

E.T.S. de Ingeniería Industrial,  
Informática y de Telecomunicación

# Evolución de la plataforma CRM "ADEL" para la implementación de una herramienta de eLearning



Grado en Ingeniería  
en Tecnologías de Telecomunicación

Trabajo Fin de Grado

Fermín Ruiz Remírez

Jesús Daniel Trigo Vilaseca

Pamplona, 29/08/2018







## Resumen

La empresa Olmix Group comenzó a desarrollar una aplicación *Customer Relationship Management* (CRM) en 2016. Esta aplicación carecía de varias funcionalidades porque la prioridad inicial era que esta fuese sencilla, y los usuarios no sabían manejar la aplicación correctamente. Debido a esto, la empresa necesitaba tanto mejorar la aplicación como implementar una solución para enseñar su uso a los usuarios.

Este trabajo recoge primeramente la evolución de la aplicación CRM, que es necesario tanto para implementar las funcionalidades adicionales que necesita la empresa, como para poder desarrollar el objetivo principal del proyecto: una herramienta eLearning.

Entre las funcionalidades adicionales, se encuentran: evolución de la interfaz clásica a *Lightning*, evolución del modelo de datos de la aplicación, configuración de una aplicación móvil, *reporting* y la integración de la aplicación con el *Enterprise Resource Planning* (ERP) de la empresa. El 100% de los usuarios de ADEL ya están utilizando la versión actualizada de la aplicación, y los nuevos objetos configurados ya son utilizados por los usuarios, habiéndose generado más de 100000 registros relativos a los nuevos objetos.

Respecto a la herramienta eLearning, esta plataforma sirve principalmente para que los usuarios aprendan a utilizar la aplicación CRM, pero también se puede usar para introducir módulos de aprendizaje de cualquier tema. Se puede utilizar tanto desde el escritorio como desde la aplicación móvil, y está implementada para ser usada en múltiples idiomas, habiendo logrado un 90% de cubrimiento satisfactorio del código en las pruebas de validación.



# Índice

ÍNDICE DE FIGURAS .....	11
ÍNDICE DE ACRÓNIMOS .....	13
<b>1. INTRODUCCIÓN .....</b>	<b>15</b>
<b>1.1. Contexto: problemas y necesidades de la empresa .....</b>	<b>15</b>
Cómo se está trabajando inicialmente con ADEL .....	16
<b>1.2. Solución inicial de integración ADEL-ERP .....</b>	<b>16</b>
<b>1.3. Estudio de alternativas de la herramienta eLearning .....</b>	<b>17</b>
<b>1.4. Motivación .....</b>	<b>18</b>
<b>1.5. Objetivos .....</b>	<b>18</b>
<b>2. MATERIALES .....</b>	<b>21</b>
<b>2.1. Descripción del entorno .....</b>	<b>21</b>
¿Qué es un CRM? .....	21
Salesforce .....	22
Seguridad en el entorno .....	23
<b>2.2. Tecnologías involucradas .....</b>	<b>24</b>
Lenguajes de programación .....	24
Lightning Component Framework .....	25
Developer Console .....	26
<b>2.3. Entornos de trabajo .....</b>	<b>26</b>
<b>2.4. Aplicaciones de terceros y APIs .....</b>	<b>27</b>
AppExchange .....	27
Data Loader .....	27
Talend ETL .....	28
Business Card Reader .....	29
MavensMate .....	30
APIs .....	31
<b>3. METODOLOGÍA .....</b>	<b>33</b>
<b>3.1. Desarrollo funcional y personalizado .....</b>	<b>33</b>
<b>3.2. Mobile First .....</b>	<b>33</b>
<b>3.3. Metodología de gestión y desarrollo del proyecto .....</b>	<b>34</b>

<b>3.4. Planificación</b> .....	<b>34</b>
<b>3.5. Estructura del equipo</b> .....	<b>35</b>
<b>3.6. Proceso de validación</b> .....	<b>35</b>
<b>4. EVOLUCIÓN DE ADEL</b> .....	<b>37</b>
<b>4.1. Evolución de interfaz: Classic a Lightning</b> .....	<b>37</b>
Habilitar funcionalidades para Lightning .....	37
Modificación de permisos.....	37
Cambios en el código .....	38
Implementación progresiva .....	38
<b>4.2. Evolución del modelo de datos de ADEL</b> .....	<b>38</b>
Configuración e implementación del objeto “leads” .....	38
Configuración de oportunidades .....	41
Automatización de procesos.....	42
<b>4.3. Clasificación geográfica</b> .....	<b>43</b>
<b>4.4. Aplicación móvil</b> .....	<b>45</b>
Instalación de calendario .....	46
Menú de navegación.....	46
Acciones globales .....	47
Acciones rápidas específicas de objetos y formatos compactos .....	48
<b>4.5. Reporting</b> .....	<b>50</b>
<b>4.6. Integración ADEL-ERP</b> .....	<b>51</b>
Talend .....	51
<b>5. HERRAMIENTA ELEARNING</b> .....	<b>55</b>
<b>5.1. Diseño</b> .....	<b>55</b>
Diseño multiplataforma .....	55
<b>5.2. Lógica de negocio</b> .....	<b>56</b>
Módulos y unidades de usuario .....	57
Multiidioma.....	57
Multiactividad .....	59
Estructura de la lógica de negocio .....	59
<b>5.3. Lado del cliente</b> .....	<b>59</b>
Componente de página home y componente de perfil .....	61
Componente de información del módulo.....	63
Componente de contenido de la unidad .....	65
Componente padre y eventos.....	67
Etiquetas multiidioma .....	67



5.4. Lado del servidor.....	68
5.5. Gamificación .....	70
5.6. Seguridad .....	70
5.7. Validación y test.....	71
6. CONCLUSIONES .....	73
6.1. Puntos de mejora y líneas de trabajo futuro .....	73
7. BIBLIOGRAFÍA .....	75



## Índice de figuras

Figura 1 - Integración inicial ADEL-ERP .....	17
Figura 2 - Funciones de un CRM.....	21
Figura 3 - Logo de Salesforce.....	22
Figura 4 - Ejemplo de registro del objeto "Property" y campos .....	23
Figura 5 - Objeto "Property" y campos .....	26
Figura 6 - Relación de entornos .....	27
Figura 7 - Data Loader .....	28
Figura 8 - Interfaz de Talend ETL.....	29
Figura 9 - MavensMate .....	30
Figura 10 - Sublime Text 3.....	31
Figura 11 - Esquema de diseño "Mobile First" .....	33
Figura 12 - Estructura del equipo .....	35
Figura 13 - Lead Path.....	39
Figura 14 - Vista Kanban.....	40
Figura 15 - Conversión de lead.....	41
Figura 16 - "Path" de oportunidades .....	42
Figura 17 - Detalles del país ITALY.....	43
Figura 18 - Cuentas y regiones relacionadas con el país ITALY .....	44
Figura 19 - Calendario MobilePlus .....	46
Figura 20 - Menú de navegación .....	47
Figura 21 - Acciones globales de la aplicación .....	48
Figura 22 - Acciones específicas del objeto contacto .....	49
Figura 23 - Formato compacto del objeto contacto .....	49
Figura 24 - Panel de ventas .....	50
Figura 25 - Configuración de componentes en Talend .....	52
Figura 26 - Mapeado de campos en talend.....	53
Figura 27 - Estructura módulos/unidades.....	55
Figura 28 - Diseño multiplataforma .....	56
Figura 29 - Módulos y unidades de usuario .....	57
Figura 30 - Descripciones de módulo y unidad .....	58
Figura 31 - Estructura de objetos completa .....	59
Figura 32 - Contenido de archivo .cmp .....	60
Figura 33 - Controlador de componente Lightning.....	60
Figura 34 - Página Home en teléfono móvil.....	61
Figura 35 - Página Home en tablet.....	62
Figura 36 - Página Home en escritorio .....	62
Figura 37 - Información de módulo en teléfono móvil .....	63
Figura 38 - Información de módulo en tablet .....	64
Figura 39 - Información de módulo en escritorio .....	64
Figura 40 - Contenido de unidad en teléfono móvil .....	65
Figura 41 - Contenido de unidad en tablet .....	66
Figura 42 - Contenido de unidad en escritorio.....	66

Figura 43 - Estructura de evento Lightning .....	67
Figura 44 - Etiqueta personalizada traducida .....	68
Figura 45 - Estructura componente/servidor.....	69
Figura 46 - Estructura de clase Apex .....	69
Figura 47 - Estructura de clase de test.....	72

## Índice de acrónimos

ADEL – Advanced assistant for DEaLs  
ADV – Departamento de Administración de Ventas  
API – Application Programming Interface  
CRM – Customer Relationship Management  
CSP – Content Security Policy  
CSS – Cascading Style Sheets  
CSV – Comma-Separated Values  
DML – Data Manipulation Language  
DOM – Document Object Model  
ERP – Enterprise Resource Planning  
ETL – Extract, Transform, Load  
HTML – HyperText Markup Language  
HTTPS – Hypertext Transport Protocol Secure  
OCR – Optical Character Recognition  
SaaS – Software as a Service  
SOQL – Salesforce Object Query Language  
SOSL – Salesforce Object Search Language  
SQL – Structured Query Language  
SSL – Secure Sockets Layer  
TIC – Tecnologías de la Información y Comunicación  
XML – Extensible Markup Language



# 1. Introducción

## 1.1. Contexto: problemas y necesidades de la empresa

Este trabajo se ha realizado en la compañía Olmix Group. Olmix Group es una empresa multinacional de origen francés especializada en biotecnología marina y química verde o sostenible. Cuenta con presencia a nivel internacional, distribuyendo sus productos y actuando en más de 100 países a lo largo del mundo.

Olmix Group empezó el desarrollo de una aplicación *Customer Relationship Management* (CRM, “gestión de la relación con los consumidores”) interna en septiembre de 2016. A esta solución se la llamó *Advanced assistant for DEaLs* (ADEL), y actualmente está siendo utilizada por 150 personas de distintos departamentos.

ADEL está basado en Salesforce, una plataforma basada en la computación en la nube que ofrece una serie de soluciones, entre ellas una plataforma CRM.

Inicialmente, los tres objetivos principales de ADEL fueron:

- Reunir todo el conocimiento relativo a cuentas y contactos dentro de un único punto de confianza.
- Compartir la información de los diferentes equipos y departamentos de una manera más eficiente.
- Saber qué se está haciendo en la empresa y con quién, para poder ser capaces de crecer, adaptarse y mejorar la forma en la que se trabaja.

Sin embargo, Olmix es una compañía relativamente difícil a la hora de adoptar herramientas relativas a las Tecnologías de la Información y Comunicación (TIC), especialmente si estas nuevas herramientas provocan un impacto en los responsables de ventas de la empresa. Por ello, ADEL fue simplificado a su mínima expresión, evitando con ello algunas de las principales características de Salesforce que hacen que esta tecnología sea utilizable al máximo.

Esta simplificación está ayudando a la empresa en la fase de adopción en la que se encuentran al comienzo de este proyecto, pero inevitablemente no deja aprovechar todo su potencial. De hecho, conseguir que la aplicación se aproveche completamente es lo que se pretende con la primera parte de este proyecto, la actualización de ADEL (“ADEL v2”).

Con este proyecto, se pretende permitir la coexistencia entre dos mundos en el mismo entorno: un proceso simplificado para los usuarios ya existentes, y un proceso más completo que se quiere construir sobre ADEL, siempre tratando de estar adaptados a las funcionalidades estándar de Salesforce.

La dificultad a la hora de adoptar herramientas TIC también supone que los empleados de Olmix Group necesitan aprender el uso de la aplicación con la ayuda de algo o alguien que les guíe, por lo que hay una necesidad de encontrar la forma de enseñar a los usuarios de ADEL a utilizar esta aplicación.

## Cómo se está trabajando inicialmente con ADEL

Olmix es una compañía con diferentes actividades y divisiones, relacionadas con distintos productos. Sin embargo, lo común en la mayoría de estas actividades y divisiones es que habitualmente trabajan a través de distribuidores. Por este motivo, la empresa raramente llega a negociar y vender a usuarios finales del producto. Los distribuidores actualmente compran los productos de Olmix directamente contactando con el Departamento de Administración de Ventas (ADV) por teléfono, email, etc. Entonces, el equipo del ADV introduce la información correspondiente en el *Enterprise Resource Planning* (ERP, “planificación de recursos empresariales”) de la empresa, ejecutando los flujos pertinentes (producción, abastecimiento, envío, etc.).

También es importante recalcar que los responsables de ventas (junto con el departamento de Servicio Técnico) pasan mucho tiempo visitando y ayudando a los usuarios finales, aunque no se les venda el producto directamente. En esos casos lo que se consigue es ayudar a los distribuidores a cerrar más ventas.

Se podría decir que el modelo de distribución de Olmix es siempre el mismo: la empresa vende sus productos a los distribuidores, y estos venden los productos de Olmix al cliente final. Sin embargo, como hay algunos casos en los que Olmix vende su producto directamente al cliente final, se debe tener la posibilidad de reflejar ese tipo de ventas en ADEL.

Aparte de esto, también hay movimientos y ventas entre subsidiarios de Olmix. Cuando un subsidiario (por ejemplo, Olmix China) requiere cierta cantidad de un producto y no tiene existencias de él, lo pide oficialmente a un subsidiario productor (por ejemplo, la sede central en Olmix Bréhan). Esto genera una “transacción” entre subsidiarios de Olmix, reflejándose un contrato por los bienes requeridos.

La integración con el ERP es un problema con el que inicialmente no se está lidiando de forma correcta, ya que la información que se obtiene de él no se puede manipular y aprovechar como se debería. Por otro lado, ADEL se ha simplificado en exceso, por lo que no puede dar las funcionalidades necesarias para cubrir los casos de uso que le daría la empresa teniendo en cuenta lo mencionado anteriormente.

### 1.2. Solución inicial de integración ADEL-ERP

El ERP está ya conectado a ADEL, aunque de una forma ineficiente, que se puede observar en la figura 1:



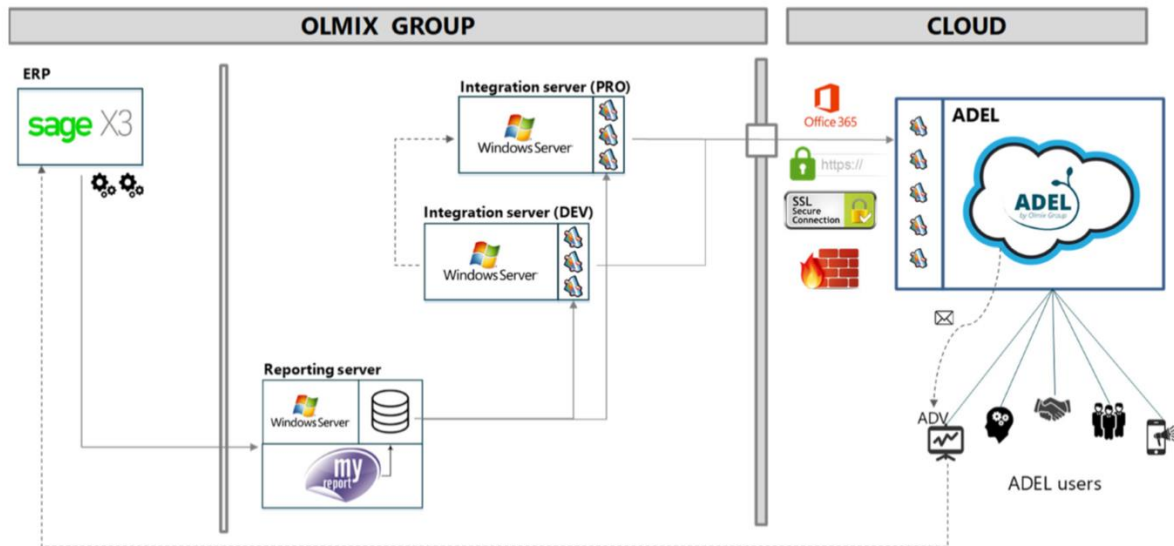


Figura 1 - Integración inicial ADEL-ERP

Internamente, la principal fuente de información financiera es el servidor SAGE X3. Sin embargo, como no podemos atacarlo directamente (debido a restricciones técnicas impuestas por los responsables de esa área), se ofreció una fuente de información intermediaria bajo la plataforma “Reporting server”.

Este es un entorno que crea una base de datos consolidada todas las noches. Después de cada proceso, la información contenida en la base de datos es destruida y regenerada otra vez. También, esta base de datos es de sólo lectura desde ADEL, por lo que no se puede escribir información, ni crear nuevos campos o tablas en ella. Esto implica que ADEL no tiene ninguna forma de introducir información en el entorno de SAGE X3. Aparte de ello, este servidor utiliza una tecnología llamada MyReport que crea una base de datos en SQL.

Como la fuente de la información financiera puede cambiar en el futuro y debe poder adaptarse rápidamente a ese cambio, se decidió no conectar ADEL directamente a la estructura de la base de datos. Esto podría crear mucho acoplamiento de datos entre ADEL y la base de datos, lo que debe ser evitado si es posible. Por ello, se han desplegado dos servidores llamados “Integration Server (PRO)” y “Integration Server (PRE/DEV)”.

Esta es una solución muy enrevesada para el problema, además de ser muy limitante a la hora de manipular los datos recibidos, por lo que se decidió cambiar la forma de recibir los datos del ERP.

### 1.3. Estudio de alternativas de la herramienta eLearning

Antes de comenzar ningún desarrollo de la herramienta eLearning, se estudió si era rentable realizar la solución desde cero, y se encontraron una serie de alternativas:

- **Aplicaciones de terceros en AppExchange:** tras explorar las opciones disponibles de aplicaciones eLearning de terceros, se llegó a la conclusión de que no era una solución apropiada, ya que las aplicaciones suficientemente profesionales valían un precio por licencia que Olmix Group no estaba dispuesto a pagar. Además, las soluciones de AppExchange no estarían integradas en la aplicación ADEL, y no permitirían un nivel de personalización que permite realizar una plataforma de cero.
- **myTrailhead:** myTrailhead es una plataforma de aprendizaje desarrollada por Salesforce que pretende llevar su propia plataforma eLearning (Trailhead) a cada organización, permitiendo un buen nivel de personalización y manteniendo muchas funcionalidades de la plataforma original. A pesar de que se trataría de una herramienta muy potente y personalizable, y que podría estar integrada en ADEL, tiene dos inconvenientes: esta funcionalidad no saldrá hasta 2019, y costará un precio por licencia, cuya cantidad aún se desconoce [1].

Por ello, la solución escogida fue la de diseñar la plataforma eLearning desde cero, ya que es la solución más barata y flexible.

#### 1.4. Motivación

Por una parte, se realizó una encuesta a usuarios de la aplicación, para examinar el nivel y la frecuencia de uso que daba cada uno y por qué. Los resultados reflejaron que la principal razón de los usuarios por la que utilizaban poco ADEL era que no sabían usarla correctamente. Los usuarios de ADEL están repartidos en muchos equipos a lo largo de todo el mundo, por lo que las diferencias horarias y la gran distancia hace difícil el enseñar a todos los empleados cómo utilizar la aplicación y la cantidad de funcionalidades disponibles.

La aplicación ADEL tiene una serie de carencias en la funcionalidad que deben ser solucionadas para que los usuarios puedan utilizarla conforme a sus necesidades, así como para poder desarrollar una herramienta para que los usuarios aprendan a utilizar ADEL. También se necesita un medio cómodo para utilizar la aplicación cuando no se dispone de un ordenador.

#### 1.5. Objetivos

El principal objetivo de este proyecto es el desarrollo y la implementación de una plataforma eLearning, que satisfaga la necesidad de la empresa de tener una herramienta ágil y duradera para que los usuarios puedan aprender las funcionalidades de ADEL. Esta plataforma debe funcionar en distintos dispositivos, como teléfono móvil, tablet o escritorio, y tiene que estar integrada en la aplicación ADEL.

Por ello, se necesita también realizar una actualización de ADEL, renovando completamente la interfaz de la aplicación, y añadiendo una serie de funcionalidades necesarias

para hacer posible el desarrollo de la plataforma eLearning, además de otras funciones requeridas por la empresa para mejorar la usabilidad y las funciones de ADEL.



## 2. Materiales

### 2.1. Descripción del entorno

¿Qué es un CRM?

Las siglas CRM quieren decir “*Customer Relationship Management*”, que en español significa “gestión de las relaciones con los clientes”. Básicamente, un CRM es una tecnología utilizada para gestionar todas las interacciones y relaciones que una empresa tiene tanto con sus clientes actuales como con clientes potenciales (figura 2). El objetivo de un sistema CRM es ayudar a las empresas a mejorar su productividad y gestionar y monitorizar los diferentes procesos de ventas y relaciones comerciales [2].



Figura 2 - Funciones de un CRM

Los sistemas CRM comienzan recolectando la información de los distintos clientes de una empresa, como su página web, email o teléfono, además de noticias sobre la actividad de empresas relacionadas y detalles personales. El CRM organiza toda esta información y ofrece registros generales de individuos y compañías. También permite visualizar datos de ventas y actividades y representarlos gráficamente, además de importar y exportar la información almacenada.

Así, además de gestionar las relaciones de una empresa con sus clientes, otras de sus principales funciones son:

- Gestión de ventas e ingresos
- Gestión de contactos
- Gestión de documentos
- Comunicación interna
- Análisis de marketing
- Presupuesto y facturación
- Informes

## Salesforce

Olmix Group utiliza una plataforma CRM en concreto llamada Salesforce (figura 3). Salesforce es la plataforma CRM líder en el mundo, siendo utilizada por más de 150000 empresas en la actualidad [3]. Esto es debido a una serie de características que hacen ventajosa a esta plataforma respecto a sus competidores.



Figura 3 - Logo de Salesforce

### Descripción y servicios

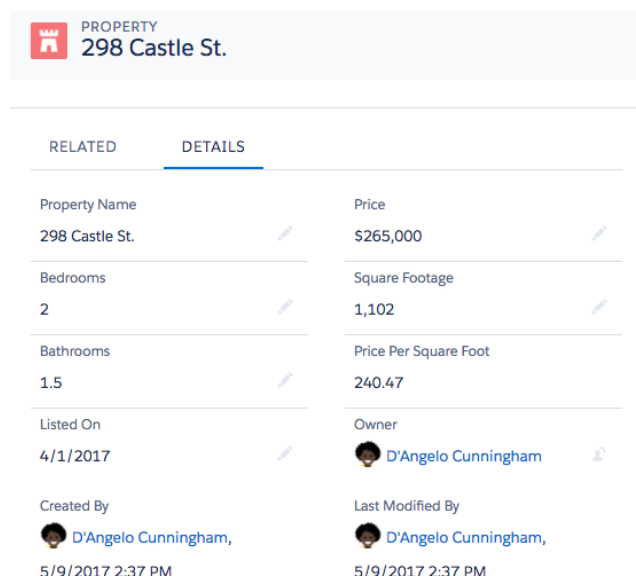
Una de las principales características de Salesforce es que está basado en la computación en la nube (Cloud Computing), lo que evita muchos problemas de implementación y mantenimiento que acarrea alojar una solución CRM en el propio servidor de la empresa. Salesforce comenzó ofreciendo soluciones *Software as a Service* (SaaS), pero ahora da distintas soluciones y ofrece una plataforma para desarrollar y distribuir software personalizado [3].

Salesforce ofrece distintos servicios y productos, como *Sales Cloud*, *Marketing Cloud*, *Service Cloud*, etc. *Sales Cloud* es la plataforma CRM de Salesforce, siendo este el principal servicio utilizado por Olmix.

Además, Salesforce proporciona a sus usuarios otros servicios como *Chatter*, una red social colaborativa dentro de la plataforma, utilizada para compartir información entre departamentos de la empresa. Salesforce también permite la implementación de *Salesforce Mobile*, una aplicación móvil que se puede personalizar para poder utilizar la información de Salesforce en un dispositivo Android o iOS.

### Estructura y modelo de datos en Salesforce

En el modelo de datos de Salesforce, el equivalente a las tablas en bases de datos se conoce como objetos, las columnas se llaman campos, y las filas serían registros. Si estamos hablando del modelo de datos de Salesforce, hablamos sobre el conjunto de objetos, campos y registros que se encuentran en la aplicación (Ilustración 1) [4].



RELATED	DETAILS
Property Name	Price
298 Castle St.	\$265,000
Bedrooms	Square Footage
2	1,102
Bathrooms	Price Per Square Foot
1.5	240.47
Listed On	Owner
4/1/2017	D'Angelo Cunningham
Created By	Last Modified By
D'Angelo Cunningham, 5/9/2017 2:37 PM	D'Angelo Cunningham, 5/9/2017 2:37 PM

Figura 4 - Ejemplo de registro del objeto "Property" y campos

Salesforce soporta distintos tipos de objetos. Los dos tipos más comunes de objeto son los objetos estándar y los objetos personalizados. Los objetos estándar son aquellos que vienen incluidos en Salesforce y los objetos personalizados son creados por los usuarios desarrolladores para guardar información específica para la propia compañía [4].

## Seguridad en el entorno

Respecto a la seguridad y la protección en ADEL, hay medidas para proteger la información tanto en el interior de la organización como frente a agentes externos.

### Seguridad interna: roles y perfiles

Los perfiles son conjuntos de permisos y configuraciones que definen cómo pueden acceder los usuarios a los distintos objetos y datos [5].

Los perfiles controlan:

- **Permisos a nivel de objeto:** qué objetos puede ver y editar el usuario

- **Permisos a nivel de campo:** qué campos de un objeto en concreto puede ver y editar el usuario
- **Permisos de usuario:** las funciones específicas que puede realizar cada usuario
- **Otros permisos:** permisos de pestaña, aplicación, horas para iniciar sesión, etc.

Por otro lado, los roles controlan los permisos de acceso a nivel de registro [5]. La jerarquía de roles sirve principalmente para permitir el acceso a los datos a empleados que se encuentran por encima en la jerarquía de la empresa.

### *Seguridad externa: certificados y claves*

La aplicación tiene una serie de medidas utilizadas para proteger la información frente a agentes externos [6]. Las principales medidas son:

- Gestión de certificados y claves
- Seguimiento de auditoría
- Control de sitios de confianza de política de seguridad de contenidos (CSP)
- Uso de HTTPS y SSL ante cualquier petición web de terceros o cualquier llamada a APIs

## 2.2. Tecnologías involucradas

### Lenguajes de programación

Como se ha mencionado en el apartado anterior, el *framework* utilizado para realizar los componentes y aplicaciones de este proyecto utiliza JavaScript en el lado del cliente y Apex en el lado del servidor.

Por un lado, JavaScript es un lenguaje de programación dinámico y completo que, cuando es aplicado a un documento HTML, proporciona interactividad en páginas web. JavaScript es un lenguaje compacto, flexible y orientado a objetos. Permite el uso de funciones de primera clase y es un lenguaje multi-paradigma [7].

Por otro lado, Apex es un lenguaje de programación orientado a objetos y fuertemente tipado, que utiliza una sintaxis similar a Java y que actúa como una serie de procedimientos de llamadas a bases de datos [8].

Apex es un lenguaje integrado. Proporciona soporte incorporado para múltiples idiomas, incluyendo:

- Llamadas de manipulación de datos (DML) como INSERT, UPDATE y DELETE.
- Peticiones en línea de Salesforce Object Query Language (SOQL) y Salesforce Object Search Language (SOSL) que devuelven registros sObject.



- Bucles para realizar procesamientos en masa.
- Llamadas a la API pública que pueden ser construidas a partir de métodos Apex.

Además, Apex es un lenguaje fuertemente tipado que usa referencias directas a esquemas de objetos y campos, y falla rápidamente a la hora de compilar si alguna de las referencias es inválida. Almacena todas las dependencias de campo, objeto y clase personalizadas en metadatos para garantizar que no se eliminen mientras el código Apex activo lo requiera.

Por último, los componentes Lightning son unidades independientes y reutilizables de una aplicación. Además del conjunto de componentes que vienen integrados en el *framework*, se pueden crear nuevos componentes, y combinarlos entre sí para realizar componentes más complejos. Los componentes son renderizados para producir elementos HyperText Markup Language (HTML) Document Object Model (DOM) en el navegador [9] [10].

Los recursos de los componentes tienen una estructura de marcas similar a Extensible Markup Language (XML) o HTML, y un componente Lightning puede contener otros componentes, así como HTML, Cascading Style Sheets (CSS) y JavaScript, o cualquier código habilitado en web.

## Lightning Component Framework

El *framework* de componentes Lightning es un *framework* de interfaz de usuario para el desarrollo de aplicaciones web dinámicas, tanto para dispositivos de escritorio como para dispositivos móviles, en concreto aplicaciones de página única diseñadas para ser escalables [11].

El *framework* soporta el desarrollo de componentes divididos en múltiples niveles que unen el cliente y el servidor. En el lado del cliente el *framework* utiliza JavaScript, y en el lado del servidor utiliza Apex.

El hecho de utilizar este *framework* tiene una serie de características que lo hacen muy provechoso:

- Viene con un conjunto de componentes ya predefinidos y disponibles para su uso inmediato, que están optimizados para ser aplicados en múltiples dispositivos.
- Tiene un amplio ecosistema de componentes, pudiendo implementarlos de forma que sean fácilmente accesibles por los usuarios. Se pueden obtener componentes adicionales en AppExchange, así como se pueden publicar y compartir los componentes propios de cada uno.
- Utiliza una arquitectura de cliente con estado y de servidor sin estado, por lo que el cliente sólo llama al servidor cuando es absolutamente necesario (como peticiones de datos o metadatos) para maximizar la eficiencia y rendimiento.
- Utiliza una arquitectura basada en eventos para tener una mejor separación entre componentes.

- Tiene una gran compatibilidad entre navegadores y un diseño responsivo, además de ser capaz de detectar el tipo de dispositivo utilizado.

## Developer Console

La Developer Console (consola de desarrollo) es un entorno de desarrollo integrado que posee un conjunto de herramientas que se pueden utilizar para crear, depurar y probar aplicaciones dentro de la organización de Salesforce (figura 5) [12].

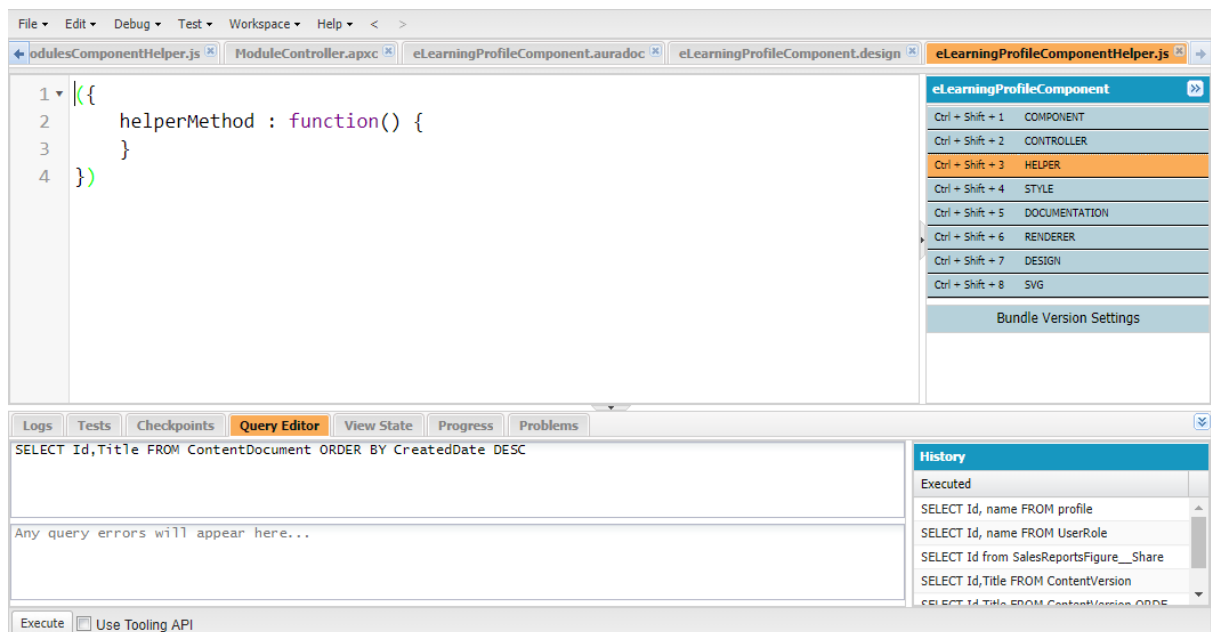


Figura 5 - Objeto "Property" y campos

Como se puede ver en la ilustración superior, la consola de desarrollo tiene múltiples usos, entre ellos se pueden destacar los siguientes:

- Crear, editar, depurar y probar código Apex.
- Ejecutar peticiones SOQL y SOSL y ver su resultado.
- Visualizar logs, errores y advertencias.
- Ver, crear y editar componentes Lightning

### 2.3. Entornos de trabajo

Respecto a cómo se ha realizado la implementación tanto de las nuevas funcionalidades de la aplicación ADEL como de la plataforma de eLearning, en ambos casos se ha seguido una metodología similar, ayudándonos de la disponibilidad de dos entornos de trabajo, el entorno

“sandbox” de desarrollo (DEV) y el entorno de producción (PRO), como se puede ver en la figura 6:

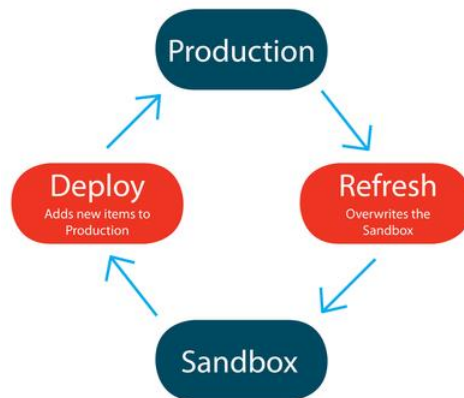


Figura 6 - Relación de entornos

Por un lado, se encuentra el entorno de producción, que es el utilizado por los usuarios de ADEL a diario y no se puede modificar con total libertad. Por otro lado, el entorno de pruebas de desarrollo es lo que se denomina una “partial copy” del entorno de producción [13]. Esto es, cuando los administradores de la organización quieran (con un límite de una vez cada 5 días) pueden realizar una copia parcial del entorno de producción. En concreto, al entorno de desarrollo se copia toda la estructura y los metadatos del entorno de producción, y una muestra de los datos y registros almacenados.

## 2.4. Aplicaciones de terceros y APIs

En la realización de este proyecto, ha sido necesario el uso de aplicaciones externas y de terceros, así como APIs (tanto internas como externas).

### AppExchange

AppExchange es una página de Salesforce en la que terceros crean, ofrecen y venden sus aplicaciones, extensiones, servicios y otras soluciones de software relacionadas con Salesforce [14]. Cualquier usuario puede crear, publicar, instalar y valorar aplicaciones, componentes y datos. Esta página también sirve para ofrecer o buscar trabajo de desarrollador o consultor.

### Data Loader

Data Loader (figura 7) es una aplicación que sirve para importar, exportar, modificar o eliminar datos en masa en Salesforce [15]. Se importan datos a partir de archivos CSV, que son mapeados según sus columnas a campos del objeto al que se quieren importar.

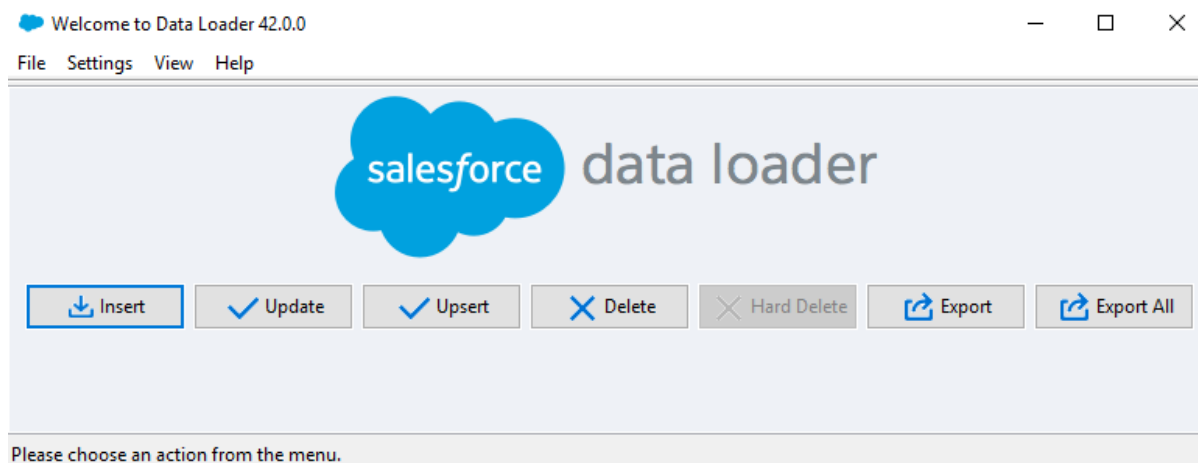


Figura 7 - Data Loader

A la hora de exportar, se realiza una petición SOQL con los campos del objeto que se desean exportar y las condiciones que deben cumplir, y la salida se recoge en un archivo *Comma-Separated Values* (CSV).

Esta herramienta ha sido de utilidad en este proyecto a la hora de importar datos en masa a los usuarios de ADEL, en especial datos del objeto “candidatos” creado en la evolución de la aplicación.

## Talend ETL

“*Extract, Transform, Load*” (ETL) es el proceso de obtener datos de múltiples orígenes, organizar y manipular esa información, y agruparla en un repositorio. Las herramientas ETL realizan estos tres procesos (extracción, transformación y carga) secuencialmente, lo que permite, si se desea, invertir los dos últimos pasos (ELT) [16].

Para este proyecto, se ha utilizado la solución ETL de Talend (figura 8). Talend es un vendedor de soluciones de integración de código abierto. El ETL de Talend está basado en Eclipse, una plataforma de desarrollo de software programada en Java.

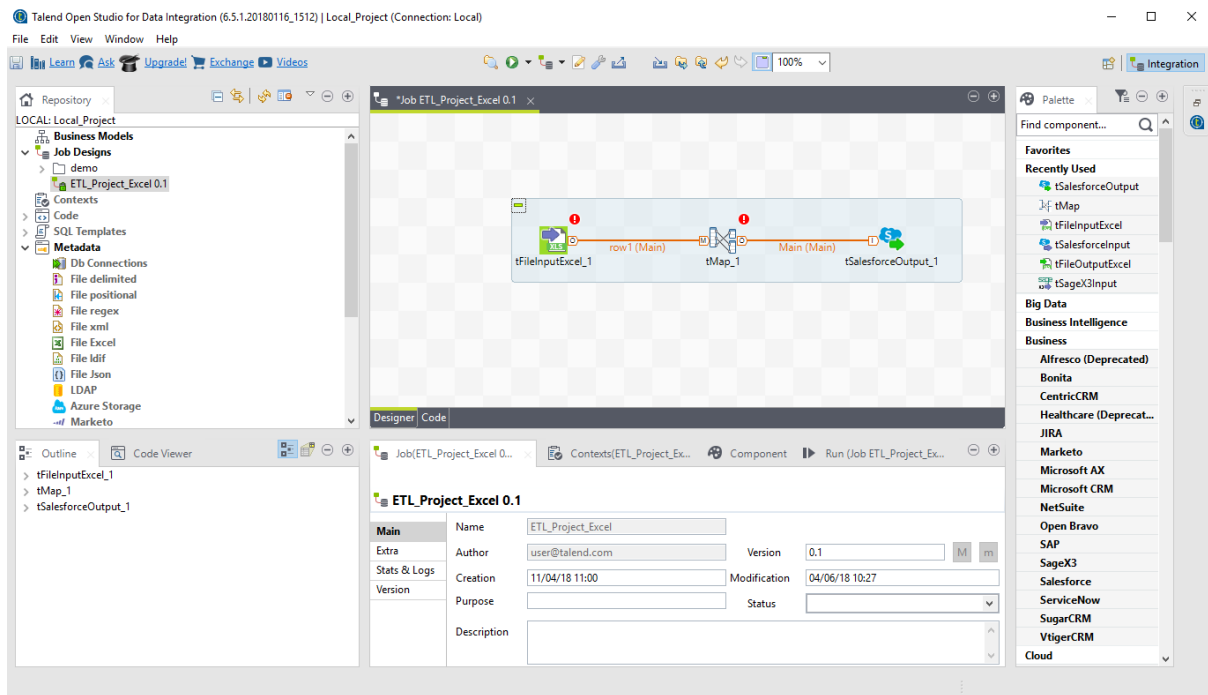


Figura 8 - Interfaz de Talend ETL

Los proyectos realizados con Talend se componen de diseños gráficos llamados jobs (trabajos). Estos contienen componentes conectados entre sí, que permiten configurar y ejecutar procesos de flujos de datos. Los componentes son conectores preconfigurados que realizan una operación de integración de datos específica. Utilizando componentes se minimiza la necesidad de escribir código [17].

## Business Card Reader

Aprovechando una de las nuevas funcionalidades de ADEL introducidas en la evolución, la configuración del objeto “candidatos”, se decidió explorar la posibilidad de añadir la función de escanear tarjetas de visita para guardar la información de posibles clientes en ADEL en forma de candidatos.

Ninguna de las soluciones encontradas en AppExchange era capaz de ser integrada en la misma aplicación, por lo que finalmente se optó por utilizar la aplicación *Business Card Reader for Salesforce CRM* de MagneticOne Mobile. Se han configurado los campos en ADEL pertinentes para que, a partir de la información escaneada, se pueda crear un registro de contacto o de candidato. La aplicación está disponible tanto en Android como en iOS, y su funcionamiento es el siguiente:

- El usuario inicia sesión en la aplicación con su cuenta de ADEL.
- Selecciona el tipo de objeto al que pertenece el futuro registro (contacto o candidato).
- Se fotografía la tarjeta de visita en la que se puede observar la información relativa al contacto o candidato.

- La imagen es enviada al servidor, donde este aplica un proceso *Optical Character Recognition* (OCR) y transforma esa información en campos del objeto en concreto.
- Se previsualiza el nuevo registro, permitiendo al usuario añadir o editar cualquier campo, y se sube la información a ADEL.

## MavensMate

MavensMate (figura 9) es una colección de proyectos de código abierto cuyo objetivo es dar la posibilidad a los desarrolladores de crear y editar aplicaciones de Salesforce utilizando editores de texto en su máquina local, como Sublime o Atom, en lugar de utilizar la “*developer console*” de Salesforce.

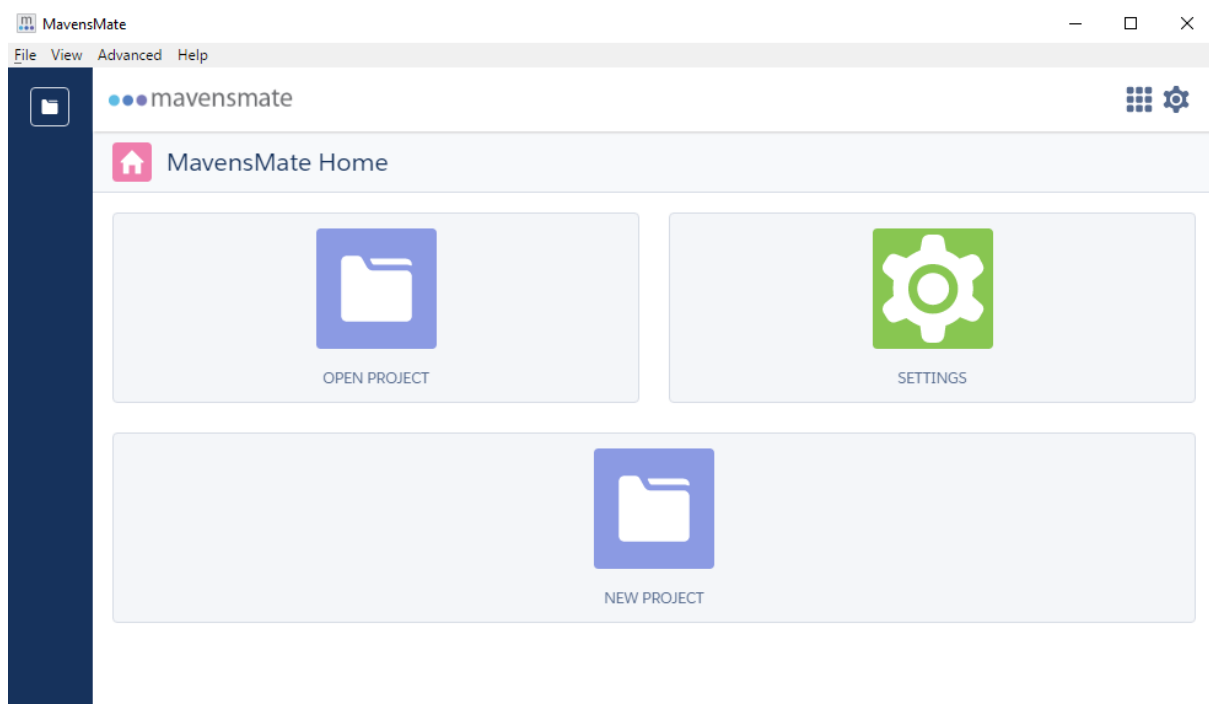


Figura 9 - MavensMate

Para este proyecto, se ha utilizado MavensMate junto con un plugin de Sublime Text 3 (figura 10) para desarrollar la herramienta eLearning, ya que es mucho más intuitivo, cómodo y funcional que la “*developer console*”. Con esta herramienta se pueden crear clases Apex, componentes Lightning, eventos, clases de test y más, y al guardar un archivo desde Sublime, se sincroniza automáticamente y se compila el código en Salesforce.

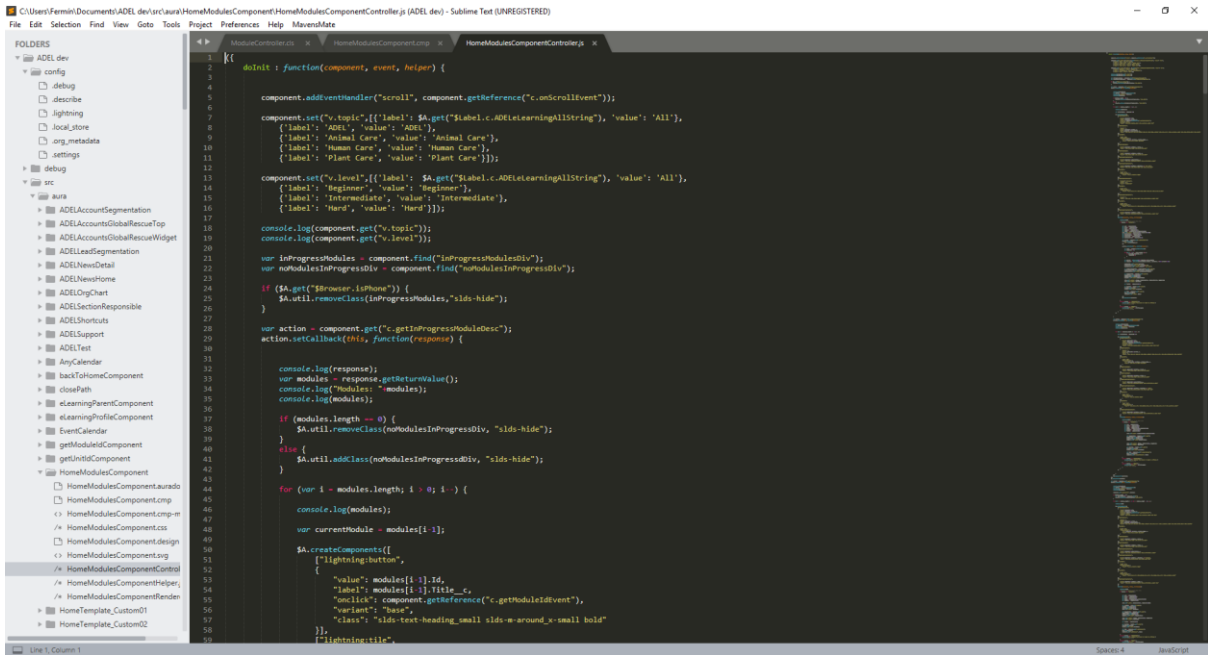


Figura 10 - Sublime Text 3

Además, se pueden personalizar los metadatos, ejecutar tests de clases, exportar las clases y los componentes, y cambiar entre los entornos de desarrollo y producción.

## APIs

Salesforce utiliza interfaces de programación de aplicaciones (APIs) para proporcionar acceso programático a la información de ADEL, de forma sencilla y segura. En este proyecto se han utilizado APIs tanto de forma interna (dentro de la organización) como externa (conectando con aplicaciones de terceros).

Entre las APIs utilizadas de forma interna, caben destacar las APIs de Apex para el intercambio de información entre componentes y clases. Respecto a las APIs externas, la API masiva es utilizada por las herramientas de Talend y Data Loader para consultar, insertar, actualizar o eliminar muchos registros enviando lotes. Y MavensMate utiliza la API de metadatos para gestionar personalizaciones del modelo de datos sin modificar los datos en sí.





## 3. Metodología

### 3.1. Desarrollo funcional y personalizado

Una de las ventajas de utilizar Salesforce es el gran abanico de funcionalidades que ya vienen incluidas, y el ahorro de tiempo escribiendo código que se consigue al utilizar esta plataforma. Salesforce ha avanzado de tal forma que se puede configurar e implementar una versión básica y estándar de una aplicación CRM con lo que se conoce como metodología de programación declarativa: el desarrollo que típicamente se haría escribiendo código se realiza vía una interfaz “*point and click*” y “*drag and drop*”.

Además de la eficiencia que supone poder realizar la mayoría del trabajo de esta forma, también tiene la ventaja de que todo lo que esté desarrollado por este método será robusto, y no tendrá errores internos ya que todo el código detrás de la interfaz utilizada ha sido verificado y comprobado.

No obstante, esta metodología de programación sirve para implementar una solución genérica, y para conseguir todas las funcionalidades deseadas se necesita escribir código personalizado, como por ejemplo para crear nuevos componentes, reglas y filtros, o campos de los objetos cuyo valor depende de fórmulas.

### 3.2. Mobile First

Para el desarrollo de este proyecto, se ha llevado la filosofía de desarrollo Mobile First. *Mobile first* (“los móviles primero”) se refiere a un modo de diseñar que tiene en cuenta primero la visualización en dispositivos móviles y de pantalla pequeña, para luego escalar a pantallas más grandes, como tablet o escritorio (figura 11) [18].



Figura 11 - Esquema de diseño "Mobile First"

La idea es conseguir mostrar todo el contenido y las funcionalidades más relevantes correctamente en los teléfonos móviles, ya que son los que más limitaciones tienen en cuanto a pantalla, ancho de banda y capacidad de procesamiento. Una vez conseguido un buen

resultado en dispositivos móviles, es mucho más sencillo adaptar el contenido a una pantalla más grande, que lo que hubiera costado realizar el proceso contrario.

Además, en muchas ocasiones, el contenido requerido por un usuario en un dispositivo móvil será diferente a la información consultada en escritorio. Por ello, conviene priorizar la muestra del contenido solicitado en los dispositivos con menos capacidad con el mejor rendimiento disponible, y actuar en otros dispositivos a partir de ahí.

### 3.3. Metodología de gestión y desarrollo del proyecto

Cada parte del proyecto se ha llevado a cabo de acuerdo con distintas metodologías de desarrollo, según la necesidad o urgencia de implementar las diferentes funcionalidades desarrolladas.

La parte de evolución de ADEL ha sido realizada siguiendo lo que se conoce como una metodología ágil. La metodología de desarrollo de software ágil está basada en el desarrollo iterativo, donde los requerimientos y soluciones evolucionan conforme el proyecto avanza. Este tipo de metodología se trabaja con iteraciones, en cada cual se debe realizar unas etapas de planificación, análisis, diseño, desarrollo y prueba [19].

Conforme se trabajaba en la evolución de ADEL, se implementaban las soluciones ya probadas en el entorno de desarrollo, y se obtenía una retroalimentación de estas para comenzar otra iteración y seguir adelante, solapándose así las etapas, en lugar de desarrollar todos los cambios planificados e implementar toda la evolución al final del desarrollo.

En cambio, con la plataforma de eLearning no se ha seguido una metodología estrictamente ágil, ya que, aunque también se ha trabajado de forma iterativa, la implementación del software no se ha llevado a cabo hasta haber finalizado el desarrollo de la plataforma por completo.

### 3.4. Planificación

Respecto a la planificación seguida en este proyecto, al haber seguido una metodología de desarrollo ágil, no se ha llegado a planificar un calendario del proyecto de forma completa, sino que la gestión del tiempo se ha realizado en intervalos a nivel semanal.

Además de tener en cuenta los hitos semanales del proyecto, se ha llevado una planificación más general a nivel mensual, pero de forma más amplia, debido a que el proyecto estaba sujeto a cambios constantes.

Finalmente, las distintas secciones del trabajo (englobando o ignorando ciertas tareas secundarias o que forman parte de una de las secciones mencionadas), son representadas según su duración y la fecha de su realización en esta tabla:

Tabla 1 - Tabla de dedicación de tiempo

Marzo			Abril			Mayo	
12-16 Adopción: aprender conceptos y uso de Salesforce	19-23 Configurar "leads", personalizar página home	26-28 Cargar registros de "leads", configurar objetos de clasificación geográfica	3-6 Configurar oportunidades	9-17 Reasignar registros de cuentas y "leads"	18-23 Configurar y personalizar app móvil	24-30 Diseño y comienzo de desarrollo de plataforma eLearning	11-31
Junio		Julio			Agosto		
1-14 Desarrollo de plataforma eLearning	15-26 Integración con ERP	27-29 Configuración de listas de selección (clasificación geográfica)	2-3 Desarrollo de plataforma eLearning	3-5 Desarrollo de plataforma eLearning	16-31 Implementación y test de plataforma eLearning	1-6 Implementación y test de plataforma eLearning	7-14 Documentación del proyecto

### 3.5. Estructura del equipo

El equipo que ha llevado a cabo la planificación y el desarrollo del proyecto sigue la jerarquía que se puede apreciar en la siguiente figura:

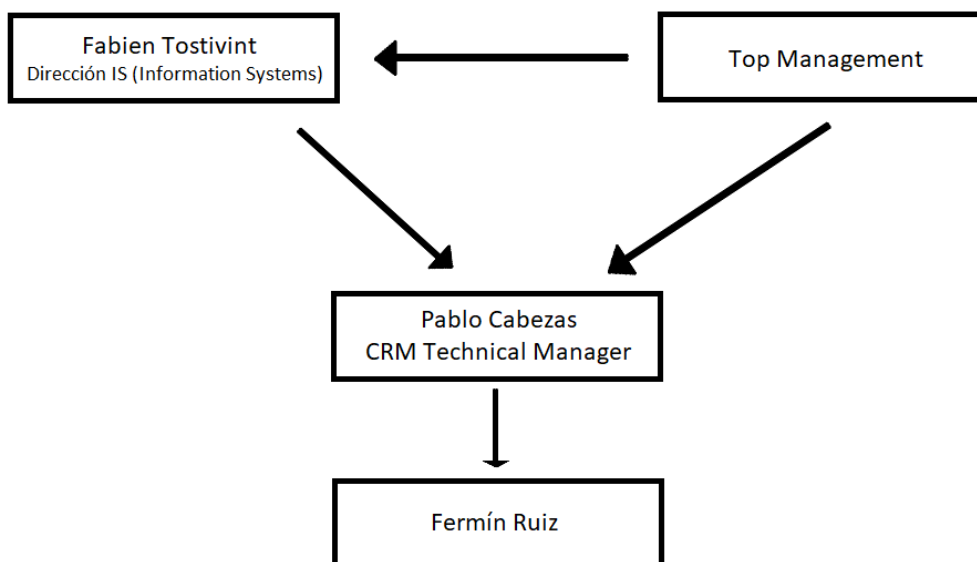


Figura 12 - Estructura del equipo

### 3.6. Proceso de validación

En cada iteración de trabajo según la metodología ágil llevada a cabo, se define una serie de objetivos y requisitos que se han de cumplir. Al final de cada iteración se realiza un proceso de validación por parte del equipo, para comprobar que el trabajo realizado funciona correctamente y que los hitos semanales han sido cumplidos en base a lo establecido.

Para validar el trabajo desarrollado, se realizan una serie de pruebas en el entorno de desarrollo para confirmar la integridad de las nuevas funcionalidades, y después estas se prueban con usuarios reales en distintos casos de uso.

## 4. Evolución de ADEL

### 4.1. Evolución de interfaz: Classic a Lightning

Al principio, ADEL era utilizado con lo que se conoce como interfaz “Classic” de Salesforce. Esta interfaz disponía de todo lo que Olmix Group necesitaba en ese momento, pero con los diferentes cambios que se iban a producir en la lógica de la aplicación y las nuevas funcionalidades con intención de ser añadidas, se decidió migrar a la nueva interfaz de Salesforce, “Lightning Experience” [20].

Esto ha supuesto una serie de cambios y adaptaciones en elementos que ya se encontraban en la aplicación, así como tener en cuenta ciertas consideraciones y la necesidad de añadir nuevas reglas y cambios en roles y perfiles.

#### Habilitar funcionalidades para Lightning

Hay una serie de funcionalidades que son prácticamente esenciales para optimizar la nueva interfaz. Estas son las adiciones más importantes realizadas ante la actualización de interfaz [21]:

- **Archivos relacionados:** añadiendo una lista de archivos relacionada a los registros, los usuarios pueden subir archivos y adjuntarlos a distintos registros.
- **Notas:** al igual que los archivos, los usuarios pueden escribir notas de texto enriquecido y relacionarlas a distintos registros.
- **Path y conversión de “candidatos”:** el “path” es una herramienta muy útil para visualizar los registros según el estado o la etapa en la que se encuentran. El principal objeto en el que se usa el “path” es en el objeto “candidato”, del que también ha sido necesario configurar su conversión. Este tema se profundiza en el apartado sobre “candidatos”.
- **Gestión de duplicados:** para alertar a los usuarios cuando están a punto de crear un registro duplicado, se pueden implementar reglas para la gestión de duplicados.

#### Modificación de permisos

Para controlar quién tenía la capacidad de cambiar a la nueva interfaz de la aplicación, se creó un nuevo conjunto de permisos (“Lightning users”), con el que se puede seleccionar a qué usuarios se les permite acceder a ello de forma individual. A su vez, con otra serie de permisos también se puede elegir si ciertos usuarios pueden cambiar libremente entre la antigua y la nueva interfaz, o una vez se encuentran en “Lightning experience” ya no pueden revertir el cambio y volver a la interfaz clásica.

## Cambios en el código

Uno de los principales cambios que acarrea la migración a la nueva interfaz, es el cambio de framework. Visualforce, el antiguo framework ligado a la interfaz clásica, es sustituido por Aura. Salvo algunas excepciones, los componentes y páginas creados con Visualforce funcionan en “Lightning”, aunque estos componentes mantienen el estilo de la interfaz antigua [22].

También hay casos en los que es necesario comprobar el funcionamiento de los componentes, especialmente si utilizan JavaScript, ya que hay ciertos cambios en las normas de uso y nuevas restricciones.

## Implementación progresiva

Antes de realizar la migración de interfaz a todos los usuarios de la empresa, se decidió probar la interfaz “Lightning” con pequeños grupos de usuarios en distintas partes del mundo. También fue necesario realizar ciertos entrenamientos con estas personas para explicar los cambios en el uso de la aplicación.

Conforme los usuarios utilizaban la nueva interfaz y se sentían cómodos con ella, se fue integrando a más gente en el cambio, y actualmente toda la organización trabaja con la nueva interfaz.

## 4.2. Evolución del modelo de datos de ADEL

Además de los cambios en la interfaz de usuario, han sido necesarias muchas modificaciones en la lógica de negocio de la aplicación para implementar todas las nuevas funciones que son requeridas por la empresa, como la creación, modificación e implementación de nuevos objetos o la configuración de distintos procesos de negocio.

### Configuración e implementación del objeto “leads”

Al comienzo del proyecto, la aplicación ADEL utilizaba dos objetos para almacenar la información de los clientes de Olmix Group. Por un lado, estaba el objeto “cuentas” que guarda los datos de empresas y organizaciones con las que se quiere mantener un seguimiento, como clientes, distribuidores, usuarios finales o competidores. Por otro lado, el objeto “contactos” guarda la información de los trabajadores y los individuos relacionados con esas empresas.

Estos dos objetos son útiles para guardar la información relacionada con los clientes, pero era necesario para la empresa tener un nuevo objeto donde almacenar los datos relativos a clientes potenciales, y para ello se ha implementado el objeto “candidatos”.

Los candidatos (en inglés *leads*) son prospectos (personas y compañías que se han identificado como clientes potenciales). Además de los datos de la persona con la que se tiene contacto y la empresa para la que trabaja, se debe almacenar otra información de importancia, como el origen de esa información, la probabilidad de efectuar una venta o el dinero estimado que se podría obtener haciendo negocios con ese candidato.

Así, el objeto “*leads*” ha sido configurado con estos campos:

- Información del candidato:
  - Compañía
  - Nombre
  - Título
  - Web
  - Estado del candidato (nivel de cualificación del candidato)
  - Valoración
  - Tipo de candidato
  - Observaciones
- Información de contacto:
  - Correo electrónico
  - Fax
  - Móvil
  - Teléfono
- Clasificación geográfica:
  - Dirección
  - País, calle, código postal, ciudad, estado o provincia
- Segmentación (divisiones de Olmix Group interesadas en el candidato)

Además de todos los campos configurados para almacenar información del *lead*, este objeto tiene una característica adicional: el “*Lead Path*”. El “*Lead Path*” (figura 13) utiliza el campo “estado del candidato” para representar de forma gráfica qué nivel de cualificación tienen los distintos candidatos, y que los usuarios puedan actuar conforme a ello.



Figura 13 - Lead Path

También, al actualizar la interfaz de “Classic” a “Lightning” y configurar el “*Lead Path*”, se habilita la opción de visualizar los registros en la vista Kanban (figura 14), con la que se pueden visualizar todos los registros organizados según su etapa.

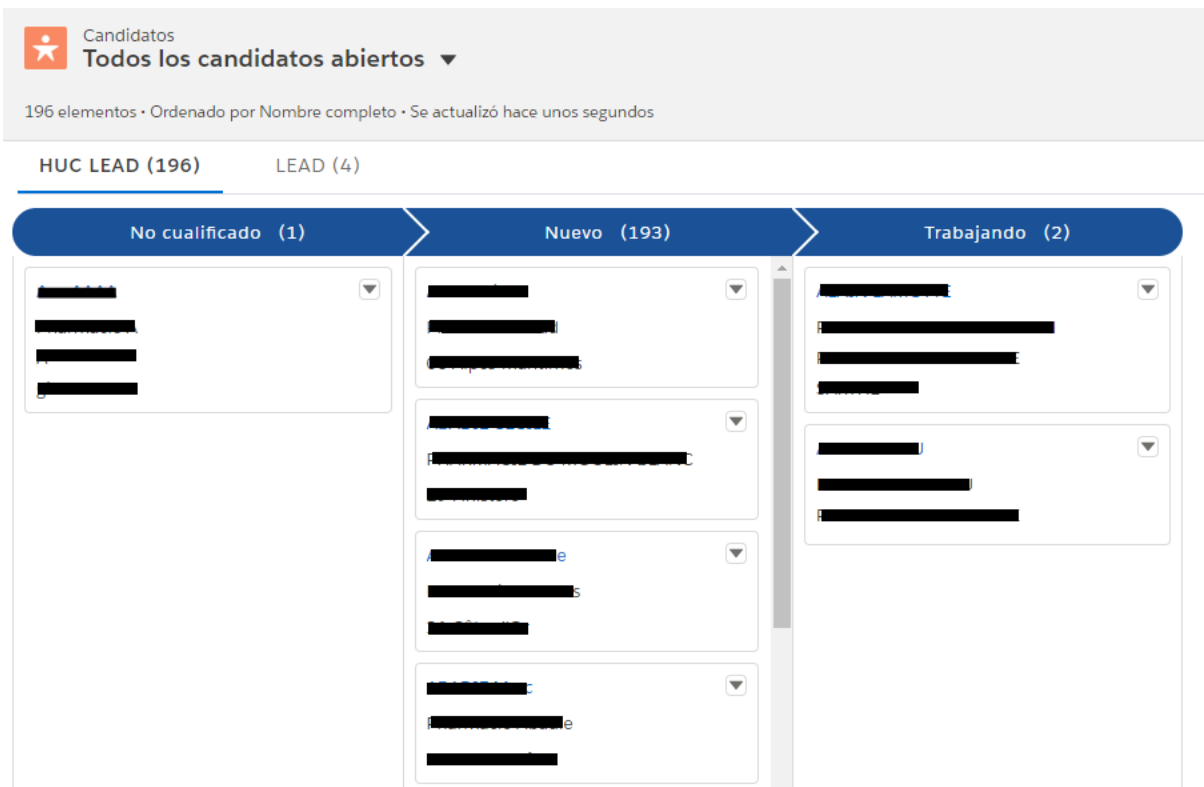


Figura 14 - Vista Kanban

El objetivo final de guardar la información de los clientes potenciales es poder trabajar con ellos para generar una oportunidad de venta, y convertirse así el candidato en un cliente (una cuenta). Por ello, también se ha configurado un mapeo de los campos del objeto “leads”, para que una vez se haya recopilado toda la información pertinente de él y se genere una oportunidad de venta, el registro de candidato se pueda convertir en una combinación de cuenta (empresa relacionada al candidato) y contacto (el candidato con el que has mantenido contacto), como se puede ver en la figura 15:



Convertir candidato

**Cuenta**  Crear nueva

\* Nombre de la cuenta

Tipo de registro

- OR -  Seleccionar existente

Búsqueda de cuenta

---

**Contacto**  Crear nueva

Sr. \*\*\*\*\*

- OR -  Seleccionar existente

Regla de duplicados no activa

---

**Oportunidad**  Crear nueva

Pharmacie A-

No crear una oportunidad tras la conversión

- OR -  Seleccionar existente

Para buscar una oportunidad, seleccione una cuenta existente

\* Propietario de registro

\* Estado convertido

*Figura 15 - Conversión de lead*

### Configuración de oportunidades

Con los objetos “cuentas”, “contactos” y “leads” es posible almacenar y manejar la información sobre clientes y prospectos, pero para aprovechar al máximo el potencial del CRM también se deberían monitorizar las ventas realizadas desde que se genera esa posibilidad de vender. Para ello se ha configurado el objeto “oportunidades”.

Básicamente, las oportunidades son negociaciones en progreso. Estas negociaciones normalmente tienen un progreso, desde un primer tanteo hasta firmar el acuerdo. Conforme el proceso de la negociación va progresando, la probabilidad de éxito de la negociación varía y la oportunidad de venta se encuentra en distintas etapas.

En ADEL han sido configuradas las siguientes etapas de negociación, con sus respectivas probabilidades aproximadas de venta:

- Cualificación (10%)
- Análisis (35%)
- Propuesta (75%)
- Negociación (90%)
- Cerrado Ganado (100%)
- Cerrado Perdido (0%)
- Cancelado (0%)

Al igual que con el objeto “leads”, el objeto “oportunidades” utiliza el campo de la etapa de negociación para formar un “path” (figura 16).



Figura 16 - "Path" de oportunidades

A su vez, esto también permite activar la vista Kanban para visualizar de forma más gráfica los distintos registros de oportunidades según su estado.

Además de los campos “etapa” y “probabilidad” explicados arriba, el objeto “oportunidades” dispone de los siguientes campos:

- Información de la oportunidad:
  - Nombre
  - IC / EXT (Cliente externo o compañía interna)
  - Nombre de la cuenta (campo relacionado con el registro del objeto “cuenta” con el que se negocia)
  - Dirección
  - Fecha de cierre
  - Importe
  - (Otros)
  
- Información Olmix Group
  - Sociedad financiera
  - Sociedad padre

A pesar de que el objeto ha sido configurado e implementado para su uso, no está siendo utilizado en su totalidad por los usuarios de ADEL, ya que se ha preferido adaptar a estos de forma gradual a las nuevas funcionalidades de la aplicación. En su lugar, están utilizando un tipo de registro alternativo para el mismo objeto, en el que solamente almacenan la información de negociaciones ya cerradas.

## Automatización de procesos

Para la automatización de algunas acciones, como crear registros o rellenar o editar campos en ciertas condiciones, se han utilizado las herramientas Process Builder y Cloud Flow Designer. Process Builder se utiliza para configurar la lógica de un proceso que ejecuta uno o más flujos (creados con Cloud Flow Designer) cuando se cumplan las condiciones establecidas. Los flujos se encargan de realizar las modificaciones de registro pertinentes.

En este trabajo, la automatización de procesos se ha utilizado, por ejemplo, para otorgar permisos de acceso a los usuarios que están relacionados con cada registro del objeto “Sales Report Figures” en la integración con el ERP. También se ha programado un proceso que asigna automáticamente al propietario de un registro de “lead” cuando es creado con la aplicación *Business Card Reader*, ya que esta aplicación asigna automáticamente el registro al administrador asociado a la clave corporativa de la aplicación.

### 4.3. Clasificación geográfica

Por parte de Olmix Group, hay un gran interés en almacenar la información geográfica de las distintas cuentas y contactos, de candidatos y de las ventas, para poder visualizar los datos clasificados por regiones y países, para poder actuar distintivamente y planificar distintas campañas según los intereses en cada país.

En un principio, se configuraron dos nuevos objetos para cumplir con el objetivo: el objeto “Country” (país), el objeto “Country Region” (región de país) y el objeto “Region” (región). Además de estos objetos que permitían dividir los registros según su localización política, se crearon tres objetos para segregar la información según su localización física: los objetos “Area”, “Territory” y “Continent”.

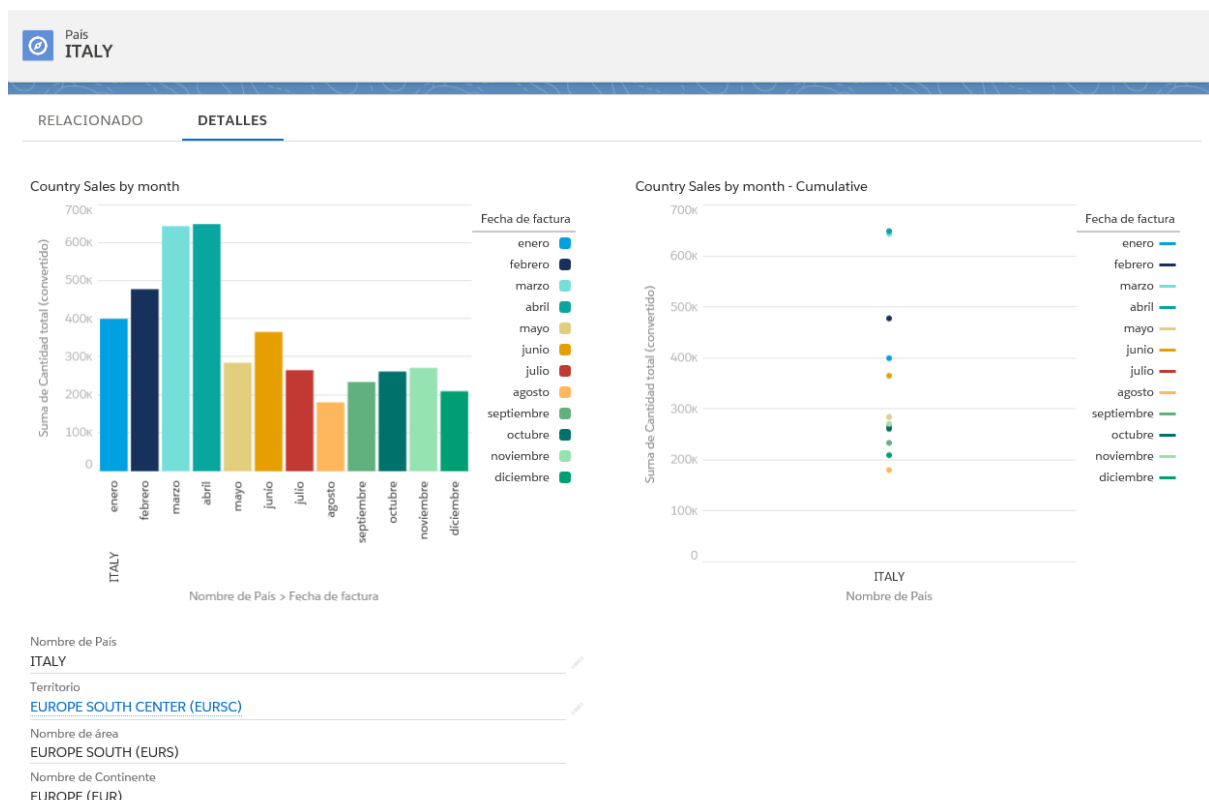


Figura 17 - Detalles del país ITALY

En cada uno de estos objetos se guardan muy pocos campos de información. Por ejemplo, en el objeto “Country” sólo se almacena el nombre del país, y el territorio, área y continente al que pertenece (figura 17), y el objeto “Country Region” sólo tiene como campos el nombre, y el país y región a la que pertenece (también tiene un campo llamado “SAGE ID” que sirve para la integración con el ERP). De esta forma, teniendo el objeto “Country Region” relacionado con el objeto “Country” y el objeto “Region”, y a su vez el objeto “Country” relacionado con el resto de los objetos, creamos una red que conecta todos los objetos (figura 18).

**Cuentas (6+)**

Nombre	Tipo de regi...	Contable	Teléfono
ABATE	Usuario final	<input type="checkbox"/>	06/99271111
AGRICOLA TRE VALLI	Distribuidor	<input checked="" type="checkbox"/>	
AGRI SI	Distribuidor	<input checked="" type="checkbox"/>	
ALBORS SRL	Dealer	<input checked="" type="checkbox"/>	
AGRI-BONINO SRL	Distribuidor	<input checked="" type="checkbox"/>	
AMINOVIT SRL	Distribuidor	<input checked="" type="checkbox"/>	

**Regiones de país (6+)**

NOMBRE DE REGIÓN DE PAÍS	REGIÓN
<a href="#">Agrigento</a>	Italy - Sicilia
<a href="#">Alessandria</a>	Italy - Piemonte
<a href="#">Ancona</a>	Italy - Marche
<a href="#">Aosta</a>	Italy - Valle d'Aosta
<a href="#">Arezzo</a>	Italy - Toscana
<a href="#">Ascoli Piceno</a>	Italy - Marche

Figura 18 - Cuentas y regiones relacionadas con el país ITALY

Otra de las ventajas que tiene utilizar esta forma de guardar estos datos, es que los nombres de los distintos países, regiones y provincias y las relaciones entre ellos son introducidos una sola vez por los desarrolladores, y después los usuarios sólo tienen que buscar la región y el país concretos en la lista ya creada. Al trabajar de esta forma, se evita uno de los mayores problemas que se dan en la introducción de datos: los errores tipográficos.

A modo de ejemplo, si tenemos una cuenta en Florencia, su información geográfica se guardaría así:

- Country Region: Florence
- Region: Toscana
- Country: Italy
- Territory: Europe South Center

- Area: Europe South
- Continent: Europe

Como se puede observar, este nivel de segmentación permite una buena distribución de la información de forma geográfica, aunque puede resultar engorroso por parte de los usuarios introducir la información de estos campos. Además de los campos “Country” y “Country Region”, los usuarios debían introducir la dirección de correo y facturación de las empresas, en la que tenían que incluir nuevamente la información de provincia y país.

Para evitar esto, se ha configurado e implementado una solución disponible en Salesforce y que se puede combinar con la solución ya implementada: las listas de selección de estado y país/territorio.

Con estas listas de selección, se cambian las entradas de provincia y país en los campos de dirección de correo y dirección de facturación, para que, en lugar de introducirlos a mano, el usuario los elija a partir de una lista creada por los administradores. A su vez, estas entradas están vinculadas a los registros de los objetos “Country Region”, “Region”, “Country”, “Territory”, “Area” y “Continent” ya configurados, por lo que con la nueva solución no se pierde el nivel de segmentación conseguido.

Para configurar las listas de selección de estado y país/territorio, ha sido necesario realizar los siguientes pasos:

1. Configurar las listas de selección de estado y país/territorio: se especifican todos los países y provincias/estados con los que se espera que la empresa vaya a tener negocios, incluyendo en cada uno sus códigos ISO 3166-1 y 3166-2 (estándar internacional que proporciona códigos únicos para todos los países y provincias/estados, respectivamente).
2. Explorar la organización: encontrar todos los datos de estado y país a lo largo de ADEL, así como referencias a los campos de estado y país en informes, vistas de lista, reglas de validación y otras personalizaciones.
3. Convertir los datos: se asignan los datos de estado y país existentes en los registros a valores de lista de selección estandarizados. Primero asignan los valores de país/territorio y luego los valores de estado.
4. Activar las listas de selección de estado y país/territorio

#### 4.4. Aplicación móvil

La aplicación Salesforce es una app para dispositivos móviles “enterprise-class” (escalable y personalizable) que permite a los usuarios acceso inmediato a los datos de ADEL desde un teléfono móvil o una tablet. El uso de esta aplicación viene con la licencia de Salesforce, está disponible tanto en iOS como en Android y puede ser utilizada para ciertas funciones sin conexión a internet.

A pesar de tener acceso a muchas funcionalidades nada más instalarla, en su estado por defecto no se aprovecha todo su potencial. Por ello, hay que implementar una serie de cambios para que esta se convierta en una aplicación profesional.

## Instalación de calendario

Para hacer que los usuarios puedan acceder al mayor número de información de forma inmediata, y teniendo en cuenta que el uso del dispositivo móvil se dará cubriendo alguna actividad, como una tarea o un evento, se decidió que la mejor solución era implementar un calendario en la aplicación que mostrara todas las actividades de cada día.

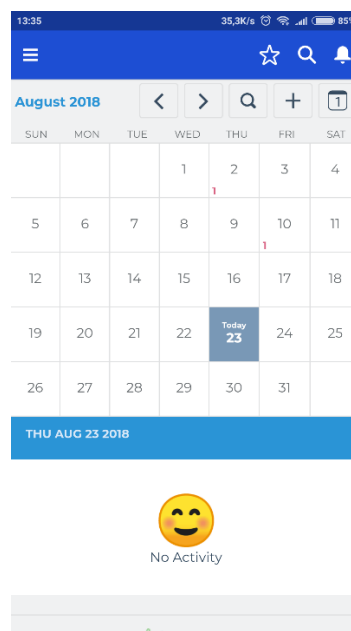


Figura 19 - Calendario MobilePlus

Tras explorar AppExchange, se encontró la aplicación MobilePlus (figura 19), que cubre todas las necesidades y además es gratuita. Tras instalarla y probar su funcionamiento, se configuró la estructura de la aplicación para que el nuevo calendario aparezca como página principal al abrir la aplicación.

## Menú de navegación

El menú de navegación (figura 20) es una barra lateral desplegable que permite acceder con rapidez a diferentes listas de objetos y secciones de la aplicación móvil. Para el uso de la aplicación previsto, se ha dividido el menú de navegación en dos secciones:

- La primera sección contiene los elementos principales de la aplicación, que siempre aparecerán en la parte superior del menú y son los elementos más utilizados desde el móvil por la organización. Estos son el calendario MobilePlus, Chatter, tareas, eventos, paneles y la aplicación de eLearning.
- La segunda sección se compone de los elementos de búsqueda inteligente. Se muestra un conjunto de objetos a los que el usuario accedió recientemente, y tiene un botón para ampliar la lista y mostrar todas las secciones de la aplicación a las que el usuario tiene acceso.

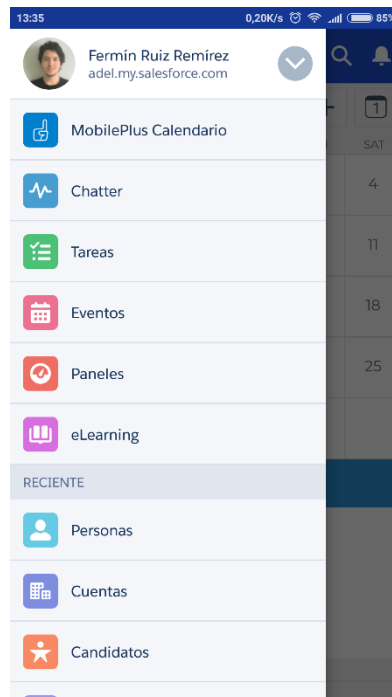


Figura 20 - Menú de navegación

## Acciones globales

Las acciones globales (figura 21) son accesos directos a las acciones más utilizadas de ADEL, simplificadas para ser utilizadas con comodidad desde un dispositivo móvil.

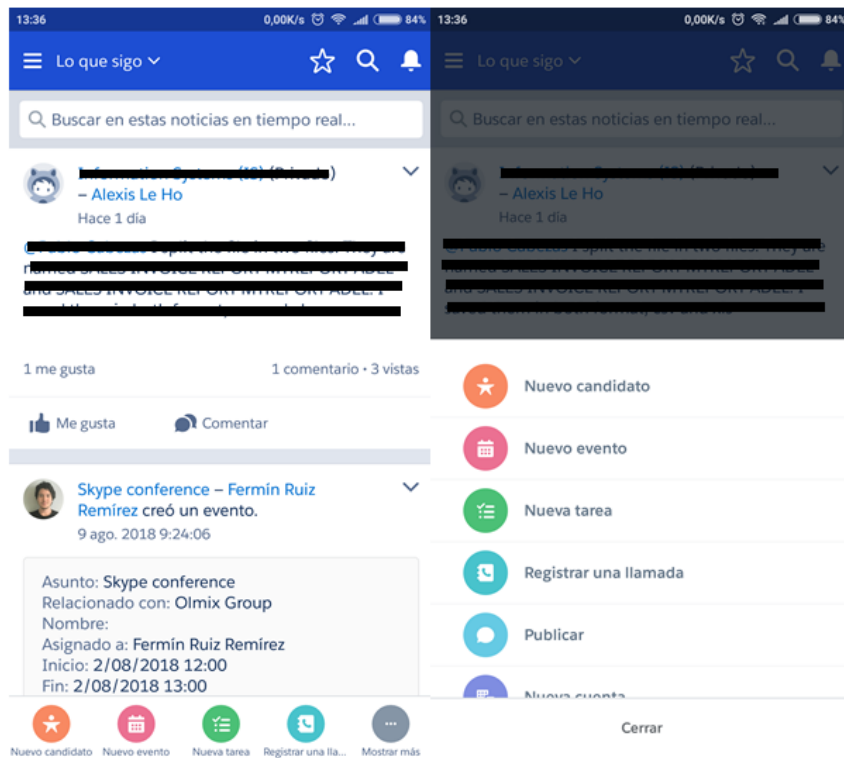


Figura 21 - Acciones globales de la aplicación

En concreto, las acciones globales implementadas en la aplicación son: nuevo candidato, nuevo evento, nueva tarea, registrar una llamada, publicar en Chatter, nueva cuenta, nuevo contacto, subir un archivo y nueva nota.

Las propias acciones globales también están personalizadas para ser más simples, ya que se supone que el uso de la aplicación móvil va a ser más dinámico que en escritorio. Por ejemplo, la acción global “nuevo candidato” sólo muestra los campos nombre, correo electrónico, teléfono, compañía y título, ocultando el resto de los campos que pueden ser añadidos después desde el escritorio.

#### Acciones rápidas específicas de objetos y formatos compactos

Las acciones específicas de objetos (figura 22) son muy similares a las acciones globales, solo que estas se configuran para los diseños de página de cada objeto, y se añaden, editan o eliminan del formato de página desde el gestor de objetos.



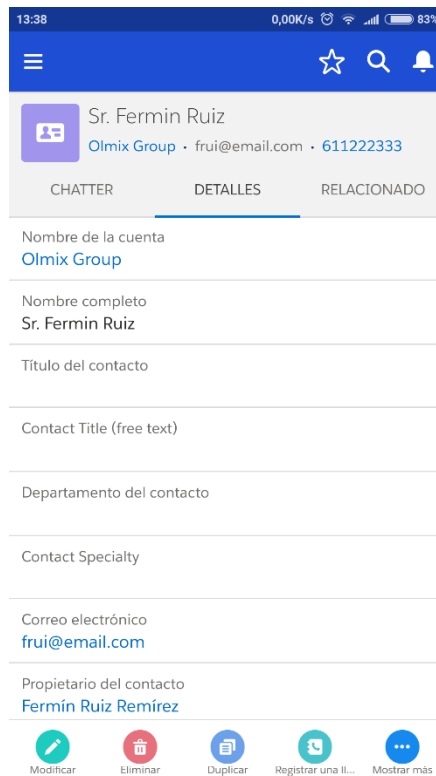


Figura 22 - Acciones específicas del objeto contacto

También, las acciones que crean un registro (nueva tarea, nuevo evento) se relacionan automáticamente con el registro desde el que se ha accedido a la acción.

Los formatos compactos (figura 23) controlan los campos que se destacan en el encabezado de un registro cuando lo visualizas en la aplicación móvil. Para cada objeto, el encabezado se puede personalizar con hasta un máximo de cuatro campos, y el uso de formatos compactos ayuda a obtener la información más útil con rapidez, sin necesidad de buscarla a lo largo de todos los campos del registro. En la siguiente figura se puede observar el formato compacto para el objeto contacto:

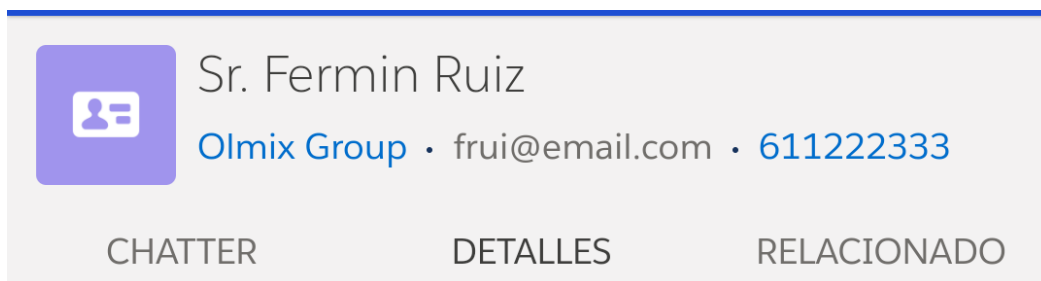


Figura 23 - Formato compacto del objeto contacto

## 4.5. Reporting

*Reporting* es la acción de realizar informes o reportes a partir de la información almacenada. En ADEL, se dispone de una potente herramienta de *reporting* para generar informes en base a los distintos objetos de la aplicación.

Para poder otorgar a los equipos de ventas cifras sobre estas y una representación gráfica de su progreso, se han configurado una serie de informes en base a un nuevo objeto desarrollado expresamente para ello. Este objeto se llama “*Sales Report Figures*” y es el objeto utilizado para importar en ADEL las cifras de ventas que se encuentran en el ERP.

Cada registro de este objeto guarda información de una orden de venta en distintos campos, como, por ejemplo: cliente, manager local, regional, de área y de territorio, fecha de factura, familia, división y línea de actividad del producto, precio en euros, cantidad en toneladas, etc.

Para toda esta información se han configurado una serie de informes que son actualizados automáticamente conforme se añaden nuevos registros, y el usuario los visualiza dinámicamente, es decir, los datos de los informes varían dependiendo de a qué sección de los registros tiene acceso el usuario teniendo en cuenta su perfil y su rol.

Los informes, además de tener la posibilidad de ser representados en gráficas, se pueden agrupar en un elemento de ADEL llamado panel. Los paneles (figura 24) contienen gráficas de distintos informes y son mostrados en una sola página al usuario. Con los informes generados a partir de los datos almacenados en el nuevo objeto, se ha compuesto un panel que muestra de forma visual todos los datos pertinentes para el usuario.

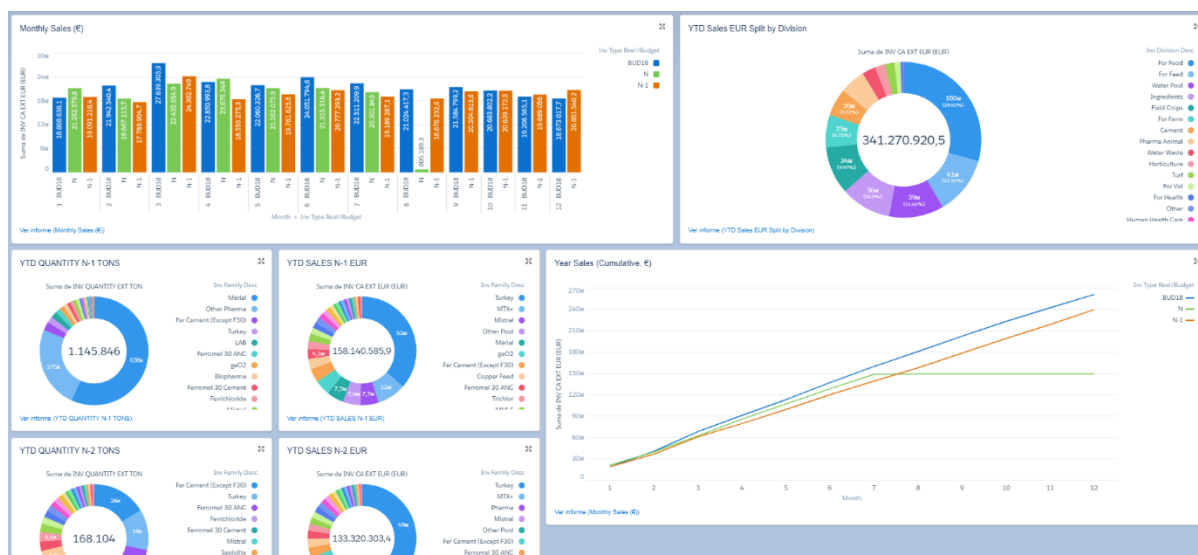


Figura 24 - Panel de ventas

## 4.6. Integración ADEL-ERP

Una de las principales necesidades que se debían cubrir en este proyecto era la integración de los datos recogidos en el ERP dentro de la aplicación ADEL. La empresa Olmix Group utiliza un ERP para almacenar y gestionar información de distintas operaciones de la empresa, como la gestión de recursos humanos, inventario, órdenes de venta, etc.

En concreto, Olmix Group utiliza el ERP Sage Business Cloud Enterprise Management, antes llamado Sage X3. Se dispone de la versión del ERP “*On premise*”, es decir, el ERP no se encuentra en la nube, sino que está instalado y se ejecuta dentro de la organización.

### Talend

La primera opción contemplada para la integración de la información del ERP fue utilizar el programa Talend. Los componentes de Talend permiten realizar una serie de manipulaciones en la información que hacen a esta herramienta ideal para la integración.

El primer requerimiento fue una forma de obtener la información del ERP. Para no comprometer la información de este, se utiliza la información del servidor de MyReport, que obtiene la información del ERP además de otras fuentes de datos (ficheros Excel, texto, etc.), y genera un cubo de datos del que se obtiene la información.

Para obtener la parte de información necesaria del servidor de MyReport (en concreto, la información de las ventas de Olmix Group), se exportan los campos pertinentes de este en un fichero Excel que contiene todas las líneas con cifras de ventas de la empresa.

La información almacenada en el servidor de MyReport cambia todas las noches, por lo que es necesario exportar el fichero Excel a diario. Por ello, se ha configurado el servidor para que genere este archivo todos los días, y automáticamente es subido a una carpeta de Dropbox.

Una vez tenemos la fuente de información configurada, se debe realizar el proceso de transformación de la información, y para ello se utiliza la herramienta Talend. En Talend, se ha creado un trabajo (*job*) con una serie de componentes, visible en la figura 25:

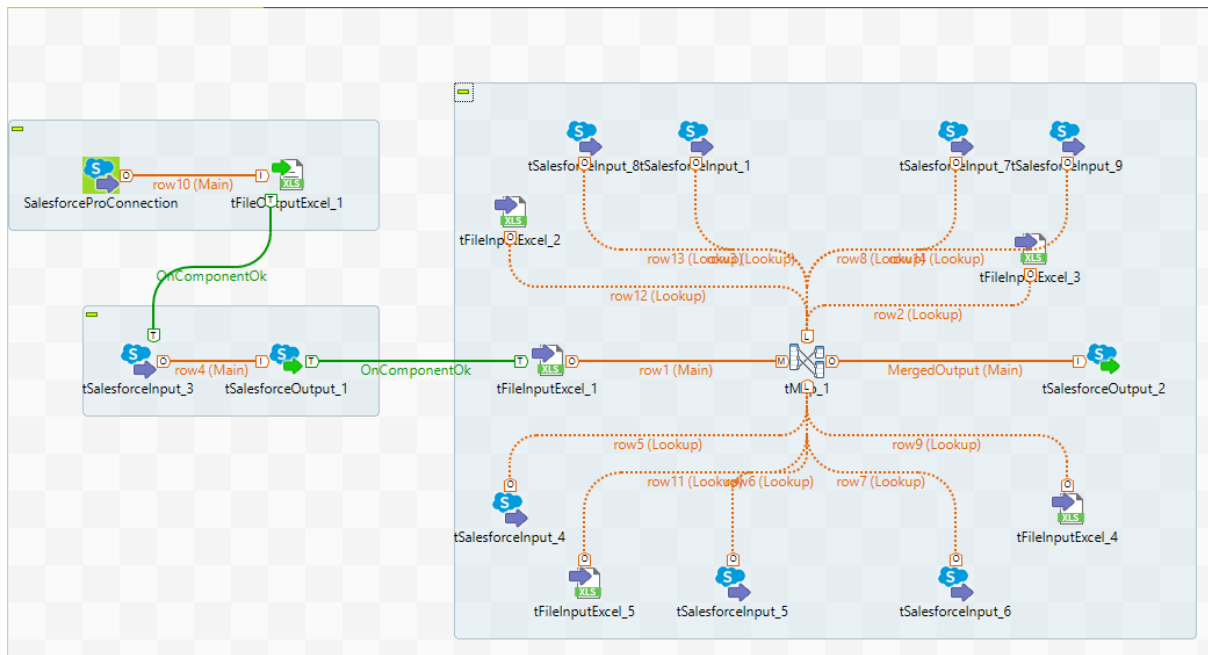


Figura 25 - Configuración de componentes en Talend

Como se puede observar, hay una serie de flujos y componentes agrupados en subtrabajos, y estos se ejecutan de forma secuencial. El primero de los subtrabajos se encarga de la obtención de todos los registros de cuenta disponibles en ADEL y los almacena en un fichero Excel auxiliar que será utilizado más adelante. Después, se procede a eliminar todos los registros del objeto “Sales Reports Figure” en ADEL, ya que la información obtenida del ERP se destruye y regenera a diario.

Una vez eliminados todos los registros, se da paso a la extracción y transformación de los nuevos datos. Primero, se extraen los registros del archivo Excel exportado por el servidor de MyReport, que se encuentra en Dropbox. Después, se obtienen los registros de todos los objetos que están relacionados con los campos del objeto “Sales Report Figure”, como cuentas (cuyos registros se han obtenido en el primer paso), usuarios, actividades, divisiones, y familias de productos.

Cuando ya se tienen todos los registros necesarios, se da paso al mapeado de la información a campos del objeto de destino, como se ve en la ilustración:

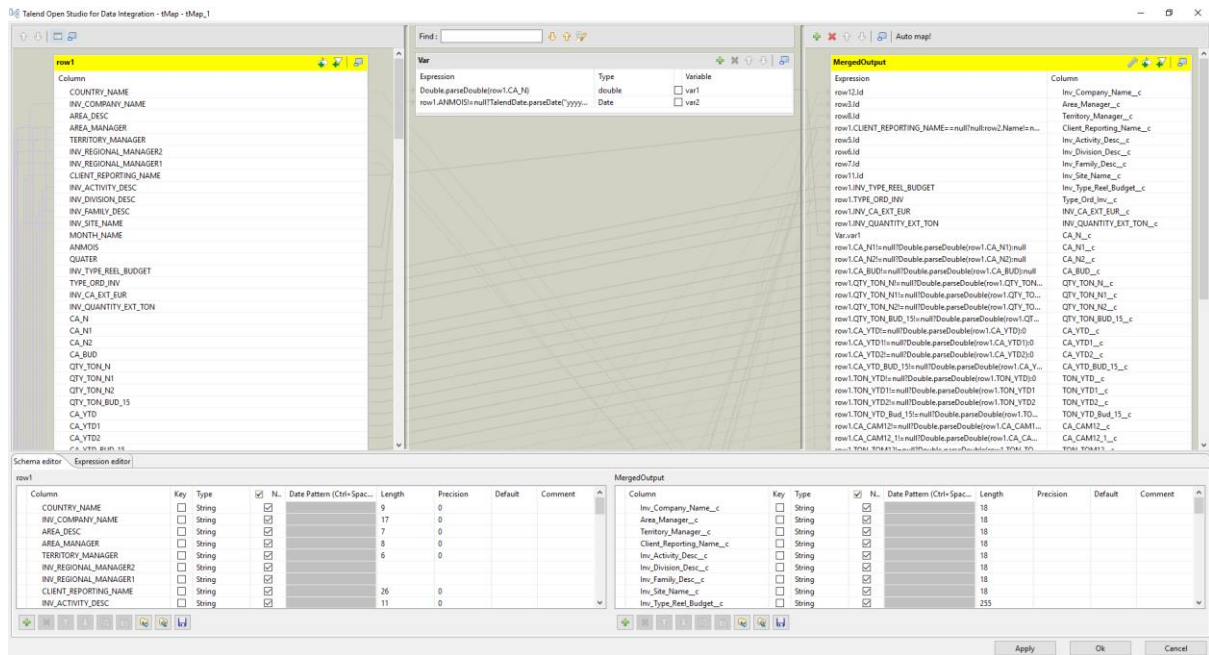


Figura 26 - Mapeado de campos en talend

Este mapeado se produce de la siguiente manera:

1. Para los campos de texto, se relaciona directamente la columna del Excel correspondiente al campo destino.
2. Para los campos de número y de fecha, se debe aplicar una función o expresión de parseo para que la información tenga el formato correcto.
3. Para los campos que son relaciones directas con otros objetos (como cuentas o usuarios), se relaciona el nombre del registro obtenido en el Excel de entrada con el campo “nombre” del objeto en cuestión, y este se utiliza como clave para indexar los registros y obtener el identificador concreto del registro. Este identificador será la información almacenada en el campo del objeto destino.

Por último, una vez realizado el mapeo, se procede a transformar la información del fichero Excel en bloques de 200 registros (de acuerdo con la API masiva de Salesforce), y se importan estos bloques de registros en la aplicación ADEL.

Como este es un proceso que debe realizarse a diario, era conveniente buscar una forma de automatizar el proceso para no tener que ejecutar el trabajo manualmente todos los días. La solución propuesta consiste en exportar el trabajo de Talend como un ejecutable de Java, y se programó la automatización del proceso en un servidor, con el programador de tareas de Windows, para que el trabajo se ejecute todos los días a las 7:00 de la mañana.



## 5. Herramienta eLearning

### 5.1. Diseño

En la etapa de diseño, se ha evaluado cómo se estructuraría la información de los distintos cursos, así como la estructura y el diseño de página, tanto en la aplicación móvil como en escritorio.

Por un lado, respecto a la estructura de los cursos, se ha pensado en un modelo similar al utilizado por otras plataformas, que divide la información en temas más generales, y a su vez en unidades más concretas.

Así, se ha pensado en la estructura de la figura 27:

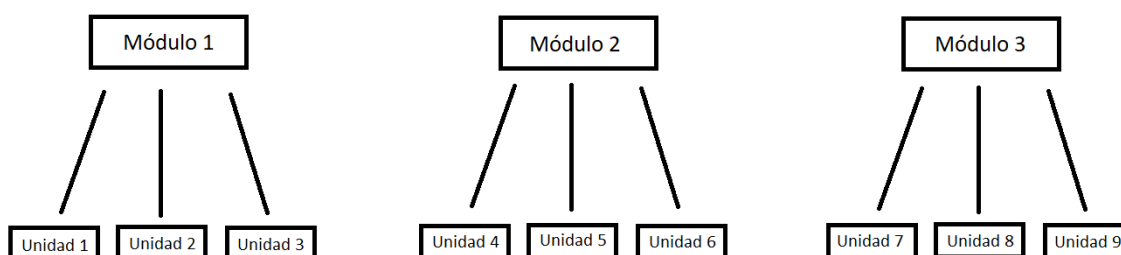


Figura 27 - Estructura módulos/unidades

El elemento más genérico es el módulo. Cada módulo trata sobre un tema específico, y esa información se presenta en unidades. Las unidades son elementos de tamaño más reducido en los que se muestra una sección concreta del contenido relativo al módulo.

### Diseño multiplataforma

En cuanto al diseño de la aplicación, el objetivo era crear una herramienta dinámica y que se pueda utilizar en una página única, en la que no se necesite refrescar el navegador o cargar nuevas páginas web. También había que tener en cuenta desde un principio que desde un teléfono móvil o desde una tablet se disponga de la máxima maniobrabilidad y funcionalidad posibles, ya que asegurado esto en pantallas pequeñas, el escalado es más sencillo.

Al utilizar componentes Lightning que se pueden comunicar por eventos, se puede jugar con la visibilidad de estos para conseguir el efecto de cambiar de página sin necesidad de refrescar la vista.

En la figura 28 se puede observar la estructura de los componentes en móvil/Tablet y en escritorio, respectivamente:

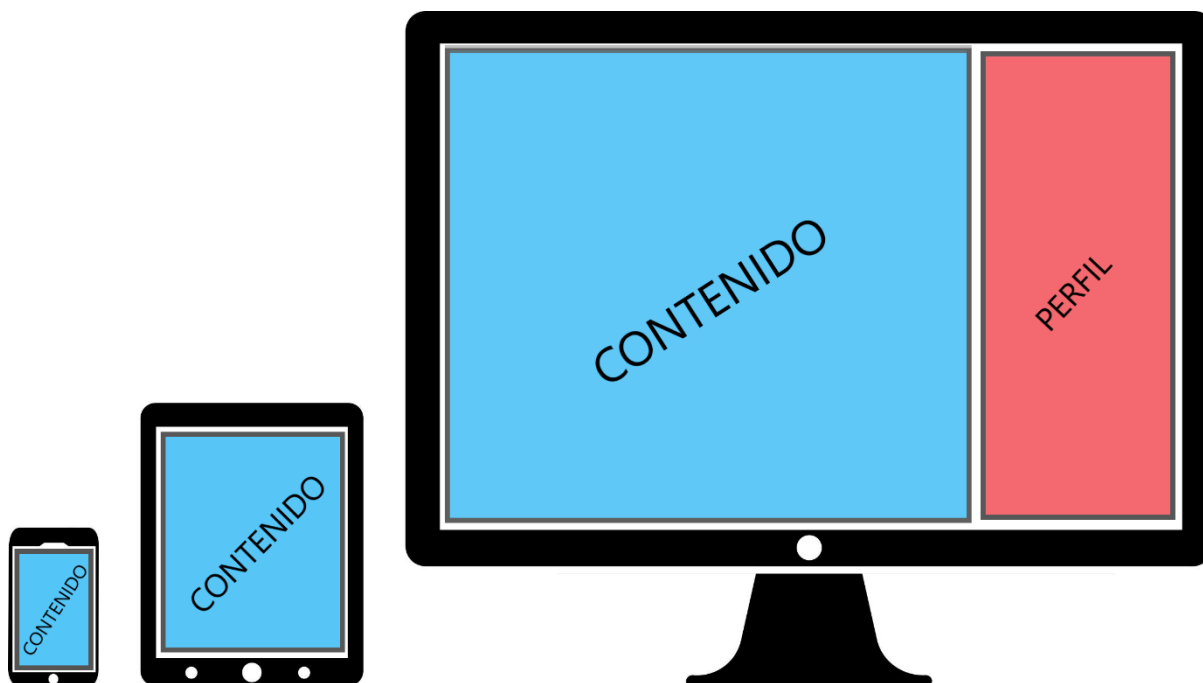


Figura 28 - Diseño multiplataforma

La principal diferencia entre el diseño para móvil y el diseño para escritorio es la incorporación de una barra lateral en el escritorio, en el que aparece el perfil del usuario y los módulos que tiene en progreso. Como la información del perfil es trivial y los módulos en progreso aparecen en el componente principal, no se compromete el funcionamiento de la plataforma.

## 5.2. Lógica de negocio

Aprovechando que la herramienta eLearning se va a implementar integrada en ADEL, se tiene la posibilidad de utilizar los objetos de Salesforce a través de las APIs de Apex para configurar los elementos de información de la plataforma. Por ello, se han creado los dos objetos fundamentales de la aplicación: el objeto “módulos” y el objeto “unidades”.

Como el verdadero contenido de aprendizaje del módulo está almacenado en las unidades, el objeto “módulos” no requiere almacenar mucha información de texto, sino las unidades que están relacionadas con él, además de un título, una descripción y una imagen que identificarán al módulo. En las unidades se guarda todo el contenido de aprendizaje, además de tener la opción de tener un videotutorial adjunto a la unidad.



## Módulos y unidades de usuario

Además de tener en cuenta la estructura y el contenido de los módulos y las unidades, deben quedar registradas, para cada usuario, todas las instancias de completitud de las unidades y los módulos. Para ello se han creado los objetos “módulos de usuario” y “unidades de usuario”, que tienen pocos campos a rellenar y son de gran sencillez (figura 29).

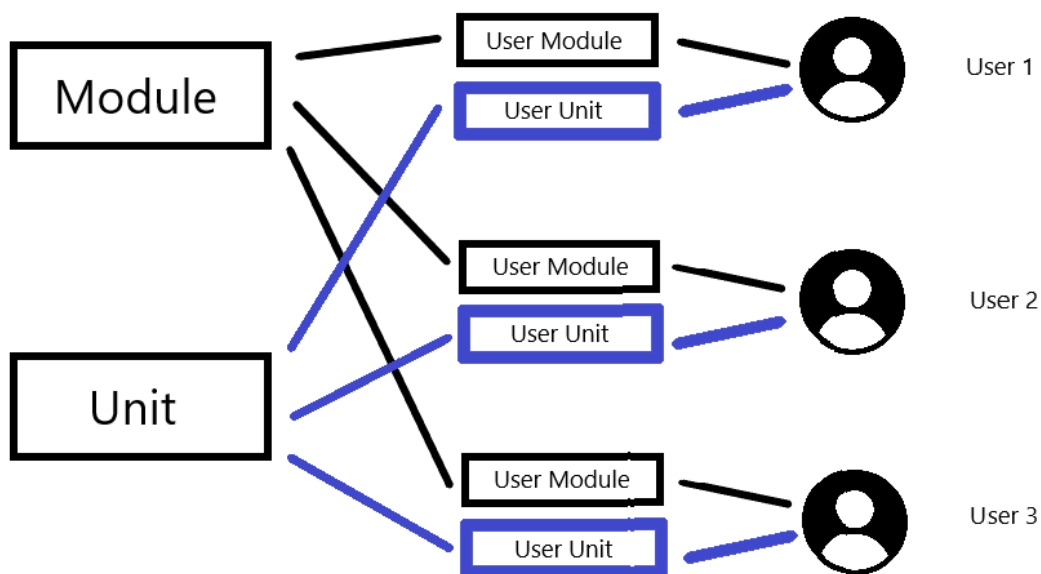


Figura 29 - Módulos y unidades de usuario

El objeto “unidades de usuario” tiene tres campos: el campo “unidad”, que relaciona al registro con el registro del objeto “unidades” correspondiente, el campo “usuario” que es asociado a un usuario, y el campo “completado” que se corresponde con un booleano que será marcado como “true” si el usuario completa la unidad en concreto.

El objeto “módulos de usuario” es similar al de “unidades de usuario”, sólo que el campo “unidad” cambia por el campo “módulo”, y el booleano “completado” pasa a ser una lista de selección llamada “etapa”, que indica si el módulo está “iniciado” (el usuario no ha completado ninguna unidad), “en progreso” (el usuario ha completado alguna unidad del módulo) o “completado” (el usuario ha realizado todas las unidades de ese módulo).

## Multiidioma

Olmix Group es una empresa que tiene empleados a lo largo de todo el planeta, por lo que no todo el mundo habla el mismo idioma. Para lograr dar accesibilidad a la plataforma en

múltiples idiomas, se han creado y configurado dos nuevos objetos que nos dan la posibilidad de mostrar el contenido de la herramienta en el idioma deseado (figura 30).

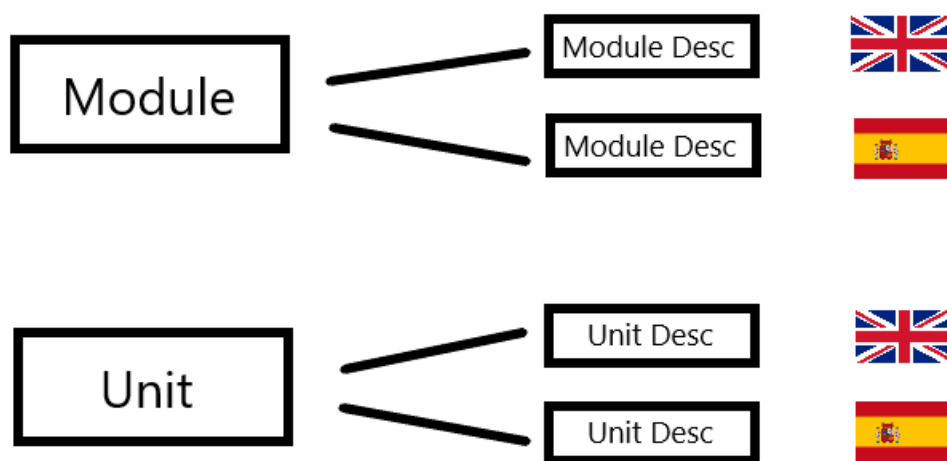


Figura 30 - Descripciones de módulo y unidad

Por un lado, tenemos el objeto “descripción de módulo”. En él se encuentran los campos “módulo” e “idioma”, que relacionan al registro de este objeto con el módulo al que corresponde e indican el idioma en el que se está guardando la información, respectivamente. En este objeto también se guarda el título y la descripción del módulo en cada idioma, a un registro de objeto por idioma.

El objeto “descripción de unidad” se comporta exactamente igual, y en cada registro se guarda el título, la descripción y el contenido de cada unidad en el idioma indicado. Además, el campo en el que se referencia al enlace al videotutorial se encuentra en este objeto, por lo que para una misma unidad se puede almacenar hasta un vídeo por idioma.

Por otro lado, al objeto “módulos” se le han añadido dos campos, el campo “idioma por defecto” y el campo “idiomas disponibles”. En el campo “idiomas disponibles” se despliega una lista de múltiple selección, en la que se deben elegir todos los idiomas para los que ha sido preparado el módulo de entre los idiomas disponibles en ADEL. En el campo “idioma por defecto” se indica uno de los idiomas disponibles para ese módulo, y si un usuario trabaja en ADEL con un idioma que no está disponible para ese módulo, el contenido de este se le mostrará en el idioma elegido.

Actualmente, la aplicación soporta francés, inglés, chino y español, y en caso de nuevas adiciones de idiomas en ADEL, el cambio de la lista de idiomas es muy sencillo y no compromete el uso de la plataforma.

## Multiactividad

Aunque el principal propósito de realizar la herramienta eLearning era proporcionar a los usuarios de ADEL la información de uso necesaria de la aplicación, la posibilidad de preparar la plataforma para aceptar cursos de otras actividades era algo muy conveniente.

Por ello, se ha configurado el campo “tema” en el objeto “módulos”, en el que se puede elegir el tópico del que tratará el módulo. Actualmente está configurado para poder elegir entre ADEL o cualquiera de las líneas de actividad de la empresa (Animal Care, Extra Care, Human Care y Plant Care). Además, la estructura de información de módulos y unidades hace que sea muy sencillo preparar cursos para los empleados que no forman parte del servicio informático. También se ha añadido el campo “nivel” al objeto “módulo”, en el que se puede indicar la dificultad de este.

## Estructura de la lógica de negocio

Tras implementar todas las configuraciones, la lógica de negocio quedaría estructurada tal y como se puede observar en la figura 31:

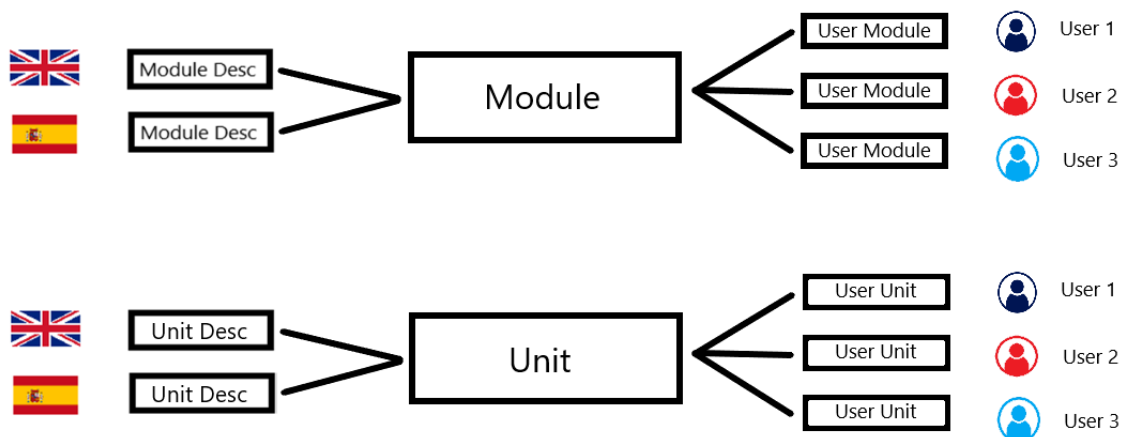


Figura 31 - Estructura de objetos completa

## 5.3. Lado del cliente

Para desarrollar el aspecto y la funcionalidad en el lado del cliente, se han utilizado componentes Lightning, eventos y JavaScript. En total se han utilizado cinco componentes Lightning: componente de página Home, de información del módulo (*module path*), de contenido de unidad, de perfil de usuario, y un componente padre que engloba al resto.

Por un lado, en el archivo .cmp relativo a los componentes Lightning se determina la estructura y el contenido estático de cierto componente (figura 32).

```

1 <aura:component access="GLOBAL" implements="flexipage:availableForAllPageTypes" controller="UserController">
2
3 <aura:dependency resource="markup://lightning:tile"/>
4 <aura:dependency resource="markup://lightning:badge"/>
5
6 <aura:attribute name="FirstName" type="String" default="" />
7 <aura:attribute name="PhotoUrl" type="String" default="/img/icon/profile32.png" />
8 <aura:attribute name="LastName" type="String" default="" />
9 <aura:attribute name="Points" type="Integer" default="100" />
10 <aura:attribute name="InProgressModules" type="String[]" default="" />
11
12 <aura:handler name="init" value="{!this}" action="{!c.doInit}" />
13
14
15 <!--<div aura:id="photoShow" class="slds-align-middle slds-m-around_small slds-show photoDiv" style="height:60px">
16 
17 </div-->
18
19 <div aura:id="profileDiv" class="slds-card-wrapper slds-m-vertical--small slds-size_1-of-1 slds-float_right">
20
21 <div class="slds-grid slds-wrap slds-m-bottom--small slds-float_right">
22
23 <div class="slds-align-middle slds-m-around--small slds-float_right">
24 <div class="slds-align-middle bold">{!v.LastName}, {!v.FirstName}</div>
25 <div class="slds-align-middle slds-float_right">{!$Label.c.ADELeLearningPointsString} {!v.Points}</div>
26 </div>
27
28 
29
30 </div>
31

```

Figura 32 - Contenido de archivo .cmp

Por otro lado, en el controlador (JavaScript) se encuentra la parte lógica del cliente, así como la parte dinámica del componente y los elementos que conectan al cliente con el servidor (figura 33).

```

1 {
2   doInit : function(component, event, helper) {
3
4
5     component.addEventListener("scroll", component.getReference("c.onScrollEvent"));
6
7     component.set("v.topic",[{label: $A.get("$Label.c.ADELeLearningAllString"), 'value': 'All'},
8       {label: 'ADEL', 'value': 'ADEL'},
9       {label: 'Animal Care', 'value': 'Animal Care'},
10      {label: 'Human Care', 'value': 'Human Care'},
11      {label: 'Plant Care', 'value': 'Plant Care'}]);
12
13     component.set("v.level",[{label: $A.get("$Label.c.ADELeLearningAllString"), 'value': 'All'},
14       {label: 'Beginner', 'value': 'Beginner'},
15       {label: 'Intermediate', 'value': 'Intermediate'},
16       {label: 'Hard', 'value': 'Hard'}]);
17
18     console.log(component.get("v.topic"));
19     console.log(component.get("v.level"));
20
21     var inProgressModules = component.find("InProgressModulesDiv");
22     var noModulesInProgressDiv = component.find("noModulesInProgressDiv");
23
24     if ($A.get("$Browser.isPhone")) {
25       $A.util.removeClass(inProgressModules, "slds-hide");
26     }
27

```

Figura 33 - Controlador de componente Lightning

## Componente de página home y componente de perfil

La página home es la primera página que se visualiza al entrar en la pestaña de la plataforma eLearning. Esta página contiene información sobre los distintos módulos disponibles en ADEL. El usuario puede observar distintas secciones en esta página.

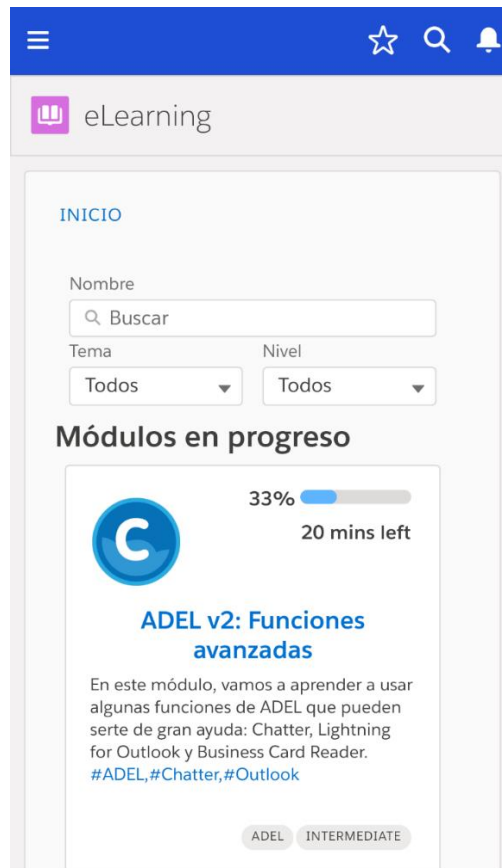


Figura 34 - Página Home en teléfono móvil

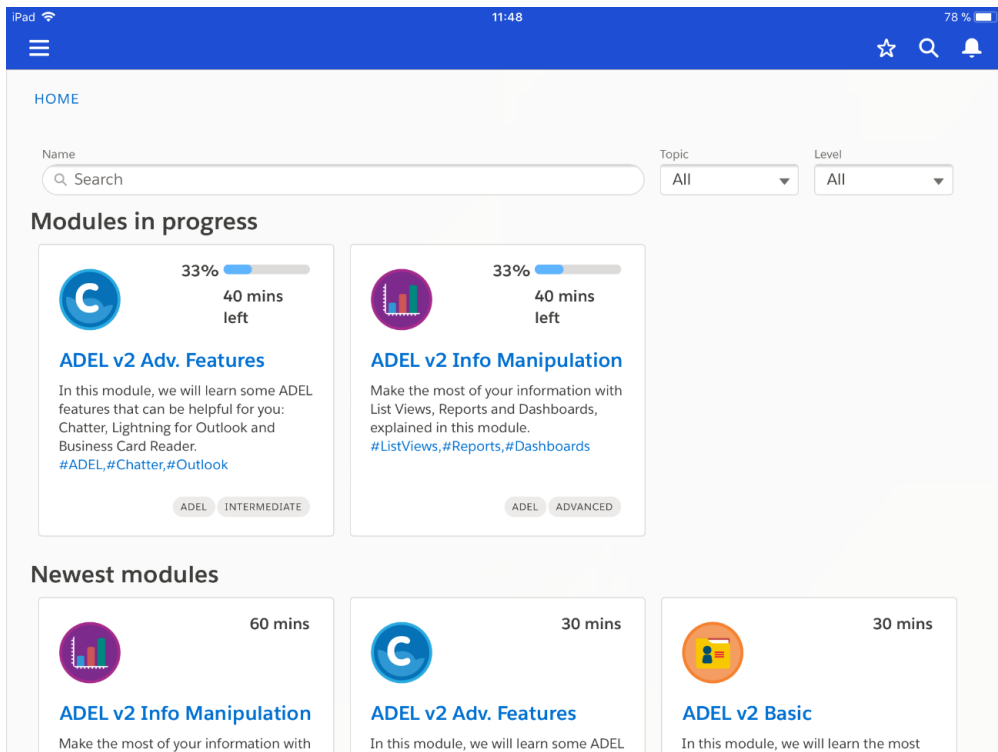


Figura 35 - Página Home en tablet

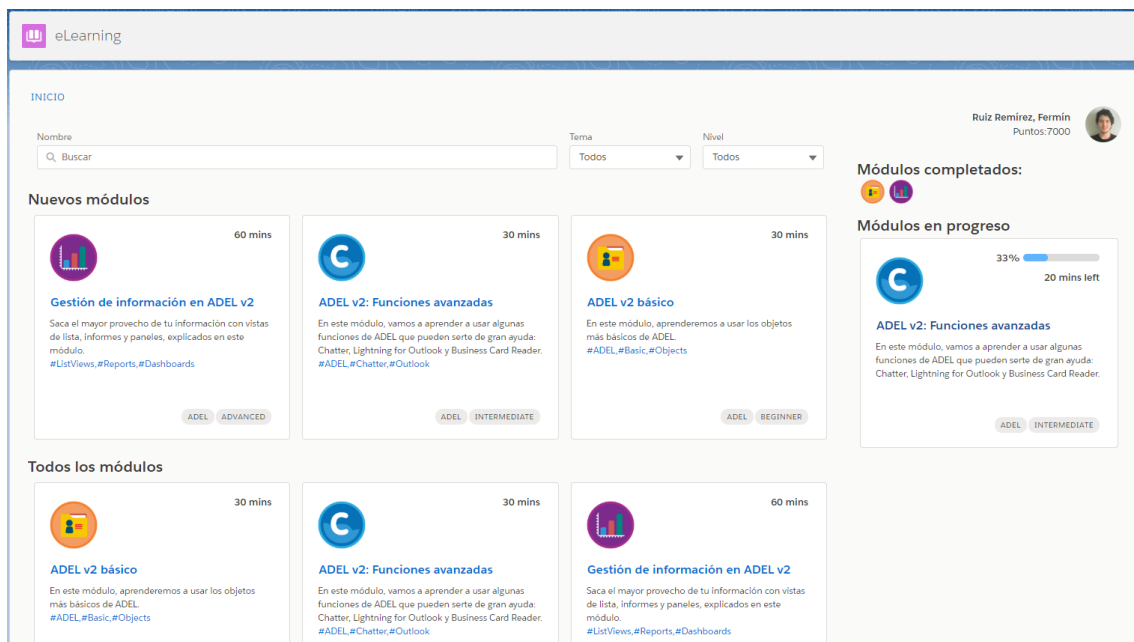


Figura 36 - Página Home en escritorio

En la parte superior de la página se puede observar una barra de búsqueda, en la que el usuario puede buscar módulos por texto (se mostrarán los módulos que contengan la cadena de texto buscada en el título, la descripción o los tags), por tema o por nivel de dificultad.

Debajo se encuentran una serie de apartados que muestran un número de módulos. Desde un teléfono móvil o tablet (figuras 34 y 35) se ven los apartados de “módulos en progreso”, “módulos recientes” y “todos los módulos”. Desde el escritorio sólo se ven estos dos últimos, ya que los módulos en progreso se ven en el componente de perfil, en una barra lateral a la derecha. Además, en la parte superior del componente de perfil se puede observar información del usuario, como su nombre, su foto de perfil o los módulos que ya ha completado (figura 36).

### Componente de información del módulo

El componente de información del módulo o *module path* se descubre cuando se hace clic en uno de los módulos de la página home. En él se muestra información sobre el módulo, además de los títulos y descripciones de todas las unidades del módulo. El botón de las unidades varía que han sido completadas por el usuario para que este pueda diferenciarlas. En las siguientes figuras se ve el componente de información del módulo para móvil (figura 37), tablet (figura 38) y escritorio (figura 39):



Figura 37 - Información de módulo en teléfono móvil

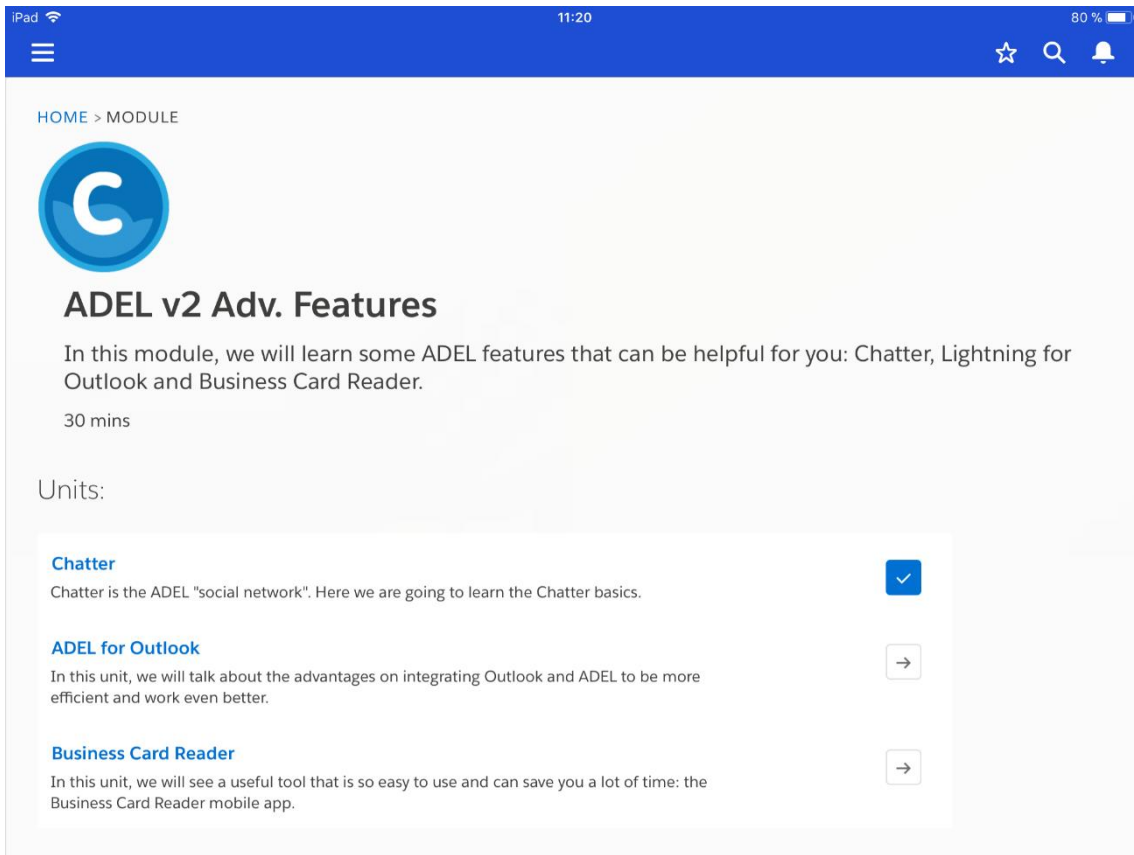


Figura 38 - Información de módulo en tablet

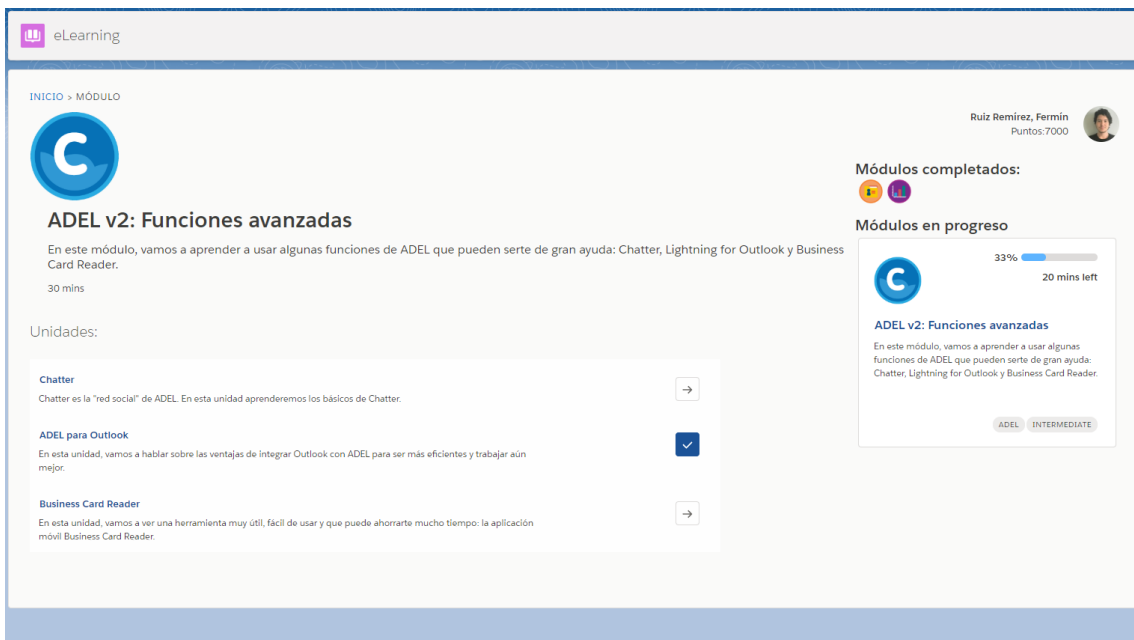


Figura 39 - Información de módulo en escritorio



## Componente de contenido de la unidad

Por último, en el componente de contenido de la unidad se muestra toda la información de la unidad seleccionada en el *module path*. Si la unidad contiene un vídeo, este aparecerá en un cuadro entre el título de la unidad y su contenido. Al final del contenido se verá un botón con el que se completa la unidad. En las figuras 40, 41 y 42 se puede apreciar el componente de contenido de la unidad en teléfono móvil, tablet y escritorio, respectivamente.



Figura 40 - Contenido de unidad en teléfono móvil

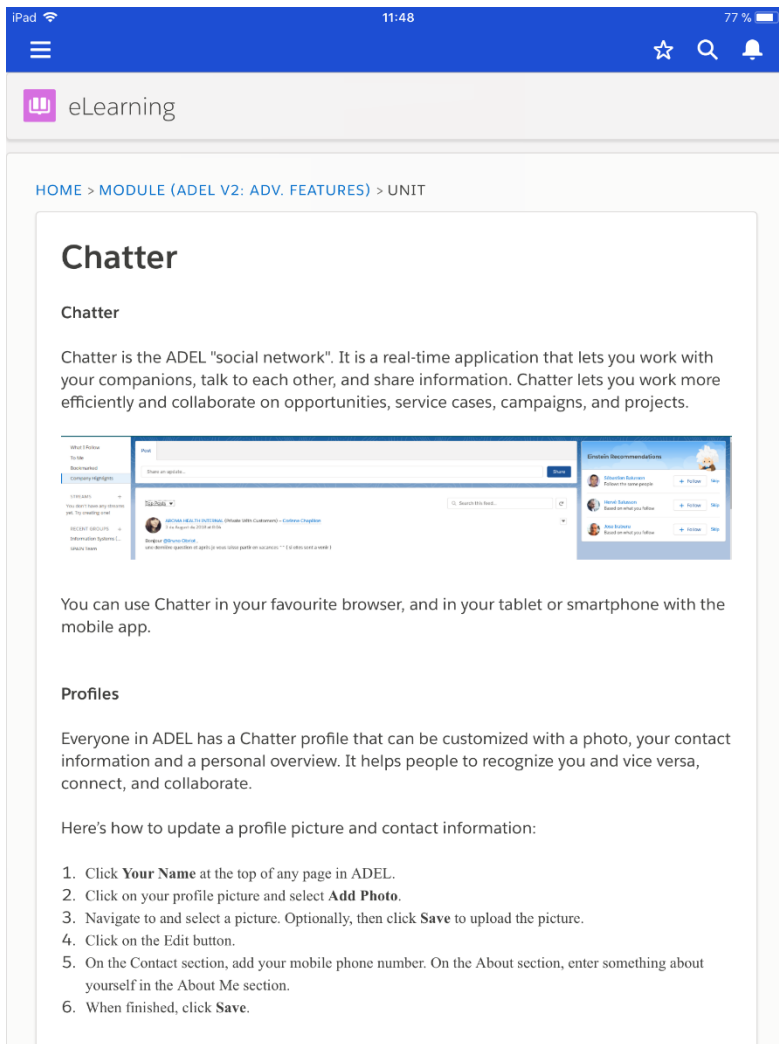


Figura 41 - Contenido de unidad en tablet

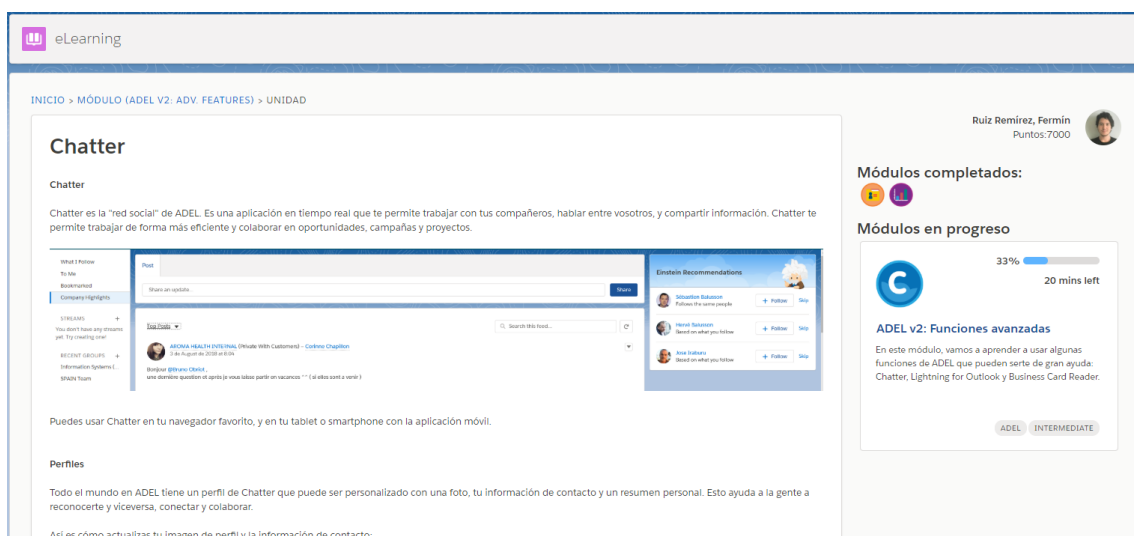


Figura 42 - Contenido de unidad en escritorio

## Componente padre y eventos

Los cambios e interacciones entre los distintos componentes son posibles gracias a los eventos (figura 43). Los eventos son emitidos y recibidos por los componentes, y hay una cantidad considerable de eventos utilizados en esta plataforma. Por ejemplo, seleccionar un módulo, seleccionar una unidad, utilizar las migas de pan de la parte superior o completar un módulo son situaciones que desencadenan un evento.

```
1 <aura:event type="APPLICATION" access="GLOBAL">
2
3     <aura:attribute name="id" type="String"/>
4     <aura:attribute name="completed" type="Boolean"/>
5
6 </aura:event>
```

Figura 43 - Estructura de evento Lightning

Todos los componentes están dentro del componente padre, por lo que este puede captar todos los eventos y aplicar clases de estilo a los distintos hijos. Por ejemplo, cuando se selecciona un módulo en la página home se desencadena un evento que es captado tanto por el componente del *module path* como el componente padre. El componente del *module path* carga el contenido del módulo seleccionado, y el componente padre se encarga de ocultar el componente de página home y mostrar el componente de información del módulo.

## Etiquetas multiidioma

Así como para poder incluir los módulos y las unidades en distintos idiomas se crearon nuevos objetos y campos, para mostrar los textos fijos de la aplicación en distintos idiomas se hace uso de las etiquetas. Las etiquetas son variables que almacenan una cadena de texto en un idioma por defecto y después se pueden añadir traducciones del texto a cualquiera de los distintos idiomas disponibles en la aplicación (figura 44).

**Detalle de Etiqueta personalizada** Modificar Eliminar

Descripción breve	ADEL eLearning Module Completion String
Idioma	Inglés
Categorías	
Valor	Congratulations, you completed the module!
Creado por	Fermin Ruiz Remirez, 7/08/2018 13:09

Modificar Eliminar

**Traducciones** Nuevo

Acción	Idioma	Texto de traducción
<a href="#">Modificar</a>   <a href="#">Eliminar</a>	Español	¡Enhorabuena, has completado el módulo!

Figura 44 - Etiqueta personalizada traducida

Para la herramienta eLearning, se ha elegido el inglés como idioma por defecto en todas las etiquetas, y se han traducido por completo al español. Así, un usuario que trabaje con ADEL en español verá los textos de la plataforma en español, y un usuario que utilice la aplicación en cualquier otro idioma verá los textos en inglés.

#### 5.4. Lado del servidor

En el desarrollo de la lógica en el lado del servidor, se han creado dos clases Apex. Una de ellas es utilizada por el componente de perfil, y la otra por el resto de los componentes. Ambas se utilizan para pedir, crear o modificar registros desde el lado del cliente, como por ejemplo modificar el campo “completado” de un registro de unidad de usuario cuando se hace clic en “completar unidad”, o se realizan peticiones de todos los registros de módulos que cumplan los criterios de búsqueda cada vez que se cambia algo en la barra de búsqueda de la página home. Al tratarse de un servidor sin estado, todo lo que desarrollado en estas clases Apex son funciones que son llamadas por los componentes desde el cliente (figura 45).

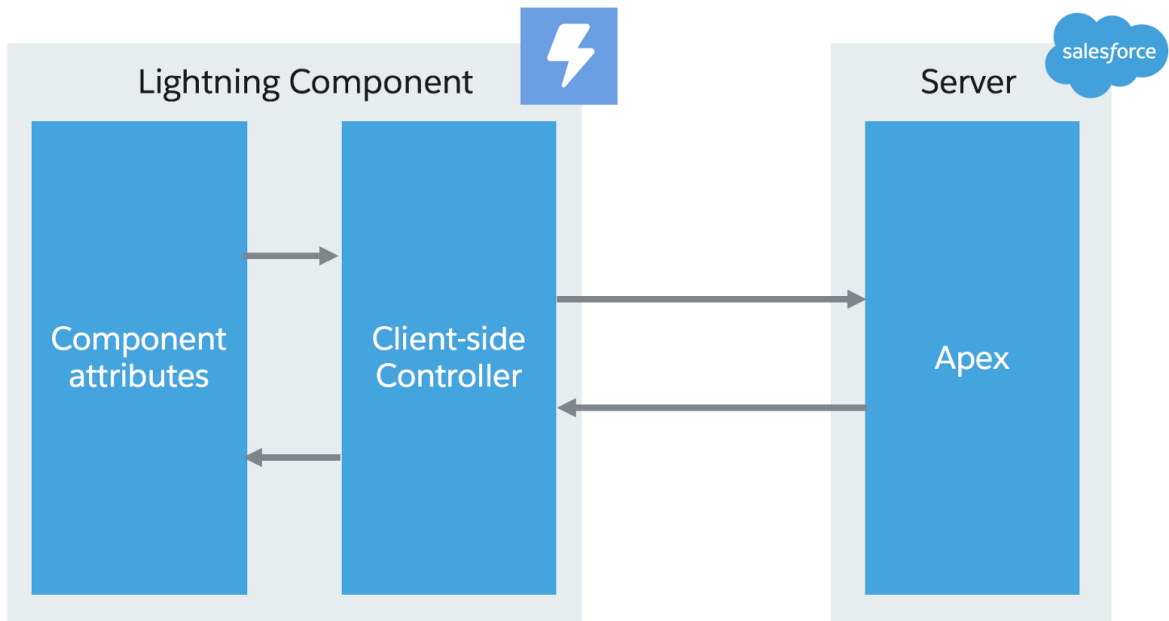


Figura 45 - Estructura componente/servidor

En la figura 46 se puede observar la estructura de una clase Apex (archivo .cls). Se puede apreciar que es similar al lenguaje de programación Java, y está compuesto de funciones que en su mayoría devuelven registros de uno de los objetos de la aplicación.

```

1 public with sharing class ModuleController {
2
3     @AuraEnabled
4     public static List<Module_Description_c> getModuleDescInfo(String id) {
5         List<Module_Description_c> result = [SELECT Id, Title_c, Description_c, Module_c, Module_r.Module_Image_c, Module
6
7         List<User_Module_c> userModules = [SELECT Id FROM User_Module_c WHERE Module_c = :result[0].Module_c AND User_c =
8
9         if(userModules.isEmpty()) {
10
11             User_Module_c userModule = new User_Module_c (User_c = UserInfo.getUserId(), Module_c = result[0].Module_c);
12             insert userModule;
13
14             List<Unit_c> units = [SELECT Id FROM Unit_c WHERE Module_c = :result[0].Module_c];
15
16             for (Integer i = 0; i < units.size(); i++) {
17
18                 User_Unit_c userUnit = new User_Unit_c (User_c = UserInfo.getUserId(), Unit_c = units[i].Id);
19                 insert userUnit;
20             }
21
22         }
23
24         return result;
25     }
26
27     @AuraEnabled
28     public static List<User_Unit_c> getUserUnits(String id) {
29
30         String language = [SELECT Language_c FROM Module_Description_c WHERE Id = :id LIMIT 1].Language_c;
31         String moduleId = [SELECT Module_c FROM Module_Description_c WHERE Id = :id LIMIT 1].Module_c;
32         List<User_Unit_c> result = [SELECT Id, Unit_r.Title_c, Unit_r.Description_c, Completed_c, Unit_r.Unit_Number_c
33         List<Unit_Description_c> titles = [SELECT Title_c, Description_c FROM Unit_Description_c WHERE Unit_r.Module_c =
34
35         for (Integer i = 0; i < titles.size(); i++) {
36             result[i].Unit_r.Title_c = titles[i].Title_c;
37             result[i].Unit_r.Description_c = titles[i].Description_c;
38         }
39
40         return result;
41     }
42 }

```

Figura 46 - Estructura de clase Apex

## 5.5. Gamificación

La gamificación es el proceso por el que se utilizan principios y técnicas relacionadas con los juegos en actividades no lúdicas con el objetivo de motivar y potenciar al usuario. Principalmente este recurso se utiliza en la educación [23].

Para intentar “enganchar” al usuario en el aprendizaje, se ha intentado aplicar una de las mecánicas de gamificación a la herramienta eLearning: la adición de puntos al completar unidades. Para ello se ha añadido al objeto “unidad” el campo “puntos”, en el que los desarrolladores, dependiendo de la duración y dificultad de la unidad, pueden determinar el número de puntos que obtienes al completarla. Así, en el componente de perfil de la herramienta se puede observar la cantidad de puntos que tiene el usuario.

También se muestran miniaturas de las imágenes de los módulos completados a modo de logros, de modo que el usuario puede ver una lista de sus éxitos.

## 5.6. Seguridad

En cuanto a la seguridad en la plataforma eLearning, se podría distinguir entre seguridad en el acceso y seguridad de edición. Respecto a la seguridad en el acceso, al estar la herramienta integrada en la aplicación ADEL, solo podrán acceder a ella aquellos empleados de Olmix Group que dispongan de una cuenta en ADEL con una licencia activa, al igual que hace falta para acceder a cualquier otra funcionalidad de ADEL.

Respecto a la seguridad de edición de los objetos y los registros implementados para la plataforma, los usuarios que no tengan un nivel de cuenta de administrador sólo podrán crear y editar registros de los objetos “módulo de usuario” y “unidad de usuario”. Es decir, sólo pueden editar sus instancias, como por ejemplo editar el campo “etapa” del registro del campo “módulo de usuario” relacionado con el usuario cuando se completan todas las unidades de ese módulo.

Todos los usuarios tienen permiso de visualizar (sin editar) los demás objetos de la plataforma eLearning, pero tienen deshabilitados todos los accesos a esa información que no sean la propia herramienta. Por ello, lo único que pueden ver o editar es lo que se permite con los movimientos que pueden realizar en el uso cotidiano de esta.

Como sólo los administradores tienen acceso para ver y editar todos los objetos y campos, si un empleado de un departamento que no sea de Sistemas de Información quiere añadir un curso a la plataforma, debe ponerse en contacto con un administrador de ADEL que, después de comprobar el contenido que quiere ser introducido en la plataforma, incluirá el contenido si lo considera correcto.

## 5.7. Validación y test

Mientras se estaba desarrollando la herramienta, se realizaron pruebas desde un teléfono Android con la aplicación de Salesforce mobile y en escritorio, desde una cuenta de administrador. Una vez terminada la plataforma por completo, se probaron todas las funcionalidades implementadas y se comprobó que funcionaba correctamente en el entorno de desarrollo.

Después de un resultado satisfactorio de las pruebas, se procedió a la migración completa de la plataforma del entorno de desarrollo al entorno de producción utilizando los *change sets* (conjuntos de cambios). Los conjuntos de cambios se utilizan para enviar personalizaciones y metadatos de una organización o entorno de Salesforce a otro.

Se han realizado *change sets* por fases, ya que algunos elementos podrían tener dependencia de otros. Los elementos se han exportado del entorno de desarrollo en *change sets* siguiendo este orden:

- Etiquetas
- Objeto “módulos”
- Objetos dependientes de “módulos”
- Resto de objetos
- Clases Apex
- Componentes y eventos
- Traducciones

Una vez exportados los elementos en *change sets* salientes, deben ser examinados y aceptados en el entorno de producción como *change sets* entrantes. La mayoría de los elementos pueden ser importados en el entorno sin necesidad de ejecutar ninguna prueba, pero con las clases Apex es obligatorio realizar lo que se conoce como pruebas de Apex.

Las pruebas de Apex evalúan el código de las clases de Apex haciendo pruebas de casos de uso de estas creando clases de test [24]. Las clases de test llaman a todas las funciones de una clase Apex con distintos parámetros para cubrir el máximo porcentaje de líneas de código posible sin errores. Se puede apreciar en la figura 47:

```

1  @isTest
2  private class ModuleControllerTest {
3
4      @isTest static void test_method_one() {
5          // Implement test code
6          Module__c module = new Module__c(Title_c='AAAA', Description_c='aaaa', Default_Language_c='English', Available_Languages_c=
7          insert module;
8          Module__c module2 = new Module__c(Title_c='AAAA2', Description_c='aaaa2', Default_Language_c='Spanish', Available_Languages_c=
9          insert module2;
10         Module_Description__c moduleDescription = new Module_Description__c(Module__c=module.Id, Title_c='AAAA', Description_c='aaaa'
11         insert moduleDescription;
12         Module_Description__c moduleDescription2 = new Module_Description__c(Module__c=module.Id, Title_c='AAAA2', Description_c='aaaa
13         insert moduleDescription2;
14         Unit__c unit = new Unit__c(Module__c=module.Id, Points_c=1000, Unit_Number_c=1);
15         insert unit;
16         Unit__c unit2 = new Unit__c(Module__c=module.Id, Points_c=1000, Unit_Number_c=2);
17         insert unit2;
18         Unit__c unit3 = new Unit__c(Module__c=module2.Id, Points_c=1000, Unit_Number_c=1);
19         insert unit3;
20         Unit_Description__c unitDescription = new Unit_Description__c(Unit__c=unit.Id, Title_c='AAAAAAA', Description_c='aaaaaaaa',
21         insert unitDescription;
22         Unit_Description__c unitDescription2 = new Unit_Description__c(Unit__c=unit2.Id, Title_c='AAAAAAA2', Description_c='aaaaaaaa
23         insert unitDescription2;
24         Unit_Description__c unitDescription3 = new Unit_Description__c(Unit__c=unit3.Id, Title_c='AAAAAAA3', Description_c='aaaaaaaa
25         insert unitDescription3;
26
27         List<Module_Description__c> moduleDescResult = ModuleController.getModuleDescInfo(moduleDescription.id);
28         List<User_Unit__c> userUnitResult = ModuleController.getUserUnits(moduleDescription.id);
29         List<Module_Description__c> allModuleDesc = ModuleController.getAllModuleDesc();
30         List<Module_Description__c> newestModuleDesc = ModuleController.getNewestModuleDesc();
31         List<Module_Description__c> searchedModuleDesc = ModuleController.getAllSearchedModuleDesc('', 'All', 'All');
32         List<Unit_Description__c> unitInfo = ModuleController.getUnitInfo(unit.Id);
33         List<Unit_Description__c> unitInfo2 = ModuleController.getUnitInfo(unit3.Id);
34         List<Unit_Description__c> nextUnitInfo = ModuleController.setUnitCompleted(unit.Id, 'English');
35         List<Module_Description__c> inProgressModuleDesc = ModuleController.getInProgressModuleDesc();
36         List<Unit_Description__c> nextUnitInfo2 = ModuleController.setUnitCompleted(unit2.Id, 'English');
37         List<Module_Description__c> descFromModule = ModuleController.getDescFromModule(module.Id, 'English');
38     }
39 }
40 }

```

Figura 47 - Estructura de clase de test

Hay que crear una clase de test por cada clase Apex importada, por lo que ha sido necesario realizar y configurar dos clases de test. Los requerimientos para pasar el test son:

- Que se prueben todas las funciones de la clase Apex
- Que se cubra al menos el 75% de las líneas de código totales de la clase Apex

Con las clases de test configuradas, se ha alcanzado un cubrimiento del código del 90% para las dos clases Apex, por lo que la prueba es exitosa y todos los elementos de la herramienta eLearning pueden ser implementados en el entorno de producción.

Una vez configurada toda la estructura de la herramienta y asignada la seguridad de acceso a los objetos en el entorno de producción, se procede a incluir en la plataforma tres cursos relacionados con ADEL de distintos niveles, para probar los posibles casos de uso.

La herramienta ha sido probada con usuarios con permisos de administrador y sin ellos, y ha sido probada en escritorio, en la aplicación Android en un teléfono móvil, y en la aplicación iOS en un iPad, sin encontrar ningún problema en la utilización de la plataforma.



## 6. Conclusiones

Para satisfacer las necesidades de la empresa Olmix Group, se han desarrollado, implementado y mejorado una serie de funcionalidades que han supuesto un incremento en el valor y de las razones por las que usar la aplicación ADEL para los empleados.

Por un lado, se ha renovado la interfaz y el modelo de datos de la aplicación, y con ello se ha aumentado el abanico de posibilidades de uso de la aplicación, así como su comodidad. También se ha configurado la aplicación móvil Salesforce para que los usuarios puedan realizar sus tareas cuando se encuentran sin acceso a su ordenador, y se ha mejorado tanto la integración con el ERP como el reporting, dando lugar a una mejor explotación y la representación gráfica de los datos almacenados. El 100% de los usuarios de ADEL ya están utilizando la versión actualizada de la aplicación, y los nuevos objetos configurados ya son utilizados por los usuarios: por ejemplo, el objeto “leads” ya cuenta con más de 25000 registros, y del objeto “sales reports figures” de la integración con el ERP han sido incluidos más de 80000 registros.

Por otro lado, se ha desarrollado e implementado una herramienta eLearning integrada en la aplicación ADEL, en la que se han añadido módulos de aprendizaje en múltiples idiomas y se puede utilizar tanto en escritorio como en otros dispositivos. Además, está configurada para poder añadir módulos de temas que no sean relativos a ADEL, por lo que se puede usar para diferentes actividades de la empresa. Aunque ya está implementada, debe pasar por unos procesos burocráticos internos antes de ser desplegada a todos los usuarios de ADEL, pero su funcionamiento es óptimo, ha superado satisfactoriamente todas las pruebas realizadas en ella y en el proceso de validación se ha logrado un cubrimiento del 90% (muy por encima del 75% requerido), por lo que la implementación ha sido exitosa.

### 6.1. Puntos de mejora y líneas de trabajo futuro

Aunque la evolución ha sido de gran magnitud y el número de funcionalidades de ADEL ha incrementado considerablemente, aún hay ciertos puntos en los que se puede mejorar.

Por ejemplo, respecto al modelo de datos de la aplicación, aunque ya se ha configurado el objeto “oportunidades”, aún no se ha implementado en la organización de forma que los usuarios empiecen a utilizarlo, ya que se ha pensado que introducir al usuario demasiadas nuevas funcionalidades de una sola vez no es lo óptimo. De todas formas, se ha planificado que en un futuro cercano los usuarios deban utilizar este objeto.

En cuanto a la integración ERP-ADEL, aunque la solución de Talend funciona correctamente, ha habido algún día en el que la sincronización diaria con el archivo Excel guardado en Dropbox no se ha realizado. Esos días, ha sido necesario acceder de forma remota al servidor para ejecutar la tarea manualmente y que los nuevos registros aparezcan en ADEL. También, se han inspeccionado alternativas para realizar la integración, y la utilización de una herramienta de Salesforce llamada Apex Triggers, que podría evitar los problemas de

sincronización. Por ello, en el futuro se ha planificado utilizar esta herramienta para rehacer la integración.

Por último, también hay varias ideas adicionales que podrían mejorar la plataforma de eLearning. Por ejemplo, se ha pensado en configurar un campo de test en las unidades, que sea necesario realizar correctamente para poder completar la unidad. También está la idea de añadir lo que se llamaría *Journeys*, que serían agrupaciones de módulos que trataran un tema común, o agrupaciones creadas por ciertos usuarios interesados en que los empleados a su cargo las realicen.

## 7. Bibliografía

- [1] «myTrailhead,» [En línea]. Available: <https://quip.com/lthKAuemOn50>.
- [2] «Salesforce,» [En línea]. Available: <https://www.salesforce.com/crm/what-is-crm/>.
- [3] «Edureka,» 19 7 2018. [En línea]. Available: <https://www.edureka.co/blog/what-is-salesforce/>.
- [4] «Trailhead,» [En línea]. Available: [https://trailhead.salesforce.com/en/modules/data\\_modeling](https://trailhead.salesforce.com/en/modules/data_modeling). [Último acceso: 14 8 2018].
- [5] «Salesforce Developers,» 2017. [En línea]. Available: [https://developer.salesforce.com/docs/atlas.en-us.fundamentals.meta/fundamentals/adg\\_securing\\_data.htm](https://developer.salesforce.com/docs/atlas.en-us.fundamentals.meta/fundamentals/adg_securing_data.htm).
- [6] «Salesforce Help,» 2017. [En línea]. Available: [https://help.salesforce.com/articleView?id=security\\_overview.htm](https://help.salesforce.com/articleView?id=security_overview.htm).
- [7] Mozilla, «MDN web docs,» 2018. [En línea]. Available: [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics).
- [8] «Salesforce Developers,» 2017. [En línea]. Available: [https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_intro\\_what\\_is\\_apex.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm).
- [9] «Salesforce Developers,» 2017. [En línea]. Available: [https://developer.salesforce.com/docs/atlas.en-us.eclipse.meta/eclipse/intro\\_components.htm](https://developer.salesforce.com/docs/atlas.en-us.eclipse.meta/eclipse/intro_components.htm).
- [10] «Salesforce Developers,» 2017. [En línea]. Available: [https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/components\\_overview.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/components_overview.htm).
- [11] «Salesforce Developers,» 2017. [En línea]. Available: [https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/intro\\_framework.htm](https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/intro_framework.htm).
- [12] «Salesforce Help,» [En línea]. Available: [https://help.salesforce.com/articleView?id=code\\_dev\\_console.htm&type=5](https://help.salesforce.com/articleView?id=code_dev_console.htm&type=5).
- [13] «Salesforce Help,» 2017. [En línea]. Available: [https://help.salesforce.com/articleView?id=create\\_test\\_instance.htm&type=5](https://help.salesforce.com/articleView?id=create_test_instance.htm&type=5).
- [14] «Salesforce Developers,» 2017. [En línea]. Available: <https://developer.salesforce.com/docs/atlas.en->

us.appExchangeInstallGuide.meta/appExchangeInstallGuide/appexchange\_install\_what  
is.htm.

- [15] «Salesforce Developers,» 2017. [En línea]. Available:  
[https://developer.salesforce.com/docs/atlas.en-us.dataLoader.meta/dataLoader/data\\_loader.htm](https://developer.salesforce.com/docs/atlas.en-us.dataLoader.meta/dataLoader/data_loader.htm).
- [16] «Talend,» [En línea]. Available: <https://www.talend.com/resources/etl-tools/>.
- [17] «Talend Help,» 2018. [En línea]. Available:  
<https://help.talend.com/reader/hKeW8H4Q2LhSMp4zunyBzg/MDauhq9nl1m0FXZX~aHp9Q>.
- [18] B. Gremillion, «UXPin,» [En línea]. Available: <https://www.uxpin.com/studio/blog/a-hands-on-guide-to-mobile-first-design/>.
- [19] «cPrime,» [En línea]. Available: <https://www.cprime.com/resources/what-is-agile-what-is-scrum/>.
- [20] «Salesforce Help,» 2017. [En línea]. Available:  
[https://help.salesforce.com/articleView?id=lex\\_welcome\\_to\\_lex.htm](https://help.salesforce.com/articleView?id=lex_welcome_to_lex.htm).
- [21] «Trailhead,» 2018. [En línea]. Available:  
[https://trailhead.salesforce.com/modules/lex\\_migration\\_introduction/units/lex\\_migration\\_introduction\\_administration](https://trailhead.salesforce.com/modules/lex_migration_introduction/units/lex_migration_introduction_administration).
- [22] «Trailhead,» 2018. [En línea]. Available:  
[https://trailhead.salesforce.com/modules/lex\\_dev\\_visualforce/units/lex\\_dev\\_visualforce\\_intro](https://trailhead.salesforce.com/modules/lex_dev_visualforce/units/lex_dev_visualforce_intro).
- [23] «Learning Theories,» [En línea]. Available: <https://www.learning-theories.com/gamification-in-education.html>.
- [24] «Trailhead,» [En línea]. Available:  
[https://trailhead.salesforce.com/en/modules/apex\\_testing](https://trailhead.salesforce.com/en/modules/apex_testing).