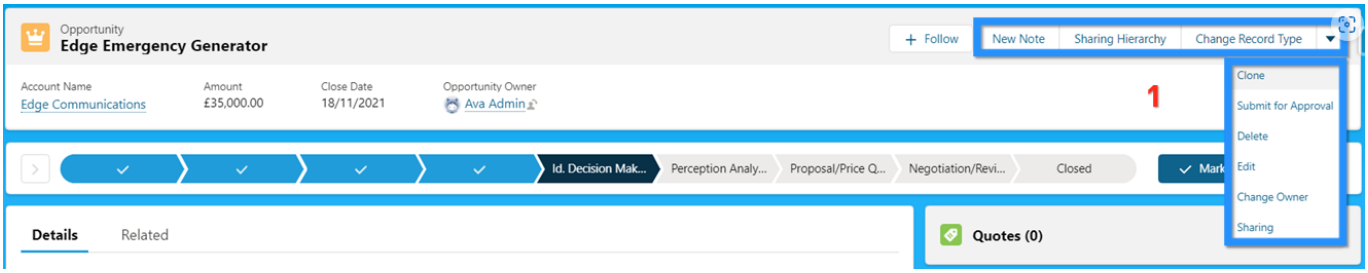
APP MANAGER

App Manager can be used to create an app which acts as the container for a set of items such as object tabs and utility items.

LIGHTNING APP BUILDER

Lightning App Builder can be used for building and customizing single-page apps, dashboard-style apps, 'point' apps that solve a particular task, custom record pages, custom Home pages, and custom email application panes.

QUICK ACTIONS Object-specific and global quick actions can be created to allow users to invoke custom functionality when using the UI. For example, they can use a quick action to quickly send an email to a customer.



DATA MANAGEMENT

Data Import Wizard and **Data Loader** can be used to insert and update multiple records. Data Loader can also be used to export and delete records.

DATA IMPORT WIZARD

Data Import Wizard supports import up to 50,000 records. Accounts, contacts, leads, solutions, campaign members, and records of custom objects can be imported.

DATA LOADER

Data Loader supports import or export up to 150,000,000 records. One can insert, update, upsert, delete, or export Salesforce records.

Característica	Data Import Wizard	Data Loader
Límite de registros	Hasta 50,000	Hasta 150,000,000
Tipos de registros soportados	Cuentas, contactos, leads, soluciones, miembros de campañas, objetos personalizados	Todos los objetos de Salesforce
Operaciones soportadas	Insertar	Insertar, actualizar, upsert, eliminar, exportar
Exportar registros	✗	✓
Soporte de objetos personalizados	✓	✓
Interfaz de usuario	Basada en navegador	Aplicación de escritorio
Acceso a través de API	✗	✓

Data Loader is generally more complete.

ANALYTICS

Reports and **Dashboards** can be built to help users analyze the data. They also can be added to folder for better management.

REPORTS

Report Builder is a visual, drag-and-drop tool that can be used to build custom reports and edit existing reports. It allows choosing a report type, report format, and the fields that should make up the report.

DASHBOARDS

Dashboard Editor is a visual, drag-and-drop tool that can be used to build custom dashboards and edit existing ones. Dashboard components can be added, edited, or arranged in a dashboard. A dashboard component contains a chart or metricthat shows data from a source report.

REPORT AND DASHBOARD FOLDERS

Report and dashboard folders can be used to organize and share reports and dashboards.

Programatic Tools

Apex

The programmatic features offered by Salesforce include Apex, Visualforce, and Lightning Components which are used when a required functionality cannot be achieved using declarative tools.

Apex Code can be used to build more complex business logic for Apps. An invocable Apex method can be created and invoked as an Apex Action in a process or a Flow. **Apex Triggers** can be created to execute custom business logic before or after a record is:

- Created
- Updated
- Deleted
- Merged
- Upserted
- Undeleted (after event only)

Apex can be used for integration with external systems. It can also be used to perform server-side actions for custom Visualforce pages and Lightning Components.

Visualforce

Visualforce is a framework that can be used to build custom UI, mainly for SF Classic. It uses tag-based markup language that consists of Visualforce tags, HTML, JS or any other Web-enable code embedded within a single apex:page tag. Visualforce pages can be used to override standard buttons, define custom tabs, create dashboard components, etc...

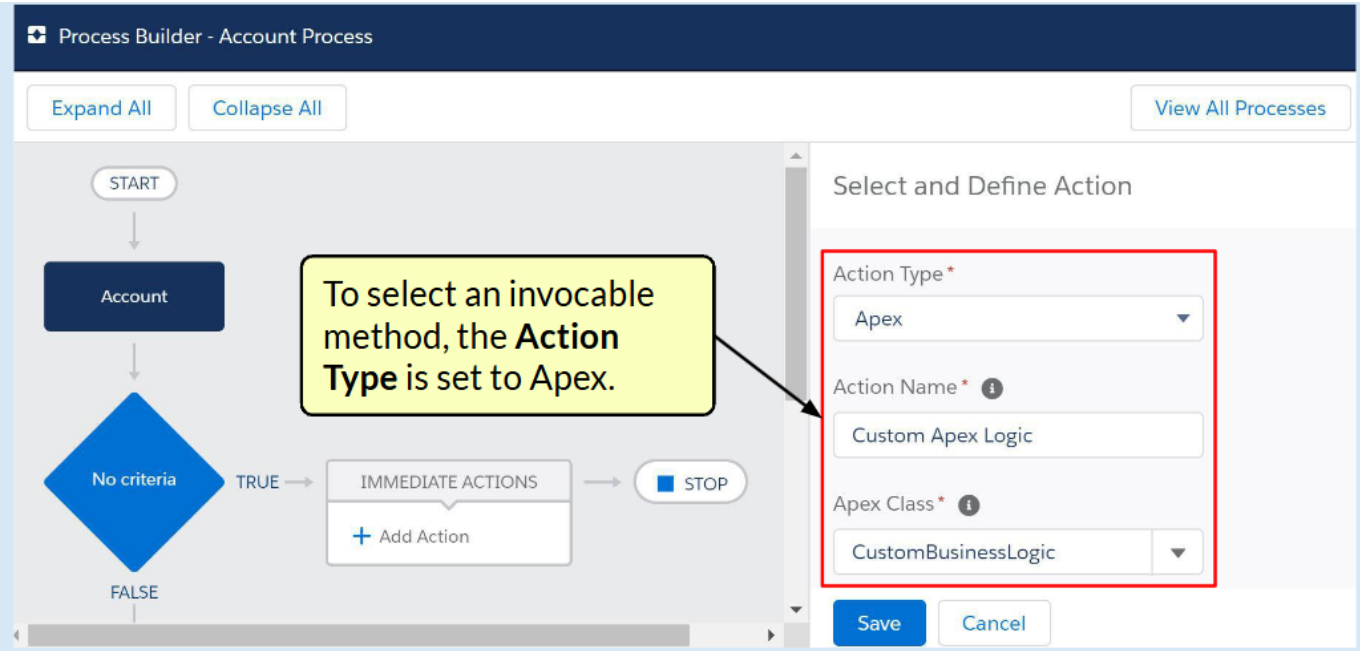
Using apex

// An Apex class is allowed to have one (1) invocable method

```
public classCustomBusinessLogic {
  @InvocableMethod
  public static void processAccounts() {
    // perform logic or process which a declarative tool is not capable of
achieving...
    }
  }
```

The following Apex class contains an invocable method which can beinvoked from declarative toolssuch as Flow Builder or Process Builder.

If the label attribute of the Apex invocable method is not defined, the Apex class nameis specified in the Apex action type



The programmatic features offered by Salesforce include Apex, Visualforce and Lightning Components which are used when a required functionality cannot be achieved using declarative tools

Lightning Aura Components.

Custom Lightning Aura Components can be created by using the original Aura Components model. Aura Components are self-contained and reusable units of an App that represent reusable sections of the UI and can range in granularity from a single line to an entire App. Various base components are available to build Aura Comps.

LIGHTNING WEB COMPONENTS Lightning components can also be created using the Lightning Web Components (LWC) model, which uses the core Web Componentsstandards. HTML and modern JavaScript can be used to build Lightning web components.

USING LIGHTNING COMPONENTSB oth Aura and Lightning web components can be used in places like custom tabs, Lightning pages, Experience Builder, Flows, Visualforce pages, etc.

APIs

Salesforce also offers a wide range of useful APIs that can be used in different scenarios

- REST API provides a way to programmatically access or process Salesforce data from an external application via REST-based web services. Use this API when developing web or mobile applications that need to integrate with Salesforce.
- SOAP API Similarly, SOAP API lets developers access, search, and perform DML operations on records from an external application via the SOAP messaging protocol. Use SOAP API in any languages that can work with web services such as Java or C++.
- CHATTER REST API Chatter REST API can be utilized to show Chatter related data such as feed, users, or groups. Use this API when integrating Chatter with external web or mobile applications.
- USER INTERFACE API User Interface API allows developers to build user interfaces that surface records, list views, quick actions, favorites, and other components. Use this API to build Salesforce UI on external native mobile applications and web applications.

- ANALYTICS REST API Analytics REST API grants programmatic access to Salesforce Analytics data such as datasets, lenses, and dashboards. Use this API when there is a need to access, query, and interact with the Analytics Platform programmatically.
- Bulk API lets developers and admins perform CRUD (Create, Read, Update, Delete) actions on large datasets asynchronously in batches. Use this API when working with large volumes of data (hundreds of thousands to millions of records).
- METADATA API Metadata API allows developers to programmatically retrieve, deploy, create, update, or delete customizations or metadata in an org. The most common use of this API is in deploying changes from one org to another, or from a local directory to an org using Salesforce CLI.
- STREAMING API Streaming API is used in receiving notifications based on record-change events or custom events in Salesforce. Use this API to subscribe and receive real-time notifications related to changes within a Salesforce org.
- APEX REST API Apex REST API allows Apex classes to be exposed and invoked by external application as REST web services. Use this API when there is a need to run Apex code from an external application and increase code reuse.
- APEX SOAP APISimilar to Apex REST API, Apex SOAP API allows Apex classes to be exposed and invoked by external application as SOAP web services. Use this API when it is required to run Apex code from an external application via SOAP API.
- TOOLING APITooling API provides the ability to write SOQL query for many metadata types in an org. This API provides SOAP and REST interfaces and can be used in creating custom Lightning Platform development tools or apps.

Identify Scenarios - Programatic vs Declarative

Identify Scenarios - Determine, create and access appropiate data model

Including object, fields, relationships and External IDs

Identify Scenarios - Options and Considerations on Importing and Exporting Data

Process Automation and Logic

User Interface

Testing, Debugging and Deployment