

# Monitor de Control Integral

Estefania García Corominas

**Resumen**— Control Integral, es un programa informático especializado en gestión de ferreterías, bricolaje, suministros industriales y centros de construcción. Este programa está formado por dos ejecutables: el primero de ellos es el de 'Gestión' y el segundo es el llamado 'Monitor'. El módulo de gestión se compone de diferentes características para satisfacer las necesidades de los clientes, actualización automática de precios de los artículos, terminal punto de venta (TPV) este permite la creación e impresión del ticket de venta mediante las referencias de productos, realizan diversas operaciones durante todo el proceso de venta, así como cambios en el inventario. También generan diversos reportes que ayudan en la gestión del negocio. El segundo módulo es el 'Monitor' que gestiona el sistema de licencias: usuarios concurrentes, fecha de expiración y módulos accesorios contratados. Valida el acceso y el uso por parte de los usuarios de acuerdo con la licencia contratada. La seguridad del sistema se basa en dispositivos USB Sentinel HASP y en el intercambio de mensajes firmados y cifrados. También permite la modificación/activación remota y desatendida de la licencia de uso. En este segundo módulo es en el que nos centraremos en este Proyecto.

**Palabras clave**—Monitor, Control Integral, Comunicación, protocolo, Cliente – Servidor, Azure, Licencias, XML, BBDD, Redes.

**Abstract**— Control Integral is a software specialized on the management of hardware stores, DIY, industrial supplies and construction centers. It consists of two executables. The first one is the 'Management' module and the second one is the 'Monitor'. The management module provides different features to fulfill customer's needs, such as: automatic update of items prices, managing the POS (Point of sales terminal), enabling the creation and printing of sales tickets by product references, and inventory update, among others. This software also generates different reports that help managing the business. Finally, the 'Monitor' module manages the licensing system: concurrent users, expiration date and accessory modules hired. It validates access and users according to the contracted license. System security is based on Sentinel HASP USB devices and signed and encrypted message exchange. It also allows modification / unattended remote activation and license use. This second module is where we will focus on this project.

**Index Terms**—Monitor, Control Integral, Communication, protocol, Client – Server, Azure, Licences, XML, BBDD, networking.



## 1 INTRODUCCIÓN

La idea original de este proyecto surge de AGC Projectes Informàtics, una empresa cuyo objetivos esenciales son los de asesorar, suministrar y dar soporte a los clientes con las mejores soluciones de *software*, respondiendo con éxito a las más altas exigencias de los clientes del mercado actual y futuro, consiguiendo de éstos la mayor satisfacción, tanto de los productos como de los servicios ofrecidos.

En AGC Projectes Informàtics surge de realizar una nueva versión del módulo de Monitor del programa de Control Integral que en su día fue implementado por Miquel Biosca en su respectivo proyecto final de carrera y que se basa en una comunicación a través de *sockets* y *hardware*.

---

- E-mail de contacto: [steffgarcia23@gmail.com](mailto:steffgarcia23@gmail.com)

- Mención realizada: Ingeniería del Software.

- Trabajo tutorizado por: Katerine Díaz Chito (CVC)

- Curso 2015/16

La empresa ha visto adecuado renovarlo, haciendo desaparecer toda la parte hardware y estableciendo otro tipo de comunicación entre un cliente y un servidor

El módulo forma parte de Control Integral [1], es un programa informático especializado en gestión de ferreterías, bricolaje, suministros industriales y almacenes de material de construcción. Está formado por dos ejecutables llamados 'Gestión' y 'Monitor'. La función de este segundo es la de gestionar el sistema de licencias (usuarios concurrentes, fecha de expiración y módulos accesorios contratados), validar el acceso y el uso de los usuarios de acuerdo con la licencia contratada y permitir la activación remota de forma desatendida de la licencia del cliente (cuando uno de los distribuidos realice la instalación inicial de un cliente), permitiendo que pueda realizar la activación autónoma sin la ayuda de ACG Projectes Informàtics.

Cabe remarcar que la seguridad del sistema se basa en dispositivos USB Sentinel HASP [2] y en el intercambio de mensajes firmados y cifrados.

Actualmente, la estructura de comunicación entre el módulo "Monitor" y el de "Gestión" se realiza mediante un HASP (Figura 1). Dicho Hasp se encuentra en el ordenador principal (Servidor) y ofrece una solución robusta para la protección de software empresarial diseñado para operar en una infraestructura de red. Esta protección es un sistema de seguridad basado en hardware el cual brinda una protección al software contra la piratería y el uso ilegal, permitiendo el acceso y ejecución únicamente cuando el HASP está conectado al ordenador. Los HASP's contienen un motor de cifrado de alta seguridad en el cual todo el proceso se realiza dentro del hardware, incluye un identificador único de 32 bits, el cifrado AES de 128 bits y soporte para 64.000 claves de cifrado. Durante la ejecución, el software protegido envía secuencias cifradas a la llave que las descifra produciendo una respuesta que no se puede emular. Si la respuesta de la llave es correcta la aplicación sigue funcionando. Si la llave no está conectada o la respuesta es incorrecta, la aplicación no se ejecuta.

Al utilizar el Sentinel HASP HL Net [2], el desarrollador puede imponer restricciones en el número de usuarios simultáneos desde un programa de seguro, además de ofrecer la opción de soporte de licencias concurrentes.

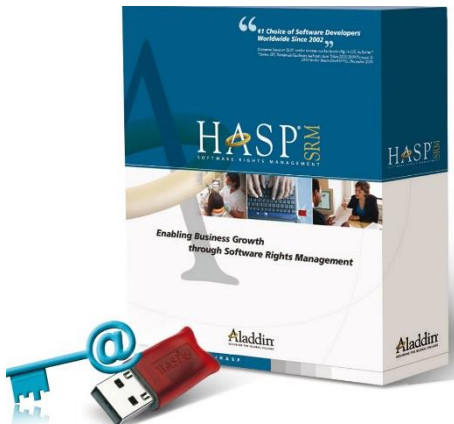


Figura 1: Imagen del dispositivo USB llamado Hasp.

Una desventaja importante es la parte en la cual es necesaria una llave por cada copia del programa, con esto el precio y costo de fabricación incrementan su valor y pueden surgir problemas en la venta del programa.

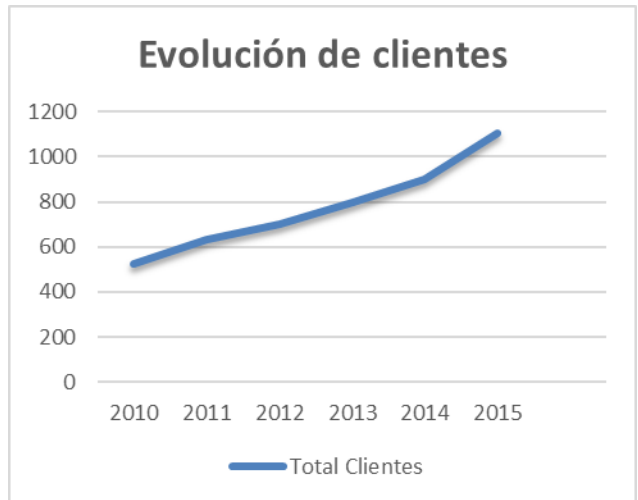


Figura 2. Evolución de los clientes dados de alta en la empresa ACG Projectes Informàtics durante el periodo de años 2010 - 2015 (Fuente: ACG Projectes Informàtics)

La Figura 2 presenta un gráfico donde se observa como el alta de los clientes en la empresa ha evolucionado considerablemente, lo cual supone un incremento de los gastos para la empresa.

También la empresa tiene un hándicap y es que en menos de 2 años estos dispositivos dejan de fabricarse, con lo cual no podrán ser adquiridos para su utilización.

Este proyecto propone definir, diseñar y desarrollar un nuevo 'Monitor', basado en el concepto cliente-servidor, y cuyo objetivo final y esencial es el de definir un entorno común para las arquitecturas de todos los servicios. Es decir, desarrollar un nuevo módulo 'Monitor' eliminando toda la parte de hardware (USB) y haciendo del 'Monitor' un servicio basado totalmente en software y que realice las mismas funciones anteriormente citadas.

Los objetivos generales del proyecto son:

- Determinar la estructura entre Cliente - Servidor
- Desarrollar de la comunicación entre Cliente - Servidor
- Determinar la estructura entre Cliente y Control Integral
- Seleccionar la plataforma a contratar en la infraestructura(backend-frontend) [7]

Este documento se estructura en secciones.

La Sección 2 presenta el contexto del proyecto. La Sección 3 presenta en detalle los objetivos del proyecto. La Sección 4 presenta expone los casos de uso prácticos. La Sección 5, detalla los cambios y modificación de los objetivos. La Sección 6 presenta la metodología utilizada, el diseño, y las funcionalidades implementadas. La Sección 7, expone las conclusiones finales del proyecto respecto a los objetivos planteados. Finalmente, La Sección 8 presenta una visión de las posibles líneas de futuro.

## 2. ESTADO DEL ARTE

Existen diversos puntos de vista sobre la manera en que debería efectuarse el procesamiento de datos. En este contexto, es necesario establecer una infraestructura de Procesamiento de información, que cuente con los elementos requeridos para proveer información adecuada, exacta y oportuna en la toma de decisiones y para proporcionar un mejor servicio a los clientes. El modelo Cliente/Servidor reúne las características necesarias para proveer esta infraestructura, independientemente del tamaño y complejidad de las operaciones de las organizaciones públicas o privadas y, consecuentemente desempeña un papel importante en este proceso de evolución. Cliente/Servidor ha evolucionado de época.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes Puntos

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
  - Generar requerimientos de bases de datos.
  - Recibir resultados del servidor.
  - Formatear resultados.

La tecnología denominada Cliente -Servidor es utilizada por todas las aplicaciones de Internet/Intranet. Un cliente funciona en su ordenador local, se comunica con el servidor remoto, y pide a éste información. El servidor envía la información solicitada. Un único servidor típicamente sirve a una multitud de clientes, ahorrando a cada uno de ellos el problema de tener la información instalada y almacenada localmente.

Algunos servidores esperan las solicitudes en puertos bien conocidos de modo que sus clientes saben a qué IP deben dirigir sus peticiones. El cliente emplea un puerto arbitrario para comunicarse.

Características de la arquitectura Cliente/Servidor

- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.

- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco e input-output devices.

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.

- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.

- La relación establecida puede ser de muchos a uno,

en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.

- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.

- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.

- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.

- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

En este proyecto, se realizará una arquitectura Cliente/servidor. Un entorno común para las arquitecturas de todos los servicios. Es decir, desarrollar un nuevo módulo 'Monitor' eliminando toda la parte de hardware (USB) y haciendo del 'Monitor' un servicio basado totalmente en software y que realice las mismas funciones que estaba realizando hasta el momento.

## 3. OBJETIVOS

El objetivo general de este proyecto de I+D consiste en definir, diseñar y desarrollar de nuevo el 'Monitor', que está basado en el concepto cliente-servidor y cuyo objetivo final y esencial será definir un entorno común para las arquitecturas de todos los servicios. La Tabla 1 presenta los objetivos específicos del proyecto.

|   | Critico | Prioritario | Secundario |
|---|---------|-------------|------------|
| Determinar comunicación entre Control Integral y cliente                          | X       |             |            |
| Determinar estructura a realizar entre Control Integral y cliente                 | X       |             |            |
| Determinar comunicación entre Monitor y programa(Cliente)                         | X       |             |            |
| Realizar comunicación entre Monitor y programa                                    | X       |             |            |
| Seleccionar tipo de servidor a contratar en la infraestructura (backend-frontend) |         | X           |            |

|  |   |   |   |
|--|---|---|---|
| Determinar tipo de cifrado en la comunicación  |   | X |   |
| Cifrar comunicación  |   |   |   |
| Especificar, para cada funcionalidad del Monitor si es accesible para todos los sistemas operativos disponibles entre los clientes |   | X |   |
| Incluir más funcionalidades como: volcado de datos en el mismo servicio de monitor   |   |   | X |
| Montar servidor de licencias   | X |   |   |

Tabla 1: Objetivos específicos del proyecto

a. *Determinar la estructura entre Servidor y el cliente.*

La base de este proyecto es la comunicación entre un cliente y un servidor en entorno local. Esta comunicación ha de establecer un entorno seguro, procesar la información de modo distribuido, es decir que los usuarios finales pueden estar dispersos en un área geográfica más o menos extensa (un edificio o una localidad) y acceder a un conjunto común de recursos compartidos y el acceso debe ser transparente, es decir el cliente puede desconocer la ubicación física del recurso que pretende utilizar.

Las características que se necesitan para cumplir dicho objetivo son:

- Transparencia.
- Independencia.
- Protocolos asimétricos.
- Recursos compartidos.
- Servicio.
- Encapsulamiento.
- Integridad.
- Acoplamiento débil.
- Escalabilidad.

b. *Desarrollo de la comunicación entre Cliente – Servidor*

El siguiente paso es crear una comunicación entre el cliente y el servidor, teniendo en cuenta las restricciones de los sistemas operativos de los clientes, para ello se tienen que crear las peticiones que va a responder el servidor y posteriormente crear las peticiones de los clientes. Para que un cliente pueda realizar dichas peticiones también se ha de generar un Cliente REST, que realizará las peticiones vía HTTP.

La Figura 3 muestra de forma gráfica los pasos para establecer la comunicación.

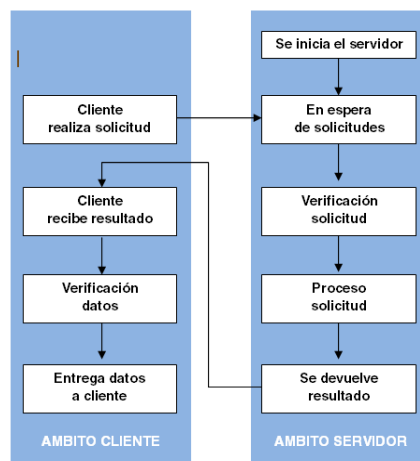


Figura 3: Funcionamiento gráfico de los 2 ámbitos Cliente/Servidor

- c. *Determinar la estructura entre Cliente y Control Integral.* El objetivo de esta parte es que la estructura tiene que ofrecer disponibilidad los servicios que ofrece, durante unas horas determinadas al máximo rendimiento.

Para ello, se tiene que crear un servidor de licencias, que será el encargado de gestionar la comunicación con los clientes.

- d. *Seleccionar plataforma a contratar en la infraestructura (backend-frontend).* El objetivo de dicha plataforma es que tiene que ser flexible, rápida y de fácil administración.

#### 4. CASOS DE USO

Para determinar el funcionamiento de cada elemento se han generado los casos de uso.

- Funciones que puede realizar en el Cliente:
  - Servidor:
    - Actualizar versión del programa
      - Actualización manual
      - Actualización online
    - Gestión licencia de acceso al programa
    - Gestión módulos
    - Gestión volcado de datos
    - Gestión activaciones de licencias
      - Activación desasistida
      - Activación online
  - Cliente
    - Petición acceso al programa
    - Gestión de Control Integral
- Funciones que puede realizar en la Empresa:
  - Actualizar versión del programa
    - Actualización manual

- Actualización online
- Comunicación servidor del cliente
- Activación módulos
- Acceso actualizar versión del programa
- Activación de licencias
  - Activación desasistida
  - Activación online

En el Apéndice A1 y A2 se muestran las imágenes de los casos de uso del cliente y del servidor [3].

## 5. CAMBIOS/MODIFICACION DE OBJETIVOS

### 5.1 Planificación

En la planificación, se decidió emplear una metodología de desarrollo ágil e incremental, centrada en las funcionalidades a implementar en cinco etapas (Sprint). En cada una, se han implementado diferentes utilidades y generado versiones entregables de la aplicación, para poder demostrar de manera visual los avances conseguidos y discutir con el tutor mejoras a realizar.

Para organizar las tareas a realizar se utilizó Trello [8], una aplicación totalmente gratuita en la que podemos organizar las tareas en diferentes columnas y marcar los periodos temporales en las que se van a realizar (Figura 4).



Figura 4: Imagen interfaz de la aplicación de Trello

El proyecto se ha llevado a cabo para la empresa ACG Projectes Informàtics [1]. La duración total del proyecto ha sido de 6 meses y constaba de las siguientes etapas:

- Etapa 1:
  - Elicitación de los requisitos funcionales y no funcionales
  - Generación de casos de uso del proyecto
  - Investigación de la arquitectura cliente y servidor
  - Especificación de los objetivos
  - Investigación de cifrado de la comunicación

- Etapa 2
  - Generación comunicación cliente y servidor
    - Peticiones y respuestas en c# [4]
    - Peticiones y respuestas en c++[5]
    - Cliente Rest [6] en cpp
- Etapa 3
  - Desarrollo comunicación abierta (sin cifrar) entre cliente y servidor
  - Desarrollo comunicación cifrada entre cliente y servidor
  - Definir una infraestructura para Control Integral y cliente. (Servidor de licencias)
- Etapa 4
  - Validación de las tareas realizadas (pruebas de test)
- Etapa 5: Entrega final y presentación del proyecto

La planificación se siguió correctamente las 2 primeras etapas, pero a partir de la 3ª se modificaron tareas a realizar. Se llegó a un acuerdo entre la empresa y el estudiante que la parte de la comunicación cifrada se realizaría al final del proyecto.

Lo esencial de este proyecto es que las comunicaciones, funcionen con total perfección, ya que es básico para el correcto funcionamiento de la aplicación tanto de forma local (Parte cliente) y/o comunicación con la empresa. De esta forma si surge algún problema en la comunicación se pueda solventar de una forma más sencilla.

Es por ello que la planificación ha quedado de la siguiente forma:

- Etapa 3
  - Desarrollo comunicación abierta (sin cifrar) entre cliente y servidor
  - Definir una infraestructura para Control Integral y cliente. (Servidor de licencias)
- Etapa 4
  - Validación de las tareas realizadas (pruebas de test)
- Etapa 5: Entrega final y presentación del proyecto

## 6. METODOLOGIA

Esta sección explica las fases de desarrollo realizadas durante el transcurso de este proyecto.

### 6.1 Análisis de requerimientos

Antes de empezar la implementación del proyecto se llevó a cabo un análisis previo a los requerimientos necesarios para cumplir los objetivos del proyecto y así poder ofrecer mejor servicio al cliente.

A continuación se define una lista con los requisitos funcionales del proyecto:

- Evitar acceder a funcionalidades no contratadas
- Mantener seguro el uso concurrente de usuarios
- Gestionar fechas de caducidad de las licencias
- Gestionar la activación /desactivación de las funcionalidades
- Licencias temporales desasistidas
- Evitar la suplantación
- Comunicaciones seguras

Seguidamente, la lista de los requisitos no funcionales:

- Monitorización y trazabilidad
- Disponibilidad
- Disponibilidad ante emergencias

## 6.2 Comunicación Cliente/Servidor (parte del cliente)

En esta parte se explica las diferentes partes de la estructura de cliente y servidor, en la Figura 5 se puede observar el funcionamiento interno de la estructura desarrollada.

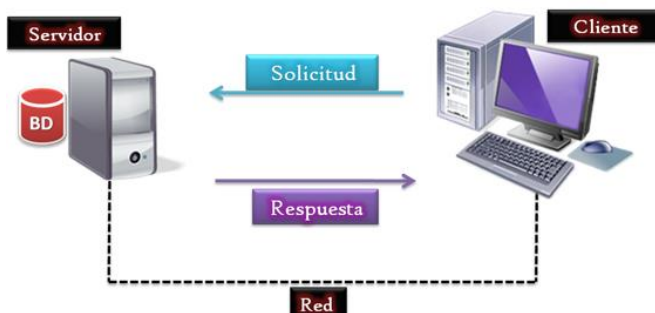


Figura 5: Estructura Cliente-Servidor desarrollada

### 6.2.1 Parte del servidor

Una vez se tenía claro el funcionamiento entre un cliente y el servidor, el objetivo a cumplir era desarrollar las respuestas de la parte del servidor en el lenguaje CSharp(C#) [4].

Las respuestas que proporciona el servidor son:

- LicenseResponse
- LoginResponse
- SessionListResponse
- Constants
- Session
- ModuleType
- Module
- Ping

Todas estas respuestas las necesita el cliente para poder trabajar de forma habitual.

- Número de licencia que tiene

- Si el cliente quiere logarse
- Lista de sesiones activas que tiene el servidor
- Constantes del trabajo
- Que ID de sesión tiene el cliente
- Que tipos de módulos tiene activados la licencia
- Módulos activos en el cliente
- Para comprobar si el cliente está activo o por el contrario hay que quitarlo de las sesiones activas del servidor.

A continuación, se muestra una captura del código para responder a la solicitud de un cliente del login al programa de gestión (Figura 6).

```
namespace CI.DistributedService.WebService.Monitor.Contracts
{
    public class MonitorLoginResponse : MonitorResponse
    {
        public string SessionId { get; set; }
    }
}
```

Figura 6: Código C# de respuesta de login al programa de gestión

Para que el servidor pueda desarrollar todas las posibles respuestas se ha generado un servicio de Windows, que permite crear aplicaciones ejecutables de larga duración, que se ejecutan en sus propias sesiones de Windows. Este servicio tendrá que estar iniciado automáticamente cuando el equipo arranque, y no mostrará ninguna interfaz de usuario.

Estas características hacen que el servicio resulte perfecto para ejecutarse en un servidor ya que se necesita una funcionalidad de ejecución larga que no interfiera con los demás usuarios que trabajen en el mismo equipo.

La Figura 7 muestra el servicio generado, ControlIntegralMonitor.

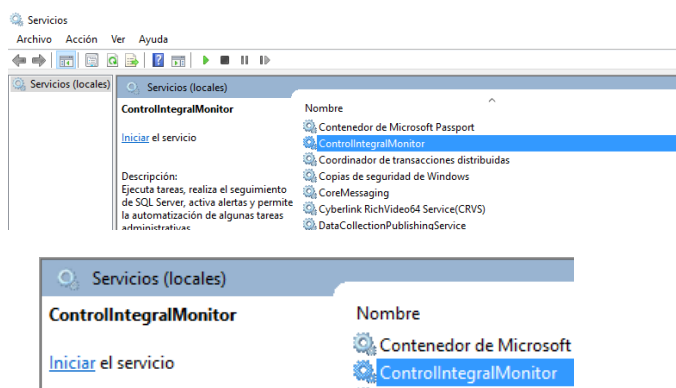


Figura 7: Captura de pantalla del servicio que genera el código para poder responder a las peticiones del cliente

Para comprobar las respuestas que ofrece el servidor, desde el explorador ponemos 127.0.0.1:22770 y llamamos al fichero de login.

El puerto donde el servidor acepta las peticiones de los Clientes es el 22770. La Figura 8 presenta una respuesta del servidor, cuando un cliente le ha realizado una petición.

```
<?xml version="1.0" encoding="ISO-8859-1"??>
<xml>
<tsResponse
  xmlns:xsi="http://w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="127.0.0.1/loginResponse">
  <credentials token="12ab34cd5678cd12cd56sa34s58s2146">
    <userName name="PCCLIENTE"/>
    <profile profile="Estefania" contentUrl=""/>
    <computerName id="ESTEFANIA PC" contentUrl=""/>
    <port id="80" contentUrl=""/>
  </credentials>
</tsResponse>
</xml>
```

Figura 8: Captura de pantalla de la respuesta del servidor cuando un cliente hace un login.

#### 4.2.1 Parte del cliente

Una vez se comprobó que su funcionamiento era correcto, con las diferentes pruebas realizadas, se pasó a la conversión del lenguaje Cplusplus (C++) [5] se utilizó dicho lenguaje ya que hay muchos ordenadores cliente que no soportan el .NET. [10]

¿Que conseguimos al realizar la traducción a este lenguaje? Conseguimos, una librería dll [9], es decir una aplicación win32 que se encarga de realizar las peticiones - respuesta entre el servidor y dicho cliente.

En esta parte desarrollada está alojada en los clientes de Control Integral, es decir la empresa no interactúa en ninguna comunicación en esta parte con el cliente.

#### 4.2.2 Cliente REST

El siguiente objetivo a cumplir era tener la estructura del cliente Rest.

Se ha utilizado REST, para la optimización del tráfico, porque se puede monitorizar y es accesible a cualquier cliente HTTP.

Es una arquitectura software para sistemas hipermedia distribuidos y lo necesitamos porque cada mensaje HTTP contiene toda la información necesaria para comprender la petición.

Y de este modo, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes.

Se pasó a realizar un cliente REST en lenguaje C++, por el mismo motivo mencionado anteriormente, es decir porque hay ordenadores cliente que no soportan .NET. Se ha desarrollado una estructura con un conjunto de operaciones básicas POST, GET, PUT y DELETE que se aplican a todos los recursos de la información HTTP. GET, pide una representación del recurso especificado. Por seguridad no debería ser usado por aplicaciones que causen efectos ya que transmite información a través de la URI agregando parámetros a la URL.

Una petición GET puede ser cacheada, pasada por un

proxy y se puede crear un bookmark.

POST, envía los datos para que sean procesados por el recurso identificado. Los datos se incluirán en el cuerpo de la petición. Esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes o ambas cosas. No se puede hacer un bookmark, no se puede hacer prefetch, no se puede cachear ni se debe hacer nada con POST sin consultar con el usuario.

PUT, Sube, carga o realiza un upload de un recurso especificado (archivo), es el camino más eficiente para subir archivos a un servidor, esto es porque en POST utiliza un mensaje multiparte y el mensaje es decodificado por el servidor.

En contraste, el método PUT nos permite escribir un archivo en una conexión socket establecida con el servidor. La desventaja del método PUT es que los servidores de hosting compartido no lo tienen habilitado. Y Finalmente DELETE, Borra el recurso especificado. Cabe destacar que PUT y DELETE están en el medio entre GET y POST. La diferencia que tienen PUT y DELETE con POST, es que son idempotentes y POST no lo es. Estas dos operaciones pueden ser repetidas si es necesario. Si se altera una entrada por PUT o DELETE y luego por algún motivo se repite la operación entonces solo la primera operación tiene efecto. No se va actualizar nuevamente lo ya actualizado y no se va a borrar dos veces la misma cosa.

La Figura 9 presenta una petición del cliente que combina codificación del cliente (Peticiones codificadas en c++) y codificación del cliente REST.

Se le pasan los parámetros necesarios para poder logarse:

\*\* Ejemplo de código de la petición.

```
MonitorLoginResponse *LoginXml (const std::wstring
&userName, const std::wstring &profile, const std::wstring
&computerName, const std::wstring &port) = 0;
```

- Nombre usuario
- Nombre perfil
- Nombre ordenador
- Puerto

```
GET PCCLIENTE/Estefania/ESTEFANIAPC/80 HTTP/1.1
Host: localhost:2270
Date: Sat, 4 Jun 2016
Accept: text/xml
```

Figura 9: Captura de pantalla de la petición del cliente para logarse

### 4.3 Pasos que se realizan en la comunicación cliente/servidor

En este apartado se va a realizar en modo de captura los pasos en la comunicación Cliente/Servidor en un Login. Con el objetivo de visualizar la comunicación desarrollada y demostrar que se realiza de forma correcta para que un cliente pueda acceder al programa de gestión.

1. Cliente envía petición al servidor ()  
http://127.0.0.1:22770/ LoginResponse

En esta petición le envía las credenciales:

- Nombre usuario
- Nombre perfil
- Nombre ordenador
- Puerto

La Figura 10 muestra la petición del cliente hacia el servidor.

```
GET PCCLIENTE/Estefania/ESTEFANIAPC/80
HTTP/1.1
Host: localhost:22270
Date: Sat, 4 Jun 2016
Accept: text/xml
```

Figura 10: Petición del cliente hacia el Servidor

Esta petición la realizara mediante la interfaz que utiliza el cliente de Login. La Figura 11 muestra la interfaz que utiliza el cliente.

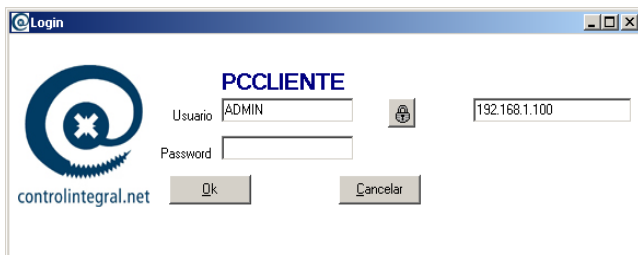


Figura 11: Imagen interfaz gráfica de usuario

2. Cliente queda a la espera de la respuesta del servidor
3. Servidor responde al cliente
  - a. Acceso OK
    - i. Le envía todas las credenciales y le añade un token único, que servirá para las futuras peticiones que este realice.
  - b. Acceso denegado
    - i. Ha llegado al número máximo de usuarios
    - ii. El usuario es incorrecto
    - iii. La contraseña es incorrecta

La Figura 12 muestra la respuesta del servidor hacia el cliente.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xml>
<tsResponse
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="127.0.0.1/loginResponse">
  <credentials token="12ab34cd5678cd12od56sa34s58s2146">
    <userName name="PCCLIENTE"/>
    <profile profile="Estefania" contentUrl=""/>
    <computerName id="ESTEFANIA PC" contentUrl=""/>
    <port id="80" contentUrl=""/>
  </credentials>
</tsResponse>
</xml>
```

Figura 12: Respuesta del servidor cuando un cliente hace un login

4. Cliente obtiene la respuesta
5. Servidor envía OK. (La Figura 13 muestra la respuesta del servidor de OK hacia el cliente).

```
HTTP/1.1 200 OK
Date: Sat, 02 Apr 2016 21:05:05 GMT
Server: localhost/1.4.19
Content-Type: text/xml
```

Figura 13: Respuesta de OK del servidor hacia el cliente

## 4.4 Comunicación Empresa VS Cliente

### 4.4.1 Servidor de licencias

El funcionamiento de este servidor es realizar una comunicación entre la empresa Control Integral y el cliente (Servidor del cliente).

Se requiere una comunicación entre el cliente y la empresa (Figura 14), para poder proporcionar al cliente diferentes servicios, como por ejemplo, poder actualizar versiones del programa, activar licencias y módulos contratados del cliente.

Cada servidor de cada cliente, realiza de forma diaria una petición a este servidor, para saber si hay algo "nuevo" según su licencia.

Un ejemplo del procedimiento que realizaría el cliente para la actualización de la versión del programa sería:

1. Control Integral, pone a disposición de los clientes una nueva versión, con novedades del programa y solución de bugs (incidencias) del programa.
2. Control Integral, abre las actualizaciones a disposición de los clientes.
3. Cliente con licencia 19587 realiza una petición al servidor de licencias
4. Servidor de licencias, revisa que ese cliente no se haya actualizado anteriormente
  - a. Si no se ha actualizado, se le descargan los ficheros de la actualización del programa
  - b. Si se ha actualizado, anteriormente no descarga nin-

guna actualización

5. En el caso que no se haya actualizado, y se descarguen los ficheros con la actualización, el servidor de licencias realiza el registro con la fecha-hora en que se han descargado los ficheros, de dicha actualización.

En el apéndice A4, se muestra un diagrama de flujo del proceso mencionado anteriormente, para que se pueda visualizar de forma gráfica.

En resumen, el servidor de licencias necesita 2 elementos básicos para su correcto funcionamiento.

Una base de datos y un fichero XML en Apéndice A3 se puede visualizar el fichero XML que se envía al cliente con la respectiva información de su número de licencia (Fechas de emisión de activaciones, módulos activados). Se diseñó la base de datos con SQL y se ha introducido la información correspondiente a las licencias de los clientes (Nº licencia, Razón social cliente, módulos activados, fechas clientes y estado de estos) en dicha base de datos.

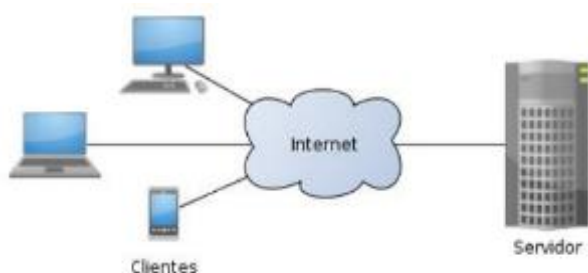


Figura 14: Captura de pantalla de la estructura entre los clientes y el servidor de licencias

## 4.5 Backend-Frontend

La selección de servidor a contratar para la infraestructura Backend-Frontend [7] será AZURE [11] quien se encargue de gestionarla.

Azure, es una plataforma abierta y flexible que admite la mayor selección de sistemas operativos, lenguajes de programación, bases de datos y dispositivos. Los servicios de Azure de pago por uso se pueden escalar o reducir verticalmente con rapidez para adaptarse a la demanda de los clientes a la hora de realizar peticiones, de forma que solo paga por lo que utiliza. Esto es algo que se tuvo en cuenta a la hora de seleccionar esta plataforma. Se seleccionó Azure, ya que los clientes no hacen peticiones en la misma franja horaria, eso conlleva a que el consumo mensual de utilización no sería muy elevado. También se seleccionó por razones de escalabilidad, disponibilidad, seguridad y facilidad de administración. Ya que en esta implementación se requiere más de una Máquina virtual, para que el servicio ofrecido al cliente tenga disponibilidad de 24 horas.

## 4.6 Disponibilidad ante emergencias

Esta parte se refiere a que un cliente puede tener un problema con el servidor, pero este debe seguir trabajando.

Se instalara un Servidor de emergencias, con el programa Monitor, para que el cliente pueda seguir trabajando de forma habitual, sin pérdida de información. Actualmente se ha creado un Monitor en las instalaciones de Control Integral donde se ha abierto el acceso para que los clientes que tengan problemas, con el monitor actual puedan seguir trabajando.

Una vez este implantada la infraestructura (máquinas virtuales, bases de datos) en la plataforma Azure, dicho servidor se trasladará siguiendo el mismo mecanismo.

Un cliente puede tener problemas de Servidor, ya sea de hardware, que en este caso el cliente tendrá que ponerse en contacto con su informático, para proceder al cambio de este.

Y a nivel de software, es decir problemas con la conexión a su base de datos, problemas de que no se conecta a la red, etc. En este caso desde el departamento de Soporte y/o departamento de I+D de Control Integral, se procede a solucionar los posibles problemas para que el cliente pueda seguir trabajando de forma habitual; Si aun así no se puede solventar el problema y necesita seguir trabajando se le facilita una IP para que pueda conectarse al servidor de emergencias y así poder seguir trabajando. Una vez se solucione el problema con su servidor, se volverá a modificar esta conexión y podrá seguir trabajando, sin perder información, ya que siempre estará trabajando bajo su base de datos.

## 5. TEST

Durante toda la realización del proyecto se han ido realizando pruebas para todas las funcionalidades que se implementaban, siguiendo la planificación se han realizado un número adecuado de pruebas de integración.

Las pruebas de rendimiento de la aplicación se realizaran una vez de tenga la plataforma de Azure para comprobar la carga que conlleva las peticiones de los clientes

## 6. CONCLUSIONES

Una vez se ha llegado a la fase final del proyecto de final de grado, los resultados son mayoritariamente positivos y muy satisfactorios. Gracias a la realización del mismo, se ha pasado de contar con un monitor que estaba solamente implementado en hardware, lo limitaba el uso a largo plazo del programa, a tener uno que ha sido programado en software.

Se ha logrado garantizar una comunicación total entre el servidor y el cliente, lo que supone haber cumplido el

requisito esencial marcado en este proyecto ya que es una de las funcionalidades básicas e imprescindibles de la aplicación implementada. Una vez superada esa fase queda por implementar la comunicación entre la empresa y el cliente, un punto en el que ya se trabaja pero que no se ha podido finalizar todavía y que se ha marcado por tanto como una línea futura de trabajo.

El hecho de ser un proyecto marcado a largo plazo y que será utilizado en la empresa anteriormente citada, en un tiempo, supone que el tiempo de elaboración de este proyecto de fin de grado es mucho mayor que el que se cuenta en un semestre, por ello a partir de ahora se seguirá trabajando para completar el trabajo hecho hasta ahora y poder lanzar la aplicación de forma satisfactoria.

Los objetivos que se han cumplido se muestran en Tabla 2.

|  | Progreso | Crítico | Prioritario | Secundario |
|--|----------|---------|-------------|------------|
| Determinar comunicación entre Control Integral y cliente   | 100%     | X       |             |            |
| Determinar estructura a realizar entre Control Integral y cliente  | 100%     | X       |             |            |
| Determinar comunicación entre Monitor y programa(Cliente)  | 100%     | X       |             |            |
| Realizar comunicación entre Monitor y programa   | 100%     | X       |             |            |
| Seleccionar tipo de servidor a contratar en la infraestructura (backend-frontend)  | 100%     |         | X           |            |
| Determinar tipo de cifrado en la comunicación  | 80%      |         | X           |            |
| Especificar, para cada funcionalidad del Monitor si es accesible para todos los sistemas operativos disponibles entre los clientes | 100%     |         | X           |            |

Tabla 2: Progreso de los objetivos específicos.

## 8. LÍNEAS FUTURAS

En este último apartado se comentaran todas aquellas funcionalidades que no se han podido realizar en este proyecto porque este es un proyecto marcado a largo plazo y no el estipulado en el marco de tiempo del trabajo de final de grado.

Ultimar los detalles y a posterior realizar la contratación de los servicios de la infraestructura de Azure. Y así poder finalizar la comunicación de una forma correcta para la implantación del proyecto.

Una vez se realicen las pruebas exhaustivas de las comunicaciones de forma abierta (sin cifrar), se procederá al desarrollo de cifrar las comunicaciones para que sean del todo seguras.

## AGRADECIMIENTOS

A mi familia, en especial a mis padres, por su apoyo incondicional siempre.

A mis compañeros de trabajo y compañeros de la carrera qué me han acompañado durante esta etapa y me han apoyado en todos los momentos.

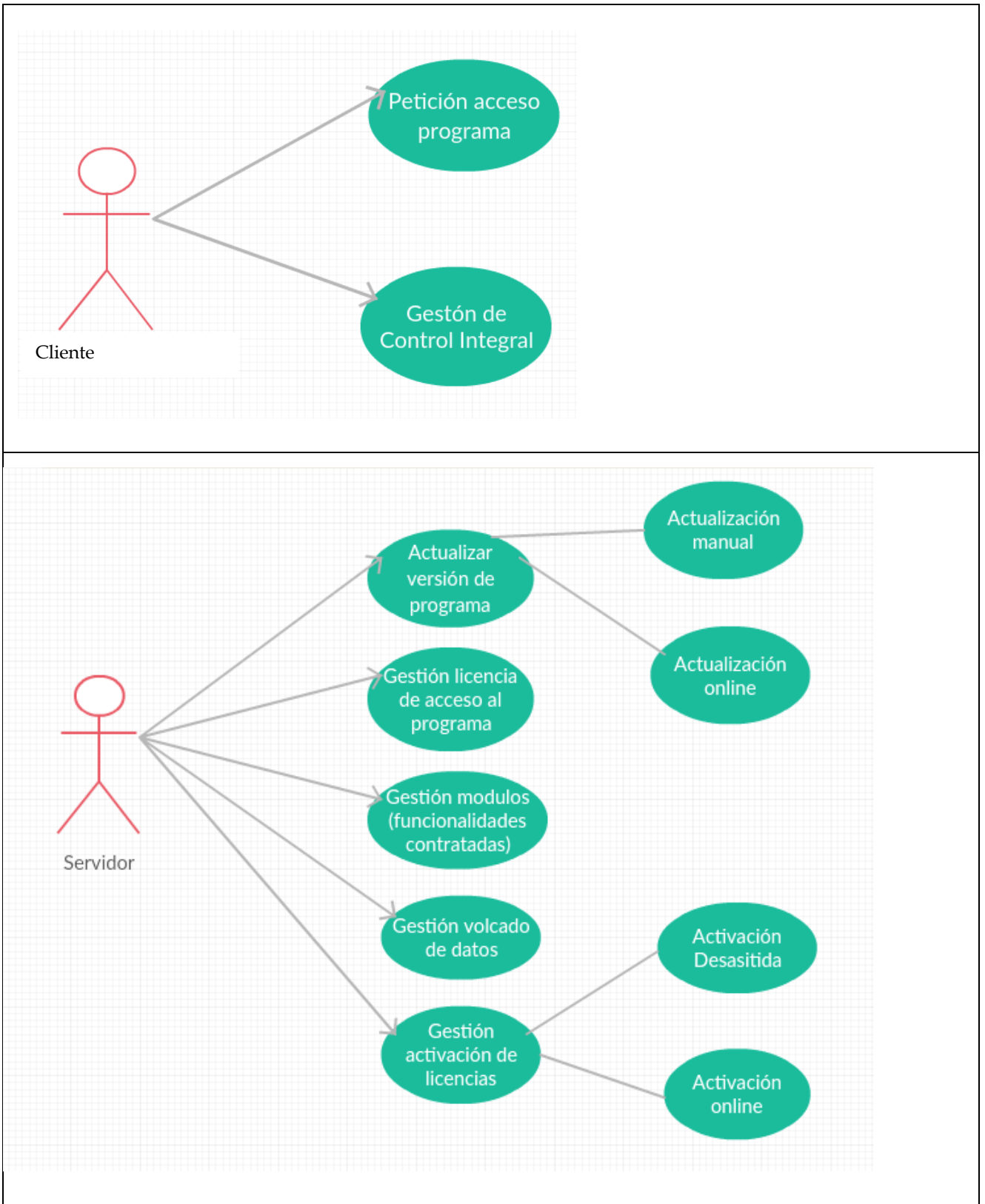
A mi tutora Katerine Díaz, y a Miquel Biosca por toda la ayuda en el desarrollo de este proyecto.

## REFERENCIAS

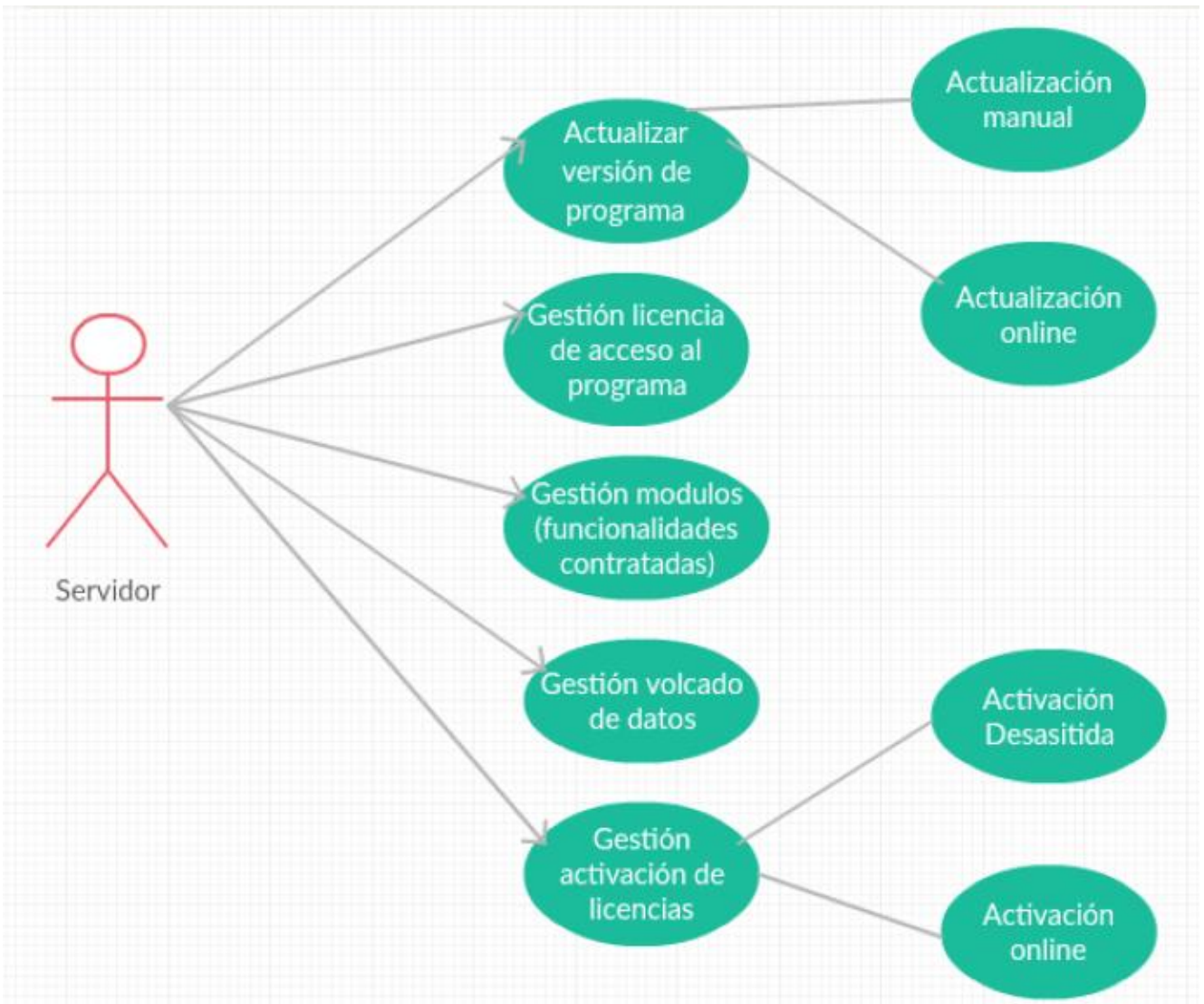
- [1] Control Integral: [www.controlintegral.net](http://www.controlintegral.net)
- [2] HaspSentinel: <http://www.safenet-inc.es/software-monetization/sentinel-hasp/>
- [3] Arquitectura Cliente/Servidor: <http://es.ccm.net/contents/148-entorno-cliente-servidor>
- [4] CSharp: <http://www.csharpya.com.ar/>
- [5] CPlusPlus: <http://www.cplusplus.com/>
- [6] Client REST: <http://www.desarrolloweb.com/articulos/que-es-rest-caracteristicas-sistemas.html>
- [7] Backend - Frontend: <http://serprogramador.es/que-es-frontend-y-backend-en-la-programacion-web/>  
<http://www.cristalab.com/blog/que-significa-backend-y-frontend-en-el-diseno-web-c106224/>
- [8] Trello: <https://trello.com/>
- [9] Dll: <https://support.microsoft.com/es-es/kb/87934>
- [10] Web oficial .NET: <https://msdn.microsoft.com/es-es/vstudio/aa496123.aspx>
- [11] Web oficial de Sql Server 2008: <https://www.microsoft.com/es-es/download/details.aspx?id=1695>

APÉNDICE

A1. DIAGRAMA DE CASO DE USO DE LA PARTE DEL CLIENTE



## A2. DIAGRAMA DE CASO DE USO DE LA PARTE DE LA EMPRESA



APENDICE

A3. Fichero XML del servidor de licencias

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<rss version="2.0">
<CustomerData>
  <LegalName>Estefania Garcia SL</LegalName>
  <cif>77745751R</cif>
  <Address>C/pau casals</Address>
  <City>Baga</City>
  <province>Barcelona</province>
</CustomerData>
<licence>
  <NLicence>99999</NLicence>
  <NSerie>23187</NSerie>
  <DateIssued>12/07/2015</DateIssued>
  <RenovationDate></RenovationDate>
  <ExpirationDate>12/08/2016</ExpirationDate>
  <DateManagement>12/07/2015</DateManagement>
  <MaxUsers>3</MaxUsers>
  <caducidadData>12/08/2016</caducidadData>
  <module>
    <module>
      <activate>True</activate>
      <description>Ecommerce</description>
      <DateIssued>12/07/2015</DateIssued>
      <ExpirationDate>12/08/2016</ExpirationDate>
    </module>
  </module>
</licence>
</rss>
  
```

A4. Diagrama de flujo: Proceso que realiza un cliente para poder actualizar el programa de Control Integral

### Como actualizar Control Integral

