# Desarrollo para Sistemas Distribuidos

Equipo: The Terribles

Proyecto: CRM Hotel

Repositorio: https://github.com/jaime264/App\_Sistemas\_Distribuidos.git

Integrantes:

Jaime Rios De La Gala – U201922339

Horario: E42B

Profesor: Héctor Saira

Ciclo: 5

## Índice

[Desarrollo para Sistemas Distribuidos](#_t24wdjehsrnc)

[Índice](#_db108ghthpbf)

[Introducción](#_ast5fg7qqypl)

[Antecedentes](#_vdveba4rixnk)

[Objetivo del proyecto](#_e7ptcljh7ir1)

[Beneficios del proyecto](#_r0nsfnrl0c1a)

[Descripción del proceso](#_c5txe4bt48dr)

[Arquitectura del sistema](#_11jef75svxhw)

[Desarrollo del sistema](#_wwqgbfjqym8e)

[Conclusiones](#_3424w6c8xq6)

### 

### 

### Introducción

Este proyecto pretende realizará una aplicación para las reservas de habitaciones en una cadena de hoteles. Con este proyecto se pretende brindar el servicio de hospedaje a personas que desean hospedarse, este servicio en la actualidad tiene un alto grado de competitividad en el mercado, ya que existen varias empresas reconocidas en el medio que lo ofrecen.

Se prestará servicios solo a personas mayores de 18 años hasta adultos mayores responsables de sus actos y destinados a un nivel socio-económico Medio, Medio-Alto y Alto.

### Antecedentes

EL caso de negocio trata sobre la implementación de una aplicación para poder hacer reservas en una cadena de hoteles, donde la aplicación será consumida a través de servicios respecto a la ubicación del hotel, de esa forma teniendo un control general de cómo se lleva las reservas en los hoteles.

### Objetivo del proyecto

* Obtener un mejor tiempo de respuesta con los servicios SOA aplicados.
* Reducir en un 50% del tiempo de carga de las reservas realizadas por el usuario.
* Tener un reporte de que hoteles son los más visitados.

### Beneficios del proyecto

* Facilidad de acceso a información actualizada
* Acceso unificado de los sistemas y base de datos
* Identificar la mayor demanda de las reservas realizadas
* Aplicación más flexible y productiva.

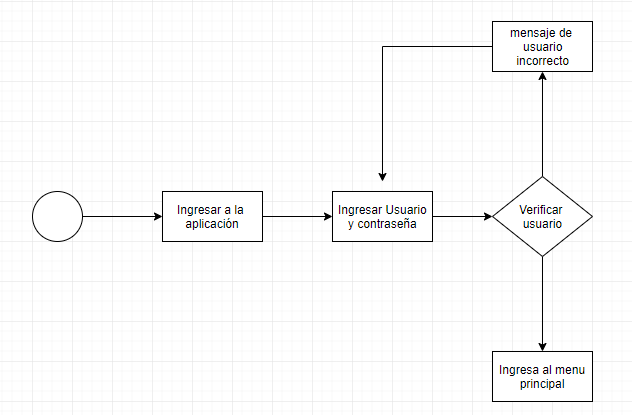
### Arquitectura de negocio (procesos)

Describa el/los proceso(s) de negocio que desea automatizar e indique las principales reglas de negocio a considerar.

**Procesos**

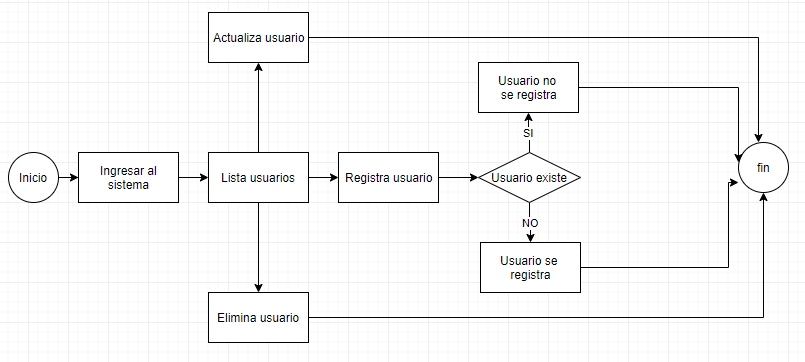
**Autenticar Usuario**

Para el ingreso del usuario deberá abrir la aplicación. Ingresar usando usuario y contraseña, en caso contrario, se contactará con el administrador.



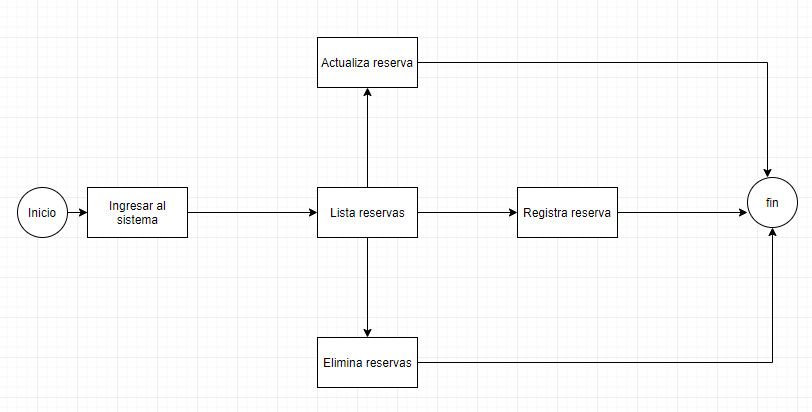
**Registrar Cliente**

Para poder tener una mejor información del tipo de cliente que tenemos, lo primero que hacemos es registrar un cliente con su nombre, apellido, tipo y numero de documento y número de teléfono.



**Registrar Reserva**

Se listan los clientes, hoteles y habitaciones disponibles por hotel, se registran las habitaciones según la solicitud del cliente, las habitaciones se pueden reservar hasta máximo un mes desde la fecha actual un cliente puede reservar varias habitaciones.



**Reglas de Negocio**

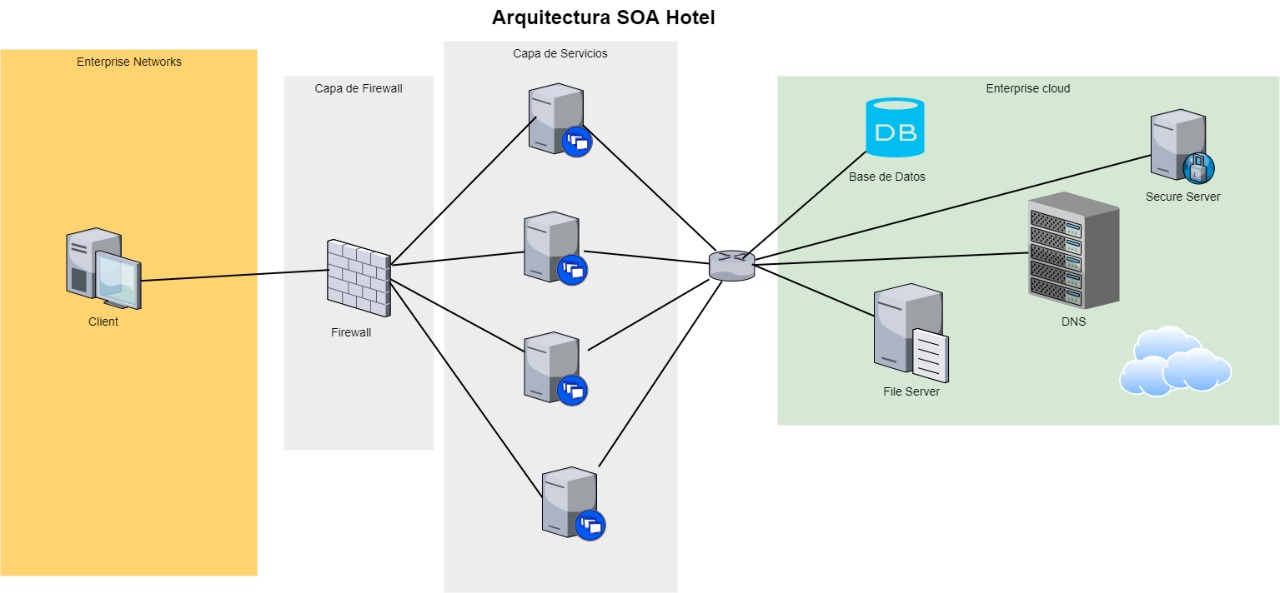
* Las reservas tendrán máximo de 5 días de vigencia para aceptar pago si no se cancela automáticamente.
* Cada tipo de habitación tiene un máximo de personas.
* No se pueden eliminar clientes que tengan reservas.
* Solo se mostrarán habitaciones disponibles por reserva.

### Arquitectura funcional (sistema)

Identifique las funcionalidades que tendrá el sistema y complete las cartillas para las historias de usuario seleccionadas; en cada cartilla se debe incluir su(s) respectiva(s) maqueta(s) y criterios de aceptación.

### Arquitectura tecnológica (software)

Modele la arquitectura de la solución. Identifique los nodos, esquema de alojamiento de cada nodo, los servicios que expone cada nodo con sus operaciones y las conexiones. Justifique la elección de las tecnologías de desarrollo distribuido a utilizar. Elabore un diagrama de despliegue del sistema utilizando UML o cualquier otra notación libre.



### Desarrollo del sistema

Se deberá ir dejando evidencia de los avances tanto para la entrega parcial como para la final (ej: pantallazos, pruebas, diagramas, enlaces a videos, etc.)

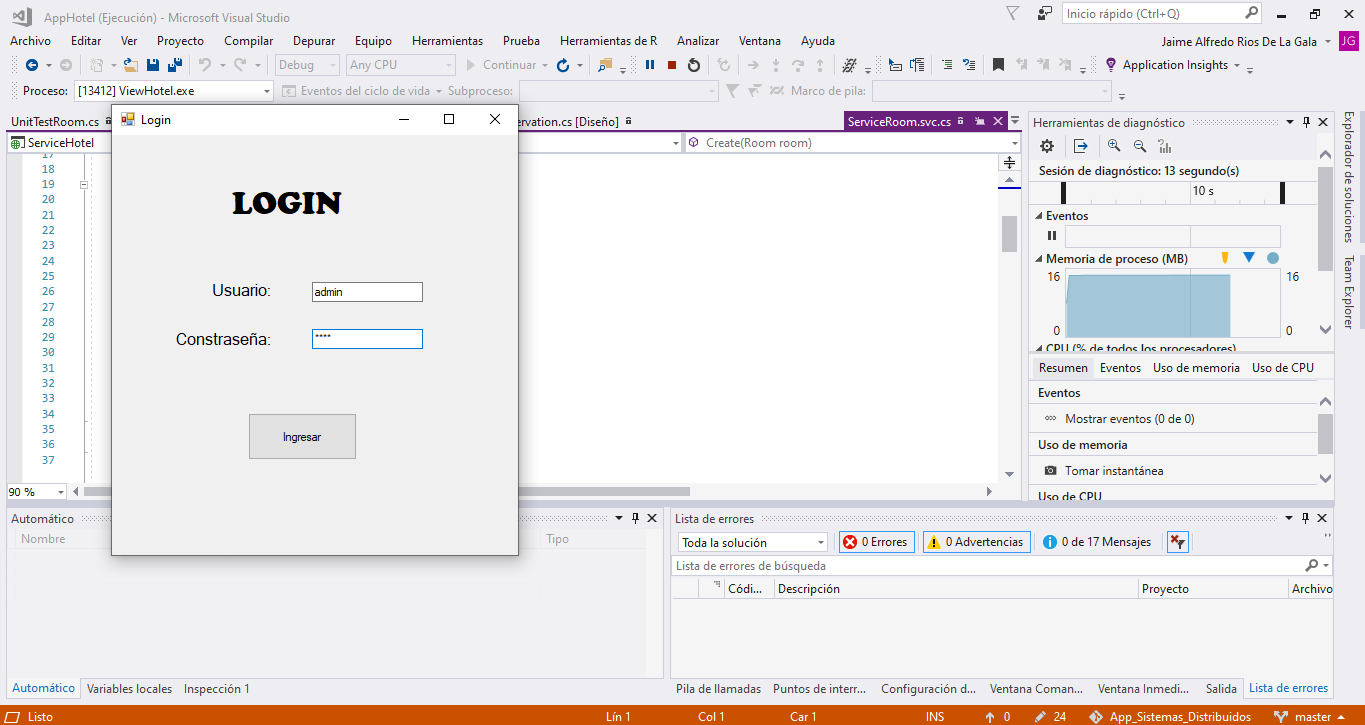
Sprint 1

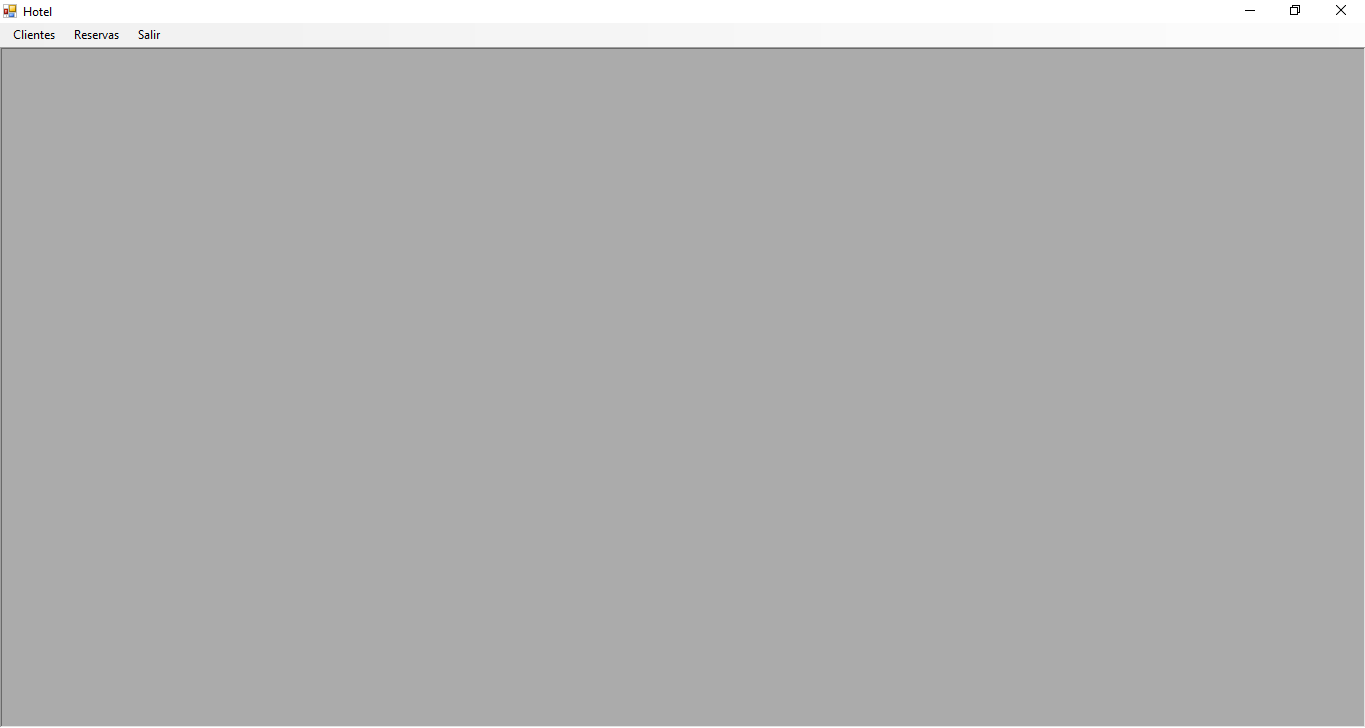
* Desarrolle la capa de presentación del sistema (prototipos navegables).
* Desarrolle los Web Services SOAP identificados (1 historia se integra con mínimo 1 servicio Web SOAP).
* Desarrolle tests unitarios automatizados para todos los servicios desarrollados.

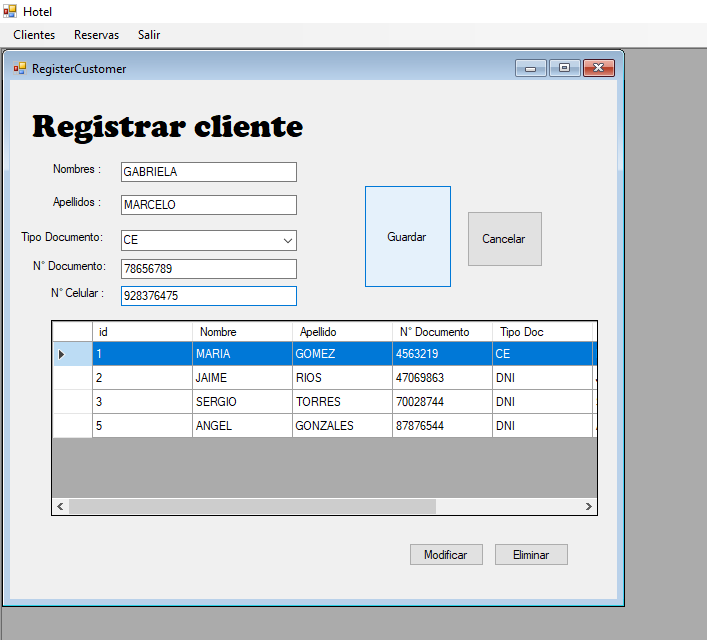
Sprint 2

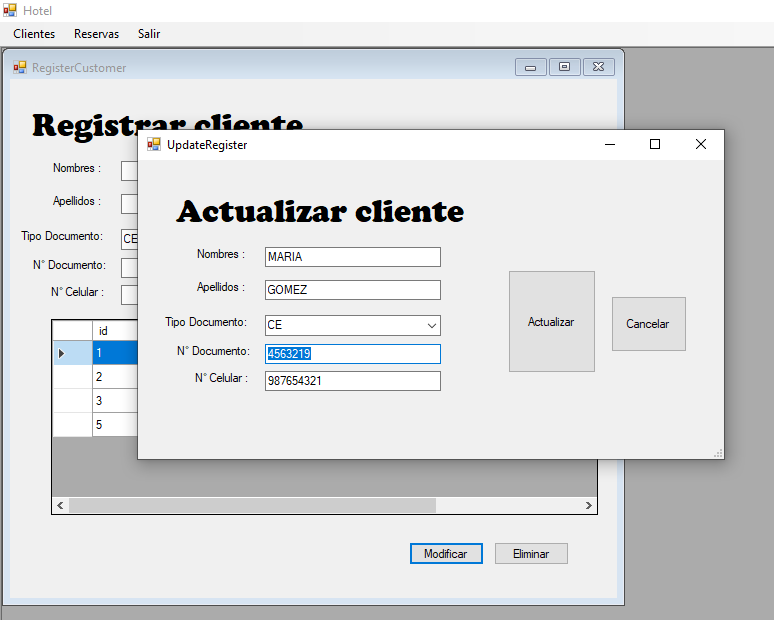
* Desarrolle la capa de presentación e integración del sistema.
* Desarrolle todos los servicios identificados (Web Services SOAP, Web Services REST y mensajería).
* Desarrolle tests unitarios automatizados para todos los servicios desarrollados.

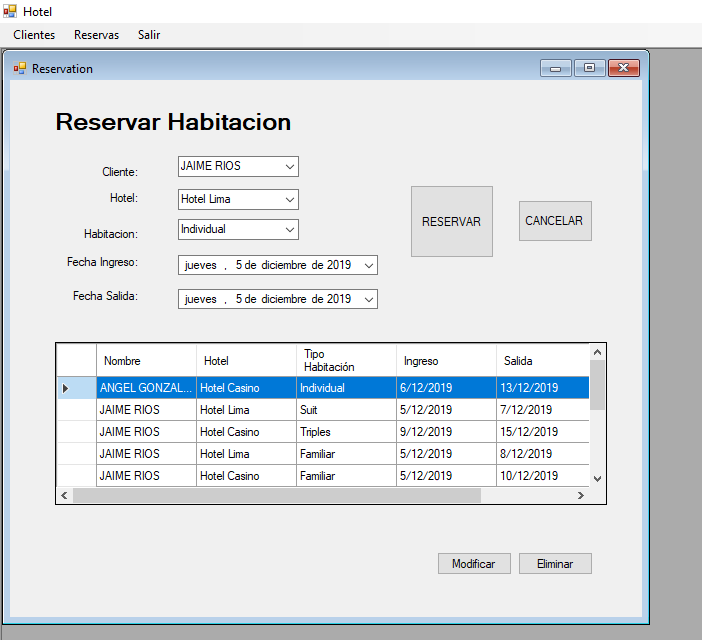
### Pantallas del sistema

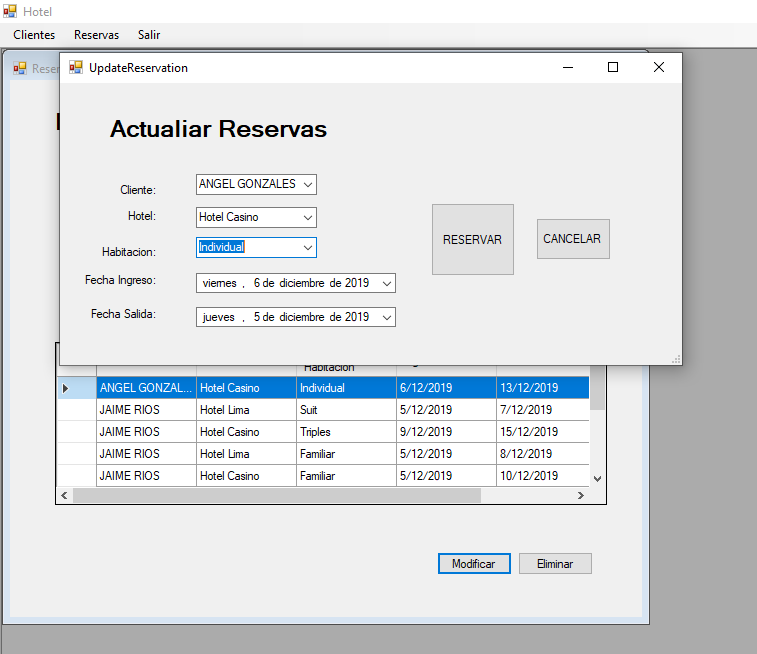












### Conclusiones

* Luego de terminar los avances del sprint, podemos validar todas las funcionalidades que tenemos y lo recomendable que es trabajar por servicios.
* Se verifica que la mayor parte de la lógica se realiza en la parte del backend en los servicios, dejando solo al frontend con pequeñas lógicas y el llamado a los servicios.
* Es fácil de implantar Windows Communication Foundation (WCF), y de desplegar en la plataforma de Microsoft.
* Al crear un servicio rest, es mucho más fácil de consumir desde web que desde aplicación de escritorio .net
* La aplicación nos permite poder realizar mantenimientos de clientes y reservas, adicional a eso se implantación los servicios de hotel y habitación para poder implementar los campos en el frontend.