



**Sistema de Gestión de Camas**  
**Centro Médico de Caracas**  
Manual para el Programador  
**Versión 1.0**

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 27/06/2014
Manual para el programador	

## Historia de Revisión

Fecha	Versión	Descripción	Autor
27/06/2014	1.0	Propuesta de trabajo pendiente 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: 27/06/2014
Manual para el programador	

## Tabla de Contenidos

1.	Introducción	4
1.1	Propósito	4
1.2	Definiciones, Siglas, y Abreviaciones	4
1.3	Referencias	4
2.	Sugerencias	5
3.	Convenciones	5
4.	Pendientes	6

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 27/06/2014
Manual para el programador	

# Manual para el Programador

## 1. Introducción

### 1.1 Propósito

El presente documento tiene como objetivo describir de manera concreta las sugerencias, metas y casos de usos pendientes a desarrollar del sistema para su total funcionalidad. Por esta razón este documento es esencial para cualquier personal futuro que se vea involucrado en el mantenimiento o actualización del sistema.

Esta no es la primera versión del sistema, pues se ha trabajado sobre la base del Sistema de Emergencias elaborado en el año 2013.

### 1.2 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.
- **USB:** Universidad Simón Bolívar.

### 1.3 Referencias

- **Documento Visión** Versión 1.0; realizado por el equipo de trabajo.
- **Documento de Arquitectura de Software** Versión 1.0; realizado por el equipo de trabajo.
- **Manual de Instalación** Versión 1.1; realizado por el equipo de trabajo.
- **Plantilla para Gestión de Camas** presentada por el cliente.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 27/06/2014
Manual para el programador	

## 2. Sugerencias

- Leer el manual de instalación e instalar el sistema de forma eficiente.
- Leer el archivo README.md para tener un conocimiento sobre las carpetas que posee el repositorio del proyecto.
- En la ruta /recursos/configuracion, encontrará el archivo **settings.py** que deberá **configurar y colocar** en /AM, (siglas para Atención Médica) para poder correr el proyecto desde su PC. Este archivo es único para cada persona y será ignorado en el .gitignore del repositorio para evitar conflictos.
- **Evitar** agregar usuarios desde la cuenta de administrador (localhost:800/admin) dado que el usuario no será reconocido en el sistema de autenticación del sistema, por problemas de hash aplicados sobre la contraseña del usuario. Por esta razón se han eliminado estos registros de la interfaz de administrador.
- Se recomiendan **Trello** (<https://trello.com/>) y **yEd** ([http://www.yworks.com/en/products\\_yed\\_about.html](http://www.yworks.com/en/products_yed_about.html)) como herramientas gratis para el manejo del proyecto y documentación (diagramas) respectivamente.
- Mantenga el orden y documentación del código.
- Es de **esencial** importancia leer los siguientes documentos para el desarrollo y mantenimiento del software:
  - **Documento Visión**
  - **Documento de Arquitectura del software (DAS)**
  - **Plantilla para Gestión de Camas** presentada por el cliente.

Es importante en caso de presentar dudas sobre las librerías instaladas o las funcionalidades del sistema.

## 3. Convenciones

- **General**
  - Por obligación, el código se escribe en español.
  - Nombres de funciones y variables en minúsculas y con palabras separadas por '\_'.
  - Indentación con 4 espacios.
  - Usar nombres de variables explícitos y completos.
- **HTML**
  - Nombres de archivos .html separados por '\_'.
  - Estilos se definen en archivos .css
  - Las clases en CSS se definen con la convención de bootstrap, es decir, palabras en minúsculas separadas por '-', por ejemplo: btn-primary
  - La asociación de eventos se hace por jQuery, no en el tag.

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 27/06/2014
Manual para el programador	

- Para escribir tags de HTML se siguen los siguientes lineamientos:

+ Se escribe un tag por línea. Los tags 'hijos' de otros deben ir en una línea separada y con un punto de indentación mayor. Ejemplo:

```
<div>
  <p> El tag hijo </p>
</div>
```

Notar que el tag para cerrar también va en una línea separada.

+ Si un tag no cabe en una línea, entonces se escribe el primer atributo al lado del nombre del tag, y el resto uno por línea, debajo, a la misma altura del primer atributo. Ejemplo:

```
<input type="button"
      class="btn btn-primary"
      value="Aceptar"
      id="boton">
```

- **Python**

- Para la definición y llamadas de funciones, aplican las mismas reglas que para los tags de HTML

## 4. Pendientes

- Finalizar los casos de uso que no se completaron en la primera versión, esencialmente el área de estadísticas (*referencias: Documento de Arquitectura del software y Plantilla para Gestión de Camas*).
- En la vista de la matriz, falta implementar que se pueda seleccionar una habitación y cambiar desde allí el estado de la misma. En el momento que la habitación esté en “mantenimiento” se debe especificar la razón por la cual está en mantenimiento. El cliente desea que esta funcionalidad sea implementada mediante un clic derecho con un pop up o ventana emergente que permita tener un campo de texto para especificar la razón del mantenimiento. Se recomienda utilizar Ajax.
- Al completar el sistema, no olvidar la primera versión del documento: **Manual de Usuario** para los usuarios del Centro Médico de Caracas.
- El paciente posee un atributo “foto” que no está funcional. Deberá ser agregado al sistema.
- Cambiar la página de inicio de la aplicación cuando no se ha iniciado sesión (/templates/index.html) para que

Sistema de Gestión de Camas	Versión: 1.0
Centro Médico de Caracas	Fecha: 27/06/2014
Manual para el programador	

coincida con el sistema.

- En los diagnósticos se deberá autocompletar todas las posibles enfermedades. El listado de todas estas enfermedades se encuentra en /recursos/base de datos/Enfermedades.sql. Para autocompletar con todos los objetos de algún modelo de la base de datos en Django, se recomienda usar django-selectable, librería instalada junto con el sistema (*referencias: Manual de Instalación*). Para cargar automáticamente todas las enfermedades, se recomienda usar archivos .json en la carpeta fixtures de cada aplicación (*ver <https://docs.djangoproject.com/en/dev/howto/initial-data/>*)
- Existe la posibilidad de reservar habitaciones. Esto no se encuentra contemplado en el sistema actual. El cliente sugirió una funcionalidad donde las habitaciones fueran reservadas para asignación cuando se prevé un ingreso de algún paciente que aún no ha hecho alguna solicitud. De ser llevada a cabo, actualizar los diagramas y documentos correspondientes.
- Propiciar la seguridad del sistema utilizando el sistema de autenticación de Django para crear Grupos y privilegios. Actualmente, las funcionalidades del sistema no están limitadas por privilegios, por lo que un usuario normal (dígase un médico) podría realizar una función de administración de conocer el url adecuado.
- Se recomienda proponer un Sistema de Autenticación Central (por ejemplo LDAP) en el Centro Médico de Caracas. Esto, por supuesto, escapa de las labores del Taller de Desarrollo.
- Culminar la funcionalidad de Listar Usuarios.
- Crear el API necesario para comunicar este sistema con el Sistema de Altas llevado a cabo por Carlos Ledezma. Contactar con el equipo desarrollador para obtenerlo.