



## **MANUAL DE TÉCNICO: PROTOTIPO DE SOFTWARE PARA GESTIÓN DE CAMAS**

**Versión: 01**

**Fecha: 29/10/2021**

Queda prohibido cualquier tipo de explotación y, en particular, la reproducción, distribución, comunicación pública y/o transformación, total o parcial, por cualquier medio, de este documento sin el previo consentimiento expreso y por escrito del Hospital Universitario de Santander.

## HOJA DE CONTROL

<b>Entidad</b>	E.S.E HOSPITAL UNIVERSITARIO DE SANTANDER		
<b>Proyecto</b>	Prototipo de Software para Gestión de Camas		
<b>Entregable</b>	Manual Técnico		
<b>Autor</b>	Fabian Alberto Espitia Muñoz, Valentina Goyeneche Calderón		
<b>Versión/Edición</b>	01	<b>Fecha Versión</b>	29/10/2021

## REGISTRO DE CAMBIOS

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
0100	Versión inicial	Fabian Alberto Espitia Muñoz, Valentina Goyeneche Calderón	29/10/2021

## **RESUMEN**

El presente documento consta de un manual de técnico creado para el prototipo de software de gestión de camas del Hospital Universitario de Santander. El siguiente manual guiará a los usuarios que harán soporte al sistema, indicando los requerimientos, las herramientas de desarrollo y despliegue y los pasos a seguir para el manejo y mantenimiento de la aplicación web diseñada.

## **TABLA DE CONTENIDO**

DESARROLLO	4
Instalación y Configuración de Herramientas de desarrollo	5
DESPLIEGUE	8
Herramientas para el despliegue	9
Configuración del Entorno	9
Ejecución	9
ANEXOS	11
BIBLIOGRAFÍA	12

# 1. DESARROLLO

## 1.1 Instalación y Configuración de Herramientas de desarrollo

- **Java:** es un popular lenguaje de programación orientado a objetos, basado en clases diseñado para reducir dependencias al momento de programar. También se define como una plataforma de desarrollo. Reduce costos, acorta los plazos de desarrollo, impulsa la innovación y mejora los servicios de las aplicaciones. (HARTMAN,2021)

**Instalación:** dirigirse al siguiente [link](#) que nos permite descargar la versión **jdk 1.8.0\_202** de java. Se debe descargar el archivo necesario para el sistema operativo de la máquina en la que se alojará la aplicación. Seguir las indicaciones del asistente de instalación.

**Nota:** Cuando se ejecute la aplicación en consola revisar que se esté usando dicha versión de java con el comando `java -version`, de lo contrario puede haber errores de compilación.

- **Spring boot:** Es una infraestructura ligera que facilita la creación de aplicaciones basadas en Spring de una manera rápida, autónoma y con características deseables para producción. (SPRING IO)

**Instalación:** Para el proyecto se trabajó con la versión **2.3.12 RELEASE**, pero no es necesario instalarla en la máquina. Este framework y todas las dependencias son gestionadas por *Maven* desde el archivo **pom.xml** ubicado en la carpeta raíz del proyecto. En caso de que una actualización en el proyecto requiera agregar o modificar una dependencia solo se debe editar este archivo siguiendo el formato establecido.

- **Maven:** Es una herramienta open source creada en el 2001 con el propósito de simplificar y automatizar los procesos de build de las aplicaciones. Principalmente usado para construir y gestionar proyectos Java, pero también puede usarse para lenguajes de programación como C# o Ruby, entre otros. (MAVEN APACHE, 2021)

**Instalación:** No es necesario que la herramienta sea descargada, ya que está incluida en el paquete del proyecto.

- **SQL Server 2012:** Base de datos proporcionada por la institución. Para poder conectar la aplicación con la base de datos de producción se debe editar el archivo **application.yml** ubicado en el directorio **src/main/resources**. Se deben cambiar las propiedades **url**, **username**, **password**. Una vez se conecte la aplicación con la base de datos, se deben crear tres tablas responsables de almacenar la información de autenticación de usuario.

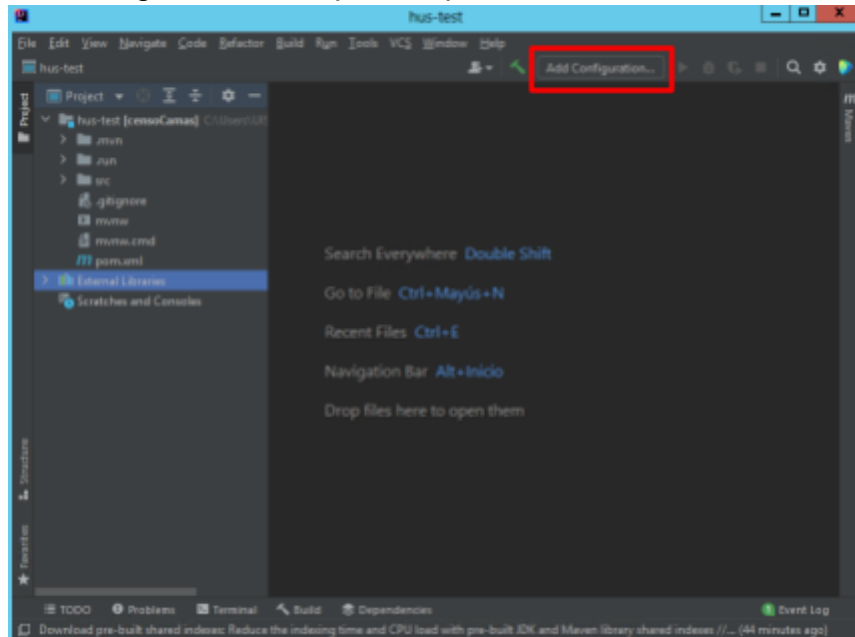
**Instalación:** El script que se encarga de crear las tablas se adjuntará como anexo al final del manual.

- **IntelliJ IDEA:** Es un IDE inteligente y sensible al contexto para trabajar con Java y otros lenguajes JVM como Kotlin, Scala y Groovy en todo tipo de aplicaciones. Proporciona asistencia a la codificación inteligente para todos los lenguajes que soporta. La indexación inicial del código fuente permite al IDE crear un mapa virtual de su proyecto. Gracias a la información del mapa virtual, puede detectar errores sobre la marcha, sugerir variantes de finalización de código con un conocimiento del contexto preciso, realizar refactorizaciones y más. (JETBRAINS)

**Instalación:** Instalar la versión *Community* más reciente disponible en la

[página oficial](#). Una vez se haya instalado siguiendo los pasos del asistente de instalación, se debe importar la carpeta de la aplicación que contiene todo el código fuente. Cuando el proyecto se haya importado por completo, se debe seleccionar la configuración de *run/debug*.

Para esto nos dirigimos a la esquina superior derecha del IDE:



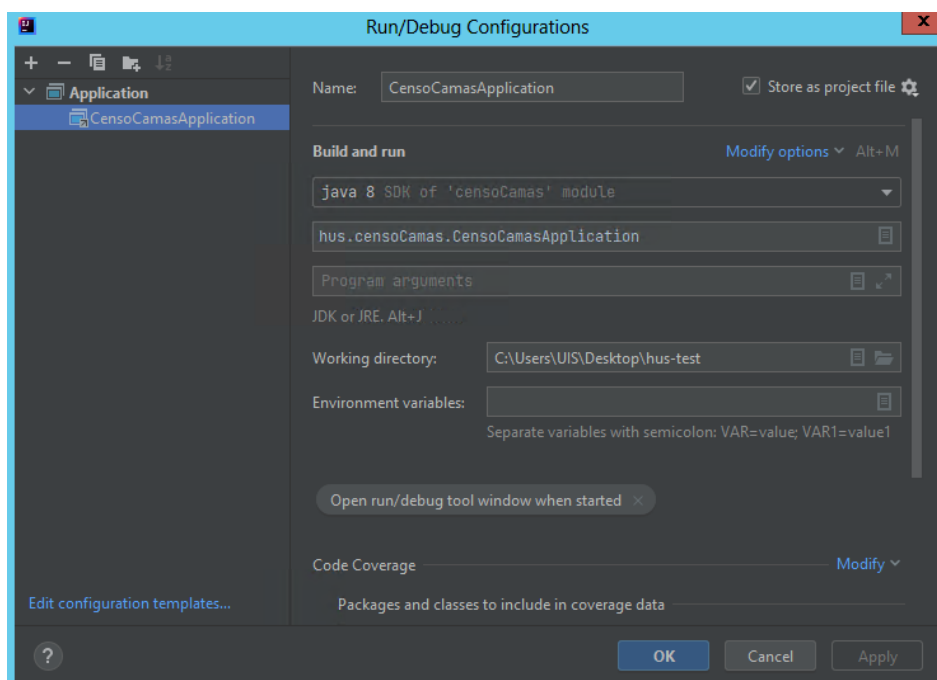
**Imagen1:** Add Configuration.

Damos click donde dice *Add Configuration*.

Se abrirá la ventana de *Run/Debug Configuration*.

Desplegamos el menú *Application*, y seleccionamos la única opción disponible *CensoCamasApplication*.

No es necesario hacer ningún cambio, clickeamos *OK*.

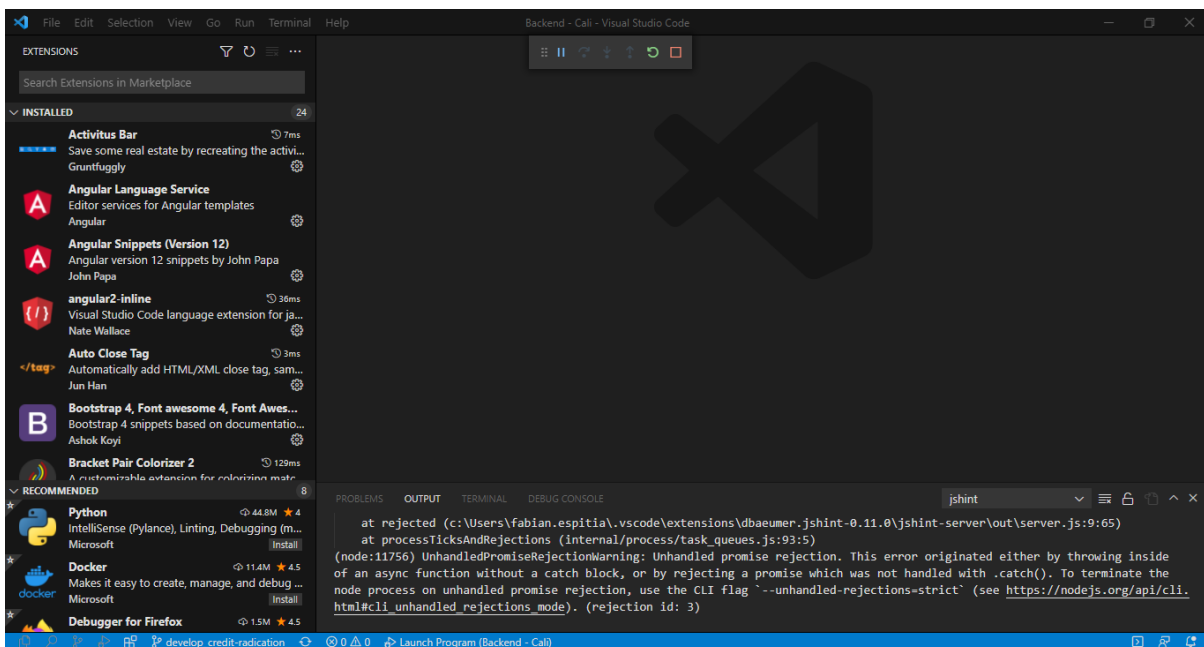


**Imagen2:** Ventana *Run/Debug Configuration*

- **Visual Studio Code:** es un editor de código fuente que permite trabajar con diversos lenguajes de programación, admite gestionar tus propios atajos de teclado y refactorizar el código. Es gratuito, de código abierto y nos proporciona una utilidad para descargar y gestionar extensiones con las que podemos personalizar y potenciar esta herramienta. (AITANA, 2018)

**Instalación:** Se debe descargar el archivo vscode.exe mediante la página oficial de Microsoft. Después, continuaremos descargando dentro del editor de código los distintos plugins mediante la sección extensiones los cuales nos permitirán una mejor lectura del código fuente, en este caso se contó con los siguientes: **(Ver imagen 3)**

- o Angular 5 Snippets – TypeScript.
- o Angular Language Service.
- o Bootstrap 4 & Font Awesome snippets.
- o HTML CSS Support.
- o JavaScript (ES6) code snippets.
- o JS-CSS-HTML Formatter.
- o JSHint.
- o Material Icon Theme
- o Terminal
- o TSLint



**Imagen3:** Imagen demostrativa de la sección “extensions”

- **Angular 11.1.4:** es un framework open source desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC

(Modelo-Vista-Controlador). (Quality Devs, 2016)

**Instalación:** Es necesario instalar la versión 11.1.4 o superior en nuestra máquina para poder leer el código Fuente o generar los archivos para poder subir al servidor. Para instalarlo se realiza mediante el comando en nuestro CMD

```
npm install -g @angular/cli@11.1.4
```



## 2. DESPLIEGUE

### 2.1 Herramientas para el despliegue

- **Java jdk 1.8.0\_202:** se debe verificar que la misma versión del jdk de desarrollo es quien ejecuta el comando de despliegue. Se puede declarar esta versión en el PATH de la máquina para que no haya riesgos de que se use una versión diferente.
- **Maven:** es la herramienta que se encarga de construir, probar y empaquetar la aplicación en formato `.jar`. No es necesaria su instalación en la máquina, se puede trabajar con el wrapper de Maven `mvnw` que está ubicado en el directorio raíz del proyecto.
- **Servidor Hosting:** se deberá contar con un servidor hosting que permita acceder dentro de un sitio web a las funcionalidades del sistema. Después se tendrá que subir los archivos generados en la carpeta `/dist`

### 2.2 Configuración del Entorno

- **Puerto Backend:** Para el despliegue del componente Backend se tiene por defecto el puerto 8080. Si no es posible usar este puerto, se debe editar el archivo `application.yml` ubicado en `src/main/resources`, agregando una nueva configuración con el siguiente formato al final del archivo:

```
server:  
    port: #número de puerto disponible para despliegue
```

- **Archivo jar:** Para desplegar el componente backend en un entorno de producción se genera un archivo `.jar`. Este archivo se crea con los siguientes pasos:
  1. Ubicarse en la carpeta del proyecto, y ejecutar el comando `mvnw install`.
  2. Se habrá generado el archivo `censoCamas-0.0.1-SNAPSHOT.jar` en la carpeta `/target` en la raíz del proyecto.
- **Archivo dist** Para desplegar los archivos en el servidor front end se debe realizar de la siguiente manera:
  1. Ubicarse en la carpeta y ejecutar el comando `ng build -prod`.
  2. Se generará en la carpeta `/dist` los archivos que se subirán al servidor.

### 2.3 Ejecución









- **Ejecución para el despliegue del back-end:**

Para la ejecución en producción del back-end se deberán realizar los siguientes pasos:

1. Dirigirse a la ubicación en el que se haya guardado el archivo `.jar` generado anteriormente, originalmente en la carpeta `/target` del proyecto.
2. Ejecutar el archivo `.jar` con el comando: **(Ver Imagen 4)**.

```
java -jar censoCamas-0.0.1-SNAPSHOT.jar
```



 assets	4/11/2021 2:23 p. m.	Carpeta de archivos	
 3rdpartylicenses.txt	4/11/2021 2:23 p. m.	Documento de te...	17 KB
 favicon.ico	21/07/2021 5:02 p. m.	Icono	1 KB
 index.html	4/11/2021 2:23 p. m.	Microsoft Edge H...	3 KB
 main.adc5d463b01b310b7edc.js	4/11/2021 2:23 p. m.	Archivo JavaScript	2.712 KB
 polyfills.995aa8d020d23b5acd01.js	4/11/2021 2:23 p. m.	Archivo JavaScript	37 KB
 runtime.7b63b9fd40098a2e8207.js	4/11/2021 2:23 p. m.	Archivo JavaScript	2 KB
 styles.6fed496e4b9936d850d0.css	4/11/2021 2:23 p. m.	Documento de ho...	142 KB

**Imagen 6:** Imagen Consola de Comandos

### **3. ANEXOS**

- **Anexo 1: Script para creación de Tablas de usuario**

Se crearon cinco tablas:

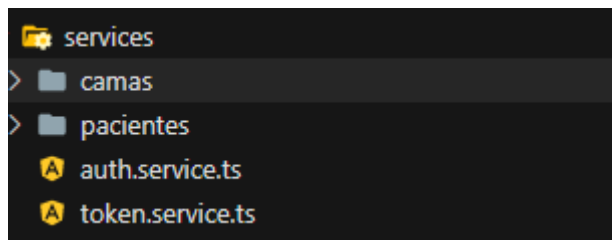
1. HPNCENSOUSU, que almacena los datos de los usuarios: OID, usuario, nombre, clave encriptada.
2. HPNCENSOROL, que almacena los datos de los roles creados: OID y nombre.
3. HPNUSUROL, tabla que almacena la relación entre el usuario y sus roles.
4. TRASLADO, tabla que almacena el registro de traslados con el usuario que lo realiza.
5. CAMALIBERADA, tabla que almacena el registro de liberaciones con el usuario que lo realiza.

Y tres secuencias para la creación de usuario, traslado y liberación, respectivamente.

**Código:**

```
create table HPNCENSOUSU(  
    OID int PRIMARY KEY,  
    HPNUSUNOMBRE varchar(255),  
    HPNUSUDESCRI varchar(255),  
    HPNUSUCLAVE varchar(255),  
    HPNUSUCAMBIOCLAVE int default 0,  
    HPNUSUESTADO int default 1  
)  
  
create table HPNCENSOROL(  
    OID int PRIMARY KEY,  
    HPNROLNOMBRE varchar(255)  
)  
  
create table HPNUSUROL(  
    OIDUSUARIO int FOREIGN KEY REFERENCES HPNCENSOUSU(OID),  
    OIDROL int FOREIGN KEY REFERENCES HPNCENSOROL(OID),  
    PRIMARY KEY(OIDUSUARIO,OIDROL)  
)  
  
create SEQUENCE [dbo].[user_sequence]  
AS [bigint]  
START WITH 1  
INCREMENT BY 1  
MINVALUE -9223372036854775808  
MAXVALUE 9223372036854775807  
CACHE
```

- Para el componente front-end, se tiene dentro del código fuente unos archivos de configuración que me permiten cambiar las rutas URL de las API, la cual están seccionadas de la siguiente manera:



**Imagen 7:** Representación de rutas de servicios API

Dentro de la carpeta “services” encontraremos 3 tipos de servicios, el primero en la carpeta “camas” se encontrarán todos los servicios relacionados con la búsqueda de cama. El segundo, en la carpeta “pacientes” se encontrarán los servicios relacionados a búsqueda y selección de pacientes. Y por último, se encontrara los servicios “auth” y “token” los cuales, contienen las configuraciones de autenticación para el sistema.(Ver Imagen 7)

## **4. BIBLIOGRAFÍA**

1. (16 de Septiembre de 2016). Obtenido de Quality Devs:  
<https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>
2. AITANA. (16 de 10 de 2018). Obtenido de  
<https://blog.aitana.es/2018/10/16/visual-studio-code/>
3. HARTMAN,J. (7 de Octubre de 2021). Obtenido de  
<https://www.guru99.com/java-platform.html>
4. MAVEN APACHE (3 de Noviembre de 2021). Obtenido de  
<https://maven.apache.org/what-is-maven.html>
5. JETBRAINS. Obtenido de <https://www.jetbrains.com/es-es/idea/features/>
6. SPRING IO . Obtenido de <https://spring.io/projects/spring-boot>