



**UNIVERSIDADE  
DE VIGO**

**ESCOLA SUPERIOR DE ENXEÑARÍA INFORMÁTICA**

Memoria do Traballo de Fin de Grao que presenta

**D. Víctor Díaz Seoane**

para a obtención do Título de Graduado en Enxeñaría Informática

**Ubiko: aplicación web para a localización de pacientes.**

# MANUAL TÉCNICO



Xuño, 2016

**Traballo de Fin de Grao Nº:** El 15/16-012

**Titor/a:** Alma María Gómez Rodríguez

**Cotitor:** David Ramos Valcárcel

**Área de coñecemento:** Linguaxes e Sistemas Informáticos

**Departamento:** Informática



## Índice

1.INTRODUCCIÓN.....	5
2.DEFINICIÓN DE REQUISITOS.....	5
2.1.Requisitos funcionales.....	6
2.2.Requisitos no funcionales.....	6
3.ANÁLISIS.....	7
3.1.Casos de Uso.....	7
3.1.1. Diagrama de Casos de Uso.....	7
3.1.2. Descripción detallada de Casos de Uso.....	8
3.2.Modelo de dominio.....	15
3.3.Diagramas de secuencia.....	16
3.3.1.Acceder al sistema:.....	16
3.3.2.Dar de alta un paciente:.....	16
3.3.3.Buscar un paciente:.....	17
3.3.4.Modificar ubicación de un paciente:.....	17
3.3.5.Dar de baja un paciente:.....	18
3.3.6.Consultar el seguimiento de un paciente:.....	18
3.3.7.Consultar el BOX del centro médico:.....	19
3.3.8.Consultar las estadísticas de un centro médico:.....	19
3.3.9.Salir del sistema:.....	20
3.4.Interfaz de usuario.....	20
3.4.1.Pantalla ‘Admisión’:.....	20
3.4.2.Pantalla ‘Seguimiento’:.....	21
3.4.3.Pestaña ‘BOX’:.....	21
3.4.4.Pantalla ‘Estadísticas’:.....	22
3.4.5.Ventana ‘LogIn’:.....	22
4.MODELO DE DATOS.....	23
4.1.Modelo entidad-relación.....	23
4.2.Paso a tablas.....	24
4.2.1.Tabla Usuario.....	24
4.2.2.Tabla Cama.....	24
4.2.3.Tabla ubicación Paciente.....	25
4.2.4.Tabla Paciente.....	26
4.2.5.Tabla Localización.....	26
5.DISEÑO.....	27
5.1.Diagramas de clases parciales y diagramas de secuencia del sistema.....	27
5.1.1.Acceder al sistema.....	27
5.1.2.Dar de alta un paciente.....	28
5.1.3.Buscar un paciente.....	29
5.1.4.Modificar ubicación de un paciente.....	30
5.1.5.Dar de baja un paciente.....	31
5.1.6.Consultar Seguimiento/BOX/Estadísticas.....	32
5.1.7.Salir del sistema.....	34

5.2.Definición de diagramas .....	35
6.DETALLES DE IMPLEMENTACIÓN .....	41
6.1.Lenguajes utilizados .....	41
6.2.Tareas más relevantes.....	42
7.PRUEBAS .....	45

## 1. INTRODUCCIÓN.

El proyecto Ubiko consiste en el desarrollo de una aplicación web que pretende solucionar la problemática que genera la localización de los diferentes pacientes que se encuentran en el área de urgencias de un centro médico. El usuario podrá tanto localizar como modificar la localización de cada paciente durante cualquier momento de su estancia en el centro. El objetivo es mantener al paciente siempre localizado de forma que se reduzcan notablemente los tiempos empleados en la búsqueda y por tanto, el traslado y la estancia de los citados pacientes en urgencias.

El sistema está diseñado para funcionar sobre los navegadores más populares del momento en un SO Microsoft.

La metodología de desarrollo utilizada en este proyecto tiene un carácter iterativo e incremental, mediante la cual en base a diferentes *Sprints* realizamos las diferentes funcionalidades del sistema. En este manual técnico se recoge toda la documentación (Análisis y Diseño) resultante del desarrollo en su versión final, es decir, la versión obtenida tras la última iteración del desarrollo.

## 2. DEFINICIÓN DE REQUISITOS.

En este apartado se exponen los requisitos que se consideran como funcionalidades o características mínimas para determinar que la aplicación está finalizada.

Los requisitos funcionales (identificados por RF#) definen aquellas acciones o funcionalidades que describen el comportamiento de la aplicación.

Los requisitos no funcionales (identificados por RNF#) describen aquellas características que aunque no influyen en el correcto funcionamiento de la aplicación, interfieren en la experiencia del usuario con la aplicación.

## 2.1.Requisitos funcionales.

Requisito	Descripción
RF1	El usuario debe poder acceder al sistema.
RF2	El usuario debe poder dar de alta en el sistema (admitir) a un número ilimitado de pacientes.
RF3	El usuario debe poder buscar un paciente previamente dado de alta en el sistema.
RF4	El usuario debe poder modificar la ubicación actual de un paciente.
RF5	El usuario debe poder dar de baja un paciente.
RF6	El usuario debe poder consultar el seguimiento de un paciente.
RF7	El usuario debe poder consultar el estado del BOX del centro médico.
RF8	El usuario debe poder consultar las estadísticas del centro.
RF9	El usuario debe poder cerrar su sesión en el sistema

## 2.2.Requisitos no funcionales.

Requisito	Descripción
RNF1	Las ubicaciones deben ser fácilmente identificables.
RNF2	El arrastre de las ubicaciones debe ser fluido.
RNF3	El cargado y visualización de las ubicaciones de un paciente debe ser instantáneo.

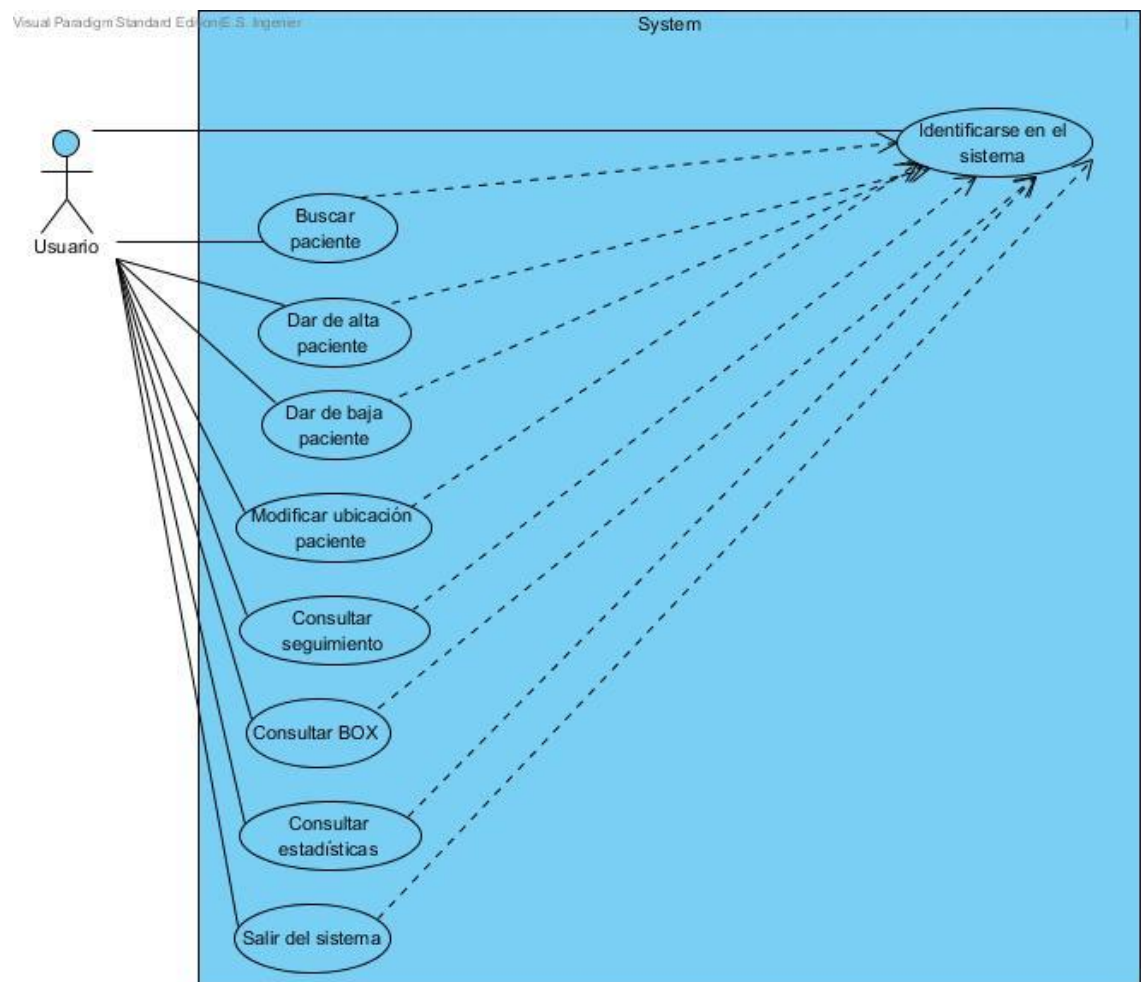
### 3. ANÁLISIS

#### 3.1.Casos de Uso

##### 3.1.1. Diagrama de Casos de Uso

Los diagramas de casos de uso, sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.

En el diagrama que se expone a continuación, podemos observar las diferentes funcionalidades expuestas en los Requisitos Funcionales (2.1).



En esta aplicación, el usuario debe estar identificado en el sistema para poder realizar el resto de acciones. De ahí la dependencia mostrada en el gráfico anterior.

### 3.1.2. Descripción detallada de Casos de Uso

En este apartado se detallarán las características de los diferentes Casos de Uso, que incluirán:

- Breve descripción.
- Actor.
- Precondiciones.
- Postcondiciones.
- Flujo de Eventos.
- Flujo de Eventos alternativo (si hubiere).

#### *Acceder al sistema:*

Caso de Uso	ACCEDER AL SISTEMA		
Actor	Usuario		
Breve Descripción	El usuario se identifica para poder comenzar a interactuar con el sistema.		
Precondiciones	La pantalla mostrada es la correspondiente al 'LogIn'.		
Postcondiciones	La pantalla mostrada es la correspondiente a 'Admisión'.		
Flujo de Eventos		Usuario	Sistema
	1	Introduce sus credenciales	
	2	Pulsa el botón (Flecha)	
	3		Permite el acceso
	4		Muestra la ventana 'Admisión'
Flujo alternativo [A3]		Usuario	Sistema
	1		Deniega el acceso
	2		Muestra la ventana 'LogIn'

En este caso, si el usuario introduce unos datos erróneos no se le permitirá el acceso al sistema, mostrándose nuevamente la pestaña de LogIn.



*Dar de alta un paciente:*

Caso de Uso	DAR DE ALTA A UN PACIENTE		
Actor	Usuario		
Breve Descripción	El usuario da de alta un nuevo paciente en el sistema		
Precondiciones	La pantalla mostrada es la correspondiente al 'Admisión'. El usuario debe estar identificado en el sistema.		
Postcondiciones	La pantalla mostrada es la correspondiente a 'Seguimiento'.		
Flujo de Eventos		Usuario	Sistema
	1	Introduce los datos del paciente	
	2	Pulsa el botón (Flecha)	
	3		Guarda en base de datos el nuevo paciente
	4		Muestra la ventana 'Seguimiento'
Flujo alternativo [A2]		Usuario	Sistema
	1		Carga los datos del paciente.
	2		Muestra la ventana 'Seguimiento'

En caso de que el usuario intente dar de alta un paciente ya existente, el sistema cargará los datos de dicho paciente y los mostrará en la ventana 'Seguimiento'

*Buscar un paciente:*

Caso de Uso	BUSCAR UN PACIENTE		
Actor	Usuario		
Breve Descripción	El usuario busca un paciente introduciendo el NHC.		
Precondiciones	La pantalla mostrada es la correspondiente a 'Seguimiento'. El usuario debe estar identificado en el sistema. El paciente a buscar debe existir en el sistema.		
Postcondiciones	La pantalla mostrada es la correspondiente a 'Seguimiento' con los datos del paciente en cuestión cargados.		
Flujo de Eventos		Usuario	Sistema
	1	Introduce el NHC del paciente en la barra de búsqueda	
	2	Pulsa el botón (Flecha)	
	3		Carga los datos del paciente en cuestión
Flujo alternativo [A3]		Usuario	Sistema
	1		Muestra un mensaje de error.
	2		Muestra la ventana 'Admisión'

Si el usuario busca un paciente no existente en el sistema, se le mostrará la ventana 'Admisión'

*Modificar ubicación de un paciente:*

Caso de Uso	MODIFICAR UBICACIÓN DE UN PACIENTE		
Actor	Usuario		
Breve Descripción	El usuario arrastra ubicaciones al contenedor para modificar la localización actual del paciente.		
Precondiciones	La pantalla mostrada es la correspondiente a 'Seguimiento'. El usuario debe estar identificado en el sistema. Debe haber un paciente cargado.		
Postcondiciones	La pantalla mostrada es la correspondiente a 'Seguimiento' con los datos del paciente en cuestión actualizados.		
Flujo de Eventos		Usuario	Sistema
	1	Arrastra la/s ubicaciones deseadas al contenedor	
	2		Guarda en base de datos la ubicación introducida.
	3		Recarga la pantalla con los datos del paciente actualizados.
Flujo alternativo [A2]		Usuario	Sistema
	1		El sistema no permite introducir la ubicación.

En caso de introducir una ubicación final o bien una ubicación igual a la anterior, el sistema no permite el guardado.

*Dar de baja un paciente:*

Caso de Uso	DAR DE BAJA UN PACIENTE		
Actor	Usuario		
Breve Descripción	El usuario da de baja un paciente.		
Precondiciones	La pantalla mostrada es la correspondiente a ‘Seguimiento’. El usuario debe estar identificado en el sistema. Los datos del paciente deben estar cargados.		
Postcondiciones	La pantalla mostrada es la correspondiente a ‘Seguimiento’.		
Flujo de Eventos		Usuario	Sistema
	1	Arrastra una ubicación equivalente a un estado final	
	2		Guarda en base de datos la ubicación y el estado final.
	3		Libera la cama ocupada en el BOX de ser el caso
	4		Muestra la ventana ‘Seguimiento’ impidiendo que se introduzcan más ubicaciones.

*Consultar el seguimiento de un paciente:*

Caso de Uso	CONSULTAR EL SEGUIMIENTO DE UN PACIENTE	
Actor	Usuario	
Breve Descripción	El usuario consulta el seguimiento de un paciente.	
Precondiciones	La pantalla mostrada es la correspondiente a 'Seguimiento'. El usuario debe estar identificado en el sistema . Los datos del paciente deben estar cargados.	
Postcondiciones	La pantalla mostrada es la correspondiente a 'Seguimiento'.	
Flujo de Eventos	Usuario	Sistema
	1	Pulsa en la opción del menú: 'Seguimiento'.
	2	Muestra la información correspondiente al paciente.

*Consultar el BOX del centro médico:*

Caso de Uso	CONSULTAR EL BOX DE UN CENTRO MÉDICO	
Actor	Usuario	
Breve Descripción	El usuario consulta el BOX de un centro médico.	
Precondiciones	La pantalla mostrada es la correspondiente a 'Estadísticas', 'Seguimiento' o 'Admisión'. El usuario debe estar identificado en el sistema.	
Postcondiciones	La pantalla mostrada es la correspondiente a 'BOX'.	
Flujo de Eventos	Usuario	Sistema
	1	Pulsa en la opción del menú: 'BOX'.
	2	Muestra la información correspondiente al BOX del centro médico.

*Consultar las estadísticas de un centro médico:*

Caso de Uso	CONSULTAR LAS ESTADÍSTICAS DE UN CENTRO MÉDICO	
Actor	Usuario	
Breve Descripción	El usuario consulta el BOX de un centro médico.	
Precondiciones	La pantalla mostrada es la correspondiente a 'Admisión', 'Seguimiento' o 'BOX'. El usuario debe estar identificado en el sistema .	
Postcondiciones	La pantalla mostrada es la correspondiente a 'Estadísticas'.	
Flujo de Eventos	Usuario	Sistema
	1 Pulsa en la opción del menú: 'Estadísticas'.	
	2	Muestra la información correspondiente a las Estadísticas del centro médico.

*Salir del sistema:*

Caso de Uso	SALIR DEL SISTEMA	
Actor	Usuario	
Breve Descripción	El usuario cierra sesión en el sistema.	
Precondiciones	La pantalla mostrada es la correspondiente a 'Admisión', 'Seguimiento', 'BOX' o 'Estadísticas'. El usuario debe estar identificado en el sistema.	
Postcondiciones	La pantalla mostrada es la correspondiente a 'LogIn'.	
Flujo de Eventos	Usuario	Sistema
	1 Pulsa el icono 'Salida'.	
	2	Cierra la sesión del usuario y muestra la pantalla 'LogIn'.

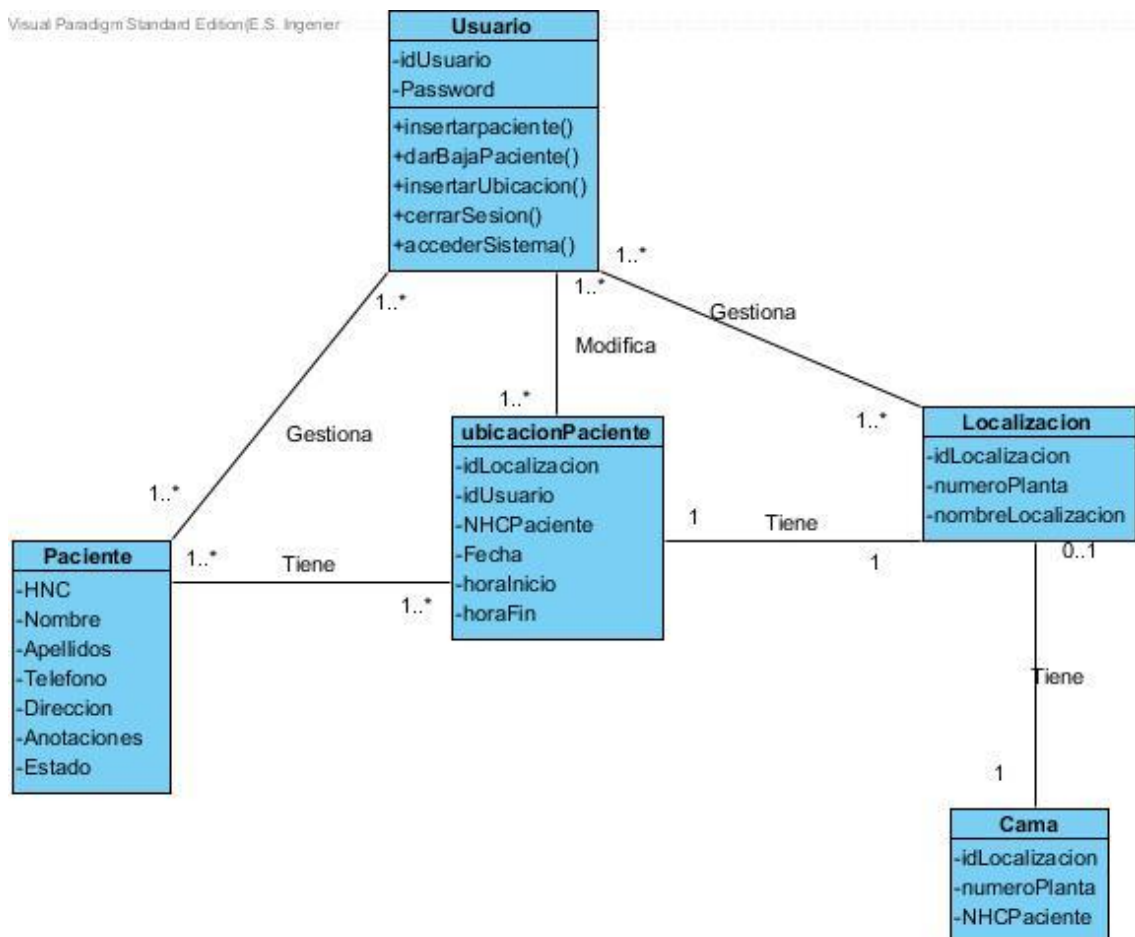
### 3.2. Modelo de dominio

El modelo de dominio representa las clases conceptuales que se consideran fundamentales en el dominio del problema.

Las clases expuestas en el diagrama no tienen por qué coincidir con las clases implementadas finalmente, ya que esto depende del entorno de desarrollo. El Modelo de Dominio de la solución se puede encontrar en el Apartado 5.1.

En el diagrama UML que se muestra a continuación podemos observar:

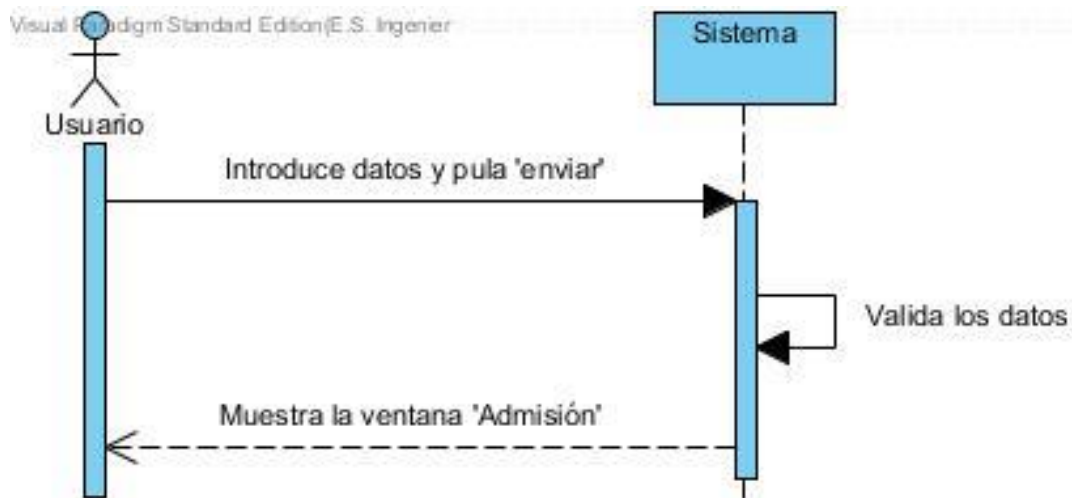
- Los objetos o clases conceptuales del dominio.
- Las relaciones que existen entre estas clases conceptuales.
- Los atributos más importantes de estas clases conceptuales.



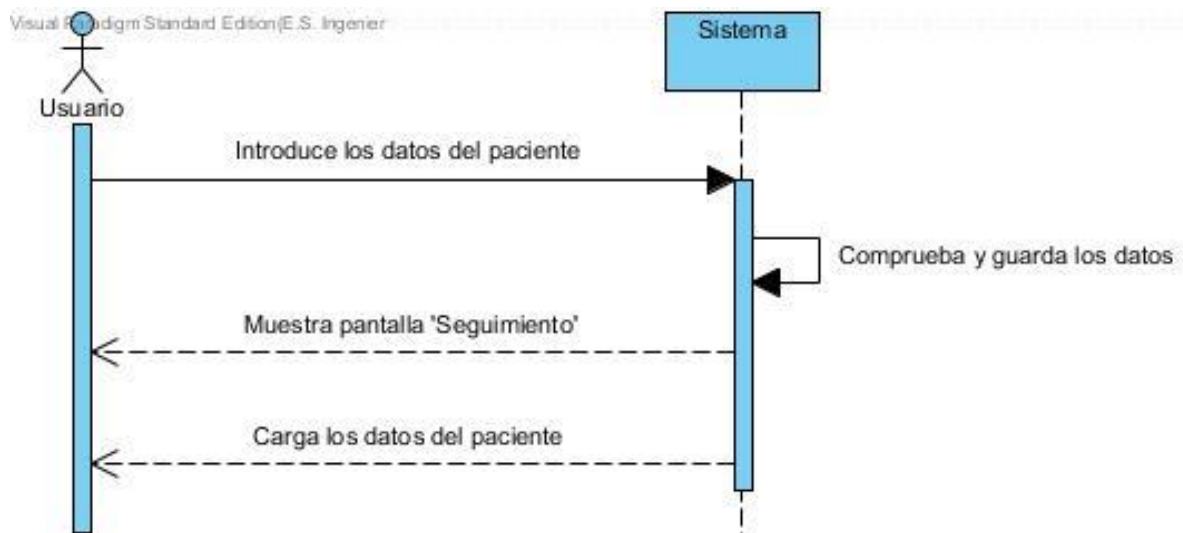
### 3.3. Diagramas de secuencia

En este apartado se detalla el comportamiento del sistema de una manera general según las acciones del usuario, pretendiendo aplicar la máxima intuitividad posible.

#### 3.3.1. Acceder al sistema:

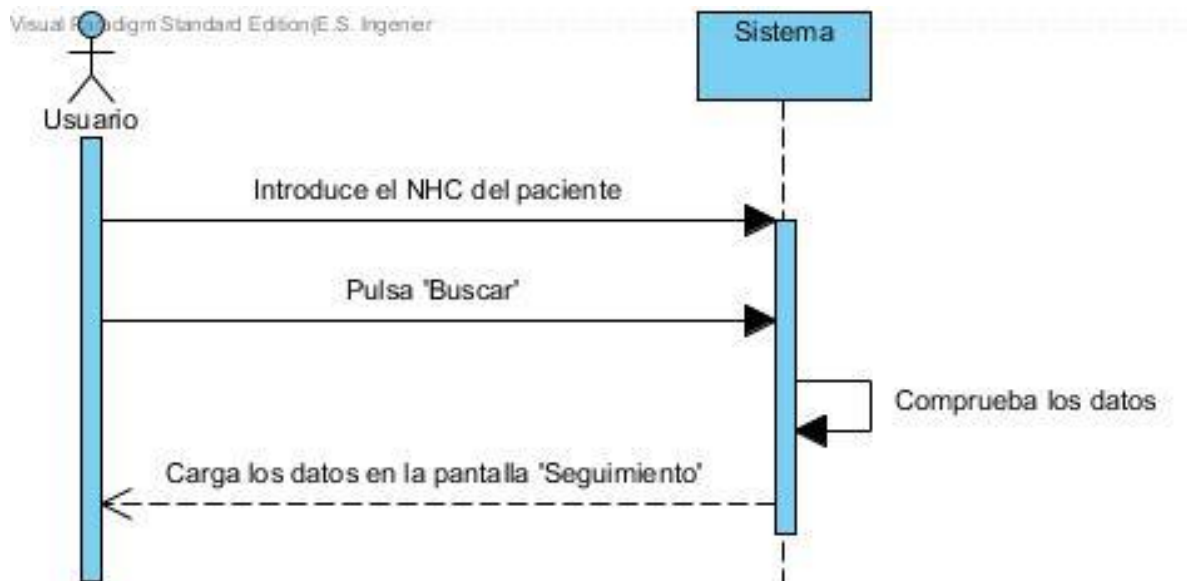


#### 3.3.2. Dar de alta un paciente:

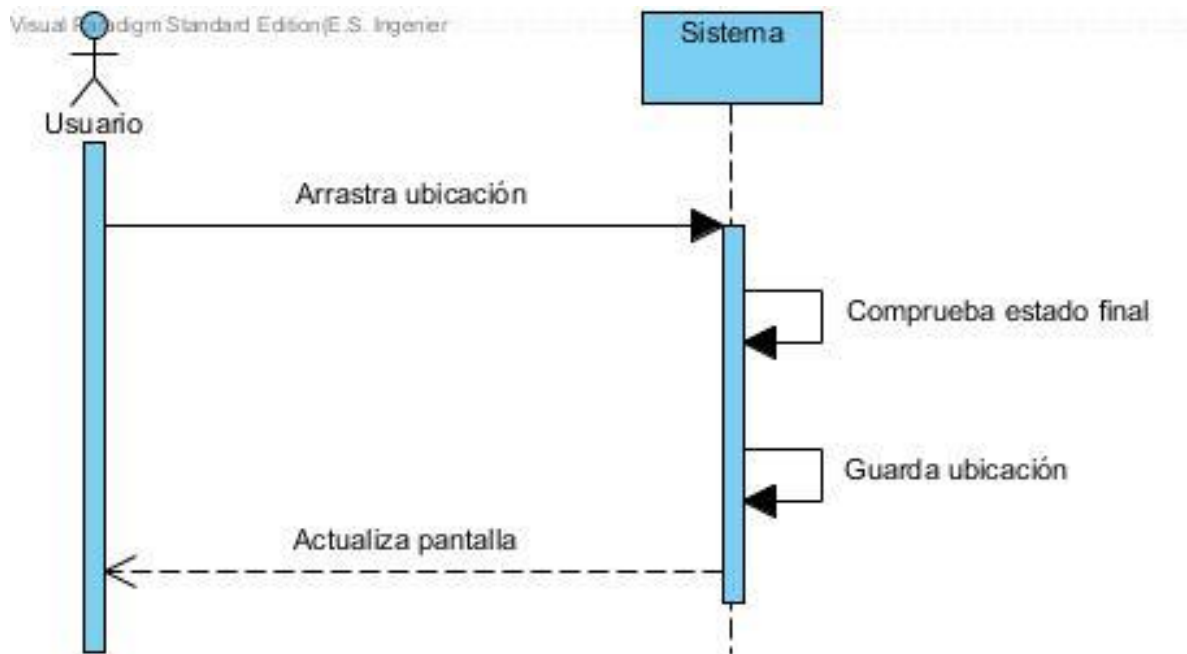




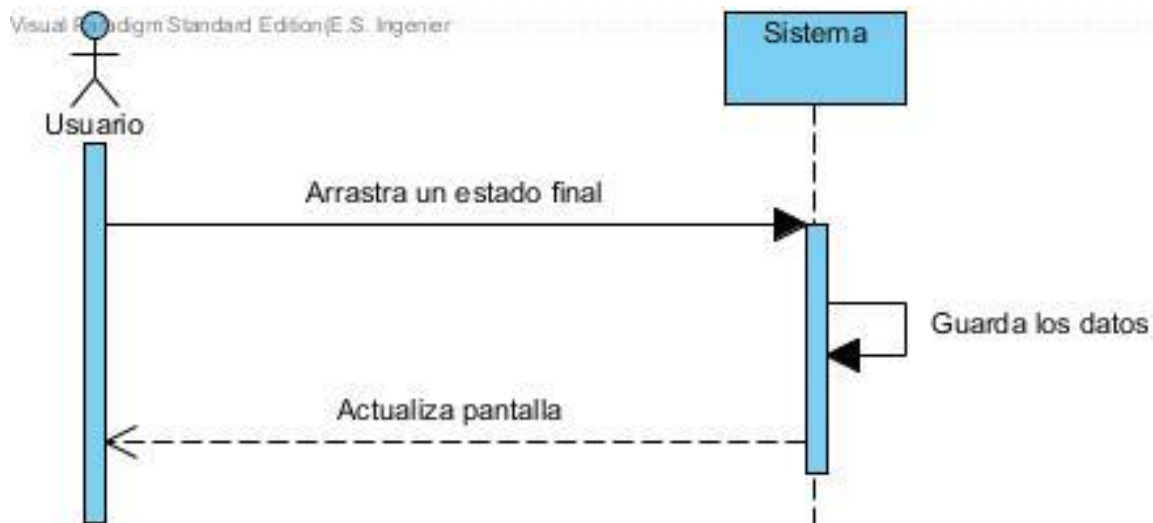
### 3.3.3. Buscar un paciente:



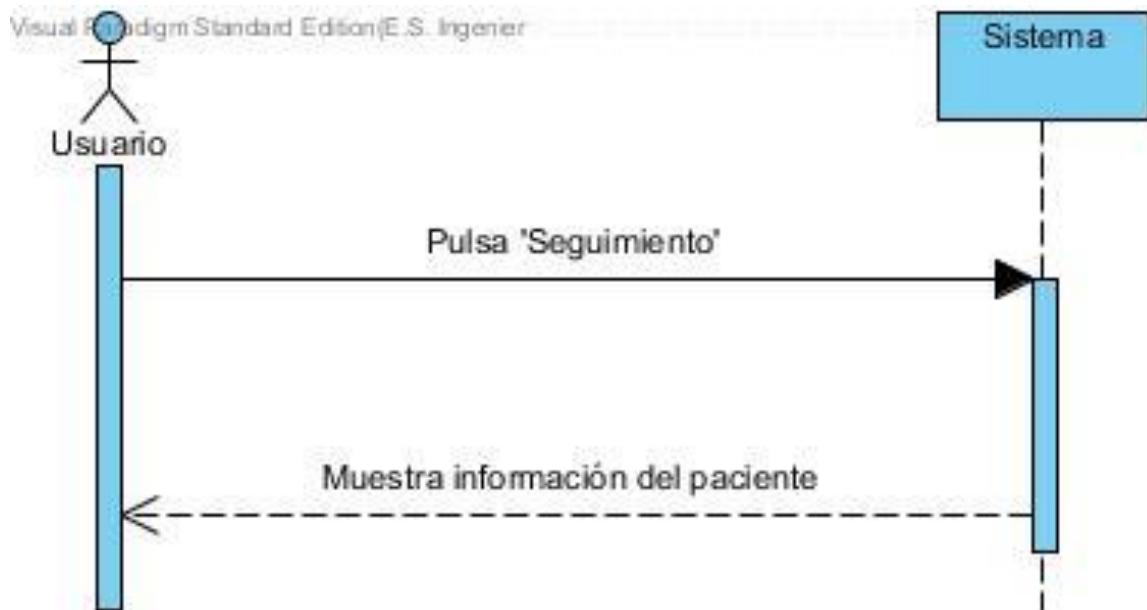
### 3.3.4. Modificar ubicación de un paciente:



### 3.3.5. Dar de baja un paciente:



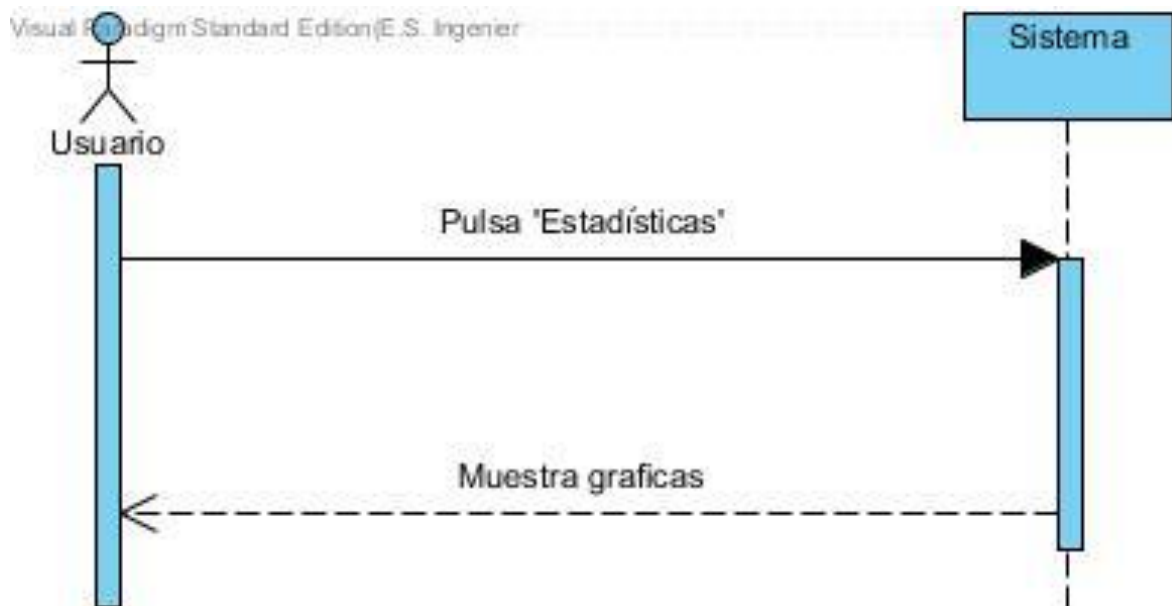
### 3.3.6. Consultar el seguimiento de un paciente:



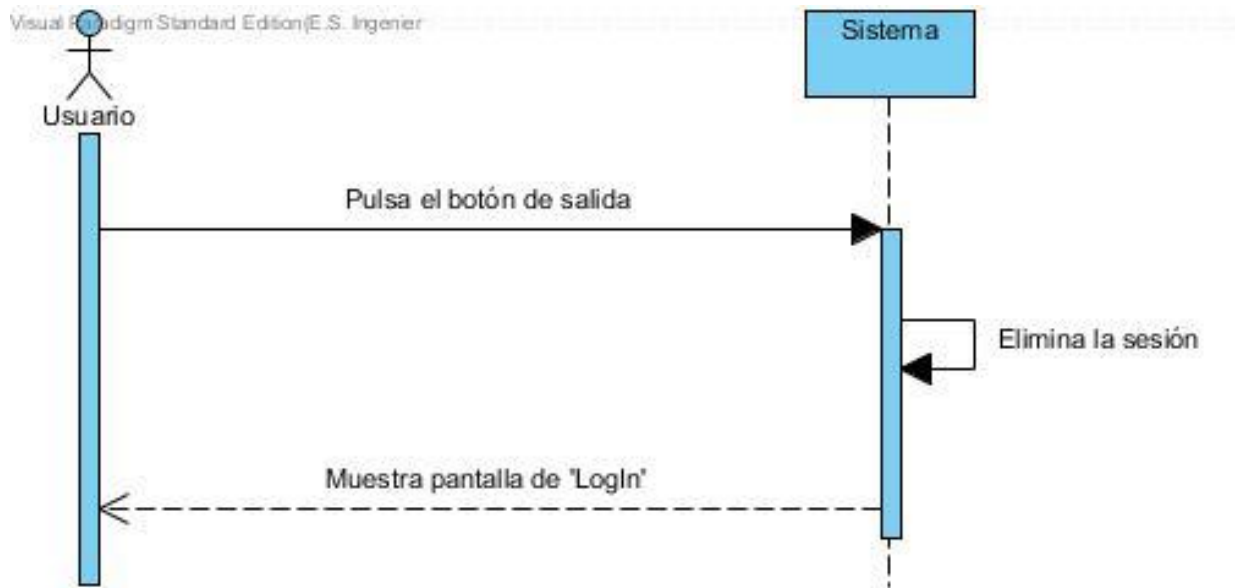
### 3.3.7. Consultar el BOX del centro médico:



### 3.3.8. Consultar las estadísticas de un centro médico:



### 3.3.9. Salir del sistema:

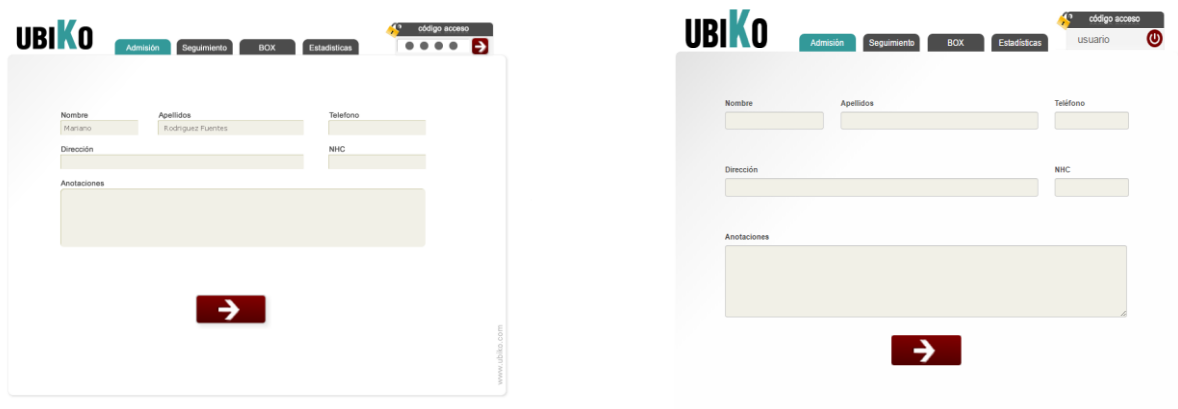


## 3.4. Interfaz de usuario

En este apartado se presentan las imágenes iniciales presentadas para la maquetación de la aplicación enfrentadas con el resultado final. El cambio más significativo es el apartado de 'Estadísticas' en el cual se ha optado por la implementación de gráficos.

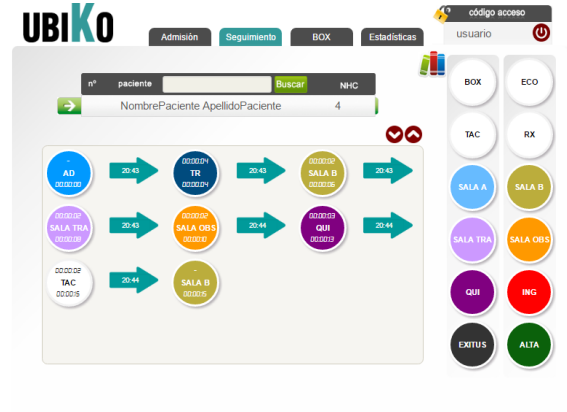
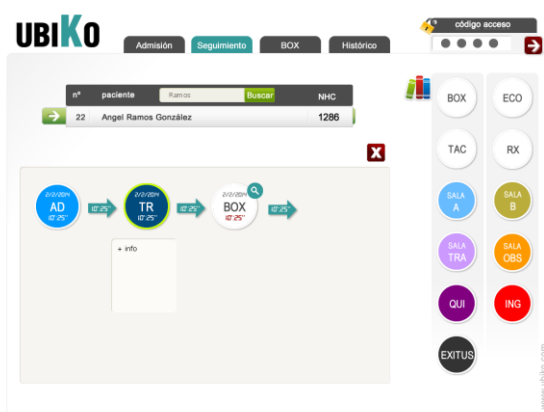
### 3.4.1. Pantalla 'Admisión':

Pantalla propuesta (izquierda) vs pantalla final (derecha):



### 3.4.2. Pantalla 'Seguimiento':

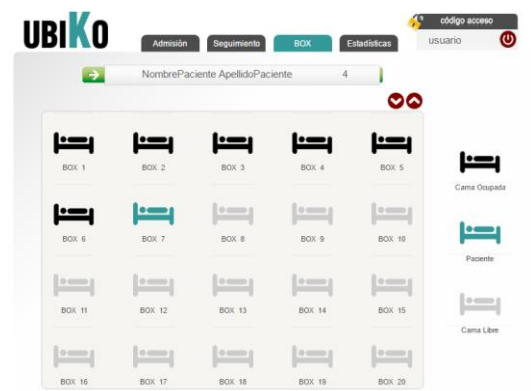
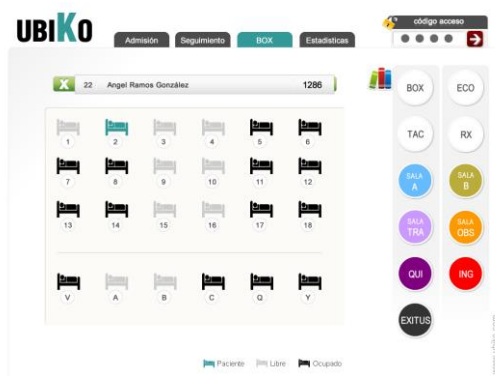
Pantalla propuesta (izquierda) vs pantalla final (derecha):



En este caso no se contemplan demasiadas diferencias, las más destacables: nueva ubicación en el menú de la pantalla final, así como la implementación de un 'carrusel' y la eliminación del apartado de selección de 'BOX'.

### 3.4.3. Pestaña 'BOX':

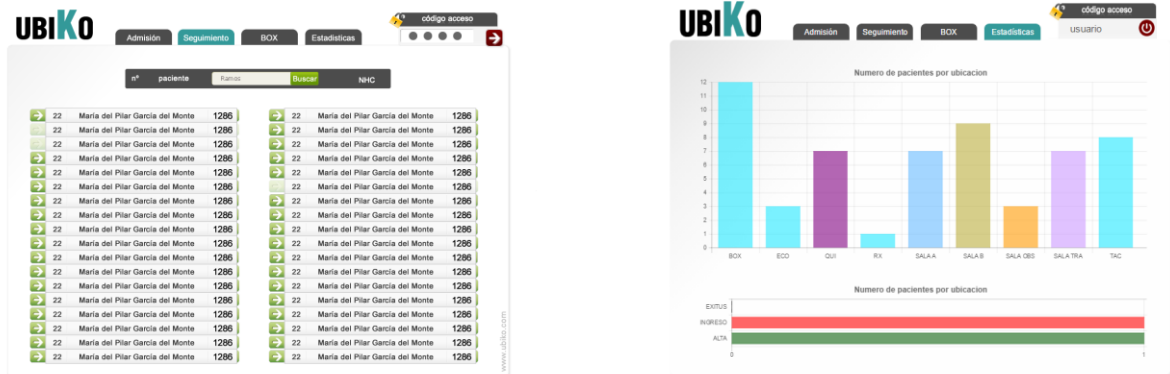
Pantalla propuesta (izquierda) vs pantalla final (derecha):



Diferencias: se suprime el menú propuesto por una leyenda, a mayores, se implementa un 'carrusel' para poder visualizar todas las camas del sistema.

### 3.4.4. Pantalla 'Estadísticas':

Pantalla propuesta (izquierda) vs pantalla final (derecha):



En la pantalla final se ha optado por un diseño basado en la visualización de gráficos para una mejor percepción de los datos.

### 3.4.5. Ventana 'LogIn':

Se ha introducido la ventana de 'LogIn' a mayores de las propuestas para permitir el acceso al sistema.

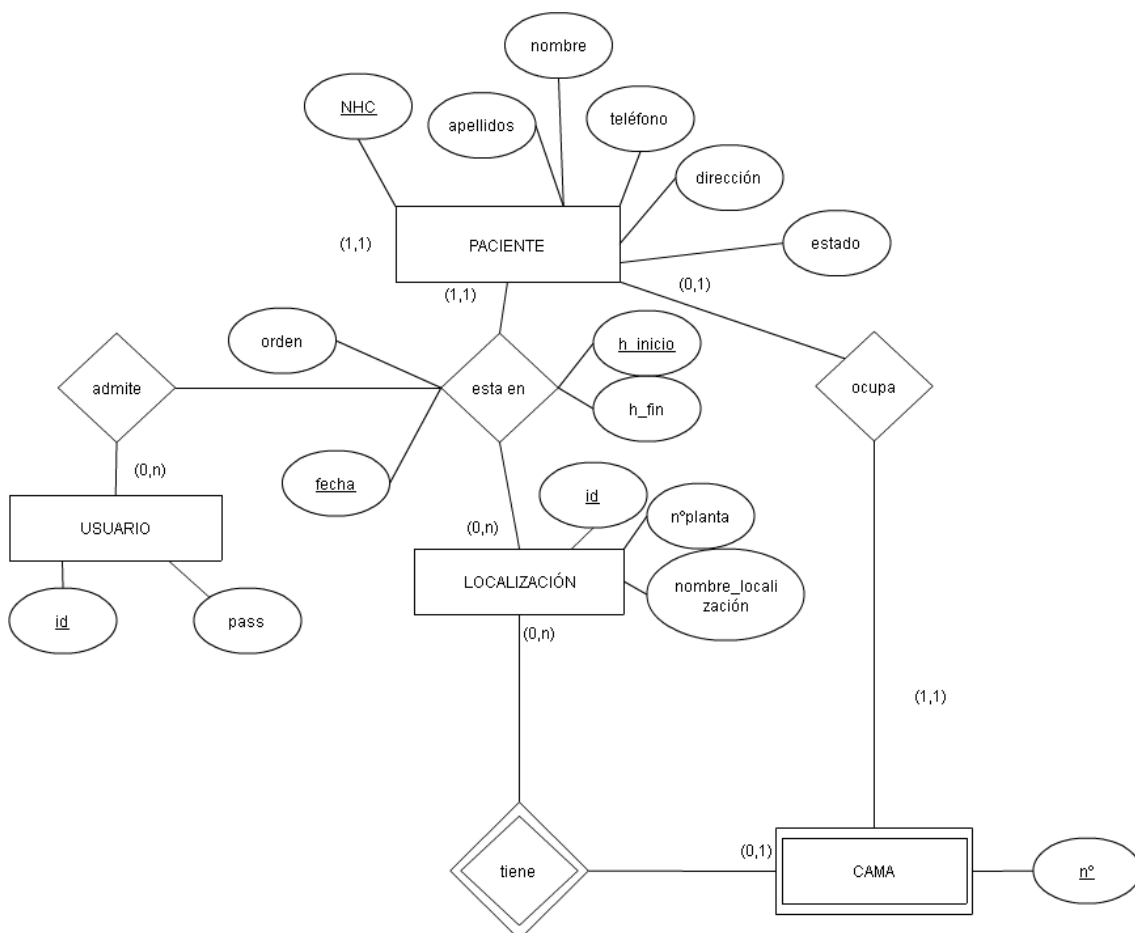
The login window (LogIn) for UBIKO features the UBIKO logo at the top. Below the logo are two input fields: 'Usuario' (User) and 'Contraseña' (Password). The 'Contraseña' field is masked with dots. Below the password field is a red button with a white right-pointing arrow.

## 4. MODELO DE DATOS

### 4.1. Modelo entidad-relación

Para el diseño del modelo de persistencia de la aplicación se utiliza el modelo entidad-relación extendido. Este modelo representa a nivel conceptual las entidades de un sistema de información, así como sus interrelaciones y propiedades.

A continuación se muestra el diagrama Entidad-Relación de la persistencia de la aplicación.



Como se puede observar, entre las clases 'Usuario', 'Paciente' y 'Localización' se genera una relación ternaria, la cual será fundamental para determinar un histórico (de ser necesario) que permita almacenar qué usuario gestionó qué paciente.

En cuanto a la clase 'Cama', surge de una dependencia de la clase 'Localización', puesto que únicamente una localización concreta (Box) posee camas. A la hora del paso a tablas, tanto la relación ternaria como la entidad débil generan una tabla independiente.

## 4.2. Paso a tablas

En esta sección pasaremos el diagrama anterior del modelo lógico a modelo relacional.

### 4.2.1. Tabla Usuario

Nombre	USUARIO		
Clave Primaria	idUsuario		
Clave Foránea			
Descripción	Tabla que contiene los datos referentes a los usuarios que tendrán acceso al sistema.		
Campo	Tipo	Not Null	Descripción
idUsuario	varchar(15)	X	Id del Usuario
Password	varchar(25)	X	Contraseña del usuario

### 4.2.2. Tabla Cama

Nombre	CAMA		
Clave Primaria	Localizacion_idLocalizacion, numeroCama		
Clave Foránea	Paciente_NHCPaciente, Localizacion_idLocalizacion		
Descripción	Tabla que contiene los datos referentes a las camas registradas en el sistema.		
Campo	Tipo	Not Null	Descripción
Localizacion_idLocalizacion	varchar(10)	X	Id de la localización.
NumeroPlanta	Int	X	Número identificativo de la cama.
Paciente_NHCPaciente	Varchar(5)		Identificador del paciente.



#### 4.2.3. Tabla ubicación Paciente

Nombre	UBICACIONPACIENTE		
Clave Primaria	Localizacion_idLocalizacion, usuario_idUsuario, fecha, horalnicio, Paciente_NHC		
Clave Foránea	Paciente_NHC, Localizacion_idLocalizacion, usuario_idUsuario		
Descripción	Tabla que contiene los datos referentes a las ubicaciones en las que se encuentra cada usuario.		
Campo	Tipo	Not Null	Descripción
Localizacion_idLocalizacion	varchar(10)	X	Id de la localización.
Usuario_idUsuario	Varchar(15)	X	Identificador del usuario que introduce la ubicación.
Paciente_NHC	Varchar(5)	X	Identificador del paciente.
Fecha	DATE	X	Fecha en la que se asigna la localización al paciente.
Horalnicio	TIME	X	Hora en la que el paciente ocupa la ubicación.
HoraFin	TIME		Hora en la que el paciente abandona la ubicación.
Orden	Int		Orden que ocupa cada ubicación del paciente.

#### 4.2.4. Tabla Paciente

Nombre	PACIENTE		
Clave Primaria	NHC		
Clave Foránea			
Descripción	Tabla que contiene los datos referentes a los pacientes registrados en el sistema.		
Campo	Tipo	Not Null	Descripción
NHC	varchar(5)	X	Número de Historia Clínica.
Nombre	varchar(25)	X	Nombre del paciente.
Apellidos	Varchar(15)	X	Apellidos del paciente.
Dirección	Varchar(30)		Dirección del paciente
Estado	Varchar(15)		Estado del paciente: Admitido, Alta, Ingresado o Fallecido.
Teléfono	Int		Teléfono del paciente.
Anotaciones	varchar(255)		Anotaciones referentes al paciente.

En un primer momento se diseñó clase paciente de forma que fuese identificada por el campo DNI, posteriormente se cambió por Número de Historia Clínica (NHC) puesto que es un valor más accesible en un centro médico que el campo DNI.

En cuanto al campo Estado, se determinó que un paciente podría tener 4 estados diferentes, uno de ellos 'Admitido' determinaría que el paciente se encuentra en urgencias, mientras que los otros tres (Alta, Ingreso o Fallecimiento) definen una localización clara en caso de ser necesaria.

#### 4.2.5. Tabla Localización

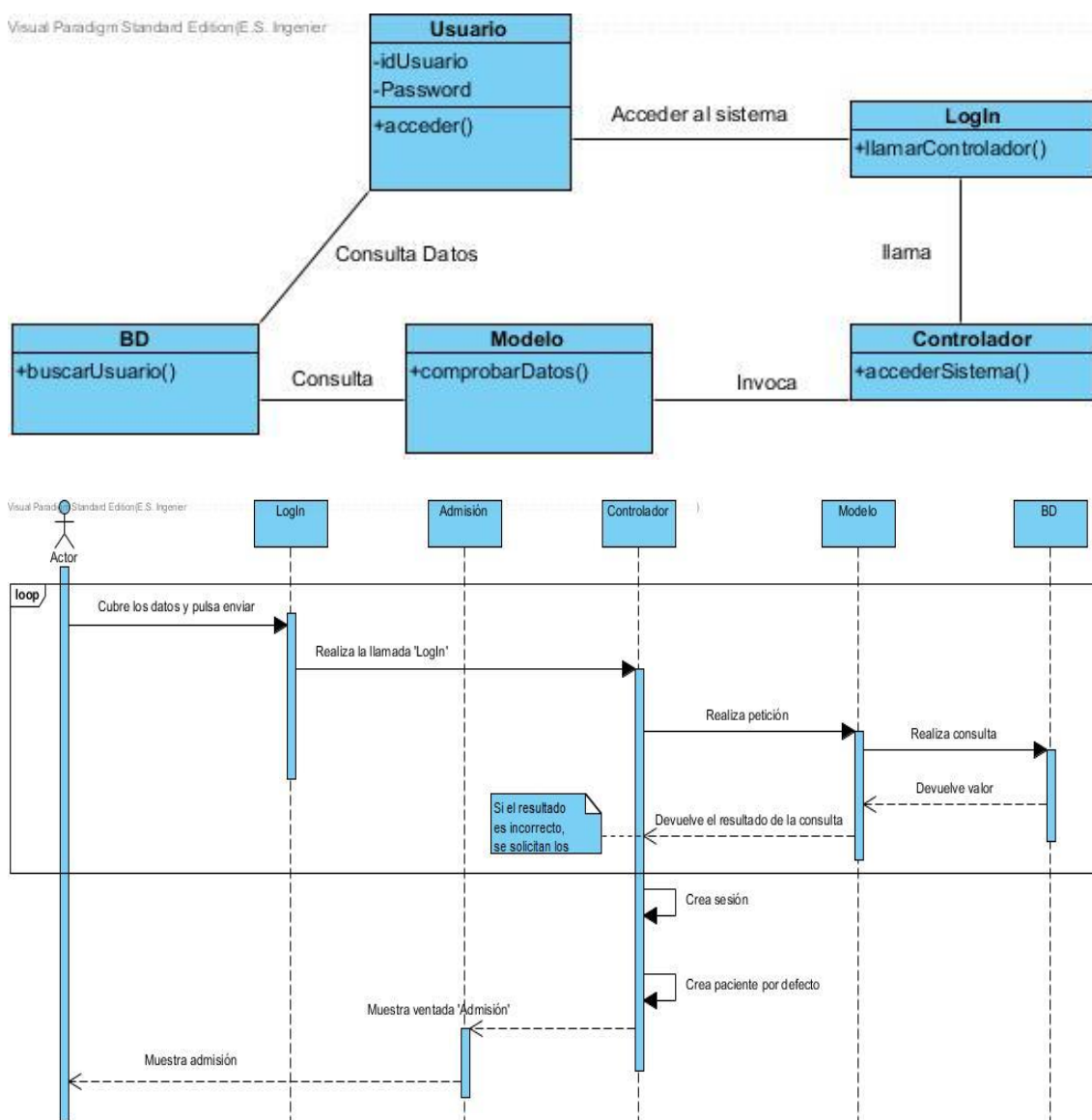
Nombre	LOCALIZACION		
Clave Primaria	idLocalizacion		
Clave Foránea			
Descripción	Tabla que contiene los datos referentes a las ubicaciones registradas en el sistema.		
Campo	Tipo	Not Null	Descripción
idLocalizacion	varchar(10)	X	Id de la localización.
numeroPlanta	Int		Piso en el que se encuentra la localización.
nombreLocalizacion	Varchar(10)		Nombre de la localización.

## 5. DISEÑO

### 5.1. Diagramas de clases parciales y diagramas de secuencia del sistema.

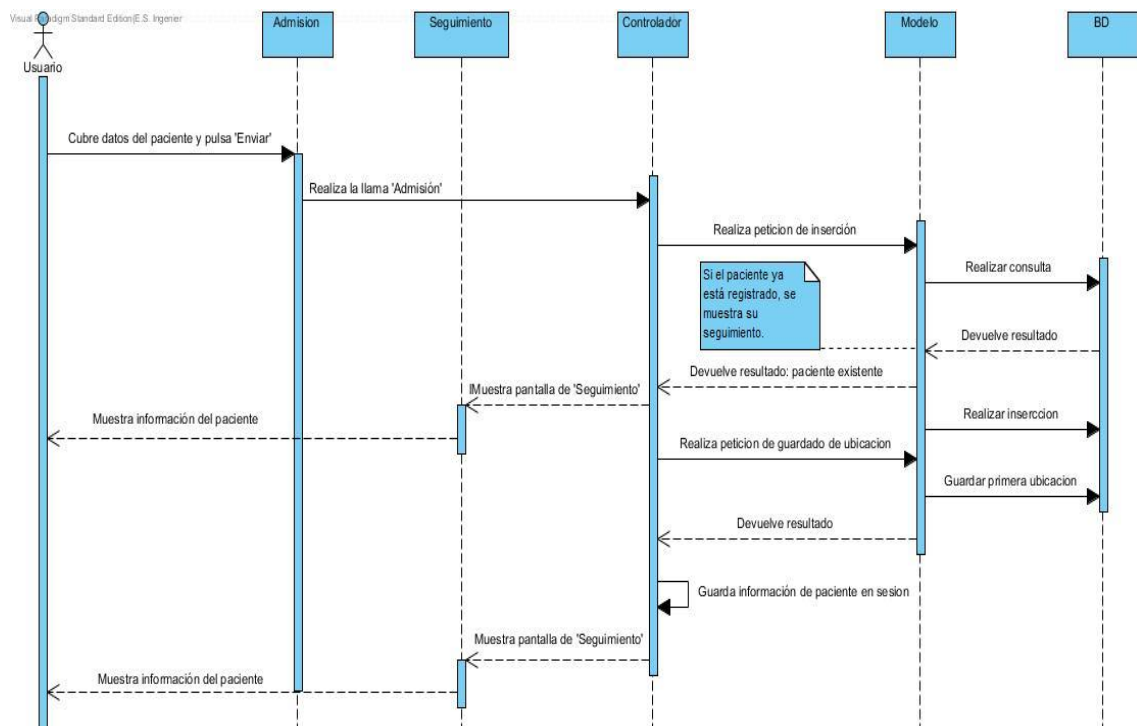
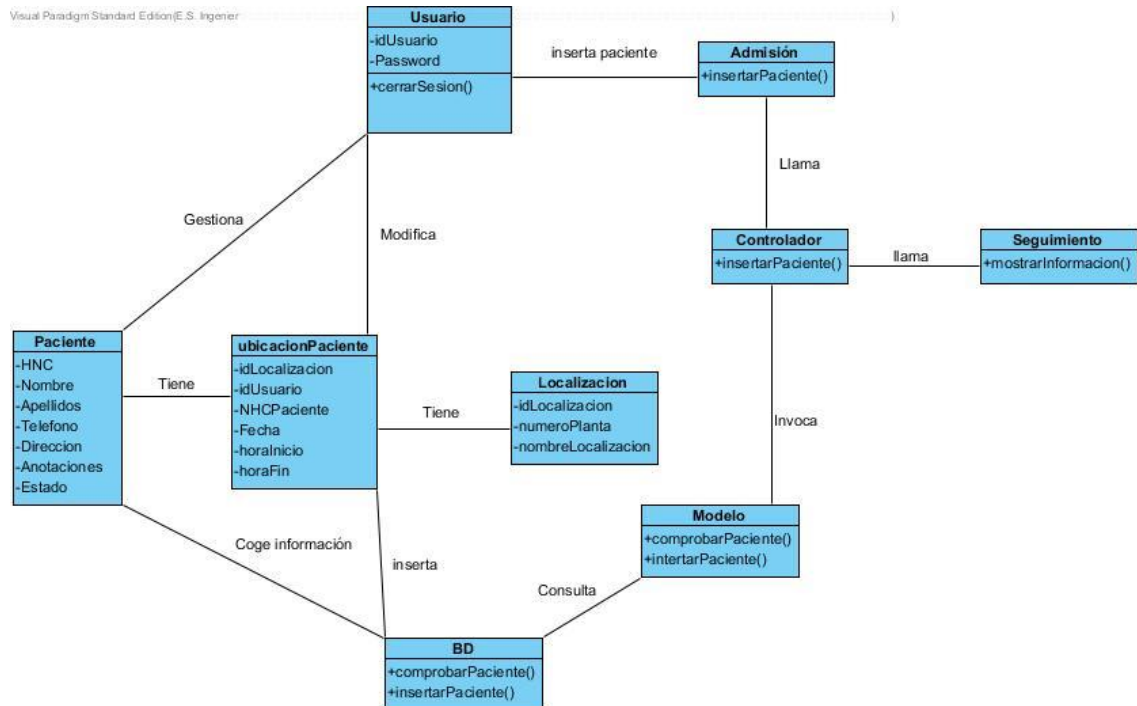
En este apartado se muestran los diferentes diagramas de clases parciales del sistema junto con el correspondiente diagrama de secuencia.

#### 5.1.1. Acceder al sistema



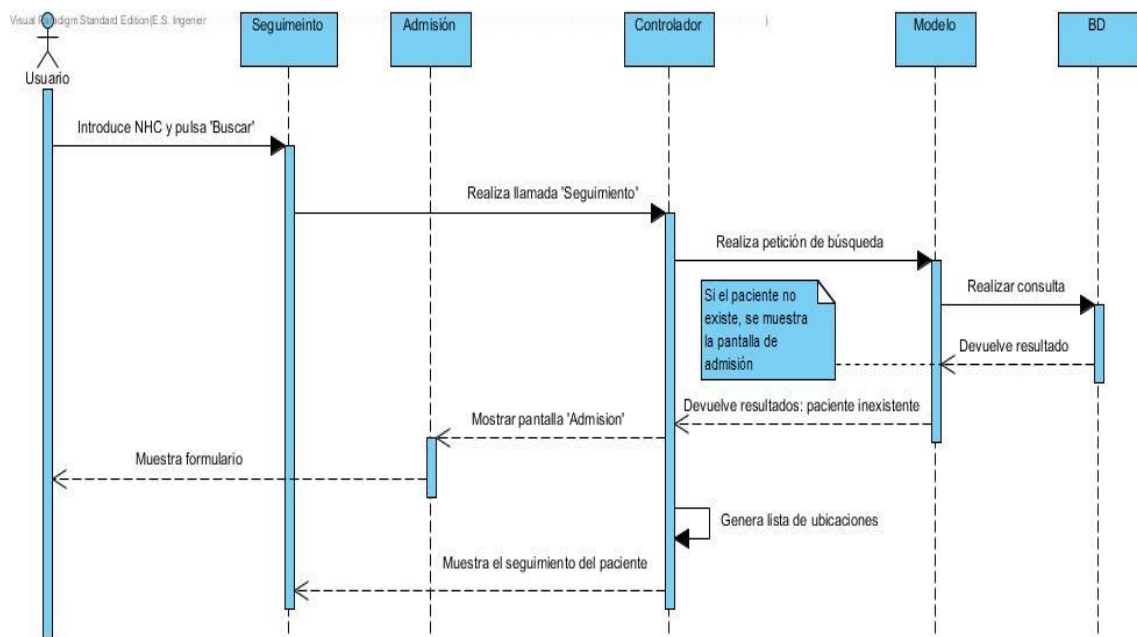
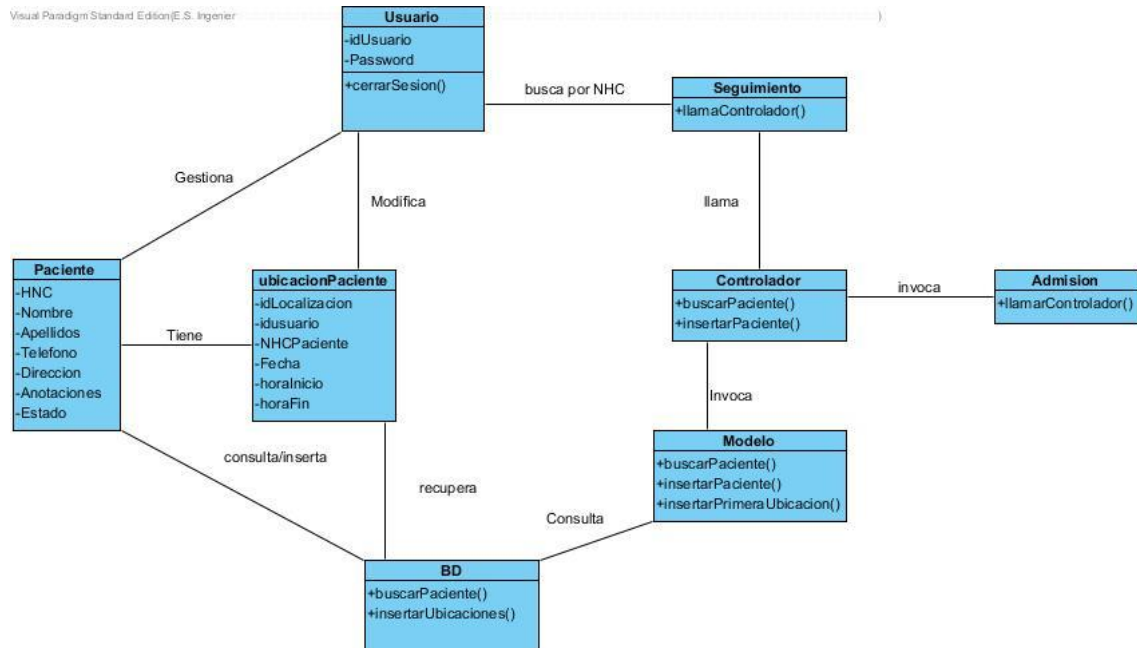
Como se observa en el diagrama, el sistema forma un bucle en caso de que los datos introducidos por el usuario no sean correctos, de forma que no se permitirá el acceso a la ventana de 'Admisión' hasta que el acceso sea permitido.

### 5.1.2. Dar de alta un paciente



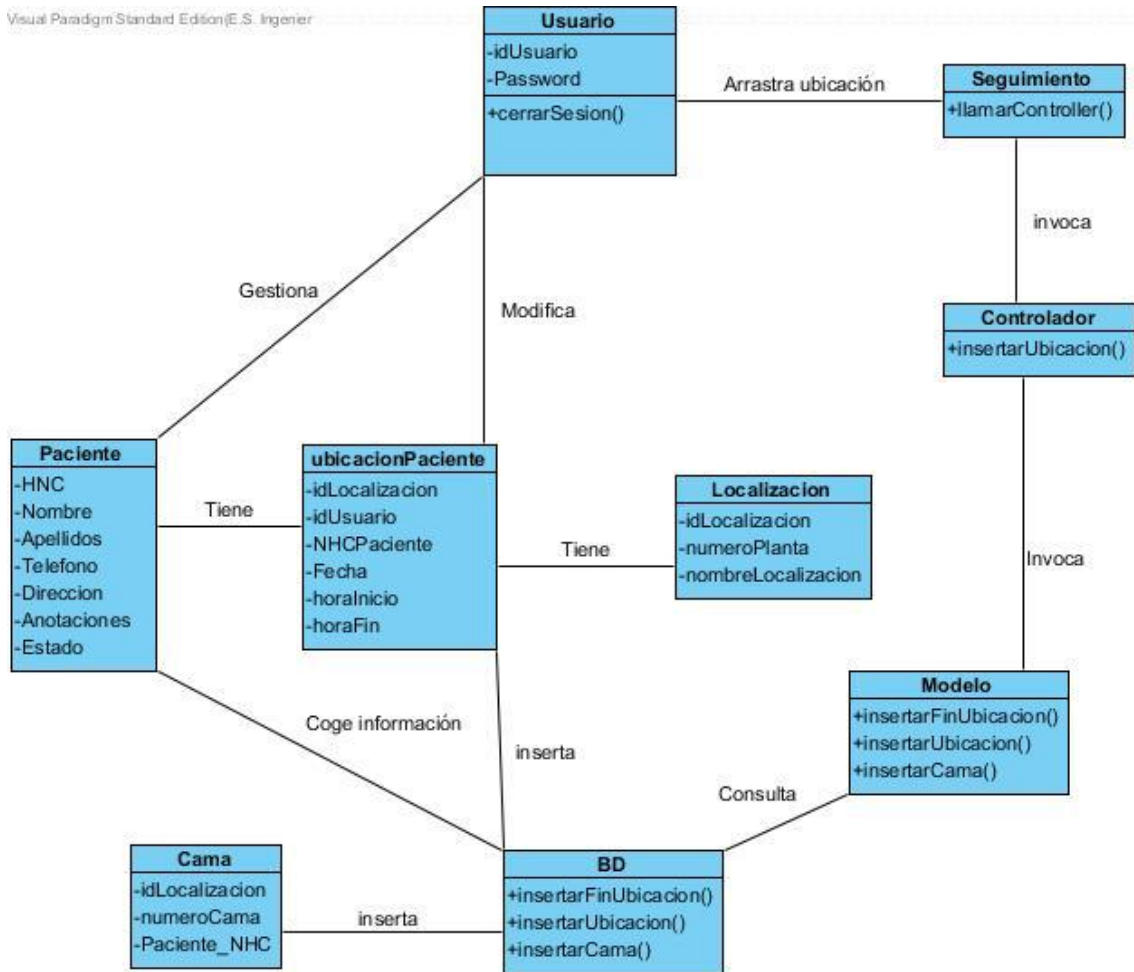
En este caso, si el usuario intenta introducir un paciente ya registrado, el sistema mostrará el seguimiento de dicho paciente. En caso contrario, se inserta el paciente en base de datos junto con sus primeras ubicaciones.

### 5.1.3. Buscar un paciente

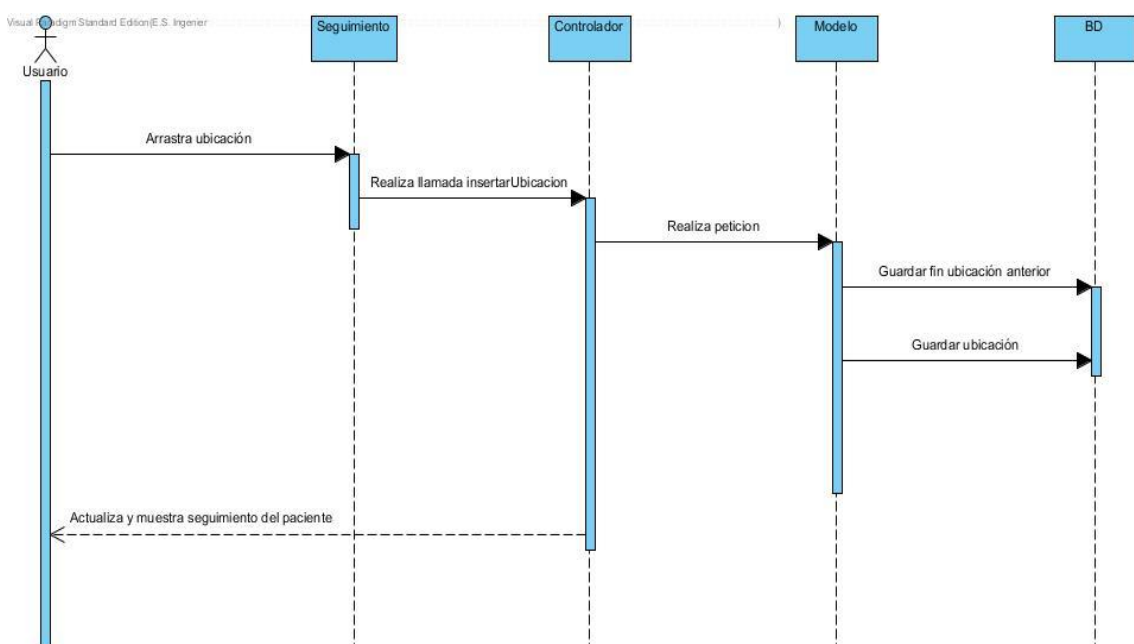


### 5.1.4. Modificar ubicación de un paciente

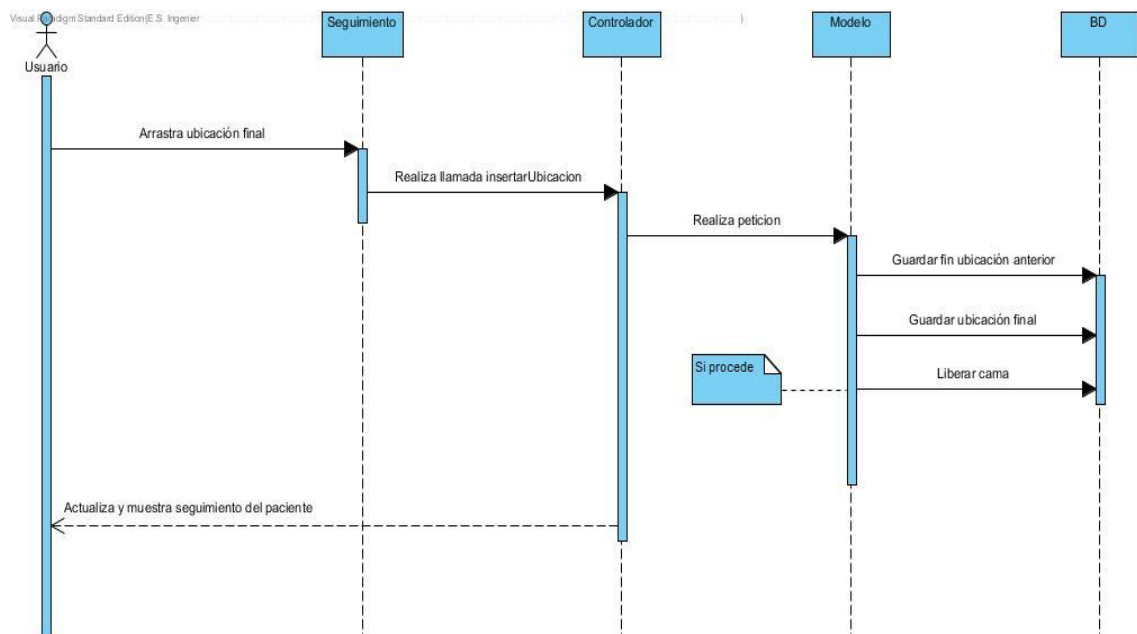
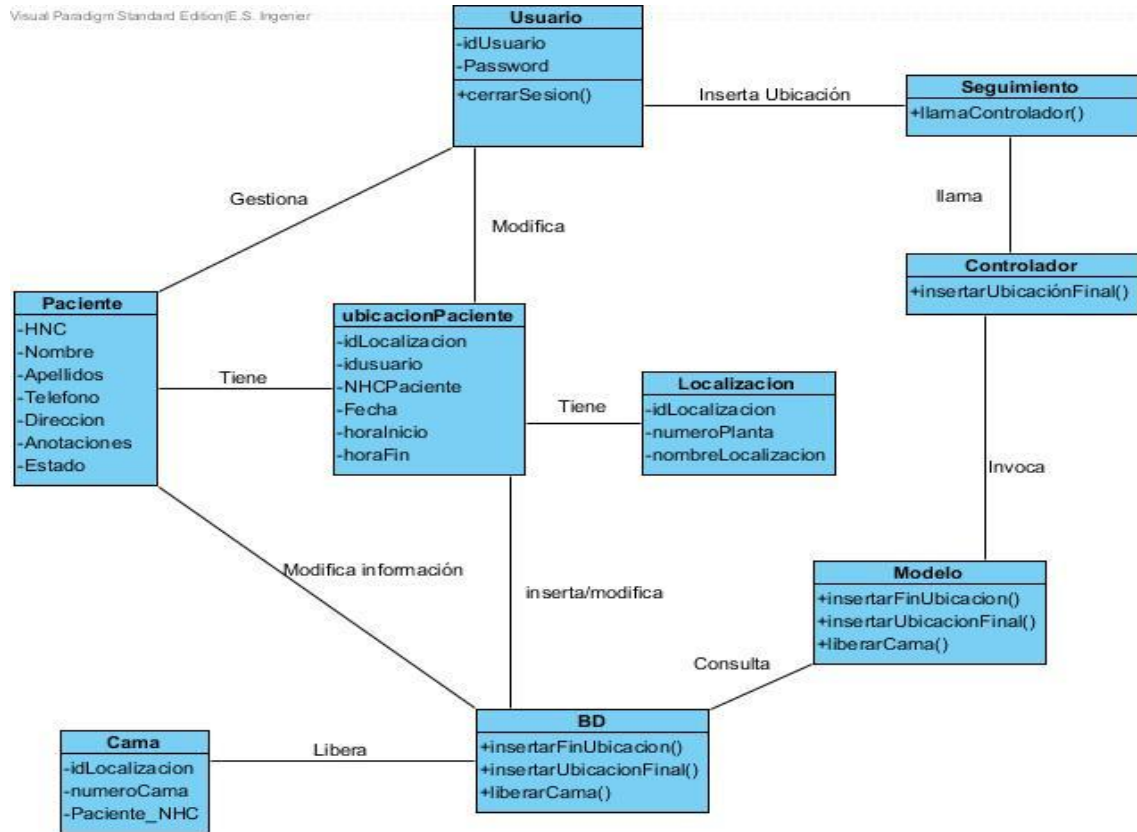
Visual Paradigm Standard Edition (E.S. Ingenier)



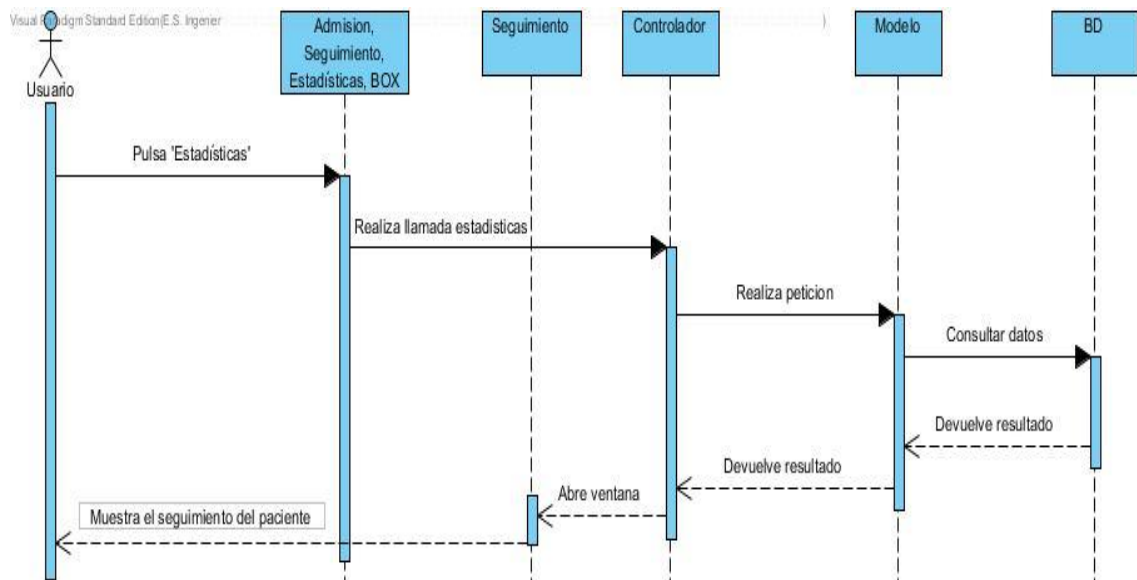
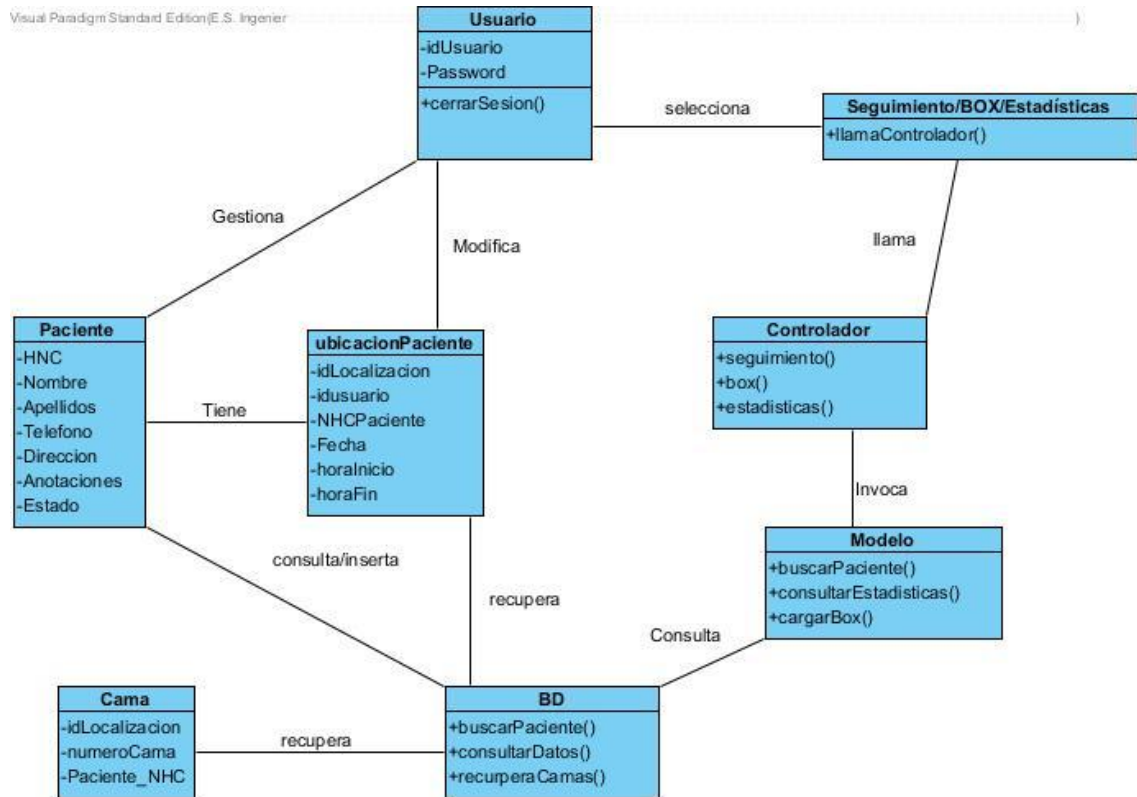
Visual Paradigm Standard Edition (E.S. Ingenier)



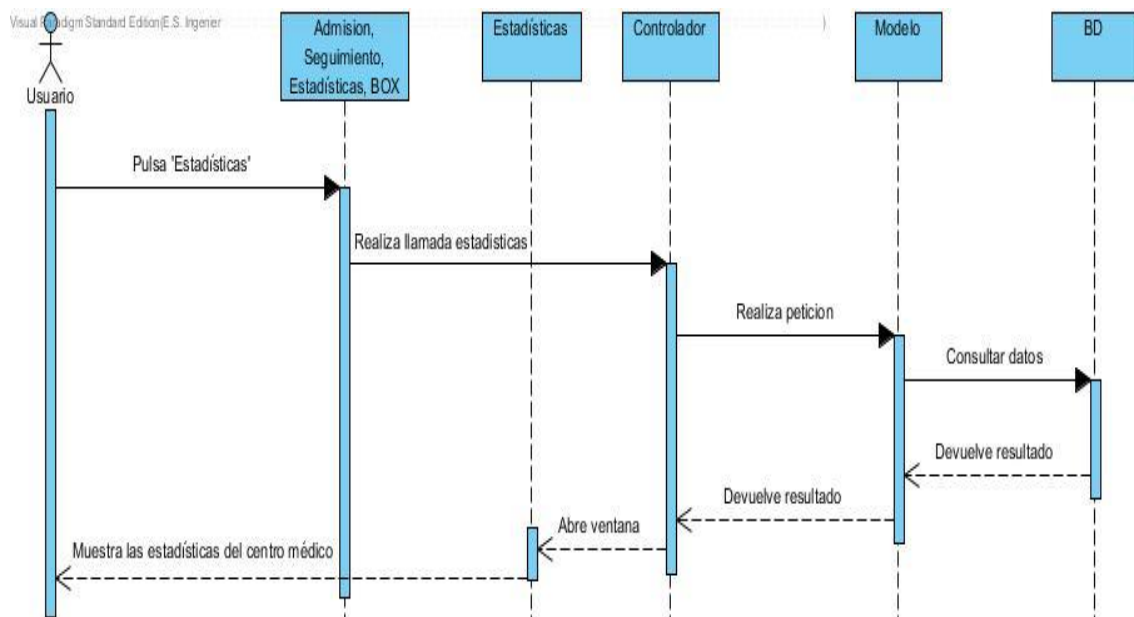
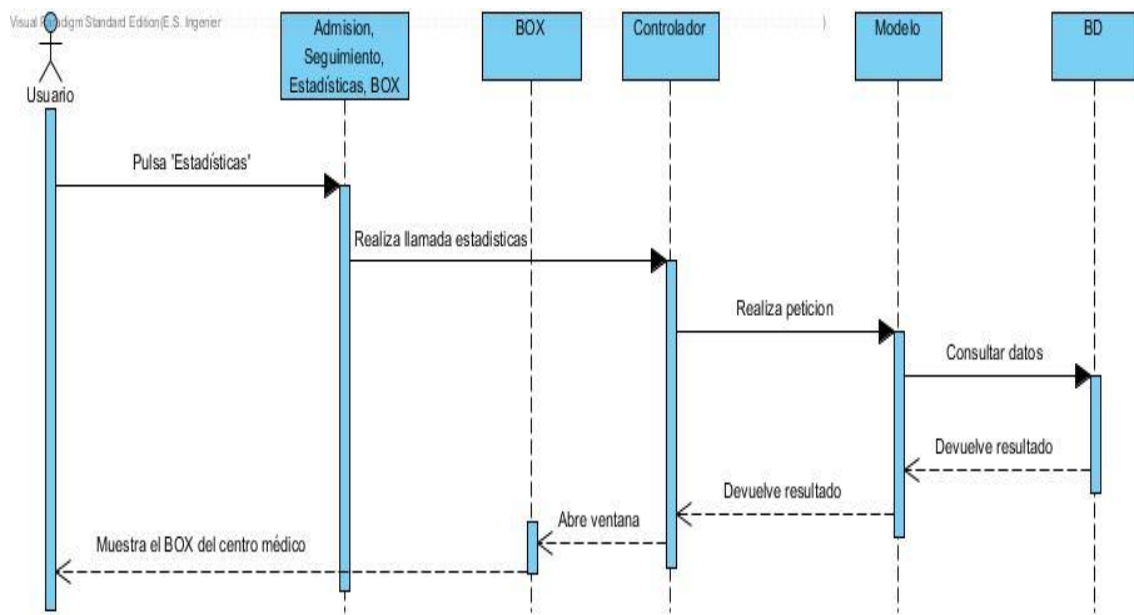
### 5.1.5. Dar de baja un paciente



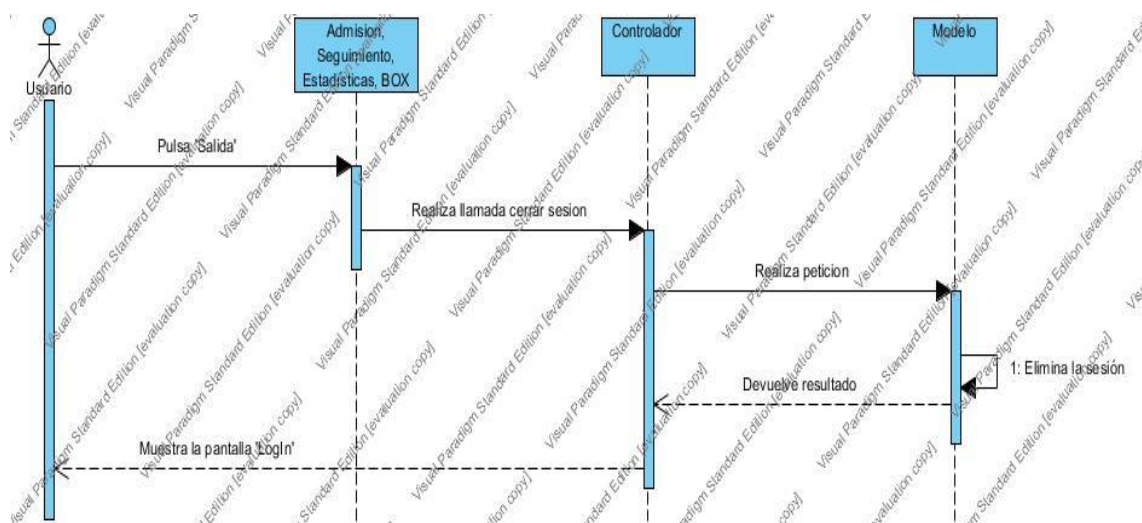
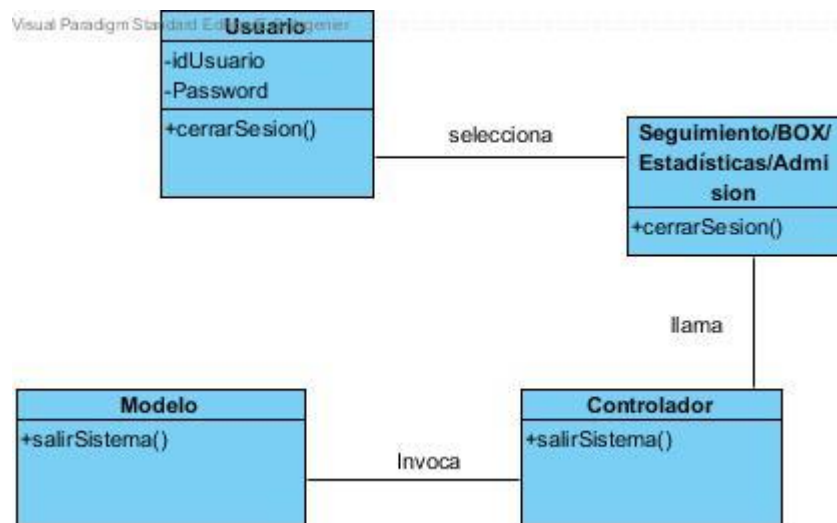
### 5.1.6. Consultar Seguimiento/BOX/Estadísticas







### 5.1.7. Salir del sistema



## 5.2. Definición de diagramas

En esta apartado se pretende definir en mayor profundidad las clases presentadas en los diagramas anteriores:

Clase	Definición
<b>Controlador</b>	Es uno de los componentes de la implementación en PHP Modelo-Vista-Controlador. Todas las vistas invocan al Controlador el cual posee todos los métodos necesarios para llamar al Model pidiendo datos, es decir, todos los accesos a base de datos se tramitan de Controlador a Model, el Controlador nunca accede directamente a ellos. Una vez el Model realiza las acciones, el Controlador decide qué vista mostrar.
Métodos	Definición
login()	Permite el acceso al sistema.
admission()	Registra un paciente en el sistema.
buscarPacienteNHC()	Busca el paciente mediante su NHC.
insertarPaciente()	Registra el paciente en BD.
insertarPrimeraUbicacion()	Inserta las dos primeras ubicaciones del paciente.
logout()	Destruye la sesión.
seguimiento()	Muestra la información de un paciente.
buscarPaciente()	Busca paciente en BD.
recuperarUbicacionesPaciente()	Busca y almacena todas las ubicaciones del paciente buscado.
insertarUbicacion()	Inserta la ubicación seleccionada en BD.
insertarPrimeraUbicacion()	Una vez se crea el paciente, se insertan sus dos primeras ubicaciones.
box()	Recupera las camas almacenadas con su respectiva información.
recuperarCamas()	Recupera las camas de BD.
insertarBox()	Asocia una cama con un paciente en BOX.
comprobarCamaBox()	Comprueba que el paciente no está ya en una cama.
recuperarCamasVacias()	Lista todas las camas libres.

insertarCamaBox()	Asocia una cama vacía con un paciente.
insertarFinUbicacionAnterior()	Introduce una hora de fin a la ubicación anterior.
insertarUbicacion()	Asocia una localización con un paciente.
recuperarUbicaciones()	Lista todas las localizaciones de un paciente.
insertarEco()	Asocia la localización ECO con el paciente.
insertarRx()	Asocia la localización RX con el paciente.
insertarTac()	Asocia la localización TAC con el paciente.
insertarTR()	Asocia la localización TR con el paciente.
insertarSalaA()	Asocia la localización SALA A con el paciente.
insertarSalaB()	Asocia la localización SALA B con el paciente.
insertarSalaObs()	Asocia la localización SALA OBS con el paciente.
insertarSalaTra()	Asocia la localización SALA TRA con el paciente.
insertarQui()	Asocia la localización QUI con el paciente.
insertarIng()	Asocia la localización ING con el paciente.
insertarAlta()	Asocia la localización ALTA con el paciente.
insertarExitus()	Asocia la localización EXITUS con el paciente.
insertarEstadoFinal()	Modifica el estado del paciente.
liberarCama()	Elimina la asociación entre paciente y cama.
restaHoras()	Operación para la resta de tiempo.
sumaHoras()	Operación para la suma de tiempo.
estadísticas()	Lista la información necesaria para crear los gráficos de estadísticas.

Clase	Definición
<b>Modelo</b>	Es uno de los componentes de la implementación en PHP Modelo-Vista-Controlador. Es invocado por el Controlador cuando este necesita de alguna consulta a base de datos.
<b>Métodos</b>	<b>Definición</b>
login()	Permite el acceso al sistema.
insertarPaciente()	Registra un paciente en el sistema.
buscarPacienteNHC()	Busca el paciente mediante su NHC.
insertarPaciente()	Registra el paciente en BD.
insertarPrimeraUbicacion()	Inserta las dos primeras ubicaciones del paciente.
buscarPaciente()	Busca paciente en BD.
recuperarUbicacionesPaciente()	Busca y almacena todas las ubicaciones del paciente buscado.
insertarUbicacion()	Inserta la ubicación seleccionada en BD.
insertarPrimeraUbicacion()	Una vez se crea el paciente, se insertan sus dos primeras ubicaciones.
recuperarCamas()	Recupera las camas de BD.
insertarBox()	Asocia una cama con un paciente en BOX.
comprobarCamaBox()	Comprueba que el paciente no está ya en una cama.
recuperarCamasVacias()	Lista todas las camas libres.
insertarCamaBox()	Asocia una cama vacía con un paciente.
insertarFinUbicacionAnterior()	Introduce una hora de fin a la ubicación anterior.
recuperarUbicaciones()	Lista todas las localizaciones de un paciente.

Clase	Definición
<b>Admisión</b>	Es la vista que permite admitir en el sistema un paciente.
Métodos	Definición
admission()	Llama al controlador para que inserte un paciente en el sistema
logOut()	Llama al controlador para que destruya la sesión.
seguimiento()	Método que llama al controlador para que recargue la información.
logOut()	Destruye la sesión.

Clase	Definición
<b>Estadísticas</b>	Vista que permite la visualización de los gráficos del centro médico.
Métodos	Definición
chart()	Método javascript que permite la creación de gráficos.
logOut()	Llama al controlador para que destruya la sesión.

Clase	Definición
<b>BOX</b>	Vista que permite visualizar las camas del centro médico, diferenciando aquellas que están ocupadas de las que no.

Clase	Definición
<b>ubicacionPaciente</b>	Clase que almacena la información referente a las ubicaciones que tiene un paciente durante su estancia en Urgencias.
Métodos	Definición
insertarPrimeraUbicacion()	Inserta las dos primeras ubicaciones del paciente.
recuperarUbicacionesPaciente()	Busca y almacena todas las ubicaciones del paciente buscado.
insertarUbicacion()	Inserta la ubicación seleccionada en BD.
insertarFinUbicacionAnterior()	Introduce una hora de fin a la ubicación anterior.

Clase	Definición
<b>Usuario</b>	Clase que almacena la información referente a un usuario.
Métodos	Definición
login()	Permite el acceso al sistema.
logout()	Llama al controlador para que destruya la sesión.

Clase	Definición
<b>Localización</b>	Clase que almacena las diferentes localizaciones posibles dentro de un hospital.
Métodos	Definición

Clase	Definición
<b>Paciente</b>	Clase que almacena toda la información personal referente al paciente.
Métodos	Definición

Clase	Definición
<b>BD</b>	Clase encargada en realizar todas las consultas a base de datos.
Métodos	Definición
login()	Permite el acceso al sistema.
insertarPaciente()	Registra un paciente en el sistema.
buscarPacienteNHC()	Busca el paciente mediante su NHC.
insertarPaciente()	Registra el paciente en BD.
insertarPrimeraUbicacion()	Inserta las dos primeras ubicaciones del paciente.
buscarPaciente()	Busca paciente en BD.
recuperarUbicacionesPaciente()	Busca y almacena todas las ubicaciones del paciente buscado.
insertarUbicacion()	Inserta la ubicación seleccionada en BD.
insertarPrimeraUbicacion()	Una vez se crea el paciente, se insertan sus dos primeras ubicaciones.
recuperarCamas()	Recupera las camas de BD.
insertarBox()	Asocia una cama con un paciente en BOX.
comprobarCamaBox()	Comprueba que el paciente no está ya en una cama.
recuperarCamasVacias()	Lista todas las camas libres.
insertarCamaBox()	Asocia una cama vacía con un paciente.
insertarFinUbicacionAnterior()	Introduce una hora de fin a la ubicación anterior.
recuperarUbicaciones()	Lista todas las localizaciones de un paciente.
liberarCama()	Elimina la asociación entre paciente y cama.

Clase	Definición
<b>LogIn</b>	Vista que muestra la pantalla de acceso al sistema.
Métodos	Definición
login()	Permite el acceso al sistema.



Clase	Definición
<b>Cama</b>	Clase que almacena las diferentes camas disponibles en BOX.
Métodos	Definición
recuperarCamas()	Recupera las camas de BD.
recuperarCamasVacías()	Lista todas las camas libres.
liberarCama()	Elimina la asociación entre paciente y cama.

## 6. DETALLES DE IMPLEMENTACIÓN

En este apartado se definen los aspectos más relevantes de la implementación. En primer lugar, es fundamental listar los diferentes lenguajes utilizados, posteriormente se explicará en profundidad los puntos a los que más tiempo se le ha dedicado.

### 6.1.Lenguajes utilizados

- HTML5: lenguaje empleado para la maquetación de la aplicación.
- CSS3: Empleado para dar estilo a la maquetación.
- Javascript: usado fundamentalmente para implementar el arrastre de las diferentes ubicaciones y para su identificación y posterior guardado en base de datos.
- JQuery: lenguaje basado en Ajax y usado para crear los efectos de arrastre y soltado de las ubicaciones en la pantalla. A mayores se implementó un efecto de 'Carousel' para permitir visualizar todas las ubicaciones guardadas así como todas las camas del apartado 'BOX'.
- PHP: usado para la parte del 'backend'. Se optó por la implementación de un modelo-vista-controlador cuyo funcionamiento es el siguiente: La parte frontal está formada por vistas, las cuales comparten elementos comunes recogidos una única vez en una plantilla denominada 'layout'. En el momento en que se desea realizar alguna acción, la vista llama al controlador el cual gestiona la petición y envía la orden al modelo. El modelo es el encargado de interactuar con la base de datos y retornar los valores solicitados al controlador. Este los analiza y decide qué hacer.

- Librería Chart.js: usada para generar los gráficos incrustados en el apartado 'Estadísticas', es una librería javascript que posee múltiples funciones de pintado de gráficos. Su manejo es muy simple, basta con llamar a la función correspondiente al gráfico deseado pasándole los datos y modificando los valores asociados a los colores, escalas, texto etc.

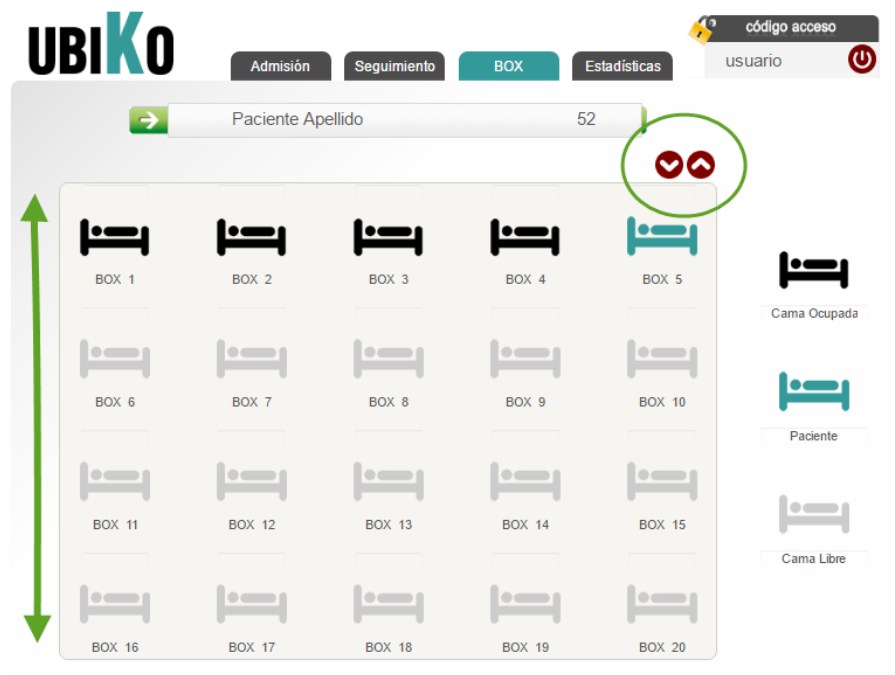
## 6.2.Tareas más relevantes

A continuación se define cómo se implementó algunas de las tareas más tediosas:

- Arrastrado y soltado de las ubicaciones en el contenedor central: JQuery dispone de múltiples funcionalidades muy visuales a la hora de realizar cambios en los elementos mostrados por pantalla, entre ellas podemos destacar la funcionalidad 'Drag and Drop' la cual nos permite arrastrar y un elemento por pantalla y soltarlo en un área concreta. Estas funcionalidades admiten a su vez subfunciones, en este caso empleadas para realizar el efecto de 'clonado' de manera que la ubicación real (la mostrada en el menú lateral) nunca se mueva de su lugar. Como complemento, a las funcionalidades ya descritas se le han añadido mejoras tales como el sombreado de los lugares válidos dónde se podrían soltar las ubicaciones o el retorno de la ubicación a su lugar original en caso de no introducirla en un lugar correcto.



- Implementación del carrusel que permite la visualización de toda la información del contenedor central: su funcionamiento es el mismo tanto para 'Seguimiento' como para 'BOX'. Para su implementación se ha usado una clase JQuery, la cual detecta las pulsaciones de cada botón y substituye la columna superior entera por la anterior o siguiente según corresponda. De esta manera, podemos se puede presentar en pantalla un contenedor del tamaño que se quiera, manteniendo gran parte oculta.



- Visualización de gráficos del apartado 'Estadísticas': implementados mediante el uso de la librería Javascript ya mencionada. Se seleccionan todas las ubicaciones existentes en el sistema, obviando aquellas en las que ha estado un paciente más de una vez de forma que el resultado final son dos gráficos informativos del número de pacientes por ubicación en el centro médico. Dichos gráficos admiten valores tanto para el eje de ordenadas como para el de abscisas. A modo complementario, se le adjudicó a cada barra el mismo color que tiene la ubicación a la que corresponden en el apartado 'Seguimiento' (a excepción de aquellas cuyo color era blanco).



## 7. PRUEBAS

En esta sección se detallan las pruebas realizadas para comprobar que el sistema cumple los Requisitos Funcionales (3.1) y Requisitos No Funcionales (3.2) establecidos en el inicio de este documento.

Para ello, se establecen una serie de pruebas de caja negra en las que se verifica que el sistema responde correctamente a las interacciones con el usuario. Para comprobar cada uno de los requisitos puede ser necesario hacer varias pruebas por lo que la identificación no corresponde con el requisito a probar.

Prueba	Descripción	Resultado
P01	El usuario puede acceder al sistema.	+
P02	Una vez identificado, la pantalla mostrada es 'Admisión'.	+
P03	El paciente se registra correctamente, mostrándose la pantalla de 'Seguimiento'.	+
P04	Si el paciente a insertar ya existe, se lanza un aviso y se cambia a la pestaña 'Seguimiento'.	+
P05	Las ubicaciones se identifican intuitiva e inequívocamente.	+
P06	Se guardan correctamente las ubicaciones una vez arrastradas al contenedor central.	+
P07	Cada ubicación muestra la hora en la que se introdujo, el tiempo total que se pasó en urgencias hasta el momento y el tiempo parcial de la ubicación.	+
P08	Si se introduce una ubicación en un lugar equivocado, la ubicación no se introduce o se reubica correctamente.	+
P09	Se carga la información del paciente a buscar.	+
P10	Si el paciente a buscar no existe, se presenta la pantalla 'Admisión'.	+
P11	Si se inserta una ubicación final, se libera la cama de 'BOX' de ser el caso y no se permite la introducción de más ubicaciones.	+

<b>P12</b>	Se localiza inequívocamente al paciente en su cama correspondiente.	+
<b>P13</b>	Los ‘carruseles’ mueven la información presentada de forma fluida.	+
<b>P14</b>	Se muestran los gráficos correctamente y de forma dinámica.	+
<b>P15</b>	El usuario es redirigido a la ventana de ‘LogIn’ una vez cerrada la sesión.	+