

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**DISEÑO DE SOFTWARE**

**PROYECTO DE CURSO**

**REPORTE**

**FEBRERO DEL 2020**

**GRUPO 11**

**INTEGRANTES:**

* León Chang David Kimmwah.
* Holguín Sarango Joselyn Elizabeth.
* Agila Pinto Nicole Salomé.

**DOCENTE**: Jurado Mosquera David Alonso.

**PARALELO: 1**

**Guayaquil-Ecuador**

**2019-2S**

**ÍNDICE**

[**1.** **DESCRIPCIÓN DEL PROYECTO** 3](#_Toc31579918)

[**2.** **PROPUESTA** 4](#_Toc31579919)

[2.2 DIAGRAMA DE CASOS DE USO 4](#_Toc31579920)

[2.3 DESCRIPCIÓN DE CASOS DE USO (CLIENTE) 5](#_Toc31579921)

[2.4 DESCRIPCIÓN DE CASOS DE USO (VENDEDOR) 5](#_Toc31579922)

[2.5 DESCRIPCION DE LOS PATRONES DE DISEÑO ESCOGIDOS PARA LA IMPLEMENTACIÓN 6](#_Toc31579923)

[DECORATOR: 6](#_Toc31579924)

[SINGLETON: 6](#_Toc31579925)

[BUILDER: 6](#_Toc31579926)

[2.6 DIAGRAMA DE CLASES (UML) 7](#_Toc31579927)

[2.7 DIAGRAMAS DE SECUENCIAS. 8](#_Toc31579928)

[2.8 DIAGRAMA NORMALIZADO DEL MODELO FÍSICO DE LA BASE DE DATOS. 14](#_Toc31579929)

[**3.** **PROTOTIPO** 15](#_Toc31579930)

[3.1 DOCKER 15](#_Toc31579931)

[3.2 INFORME CALIDAD DEL CODIGO 15](#_Toc31579932)

[3.3 PRUEBAS UNITARIAS 16](#_Toc31579933)

# **DESCRIPCIÓN DEL PROYECTO**

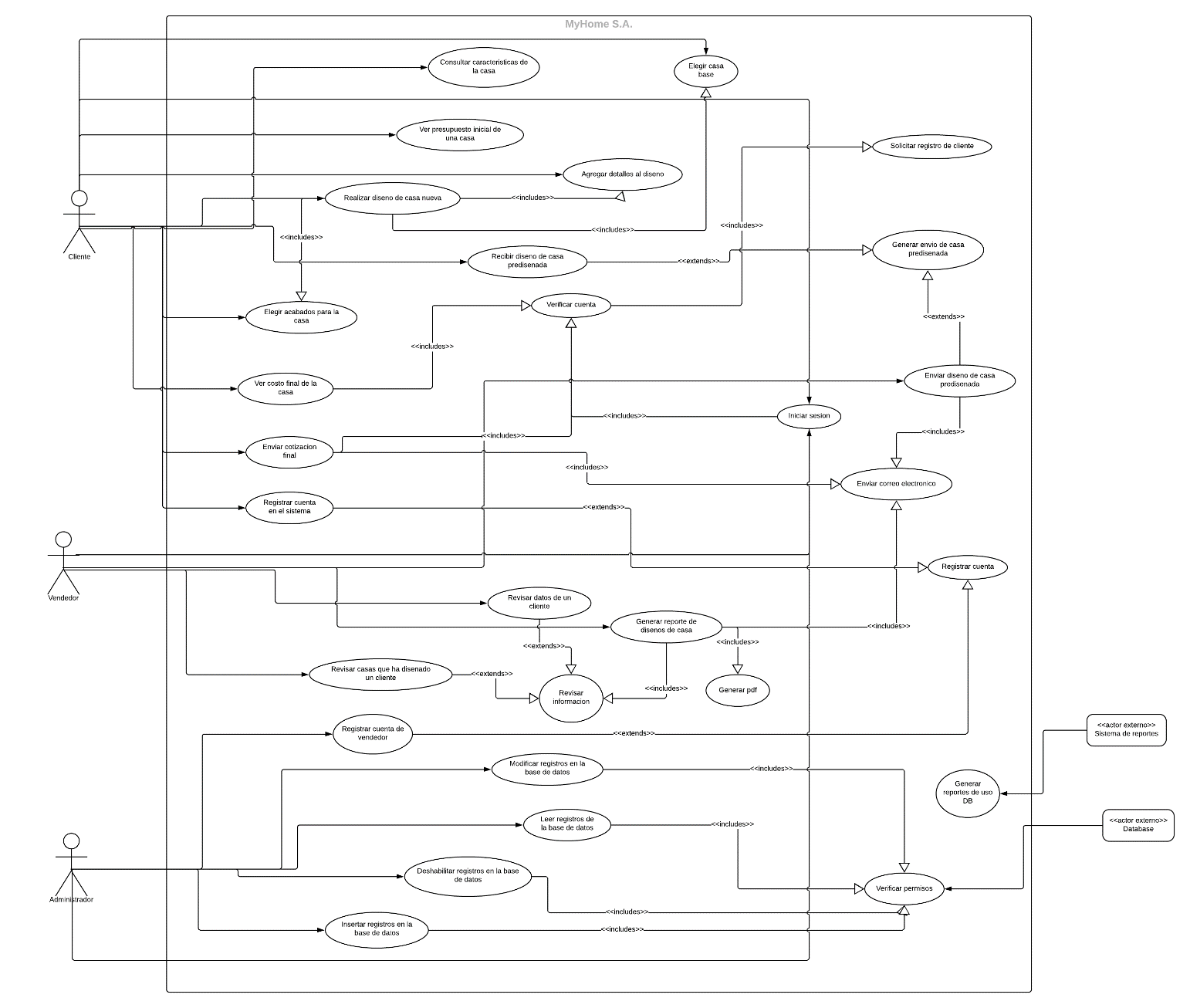
La constructora MyHome S.A desea crear un sistema que le permita diseñar a sus clientes y clientes potenciales las casas de sus sueños, pueden ser estas casas básicas como Cielo, Paraíso y Oasis, estos clientes pueden ser registrados o sin registrar, a su vez es importante recalcar que su modelo de negocio va a funcionar como un quiosco y por el momento contará con una sola oficina, la misma que se va a conectar al sistema que se enlaza directamente con la base de datos, en donde se muestra datos referentes a empleados y a los clientes registrados.

Este sistema permite el acceso a clientes registrados, vendedores y administrador, sin embargo, los clientes no registrados también tienen ciertas funcionalidades a las cuales pueden acceder pero para completar cualquier proceso deben registrarse de manera obligatoria, los clientes pueden diseñar su casa, agregar detalles y enviar su diseño vía email en PDF, también pueden observar el costo final, si no cuenta como un cliente registrado y ha diseñado su casa, solo puede obtener su plano simplemente si se registra y su diseño se envía automáticamente, además es importante considerar que si agrega algún detalle extra como acabados o elementos se les incorpora un valor adicional a su cotización.

Un empleado vendedor sólo puede ser registrado por un administrador, también puede inspeccionar datos del cliente y los diseños que han realizado, puede generar reportes en PDF y recibirlos mediante correo, ofrece opciones de casas prediseñadas a los clientes mediante email, mientras tanto un administrador puede hacer operaciones CRUD dentro del sistema, pero para poder realizar DELETE (eliminación) se debe modificar su estado a inactivo, para no perder informacion.

# **PROPUESTA**

## DIAGRAMA DE CASOS DE USO



**Figura n°1:** Diagrama de casos de uso de los actores que interactúan en el sistema (Cliente, vendedor, administrador).

## DESCRIPCIÓN DE CASOS DE USO (CLIENTE)

Imagen que contiene texto

Descripción generada automáticamente

**Figura n°2:** Descripción de caso de uso de la cliente.

## DESCRIPCIÓN DE CASOS DE USO (VENDEDOR)

Imagen que contiene texto

Descripción generada automáticamente

**Figura n°3:** Descripción de caso de uso de vendedor.

## DESCRIPCION DE LOS PATRONES DE DISEÑO ESCOGIDOS PARA LA IMPLEMENTACIÓN

### 

### DECORATOR:

Utilizamos el patrón de diseño estructural DECORATOR, para agregar nuevos acabados a la casa base escogida por el usuario. Cada que se agrega un nuevo acabado, se suma el valor del acabado al precio final de la casa diseñada, también se agrega un objeto acabado a la lista de acabados de la casa para tener un registro de los acabados que se agregaron.

### 

### SINGLETON:

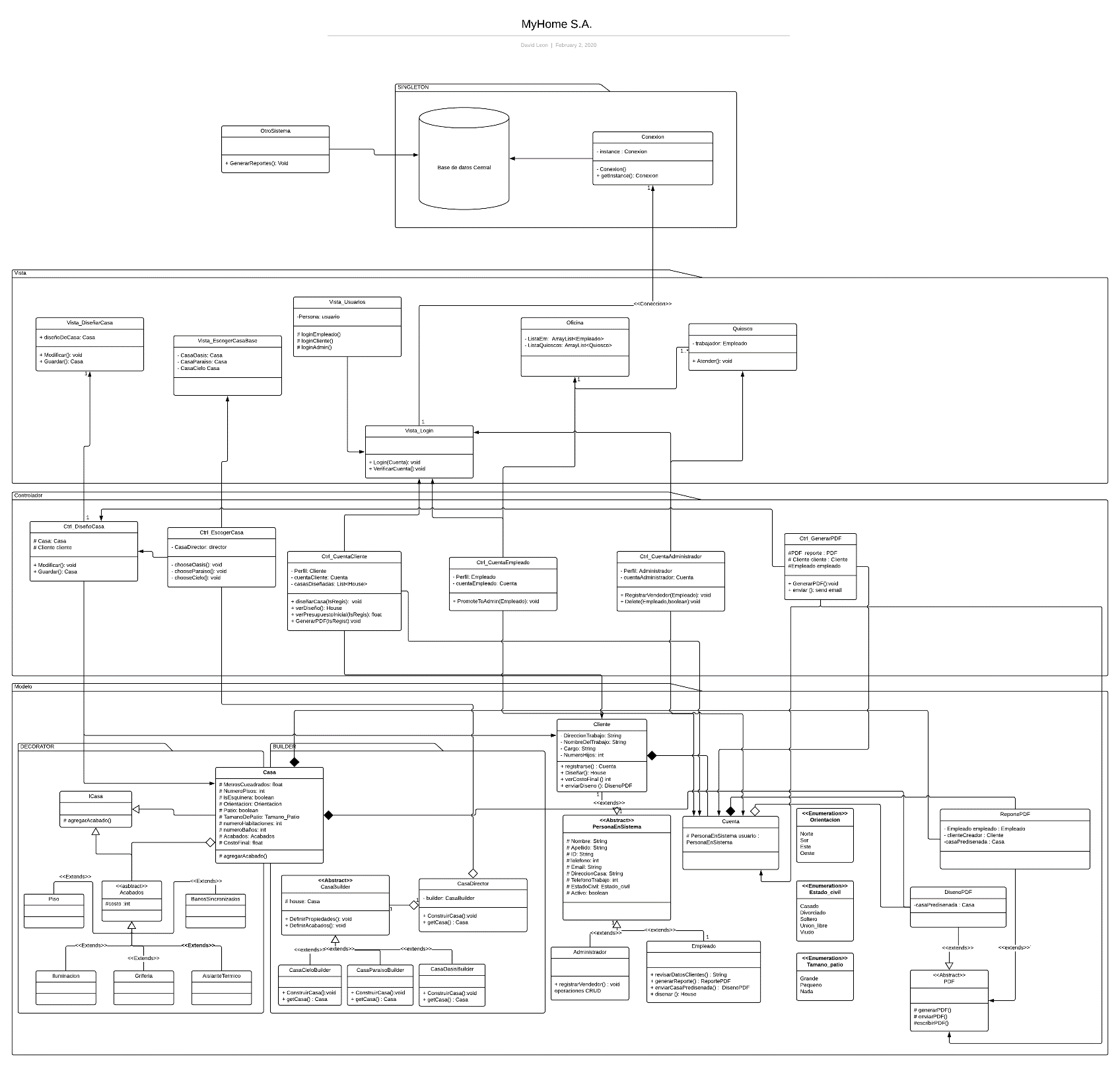
Utilizamos el patrón de diseño creacional SINGLETON para crear una sola instancia de la conexión a la base de datos. Para que solo se use esa única instancia en todos los querys que se creen para obtener y agregar informacion a la base.

### 

### BUILDER:

Utilizamos el patrón de diseño creacional BUILDER para crear los 3 tipos de casas base, con sus diferentes características entre ellas.

## DIAGRAMA DE CLASES (UML)



**Figura n°5:** UML de la empresa MyHome S.A.

## DIAGRAMAS DE SECUENCIAS.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

**Figura n°6:** Diagrama de secuencia referente al ingreso al sistema del cliente registrado.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

**Figura n°7:** Diagrama de secuencia referente a generar reporte por parte del cliente.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

**Figura n°8:** Diagrama de secuencia referente a realizar un diseño de casa nuevo

**Figura n°8:** Diagrama de secuencia referente a realizar diseño de casa nueva.

Imagen que contiene mapa, texto

Descripción generada automáticamente

**Figura n°9:** Diagrama de secuencia referente a registrar un vendedor.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

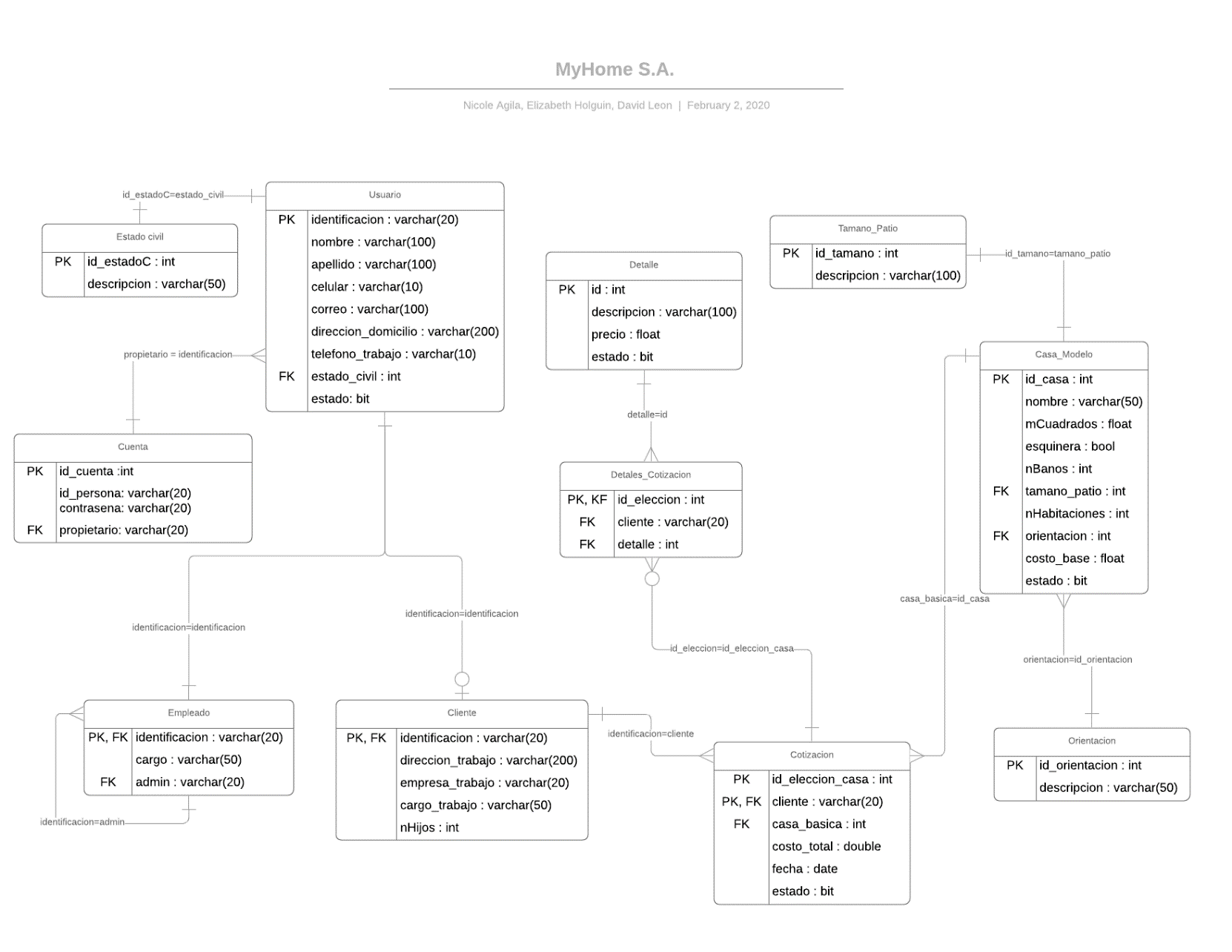
**Figura n°10:** Diagrama de secuencia referente a registro de un nuevo cliente.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

**Figura n°11:** Diagrama de secuencia referente al ingreso de vendedores a MyHome S.A.

## DIAGRAMA NORMALIZADO DEL MODELO FÍSICO DE LA BASE DE DATOS.



**Figura n°12:** Diagrama normalizado del modelo físico de la base de datos MyHome S.A.

# **PROTOTIPO**

## DOCKER

**Comandos para crear la base en Docker**

docker images

docker ps

docker pull mysql

docker run -p 3308:3306 --name MyHomeDataBase -e MYSQL\_ROOT\_PASSWORD=root -d mysql

docker exec -it MyHomeDataBase mysql -uroot -p

show databases;

**Conexión desde el proyecto en NetBeans para conexión por IP.**

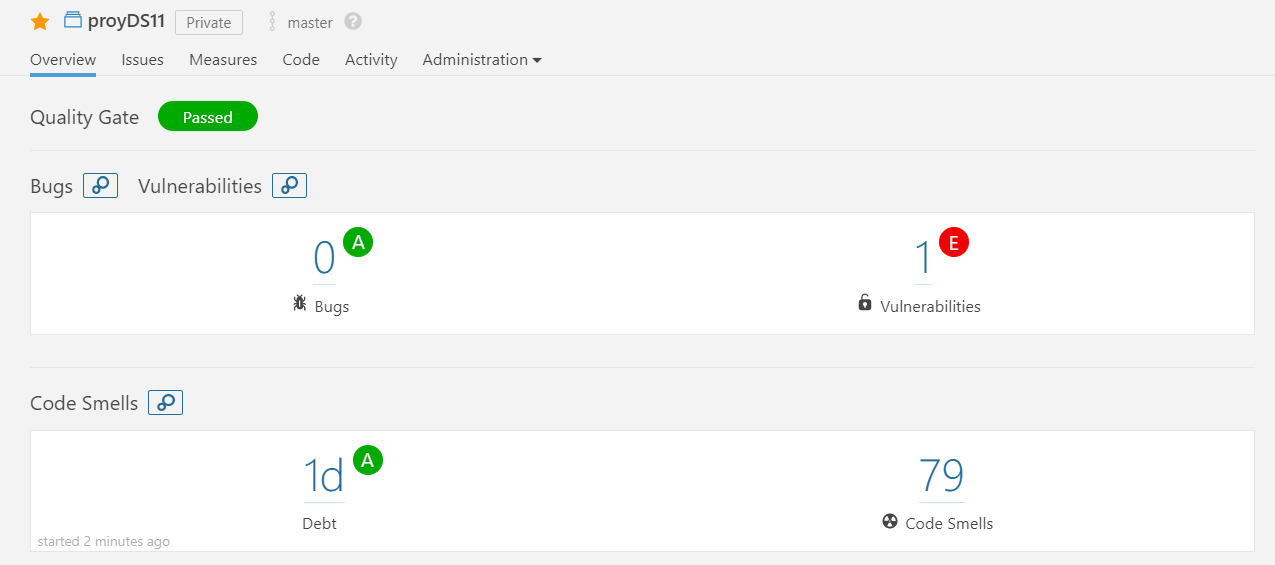
Connection con = null;

String url = "jdbc:mysql://192.168.99.100:3308/MyHomeDataBase";

String user= "root";

String pass= "root";

## INFORME CALIDAD DEL CODIGO



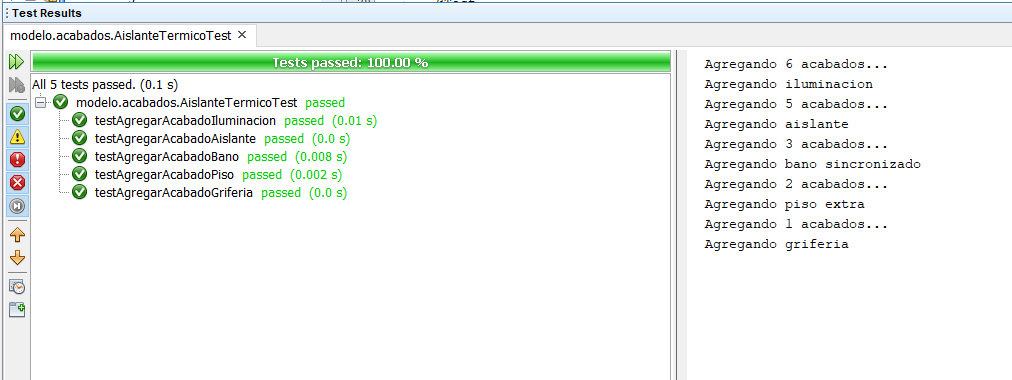
<http://200.126.12.164/dashboard?id=nagila2019%3AproyDS11.3>

Tras revisar los code smells que detectaba Sonar Qube, nos dimos cuenta de que casi el 90% es por los nombres de las variables, que no siguen el estándar de java, sin embargo, no las cambiamos, ya que, si se entienden, se refiere mas a los nombres de las variables de las vistas, que están como btn\_guardar, txt\_user, etc.

El otro 10% es por unos métodos que todavía no están terminados y salen vacíos.

La vulnerabilidad es la clave de la base de datos, que se ve en texto tal cual.

## PRUEBAS UNITARIAS



Unas de las pruebas unitarias que realizamos fueron al agregar acabados a la casa seleccionada, comparamos el valor final de la casa al agregar cierta cantidad de un acabado en específico.

