



**Sistema de Gestión de Camas**  
**Centro Médico de Caracas**  
Documento de Arquitectura de Software  
**Versión 1.0**

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## Historia de Revisión

Fecha	Versión	Descripción	Autor
27/06/2014	1.0	Propuesta inicial del proyecto a desarrollar	Rebeca Machado Karen Troiano Leonardo Ramos Ronier Rodríguez

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## Tabla de Contenidos

1.	Introducción	4
1.1	Propósito	4
1.2	Alcance	4
1.3	Definiciones, Siglas, y Abreviaciones	4
1.4	Referencias	5
2.	Representación Arquitectónica	5
3.	Metas y Restricciones Arquitectónicas	6
4.	Vista de Casos de Uso	7
4.1	Modelo de Casos de Uso	7
4.2	Detalle de Actores	8
4.3	Resumen de Casos de Uso y Actores	8
4.4	Detalle de Casos de Uso	10
5.	Vista Lógica	14
5.1	Visión general	14
5.1.1	Diagrama de Clases	14
5.1.2	Diagrama de Actividad del Negocio	16
6.	Vista de Implementación	17
6.1	Diagrama de Componentes	17
7.	Vista de Datos	18
7.1	Modelo ER-E	18
7.2	Diccionario de datos	18
7.2.1	Entidades	18
7.2.2	Interrelaciones	21
8.	Librerías y Componentes a Considerar	21
9.	Calidad	22

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

# Documento de Arquitectura de Software

## 1. Introducción

### 1.1 Propósito

El presente documento tiene como objetivo describir de manera concreta la arquitectura del software y los componentes más relevantes del sistema, explicándolos mediante el uso de la vista lógica y de implementación.

El documento de arquitectura de software busca mostrar de forma clara y concisa la estructura y diseño de la aplicación según las especificaciones de las reglas del negocio con los clientes, también sus componentes y su funcionamiento a cualquier persona interesada en la misma, desde una percepción amplia y general hasta una visión más detallada y puntual. Por esta razón es el documento esencial para cualquier personal futuro que se vea involucrado en el mantenimiento o actualización del software.

### 1.2 Alcance

Este documento muestra las distintas vistas del sistema en detalle, los casos de uso, la funcionalidad esperada, los componentes de la aplicación, las metas y restricciones arquitectónicas; que serán de gran utilidad para cualquier desarrollador interesado en la actualización del sistema.

El DAS no pretende ser un manual de instalación o una especificación rápida de la funcionalidad del software; sino que muestra detalladamente la concepción, análisis y diseño de toda la arquitectura del mismo después de haberse razonado, planteado, evaluado y aceptado.

### 1.3 Definiciones, Siglas, y Abreviaciones

- **CMC:** Centro médico de Caracas
- **DAS:** Documento de Arquitectura de Software.
- **Entorno o ambiente de producción:** El producto final, instalado en los equipos del cliente o usuario final.
- **Framework:** Software o herramienta utilizada para implementar el sistema.
- **Modelo ER-E:** Modelo Entidad-Relación Extendido. Corresponde a una herramienta de representación de datos de manera gráfica; utilizado principalmente para modelar bases de datos relacionales.
- **MVC:** Modelo Vista Controlador. Patrón de desarrollo de software que separa la vista o interfaz de la lógica del negocio.
- **Stakeholders:** Son los entes interesados o involucrados en la realización del sistema

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

- **RUP:** Siglas correspondientes a *Rational Unified Process*, que es una metodología de desarrollo de software que se lleva a cabo mediante iteraciones, basándose en la arquitectura del sistema y en sus casos de uso.
- **USB:** Universidad Simón Bolívar.

## 1.4 Referencias

- **Documento Visión** Versión 1.0; realizado por el equipo de trabajo.
- **Manual de Instalación** Versión 1.0; realizado por el equipo de trabajo.
- **Manual del Programador** Versión 1.0; realizado por el equipo de trabajo.
- **Plantilla para Gestión de Camas** presentada por el cliente.

## 2. Representación Arquitectónica

Con el objetivo de describir a cabalidad la arquitectura del Sistema de Gestión de Camas, se utilizará el modelo de las 4+1 vistas, desarrollado por el profesor Philippe Kruchten, que se estructura de la siguiente manera:



La vista lógica describe los conceptos y clases relacionados al sistema, y la manera en la que se relacionan. Para el caso de este documento, la vista lógica se explicará mediante el uso del diagrama de clases y la actividad del negocio.

Por su parte, la vista de despliegue, o vista de implantación y la vista de implementación exponen al sistema desde el punto de vista de los desarrolladores, los componentes que conforman al sistema y cómo éstos están

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

relacionados.

En la vista de procesos, se describen requerimientos no funcionales del sistema, tales como la tolerancia a fallas, el rendimiento, y la disponibilidad del sistema. De la misma manera, en esta vista se incluyen los procesos que se llevan a cabo dentro del sistema y su comunicación, la sincronización y control de concurrencia entre ellos. Esta vista no será desarrollada para efectos de este documento.

Finalmente, la vista de casos de uso contiene los casos de uso del sistema y la relación existente entre ellos. Su importancia radica en que son el punto de partida para el proceso de desarrollo del software; deben ser claros, sencillos y comprensibles tanto para los desarrolladores como para el cliente. Esta vista se apoya en el uso del diagrama de casos de uso, y la descripción y flujo de cada uno.

Aunque este modelo no incluye una vista de datos, en este documento es incluida, ya que su desarrollo permite exponer los datos que maneja el sistema y cómo estos se relacionan. Para cumplir con el objetivo de esta vista, se utiliza un diagrama ER-E en el que se observan los datos que son manejados por el software.

### 3. Metas y Restricciones Arquitectónicas

El desarrollo del Sistema de Gestión de Camas tiene como principal meta el establecimiento de un software confiable, eficiente y de calidad, cuya interfaz sea amigable al usuario y que sea sencilla de utilizar. Como el sistema será utilizado vía web, resulta indispensable cumplir el requerimiento de que pueda ser accedido desde cualquier navegador y siga cumpliendo con todos los estándares de calidad.

El Sistema de Gestión de Camas será utilizado en el Centro Médico de Caracas, por lo que es estrictamente necesario utilizar las herramientas que hacen posible el mantenimiento y reutilización del sistema, tales como **Django** (<https://www.djangoproject.com/>), framework que permite el desarrollo del software basado en el Modelo Vista Controlador de forma rápida y eficiente.

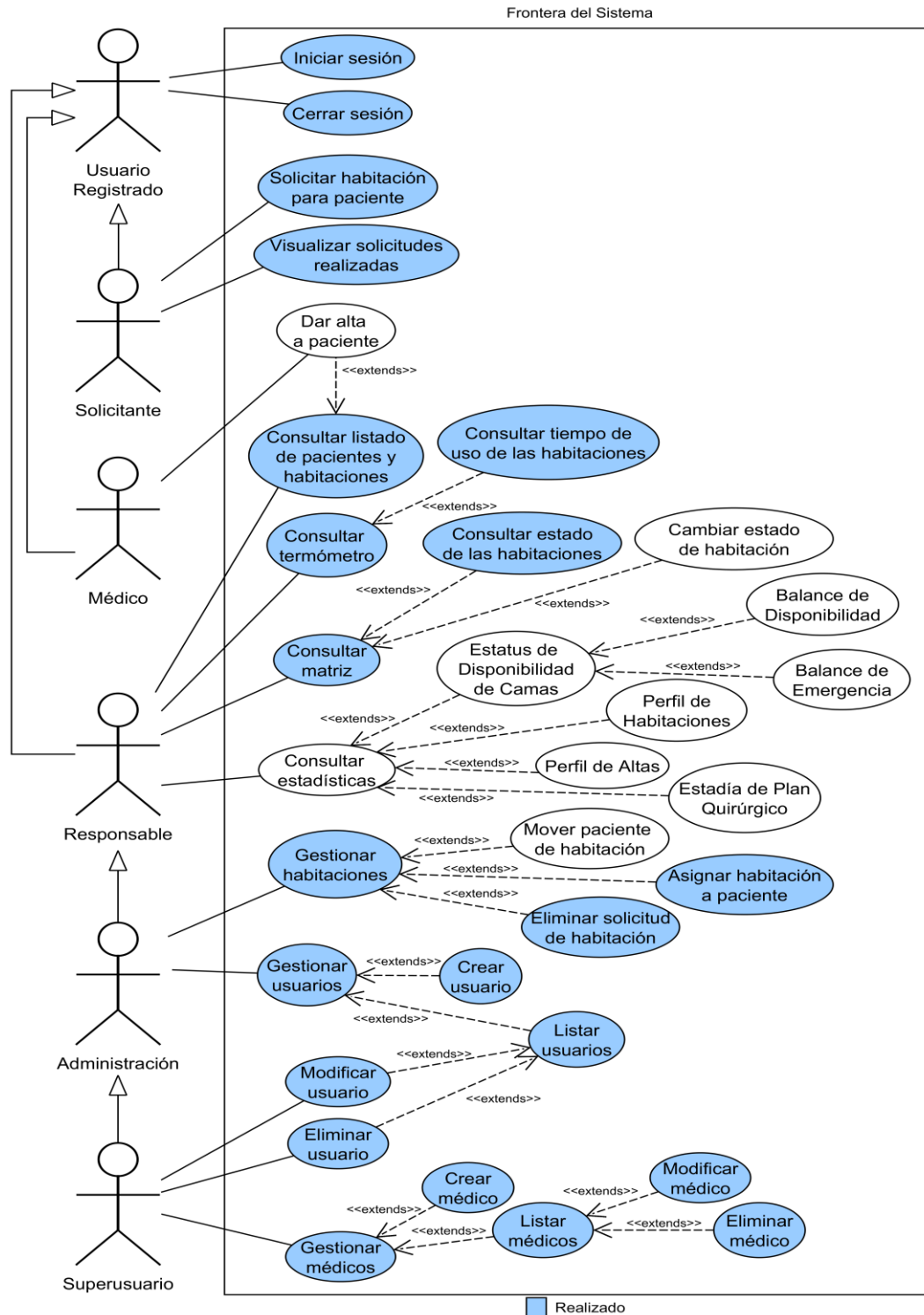
Igualmente, con el fin de llevar un control sobre las versiones del código que se van desarrollando, es necesario el correcto uso de un gestor de configuraciones. El equipo desarrollador ha utilizado GitHub (<https://github.com/>) para efectos de este documento.

Es posible que situaciones expresadas dentro de este documento cambien a petición del cliente; bien sea funcionalidades, componentes externos e/o internos, conexiones o incluso ambiente de implantación.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## 4. Vista de Casos de Uso

### 4.1 Modelo de Casos de Uso



Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## 4.2 Detalle de Actores

- **Usuario Registrado:** Se refiere a cualquier usuario que pueda acceder al sistema con un nombre de usuario y contraseña. Cada usuario registrado tiene un rol dentro de la aplicación: Solicitante, Médico, Responsable, Administración o Superusuario.
- **Solicitante:** Corresponde a un único usuario con el rol de Solicitante, que puede completar un formulario para solicitar una habitación para un paciente de cualquier sección médica del hospital (partos, emergencias, quirófanos, traslados, entre otros).
- **Médico:** Se refiere a cualquier usuario registrado con el rol de Médico; que puede dar el alta a sus pacientes a través del sistema.
- **Responsable:** Corresponde a aquella persona encargada del proceso de gestión de habitaciones. Tendrá un usuario dentro del sistema, pudiendo realizar las funciones que le competen.
- **Administración:** Representa a la administración del hospital, encargada principalmente de autorizar la asignación de camas; pudiendo también acceder a la funcionalidad completa del sistema mediante su usuario.
- **Superusuario:** Gestiona principalmente los catálogos de usuarios y médicos en caso de que se deba hacer alguna modificación a los datos cargados. Accede a todas las funcionalidades del sistema.

## 4.3 Resumen de Casos de Uso y Actores

Caso de Uso	Actor
Iniciar sesión	Usuario Registrado, Solicitante, Médico, Responsable, Administración, Superusuario
Cerrar sesión	Usuario Registrado, Solicitante, Médico, Responsable, Administración, Superusuario
Solicitar habitación para paciente	Solicitante
Visualizar solicitudes realizadas	Solicitante
Dar alta a paciente	Médico
Consultar listado de pacientes y habitaciones	Responsable, Administración, Superusuario
Consultar termómetro	Responsable, Administración, Superusuario
Consultar tiempo de uso de la habitaciones	Responsable, Administración, Superusuario



Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

Consultar matriz	Responsable, Administración, Superusuario
Consultar estado de las habitaciones	Responsable, Administración, Superusuario
Cambiar estado de habitación	Responsable, Administración, Superusuario
Consultar estadísticas	Responsable, Administración, Superusuario
Consultar estadística “Estatus de Disponibilidad de Camas”	Responsable, Administración, Superusuario
Consultar estadística “Balance de Disponibilidad”	Responsable, Administración, Superusuario
Consultar estadística “Balance de Emergencia”	Responsable, Administración, Superusuario
Consultar estadística “Perfil de Habitaciones”	Responsable, Administración, Superusuario
Consultar estadística “Perfil de Altas”	Responsable, Administración, Superusuario
Consultar estadística “Estadía de Plan Quirúrgico”	Responsable, Administración, Superusuario
Gestionar habitaciones	Administración, Superusuario
Mover paciente de habitación	Administración, Superusuario
Asignar habitación a paciente	Administración, Superusuario
Eliminar solicitud de habitación	Administración, Superusuario
Gestionar usuarios	Administración, Superusuario
Crear usuario	Administración, Superusuario
Listar usuarios	Administración, Superusuario
Modificar usuario	Superusuario
Eliminar usuario	Superusuario
Gestionar médicos	Superusuario
Crear médico	Superusuario
Listar médicos	Superusuario
Modificar médico	Superusuario
Eliminar médico	Superusuario

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

#### 4.4 Detalle de Casos de Uso

- **Iniciar sesión:** Los actores asociados deben poder iniciar sesión en el sistema mediante su nombre de usuario y contraseña para poder realizar el resto de las funcionalidades. No se debe presentar ninguna opción o poder realizar alguna función sin haber efectuado antes este caso de uso; favoreciendo los requisitos no funcionales de seguridad respecto a autenticación y autorización. A su vez, debe garantizarse que la vista mostrada de la aplicación –una vez iniciada la sesión– corresponda al rol del usuario que ingresa. Se puede iniciar sesión desde la pantalla principal del sistema o “home” y esta debe existir hasta que se cierre.

*Nota para el programador:* Actualmente se está manejando el sistema de autenticación de Django. Se recomienda continuar con el mismo.

- **Cerrar sesión:** Los actores asociados pueden cerrar una sesión iniciada en cualquier momento. Una vez cerrada la sesión, no se debe poder realizar ninguna funcionalidad del sistema.
- **Solicitar habitación para paciente:** El actor correspondiente debe poder solicitar una habitación para un paciente proveniente de todas las secciones del hospital: Quirófano (operaciones planificadas), Emergencias de Adultos y Pediátricas (pacientes ingresados por emergencia), Partos (madres que necesitan hospitalización), UCI (pacientes de la Unidad de Cuidados Intensivos que son hospitalizados) y Especiales (pacientes que requieren una habitación con una prioridad superior). A su vez, otras razones para solicitar una habitación pueden referirse a Adicionales (pacientes a los que se le realizan exámenes y deben permanecer en el hospital), Traslado (un paciente movido de una habitación a otra) y Otros (no especificados). Todas estas secciones utilizarán el mismo usuario para realizar sus solicitudes.

*Nota para el programador:* Actualmente se maneja la obtención de los datos del paciente como una consulta Ajax a la base de datos; para evitar inconsistencias con los datos ingresados en el formulario y los reales del paciente. Estos datos deben provenir de un sistema externo. Los datos de los médicos se encuentran en la base de datos y son obtenidos mediante la librería django-selectable en el formulario.

- **Visualizar solicitudes realizadas:** El solicitante debe poder listar las solicitudes que se han realizado, la sección del hospital que la realizó y su estado (Activa o No Activa). Una solicitud está activa cuando aún está en espera de procesamiento; o, aún no se ha asignado la habitación al paciente relacionado a esta solicitud. Las solicitudes No Activas funcionan como historial de solicitudes realizadas.
- **Dar alta a paciente:** Un médico debe poder acceder al sistema y darle de alta a los pacientes bajo su cuidado o tratamiento. Esto se realiza en la vista de censo, con la opción de dar de alta a cada paciente. En la actual versión, el sistema no limita que la opción aparezca solo en aquellos pacientes correspondientes al médico que ha iniciado sesión; ya que no ha sido especificado por el cliente.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

*Nota para el programador:* Al dar de alta a un paciente, sus datos y los de la habitación deben pasar a un sistema externo para manejar el estado del alta. Esto se maneja mediante una operación Ajax. Este Web Service debe ser implementado en futuras versiones.

- **Consultar listado de pacientes y habitaciones:** Dentro del Sistema de Gestión de Camas, los actores correspondientes pueden consultar el listado de todas las camas ocupadas del hospital, junto con el paciente que la ocupa, el médico tratante y la fecha estimada de salida además del número de días que lleva ingresado en la habitación y en la institución (dos cálculos diferentes); coloreados según el número: verde (0-1 día ocupada), amarillo (2-4 días ocupada) y rojo (5 o más días ocupada). Esta lista de ingresos se corresponde a la vista de “Censo” dentro de las opciones del sistema. Provee filtros de búsqueda y ordenamiento para facilidades del usuario.
- **Consultar termómetro:** Los actores asociados pueden consultar la vista de termómetro semanal (actual y anterior), mediante un historial. La vista de termómetro de un día provee una pila de habitaciones coloreadas según el número de días que lleva ocupada. Se debe presentar la opción de ver una semana anterior o una fecha específica.

*Nota para el programador:* El historial se maneja con la librería django-simple-history.

- **Consultar tiempo de uso de las habitaciones:** Caso de uso inherente al termómetro. Las habitaciones deben estar coloreadas según lo siguiente: azul (libre), verde (0-1 día ocupada), amarillo (2-4 días ocupada) y rojo (5 o más días ocupada).
- **Consultar matriz:** Muestra la vista en matriz de todas las habitaciones del día en curso, coloreadas según su estado.
- **Consultar estado de las habitaciones:** Caso de uso inherente a la matriz. Las habitaciones deben estar coloreadas según lo siguiente: blanco (libre), azul (en proceso de alta), verde (en limpieza), rojo (en mantenimiento) y amarillo (ocupada).
- **Cambiar estado de habitación:** Los actores asociados pueden, mediante el sistema, reportar el cambio de estado de una habitación a “En Mantenimiento”. Esto se puede realizar desde la vista de la matriz y agregando una pequeña observación al respecto del cambio. (P.E.: “En mantenimiento por reparación del aire acondicionado.”)

*Nota para el programador:* El cliente desea que esta funcionalidad esté adherida a la vista de la matriz, por lo que recomendó una funcionalidad de “clic derecho” sobre la habitación, con una ventana emergente que permitiera añadir la observación mencionada y seleccionar el nuevo estado de “En Mantenimiento”.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

- **Consultar estadística:** Este caso de uso, activado por la Administración y el Responsable, permite consultar las estadísticas del proceso de asignación y liberación de camas, con gráficos y tablas de datos que genera el sistema. Las vistas son semanales, por lo que debe manejarse un historial. Se debe presentar la opción de consultar una semana anterior o una fecha en particular.
- **Consultar estadística de Estatus de Disponibilidad de Camas:** El sistema muestra, por día (vista semanal), la cantidad de camas disponibles al inicio del día, las altas dadas, las camas asignadas a intervenciones quirúrgicas planificadas, las asignadas a emergencias y el saldo final de camas para presentar una estadística de disponibilidad y asignación.
- **Consultar estadística de Balance de Disponibilidad:** El sistema muestra, por día (vista semanal), la cantidad de camas disponibles al inicio del día y saldo final después de las altas y asignaciones competentes para generar una estadística de balance final.
- **Consultar estadística de Balance de Emergencia:** El sistema muestra, por día (vista semanal), la cantidad de camas requeridas por emergencia y la cantidad de camas asignadas a la misma para generar una estadística de balance final.
- **Consultar estadística de Perfil de Habitaciones:** El sistema muestra, por día (vista semanal), la cantidad de camas con uno, dos, tres o más días de uso; así como también las libres, para generar una estadística de perfil de habitaciones.
- **Consultar estadística de Perfil de Altas:** El sistema muestral, por día (vista semanal), la cantidad de camas liberadas con cierta cantidad de días de uso (uno, dos, tres o más) para generar una estadística de egresos.
- **Consultar estadística de Estadía de Plan Quirúrgico:** El sistema muestra, por día (vista semanal), el conteo de camas con cierta cantidad estimada de días de uso (uno, dos, tres o más) para pacientes provenientes de quirófano.
- **Gestionar habitaciones:** La Administración del hospital puede asignarle una cama a un paciente según su prioridad y juicio, trasladar a un paciente de habitación y eliminar solicitudes realizadas.
- **Mover paciente de habitación:** La Administración puede efectuar traslados de pacientes que ya ocupaban una habitación, una vez se haya realizado una solicitud de habitación correspondiente. Esto consiste en liberar una habitación o reportarla en mantenimiento y asignar el mismo paciente a una nueva cama.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

- **Asignar habitación a paciente:** La Administración del hospital puede asignar un paciente a una cama libre. La sección de la cual proviene el paciente debe haber realizado una solicitud correspondiente al mismo paciente antes de poder ser asignado, sin excepción.
- **Eliminar solicitud de habitación:** La Administración puede eliminar una solicitud de habitación realizada por alguna sección del hospital.  
*Nota para el programador:* Se maneja actualmente como una operación Ajax en la vista de asignación de camas.
- **Gestionar usuarios:** La Administración y el Superusuario pueden gestionar el catálogo de usuarios del sistema. El actor con el rol de Administración solo podrá crear y listar los usuarios, mas no podrá eliminarlos o modificarlos; a diferencia del Superusuario.
- **Crear usuario:** Se debe poder crear un usuario mediante un formulario o mediante la página de administración del framework.
- **Listar usuarios:** Se deben poder listar todos los usuarios del sistema; mediante una opción en el menú o mediante la página de administración del framework.
- **Modificar usuario:** Se pueden cambiar los datos y atributos de un usuario registrado en el sistema mediante la página de administración del framework.
- **Eliminar usuario:** Se puede eliminar un usuario mediante la página de administración del framework.
- **Gestionar médicos:** El Superusuario podrá gestionar el catálogo de médicos del hospital.  
*Nota para el programador:* Actualmente todos los médicos son ingresados automáticamente en la base de datos con un script de Django; por lo que esto solo debería ser realizado en casos excepcionales.
- **Crear médico:** Se puede crear un médico mediante la página de administración del framework.
- **Listar médicos:** Se pueden listar todos los médicos del sistema mediante la página de administración del framework.
- **Modificar médico:** Se pueden modificar los atributos de un médico en particular mediante la página de administración del framework.
- **Eliminar médico:** Se puede eliminar un médico en particular mediante la página de administración del framework.

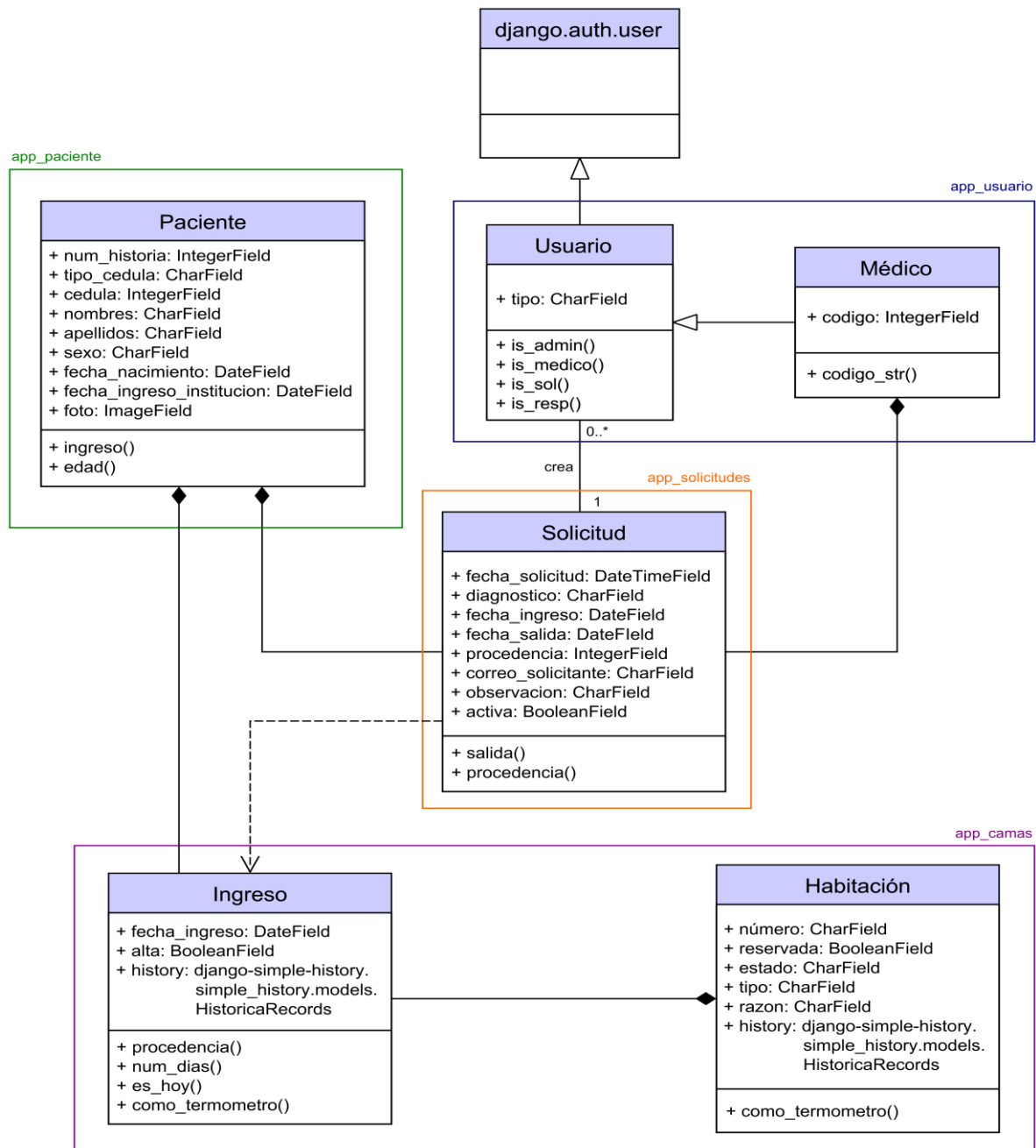
Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## 5. Vista Lógica

### 5.1 Visión general

En esta vista se definirán las clases y conceptos relativos al sistema, y la relación que existe entre ellos. A continuación se presentan el diagrama de clases y el diagrama de actividad que representa la lógica del negocio.

#### 5.1.1 Diagrama de Clases



Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

El diagrama de clases muestra la estructura y jerarquía de las clases presentadas en el proyecto mediante el framework de Django; divididas e identificadas según las aplicaciones creadas y con los tipos de atributos asociados al mismo ambiente de desarrollo.

Algunos atributos han sido agregados o modificados en comparación con el diagrama ER-E por fines conceptuales, ya que Django transforma todas las clases en relaciones de la base de datos; esto significa que, en un caso general, no debería existir ninguna diferencia entre el diagrama ER-E y el aquí presentado diagrama de clases. El equipo desarrollador ha decidido introducir estas diferencias en caso de que se desee trasladar la aplicación a otro framework de desarrollo que no se maneje de la misma forma.

Se recomienda mantener actualizado en la medida de lo posible este diagrama con los cambios que se realicen en otra etapa del desarrollo. De esta forma se promueve la mantenibilidad del proyecto.

La clase “Paciente” debe desaparecer al momento de implantación pues representa un componente externo aún no adherido (véase diagrama de componentes en la vista de implantación). Se ha mantenido a lo largo de esta etapa del desarrollo por facilidades obvias del equipo programador. Por esta razón, no se representa en los casos de uso alguno referente al manejo de catálogos de pacientes. Esto permanecerá así hasta que sea implantado el sistema, unido al componente externo; o hasta que el cliente exprese un nuevo requerimiento.

De ser necesario para mayor comprensión del diagrama, se adjuntan dos referencias utilizadas a la hora del desarrollo:

[1] **django.auth.user**: Modelo “user” del sistema de autenticación de django. Este sistema añade robustez al resto de la aplicación, además de hacerla más estándar y sencilla de comprender. Para más información sobre atributos y métodos, ver:

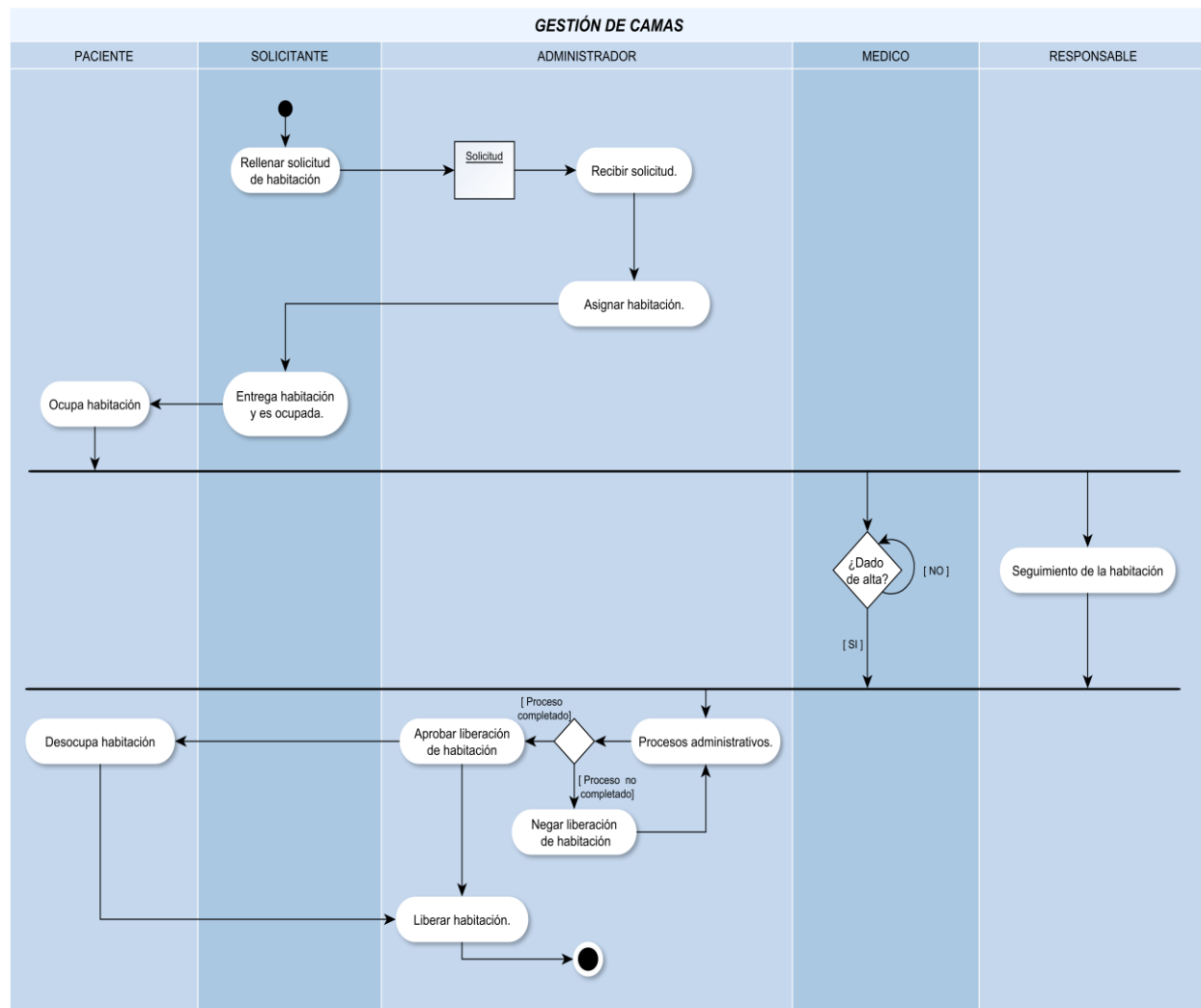
<https://docs.djangoproject.com/en/dev/topics/auth/>

[2] **django-simple-history.simple\_history.models.HistoricalRecords**: Utilidad desarrollada para Django que permite almacenar cambios y modificaciones hechos a una instancia de un modelo o clase. De esta manera se maneja el historial necesario para ciertos casos de uso de la aplicación. Para más información sobre este paquete, ver:

<https://django-simple-history.readthedocs.org/en/latest/>

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

### 5.1.2 Diagrama de Actividad del Negocio



El diagrama de actividad del negocio muestra una vista genérica de cómo funciona aquello que se quiere modelar, automatizar, optimizar o mejorar con la aplicación en desarrollo. En este caso se representa lo siguiente:

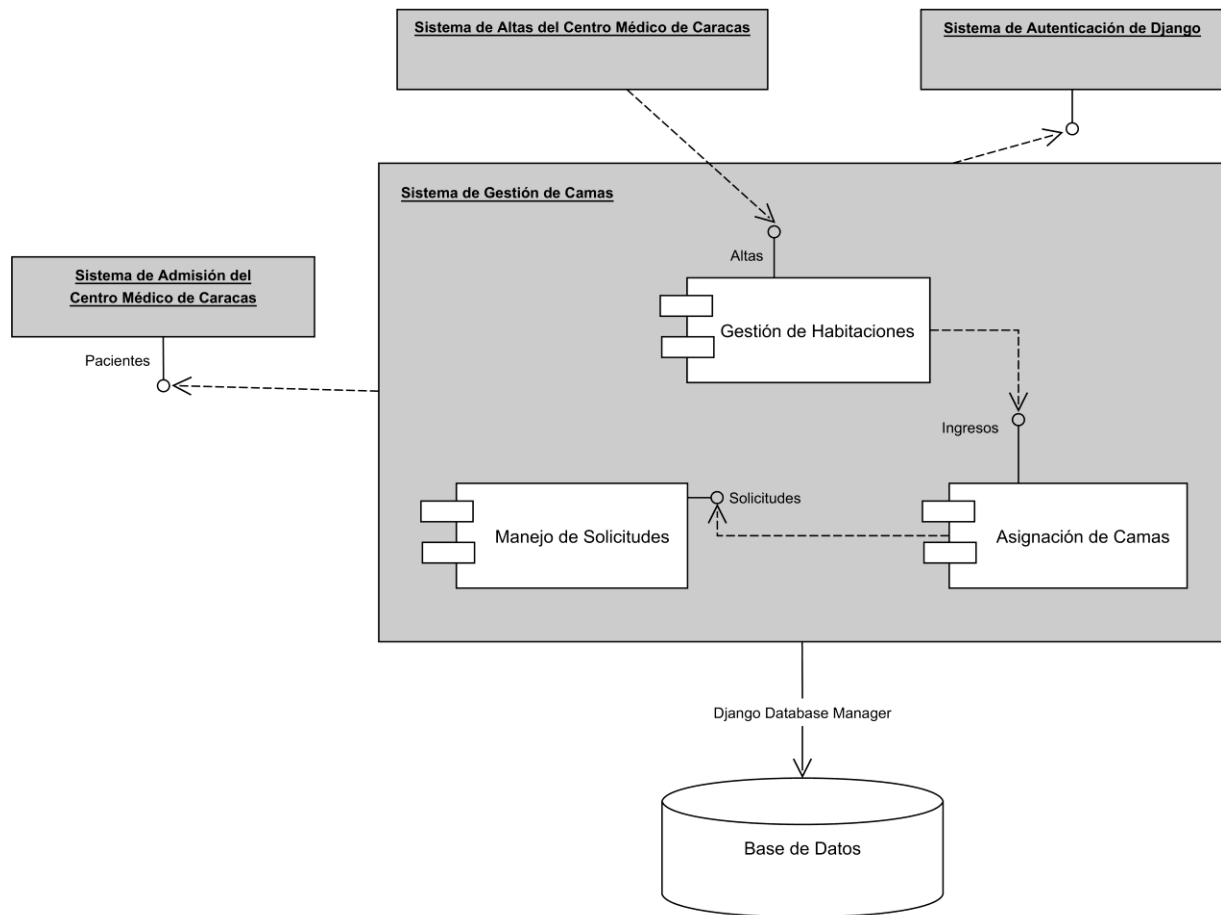
El proceso de negocio se activa cuando un actor solicitante (véase detalle de actores) realiza una solicitud asociada a un paciente de su área. Esta solicitud es recibida por administración, quien asigna una habitación a este paciente. La notificación de asignación es enviada y recibida de nuevo por el solicitante, que hace ocupar la habitación a dicho paciente. Durante su estadía en la habitación, el responsable de la gestión de camas en el hospital monitorea continuamente el estado de las habitaciones. Una vez el médico asociado decide dar el alta al paciente ingresado, este desocupa la habitación y después de diferentes procesos administrativos (facturación, caja y limpieza) se declara “liberada” la cama para dar paso a un nuevo proceso con un nuevo paciente.



Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## 6. Vista de Implementación

### 6.1 Diagrama de Componentes



La aplicación opera con diferentes componentes externos explicados a continuación:

El **Sistema de Admisión del Centro Médico de Caracas** posee el registro de todos los pacientes ingresados a la institución. De allí se obtienen los datos de los mismos para realizar solicitudes. Esto está simulado actualmente en la base de datos.

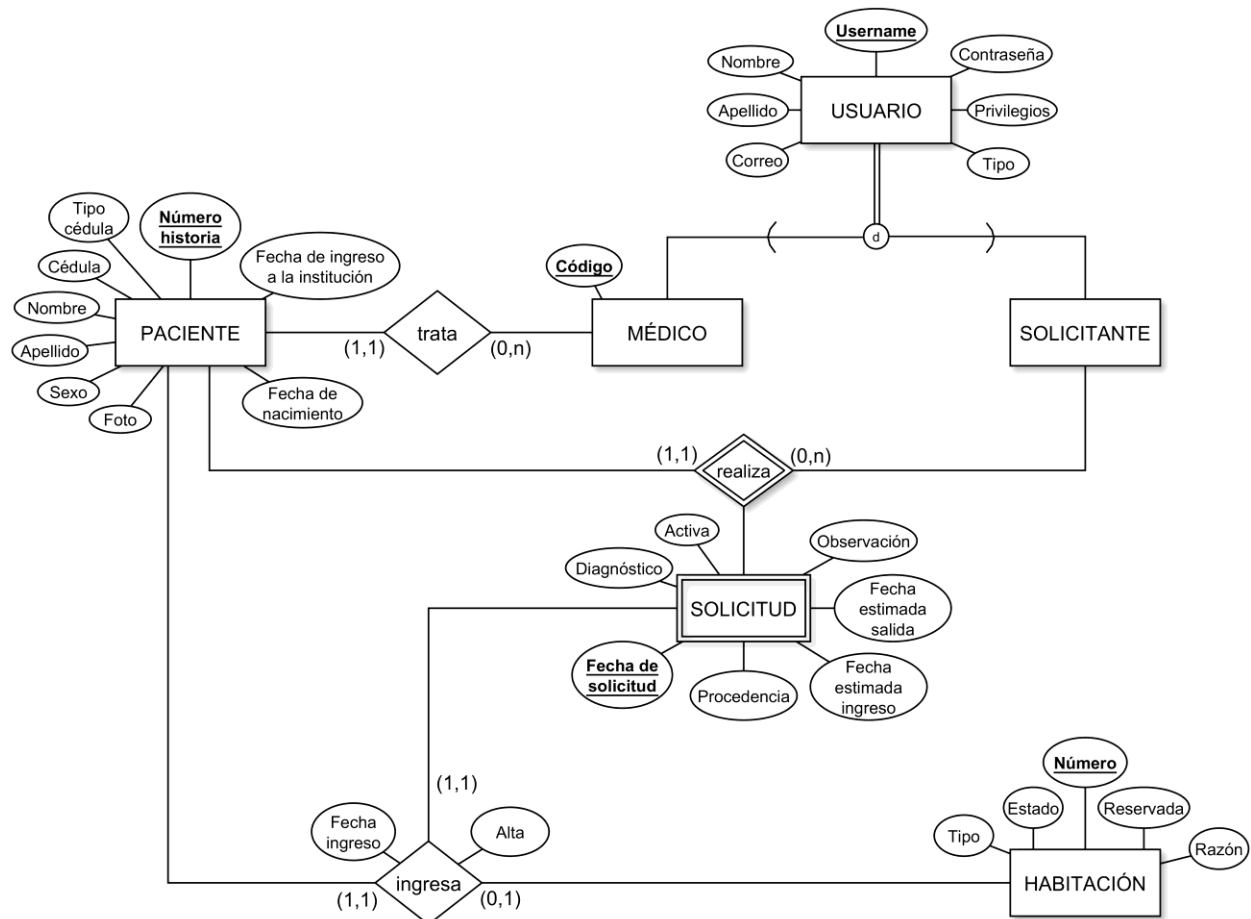
El **Sistema de Altas del Centro Médico de Caracas** se encuentra en desarrollo paralelo con el presentado en este documento. Se prevé que se maneje el proceso de las altas de un paciente una vez el médico realiza esta acción. Tal sistema obtiene los datos del Sistema de Gestión de Camas.

El **Sistema de Autenticación de Django** funciona como base para manejar las sesiones y los privilegios. Se recomienda mantenerlo y utilizarlo apropiadamente.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## 7. Vista de Datos

### 7.1 Modelo ER-E



### 7.2 Diccionario de datos

#### 7.2.1 Entidades

Entidad	Semántica	Atributo	Semántica del atributo	Dominio
<b>Habitación</b>	Cama dentro de la institución, destinada a la hospitalización de pacientes.	<u>Número</u>	Identificador de la cama dentro del edificio.	VARCHAR(6)
		Estado	Identifica el estado de la habitación: en proceso de limpieza, ocupada, libre, en proceso de alta o en mantenimiento.	VARCHAR(1)

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

		Reservada	Indica si la cama se encuentra reservada.	BOOLEAN
		Tipo	Identifica el tipo de habitación a ser solicitada: Hospitalización, UCI-A, UCI-P. (Unidad de Cuidados Intensivos Adultos o Pediátrica)	VARCHAR(1)
		Razón	Posibles detalles y observaciones con respecto la habitación que se encuentra en mantenimiento.	VARCHAR(140)
<b>Médico</b>	Médico que atiende a pacientes dentro de la clínica.	<b><u>Código</u></b>	Identificación del médico dentro del hospital.	NUMERIC(3)
<b>Paciente</b>	Paciente que es atendido en la clínica.	<b><u>Número historia</u></b>	Identificador de su historia médica.	NUMERIC(6)
		Tipo cédula	Especifica si el paciente es venezolano o extranjero (V o E).	VARCHAR(1)
		Cédula	Número de cédula del paciente.	NUMERIC(9)
		Nombre	Nombre del paciente.	VARCHAR(20)
		Apellido	Apellido del paciente.	VARCHAR(20)
		Sexo	Sexo del paciente. Masculino o Femenino.	VARCHAR(1)
		Fecha de ingreso a la institución	Fecha de ingreso en la institución (CMC).	DATE
		Fecha de nacimiento	Fecha de nacimiento del paciente.	DATE
		Foto	Ruta a la foto del paciente dentro del sistema.	VARCHAR(100)

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

<b>Usuario</b>	Usuario del sistema.	<b><u>Username</u></b>	Nombre del usuario en el sistema.	VARCHAR(60)
		Contraseña	Contraseña de acceso del usuario al sistema.	VARCHAR(60)
		Privilegios	Tipo de privilegio que tiene el usuario.	VARCHAR(60)
		Tipo	Tipo del usuario: Médico, Administrador, Solicitante, Responsable	VARCHAR(1)
		Nombre	Nombre real del usuario.	VARCHAR(20)
		Apellido	Apellido real del usuario.	VARCHAR(20)
		Correo	Dirección de correo electrónico del usuario.	VARCHAR(100)
<b>Solicitante</b>	Usuario que solicita habitaciones.	-	-	-
<b>Solicitud</b>	Solicitud que realiza un solicitante para solicitar una habitación a un paciente.	<b><u>Fecha de solicitud</u></b>	Fecha en que se realizó la solicitud.	DATETIME
		Procedencia	Sección de procedencia (Quirófano, Emergencia Adultos, Emergencia Pediátrica, Partos, UCI, Adicional, Especial, Traslado, Otros).	NUMERIC(1)
		Fecha estimada ingreso	Fecha estimada en la que el paciente ingresará a la habitación.	DATE
		Fecha estimada egreso	Fecha estimada en la que el paciente egresará de la habitación.	DATE
		Diagnóstico	Diagnóstico del paciente.	VARCHAR(64)
		Activa	Indica si la solicitud está activa	BOOLEAN

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

		Observación	Posibles detalles y observaciones con respecto al paciente y la solicitud.	VARCHAR(140)
--	--	-------------	--	--------------

### 7.2.2 Interrelaciones

Entidad	Semántica	Atributo	Semántica del atributo	Dominio
Ingresa (p, h, s)	Representa el ingreso de un paciente p a una habitación h de la clínica por medio de una solicitud s.	Fecha ingreso	Fecha en la que ingresó el paciente a la habitación.	DATE
		Alta	Indica si el paciente está en proceso de alta o no.	BOOLEAN
Realiza(s,h)	Representa que un solicitante s solicite una habitación h.	-	-	-
Trata(p,m)	Representa que un paciente p está siendo tratado por un médico m.	-	-	-

## 8. Librerías y Componentes a Considerar

Esta aplicación se conecta con diferentes componentes externos también desarrollados en paralelo y que se espera logren ser unidos a la hora de implantar. Sin embargo, la aplicación internamente utiliza ciertas librerías y utilidades que les son buenas conocer al programador; como la muy utilizada compatibilidad con **jQuery**, pensada para incrementar notablemente la usabilidad de la aplicación. A su vez, se manejan peticiones **AJAX** para lograr ciertos efectos solicitados por el cliente al discutir los requerimientos.

Entre las librerías asociadas a Django, se utilizó **django-simple-history** para el manejo de modificaciones sobre ciertos modelos de la aplicación, consecuentemente con el requerimiento de historial que presentó el cliente. La librería **django-selectable** funcionó como apoyo a la usabilidad en los formularios. Por último, la utilidad **south** maneja la migración de datos a la hora de modificar la base de datos de la aplicación. Puede consultar más información sobre estas librerías en el Manual de Instalación (véase referencias).

Se recomienda al programador tener un conocimiento básico de estas herramientas en caso de que se desee continuar con el desarrollo o realizar alguna modificación a la aplicación existente.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 09/06/2014
Documento de Arquitectura de Software	

## 9. Calidad

El Sistema de Gestión de Camas será desarrollado para utilizarse vía web, por lo que es indispensable una conexión a Internet para que los usuarios tengan acceso a las funcionalidades del sistema. Se recuerda que está pensado para usuarios de un centro médico con bajo o medio conocimiento en el área técnica de tecnología; por lo que es necesario mantener unos estándares de calidad altos.

Toda la arquitectura que fue descrita a lo largo de este documento, busca garantizar estos estándares de calidad:

- Alta portabilidad, ya que las herramientas utilizadas para la implementación del sistema permitirán que éste funcione correctamente en diferentes plataformas y sistemas operativos, como Windows y Linux.
- Las librerías utilizadas y el patrón creativo del desarrollo garantizan que la interfaz sea intuitiva, sencilla y agradable al usuario; promoviendo la usabilidad.
- Este sistema contará con una alta mantenibilidad, respaldada por el uso del MVC (MVT en Django) como patrón de desarrollo del software y por la disponibilidad de documentación sobre todos los aspectos relevantes del sistema; que podrán ser consultados en caso de necesitar conocer en profundidad información sobre su arquitectura, bien sea para realizarle cambios, mejoras o extensiones.
- El acceso a una conexión a internet o a la red interna, y la estabilidad del servidor en el que sea instalado el sistema, serán los factores determinantes en la disponibilidad del sistema.
- Buen tiempo de respuesta, pues la aplicación ha sido diseñada de forma que utilice la menor cantidad de memoria; gracias a la creación de consultas simples y la eliminación de las consultas innecesarias.
- Este sistema, al ser desarrollado de forma iterativa siguiendo la metodología ágil, garantiza que al final de cada iteración se verifique si se está cumpliendo con los requerimientos, metas y restricciones del desarrollo del sistema, de forma tal que se provea un software de calidad y que satisfaga las necesidades del cliente.