**GestApaMS 0.1 (Gestion Apartamentos Microservicios)**

*Memoria borrador(muchos de los datos son notas para mi.)*

Índice

1. Información general.

3. Funcionamiento previsto (lo que quiero que haga de forma generalizada)

4. Funcionamiento (Hasta lo que he podido hacer)

2. Bases de Datos y Modelos

5. Spring Security

6. Microservicios

7. Biografía y búsquedas

8. Herramientas utilizadas

**1.Información general de la Aplicación**

Se trata de una aplicación con la que poder gestionar apartamentos vacacionales. No se trata de una gestión de reservas (aunque se podría incluir si fuera necesario). Pensado para empresas que gestione una cantidad elevada de apartamentos o pisos. Valido también para hoteles y hostales (con pocas modificaciones).

A parte del apartamento, es posible gestionar otros servicios como Parkings, Servicio Limpieza y escalonar la aplicación para cualquier otro servicio necesario.

La aplicación está orientada a funcionar con Microservicios. Debido a limitaciones con el hardware, no se ha podido fraccionar la aplicación en el proyecto, pero en la memoria se indican los distintos microservicios que habría.

**2.Microservicios finalizados**

* ClienteWeb
  + Login
  + Seguridad
  + Página de gestión de usuarios
  + Página de gestión de Apartamentos
  + Página de gestión de Reservas
* Microservicio de Usuarios
* Microservicio de Apartamentos
* Microservicio de Reservas
* Microservicio de Conversión de entidades
* Microservicio de gestión de fotos

**3.Microservicios pendientes**

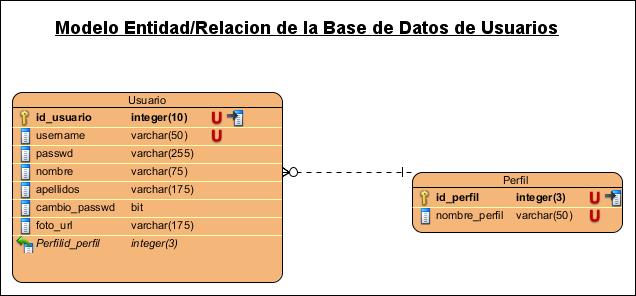
* ClienteWeb
  + Limpieza
  + Parking
  + Servicios
* Microservicio de Limpieza
* Microservicio de Parking
* Microservicio de Servicios

**4.Bases de Datos y Modelos**

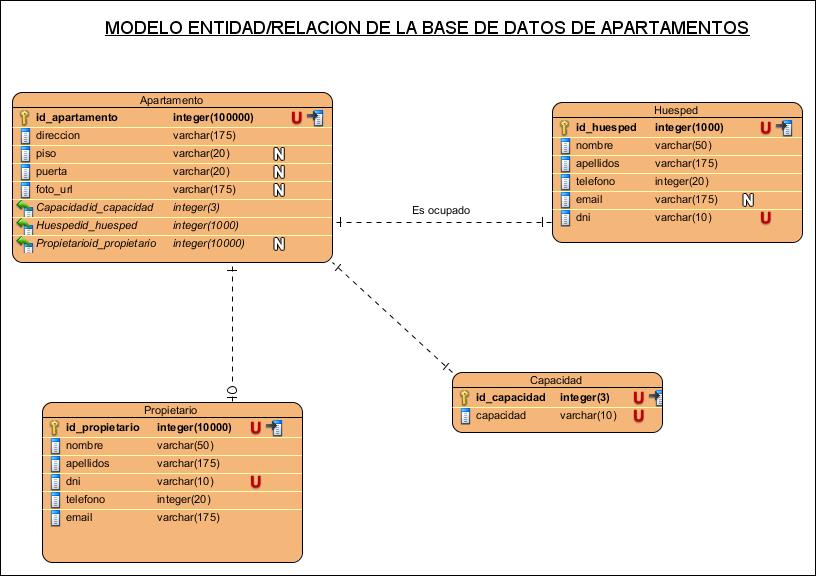
Las bases de datos se han realizado en H2(in memory) para que se pueda probar mejor. Cada una de ellas incluye un archivo data.sql con instrucciones para la inserción de datos de prueba.

1. UsuariosDB: Base de datos de los usuarios que tendrán acceso y su nivel de acceso o perfil;

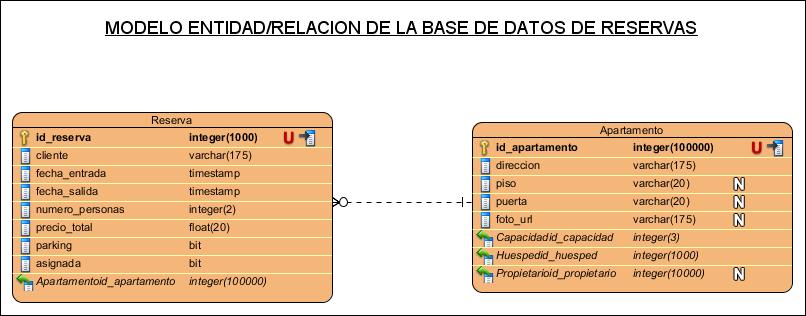
ROLE\_ADMIN, ROLE\_USER, ROLE\_CLEAN, etc.



1. ApartamentosDB: Base de datos de los apartamentos/pisos que dispone el cliente para su ocupación. Incluye información del huésped que estaría al cargo de la habitación, la capacidad del apartamento y el propietario del mismo.



1. ReservasDB: Base de datos de las reservas que tiene el cliente y los apartamentos relacionados.



* Generadores de Datos
* Modelos utilizados (Usuario, Apartamento, PlazaParking, etc. )

**Spring Security**

Explicar el Microservicio para autentificarse en la web. Uso de Feign para consultar usuario y password. Definir recursos y terminar de definir el acceso a las paginas según el perfil.

**Microservicios**

1. Usuarios(“gestión-usuarios”)

Microservicio encargado de la gestión de los usuarios y todos los datos relacionados. Comunica directamente con la base de datos, en este caso H2 en memoria para pruebas.

1. Apartamentos(“gestión-apartamentos”)

Microservicio encargado de la gestión de los apartamentos/pisos/habitaciones que pueda tener el cliente. Base de datos H2 en memoria para pruebas. También se incluye la gestión de los propietarios. Esta función podría separarse en otro Microservicio.

1. Reservas (“gestión-reservas”)

Microservicio encargado de gestionar las reservas. Base de datos propia en H2 en memoria. También se encargaría de procesar las reservas entrantes de otros sitios, he utilizado por ejemplo de Booking.com, utilizo un response de prueba y gracias a un conversor de XML a objeto puedo tratar y transformar esas reservas a las de la propia aplicación.

**Biografía y búsquedas**

Lo que he tenido que buscar y de donde lo he buscado para hacerlo.

* Thymeleaf
* SpringSecurity
* JAXB API (xml to Object)

**Herramientas utilizadas**

* Intellij