

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Sistemas de Información



Trabajo Fin de Grado

Optimización en la gestión de clientes: IoT

ESCUELA POLITECNICA
SUPERIOR

Autor: Eduardo Ventura Izquierdo Nuñez

Tutor/es: José Ramón Hilera González

2017



Universidad
de Alcalá

Escuela Politécnica Superior

Grado en Sistemas de Información

Trabajo Fin de Grado

Optimización en la gestión de clientes: IoT

Autor: Eduardo Ventura Izquierdo Nuñez

Tutor: José Ramón Hilera González

Tribunal

Presidente:

Vocal 1º:

Vocal 2º:

Fecha:

Agradecimientos

Me gustaría dedicar este proyecto, así como cada uno de los logros alcanzados a lo largo de mi formación en el grado en sistemas de información a mi abuelo; responsable de mi iniciación en el mundo de la informática y con el cual pude compartir mi pasión.

A Patricia Sotodosos San Clemente; compañera desde el inicio de mis estudios en la Universidad de Alcalá y cuya amistad supone uno de los mayores regalos que me llevo de mis estudios universitarios.

A mi madre, padre y hermana cuyos ánimos me han ayudado no solo a superar distintas dificultades en mis estudios desde pequeño, sino que me han enseñado la importancia de la constancia y responsabilidad.

A mis profesores, gracias a los cuales no solo he logrado adquirir las competencias necesarias para triunfar en el mundo laboral, sino que me han ayudado a crecer como persona y en valores.

Índice

Optimización en la gestión de clientes: IoT

| | |
|---|-----------|
| AGRADECIMIENTOS..... | 2 |
| ÍNDICE | 3 |
| CAPÍTULO 1: DESCRIPCIÓN DEL PROYECTO..... | 4 |
| INTRODUCCIÓN | 4 |
| OBJETIVOS | 4 |
| RECURSOS EMPLEADOS..... | 6 |
| CAPÍTULO 2: TECNOLOGÍAS EMPLEADAS..... | 7 |
| INTERNET OF THINGS | 8 |
| <i>Soluciones disponibles en el mercado.....</i> | <i>8</i> |
| <i>Conclusiones para la solución seleccionada</i> | <i>9</i> |
| CUSTOMER RELATIONSHIP MANAGEMENT..... | 11 |
| <i>Soluciones disponibles en el mercado.....</i> | <i>11</i> |
| <i>Conclusiones para la solución seleccionada</i> | <i>11</i> |
| CAPÍTULO 3: INTERNET OF THINGS..... | 13 |
| PASOS INICIALES..... | 13 |
| DESARROLLO DEL PROTOTIPO | 13 |
| <i>IOT_Input.py</i> | <i>14</i> |
| <i>IOT_Output.py</i> | <i>16</i> |
| <i>ServerManagement.py</i> | <i>17</i> |
| INTERFAZ DE COMUNICACIÓN EXTERNA | 20 |
| <i>Servicio de recuperación de información ambiental.....</i> | <i>20</i> |
| <i>Servicio de control de recursos domóticos.....</i> | <i>21</i> |
| DIAGRAMA DE CLASES DEL PROTOTIPO | 24 |
| ESQUEMA ELÉCTRICO DEL PROTOTIPO | 25 |
| CAPÍTULO 4: CUSTOMER RELATIONSHIP MANAGEMENT..... | 26 |
| PASOS INICIALES..... | 26 |
| IMPLEMENTACIÓN DE OBJETOS | 28 |
| <i>Contactos.....</i> | <i>28</i> |
| <i>Reservas.....</i> | <i>35</i> |
| <i>Habitaciones.....</i> | <i>41</i> |
| <i>Servidores</i> | <i>44</i> |
| ASISTENTE DE ASIGNACIÓN DE RECURSOS..... | 47 |
| <i>Validación de registros</i> | <i>47</i> |
| <i>Aura Definition Bundle.....</i> | <i>48</i> |
| <i>Interfaces de usuario</i> | <i>58</i> |
| SERVICIOS REMOTOS..... | 60 |
| <i>Implementación de tareas programadas</i> | <i>60</i> |
| <i>Implementación de servicios.....</i> | <i>61</i> |
| IMPLEMENTACIÓN DE APLICACIONES PERSONALIZADAS..... | 66 |
| BIBLIOGRAFÍA | 68 |

Capítulo 1: Descripción del proyecto

Introducción

Este documento pretende proporcionar la información necesaria para implementar un completo sistema de gestión de clientes orientado a empresas del sector hotelero, capaz de integrarse completamente con un sistema de control domótico, el cual será instalado en cada una de las habitaciones; para comenzar, se realizará un estudio de las soluciones disponibles en el mercado para la implementación de dichos sistemas; y finalmente, se dará paso a la implementación de los mismos acorde a las características solicitadas en los objetivos del proyecto, siendo preciso añadir, que la implementación de los sistemas será descrita con gran detalle técnico, permitiendo la reproducción del sistema.

La competitividad empresarial alcanza los mayores picos de la historia, ocasionando que la captación y retención de clientes suponga un reto para asegurar la supervivencia de una determinada firma empresarial en el mercado; es preciso comentar, que el sector de la hostelería y el turismo es uno de los más afectados como consecuencia del elevado número de competidores que luchan por un cliente.

Las grandes empresas hoteleras han actuado tradicionalmente recopilando información valiosa de forma manual en torno a sus clientes, almacenándola con el fin de mejorar el desempeño de algunos procesos empresariales tales como la asignación de habitaciones o registro de nuevos clientes, apoyándose para ello en grandes sistemas de información.

Estos sistemas hacen posible el procesamiento y centralización de la información de los clientes de forma colaborativa para toda la organización, permitiendo ofrecer soporte a todos los procesos empresariales implicados en dicha información; actualmente, resulta indispensable la participación de los empleados, haciendo posible la coherencia entre el sistema de información y la realidad.

Actualmente, nos encontramos ante un mundo de nuevas posibilidades consecuencia de los increíbles avances tecnológicos acontecidos en estos últimos años como es el caso de *Internet of Things*; tecnología la cual abre las puertas a una nueva realidad en la que la automatización hará posible que la gestión masiva de clientes no suponga un reto inviable dado un conjunto limitado de recursos; maximizando la fidelización de clientes como resultado.

Objetivos

El proyecto aquí desarrollado supone la implementación simplificada de un completo y funcional sistema *Customer Relationship Management*, el cual sea capaz de aprovechar la tecnología de *Internet of Things* para realizar una mejor gestión de los recursos de un complejo hotelero ficticio en base a la automatización de ciertas actividades hasta ahora gestionadas de forma manual por los empleados.

El sistema *Customer Relationship Management* ha de disponer de las configuraciones y desarrollos precisos para soportar el almacenamiento y gestión de la información crítica

de un supuesto complejo hotelero, de tal forma que esta sea accesible para la gestión de los procesos empresariales básicos implicados en la operativa diaria de la empresa.

El carácter diferencial de esta propuesta es la comunicación bidireccional con un sistema de control domótico, el cual permita tanto la recuperación de información como son los valores de humedad y temperatura, así como la ejecución de acciones de control sobre los recursos críticos como son la energía, agua o cierre electrónico de puertas.

A continuación, se define el conjunto de características que ha de satisfacer el prototipo desarrollado.

I. Requisitos para el sistema de control domótico

- a. El sistema ha de permitir la recuperación de la información ambiental, así como el control de los recursos críticos de una habitación.
- b. El sistema ha de ser escalable, permitiendo la ampliación del alcance del sistema sin impactar a la funcionalidad ya implementada.
- c. El sistema ha de permitir el acceso a la funcionalidad desarrollada en base a servicios; basándose en las tecnologías REST y JSON.
- d. El sistema ha de proteger mediante credenciales de usuario el acceso al servicio de control de recursos críticos.
- e. El sistema ha de disponer de un control de errores, informando en todo momento de los eventos que impiden la explotación de los servicios.

II. Requisitos para el sistema de gestión de clientes

- a. El sistema ha de almacenar de forma estructurada los datos originados por un supuesto complejo hotelero.
- b. El sistema ha de disponer de los mecanismos necesarios para asegurar la integridad y fiabilidad de los datos del sistema.
- c. El sistema ha de simplificar al máximo el proceso de gestión de reservas; guiando al usuario en la configuración, y haciendo uso de los datos de la plataforma de forma coherente.
- d. El sistema ha de implementar un buscador de habitaciones configurable en base a distintos parámetros.
- e. El sistema ha de automatizar la gestión de disponibilidad de habitaciones en función de los procesos de reserva activos.
- f. El sistema ha de permitir la configuración de un sistema domótico para cada habitación de forma opcional.
- g. El sistema ha de gestionar las incidencias del sistema domótico de forma automática, bloqueando el servicio a la espera de su resolución.
- h. El sistema ha de ser compatible con la nueva interfaz de usuario Lightning Experience, así como con la aplicación móvil Salesforce1.

Recursos empleados

La construcción del prototipo requiere de un conjunto de componentes electrónicos, así como de una placa computadora y del acceso a una plataforma *Customer Relationship Management*; permitiendo la correcta operación del sistema a través de internet.

A continuación, se proporciona el listado completo de componentes empleados para el desarrollo del sistema de control domótico.

| Recursos empleados | | | |
|-----------------------------|--------|----------|---|
| Sistema de control domótico | | | |
| Nombre del producto | Coste | Unidades | Características |
| Equipo de placa reducida | €10.00 | 1 | Modelo: Raspberry Pi 1 Modelo B Sistema Operativo: Raspbian Jessie |
| Tarjeta de memoria | €4.69 | 1 | Tipo: SDHC Capacidad: 8Gb |
| Red | €1.99 | 1 | Tipo: RJ-45 Longitud: 1m |
| Alimentación | €9.99 | 1 | Terminación: Micro USB B Longitud: 1m |
| Placa de pruebas | €2.00 | 1 | Dimensiones: 4.7 cm x 3.5 cm x 0.9 cm Puntos de contacto: 170 |
| Cableado | €4.01 | 1 | Terminaciones: M-M, M-H y H-H |
| Sensor | €3.00 | 1 | Módulo: DHT11 Medición: Humedad y temperatura |
| Resistencia variable | €0.36 | 1 | Valor de resistencia variable: 10K |
| Relé | €3.00 | 3 | Controlador: 5V Canal: 250V ~ 10A |
| | | | |
| Flask Framework | €0.00 | N/A | Versión: 0.12 |

Finalmente, indicar que el sistema *Customer Relationship Management* requiere de una cuenta de usuario en la plataforma; indicándose los pasos necesarios para obtenerla en el capítulo destinado a la construcción de dicho sistema (véase capítulo 4).

A continuación, se proporciona el desglose de componentes empleados en el desarrollo del sistema de gestión de clientes.

| Recursos empleados | | | |
|--------------------------------|-------|----------|---------------------------------------|
| Sistema de gestión de clientes | | | |
| Nombre del producto | Coste | Unidades | Características |
| Cuenta de usuario | €0.00 | 1 | Edición: Salesforce Developers |

Capítulo 2: Tecnologías empleadas

En la actualidad, el mercado cuenta con numerosas y variadas soluciones tecnológicas capaces de satisfacer los requisitos de cualquier tipo de proyecto; siendo indispensable la identificación y posterior comparación entre las mismas, a fin de seleccionar aquella opción más adecuada para el proyecto que se desea construir.

El prototipo propuesto requiere implementar dos sistemas independientes capaces de comunicarse entre sí a través de internet; por lo que será necesario identificar cada una de las soluciones de forma aislada.

La búsqueda de soluciones comerciales requiere la identificación de los requisitos que han de ser cubiertos por la tecnología; para ello se toman los objetivos del proyecto a fin de extraer el alcance tecnológico de cada una de las necesidades.

Las necesidades identificadas para cada uno de los sistemas son las siguientes.

- I. Requisitos tecnológicos del sistema de control domótico
 - a. El dispositivo contará con un tamaño mínimo, permitiendo su instalación en espacios limitados.
 - b. El dispositivo ha de realizar un mínimo consumo energético.
 - c. El dispositivo ha de permitir un funcionamiento continuo 24/7.
 - d. El dispositivo ha de disponer de conectividad a la red.
 - e. El dispositivo ha de disponer de puertos IO.
 - f. El sistema ha de permitir el uso de lenguajes de programación.

- II. Requisitos tecnológicos del sistema de gestión de clientes
 - a. El sistema ha de ser gratuito.
 - b. El sistema ha de permitir el almacenamiento de datos estructurados.
 - c. El sistema ha de permitir operaciones DML.
 - d. El sistema ha de permitir el desarrollo de interfaces personalizadas.
 - e. El sistema ha de disponer de mecanismos de control de errores.
 - f. El sistema ha de permitir la creación de nuevas entidades de datos.
 - g. El sistema ha de permitir desarrollos basados en programación orientada a objetos.
 - h. El sistema ha de permitir la conectividad con sistemas externos.

Internet of Things

El término de *Internet of Things* hace referencia a la capacidad de conectar a internet todo tipo de dispositivo de uso cotidiano; actualmente, esta filosofía se encuentra en constante crecimiento con el conocido movimiento *maker*.

Soluciones disponibles en el mercado

Las soluciones disponibles en el mercado pueden agruparse en dos grandes categorías, permitiendo realizar una selección más sencilla en base a los aspectos generales que las definen.

Equipos de placa reducida

Los equipos de placa reducida son sistemas informáticos completamente funcionales en los que todos los componentes que los definen se encuentran integrados en una placa base de pequeño tamaño; normalmente, estos dispositivos llevan consigo un conjunto de puertos IO, los cuales permiten acoplar distintos componentes electrónicos.

Estos dispositivos gozan de autonomía con respecto a otros sistemas al estar equipados con un sistema operativo propio, generalmente basado en Linux.

A continuación, se indican los ordenadores de placa reducida más populares.

| Internet of Things | | |
|---------------------------|----------------------|---|
| Equipos de placa reducida | | |
| Nombre del producto | Coste | Características |
| Raspberry Pi | \$35.00 ¹ | Los productos de la familia Raspberry Pi son los más conocidos en el ámbito de IoT; obteniendo en consecuencia, un excelente soporte para componentes de terceros. |
| Jaguarboard | \$79.00 ² | Los productos de la familia Jaguarboard son pioneros en instalar procesadores X86 en lugar de ARM, logrando una mayor versatilidad en el software que se puede ejecutar. |
| Beagleboard | \$55.00 ³ | Los productos de la familia Beagleboard son una de las soluciones más adecuadas para proyectos del ámbito IoT; en consecuencia, a su reducido tamaño y la cantidad de puertos IO. |

1. Precio de Raspberry Pi 3 Modelo B
2. Precio de Jaguar One
3. Precio de BeagleBone Black

Microcontroladores

Los microcontroladores de placa simple son productos hardware con capacidad de ser programados a través de un entorno de desarrollo; estos dispositivos incluyen todo tipo de puertos, ya sean digitales o analógicos, permitiendo una mejor experiencia en el uso de componentes electrónicos.

El funcionamiento de estos dispositivos depende de la programación ajustada a través del entorno de desarrollo instalado en un sistema externo.

A continuación, se indican los microcontroladores de placa simple más populares.

| Internet of Things | | |
|------------------------------------|----------------------|---|
| Microcontroladores de placa simple | | |
| Nombre del producto | Coste | Características |
| Arduino | \$24.95 ¹ | Los productos de la familia Arduino son los más conocidos en el ámbito <i>maker</i> ; siendo posible obtener a través de internet gran cantidad de documentación y soporte. Al tratarse de una compañía de software libre, el entorno de desarrollo es compatible con la mayoría de productos de otras firmas. |
| Adafruit | \$14.95 ² | Los productos de la familia Adafruit ofrecen soluciones más adaptadas a las necesidades particulares de los usuarios. Los productos de Adafruit son compatibles con el entorno de desarrollo de Arduino. |

1. Precio de Arduino One
2. Precio de Adafruit Flora

Conclusiones para la solución seleccionada

Evaluadas las soluciones hardware identificadas previamente en el análisis de mercado, los productos de la marca Raspberry Pi son los más adecuados para aquellos requisitos técnicos identificados en la construcción del sistema de control domótico.

Es posible identificar algunas ventajas adicionales como es la diversidad de modelos que ofrece la marca, permitiendo al consumidor adquirir aquella que mejor se adapte a sus necesidades particulares, así como la sencillez en cuanto la adquisición de sus productos ya sea físicamente o a través de internet.

A continuación, se realiza la evaluación de los requisitos identificados en base a uno de los modelos más económicos que ofrece la marca dada su antigüedad.

Raspberry Pi 1 Modelo B

| Requisito de proyecto | Requisito cumplido | Descripción |
|--|--------------------|---|
| Requisito 1: El dispositivo contará con un tamaño mínimo, permitiendo su instalación en espacios limitados. | Sí | Raspberry Pi 1 Modelo B cuenta con unas dimensiones de 8.56 cm x 5.39 cm por lo que podrá instalarse en espacios reducidos. |
| Requisito 2: El dispositivo ha de realizar un mínimo consumo energético. | Sí | Raspberry Pi 1 Modelo B realiza un consumo energético de 700 mA (3,5W). |
| Requisito 3: El dispositivo ha de permitir un funcionamiento continuo 24/7. | Sí | Raspberry Pi permite implementar proyectos domóticos de todo tipo. |
| Requisito 4: El dispositivo ha de disponer de conectividad a la red. | Sí | Raspberry Pi 1 Modelo B cuenta con un puerto de red 10/100 Ethernet (RJ-45) vía hub USB. |
| Requisito 5: El dispositivo ha de disponer de puertos IO. | Sí | Raspberry Pi 1 Modelo B dispone de 17 puertos de IO entre los 26 pines ofrecidos por GPIO en este modelo. |
| Requisito 6: El sistema ha de permitir el uso de lenguajes de programación. | Sí | Raspberry Pi es capaz de ejecutar sistemas operativos basados en Linux por lo que es posible instalar cualquier compilador compatible con el sistema. |

Una vez evaluados los distintos requisitos, se asegura que el modelo seleccionado para la implementación del proyecto cumple con todas las necesidades y por tanto no será preciso modificar la tecnología una vez iniciados los procesos de desarrollo.

Customer Relationship Management

Los sistemas *Customer Relationship Management* hacen referencia a aquellos productos software destinados a la optimización de las tareas que guardan relación con el cliente, permitiendo reducir los costes y mejorar la experiencia del mismo.

Soluciones disponibles en el mercado

La optimización en la gestión de clientes ha adquirido una importancia crítica para todo tipo de empresas, desembocando en la aparición de nuevas soluciones en el mercado o la mejora de los productos existentes, haciendo posible la adaptación de empresas de todos los tamaños y sectores a esta nueva realidad.

Los sistemas *Customer Relationship Management* acostumbran a basar su facturación en cuotas variables, relacionando la cuantía con las necesidades particulares de cada compañía.

A continuación, se indican los sistemas más populares.

| Customer Relationship Management | | |
|----------------------------------|-------------|---|
| Nombre del producto | Coste | Características |
| Salesforce | Gratuito | La edición Salesforce Developers garantiza el acceso ilimitado a una instancia del sistema, aunque con ciertas limitaciones en materia de usuarios y datos. |
| Dynamics | Suscripción | La edición Dynamics 365 divide el sistema en distintas aplicaciones, adquiriendo cada una de ellas en base a suscripciones individuales o planes conjuntos. |
| Zoho | Gratuito | La edición Zoho Free ofrece una alternativa interesante ante grandes soluciones como son Salesforce o Dynamics. Zoho Free ofrece las mismas limitaciones técnicas que Salesforce Developers. |

Conclusiones para la solución seleccionada

Evaluadas las soluciones software identificadas previamente en el análisis de mercado, los productos ofrecidos por la marca Salesforce ofrecen la solución más adecuada a los requisitos técnicos identificados para la construcción del sistema de gestión de clientes.

Una de las características más destacables del producto seleccionado es la posibilidad de hacer uso del sistema sin realizar pago alguno al contar con una edición destinada a

desarrolladores; la edición Salesforce Developers permite hacer uso de prácticamente todas las funciones disponibles sin apreciar las limitaciones técnicas.

A continuación, se realiza una evaluación de los requisitos identificados en base a la edición Salesforce Developers.

| Salesforce Developers | | |
|--|--------------------|--|
| Requisito de proyecto | Requisito cumplido | Descripción |
| Requisito 1: El sistema ha de ser gratuito. | Sí | La edición Salesforce Developers es completamente gratuita. |
| Requisito 2: El sistema ha de permitir el almacenamiento de datos estructurados. | Sí | Salesforce dispone de un sistema de gestión de datos basados en tablas. |
| Requisito 3: El sistema ha de permitir operaciones DML. | Sí | Salesforce es capaz de comprender sentencias de SQL clásicas. |
| Requisito 4: El sistema ha de permitir el desarrollo de interfaces personalizadas. | Sí | Salesforce permite tanto la personalización de las interfaces estándares; como la implementación de interfaces personalizadas. |
| Requisito 5: El sistema ha de disponer de mecanismos de control de errores. | Sí | Salesforce permite el control de datos en tres niveles; a nivel de campo, a nivel de validación y a nivel de apex. |
| Requisito 6: El sistema ha de permitir la creación de nuevas entidades de datos. | Sí | Salesforce permite la creación de objetos personalizados, así como de clases de apex. |
| Requisito 7: El sistema ha de permitir desarrollos basados en programación orientada a objetos. | Sí | Salesforce se basa en el paradigma de programación orientada a objetos. |
| Requisito 8: El sistema ha de permitir la conectividad con sistemas externos. | Sí | Salesforce dispone de los objetos apex precisos para invocar servicios web. |

Evaluados los distintos requisitos, se asegura que la edición seleccionada para la implementación del proyecto cumple con todas las necesidades y por tanto no será preciso modificar la tecnología una vez iniciados los procesos de desarrollo.

Capítulo 3: Internet of Things

Los sistemas satélites han de ser desarrollados como si de servidores independientes se trataran, haciendo posible el consumo de los servicios disponibles en base a un acuerdo de interfaz; el sistema satélite implementado, es el responsable de la automatización de los procesos de recuperación de información ambiental y ejecución de órdenes.

El sistema desarrollado se basa en las características técnicas ofrecidas por la Raspberry Pi 1 Modelo B; permitiendo hacer uso de los puertos IO ofrecidos por el módulo de GPIO, así como aprovechar las facilidades y comodidades del desarrollo software como si de un equipo tradicional se tratase.

A continuación, se especifica de forma ordenada las actividades llevadas a cabo para la implementación del sistema satélite.

Pasos iniciales

El sistema operativo recomendado por el fabricante dispone de todas las librerías requeridas para la implementación del proyecto; haciendo que Python tome especial relevancia como lenguaje de programación por defecto.

La implementación del sistema precisa de la instalación previa de un micro framework conocido como Flask, el cual facilita enormemente el desarrollo de servicios consumidos a través de REST.

La instalación de Flask es muy sencilla; únicamente es necesario ejecutar la siguiente orden en la terminal del sistema con permisos de súper usuario.

```
Sudo pip install Flask
```

Una vez autorizados todos los cambios solicitados por la terminal; será posible hacer uso de Flask para cualquier desarrollo de Python una vez importada la librería.

Desarrollo del prototipo

El desarrollo será realizado siguiendo los principios del paradigma de la programación orientada a objetos; permitiendo generar nuevas estructuras sin apenas impactar en los desarrollos previos.

A continuación, se especifica el conjunto de componentes que representa al sistema desarrollado.

IOT_Input.py

La implementación desarrollada hace posible la recuperación y transformación de los valores digitales generados por el sensor de humedad y temperatura a aquellos valores numéricos correspondientes.

La implementación sigue los principios expuestos en el paradigma de la programación orientada a objetos; permitiendo describir aquellos métodos y atributos que definen la clase.

- **init**: Identifica al constructor de la clase; permite la declaración de las variables globales de temperatura y humedad, así como la inicialización de las mismas con el valor por defecto.
- **binary**: Identifica aquel método privado destinado a la conversión de valores binarios en los correspondientes numéricos.
- **read**: Identifica aquel método privado destinado a la recuperación de los datos digitales del sensor, así como su conversión en valores del tipo entero y posterior asignación a las variables globales destinadas a la humedad y temperatura; para ello es preciso invocar al método binary.
- **getInfo**: Identifica aquel método público destinado al retorno de los valores de humedad y temperatura definidos en las variables globales; para ello es preciso invocar al método read.

A continuación, se facilita el código desarrollado.

```
# -*- coding: utf-8 -*-

#Python#
#Environmental Actions#

#Project name: Optimización en la gestión de clientes: IoT#
#Developer: Eduardo Ventura#

#Import Library#
import RPi.GPIO as GPIO
import time

class Environment(object):
    def __init__(self):
        self.__humidity = 0
        self.__temperature = 0

    def __binary(self, value):
        return int(value, 2)

    def __read(self):
        #Board Configuration#
        GPIO.setmode(GPIO.BCM)
```

```

GPIO.setup(4,GPIO.OUT)

#Setup Configuration#
GPIO.output(4,GPIO.HIGH)
time.sleep(0.025)
GPIO.output(4,GPIO.LOW)
time.sleep(0.02)

#Retrieve Data#
data = []
GPIO.setup(4, GPIO.IN, pull_up_down=GPIO.PUD_UP)

for i in range(0,500):
    data.append(GPIO.input(4))

#Variable Configuration#
count = 0
init = 0
bit = ""
control = 0

humidity = 0
temperature = 0

#Process Data#
try:
    init = data.index(0)
    flag = 0

    for signal in data[init:500]:
        if signal == 1:
            flag = 1
            count = count + 1

        if signal == 0 and flag == 1:
            flag = 0
            if count < 4: bit = bit + "0"
            else: bit = bit + "1"
            count = 0

    humidity = self.__binary(bit[0:8])
    temperature = self.__binary(bit[16:24])
    control = self.__binary(bit[32:40])

except:
    pass

#Validation#
check = humidity + temperature - control

```

```

    if check < 2 and check > -2:
        self.__humidity = humidity
        self.__temperature = temperature
    else:
        self.__humidity = 0
        self.__temperature = 0

def getInfo(self):
    self.__humidity = 0
    self.__temperature = 0

    while self.__humidity == 0 and self.__temperature == 0:
        self.__read()
        time.sleep(0.25)

    return self.__humidity, self.__temperature

```

IOT_Output.py

La implementación desarrollada hace posible la activación o desactivación de los relés responsables del control de los recursos domóticos disponibles en el sistema, como son la energía, agua y cierre electrónico de puertas.

La implementación sigue los mismos principios de programación orientada a objetos ya mencionados con anterioridad; permitiendo describir aquellos métodos y atributos que definen la clase.

- **init:** Identifica al constructor de la clase; sobre el mismo se definen los pines de control para cada uno de los relés, así como la configuración de los puertos GPIO.
- **energy:** Identifica aquel método público relativo a la activación o desactivación del relé destinado al control de la energía en base a un parámetro de entrada.
- **water:** Identifica aquel método público relativo a la activación o desactivación del relé destinado al control del agua en base a un parámetro de entrada.
- **block:** Identifica aquel método público relativo a la activación o desactivación del relé destinado al cierre de puertas en base a un parámetro de entrada.

A continuación, se facilita el código desarrollado.

```
# -*- coding: utf-8 -*-
```

```
#Python#
```

```
#Resource Actions#
```

```
#Project name: Optimización en la gestión de clientes: IoT#
```

```
#Developer: Eduardo Ventura#
```

```
#Import Library#
```

```
import RPi.GPIO as GPIO
```

```

import time
import IOT_Input

class Resource(object):
    def __init__(self):
        #Variable Configuration#
        self.__pin = [10, 9, 11]

        #Board Configuration#
        GPIO.setmode(GPIO.BCM)

    #Methods#
    def energy (self, action):
        #GPIO Configuration#
        GPIO.setup(self.__pin[0], GPIO.OUT)

        if action == True:
            GPIO.output(self.__pin[0], GPIO.HIGH)
        else:
            GPIO.output(self.__pin[0], GPIO.LOW)

    def water (self, action):
        #GPIO Configuration#
        GPIO.setup(self.__pin[1], GPIO.OUT)

        if action == True:
            GPIO.output(self.__pin[1], GPIO.HIGH)
        else:
            GPIO.output(self.__pin[1], GPIO.LOW)

    def block (self, action):
        #GPIO Configuration#
        GPIO.setup(self.__pin[2], GPIO.OUT)

        if action == True:
            GPIO.output(self.__pin[2], GPIO.HIGH)
        else:
            GPIO.output(self.__pin[2], GPIO.LOW)

```

[ServerManagement.py](#)

La implementación desarrollada hace posible acceder a través de la red a las funciones de lectura de humedad y temperatura ambiente, así como a las funciones de activación o desactivación de los relés que gestionan los recursos domóticos.

El sistema ha de proteger el acceso a las funciones más críticas como es el control de los recursos domóticos; en este caso, el cliente ha de introducir las credenciales facilitadas por el administrador del sistema en la petición del recurso.

Las peticiones enviadas por el cliente pueden estar formuladas incorrectamente, lo que hace esencial la evaluación de posibles errores en las peticiones, ofreciendo como consecuencia del error, un código y mensaje de error asociado; los errores informados por el sistema son los siguientes.

- **400:** Petición incorrecta; El error es informado para aquellas peticiones que no disponen de la estructura esperada para el recurso solicitado.
- **401:** No autorizado; El error es informado para aquellas peticiones que solicitan el consumo de recursos críticos, pero no disponen de credenciales de usuario o bien, estas son incorrectas.
- **404:** No encontrado; El error es informado para aquellas peticiones que solicitan el consumo de recursos inexistentes.

La implementación sigue los principios de programación imperativa; siendo necesaria la instanciación de las clases desarrolladas previamente como apoyo a los servicios.

- **get_tasks:** Servicio consumible sin necesidad de autenticación, destinado a la recuperación de información de humedad y temperatura ambiente; es preciso hacer uso de la clase Environment, responsable de recuperar y transformar los datos obtenidos por el sensor.
- **post_tasks:** Servicio consumible mediante autenticación, destinado al control de los recursos domóticos habilitados en el sistema; el servicio espera como cuerpo de la petición los valores de activación o desactivación para cada uno de los recursos mediante una estructura de datos JSON; es preciso hacer uso de la clase Resource, responsable de activar o desactivar los relés.

A continuación, se facilita el código desarrollado.

```
# -*- coding: utf-8 -*-

#Python#
#Web Service#

#Project name: Optimización en la gestión de clientes: IoT#
#Developer: Eduardo Ventura#

#!/flask/bin/python
from flask import Flask, jsonify, abort, request
import IOT_Input
import IOT_Output
import time

#Object Oriented Programming#
environment = IOT_Input.Environment()
resource = IOT_Output.Resource()

#Rest Server#
app = Flask(__name__, static_url_path = "")
```

```

#Authorization#
username = 'admin'
password = 'admin'

#Variable Configuration#
#GET#
current_humidity = 0
current_temperature = 0

#Variable Configuration#
#POST#
selected_energy = False
selected_water = False
selected_block = False

#Error Handler#
@app.errorhandler(400)
def bad_request(error):
    return jsonify( { 'error': 'Bad request' } ), 400

@app.errorhandler(401)
def unauthorized(error):
    return jsonify( { 'error': 'Unauthorized' } ), 401

@app.errorhandler(404)
def not_found(error):
    return jsonify( { 'error': 'Not found' } ), 404

#Services#
@app.route('/api/v1.0/automation', methods = ['GET'])
def get_tasks():
    #Variable Configuration#
    global current_humidity, current_temperature

    #Retrieve Data#
    current_humidity, current_temperature =
environment.getInfo()

    return jsonify( { "humidity": current_humidity,
'temperature': current_temperature } ), 200

@app.route('/api/v1.0/automation', methods = ['POST'])
def post_tasks():
    if not request.json:
        abort(400)
    else:
        if request.authorization and
request.authorization.username == username and

```

```

request.authorization.password == password:
    #Variable Configuration#
    global selected_energy, selected_water,
selected_block, selected_temperature
    selected_energy = request.json['energy']
    selected_water = request.json['water']
    selected_block = request.json['block']

    #Output#
    resource.energy(selected_energy)
    resource.water(selected_water)
    resource.block(selected_block)

    return jsonify(
        {
            'energy' : selected_energy,
            'water' : selected_water,
            'block' : selected_block
        }
    ), 200
else:
    abort(401)

```

#Fire#

```
app.run(debug = True, host='0.0.0.0')
```

Interfaz de comunicación externa

La explotación de las funciones implementadas en el sistema satélite implica el envío y la recepción de mensajes; siendo indispensable que los mismos sean comprendidos por todo sistema que participe en la comunicación.

A continuación, se documentan todos los servicios habilitados en el sistema de control domótico; describiendo toda aquella información necesaria para el correcto consumo de los servicios.

Servicio de recuperación de información ambiental

Servicio de consumo público el cual hace posible la recuperación de los valores actuales de humedad y temperatura recuperados por el sensor asociado al servidor indicado en el punto de acceso.

Información para peticiones

- **Punto de acceso:** `http://{server root:5000}/api/v1.0/automation`
- **Método:** GET
- **Cabecera:** No requerido para la petición del servicio.
- **Cuerpo:** No requerido para la petición del servicio.

Información para respuestas

- **Mensajes de error:** Una vez recuperada la información solicitada en la petición, el servicio adjunta el código de éxito en la respuesta, indicando que la operación se ha completado satisfactoriamente.
- **Cabecera:** No empleado.
- **Cuerpo:** Información de temperatura y humedad contenidas en una estructura de datos del tipo EnvironmentResponseType.

Ejemplo de respuesta en la respuesta una vez consumido el servicio.

```
{
  "temperature" : 10,
  "humidity" : 30
}
```

Tipo de datos empleados

| EnvironmentResponseType | | | |
|-------------------------|-------------|--------------|---|
| Variable | Obligatorio | Tipo de dato | Descripción |
| temperature | Sí | Integer | Información de temperatura ambiente recuperada por el sensor. |
| humidity | Sí | Integer | Información de humedad ambiente recuperada por el sensor. |

Servicio de control de recursos domóticos

Servicio de consumo controlado el cual hace posible la activación o desactivación de los distintos recursos domóticos habilitados en el servidor indicado en el punto de acceso.

Información para peticiones

- **Punto de acceso:** `http://{server root:5000}/api/v1.0/automation`
- **Método:** POST
- **Cabecera:** Requiere el envío de dos cabeceras: *Content-Type* y *Authorization*; la primera de las cabeceras indica el tipo de datos contenidos en el cuerpo de la petición, el cual ha de ser ***application/json***; la segunda contiene la información de usuario y contraseña en formato ***usuario : contraseña*** y codificado en base64.
- **Cuerpo:** Requiere el envío del estado de cada uno de los recursos domóticos que han sido habilitados en servidor contenidos en una estructura de datos del tipo ResourceRequestType.

Ejemplo de petición para el consumo del servicio.

```
{  
  "energy" : True,  
  "water" : True,  
  "block" : True  
}
```

Información para respuestas

- **Mensajes de error:** Una vez efectuadas las ordenes solicitadas en la petición, el servicio adjunta el código de éxito en la respuesta indicando que la operación se ha completado satisfactoriamente.
- **Cabecera:** No empleado.
- **Cuerpo:** Información del estado de cada uno de los recursos domóticos una vez aplicadas las configuraciones indicadas en la solicitud del servicio contenidos en una estructura de datos del tipo ResourceResponseType.

Ejemplo de respuesta en la respuesta una vez consumido el servicio.

```
{  
  "energy" : True,  
  "water" : True,  
  "block" : True  
}
```

Tipo de datos empleados

| ResourceRequestType | | | |
|---------------------|-------------|--------------|--|
| Variable | Obligatorio | Tipo de dato | Descripción |
| energy | Sí | Boolean | Control de la energía eléctrica. True: Electricidad habilitada. False: Electricidad deshabilitada. |
| water | Sí | Boolean | Control del agua corriente. True: Agua corriente habilitada. False: Agua corriente deshabilitada. |
| block | Sí | Boolean | Control del cierre electrónico de las puertas. True: Cierre activo. False: Cierre inactivo. |

ResourceResponseType

| Variable | Obligatorio | Tipo de dato | Descripción |
|----------|-------------|--------------|---|
| energy | Sí | Boolean | Estado del control de la energía eléctrica. True: Electricidad habilitada. False: Electricidad deshabilitada. |
| water | Sí | Boolean | Estado del control del agua corriente. True: Agua corriente habilitada. False: Agua corriente deshabilitada. |
| block | Sí | Boolean | Estado del control del cierre electrónico de las puertas. True: Cierre activo. False: Cierre inactivo. |

Diagrama de clases del prototipo

La complejidad del sistema puede ser descompuesta en los distintos componentes que lo definen, de forma que las labores de optimización puedan ser realizadas de forma visual.

Esta optimización es posible gracias al diagrama de clases y ficheros desarrollado, el cual no solo define los métodos y atributos para cada una de las clases desarrolladas; sino que especifica cada uno de los archivos que representa al sistema.

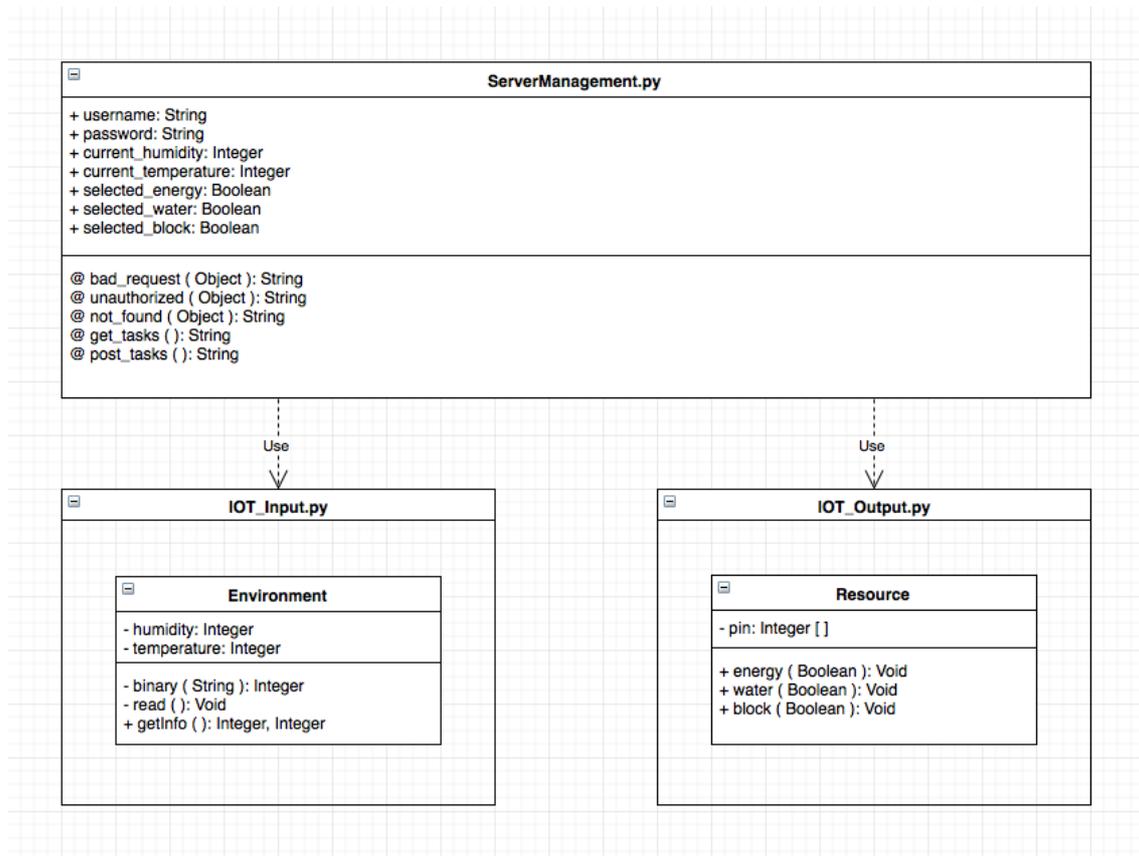


Figura 1. Diagrama de clases y ficheros.

A continuación, se facilita la leyenda relativa al diagrama, facilitando la comprensión de la simbología empleada.

| Leyenda del diagrama | |
|----------------------|------------------------------|
| Símbolo | Descripción |
| + | Métodos y atributos públicos |
| - | Métodos y atributos privados |
| @ | Métodos orientados a eventos |

Esquema eléctrico del prototipo

La implementación del sistema *Internet of Things* requiere la realización de una estructura de componentes electrónicos, de forma que el software sea capaz de recuperar la información del sensor, así como de enviar las órdenes recibidas al relé de cada uno de los recursos domóticos.

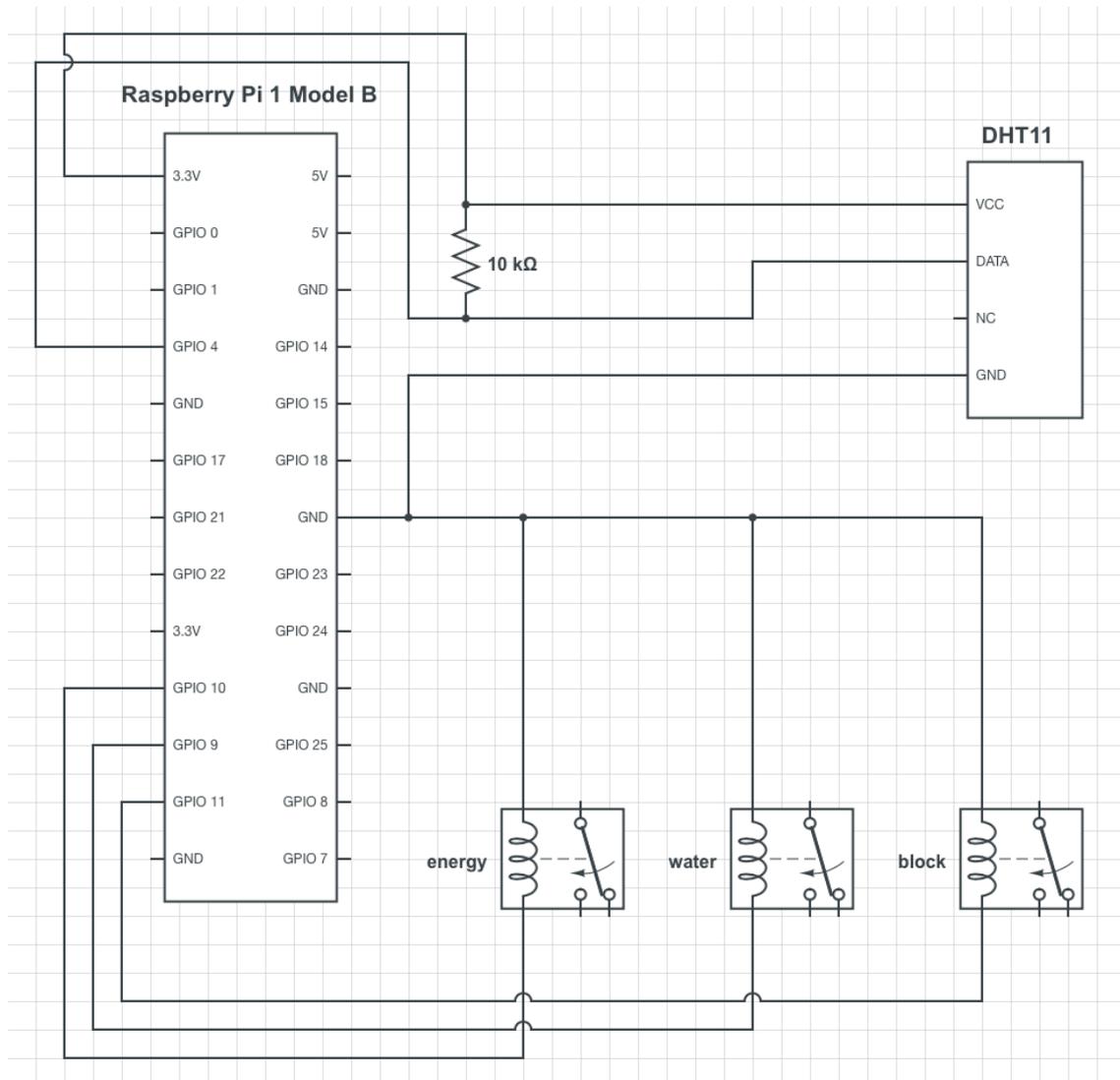


Figura 2. Esquema eléctrico del prototipo.

El esquema eléctrico hace uso de dos componentes electrónicos personalizados como son el módulo GPIO relativo a la Raspberry Pi 1 Modelo B, así como al esquema eléctrico del sensor de humedad y temperatura modelo DHT11.

Capítulo 4: Customer Relationship Management

El sistema *Customer Relationship Management* hace posible la simplificación de todo proceso de negocio relacionado con el cliente, permitiendo en base a las herramientas suministradas por la plataforma, la automatización de la mayoría de ellos.

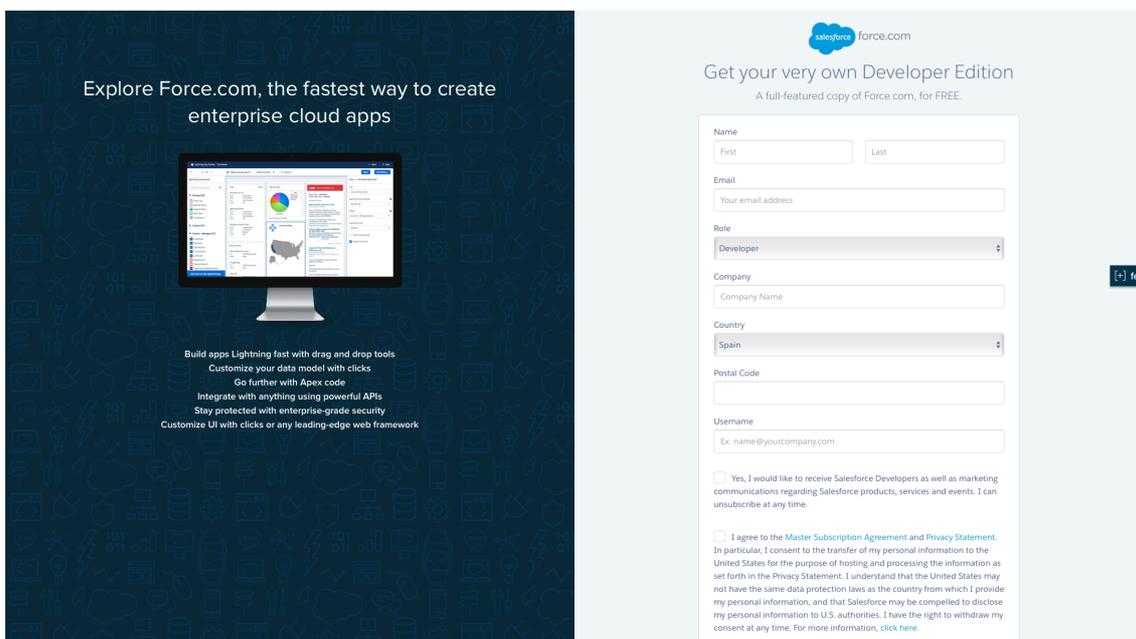
Es posible establecer una clasificación de las herramientas ofrecidas por la plataforma en dos secciones.

La primera se relaciona con las labores de administración del entorno, haciendo posible la generación de campos calculados dinámicamente, así como la definición de ciertas restricciones sobre los datos entre otros muchos aspectos; la segunda se apoya en las labores de desarrollo, haciendo posible el procesamiento de los datos del sistema en base a un lenguaje de programación, estos desarrollos pueden ser lanzados en base a ciertos eventos como crear, modificar o eliminar un registro de datos o bien pulsando sobre un botón en la interfaz.

A continuación, se indica el conjunto de actividades llevadas a cabo para implementar el sistema organizadas en base a los objetos implementados.

Pasos iniciales

El acceso al entorno solo es posible empleando una cuenta de usuario registrada en la plataforma, para ello, se accede a la página Salesforce Developers en la cual dispone del formulario de registro correspondiente; es preciso recordar que Salesforce Developers en la única vía de acceso al entorno de forma gratuita.



Explore Force.com, the fastest way to create enterprise cloud apps

Build apps Lightning fast with drag and drop tools
Customize your data model with clicks
Go further with Apex code
Integrate with anything using powerful APIs
Stay protected with enterprise-grade security
Customize UI with clicks or any leading-edge web framework

Get your very own Developer Edition
A full-featured copy of Force.com, for FREE.

Name
First Last

Email
Your email address

Role
Developer

Company
Company Name

Country
Spain

Postal Code

Username
Ex: name@yourcompany.com

Yes, I would like to receive Salesforce Developers as well as marketing communications regarding Salesforce products, services and events. I can unsubscribe at any time.

I agree to the Master Subscription Agreement and Privacy Statement. In particular, I consent to the transfer of my personal information to the United States for the purpose of hosting and processing the information as set forth in the Privacy Statement. I understand that the United States may not have the same data protection laws as the country from which I provide my personal information, and that Salesforce may be compelled to disclose my personal information to U.S. authorities. I have the right to withdraw my consent at any time. For more information, [click here](#).

Figura 3. Formulario de registro en Salesforce Developers.

Una vez formalizada toda la información solicitada en el proceso de registro, se accede por primera vez a la plataforma, habilitándose la nueva interfaz de usuario Lightning Experience basada en componentes HTML5, CSS3 y JavaScript.

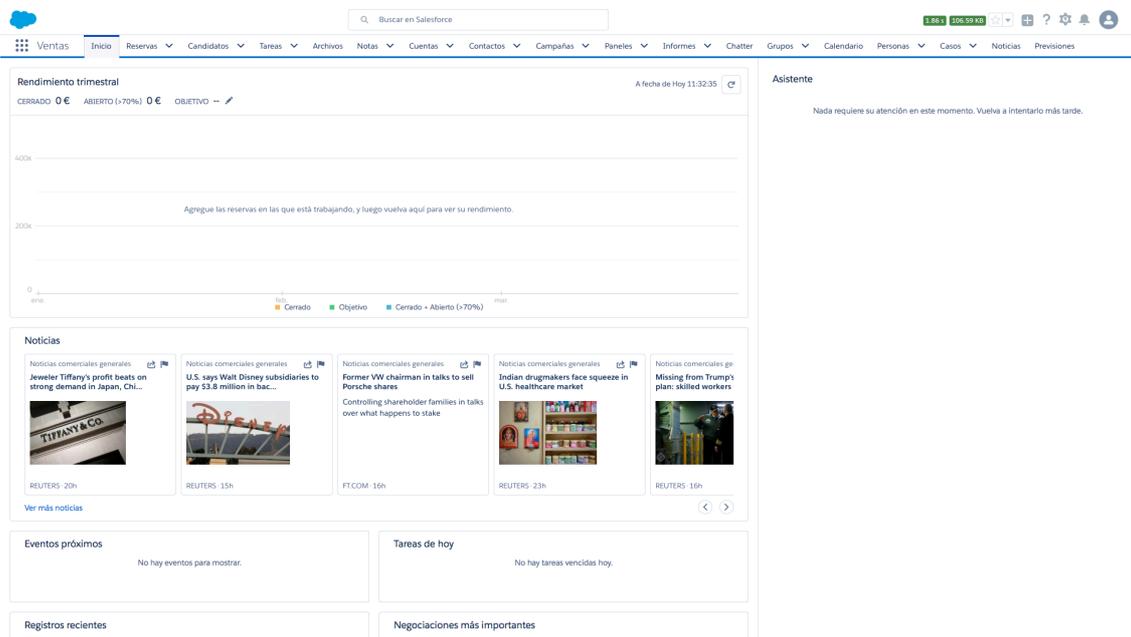


Figura 4. Vista inicial de la interfaz de usuario Lightning Experience.

Es preciso mencionar que toda configuración o seguimiento ha de realizarse a través del panel de configuración del sistema, siendo posible acceder al mismo a través del panel superior.

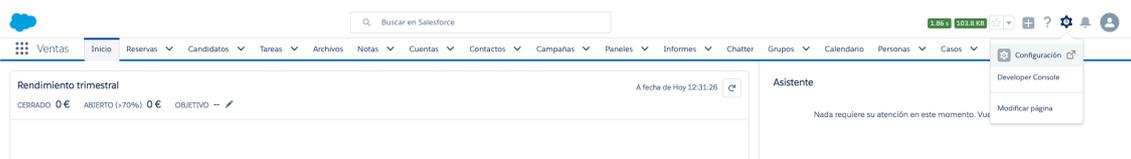


Figura 5. Acceso al panel de configuración del sistema.

La configuración del sistema ofrece al administrador la posibilidad de adaptar cualquier aspecto configurable de la plataforma a las necesidades específicas del negocio; por lo que toda configuración indicada en los puntos posteriores, será efectuada desde dicho panel.

Los puntos adyacentes indican las actividades llevadas a cabo para la implementación del sistema *Customer Relationship Management* acorde a las necesidades especificadas en los requisitos; las actividades son organizadas por objetos, siendo posible una mejor comprensión del comportamiento particular de cada una de las entidades que precisa el negocio.

Implementación de objetos

Uno de los puntos críticos de la plataforma resulta de la gestión de datos, siendo preciso almacenar los datos capturados en distintas entidades llamadas objetos; la plataforma ofrece un conjunto de objetos estándar los cuales representan aquellas entidades de datos comúnmente empleadas en negocios de distintos sectores como son los clientes, oportunidades y contactos, así como la posibilidad de diseñar otros personalizados de forma casi ilimitada.

Contactos

El objeto contactos es un objeto estándar ofrecido por el sistema para gestionar los datos personales de cualquier individuo que guarde relación con la entidad; el proyecto hace uso del objeto para gestionar la información del personal de la organización, así como de sus clientes potenciales.

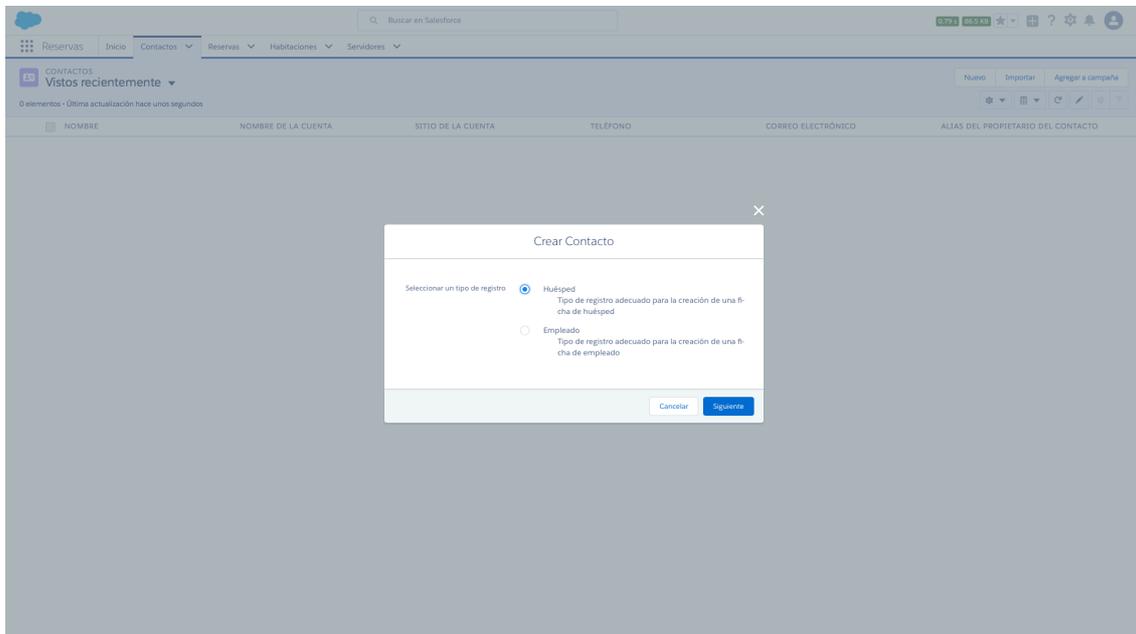


Figura 6. Tipos de contactos disponibles.

En el proceso de creación, el usuario podrá definir el tipo de registro que desea crear, accediendo a la vista de datos correspondiente; los tipos de registro son empleados para aprovechar los desarrollos de un determinado objeto en entidades que guardan cierto parentesco en los datos que desean gestionar.

Campos

La información relativa al personal y a los clientes de la organización es almacenada de forma conjunta en el objeto contactos, siendo preciso configurar de forma conjunta los campos requeridos para dichas entidades, evitando repetir campos.

A continuación, se indican los distintos campos implementados sobre el objeto.

| Contactos | | | | |
|---------------------|-------------|-----------------------------|--------------------|--|
| Objeto estándar | | | | |
| Nombre | Obligatorio | Tipo de dato | Tipo de registro | Particularidades |
| Número de empleado | No | Numeración automática | Empleado | Identificador único. |
| Saludo | No | Lista de selección | Empleado y huésped | Valores de lista habilitados. ✓ Mr. ✓ Ms. ✓ Mrs. ✓ Dr. ✓ Prof. |
| Nombre | No | Texto (40) | Empleado y huésped | |
| Apellidos | Sí | Texto (80) | Empleado y huésped | |
| Segundo nombre | No | Texto (40) | Empleado y huésped | |
| Fecha de nacimiento | Sí | Fecha | Empleado y huésped | |
| Edad | No | Formula (Número) | Empleado y huésped | Valor de campo dinámico. - Fecha de nacimiento |
| Idiomas | Sí | Lista de selección múltiple | Empleado y huésped | Valores de lista habilitados. ✓ Español ✓ Inglés ✓ Francés ✓ Alemán ✓ Portugués ✓ Italiano ✓ Ruso ✓ Chino ✓ Japonés |
| Teléfono | Sí | Teléfono | Empleado y huésped | |
| Correo electrónico | Sí | Email | Empleado y huésped | |
| País | Sí | Lista de selección | Empleado y huésped | Valores de lista habilitados. |

| | | | | |
|--------------------------|----|--------------------|--------------------|---|
| | | | | ✓ España |
| Tipo de documento | Sí | Lista de selección | Empleado y huésped | Valores de lista habilitados. ✓ DNI ✓ Pasaporte |
| Número de documento | Sí | Texto (9) | Empleado y huésped | Valor de campo exclusivo en el sistema. |
| Fecha de incorporación | Sí | Fecha | Empleado | |
| Fecha de baja | No | Fecha | Empleado | |
| Fecha de reincorporación | No | Fecha | Empleado | |
| Seguridad social | Sí | Texto (12) | Empleado | Valor de campo exclusivo en el sistema. |
| Entidad bancaria | Sí | Lista de selección | Empleado | Valores de lista habilitados. ✓ Banesto ✓ Bankia ✓ Bankinter ✓ BBVA ✓ Cajamar ✓ CatalunyaCaixa ✓ Ibercaja ✓ La Caixa ✓ Popular ✓ Sabadell ✓ Santander ✓ Unicaja |
| IBAN | Sí | Texto (24) | Empleado | Valor de campo exclusivo en el sistema. |
| Salario base | Sí | Divisa (7, 2) | Empleado | |
| Complementos salariales | No | Divisa (7, 2) | Empleado | |
| Salario bruto mensual | No | Formula (Divisa) | Empleado | Valor de campo dinámico. - Salario base - Complementos salariales |
| Calle de correo | No | Texto | Empleado y huésped | |

| | | | | |
|------------------------------|-----|-----------------------------|--------------------|--|
| Ciudad de correo | No | Texto | Empleado y huésped | |
| Estado o provincia de correo | No | Texto | Empleado y huésped | |
| Código postal de correo | No | Texto | Empleado y huésped | |
| País de correo | No | Texto | Empleado y huésped | |
| Otra calle | No | Texto | Empleado y huésped | |
| Otra ciudad | No | Texto | Empleado y huésped | |
| Otro estado o provincia | No | Texto | Empleado y huésped | |
| Otro código postal | No | Texto | Empleado y huésped | |
| Otro país | No | Texto | Empleado y huésped | |
| Permiso de conducir | Sí | Lista de selección | Empleado y huésped | Valores de lista habilitados. - Si - No |
| Vehículos | Sí* | Lista de selección múltiple | Empleado y huésped | Campo obligatorio. - Permiso de conducir (*) |
| Tipo de empleado | Sí | Lista de selección | Empleado | Valores de lista habilitados. - Administración - Publicidad - Servicio - Mantenimiento - Limpieza |
| Rango | Sí | Lista de selección | Empleado | Valores de lista habilitados. - Responsable - Líder - Trabajador - Becario |
| Recurso activo | Sí | Casilla | Empleado | |
| Disponible | Sí | Casilla | Empleado | |

| | | | | |
|---------------------------|----|-----------------------|--------------------|----------------------|
| Trabajo realizado (Horas) | No | Número (18, 0) | Empleado | |
| Descripción | No | Texto (320000) | Empleado y huésped | |
| Número de cliente | No | Numeración automática | Huésped | Identificador único. |
| Cliente VIP | Sí | Casilla | Huésped | |

Reglas de validación y restricciones de campo

Atendiendo a los campos disponibles sobre el objeto contactos, es indispensable definir un conjunto de restricciones que evalúen aquellos datos introducidos por el usuario, bloqueando el almacenamiento de aquellos registros sujetos a posibles incoherencias; la coherencia e integridad de los datos puede ser evaluada sin necesidad de desarrollar reglas de validación, configurando correctamente el tipo de dato, la dependencia y la exclusividad de los campos.

La dependencia de campos definida entre los campos *Permiso de conducir* y *Vehículos* condiciona los valores disponibles en el segundo campo, de forma que el usuario ha de seleccionar **Si** en *Permiso de conducir* para poder seleccionar algún tipo de permiso en *Vehículos*.

Las restricciones de exclusividad configuradas sobre los campos *Número de documento*, *Seguridad social* e *IBAN* evitan que el usuario repita el mismo valor en un nuevo registro al considerarse únicos.

Interfaces de usuario

El correcto desempeño sobre el objeto de contactos implica la asignación de un formato de página a cada uno de los tipos de registro previamente configurados, de forma que únicamente se muestren aquellos campos que guardan relación con dicha entidad, y sea posible la asignación de permisos y obligatoriedad de campos.

The screenshot displays the Salesforce interface for a contact named Ms. Juan Fernández. The page is divided into two main sections: 'RELACIONADO' (related) and 'DETALLES' (details). The 'DETALLES' section is further divided into 'Información del contacto' (contact information) and 'Información laboral' (labor information). The 'Información del contacto' section includes fields for 'Número de empleado' (Employee ID: 032), 'Nombre' (Ms. Juan Fernández), 'Segundo nombre' (Javier), 'Fecha de nacimiento' (15/01/1990), and 'Edad' (27). The 'Información laboral' section includes fields for 'País' (España), 'Tipo de documento' (DNI), 'Número de documento' (28704785A), 'Fecha de incorporación' (1/01/2017), and 'Fecha de baja' (500,00 €). The 'Información laboral' section also includes fields for 'Seguridad social' (990123261722), 'Entidad bancaria' (Santander), 'IBAN' (004900210092911223321112), 'Salario base' (1.000,00 €), and 'Complementos salariales' (500,00 €). The 'SALARIO BRUTO MENSUAL' is 1.500,00 €. The 'ACTIVIDAD' and 'CHATTER' sections are also visible, showing 'Filtrar cronología' and 'Sigüientes pasos' (No hay pasos sigüientes. Para poner las cosas en marcha, agregue una tarea o configure una reunión.) and 'Actividad anterior' (No hay actividad anterior. Las tareas y las reuniones anteriores marcadas como realizadas aparecen aquí.).

Figura 7. Formato de página asociado a los registros de empleado.

The screenshot displays the Salesforce interface for a contact named Dr. Luis Lopez. The page is divided into two main sections: 'RELACIONADO' (related) and 'DETALLES' (details). The 'DETALLES' section is further divided into 'Información del contacto' (contact information), 'Información de dirección' (direction information), 'Información personal' (personal information), and 'Información adicional' (additional information). The 'Información del contacto' section includes fields for 'Número de cliente' (Cliente - 034), 'Nombre' (Dr. Luis Lopez), 'Segundo nombre' (Fernando), 'Fecha de nacimiento' (19/03/1990), and 'Edad' (27). The 'Información de dirección' section includes fields for 'Dirección de correo' (Otra dirección). The 'Información personal' section includes fields for 'País' (España), 'Tipo de documento' (DNI), 'Número de documento' (52112485M), and 'Cliente VIP' (checked). The 'Información adicional' section includes fields for 'Permiso de conducir' (SI). The 'ACTIVIDAD' and 'CHATTER' sections are also visible, showing 'Filtrar cronología' and 'Sigüientes pasos' (No hay pasos sigüientes. Para poner las cosas en marcha, agregue una tarea o configure una reunión.) and 'Actividad anterior' (No hay actividad anterior. Las tareas y las reuniones anteriores marcadas como realizadas aparecen aquí.).

Figura 8. Formato de página asociado a los registros de huésped.

Los contactos pueden establecer relaciones con otros objetos como son las reservas y habitaciones, siendo posible visualizar dichas dependencias entre entidades gracias a la pestaña *relacionado* disponible al visualizar los detalles de un registro.

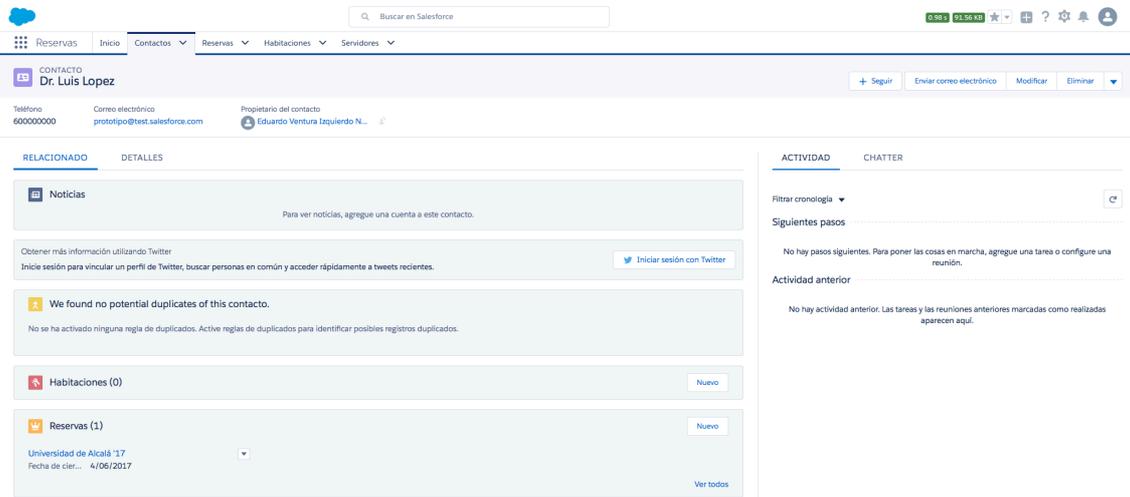


Figura 9. Relaciones activas del huésped seleccionado.

La pestaña indicada permite al usuario de la plataforma realizar un mejor seguimiento de los registros, ya sean empleados o huéspedes, visualizando en una misma página las dependencias activas entre los distintos registros de la organización.

Reservas

El objeto oportunidades es un objeto estándar ofrecido por el sistema destinado a la gestión de ventas y negociaciones pendientes; este es empleado en el proyecto bajo el nombre de *reservas* haciendo posible la gestión del proceso de registro del hotel.

Las oportunidades disponen de un conjunto de etapas sobre las que se identifica el nivel de evolución de la misma; la realización de reservas por etapas, permite completar los datos requeridos de forma escalonada, asegurando el éxito de las negociaciones.

1. **Información básica:** Información crítica para iniciar la reserva.
2. **Asignación de recursos:** Localización de habitaciones adecuadas.
3. **Información de pago:** Información de los medios de pago dispuestos por el cliente.
4. **Reserva asignada:** Asignación de recursos y facturación al cliente.
5. **Reserva cancelada:** Cancelación de la reserva.

La localización y asignación de recursos se realiza de forma transparente y automática al usuario en base a un conjunto de desarrollos comentados en los puntos posteriores.

Campos

La información relativa a las reservas es almacenada en el objeto oportunidades, al igual que en otros objetos, se diseñarán un conjunto de campos dinámicos y restricciones que faciliten y normalicen el desempeño del personal de la organización.

A continuación, se indican los distintos campos implementados sobre el objeto.

| Oportunidades | | | | |
|----------------------|-------------|-------------------------|------------------------|---|
| Objeto estándar | | | | |
| Nombre | Obligatorio | Tipo de dato | Necesario en... | Particularidades |
| Número de reserva | No | Numeración automática | Información básica | Identificador único. |
| Nombre de la reserva | Sí | Texto (120) | Información básica | |
| Contacto principal | Sí | Búsqueda (Contactos) | Información básica | Filtro de búsqueda aplicado sobre el campo. - Empleado |
| Habitación | No | Búsqueda (Habitaciones) | Asignación de recursos | |
| Fecha de inicio | Sí | Fecha | Información básica | |
| Fecha de cierre | Sí | Fecha | Información básica | |

| | | | | |
|-----------------------|----|--------------------|------------------------|---|
| Etapa | Sí | Lista de selección | N/A | Valores de lista habilitados. <ul style="list-style-type: none"> ✓ Información básica ✓ Asignación de recursos ✓ Información de pago ✓ Reserva asignada ✓ Reserva cancelada |
| Probabilidad (%) | No | Automático | N/A | Valor de campo dinámico. <ul style="list-style-type: none"> - Etapa |
| Adultos | Sí | Número (2, 0) | Asignación de recursos | |
| Niños | No | Número (2, 0) | Asignación de recursos | |
| Bebés | No | Número (2, 0) | Asignación de recursos | |
| Total de inquilinos | No | Formula (Número) | Asignación de recursos | Valor de campo dinámico. <ul style="list-style-type: none"> - Adultos - Niños - Bebés |
| Medio de pago | No | Lista de selección | Información de pago | Valores de lista habilitados. <ul style="list-style-type: none"> ✓ Efectivo ✓ Tarjeta de crédito |
| Importe | No | Fórmula (Divisa) | Asignación de recursos | Valor de campo dinámico. <ul style="list-style-type: none"> - Habitación - Fecha de inicio - Fecha de cierre - Cuantía de descuento |
| Cuantía abonada | No | Casilla | Información de pago | |
| Descuento o promoción | Sí | Lista de selección | N/A | Valores de lista habilitados. |

| | | | | |
|----------------------|-----|--------------------|---------------------|--|
| | | | | <input checked="" type="checkbox"/> Sí <input checked="" type="checkbox"/> No |
| Cuantía de descuento | Sí* | Porcentaje | N/A | Campo obligatorio. - Descuento o promoción (*) |
| Tipo de tarjeta | Sí* | Lista de selección | Información de pago | Campo obligatorio. - Medio de pago (*) Valores de lista habilitados. <input checked="" type="checkbox"/> Visa <input checked="" type="checkbox"/> Mastercard |
| Número de tarjeta | Sí* | Texto (16) | Información de pago | Campo obligatorio. - Medio de pago (*) |
| Fecha de caducidad | Sí* | Fecha | Información de pago | Campo obligatorio. - Medio de pago (*) |
| CVV | Sí* | Texto (3) | Información de pago | Campo obligatorio. - Medio de pago (*) |
| Descripción | No | Texto (320000) | N/A | |

Reglas de validación y restricciones de campo

Los datos introducidos por el usuario pueden estar sujetos a error, siendo necesario definir un conjunto de controles que aseguren la coherencia entre los campos, así como con respecto a la lógica de negocio aplicada en la organización.

Además de la configuración del tipo de datos para asegurar la integridad de los datos, ha sido precisa la relación entre los campos *Etapa* y *Probabilidad (%)* permitiendo la asignación dinámica de valores del segundo en función del primero.

La lógica de negocio ha de verse reflejada en la evolución de las oportunidades, por lo que es preciso establecer un conjunto de restricciones basadas en reglas de validación que dirijan a los usuarios en la formulación de las reservas.

Las reglas de validación implementadas aseguran la correcta gestión de las reservas, de forma que todo registro que no cumple con las condiciones especificadas en las fórmulas de las reglas, es bloqueado y notificado al usuario para su modificación.

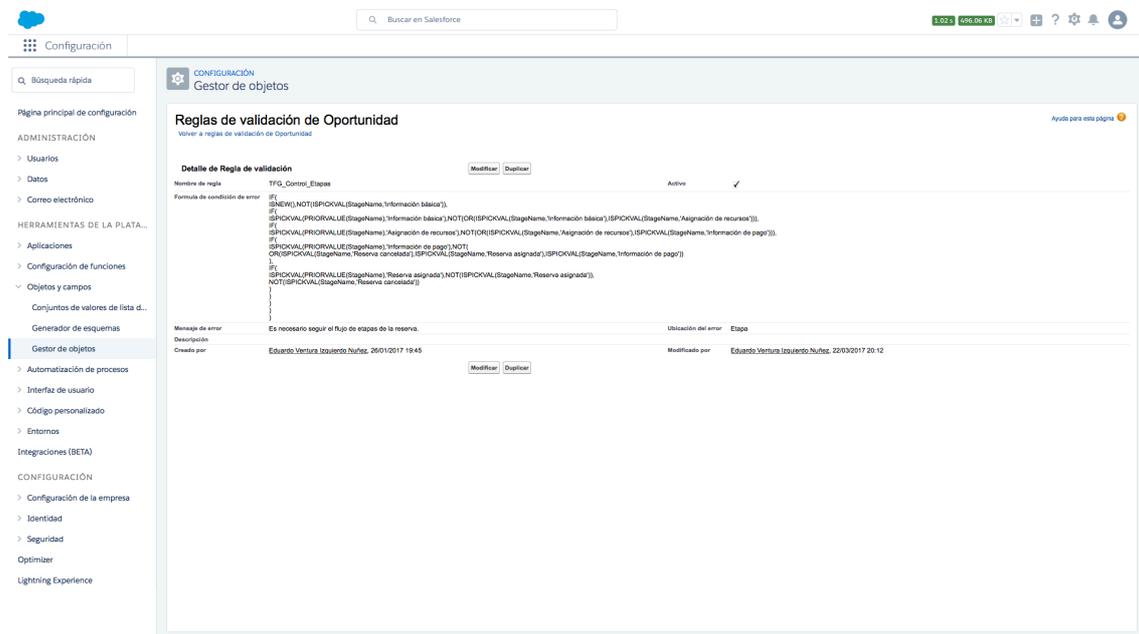


Figura 10. Formato empleado para la implementación de reglas de validación.

A continuación, muestra el listado de reglas de validación implementadas en torno a las reservas, asegurando el cumplimiento de la lógica de negocio de la organización.

| Reglas de validación | |
|----------------------|--|
| Reservas | |
| Nombre de la regla | Descripción detallada |
| TFG_Control_Etapas | <p>Condición de error: No es posible alterar el flujo de etapas definido en las reservas, bloqueándose los saltos de etapa y retrocesos.</p> <p>* Campos implicados: <i>Etapas.</i></p> |
| TFG_Fecha_Inicio | <p>Condición de error: No es posible configurar una fecha de inicio anterior al día actual en la reserva.</p> <p>* Campos implicados: <i>Fecha de inicio.</i></p> |
| TFG_Fecha_Final | <p>Condición de error: No es posible definir una fecha de cierre anterior a la fecha de inicio configurada en la reserva.</p> <p>* Campos implicados: <i>Fecha de inicio, Fecha de cierre.</i></p> |
| TFG_Adultos | <p>Condición de error: No es posible configurar reservas en las que no exista al menos un adulto.</p> |

| | |
|------------------------|--|
| | <p>* Campos implicados: <i>Adultos.</i></p> |
| TFG_Ninos | <p>Condición de error: No es posible introducir valores negativos.</p> |
| | <p>* Campos implicados: <i>Niños.</i></p> |
| TFG_Bebes | <p>Condición de error: No es posible introducir valores negativos.</p> |
| | <p>* Campos implicados: <i>Bebés.</i></p> |
| TFG_Cuantia_Descuento | <p>Condición de error: No es posible asignar un descuento o promoción a la reserva sin definir su cuantía.</p> |
| | <p>* Campos implicados: <i>Descuento o promoción, Cuantía de descuento.</i></p> |
| TFG_Informacion_Pago | <p>Condición de error: No es posible avanzar la etapa de información de pago sin informar todos los campos dependientes del medio seleccionado.</p> |
| | <p>* Campos implicados: <i>Etapa, Medio de pago, Tipo de tarjeta, Número de tarjeta, Fecha de caducidad, CVV.</i></p> |
| TFG_Numero_Tarjeta | <p>Condición de error: No es posible configurar números de tarjeta de crédito que dispongan de longitud distinta a dieciséis o involucren caracteres alfabéticos.</p> |
| | <p>* Campos implicados: <i>Número de tarjeta.</i></p> |
| TFG_Fecha_Caducidad | <p>Condición de error: No es posible configurar medios de pago cuya caducidad sea previa a la fecha de finalización de la reserva.</p> |
| | <p>* Campos implicados: <i>Fecha de cierre, Fecha de caducidad.</i></p> |
| TFG_CVV | <p>Condición de error: No es posible configurar números de CVV que dispongan de longitud distinta a tres o involucren caracteres alfabéticos.</p> |
| | <p>* Campos implicados: <i>CVV.</i></p> |
| TFG_Habitacion | <p>Condición de error: No es posible avanzar a la etapa de información de pago sin asignar una habitación.</p> |
| | <p>* Campos implicados: <i>Etapa, Habitación.</i></p> |
| TFG_Total_Inquilinos | <p>Condición de error: No es posible modificar el número de inquilinos una vez asociada la habitación.</p> |
| | <p>* Campos implicados: <i>Habitación, Total de inquilinos.</i></p> |
| TFG_Cuantia_Habitacion | <p>Condición de error: No es posible abonar la cuantía de la reserva sin configurar una habitación.</p> |
| | <p>* Campos implicados: <i>Habitación, Cuantía abonada.</i></p> |
| TFG_Cuantia_Abonada | <p>Condición de error: No es posible cerrar una reserva sin abonar el importe de la misma.</p> |

| | |
|---------------------|---|
| | * Campos implicados: <i>Etapa, Cuantía abonada.</i> |
| TFG_Gestion_Reserva | Condición de error: No es posible efectuar cambios en las reservas cerradas. |
| | * Campos implicados: <i>Etapa.</i> |

Interfaces de usuario

El formato de página desarrollado para la gestión de las reservas dispone al igual que para contactos, una organización de campos basada en secciones, lo que hace posible la rápida localización de campos en función de su categoría.

The screenshot shows a web application interface for managing reservations. The main content area is titled 'RESERVA Universidad de Alcalá '17'. It features a navigation bar with tabs for 'ACTIVIDAD', 'CHATTER', and 'DETALLES'. The 'DETALLES' tab is active, displaying a form with the following sections:

- Información de la reserva:**
 - Número de reserva: Reserva - 094
 - Fecha de inicio: 3/06/2017
 - Nombre de la reserva: Universidad de Alcalá '17
 - Fecha de cierre: 4/06/2017
 - Contacto principal: Luis Lopez
 - Etapas: Información de pago
 - Habitación: Grand Royal Suite III
 - Probabilidad (%): 75 %
- Otra información:**
 - Adultos: 2
 - Total de inquilinos: 2
 - Niños: 0
 - Bebés: 0
- Información de pago:**
 - Medio de pago: Efectivo
 - Tipo de tarjeta: Visa
 - Importe: 1.125,00 €
 - Número de tarjeta: [redacted]
 - Cuánta abonada:
 - Fecha de caducidad: [redacted]
 - CVV: [redacted]

On the right side, there are two panels: 'Noticias' (Noticias) and 'Notas y archivos adjuntos (0)'. The 'Noticias' panel shows a message: 'No hay perspectivas de Data.com disponibles para esta empresa.' and 'No hay elementos de noticias disponibles en este momento.'

Figura 11. Formato de página asociado a las reservas.

Es recomendable el uso de objetos estándar; permitiendo al usuario consumir aquellos componentes desarrollados por el proveedor y logrando como consecuencia interfaces más funcionales y estéticas.

Habitaciones

El objeto habitaciones es un objeto personalizado definido por el administrador del sistema con el objetivo de mantener toda la información de dichos recursos accesible a todos los niveles de la organización.

Los registros de las habitaciones, opcionalmente, podrán asociar un sistema de control domótico a las habitaciones, posibilitando la gestión remota de los recursos críticos de cada habitación, así como un seguimiento actualizado de las condiciones de la misma.

Campos

Algunos campos definidos en el objeto de habitaciones son tomados por la plataforma, haciendo posible la automatización de ciertos procesos de negocio como la asignación de recursos a las reservas.

Los registros de las habitaciones cuentan con un conjunto de relaciones a registros de otras entidades, permitiendo realizar un seguimiento óptimo de aquellas entidades con las que se relaciona en un determinado instante.

A continuación, se indica el conjunto de campos que constituye el objeto.

| Habitaciones | | | |
|--------------------------|-------------|-------------------|---|
| Objeto personalizado | | | |
| Variable | Obligatorio | Tipo de dato | Descripción |
| Nombre | Sí | Texto (80) | |
| Responsable | No | Buscar (Contacto) | Filtro de búsqueda aplicado sobre el campo. - Empleado |
| Máximo de inquilinos | Sí | Número (1, 0) | |
| Tamaño (m ²) | Sí | Número (3, 2) | |
| Precio | Sí | Divisa (4, 2) | |
| Última limpieza | No | Fecha y hora | |
| Movilidad reducida | No | Casilla | |
| Apto para niños | No | Casilla | |
| Chimenea | No | Casilla | |
| Aire acondicionado | No | Casilla | |
| Internet | No | Casilla | |
| Televisión | No | Casilla | |

| | | | |
|-------------------------|----|------------------------|---|
| Televisión por satélite | No | Casilla | |
| Champú | No | Casilla | |
| Secador de pelo | No | Casilla | |
| Plancha | No | Casilla | |
| Código de habitación | No | Formula (Texto) | Valor de campo dinámico. - Piso - Letra |
| País | Sí | Lista de selección | Valores de lista habilitados. ✓ España |
| Provincia | No | Texto (40) | |
| Ciudad | Sí | Texto (40) | |
| Calle | No | Texto (255) | |
| Número | No | Número (3, 0) | |
| Piso | Sí | Número (2, 0) | |
| Letra | Sí | Texto (2) | |
| Temperatura (°C) | No | Fórmula (Número) | Valor de campo dinámico. - Servidor |
| Humedad | No | Fórmula (Número) | Valor de campo dinámico. - Servidor |
| Última actualización | No | Fórmula (Fecha y hora) | Valor de campo dinámico. - Servidor |
| Energía | No | Fórmula (Casilla) | Valor de campo dinámico. - Servidor |
| Agua | No | Fórmula (Casilla) | Valor de campo dinámico. - Servidor |
| Puerta | No | Fórmula (Casilla) | Valor de campo dinámico. - Servidor |
| Servidor | No | Buscar (Servidor) | Filtro de búsqueda aplicado sobre el campo. - Funcionamiento |

Interfaces de usuario

La interfaz de usuario desarrollada para la administración de las habitaciones cuenta, al igual que para aquellos objetos ya descritos, con una estructura basada en secciones, facilitando la localización de la información en base a distintas categorías.

The screenshot shows a web application interface for managing rooms. At the top, there is a navigation bar with a search field and a menu with options: Reservas, Inicio, Contactos, Reservas, Habitaciones, and Servidores. Below the navigation bar, the page title is 'HABITACION Grand Royal Suite III'. The main content area is divided into two tabs: 'RELACIONADO' and 'DETALLES', with 'DETALLES' selected. The details section is organized into several expandable sections:

- Información:** Includes fields for 'Nombre' (Grand Royal Suite III), 'Responsable' (Juan Fernández), 'Máximo de Inquilinos' (4), 'Tamaño (m²)' (250,00), and 'Precio' (1.500,00 €).
- Propiedades de la Habitación:** A list of amenities with checkboxes, including 'Muebles reducidos', 'Apto para niños', 'Chimenea', 'Aire acondicionado', 'Internet', 'Televisión', 'Televisión por satélite', 'Champú', 'Secador de pelo', and 'Plancha'.
- Información de Localización:** Includes 'Código de Habitación' (3A), 'País', 'Calle' (Gran vía), and 'Número'.

Figura 12. Formato de página asociado a las habitaciones.

La particularidad en esta interfaz consiste en la posibilidad de visualizar de forma rápida todo el conjunto de datos del sistema domótico sin necesidad de acceder a la interfaz.

This screenshot shows the same user interface as Figure 12, but with the 'DETALLES' tab expanded to show smart control (domotic) fields. The 'Información' and 'Propiedades de la Habitación' sections are collapsed. The 'Información de Localización' section is also collapsed. The 'Información de la Habitación' section is expanded, showing:

- Temperatura (°C):** Última actualización
- Humedad:**
- Asistencia Remota:** Includes checkboxes for 'Energía', 'Agua', and 'Puerta', all of which are checked.
- Domótica:** Includes a 'Servidor' field with the value 'Servidor-013'.

Figura 13. Campos asociados al sistema de control domótico.

El valor de los campos asociados al sistema domótico será actualizado en función del valor que toman los correspondientes campos del objeto de servidores, manteniendo ambos registros alineados.

Servidores

El objeto servidores en un objeto personalizado, configurado con el fin de centralizar los controles domóticos de cada una de las habitaciones que dispongan de dicha capacidad habilitada.

Es preciso remarcar la importancia del objeto para el consumo de los servicios, al ser el responsable de orquestar la ejecución de los métodos desarrollados para la explotación de los servicios en red expuestos por los distintos servidores.

Campos

Los campos configurados en el objeto servidores permiten al usuario la configuración del acceso a los servicios, así como la capacidad de explotación de los mismos; siendo indispensable el almacenamiento de los datos relativos a las credenciales de usuario, así como aquellos referentes al direccionamiento de red.

A continuación, se indican los campos configurados en la implementación del objeto.

| Servidores | | | |
|--------------------------|-------------|-----------------------|-------------------------------|
| Objeto personalizado | | | |
| Variable | Obligatorio | Tipo de dato | Descripción |
| Nombre | No | Numeración automática | Identificador único. |
| Nodo de direccionamiento | Sí | Texto (15) | |
| Dirección de red | Sí | Texto (15) | |
| Usuario | Sí | Texto (25) | |
| Contraseña | Sí | Texto (Cifrado) (25) | |
| Fecha de instalación | No | Fecha | |
| Energía | No | Casilla | |
| Agua | No | Casilla | |
| Puerta | No | Casilla | |
| Temperatura (°C) | No | Número (2, 2) | |
| Humedad | No | Porcentaje (3, 2) | |
| Última actualización | No | Fecha y hora | |
| Habilitado | No | Casilla | |
| Funcionamiento | No | Lista de selección | Valores de lista habilitados. |

| | | | |
|------------------------------|----|--------------|-------------------------------|
| | | | ✓ Correcto ✓ Mantenimiento |
| Fecha estimada de resolución | No | Fecha y hora | |

Reglas de validación y restricciones de campo

La plataforma ha de mantener un control de las conexiones con el servidor, previniendo cualquier acceso a los servicios de aquellos servidores cuyo estado sea *Mantenimiento* o bien no estén habilitados en la plataforma.

No es posible configurar este tipo de controles mediante el uso de reglas de validación, por lo que ha sido preciso definirlos a nivel del desencadenador de apex.

```

/*-----
Author:          Eduardo Ventura
Company:         Universidad de Alcalá
Description:     Apex Trigger

History:

<Date>          <Author>          <Change Description>
02/02/2017      Eduardo Ventura    Initial Version
-----*/

```

trigger TFG_Servidor on Servidor__c (before insert, before update, before delete, after insert, after update, after delete) {

```

    if(trigger.isBefore){
        /* TRIGGER IS BEFORE */
        if(trigger.isInsert){
            /* Trigger is Before Insert */
        }else if(trigger.isUpdate){
            /* Trigger is Before Update */
            /* C001 */
            Map<Id, Servidor__c> oldValues = new Map<Id, Servidor__c>(trigger.old);
            for(Servidor__c item : trigger.new){
                if(oldValues.get(item.Id).TFG_Energia__c != item.TFG_Energia__c ||
oldValues.get(item.Id).TFG_Agua__c != item.TFG_Agua__c ||
oldValues.get(item.Id).TFG_Puerta__c != item.TFG_Puerta__c){
                    if(item.TFG_Habilitado__c && item.TFG_Funcionamiento__c ==
'Correcto') TFG_ServidorMethods.changesToRaspberry(item.Id);
                    else item.addError(' El servidor se encuentra en mantenimiento o
está deshabilitado; inténtelo de nuevo más tarde.');
```

```

        /* C001 */
    }else if(trigger.isUpdate){
        /* Trigger is After Update */
    }else{
        /* Trigger is After Delete */
    }
}
}
}

```

En caso de no cumplirse las condiciones necesarias para acceder a los métodos de apex relativos al consumo de los servicios web, se notificará al usuario a través de la interfaz, de forma que pueda identificar rápidamente las causas del bloqueo.

Interfaces de usuario

La interfaz de usuario desarrollada para la administración de los servidores cuenta con una estructura de campos basada en un total de cuatro secciones, identificando aquellos puntos de acción que dispone el usuario.

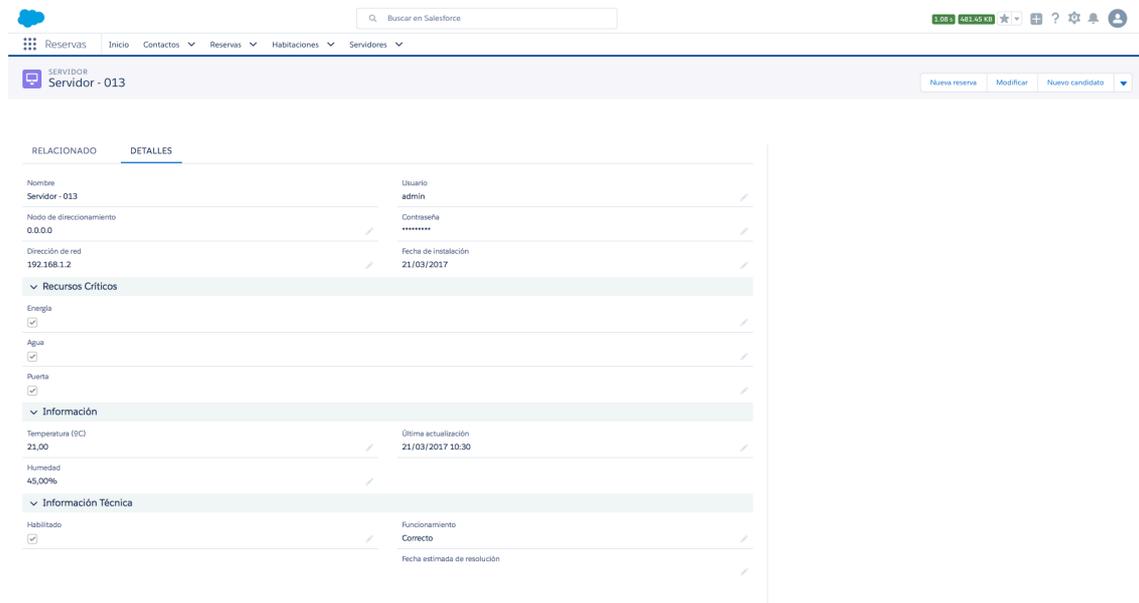


Figura 14. Formato de página asociado a los servidores.

La primera sección de campos, permite al usuario configurar la ruta de direccionamiento con el servidor, así como el ajuste de las credenciales de usuario, indispensables para el acceso OAuth requerido en la ejecución de acciones domóticas.

La segunda y tercera sección permiten al usuario interactuar con los servicios ofrecidos por el servidor, permitiendo el control de los recursos domóticos, así como la consulta de información ambiental.

La cuarta sección se destina a las labores de mantenimiento, permitiendo habilitar o deshabilitar el consumo de los servicios en base al estado del servidor.

Asistente de asignación de recursos

La asignación de habitaciones entre las distintas reservas se realiza de forma guiada, permitiendo al usuario localizar la habitación más adecuada para las necesidades de un cliente en particular.

El asistente realiza un conjunto de validaciones sobre el conjunto de habitaciones activas en la organización, eliminando aquellas que no cumplan las necesidades especificadas en la reserva o bien aquellos parámetros definidos por el usuario a través de la interfaz del asistente.

Validación de registros

La capacidad de automatización de los procesos de filtrado requiere de un conjunto de datos básicos solicitados en la creación de las reservas; una vez formalizados, el usuario es capaz de acceder a la etapa de asignación de recursos donde dispondrá de capacidad de acceso al asistente de asignación de recursos a través de la interfaz del registro.

Existe una fase previa a la inicialización del asistente en la que se realizan un conjunto de validaciones para asegurar la coherencia en los registros que acceden al desarrollo.

| Validación de registros | |
|-------------------------------------|--|
| Asistente de asignación de recursos | |
| Nombre de la regla | Descripción detallada |
| Validación de etapa | Condición de advertencia: No es posible acceder al componente si la reserva no se encuentra en la etapa de asignación de recursos. * Campos implicados: <i>Etapa.</i> |
| Validación de recursos | Condición de advertencia: No es posible acceder al componente si la reserva cuenta con una habitación previamente asignada. * Campos implicados: <i>Habitación.</i> |

En caso de no asegurarse la coherencia de los registros a través de los cuales se accede al asistente, se notifica al usuario con un mensaje de advertencia y se retrocede a la vista original de la reserva.

El acceso al asistente no supone alteración de datos alguna, no siendo posible realizar las validaciones comentadas a través de reglas de validación o desencadenador de apex; requiriendo actuar a nivel del lanzador del desarrollo.

TFG_Asignacion_Reservas.vfp

El asistente de asignación de recursos es construido sobre la siguiente página visualforce donde previamente se realiza una validación de los datos de la reserva que desencadena la ejecución del recurso.

```
<apex:page standardController="Opportunity"
extensions="TFG_Asignacion_Reservas_Extension">
  <apex:includeLightning />
  <div id="LightningComponent"/>

  <script>
    var warning = '';
    if('{!opportunity.StageName}' != 'Asignación de recursos') warning = 'La
reserva en curso no se encuentra en la etapa adecuada; actualice la reserva y vuelva
a intentar acceder a la aplicación.';
    else if('{!opportunity.TFG_Habitacion__c}' != '') warning = 'La reserva en
curso ya dispone de una habitación asignada; actualice la reserva y vuelva a intentar
acceder a la aplicación.';

    $Lightning.use("c:TFG_Asignacion_Reservas_App", function() {
      $Lightning.createComponent(
        "c:TFG_Asignacion_Reservas",
        {reserva : '{!opportunity.Id}', warning: warning},
        "LightningComponent",
        function(cmp) {
        });
    });
  </script>
</apex:page>
```

TFG_Asignacion_Reservas_Extension.apxc

La recuperación de información de la reserva es posible gracias a la siguiente extensión, donde se asigna la instancia de dicha reserva a una variable accesible desde la página visualforce.

```
public class TFG_Asignacion_Reservas_Extension {
  public final Opportunity opportunity;

  public TFG_Asignacion_Reservas_Extension(ApexPages.StandardController
stdController) {
    this.opportunity = (Opportunity) stdController.getRecord();
  }
}
```

Aura Definition Bundle

Una vez ejecutadas las validaciones sobre los datos de la reserva a través de la página visualforce y obtenido el correspondiente mensaje de advertencia, se generan aquellos componentes relacionados con el asistente aportando como parámetros el identificador de la reserva y el mensaje de advertencia.

La inicialización de los componentes que guardan relación con el asistente hace posible la asignación de los datos enviados por el lanzador a las correspondientes variables de dichos componentes, manteniendo toda la información necesaria fácilmente accesible.

Notificación de advertencias

En caso de informarse el campo relacionado con el mensaje de advertencia, la interfaz habilitará un cuadro de diálogo personalizado; informando al usuario de las condiciones que bloquean el acceso al asistente y bloqueando cualquier acción sobre el mismo.

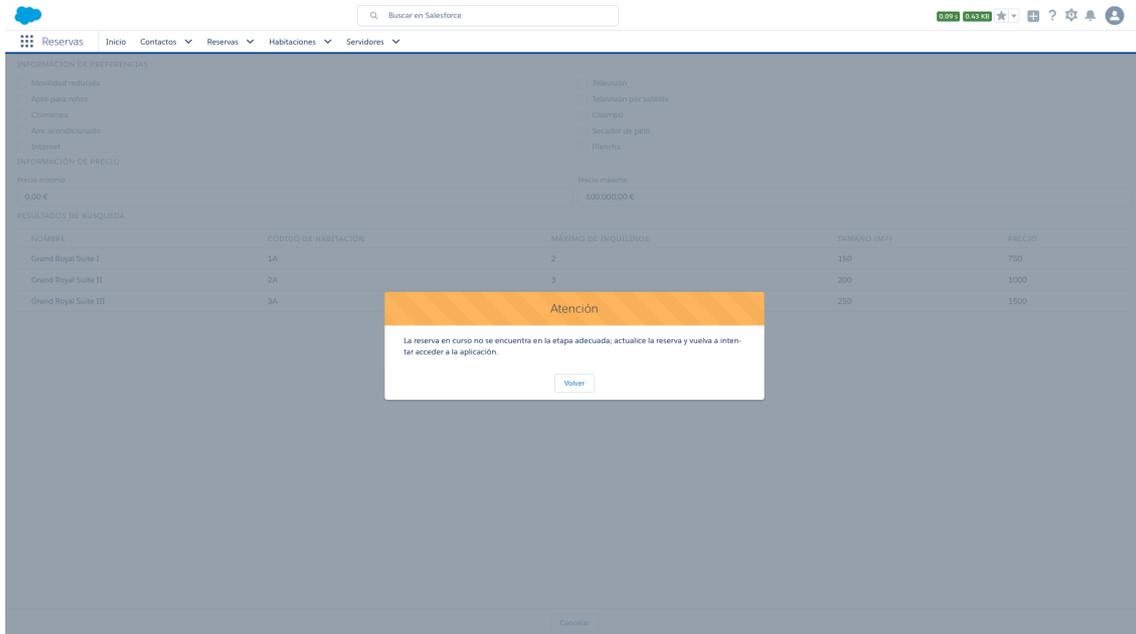


Figura 15. Cuadro de diálogo destinado a las advertencias.

En caso de notificación de advertencia, el usuario únicamente podrá recuperar la vista original de la reserva permitiendo realizar los cambios necesarios para iniciar el asistente correctamente.

Asistente de asignación de recursos

El componente implementado ofrece al usuario la posibilidad de asignar habitaciones a través de una interfaz sencilla; la interfaz sigue los principios estéticos recomendados en la guía de diseño de componentes, de la misma forma que las implementaciones se basan en el framework ofrecido por Salesforce para la implementación de componentes Lightning Experience, obteniendo como resultado un componente Lightning Experience completamente integrado en el sistema, capaz de ser actualizado conforme a las nuevas versiones del entorno.

El asistente de asignación de recursos queda representado por tres subcomponentes, descomponiendo su complejidad en distintos elementos, los cuales se relacionan entre sí gracias a la estructura propia de los elementos Aura Definition Bundle.

TFG_Asignacion_Reservas.cmp

Este subcomponente permite la declaración de la apariencia del componente, así como la declaración de variables y otras estructuras de gestión de datos las cuales dotan de dinamismo al componente.

A continuación, se facilita el código desarrollado.

```
<aura:component access="GLOBAL" controller="TFG_Asignacion_Reservas_Controller"
implements="force:lightningQuickAction">
  <!-- Attributes -->
  <aura:attribute name="habitaciones" type="Habitacion__c[]"/>
  <aura:attribute name="habitacion" type="Habitacion__c"/>

  <!-- INIT Attributes -->
  <aura:attribute name="reserva" type="String"/>
  <aura:attribute name="warning" type="String"/>

  <!-- INIT -->
  <aura:handler name="init" value="{!this}" action="{!c.doInit}"/>

  <!-- Error -->
  <aura:if isTrue="{!v.warning != ''}">
    <div role="alertdialog" tabIndex="-1" aria-labelledby="prompt-heading-id"
aria-describedby="prompt-message-wrapper" class="slds-modal slds-fade-in-open slds-
modal--prompt">
      <div class="slds-modal__container">
        <div class="slds-modal__header slds-theme--warning slds-theme--
alert-texture">
          <h2 class="slds-text-heading--medium" id="prompt-heading-
id">Atención</h2>
        </div>
        <div class="slds-modal__content slds-p-around--medium">
          <p>{!v.warning}</p>
        </div>
        <div class="slds-modal__footer slds-theme--default">
          <button class="slds-button slds-button--neutral"
onclick="{!c.cancel}">Volver</button>
        </div>
      </div>
    </div>
    <div class="slds-backdrop slds-backdrop--open"></div>
  </aura:if>

  <!-- Lightning Form -->
  <div>
    <fieldset class="slds-form--compound">
      <legend style="margin-bottom: 15px" class="slds-form-element__label slds-
text-title--caps">Información de preferencias</legend>
      <div class="slds-form-element__group">
        <div class="slds-form-element__row">
          <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
              <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Movilidad_Reducida" type="checkbox" label="Movilidad reducida"
name="TFG_Movilidad_Reducida" />
            </div>
          </div>
          <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
              <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Televisión" type="checkbox" label="Televisión" name="TFG_Televisión" />
            </div>
          </div>
        </div>
      </div>
    </fieldset>
  </div>
```

```

        </div>
    </div>
    <div class="slds-form-element__row">
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Apto_Ninos" type="checkbox" label="Apto para niños"
name="TFG_Apto_Ninos" />
            </div>
        </div>
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <div class="slds-form-element__control">
                    <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Television_Satelite" type="checkbox" label="Televisión por satélite"
name="TFG_Television_Satelite" />
                </div>
            </div>
        </div>
    </div>
    <div class="slds-form-element__row">
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Chimenea" type="checkbox" label="Chimenea" name="TFG_Chimenea" />
            </div>
        </div>
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Champu" type="checkbox" label="Champú" name="TFG_Champu" />
            </div>
        </div>
    </div>
    <div class="slds-form-element__row">
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Aire_Acondicionado" type="checkbox" label="Aire acondicionado"
name="TFG_Aire_Acondicionado" />
            </div>
        </div>
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Secador_Pelo" type="checkbox" label="Secador de pelo"
name="TFG_Secador_Pelo" />
            </div>
        </div>
    </div>
    <div class="slds-form-element__row">
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Internet" type="checkbox" label="Internet" name="TFG_Internet" />
            </div>
        </div>
        <div class="slds-form-element slds-size--1-of-2">
            <div class="slds-form-element">
                <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Plancha" type="checkbox" label="Plancha" name="TFG_Plancha" />
            </div>
        </div>
    </div>
</div>
</fieldset>

```

```

<fieldset class="slds-form--compound">
  <legend style="margin-bottom: 15px" class="slds-form-element__label slds-
text-title--caps">Información de precio</legend>
  <div class="slds-form-element__group">
    <div class="slds-form-element__row">
      <div class="slds-form-element slds-size--1-of-2">
        <div class="slds-form-element">
          <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Precio_Minimo" type="number" name="TFG_Precio_Minimo" label="Precio
mínimo" value="0" formatter="currency"/>
        </div>
      </div>
      <div class="slds-form-element slds-size--1-of-2">
        <div class="slds-form-element">
          <lightning:input onchange="{!c.loadPreferences}"
aura:id="TFG_Precio_Maximo" type="number" name="TFG_Precio_Maximo" label="Precio
máximo" value="100000" formatter="currency"/>
        </div>
      </div>
    </div>
  </div>
</fieldset>
<fieldset class="slds-form--compound">
  <legend style="margin-bottom: 15px" class="slds-form-element__label slds-
text-title--caps">Resultados de búsqueda</legend>
  <div class="slds-form-element__group">
    <div class="slds-form-element__row">
      <table class="slds-table slds-table--bordered slds-table--cell-
buffer">
        <thead>
          <tr class="slds-text-title--caps">
            <th scope="col">
              <div class="slds-truncate"
title="TFG_Nombre">Nombre</div>
            </th>
            <th scope="col">
              <div class="slds-truncate"
title="TFG_Codigo_Habitacion">Código de Habitación</div>
            </th>
            <th scope="col">
              <div class="slds-truncate"
title="TFG_Maximo_Inquilinos">Máximo de inquilinos</div>
            </th>
            <th scope="col">
              <div class="slds-truncate"
title="TFG_Tamano">Tamaño (m²)</div>
            </th>
            <th scope="col">
              <div class="slds-truncate"
title="TFG_Precio">Precio</div>
            </th>
          </tr>
        </thead>
        <tbody>
          <!-- Search Results -->
          <aura:iteration items="{!v.habitaciones}" var="item"
indexVar="index">
            <tr style="cursor: pointer" onclick="{!c.pickRoom}"
data-index="{!index}">
              <th scope="row" data-label="TFG_Nombre">
                <div class="slds-truncate"
title="{!item.Name}">{!item.Name}</div>
              </th>
              <td data-label="TFG_Codigo_Habitacion">
                <div class="slds-truncate"
title="{!item.TFG_Codigo_Habitacion__c}">{!item.TFG_Codigo_Habitacion__c}</div>

```

```

        </td>
        <td data-label="TFG_Maximo_Inquilinos">
            <div class="slds-truncate"
title="{!item.TFG_Maximo_Inquilinos__c}">{!item.TFG_Maximo_Inquilinos__c}</div>
        </td>
        <td data-label="TFG_Tamano">
            <div class="slds-truncate"
title="{!item.TFG_Tamano__c}">{!item.TFG_Tamano__c}</div>
        </td>
        <td data-label="TFG_Precio">
            <div class="slds-truncate"
title="{!item.TFG_Precio__c}">{!item.TFG_Precio__c}</div>
        </td>
    </tr>
</aura:iteration>
</tbody>
</table>
</div>
</div>
</fieldset>
</div>

<!-- Confirmation -->
<div aura:id="alert-dialog" role="alertdialog" tabindex="-1" aria-
labelledby="prompt-heading-id" aria-describedby="prompt-message-wrapper" class="slds-
modal slds-fade-in-open slds-modal--prompt slds-hide">
    <div class="slds-modal__container">
        <div class="slds-modal__header slds-theme--info slds-theme--alert-
texture">
            <h2 class="slds-text-heading--medium" id="prompt-heading-
id">Asignación</h2>
        </div>
        <div class="slds-modal__content slds-p-around--medium">
            <p>Está a punto de asignar la habitación {!v.habitacion.Name} a la
reserva en curso; el coste por noche será de {!v.habitacion.TFG_Precio__c} €. ¿Desea
continuar?</p>
        </div>
        <div class="slds-modal__footer slds-theme--default">
            <button class="slds-button slds-button--brand"
onclick="{!c.confirm}">Asignar</button>
            <button class="slds-button slds-button--neutral"
onclick="{!c.abort}">Cancelar</button>
        </div>
    </div>
</div>
<div aura:id="alert-backdrop" class="slds-backdrop slds-backdrop--open slds-
hide"></div>

<!-- Spinner -->
<div aura:id="spinner_container" class="slds-spinner_container slds-hide">
    <div role="status" class="slds-spinner slds-spinner--medium slds-spinner--
brand">
        <span class="slds-assistive-text">Loading</span>
        <div class="slds-spinner__dot-a"></div>
        <div class="slds-spinner__dot-b"></div>
    </div>
</div>

<!-- Lightning Action -->
<div class="slds-docked-form-footer">
    <button class="slds-button slds-button--neutral" type="button"
onclick="{!c.cancel}">Cancelar</button>
</div>
</aura:component>

```

TFG_Asignacion_ReservasController.js

Este subcomponente permite interactuar con las estructuras y variables definidas en el subcomponente de interfaz; el subcomponente será responsable de realizar pequeños cambios en la interfaz, así como la invocación de métodos más complejos localizados en otros subcomponentes.

- **doInit:** Identifica aquel método ejecutado en la inicialización del componente, responsable de la invocación del método destinado a la búsqueda de recursos.
- **loadPreferences:** Identifica aquel método responsable de la recuperación de los parámetros de búsqueda definidos en la interfaz, y la invocación del método de búsqueda de recursos.
- **pickRoom:** Identifica al método responsable de la selección de cada habitación, almacenando los datos en la correspondiente variable y lanzando el diálogo de confirmación para la asignación de dicho recurso.
- **confirm:** Identifica aquel método responsable de coordinar las invocaciones a los métodos precisos para la asignación de la habitación previamente seleccionada en la correspondiente reserva.
- **abort:** Identifica aquel método responsable de cerrar el diálogo de confirmación en caso de error o desacuerdo.
- **cancel:** Identifica aquel método responsable de cerrar el asistente, recuperando la vista original de la reserva.

A continuación, se facilita el código desarrollado.

```
{
  doInit : function(component, event, helper){
    var reserva = component.get('v.reserva');

    helper.search(component, event, helper, false, false, false, false, false,
false, false, false, false, false, null, null, reserva);
  },
  loadPreferences : function(component, event, helper){
    var TFG_Movilidad_Reducida =
component.find('TFG_Movilidad_Reducida').get('v.checked');
    var TFG_Apto_Ninos = component.find('TFG_Apto_Ninos').get('v.checked');
    var TFG_Chimenea = component.find('TFG_Chimenea').get('v.checked');
    var TFG_Aire_Acondicionado =
component.find('TFG_Aire_Acondicionado').get('v.checked');
    var TFG_Internet = component.find('TFG_Internet').get('v.checked');
    var TFG_Television = component.find('TFG_Television').get('v.checked');
    var TFG_Television_Satelite =
component.find('TFG_Television_Satelite').get('v.checked');
    var TFG_Champu = component.find('TFG_Champu').get('v.checked');
    var TFG_Secador_Pelo = component.find('TFG_Secador_Pelo').get('v.checked');
    var TFG_Plancha = component.find('TFG_Plancha').get('v.checked');

    /* Search Parameters */
    var reserva = component.get('v.reserva');

    var TFG_Precio_Maximo, TFG_Precio_Minimo;
    if (component.find('TFG_Precio_Maximo').get('v.value') != '')
TFG_Precio_Maximo = component.find('TFG_Precio_Maximo').get('v.value');
    else TFG_Precio_Maximo = null;
    if (component.find('TFG_Precio_Minimo').get('v.value') != '')
TFG_Precio_Minimo = component.find('TFG_Precio_Minimo').get('v.value');
```

```

    else TFG_Precio_Minimo = null;

    var automate = component.find('TFG_Precio_Maximo');
    if (TFG_Precio_Maximo != null && TFG_Precio_Minimo != null &&
TFG_Precio_Maximo < TFG_Precio_Minimo) automate.set("v.value", TFG_Precio_Minimo);

    helper.search(component, event, helper, TFG_Movilidad_Reducida,
TFG_Apto_Ninos, TFG_Chimenea, TFG_Aire_Acondicionado, TFG_Internet, TFG_Television,
TFG_Television_Satelite, TFG_Champu, TFG_Secador_Pelo, TFG_Plancha$
    },
    pickRoom : function(component, event, helper) {
        var habitaciones = component.get('v.habitaciones');
        component.set('v.habitacion',
habitaciones[event.currentTarget.dataset.index]);

        helper.dialog(component, event, helper, true);
    },
    confirm : function(component, event, helper) {
        var habitacion = component.get('v.habitacion');
        var reserva = component.get('v.reserva');

        helper.dialog(component, event, helper, false);

        helper.loading(component, event, helper, true);
        helper.assign(component, event, helper, reserva, habitacion.Id);
        setTimeout(function(){ sforce.one.back(true); }, 10000);
    },
    abort : function(component, event, helper) {
        helper.dialog(component, event, helper, false);
    },
    cancel : function(component, event, helper) {
        sforce.one.back(true);
    }
}
})

```

TFG_Asignacion_ReservasHelper.js

Este subcomponente dispone de aquellos métodos responsables de la invocación de los métodos del controlador del componente diseñado para la interacción con los datos del sistema.

- **loading:** Identifica aquel método de apoyo responsable de gestionar la visibilidad del spinner de carga en la interfaz del componente.
- **dialog:** Identifica aquel método de apoyo responsable de gestionar la visibilidad del diálogo de confirmación en la interfaz del componente.
- **search:** Identifica aquel método responsable de la invocación de los recursos del controlador del componente para la búsqueda de habitaciones.
- **assign:** Identifica aquel método responsable de la invocación de los recursos del controlador del componente para la asignación de una habitación a la reserva en curso.

A continuación, se facilita el código desarrollado.

```

({
    loading : function(component, event, helper, show){
        /* Variable Configuration */
        var spinner = component.find('spinner_container');
    }
})

```

```

        if(show) $A.util.removeClass(spinner,'slds-hide');
        else $A.util.addClass(spinner,'slds-hide');
    },
    dialog : function(component, event, helper, show){
        /* Variable Configuration */
        var backdrop = component.find('alert-backdrop');
        var dialog = component.find('alert-dialog');

        if(show){
            $A.util.removeClass(backdrop,'slds-hide');
            $A.util.removeClass(dialog,'slds-hide');
        }else{
            $A.util.addClass(backdrop,'slds-hide');
            $A.util.addClass(dialog,'slds-hide');
        }
    },
    search : function(component, event, helper, TFG_Movilidad_Reducida,
TFG_Apto_Ninos, TFG_Chimenea, TFG_Aire_Acondicionado, TFG_Internet, TFG_Television,
TFG_Television_Satelite, TFG_Champu, TFG_Secador_Pelo, TFG_Plancha,
TFG_Precio_Maximo, TFG_Precio_Minimo, reserva){
        component.set('v.habitaciones',[]);
        helper.loading(component, event, helper, true);

        var action = component.get("c.searchRooms");
        action.setParams({TFG_Movilidad_Reducida: TFG_Movilidad_Reducida,
TFG_Apto_Ninos: TFG_Apto_Ninos, TFG_Chimenea: TFG_Chimenea, TFG_Aire_Acondicionado:
TFG_Aire_Acondicionado, TFG_Internet: TFG_Internet, TFG_Television: TFG_Television,
TFG_Television_Satelite: TFG_Television_Satelite, TFG_Champu: TFG_Champu,
TFG_Secador_Pelo: TFG_Secador_Pelo, TFG_Plancha: TFG_Plancha, TFG_Precio_Maximo:
TFG_Precio_Maximo, TFG_Precio_Minimo: TFG_Precio_Minimo, reservaId: reserva});

        action.setCallback(this, function(response) {
            if (component.isValid() && response.getState() === "SUCCESS") {
                var returnValue = response.getReturnValue();
                if(returnValue != null){
                    component.set('v.habitaciones', returnValue);
                }

                helper.loading(component, event, helper, false);
            }
        });
        $A.enqueueAction(action);
    },
    assign : function(component, event, helper, reserva, habitacion){
        var result = false;
        var action = component.get("c.assignRoom");
        action.setParams({reservaId: reserva, habitacionId: habitacion});

        action.setCallback(this, function(response) {
            if (component.isValid() && response.getState() === "SUCCESS") {
                var returnValue = response.getReturnValue();
                if(returnValue != null){
                    result = returnValue;
                }
            }
        });
        $A.enqueueAction(action);
    }
}
})

```

Controlador del componente Lightning Experience

La versión del entorno en el que se desarrolla el proyecto no ofrece la posibilidad de trabajar con los registros disponibles en los objetos del entorno directamente desde el componente Lightning Experience, siendo necesaria la implementación de una clase de apex responsable de atender toda petición que implique la realización de una operación de consulta, inserción, actualización o eliminación sobre los registros de la plataforma.

A continuación, se facilita el desarrollo del controlador relacionado al asistente.

TFG_Asignacion_Reservas_Controller.apxc

El controlador diseñado para el asistente dispone de aquel conjunto de métodos que invocados por el asistente permiten la interacción del mismo con los datos del sistema; la funcionalidad del asistente queda representada a través de dos métodos.

- **searchRooms:** Identifica aquel método global responsable de la búsqueda de habitaciones en base a un conjunto de características previamente definidas en la reserva o aquellos parámetros indicados por el usuario en el asistente.
- **assignRoom:** Identifica aquel método global responsable de la asignación de aquellos recursos seleccionados en el asistente a las reservas en curso.

A continuación, se facilita el código desarrollado.

```
/*-----  
Author:          Eduardo Ventura  
Company:         Universidad de Alcalá  
Description:     Lightning Experience Controller  
  
History:  
  
<Date>          <Author>          <Change Description>  
05/02/2017      Eduardo Ventura      Initial Version  
22/03/2017      Eduardo Ventura      Remove Rooms  
-----*/  
  
global with sharing class TFG_Asignacion_Reservas_Controller {  
    /* SALESFORCE VARIABLE CONFIGURATION */  
    /* Empty */  
  
    @AuraEnabled  
    public static Habitacion__c[] searchRooms(boolean TFG_Movilidad_Reducida, boolean  
TFG_Apto_Ninos, boolean TFG_Chimenea, boolean TFG_Aire_Acondicionado, boolean  
TFG_Internet, boolean TFG_Television, boolean TFG_Television_Satelite, boolean  
TFG_Champu, boolean TFG_Secador_Pelo, boolean TFG_Plancha, Double TFG_Precio_Maximo,  
Double TFG_Precio_Minimo, String reservaId) {  
        /* Variable Configuration */  
        Habitacion__c[] returnValue;  
  
        /* Retrieve Data */  
        Opportunity opportunity = [SELECT TFG_Fecha_Inicio__c, CloseDate,  
TFG_Total_Inquilinos__c FROM Opportunity WHERE Id =: reservaId][0];  
        List <Opportunity> occupiedRooms = new List <Opportunity>([SELECT  
TFG_Habitacion__c FROM Opportunity WHERE StageName != 'Reserva cancelada' AND  
(TFG_Fecha_Inicio__c >= :opportunity.TFG_Fecha_Inicio__c AND TFG_Fecha_Inicio__c <  
:opportunity.CloseDate) OR (TFG_Fecha_Inicio__c >= :opportunity.TFG_Fecha_Inicio__c
```

```

AND CloseDate < :opportunity.CloseDate) OR (CloseDate >=
:opportunity.TFG_Fecha_Inicio__c AND CloseDate < :opportunity.CloseDate))]];

    /* Query Parameters */
    String query = 'WHERE TFG_Maximo_Inquilinos__c >= ' +
opportunity.TFG_Total_Inquilinos__c + ' ';
    if(TFG_Movilidad_Reducida) query += 'AND TFG_Movilidad_Reducida__c = true ';
    if(TFG_Apto_Ninos) query += 'AND TFG_Apto_Ninos__c = true ';
    if(TFG_Chimenea) query += 'AND TFG_Chimenea__c = true ';
    if(TFG_Aire_Acondicionado) query += 'AND TFG_Aire_Acondicionado__c = true ';
    if(TFG_Internet) query += 'AND TFG_Internet__c = true ';
    if(TFG_Television) query += 'AND TFG_Television__c = true ';
    if(TFG_Television_Satelite) query += 'AND TFG_Television_Satelite__c = true
';

    if(TFG_Champu) query += 'AND TFG_Champu__c = true ';
    if(TFG_Secador_Pelo) query += 'AND TFG_Secador_Pelo__c = true ';
    if(TFG_Plancha) query += 'AND TFG_Plancha__c = true ';
    if(TFG_Precio_Maximo != null) query += 'AND TFG_Precio__c <= ' +
TFG_Precio_Maximo + ' ';
    if(TFG_Precio_Minimo != null) query += 'AND TFG_Precio__c >= ' +
TFG_Precio_Minimo + ' ';

    List <Habitacion__c> rooms = Database.query('SELECT Id, Name,
TFG_Codigo_Habitacion__c, TFG_Maximo_Inquilinos__c, TFG_Tamano__c, TFG_Precio__c FROM
Habitacion__c ' + query + 'ORDER BY TFG_Precio__c ASC');

    /* Remove Rooms */
    Map <Id, Habitacion__c> availableRooms = new Map <Id, Habitacion__c>(rooms);
    for(Opportunity item :
occupiedRooms){if(availableRooms.containsKey(item.TFG_Habitacion__c))
availableRooms.remove(item.TFG_Habitacion__c);}

    if(!availableRooms.isEmpty()) returnValue = (Habitacion__c[])
availableRooms.values();
    else returnValue = null;

    return returnValue;
}

@AuraEnabled
public static Boolean assignRoom(String reservaId, String habitacionId){
    /* Variable Configuration */
    Boolean returnValue = true;

    try{
        Opportunity opportunity = [SELECT Id, TFG_Habitacion__c FROM Opportunity
WHERE Id =: reservaId LIMIT 1][0];

        opportunity.TFG_Habitacion__c = habitacionId;
        update opportunity;
    }catch(Exception exc){returnValue = false;}

    return returnValue;
}
}

```

Interfaces de usuario

El asistente de asignación de recursos se beneficia de una interfaz estructurada en tres secciones, disponiendo las dos primeras secciones de los controles para cada uno de los aspectos diferenciales de una habitación como es la configuración o el precio; la tercera sección consiste en una tabla dinámica, donde se muestran las habitaciones por orden de precio.

La interfaz se encuentra estrechamente relacionada con el controlador del componente, lo que hace posible la actualización de los resultados de forma instantánea.

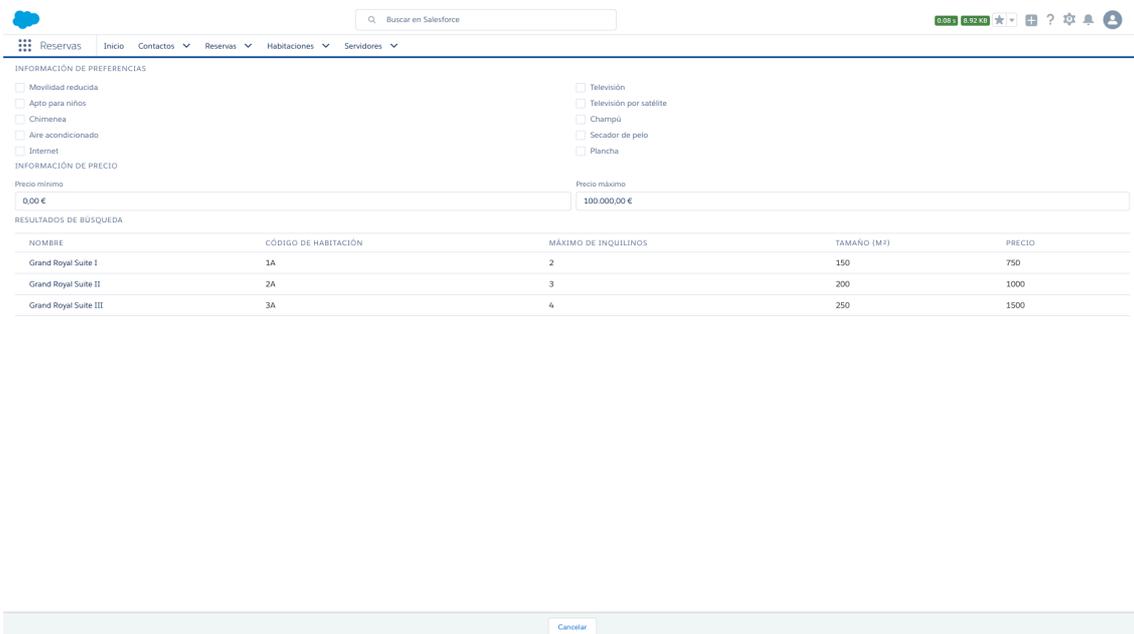


Figura 16. Interfaz de usuario del asistente de asignación de recursos.

Al seleccionarse una de las habitaciones disponibles entre los resultados, el asistente muestra al usuario un cuadro de diálogo donde se expone el resumen de la asignación.

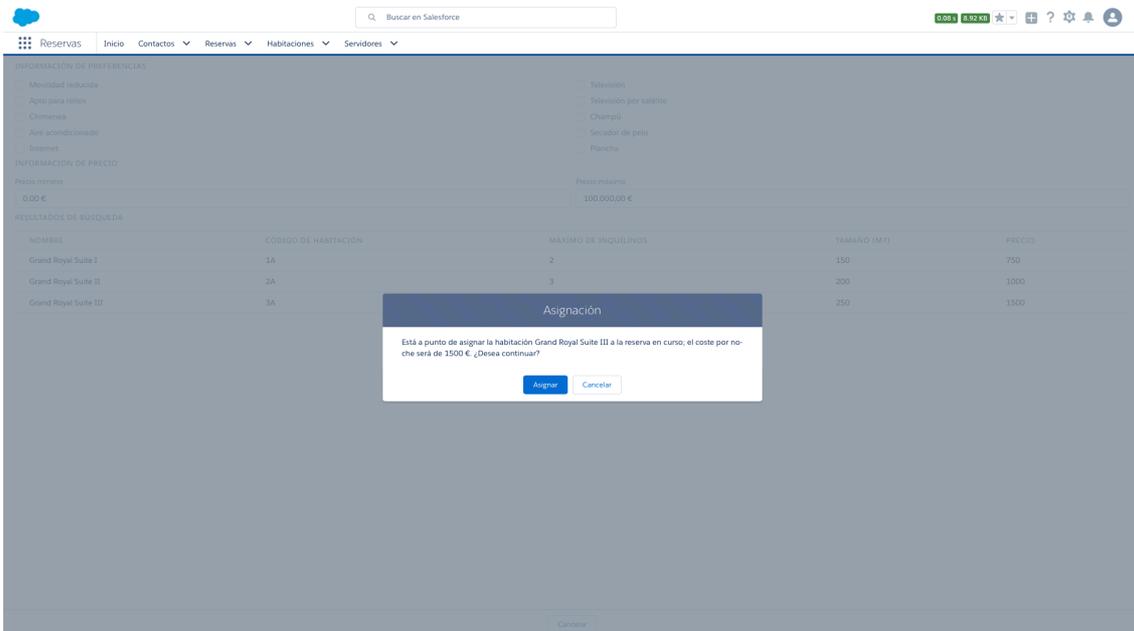


Figura 17. Confirmación para la asignación de recursos a la reserva.

Una vez se ha confirmado la selección, el sistema guardará la nueva configuración de la reserva en curso y dirigirá al usuario a la vista principal de la misma.

Servicios remotos

El sistema de control domótico implementado en la plataforma requiere interactuar con los servicios implementados en los servidores domóticos, siendo necesaria la realización de un conjunto de ajustes y desarrollos.

La invocación a los servicios se realiza a través de la dirección de red pública asignada a la organización en la que se encuentra el servidor; siendo necesario incorporar dicha dirección de red al listado de direcciones de confianza de la plataforma.

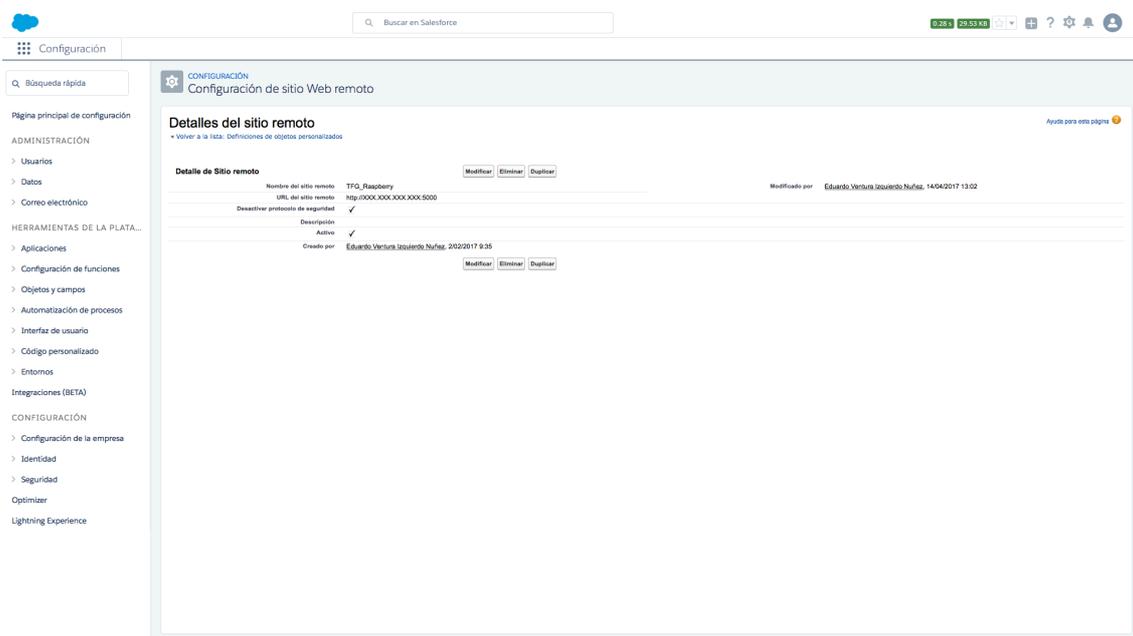


Figura 18. Ajuste ficticio de la dirección de red de la organización.

Es preciso implementar una configuración por cada una de las direcciones de red a las que se desee acceder desde la plataforma.

Implementación de tareas programadas

Algunos servicios precian ser invocados automáticamente por el sistema sin necesidad de intervención del usuario, para ello es preciso definir un lanzador el cual ejecute aquel método responsable de desencadenar el servicio cada cierto intervalo de tiempo.

En primer lugar, es necesario implementar una clase objetivo para el lanzador, siendo la responsable de invocar aquellos métodos necesarios para desencadenar el servicio.

/*-----
Author: Eduardo Ventura
Company: Universidad de Alcalá

Description: Apex Schedule

History:

```
<Date>                <Author>                <Change Description>
14/04/2017            Eduardo Ventura            Initial Version
-----*/
```

```
global class TFG_Scheduler implements Schedulable {
    /* SALESFORCE VARIABLE CONFIGURATION */
    /* Empty */

    global void execute(SchedulableContext ctx) {
        /* Variable Configuration */
        List<Servidor__c> servers = [SELECT Id FROM Servidor__c WHERE
TFG_Habilitado__c = true AND TFG_Funcionamiento__c = 'Correcto'];

        for(Servidor__c item : servers){
            TFG_ServidorMethods.infoFromRaspberry(item.Id);
        }
    }
}
```

Finalmente, es necesario ajustar el instante de ejecución de la clase objetivo evitando la sobrecarga de los sistemas involucrados; precisando de la confección de un comando de configuración personalizado donde se ajusta la ejecución a cada hora.

```
System.schedule('Web Service', '0 0 * * * ?', new TFG_Scheduler());
```

El comando ha de configurarse a través de una ejecución de apex; siendo preciso el uso de la herramienta Workbench.

Implementación de servicios

La invocación de los servicios ha de realizarse respetando la interfaz de comunicación externa definida por el sistema domótico, garantizando la correcta comunicación entre los distintos sistemas involucrados; para ello, se centraliza en una misma clase todas las estructuras de datos necesarias para la explotación de los servicios.

```
/*-----*/
Author:                Eduardo Ventura
Company:               Universidad de Alcalá
Description:           Apex Structures

History:

<Date>                <Author>                <Change Description>
02/02/2017            Eduardo Ventura            Initial Version
-----*/
```

```
global class TFG_Raspberry {

    /* Automation: Post Method */
    global class ResourceRequestType {
        public boolean energy;
        public boolean water;
        public boolean block;
    }

    /* Automation: Get Method */
```

```

global class EnvironmentResponseType {
    public double temperature;
    public double humidity;
}
}

```

A continuación, se indican aquellos desarrollos relacionados con la explotación de los servicios del sistema domótico.

TFG_ServidorMethods.apxc

Este componente permite la conversión de los datos de la plataforma en las estructuras de datos requeridas por los servicios domóticos, así como el procesamiento de aquellas estructuras de datos recuperadas.

Es preciso mencionar la necesidad de ejecutar los métodos asincrónicamente, empleando para ello la sentencia *@future*.

- **changesToRaspberry:** Identifica aquel método responsable de enviar los ajustes definidos por el usuario en el panel del servidor al sistema domótico.
- **infoFromRaspberry:** Identifica aquel método responsable de la recuperación de los valores de información ambiental del sistema domótico.

A continuación, se facilita el código desarrollado.

```

/*-----*/
Author:          Eduardo Ventura
Company:         Universidad de Alcalá
Description:     Apex Methods

History:

<Date>          <Author>          <Change Description>
02/02/2017      Eduardo Ventura      Initial Version
-----*/

global class TFG_ServidorMethods {
    /* SALESFORCE VARIABLE CONFIGURATION */
    /* Empty */

    @future (callout = true)
    public static void changesToRaspberry (Id identification) {
        /* Variable Configuration */
        boolean result;
        List<Servidor__c> servers = [SELECT Id, TFG_Usuario__c, TFG_Contrasena__c,
TFG_Energia__c, TFG_Agua__c, TFG_Puerta__c, TFG_Nodo_Direccionamiento__c,
TFG_Funcionamiento__c FROM Servidor__c WHERE Id =: identification];

        /* Transformation */
        if (servers.size() == 1){
            Servidor__c server = servers.get(0);

            TFG_Raspberry.ResourceRequestType resourceRequest = new
TFG_Raspberry.ResourceRequestType ();
            resourceRequest.energy = server.TFG_Energia__c;
            resourceRequest.water = server.TFG_Agua__c;
            resourceRequest.block = server.TFG_Puerta__c;

```


History:

| <Date> | <Author> | <Change Description> |
|------------|-----------------|----------------------|
| 02/02/2017 | Eduardo Ventura | Initial Version |

-----*/

```
public class TFG_Salesforce_To_Raspberry {
    /* SALESFORCE VARIABLE CONFIGURATION */
    private final static Integer port = 5000;
    private final static Integer timeout = 20000;

    public static boolean postValues (String host, String username, String password,
TFG_Raspberry.ResourceRequestType resourceRequest){
    /* Variable Configuration */
    boolean result = false;
    Http http = new Http();
    HttpRequest request = new HttpRequest();
    HttpResponse response = new HttpResponse();

        /* Authorization */
        String auth = 'Basic ' +
EncodingUtil.base64Encode(Blob.valueOf(username + ':' + password));

    /* Request Management */
    request.setEndpoint('http://'+host+':'+port+'/api/v1.0/automation');
    request.setMethod('POST');
    request.setHeader('Content-Type', 'application/json');
    request.setHeader('Authorization', auth);
    request.setTimeout(timeout);

    try{
        /* JSON */
        request.setBody(JSON.serialize(resourceRequest));

        /* Invoke */
        if(System.Test.isRunningTest()){System.debug('Mock');}else response =
http.send(request);

        /* Response Management */
        if(response.getStatusCode() == 200){
            /* Successfully Created */
            result = true;
        }
    }catch(Exception exc){}
    return result;
}

public static TFG_Raspberry.EnvironmentResponseType getValues (String host){
    /* Variable Configuration */
    TFG_Raspberry.EnvironmentResponseType environmentResponse;

    Http http = new Http();
    HttpRequest request = new HttpRequest();
    HttpResponse response = new HttpResponse();

    /* Request Management */
    request.setEndpoint('http://'+host+':'+port+'/api/v1.0/automation');
    request.setMethod('GET');
    request.setTimeout(timeout);

    try{
        /* Invoke */
        if(System.Test.isRunningTest()){System.debug('Mock');}else response =
http.send(request);

        /* Response Management */
```

```
        if(response.getStatusCode() == 200){
            /* Successfully Created */
            environmentResponse = (TFG_Raspberry.EnvironmentResponseType)
JSON.deserialize(response.getBody(), TFG_Raspberry.EnvironmentResponseType.class);
        }
    } catch (Exception exc) {}
    return environmentResponse;
}
}
```

Implementación de aplicaciones personalizadas

Las aplicaciones personalizadas se relacionan estrechamente con los departamentos de la organización, permitiendo al administrador del sistema generar distintas aplicaciones adaptadas a las necesidades particulares de cada departamento involucrado en la plataforma.

Las aplicaciones personalizadas hacen posible controlar la visibilidad y organización de los distintos objetos disponibles en el sistema, de forma que cada departamento pueda acceder a la información de interés de una forma más óptima.

La ejecución del proyecto requiere la configuración de una aplicación personalizada adaptada a las necesidades del personal de recepción; donde se facilite el acceso a los objetos relativos a contactos, reservas, habitaciones y servidores.

A continuación, se adjuntan los distintos parámetros definidos en la configuración de la aplicación personalizada.

The screenshot shows a web interface for configuring a custom application. At the top, there is a dark blue header with the text 'Modificar aplicación' and a close button. Below the header is a navigation bar with five tabs: 'DETALLES DE APLICACIÓN Y MARCA' (selected), 'OPCIONES DE APLICACIÓN', 'BARRA DE UTILIDADES', 'SELECCIONAR ELEMENTOS', and 'ASIGNAR A PERFILES DE USUARIO'. The main content area is titled 'Detalles de aplicación y Marca' and includes a sub-header: 'Proporcione a su aplicación Lightning un nombre y una descripción. Cargue una imagen y seleccione el color de resaltado para su barra de navegación.' The interface is divided into two columns. The left column, 'Detalles de la aplicación', contains three input fields: 'Nombre de aplicación' (with 'Reservas' entered), 'Nombre del desarrollador' (with 'TRG_Reservas' entered), and 'Descripción' (with 'Gestión de las reservas del complejo hotelero.' entered). The right column, 'Marca de aplicación', contains an 'Imagen' section with a 'Cargar' button and a 'Valor de color hexadecimal principal' section with a dropdown menu and the value '#004495'. Below these is a 'Vista previa del Iniciador de aplicación' showing a blue square with 'Re' and a grey box with 'Reservas' and the description. At the bottom right of the interface is a blue 'Listo' button.

Figura 19. Definición de los detalles de la aplicación personalizada.

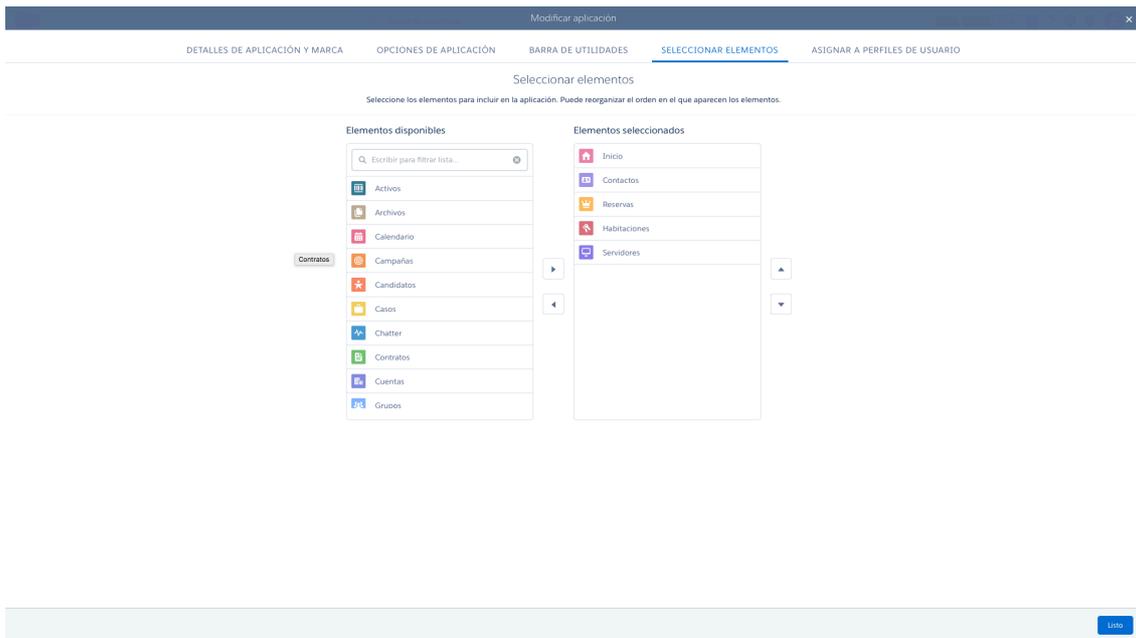


Figura 20. Ajuste de objetos visibles en la aplicación personalizada.

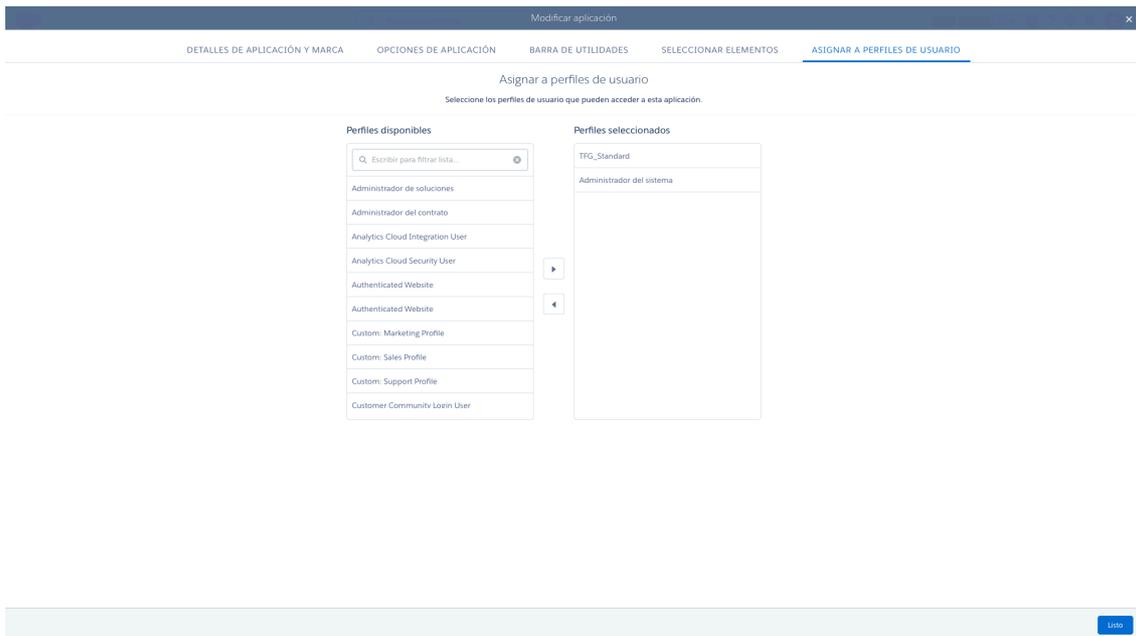


Figura 21. Ajuste de visibilidad por perfiles en la aplicación personalizada.

Al finalizar, todos los perfiles de usuario indicados en la configuración de la aplicación personalizada dispondrán de total libertad para acceder a la misma desde el selector de aplicaciones.

Es posible personalizar la aplicación iniciada por defecto en cada perfil de usuario, para ello, es preciso modificar la configuración de aplicaciones en cada uno de los perfiles deseado.

Bibliografía

Adafruit. (2017). *Flora*. Obtenido de <https://www.adafruit.com/product/659>

Arduino. (2017). *Arduino One & Genuino One*. Obtenido de <https://www.arduino.cc/en/main/arduinoBoardUno>

Beagleboard. (2017). *BeagleBone Black*. Obtenido de <https://beagleboard.org/Black>

Flask Framework. (2017). *Quickstart*. Obtenido de <http://flask.pocoo.org/docs/0.12/quickstart>

Flask Framework. (2017). *API*. Obtenido de <http://flask.pocoo.org/docs/0.12/api>

Francesc. (15 de agosto de 2015). *Sensor DHT11 de temperatura y humedad*. [Foro]. Obtenido de <http://fpaez.com/sensor-dht11-de-temperatura-y-humedad>

Jaguarboard. (2017). *Jaguar One*. Obtenido de http://www.jaguarboard.org/index.php/com_virtuemart_menu_configuration/products/buy/jaguarboard/207/jaguarboard-detail.html

Microsoft. (2017). *Precios de Dynamics 365*. Obtenido de <https://www.microsoft.com/es-es/dynamics365/pricing>

Raspberry Pi Foundation. (2017). *Configuring Raspbian with Raspberry Pi configuration file*. Obtenido de <https://www.raspberrypi.org/documentation/configuration/config-txt/README.md>

Raspberry Pi Foundation. (2017). *Configuring Raspbian with Raspberry Pi configuration tool*. Obtenido de <https://www.raspberrypi.org/documentation/configuration/raspi-config.md>

Raspberry Pi Foundation. (2017). *Installing operating system images*. Obtenido de <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

Raspberry Pi Foundation. (2017). *Installing software using Python's package manager*. Obtenido de <https://www.raspberrypi.org/documentation/linux/software/apt.md>

Raspberry Pi Foundation. (2017). *Introduction to GPIO and physical computing on the Raspberry pi*. Obtenido de <https://www.raspberrypi.org/documentation/usage/gpio/README.md>

Raspberry Pi Foundation. (2017). *Raspberry Pi 3 Model B*. Obtenido de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b>

Raspberry Pi Foundation. (2017). *Schematics for the Raspberry Pi 1 Model A and B board versions, revision 2.1*. Obtenido de <https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/Raspberry-Pi-Rev-2.1-Model-AB-Schematics.pdf>

Raspberry Pi Foundation. (2017). *Updating and upgrading Raspbian*. Obtenido de <https://www.raspberrypi.org/documentation/raspbian/updating.md>

Salesforce. (2017). *About Developer Edition*. Obtenido de https://developer.salesforce.com/page/Developer_Edition

Salesforce. (2017). *Apex Developer Guide*. Obtenido de https://developer.salesforce.com/docs/atlas.en-us.206.0.apexcode.meta/apexcode/apex_intro.htm

Salesforce. (2017). *Contacts*. Obtenido de https://help.salesforce.com/articleView?id=contacts_overview.htm&type=0

Salesforce. (2017). *Custom Fields*. Obtenido de https://help.salesforce.com/articleView?id=customize_customfields.htm&type=0

Salesforce. (2017). *Lightning Components Developer Guide*. Obtenido de https://developer.salesforce.com/docs/atlas.en-us.lightning.meta/lightning/intro_framework.htm

Salesforce. (2017). *Lightning Design System*. Obtenido de <https://www.lightningdesignsystem.com>

Salesforce. (2017). *Lookup Filters*. Obtenido de https://help.salesforce.com/articleView?id=fields_lookup_filters.htm&type=0

Salesforce. (2017). *Opportunities*. Obtenido de <https://help.salesforce.com/articleView?id=opportunities.htm&type=0>

Salesforce. (2017). *Page Layouts*. Obtenido de https://help.salesforce.com/articleView?id=customize_layout.htm&type=0

Salesforce. (2017). *Record Types*. Obtenido de https://help.salesforce.com/articleView?id=creating_record_types.htm&type=0

Salesforce. (2017). *Store information that's unique to your organization*. Obtenido de https://help.salesforce.com/articleView?id=dev_object_def.htm&type=0

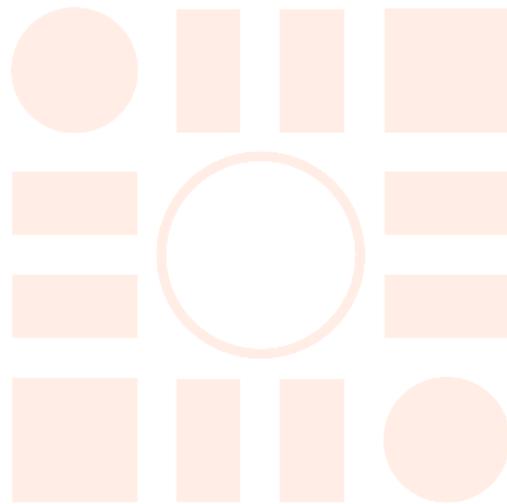
Salesforce. (2017). *Validation Rules*. Obtenido de https://help.salesforce.com/articleView?id=fields_about_field_validation.htm&type=0

Zoho. (2017). *Precios de Zoho*. Obtenido de <https://www.zoho.eu/crm/zohocrm-pricing.html>



Escuela Politécnica Superior

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá