

Welcome to

dreamOle[®]



Cognizant



FORMTITAN



MAKEMECLLOUD
consulting



aircall

blue-
infinity

Linked by isobar

Dentsu Aegis Network



FINANCIALFORCE



wefox

BLACKBIRD



BUILDING A MULTICLOUD APP WITH LIGHTNING COMPONENTS

José María González Sanz



Sobre mí

Empecé a trabajar con Salesforce en 2016

Actualmente trabajo en Omega CRM



OMEGA EN CIFRA 2019

+15 AÑOS
EXPERIENCIA



EMPRESA
FINANCIERAMENTE
INDEPENDIENTE



SOCIOS
IMPLICADOS EN
EL PROYECTO



106
EQUIPO
HUMANO

**MULTI-PLATFORM CRM
ENTERPRISE**

PARTNER DE LOS
PRINCIPALES
FABRICANTES CRM
DEL MERCADO



5

SEDES



CENTROS DE SOPORTE GLOBAL
Y DE IMPLEMENTACIÓN LOCAL, NEARSHORE Y
OFFSHORE



160
CLIENTES
+60
ACTIVOS



9.79

SATISFACCIÓN
DE CLIENTES

omega^{crm}

**+15 años de experiencia
implantando
soluciones CRM.**

*“Animamos a nuestros clientes a que adopten y utilicen la tecnología. Les ayudamos a entenderla ofreciéndoles **una experiencia única adaptada a sus necesidades** permitiendo la mejora su negocio.”*

Survey Multicloud App

Funcionalidades y Requerimientos



Crear / Editar

 **DreamOle Questionnaire**

Survey URL : <https://omegacrmsurvey-dev-ed.lightning.force.com/c/Survey.app?surveyId=a021r0000de1mxAAA>

¿Sueñan los robots con ovejas eléctricas?

Delete Save

Salesforce Sans 12 B I U ↺ ☰ ☷ ☹ ☺ ☻ ☼ ☽ ☿

Question subtype
One option

Si



Distribuir / Responder

Survey URL : <https://omegacrmsurvey-dev-ed.lightning.force.com/c/Survey.app?surveyId=a021r00000de1mxAAA>



DreamOle Questionnaire

1. ¿Sueñan los robots con ovejas eléctricas?

Si

No

Tal vez



DreamOle Exam

1. Selecciona las frutas

Manzana

Pera

Piedra

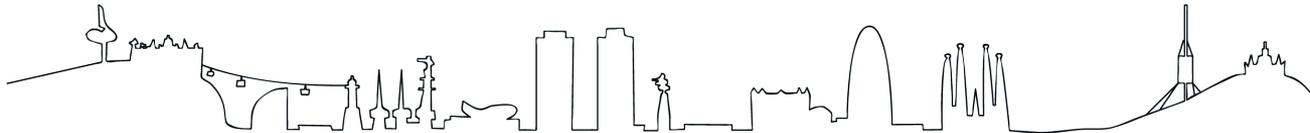


Salesforce

Acceso a las Herramientas de Salesforce

Queremos que nuestra solución esté integrada con Salesforce

- Mismo estilo de interfaz
- Reports & Dashboards
- Accesible para las diversas nubes
- Otras características propias de la plataforma



Survey Multicloud App

Implementación mediante
Lightning Components



Lightning Component Bundle

Component

Markup

CSS

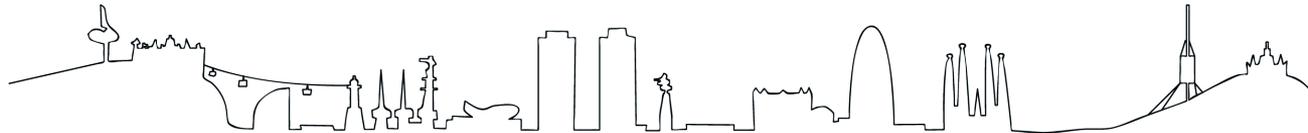
Style

Controller

Event Handler

Helper

Code Library



Lightning Component Bundle

Design

Parameters

Documentation

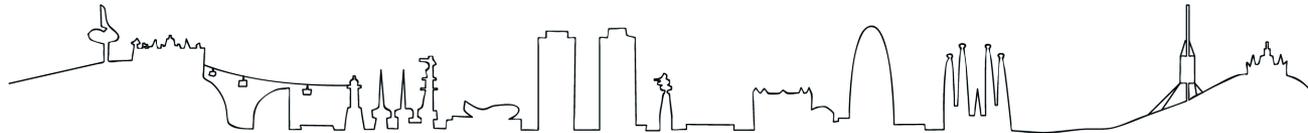
Reference

Renderer

Customize

SVG

Icon



Arquitectura

Contenedor

Vista

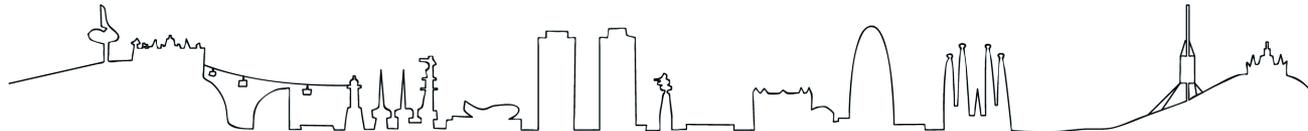
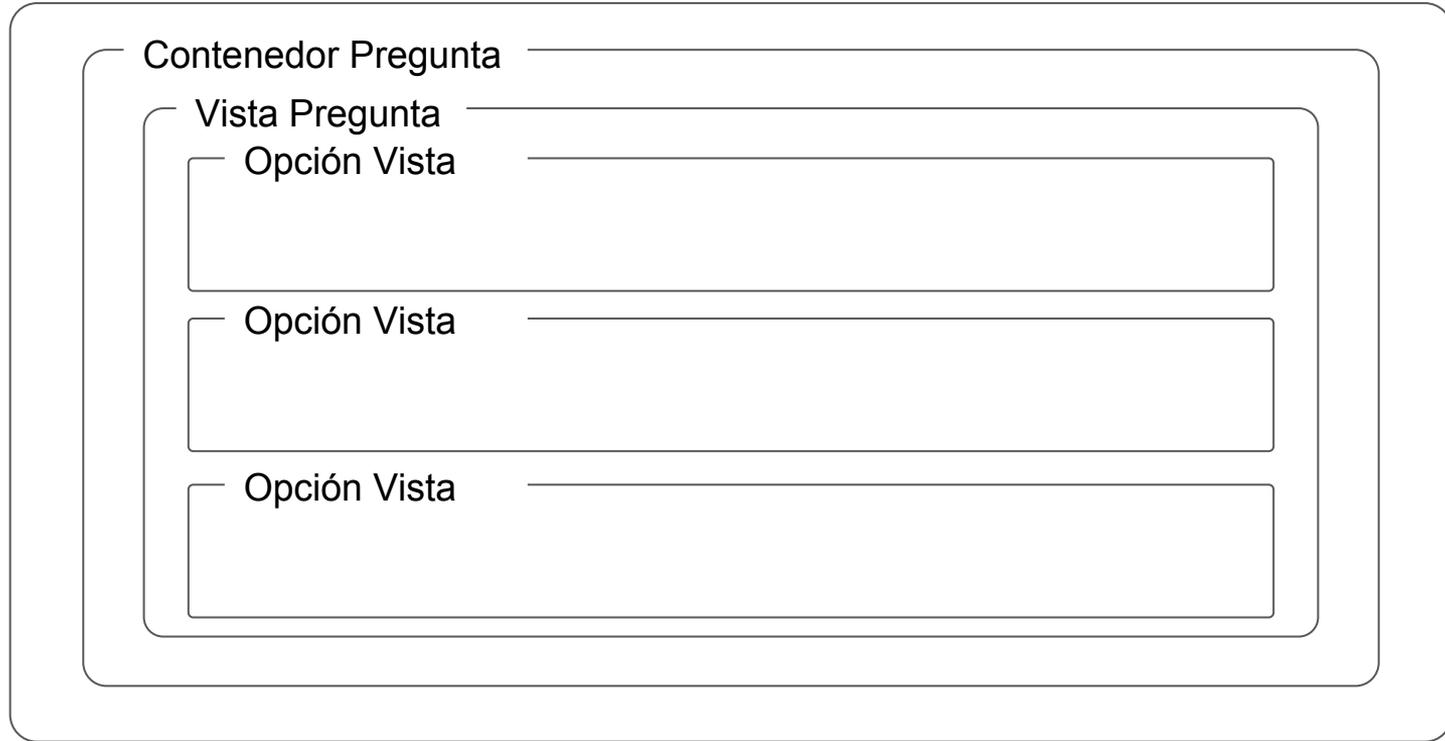
Pregunta Vista

Pregunta Vista

Pregunta Vista



Arquitectura



Lightning Data Service

force:recordData

Lightning Data Service nos permite realizar operaciones DML básicas sobre la base de datos

Pros:

- No se requiere Apex
- Sincronización de los datos entre componentes

Cons:

- No acepta lógica compleja
- Un solo registro
- Uso limitado a Lightning Experience y Salesforce App



Apex Controller

\$A.enqueueAction(action)

El uso de controladores apex nos permite realizar operaciones complejas sobre la base de datos

Pros:

- Permite trabajar con colecciones de registros
- Implementación de lógica compleja
- Accesible fuera de Lightning Experience y Salesforce App

Cons:

- Sin sincronización de datos entre componentes
- Require implementación explícita



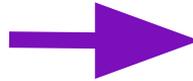
Null / Undefined

Query Database

```
return [ SELECT Id, Name, Description FROM Custom_sObject_c ]
```

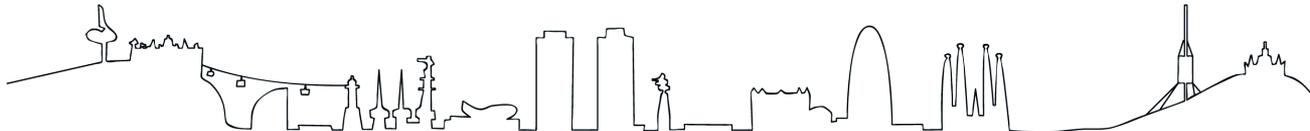
Apex

Id: a011r00003bybr7AAA
Name: RecordName
Description: null



JavaScript

Id: a011r00003bybr7AAA
Name: RecordName
Description: undefined



Null / Undefined

Update Database

Apex

Id: a011r00003bybr7AAA
Name: RecordName
Description: **Description**



JavaScript

Id: a011r00003bybr7AAA
Name: RecordName
Description: **Description**

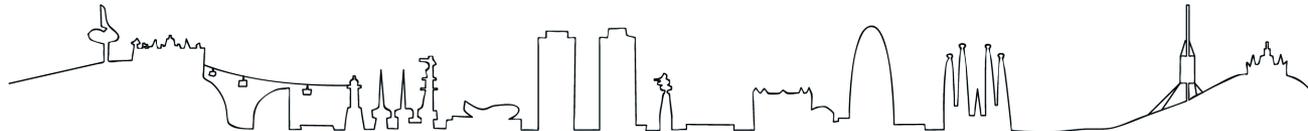
JavaScript

Id: a011r00003bybr7AAA
Name: RecordName
Description: **undefined**



Apex

Id: a011r00003bybr7AAA
Name: RecordName
Old Value



Null / Undefined

Update Database

Apex

Id: a011r00003bybr7AAA
Name: RecordName
Description: **Description**



JavaScript

Id: a011r00003bybr7AAA
Name: RecordName
Description: **Description**

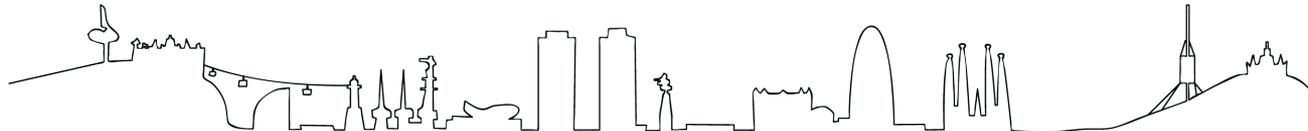
JavaScript

Id: a011r00003bybr7AAA
Name: RecordName
Description: **null**

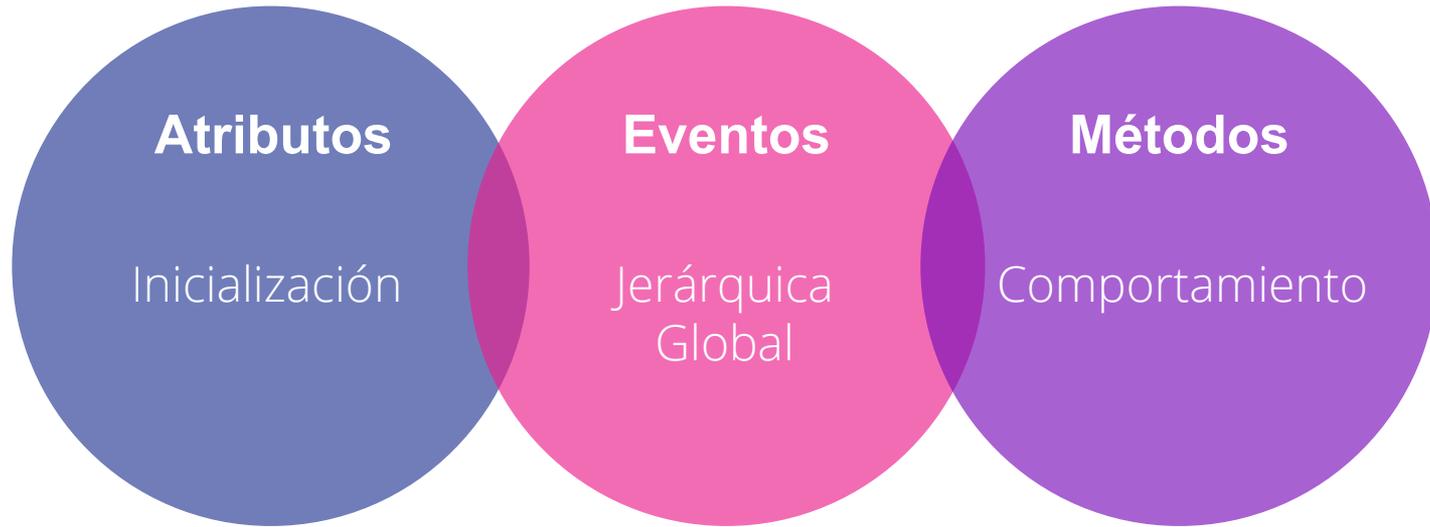


Apex

Id: a011r00003bybr7AAA
Name: RecordName
Description: **null**



Comunicación entre componentes



Atributos

Usos

Como en los Objetos, los atributos nos sirven para definir las propiedades de los componentes

- Los atributos pueden ser referenciados dinámicamente en el markup (!) o estáticamente (#)
- Almacenamiento y control del estado del componente



Caso de Uso

```
<aura:attribute name="questionRecordList" type="List" default="[]" />
```

```
<ul>
```

```
<aura:iteration items="{!v.questionRecordList}" var="questionRecord" >
```

```
<li class="slds-m-bottom_large" >
```

```
<c:SurveyQuestionEditorDisplay questionRecord="{!questionRecord}" />
```

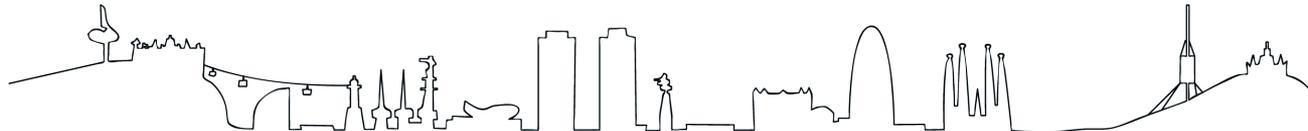
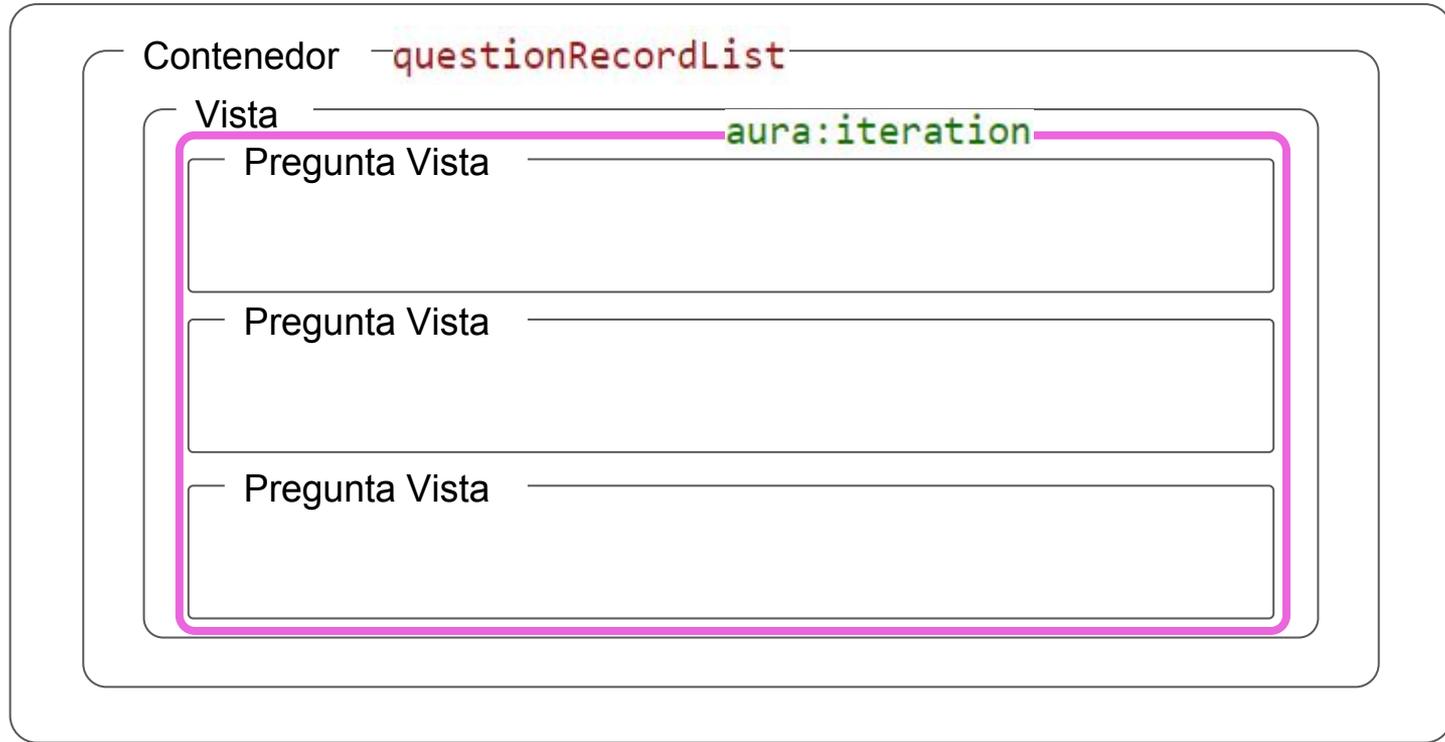
```
</li>
```

```
</aura:iteration>
```

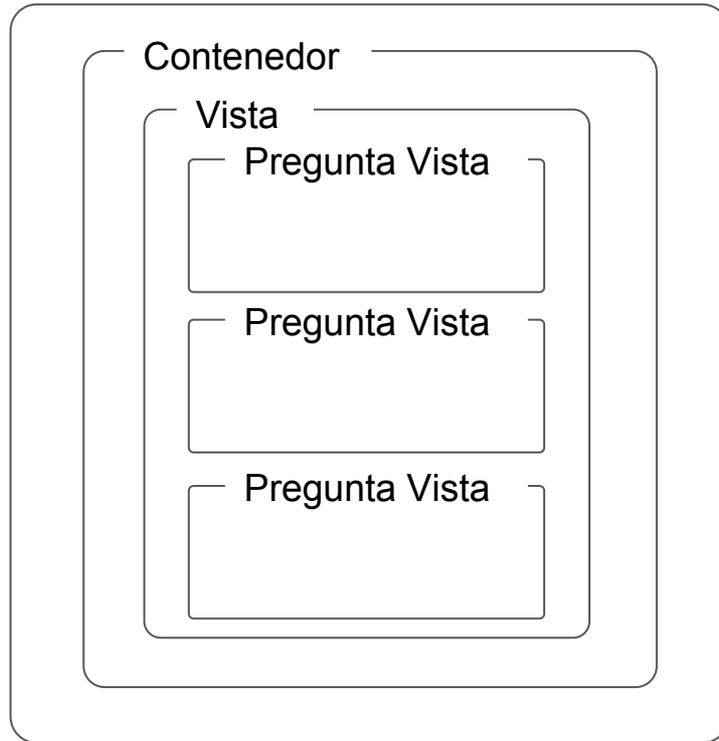
```
</ul>
```



Caso de Uso

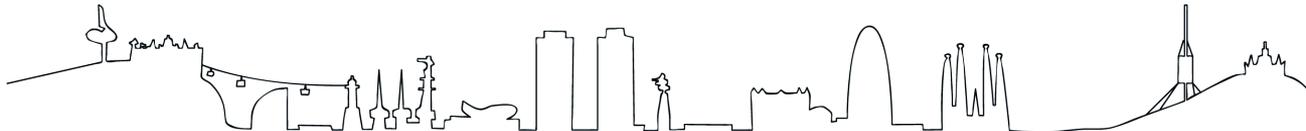


Eventos



Los eventos permiten la comunicación entre componentes

- A través de la Jerarquía
- A nivel de Aplicación



Caso de Uso

Definición del evento

```
<aura:event type="COMPONENT"  
    description="Fire event to delete a question" >  
    <aura:attribute name="questionRecord" type="Question__c"/>  
</aura:event>
```

- No estamos obligados a definir atributos
- Para eventos de tipo aplicación se utiliza como type APPLICATION



Caso de Uso

Lanzamiento del evento

```
<aura:registerEvent name="questionDeleteEvent" type="c:questionDeleteEvent"/>
```

```
deleteQuestion : function( component, event, helper ) {  
    var questionDeleteEvent = component.getEvent( "questionDeleteEvent" );  
    var questionRecord = component.get( "v.questionRecord" );  
    questionDeleteEvent.setParams( { "questionRecord" : questionRecord } );  
    console.log( questionRecord );  
    questionDeleteEvent.fire();  
},
```



Caso de Uso

Captura del evento

```
<aura:handler name="questionOptionDeleteEvent"  
    event="c:QuestionOptionDeleteEvent"  
    action="{!c.handleQuestionOptionDeleteEvent}"/>
```

```
handleQuestionDeleteEvent : function( component, event, helper ) {  
    var questionRecordList = component.get( "v.questionRecordList" );  
    var questionRecord = event.getParam( "questionRecord" );
```

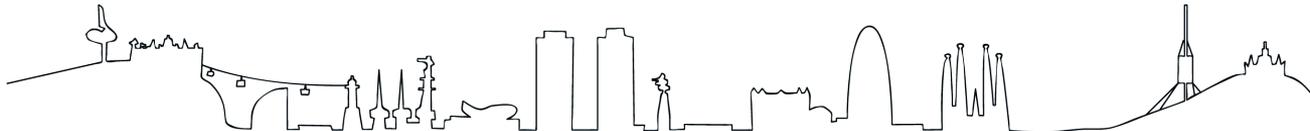


Métodos

Usos

Podemos exponer los Métodos de un componente para que puedan ser llamados desde otros componentes

- Ejecutar lógica sin forzar el renderizado completo del componente
- Recuperar información

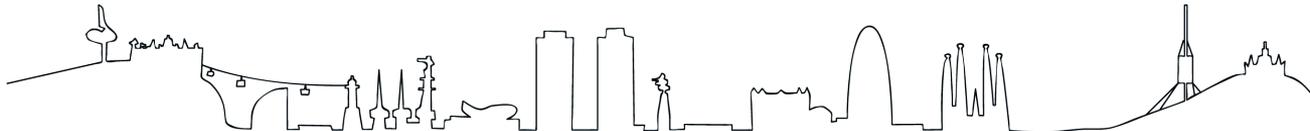


Caso de Uso

Exposición del Método

```
<aura:method name="getOptionInfo"  
             action="{!c.returnOptionInfo}"  
             description="Returns option data">  
</aura:method>
```

```
returnOptionInfo : function( component, event, helper ) {  
    var optionInfo = {};  
    optionInfo.optionText = component.find( "optionText" ).get("v.value");  
    optionInfo.optionChecked = component.get( "v.checked" );  
    return optionInfo;  
},
```



Caso de Uso

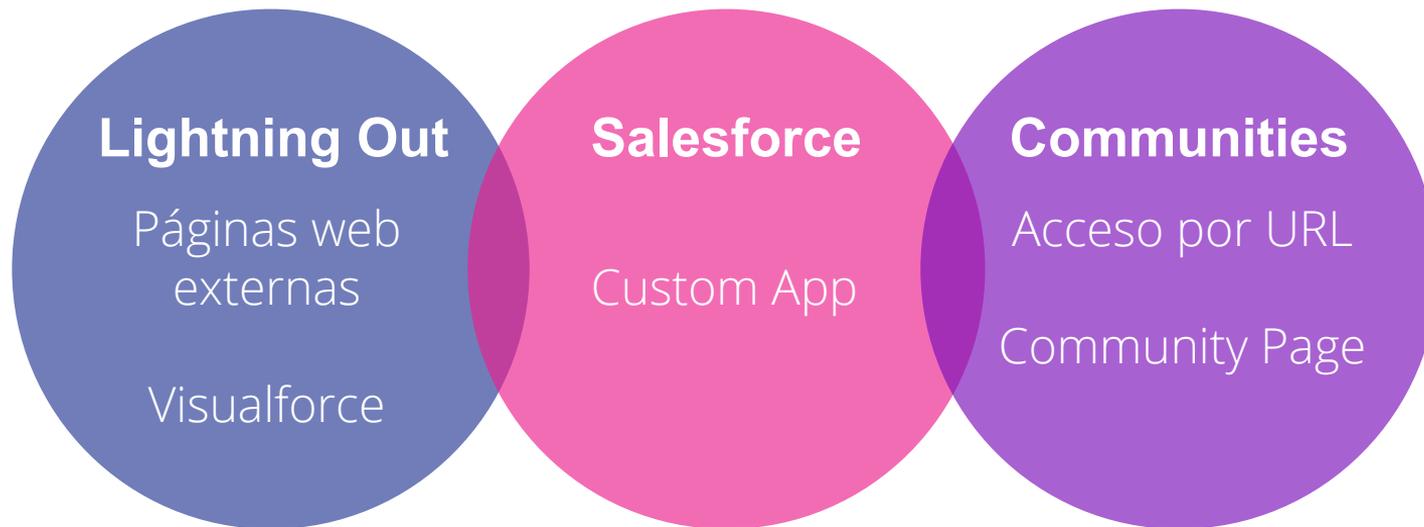
Llamada del Método

```
<c:QuestionOptionEditor aura:id="questionOption"
```

```
var questionOptionList = component.find( "questionOption" );  
for( var qol = 0; qol < questionOptionList.length; ++qol ){  
    var optionInfo = questionOptionList[ qol ].getOptionInfo();
```

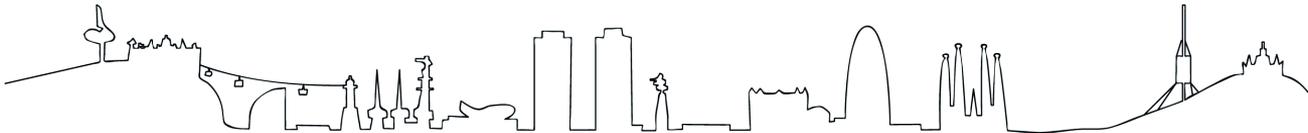


Distribución



Survey.app

```
<aura:application access="GLOBAL" extends="force:slds" implements="ltng:allowGuestAccess">  
  
  <aura:dependency resource="c:SurveyContainer"/>  
  <aura:dependency resource="c:SurveyDisplay"/>  
  <aura:dependency resource="c:SurveyQuestionDisplay"/>  
  <aura:dependency resource="c:QuestionOption"/>  
  <aura:dependency resource="c:QuestionOptionCheckEvent" type="EVENT"/>  
  <aura:dependency resource="c:SaveAnswerEvent" type="EVENT"/>  
  
  <aura:attribute name="surveyId" type="String" />  
  
  <c:SurveyContainer recordId="{!v.surveyId}" />  
  
</aura:application>
```



Q & A



Cognizant



FORMTITAN



MAKEMECLLOUD
consulting



#dreamOle18



Barcelona 2018