



**Diseño de un aplicativo para control de inventario y programa de mantenimiento preventivo de una institución de salud**

**Autor(a):**

**Angelly Del Carmen Rebolledo Ortega**

**Trabajo de grado como prerrequisito para la obtención de grado de  
Tecnólogo en Electromedicina**

**Director(a):**

**Claudia Basto Vera**

**Facultad de Ingeniería**

**Programa de Tecnología en Electromedicina**

**Barranquilla**

**2022**



**Diseño de un aplicativo para control de inventario y programa de mantenimiento preventivo de una institución de salud**

**Autor(a):**

**Angelly Del Carmen Rebolledo Ortega**

**Director(a):**

**Claudia Basto Vera**

**Facultad de Ingeniería**

**Programa de Tecnología en Electromedicina**

**Barranquilla**

**2022**

## ÍNDICE

LISTADO DE TABLAS.....	5
LISTADO DE IMÁGENES .....	6
RESUMEN .....	7
ABSTRACT.....	7
INTRODUCCION .....	8
PLANTEAMIENTO DEL PROBLEMA .....	9
OBJETIVOS .....	10
OBJETIVO GENERAL.....	10
OBJETIVOS ESPECIFICOS .....	10
JUSTIFICACION .....	10
MARCOS DE REFERENCIA.....	11
MARCO TEORICO .....	11
MARCO CONCEPTUAL .....	14
Hoja de vida .....	14
Inventario .....	14
Lenguaje de programación.....	22
Mantenimiento .....	26
Mantenimiento correctivo.....	27
Mantenimiento preventivo .....	27

MARCO LEGAL.....	30
Decreto 4725 de 2005 .....	30
Resolución 2981 de 2011 .....	31
Resolución 3100 de 2019.....	31
METODOLOGIA.....	32
DISEÑO.....	32
TIPO DE INVESTIGACIÓN .....	33
PROCEDIMIENTO.....	33
RECOLECCIÓN DE DATOS PARA EL INVENTARIO .....	33
DISEÑO DEL SOFTWARE DE INVENTARIO Y MANTENIMIENTO.....	34
RESULTADOS.....	36
DISCUSION (ANALISIS DE RESULTADOS).....	38
CONCLUSIONES Y RECOMENDACIONES .....	40
REFERENCIAS.....	42
APENDICE A - Python .....	44

**LISTADO DE TABLAS**

<b>Tabla 1.</b> Datos de inventario. (Organización Mundial de la Salud (OMS), 2012).....	15
<b>Tabla 2.</b> Función del equipo. (Organización Mundial de la Salud (OMS), 2012).....	28
<b>Tabla 3.</b> Riesgo físico asociado con la aplicación clínica. (Organización Mundial de la Salud (OMS), 2012). .....	29
<b>Tabla 4.</b> Requisitos de mantenimiento (siguiendo indicaciones del fabricante o basados en la experiencia). (Organización Mundial de la Salud (OMS), 2012). .....	29
<b>Tabla 5.</b> Antecedentes de problemas del equipo. (Organización Mundial de la Salud (OMS), 2012). .....	29
<b>Tabla 6.</b> Información solicitada en el ítem Proveedores.....	35
<b>Tabla 7.</b> Información solicitada en el ítem Datos .....	35
<b>Tabla 8.</b> Información solicitada en el ítem Inventario .....	36

## LISTADO DE IMÁGENES

<b>Figura 1.</b> Factores clave para la planificación de un programa de mantenimiento. (Organización Mundial de la Salud (OMS), 2012) .....	15
<b>Figura 2.</b> Intérprete de Python (Aula21, s.f.).....	26
<b>Figura 3.</b> Componentes de un programa de mantenimiento (Organización Mundial de la Salud (OMS), 2012) .....	27
<b>Figura 4.</b> Creación del proveedor y contratista de mantenimiento (si lo requiere) .....	37
<b>Figura 5.</b> Información técnica de los equipos y dispositivos médicos dispuestos en la Institución. .....	37
<b>Figura 6.</b> Creación de cada equipo que contiene información única.....	38
<b>Figura 7.</b> Tabla de datos generada en el apartado de Proveedores.....	39
<b>Figura 8.</b> Tabla de datos generada en el apartado de Datos .....	39
<b>Figura 9.</b> Tabla de datos generada en el apartado de Inventario .....	40

## **RESUMEN**

Con este proyecto se diseñó un aplicativo, utilizando el lenguaje de programación Python, para controlar el inventario de dispositivos y equipos biomédicos que se encuentran en una institución de salud, por medio de una amplia base de datos con la información más relevante de estos y que pueden constituir su hoja de vida. Lo anterior con la finalidad de mejorar los tiempos de respuesta y optimizar el programa de mantenimiento preventivo.

Con el aplicativo se logra alimentar la base de datos, inventario y programa de mantenimiento preventivo de manera conjunta. Asimismo, se logran identificar detalles técnicos de los equipos biomédicos, incluso los proveedores de estas tecnologías.

Palabras clave: dispositivos médicos, equipos biomédicos, inventario, mantenimiento.

## **ABSTRACT**

With this project an application was designed, using the Python programming language, to control the inventory of biomedical devices and equipment found in a health institution, by means of a large database with the most relevant information of these and that can constitute its resume. The purpose of this is to improve response times and optimize the preventive maintenance program.

With the application it is possible to feed the database, inventory, and preventive maintenance program together. Likewise, it is possible to identify technical details of biomedical equipment, including the suppliers of these technologies.

Key words: medical devices, biomedical equipment, inventory, maintenance.

## INTRODUCCION

Un programa de mantenimiento para equipos biomédicos lo que busca es que los equipos que le pertenecen sean seguros, eficientes y confiables. El eje principal de este programa es el mantenimiento de los equipos biomédicos asegurando su correcto funcionamiento (mantenimiento preventivo, mantenimiento correctivo y, de ser posible, mantenimiento predictivo) y calibración (en caso de que aplique). Todo esto se hace en búsqueda de garantizar la seguridad de los pacientes y del personal que interviene en el uso de estos.

Cada equipo contiene características que difieren de otros, la variedad de fabricantes e incluso proveedores hacen que el proceso se torne más complejo. Optimizar los tiempos de respuesta en caso de alguna falla, también hacen parte de un programa de mantenimiento organizado. Con la gestión del sistema de mantenimiento se busca ofrecer seguridad y calidad en la prestación de los servicios, economizar el tiempo requerido para los mantenimientos de cada equipo, sistematizar los protocolos/rutinas de mantenimiento y conocer a profundidad la información técnica de estos.

Un buen programa de mantenimiento va muy de la mano del inventario con el que se cuente en la institución. Conociendo la información técnica de los equipos que se tienen, sus frecuencias de uso, cuidados se puede determinar la frecuencia de mantenimiento. También, permite coordinar con los proveedores que sean necesarios las fechas de realización de estos.

Un problema frecuente al interior de una institución de salud sea Clínica u Hospital, es determinar la ubicación exacta de los equipos pues hay ocasiones en que entre servicios se intercambian o prestan, pero no se informa al personal del departamento Biomédico para hacerle el respectivo seguimiento al caso. A largo plazo esto afecta el control interno de los equipos,

afecta al momento de realizar alguna auditoría o al momento de requerirlo para su mantenimiento no se encuentra en el sitio asignado.

Teniendo en cuenta lo anteriormente descrito y que es una problemática muy frecuente, organizar el inventario y programa de mantenimiento se convierte en una necesidad primordial y de ahí nace una herramienta que permita a la institución controlar estos aspectos. Esto puede ser provechoso también para tener presentes las recomendaciones dadas por el fabricante para el cuidado de los equipos. Se pretende el diseño y desarrollo de una aplicación que permita al personal de ingeniería biomédica de la institución conocer la hoja de vida de los dispositivos y equipos que tiene la institución, manejar una amplia base de datos amplia con la información técnica, seguridad eléctrica, riesgos asociados, accesorios, apoyo técnico, entre otros parámetros y, asimismo, incluir el cronograma de mantenimientos preventivos.

### **PLANTEAMIENTO DEL PROBLEMA**

Con frecuencia, al interior de las instituciones de salud se logra evidenciar la falta de control en cuanto al inventario existente. Es muy común ver que entre servicios se comparten los equipos y no se hacen las respectivas devoluciones y/o no se informa al personal del departamento Biomédico para hacerle el respectivo seguimiento al caso. Esto afecta el control interno de los equipos pues se pueden extraviar, dañar o perder la trazabilidad de sus movimientos y por ende no se garantizaría calidad total en la prestación de los servicios.

Otro problema frecuente es el desconocimiento, sea porque no se ha organizado u otras razones, del cronograma de mantenimiento preventivo de los equipos. Tener pleno conocimiento de esta información le permite a la institución ahorrar dinero al evitar un daño futuro en sus equipos y optimizar los tiempos de respuesta en caso de que se requiera un correctivo.

¿Se puede diseñar un aplicativo para controlar el inventario de una institución, además de optimizar el cronograma de mantenimientos?

## **OBJETIVOS**

### **OBJETIVO GENERAL**

Diseñar un aplicativo para control de inventario de los diferentes dispositivos y equipos biomédicos dentro de una institución de salud, por medio de una amplia base de datos, con la finalidad de mejorar tiempos de respuesta y optimizar el programa de mantenimiento preventivo.

### **OBJETIVOS ESPECIFICOS**

- Generar una base de datos completa de dispositivos y equipos biomédicos que permita la implementación del aplicativo.
- Identificar los parámetros e información necesaria para el manejo del inventario.
- Diseñar un aplicativo utilizando el lenguaje de programación Python que permita alimentar la base de datos, inventario y programa de mantenimiento preventivo.

## **JUSTIFICACION**

Ante la necesidad de una mejora continua del proceso de gestión de tecnologías biomédicas, incluyendo el control de inventario y gestión de mantenimiento de equipos biomédicos, se busca garantizar calidad de los servicios prestados por una institución de salud. Por esto, es importante tener la información detallada, organizada y, de algún modo, sintetizada para la hora de la toma de decisiones. Esta toma de decisiones puede influir en los recursos asignados al área biomédica

y a la adquisición de nuevas tecnologías y/o repuestos de las ya existentes mediante la planificación del programa de mantenimiento preventivo.

Por otro lado, también es importante tener en cuenta que con este tipo de organización de la información se puede mejorar la productividad por parte del grupo de biomédicos de la institución. Este proyecto puede constituir una pieza fundamental al interior de una institución de salud pues permitirá conocer con certeza los equipos con los que cuenta, las condiciones de estos y controlar su correcto funcionamiento a través del programa de mantenimiento.

## **MARCOS DE REFERENCIA**

### **MARCO TEORICO**

Los servicios de salud tienden a modernizarse para brindar una atención cada vez mejor a los usuarios, razón por la cual requieren tecnologías más complejas, las cuales exigen ambientes y sistemas de ingeniería apropiados para su instalación y su funcionamiento. A medida que las instituciones de salud incursionan en nuevos procedimientos médicos, exigen tecnologías más complejas o de última generación.

El soporte técnico que requiere toda unidad de salud lo proporciona la ingeniería clínica, que nació como ciencia en 1960, pero alcanzó su mayor desarrollo en 1970, cuando las instituciones de salud empezaron a incorporar de manera masiva las tecnologías médicas complejas, como resultado de las investigaciones médico-científicas, las cuales, a su vez, exigen soportes especializados de los sistemas de ingeniería. Este soporte lo brindan los ingenieros clínicos y los tecnólogos biomédicos. (Granados, 2016).

Las actividades del programa de mantenimiento son el núcleo principal para el mantenimiento de los equipos médicos, así como para asegurar el funcionamiento, la calibración y el mantenimiento preventivo requeridos a la hora de apoyar al personal médico y de ingeniería. El fin es garantizar el funcionamiento seguro y adecuado de los equipos utilizados en la atención del paciente, bien sea de forma directa o indirectamente. (Granados, 2016).

Existen varios elementos que contribuyen a aumentar el acelerado deterioro de las tecnologías y, por tanto, las probabilidades de fallas en los equipos y los dispositivos médicos; entre dichos elementos están la insuficiencia de información técnica y las condiciones ambientales impropias para almacenar o instalar los equipos, que son afectados por la humedad, el polvo, la mala ventilación o las deficiencias en los sistemas de suministros energéticos (electricidad, agua, gases medicinales y otros). (Granados, 2016).

Una buena gestión del mantenimiento de los equipos médicos se logra teniendo en cuenta lo siguiente:

- Establecer el programa de mantenimiento de manera que sea seguro y funcional. Es decir, brindar seguridad con los equipos se están revisando y garantizando calidad en su funcionamiento.
- Proporcionar facilidades en el proceso de gestión. Se busca economizar tiempo en el desarrollo de las actividades del mantenimiento y soportar con documentos avalados por la institución toda la actividad realizada.

El objetivo de toda actividad de mantenimiento es la plena identificación de los aspectos más relevantes durante la inspección del equipo sin ser muy específicos. Con el desarrollo de un programa de mantenimiento preventivo se busca que los equipos, en condiciones normales,

funcionen correctamente, sean seguros y a la larga generen un ahorro significativo en reparaciones para la Institución.

Los criterios para la inclusión dentro del inventario de mantenimiento del equipo médico incluyen un examen de la aplicación clínica, los riesgos físicos y los requisitos de mantenimiento preventivo, entre otros. Los criterios para inventariar un equipo y tomarlo como válido y apto para ser incluido en un sistema de control de activos pueden ser muy diversos, ya que en dicho sistema se incluyen las consideraciones del valor activo, la depreciación y la capacidad de investigar los costos de mantenimiento correctivo, entre otros.

Las actividades de mantenimiento preventivo son planificadas, organizadas y desarrolladas, en gran parte, por personal de la Institución. Las actividades de este mismo mantenimiento, pero para tecnologías complejas serán programadas por la Institución, aunque desarrolladas por personal calificado externo a esta, siguiendo las indicaciones dadas por el fabricante y/o proveedor del equipo.

## MARCO CONCEPTUAL

***Hoja de vida:*** Registro continuo de la información básica y específica de cada acción de mantenimiento, reparación o adecuación realizada a un equipo. Está conformada por la ficha técnica del mismo y se soporta con los informes de mantenimientos. Por medio de este documento se puede determinar la trazabilidad del equipo, estado de funcionalidad, obsolescencia y demás. (Muñoz Salazar, 2008).

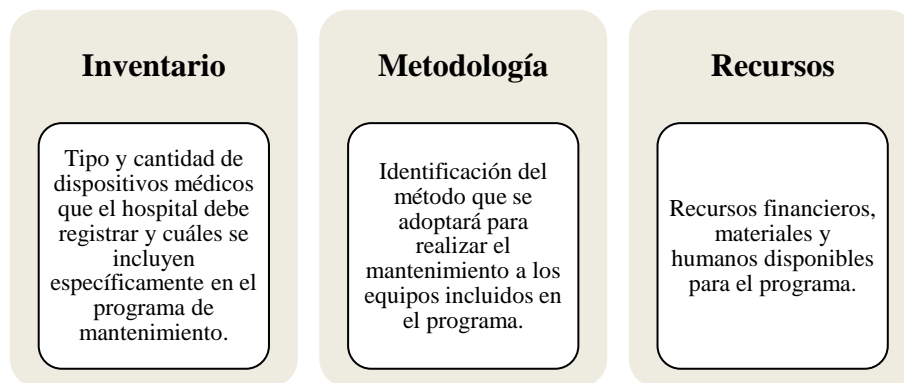
***Inventario:*** La experiencia demuestra que el inventario de los equipos incluidos en los programas para el mantenimiento se debe limitar a los equipos más significativos; de lo contrario, se hace inmanejable o ineficiente. Se recomienda priorizar el mantenimiento de los equipos basándose en criterios de riesgo. (Granados, 2016).

Por ejemplo, los dispositivos manuales para medir la presión arterial tienen solo algunos componentes que son fáciles de reparar si se dispone de los elementos necesarios para ello. Los dispositivos de imagenología o de laboratorio se sitúan en el lado opuesto. La reparación de un sistema de resonancia magnética involucra gran cantidad de recursos financieros, materiales y humanos. Entre estos extremos están las bombas de infusión, los desfibriladores, los electrocardiógrafos, y cientos de otros dispositivos de complejidad variable. En las fases iniciales del proceso de planificación de un programa de mantenimiento, es esencial determinar qué tipos de dispositivos se deben incluir en el programa. (Organización Mundial de la Salud (OMS), 2012).

Se considera que el área de ingeniería biomédica debe identificar y seleccionar los equipos y dispositivos que se incluirán en el inventario de la institución. Del mismo modo, identifican

cuales de esos equipos deben estar incluidos en el programa de mantenimiento y la modalidad de este.

**Figura 1.** Factores clave para la planificación de un programa de mantenimiento. (Organización Mundial de la Salud (OMS), 2012)



Periódicamente, se debe verificar la existencia de los equipos dispuestos en la institución de tal manera que el inventario se mantenga actualizado con la cantidad de equipos correspondiente, ubicaciones y estado de estos. Esta tarea se puede realizar previo a las inspecciones previas al mantenimiento preventivo. Es importante tener presente que, con la llegada de una tecnología nueva a la institución, ésta también debe ser incluida en el inventario. De acuerdo con la necesidad de la institución, se identifica qué información debe ser incluida en el inventario sobre cada artículo.

**Tabla 1.** Datos de inventario. (Organización Mundial de la Salud (OMS), 2012)

<b>Elemento</b>	<b>Finalidad</b>	<b>Tipo de inventario</b>
<b>Número de identificación de inventario</b>	Identificador único para cada equipo	Equipos médicos

<b>Tipo de equipo/artículo</b>	Informa sobre la naturaleza del artículo, utilizando una nomenclatura uniforme y estándar. Por ejemplo, el Universal Medical Device Nomenclature System (sistema universal de nomenclatura de dispositivos médicos, UMDNS) o la Global Medical Device Nomenclature (nomenclatura mundial de dispositivos médicos, GMDN)	Todos
<b>Breve descripción del equipo/artículo</b>	Describe el artículo y su función o finalidad	Todos
<b>Fabricante</b>	Identifica la empresa que fabrica el artículo, indicando su nombre, dirección y datos de contacto	Todos
<b>Modelo/número de catálogo</b>	Identificador único de la línea de productos (asignado por el fabricante)	Todos
<b>Número de serie</b>	Identificador único del artículo (asignado por el fabricante)	Todos
<b>Ubicación física en el centro de atención sanitaria</b>	Incluye el número de habitación, o área; permite localizar el equipo médico cuando deba someterse a mantenimiento preventivo; puede incluir información sobre el lugar de almacenamiento del material fungible y las refacciones.	Todos

<b>Estado/estatus operativo</b>	Indica si el equipo está “en servicio” o “fuera de servicio”, y en el segundo caso indica el motivo: por ejemplo, pendiente de calibración o de mantenimiento preventivo, en reparación, en espera de refacciones o dañado sin posibilidad de reparación.	Equipos médicos, equipos de medición
<b>Alimentación eléctrica</b>	Indica la alimentación eléctrica necesaria para el equipo. Por ejemplo, de 110 V, 220 V, 380 V o trifásica; puede ser útil para señalar los equipos que necesiten transformadores u otras medidas especiales.	Equipos médicos, equipos de medición
<b>Requisitos de funcionamiento y mantenimiento</b>	Describe cualquier requisito especial necesario para el funcionamiento o el mantenimiento del equipo.	Equipos médicos
<b>Fecha inicial de registro en el inventario y de actualización</b>	Fecha en la que se registró el equipo en el inventario y fecha de la actualización más reciente de la información	Todos
<b>Proveedor del servicio de mantenimiento</b>	Proporciona información sobre el proveedor incluyendo el nombre, contacto y, en caso de contar con una empresa o taller externo para el mantenimiento del equipo, incluir detalles del contrato (incluyendo vigencia de la garantía);	Equipos médicos, equipos de medición

	información que indique fecha en que se realizó el mantenimiento.	
<b>Proveedor de compra</b>	Se usa como contacto para compras, pedidos de reposición, sustituciones en garantía, etc.	Todos
<b>Número de lote</b>	Suele asignarse al material fungible o los reactivos fabricados en un mismo lote o partida; puede ayudar a detectar defectos; útil para los sistemas de control de existencias de material fungible.	Material fungible
<b>Números de la versión actual del software y firmware</b>	Usados para equipos que funcionan con programas informáticos (software) o circuitos electrónicos (firmware); pueden facilitar la detección de problemas asociados al software o el firmware.	Equipos médicos, equipos de medición
<b>Datos del departamento propietario</b>	Designa el contacto para notificar retrasos en el servicio técnico y programar el mantenimiento preventivo.	Equipos médicos
<b>Costo de compra</b>	Dato de interés para el cálculo de los valores de inventario de activos fijos y para fines presupuestarios.	Todos
<b>Fecha de compra</b>	En el caso de activos fijos, se usa para calcular los valores de depreciación o determinar la sustitución/obsolescencia En el caso de material	Todos

	fungible o refacciones, puede emplearse para determinar las tasas de uso, las necesidades de pedidos de reposición y las fechas de caducidad	
<b>Fecha de vencimiento de la garantía</b>	Útil para el seguimiento de la validez y vencimiento de las garantías.	Todos
<b>Fecha de instalación e información y resultados sobre las pruebas de aceptación</b>	Sirve como base para la documentación del historial de mantenimiento y se usa como referencia cuando deba solucionarse un problema.	Equipos médicos, equipos de medición
<b>Evaluación/ clasificación en materia de seguridad y riesgos</b>	Incluye la evaluación de riesgos realizada (u otra justificación en caso necesario) que determinó la inclusión del equipo en el inventario; también puede servir para determinar la prioridad asignada al equipo en lo que respecta a su reparación y pruebas.	Equipos médicos
<b>Calendario y procedimientos de mantenimiento preventivo</b>	Resume la frecuencia del mantenimiento preventivo, así como los procedimientos de mantenimiento.	Equipos médicos, equipos de medición
<b>Fecha y resultado de las calibraciones realizadas; fecha de las próximas</b>	Sirve como referencia al solucionar problemas del equipo y permite asegurarse de que se cumplen los plazos de calibración	Equipos médicos, equipos de medición

<b>calibraciones necesarias y procedimientos</b>		
<b>Cantidad de refacciones</b>	Cuando se usa en sistemas de control de existencias, sirve como indicador para realizar un pedido de reposición cuando la cantidad en existencias alcanza un nivel determinado.	Refacciones, material fungible
<b>Dispositivos, sistemas, accesorios, material fungible o refacciones asociada</b>	Señala equipos complementarios importantes, entre ellos cualquier dispositivo o accesorio necesario para el funcionamiento de un equipo.  Resulta útil disponer de los números de catálogo de los accesorios, las refacciones y el material fungible.	Equipos médicos, equipos de medición
<b>Año de fabricación</b>	Se emplea para calcular la antigüedad del equipo; en combinación con la vida útil prevista del equipo, sirve para determinar cuándo debe ser sustituido, retirado o eliminado	Equipos médicos, equipos de medición
<b>Vida útil prevista del equipo</b>	Indica el tiempo esperado (típicamente, expresado en años) durante el cual el equipo puede funcionar de forma segura y eficaz; puede usarse como dato para determinar cuándo debe ser sustituido, retirado o eliminado.	Todos

<p><b>Historial de funcionamiento y mantenimiento</b></p>	<p>Puede incluir cuadernos de uso o de mantenimiento (para el funcionamiento o el servicio técnico), órdenes de trabajo o informes de servicio técnico, informes de mantenimiento preventivo y otra información sobre el funcionamiento y mantenimiento del equipo; puede usarse en el diagnóstico de averías, para evaluar posibles compras de equipos nuevos similares y para determinar cuándo debe ser sustituido, retirado o eliminado un artículo.</p>	<p>Equipos médicos, equipos de medición</p>
<p><b>Historial de órdenes de retirada y peligros notificados</b></p>	<p>Se usa para detectar y controlar posibles peligros asociados al uso del aparato.</p>	<p>Equipos médicos, equipos de medición</p>
<p><b>Cualquier otra información que se desee</b></p>	<p>Un inventario solo es útil para un centro de atención sanitaria si contiene la información importante que necesita el centro; por consiguiente, pueden añadirse todos los campos de datos que se consideren necesarios</p>	<p>Todos</p>

**Lenguaje de programación:** Python se lanzó en 1991 y fue creado por Guido Van Rossum. Es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). El software se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo (aws amazon, s.f.). Python además contiene varias librerías las cuales contienen funciones incorporadas en el lenguaje que simplifica la programación. Un programa realizado en Python puede constar de hasta 5 líneas menos que otros programas como Java y es multiplataforma; adicionalmente, Python es de libre uso, es decir, es gratuito incluso para las empresas que deseen adquirirlo.

**Características** (Campodónico Chávez, 2019):

- Ofrece soporte para una gran variedad de Base de Datos.
- Existen muchas librerías y funciones preestablecidas que se pueden emplear en cualquier documento y ahorran mucho tiempo.
- Es fácil de aprender.
- Es multiparadigma, es decir que puede soportar gran variedad de paradigmas de programación como: estructurada, imperativa, orientada a objetos y funcional.
- Es multiplataforma, es compatible con Linux, Unix, Mac Os y Windows.
- Es un Software libre (2018).

**¿Cómo se utiliza Python?** (aws amazon, s.f.).

El lenguaje Python se aplica a varios casos de uso en el desarrollo de aplicaciones.

- **Desarrollo web del lado del servidor:** El desarrollo web del lado del servidor incluye las funciones complejas de *backend* que los sitios web llevan a cabo para mostrar información al usuario. Por ejemplo, los sitios web deben interactuar con las

bases de datos, comunicarse con otros sitios web y proteger los datos cuando se los envía a través de la red. Los desarrolladores también utilizan un amplio rango de marcos de Python que proporcionan todas las herramientas necesarias para crear aplicaciones web con mayor rapidez y facilidad.

- **Realizar tareas de ciencia de datos y machine learning:** Consiste en extraer conocimientos valiosos a partir de los datos, mientras que el enseña a las computadoras a aprender automáticamente de los datos y a efectuar predicciones precisas.
  - Corregir y eliminar datos incorrectos, lo que se conoce como limpieza de datos.
  - Extraer y seleccionar varias características de los datos.
  - Buscar diferentes estadísticas a partir de los datos.
  - Visualizar los datos mediante el uso de tablas y gráficos, como los gráficos de líneas, los de barras, los circulares y los histogramas.

### **Bibliotecas de Python** (aws amazon, s.f.).

Una biblioteca es una colección de códigos usados con frecuencia que los desarrolladores pueden incluir en sus programas de Python para evitar tener que escribir el código desde cero. De forma predeterminada, Python incluye la biblioteca estándar, que contiene una gran cantidad de funciones reutilizables. Además, más de 137 000 bibliotecas de Python están disponibles para diversas aplicaciones, incluidos el desarrollo web, la ciencia de datos y el machine learning (ML).

- **Pandas:** Proporciona estructuras de datos optimizadas y flexibles que se pueden utilizar para manipular datos de serie temporal y datos estructurados, como las tablas

y las matrices. Por ejemplo, puede utilizar Pandas para leer, escribir, combinar, filtrar y agrupar datos. Muchas personas lo utilizan para las tareas de ciencia de datos, análisis de datos y ML.

- **XlsxWriter** (PyPI - Python Software Foundation, 2022): es un módulo de Python para escribir archivos en el formato de archivo Excel 2007+ XLSX. XlsxWriter se puede usar para escribir texto, números, fórmulas e hipervínculos a varias hojas de trabajo y admite funciones como formato y muchas más, que incluyen:
  - Archivos Excel XLSX 100% compatibles.
  - Formateo completo.
  - Celdas combinadas.
  - Nombres definidos.
  - Gráficos.
  - Autofiltros.
  - Validación de datos y listas desplegables.
  - Formato condicional.
  - Imágenes PNG/JPEG/GIF/BMP/WMF/EMF de la hoja de trabajo.
  - Cadenas multiformato ricas.
  - Comentarios de celda.
  - Integración con Pandas.
  - Cajas de texto.
  - Soporte para agregar Macros.
  - Modo de optimización de memoria para escribir archivos grandes.

### **Marcos de Python** (aws amazon, s.f.).

Un marco de Python es una colección de paquetes y módulos. Un módulo es un conjunto de código relacionado, y un paquete es un conjunto de módulos. Los desarrolladores pueden usar los marcos de Python para crear aplicaciones de Python más rápido debido a que no tienen que preocuparse por los detalles de nivel inferior, como la forma en que se producen las

comunicaciones en la aplicación web o el modo en que Python hará que el programa sea más rápido. Python tiene dos tipos de marcos:

- El marco de pila completa incluye casi todo lo que se necesita para crear una aplicación grande.
- El micromarco es un marco básico que proporciona funcionalidades mínimas para crear aplicaciones de Python simples. También proporciona extensiones si las aplicaciones necesitan funciones más sofisticadas.

Los desarrolladores pueden utilizar varios marcos de Python para que su desarrollo sea eficiente. El utilizado en el presente trabajo fue:

- Django: Es uno de los marcos web de Python de pila completa más utilizados para el desarrollo de aplicaciones web a gran escala. Proporciona varias características útiles, incluidos un servidor web para el desarrollo y las pruebas, un motor de plantillas para crear el sitio web de *frontend* y diversos mecanismos de seguridad.

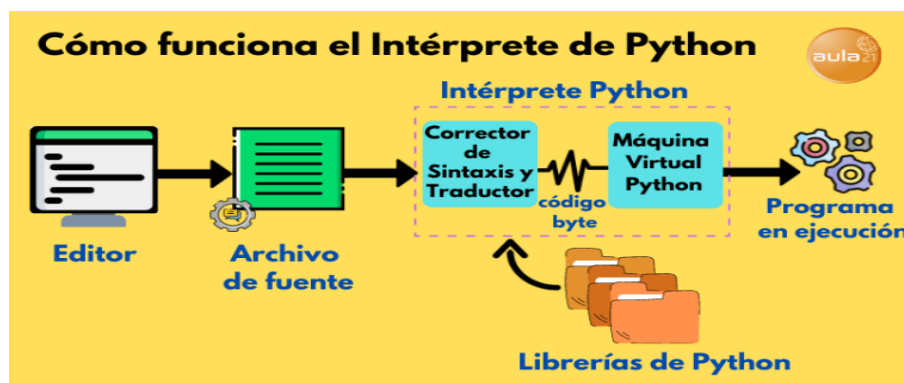
### **¿Cómo funciona Python?** (Aula21, s.f.).

El lenguaje de programación Python utiliza módulos de código que son intercambiables en lugar de una larga lista de instrucciones que era estándar para los lenguajes de programación funcional. La implementación estándar de Python se llama «cpython». En definitiva, no convierte su código en lenguaje de máquina o código máquina, algo que el hardware puede entender. En realidad, lo convierte en algo llamado código de byte. Este código de bytes no puede ser entendido por la CPU. Así que necesitamos un intérprete llamado Máquina Virtual Python (PVM) que ejecuta los códigos de bytes.

El intérprete de Python realiza las siguientes tareas para ejecutar un programa:

1. El intérprete lee un código o instrucción Python. Luego verifica que la instrucción esté bien formateada, es decir, comprueba la sintaxis de cada línea. Si encuentra algún error, detiene inmediatamente la traducción y muestra un mensaje de error.
2. Si no hay ningún error, es decir, si la instrucción o el código Python está bien formateado, el intérprete lo traduce a su forma equivalente en un lenguaje intermedio llamado «código Byte». Así, después de la ejecución exitosa de la escritura o el código Python, se traduce completamente en código Byte.
3. El código del byte se envía a la Máquina Virtual Python, donde de nuevo se ejecuta el código del byte en PVM. Si se produce un error durante esta ejecución, ésta se detiene con un mensaje de error.

**Figura 2.** Intérprete de Python (Aula21, s.f.)

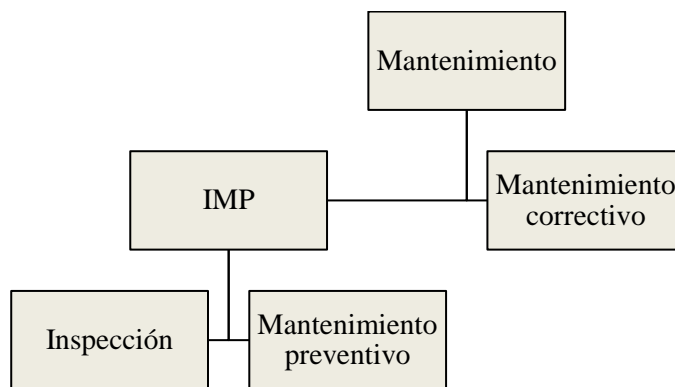


**Mantenimiento:** es el procedimiento por el cual se trata un bien determinado de manera que el paso del tiempo, el uso o el cambio de circunstancias externas no lo afecten. En áreas críticas donde los equipos deben estar siempre operativos, existen protocolos y un alto grado de sofisticación en las tareas de mantenimiento. (Todo Mantenimiento, s.f.).

Las actividades de mantenimiento han de ser programadas por la institución y dependiendo el nivel de complejidad de la tecnología biomédica en cuestión serán desarrolladas por personal

propio de esta o por terceros expertos en el área. En ambos casos, el mantenimiento debe ser teniendo en cuenta las recomendaciones dadas por el fabricante de la tecnología.

**Figura 3.** Componentes de un programa de mantenimiento (Organización Mundial de la Salud (OMS), 2012)



**Mantenimiento correctivo:** Busca la corrección de averías o fallas que presentan los equipos. Este mantenimiento no es predecible, por tanto, no es planificado. (Todo Mantenimiento, s.f.).

El mantenimiento correctivo (MC) restituye la función de un dispositivo averiado y permite ponerlo nuevamente en servicio. Las tareas de mantenimiento correctivo no se programan y aumentarán o disminuirán según la demanda. Por lo tanto, es importante contar con un esquema de prioridades de modo que los recursos para el MC se destinen a cubrir las necesidades más apremiantes. (Organización Mundial de la Salud (OMS), 2012).

**Mantenimiento preventivo:** Tiene como objetivo garantizar la fiabilidad en el buen funcionamiento del equipo y así evitar que se presente alguna avería o deterioro de este. Es planificado. (Todo Mantenimiento, s.f.).

Regularmente, los equipos incluidos en el programa de mantenimiento son equipos de apoyo vital, de laboratorio, de cirugía, de cuidados intensivos, de imagenología, equipos que si llegasen a fallar podrían provocar lesiones o muerte, equipos cuyo mantenimiento será realizado por

personal externo a la institución. Los procedimientos de Inspección y Mantenimiento Preventivo correctos y apropiados marcan la diferencia y son los que garantizan equipos fiables y que funcionan adecuadamente. Los procedimientos se deben definir antes de realizar los trabajos de inspección o de mantenimiento, examinando cuidadosamente cada tipo de equipo (o modelo). (Organización Mundial de la Salud (OMS), 2012).

Siguiendo la orientación dada por la OMS en cuanto a la gestión de equipos biomédicos basados, se establecen algunos criterios de inclusión para evaluar los equipos dispuestos en la institución. Se tiene en cuenta el modelo de Fennigkon y Smith que asigna valores numéricos a cada tipo de dispositivo de acuerdo con su función, aplicación y requisitos de mantenimiento.

$$\text{Número de gestión del equipo} = \# \text{ función} + \# \text{ aplicación} + \# \text{ mantenimiento} + \# \text{ antecedentes}$$

**Tabla 2.** Función del equipo. (Organización Mundial de la Salud (OMS), 2012).

<b>Categoría</b>	<b>Descripción de la función</b>	<b>Puntuación</b>
<b>Terapéutico</b>	Apoyo vital	10
	Cirugía y cuidados intensivos	9
	Fisioterapia	8
<b>Diagnóstico</b>	Control de cirugía y cuidados intensivos	7
	Control fisiológico adicional y diagnóstico	6
<b>Analítico</b>	Análisis del laboratorio	5
	Accesorios del laboratorio	4
	Computadoras y afines	3
<b>Otros</b>	Relacionados con el paciente y otros	2

**Tabla 3.** Riesgo físico asociado con la aplicación clínica. (Organización Mundial de la Salud (OMS), 2012).

<b>Descripción del riesgo durante el uso</b>	<b>Puntuación</b>
Riesgo de muerte del paciente	5
Posible lesión del paciente o el operador	4
Tratamiento inapropiado o error de diagnóstico	3
Daño al equipo	2
Sin riesgo significativo identificado	1

**Tabla 4.** Requisitos de mantenimiento (siguiendo indicaciones del fabricante o basados en la experiencia). (Organización Mundial de la Salud (OMS), 2012).

<b>Requisitos de mantenimiento</b>	<b>Puntuación</b>
Importantes: exige calibración y reemplazo de piezas periódicas	5
Superiores al promedio	4
Usuales: verificación de funcionamiento y pruebas de seguridad	3
Inferiores al promedio	2
Mínimo: inspección visual	1

**Tabla 5.** Antecedentes de problemas del equipo. (Organización Mundial de la Salud (OMS), 2012).

<b>Promedio de averías del equipo</b>	<b>Factor</b>
Significativo: más de una cada seis meses	+2
Moderado: una cada 6-9 meses	+1
Usual: una cada 9-18 meses	0
Mínimo: una cada 18-30 meses	-1
Insignificante: menos de una en los 30 meses anteriores	-2

**Frecuencia del mantenimiento.** (Organización Mundial de la Salud (OMS), 2012)

Los valores correspondientes a los requisitos de mantenimiento también sirven para determinar el intervalo entre inspecciones y procedimientos de mantenimiento, según el tipo de dispositivo.

- Para todos los dispositivos con requisitos importantes de mantenimiento en la clasificación (valor característico de 4 o 5), se programarán tareas de mantenimiento preventivo cada seis meses.
- Para los dispositivos con requisitos usuales o mínimos de mantenimiento (valores de 3, 2 o 1) se programarán tareas de mantenimiento preventivo anuales.
- Para los dispositivos con un valor de gestión de equipo de 15 o más se programarán inspecciones por lo menos cada seis meses.
- Para los dispositivos con un valor de gestión de equipo de 19 o 20 se programarán inspecciones cada cuatro meses.

**MARCO LEGAL**

*Decreto 4725 de 2005.* Ministerio de la Protección Social. Por el cual se reglamenta el régimen de registros sanitarios, permiso de comercialización y vigilancia sanitaria de los dispositivos médicos para uso humano.

**Artículo 1º. Objeto y ámbito de aplicación.** El presente decreto tiene por objeto, regular el régimen de registros sanitarios, permiso de comercialización y vigilancia sanitaria en lo relacionado con la producción, procesamiento, envase, empaque, almacenamiento, expendio, uso, importación, exportación, comercialización y mantenimiento de los dispositivos médicos para uso humano, los cuales serán de obligatorio cumplimiento por parte de todas las personas

naturales o jurídicas que se dediquen a dichas actividades en el territorio nacional.

(MINISTERIO DE PROTECCION SOCIAL, 2005)

**Resolución 2981 de 2011.** Ministerio de la Protección Social. Por la cual se dictan disposiciones relacionadas con la codificación estandarizada de insumos y dispositivos médicos.

**Artículo 3°. Objetivos de la codificación estandarizada.** La codificación estandarizada de insumos y dispositivos médicos tiene como propósito fundamental la identificación y clasificación de los insumos y dispositivos médicos comercializados y usados en el territorio nacional y como objetivos específicos, los siguientes:

2. Facilitar el manejo, suministro, adquisición y uso de los insumos y dispositivos médicos para uso humano en los procesos de prestación de servicios de salud, actividades de promoción y prevención, y el inventario y mantenimiento de los equipos biomédicos. (MINISTERIO DE LA PROTECCION SOCIAL, 2011)

**Resolución 3100 de 2019.** Ministerio de Salud y Protección Social. Por la cual se definen los procedimientos y condiciones de inscripción de los prestadores de servicios de salud y de habilitación de los servicios de salud y se adopta el Manual de Inscripción de Prestadores y Habilitación de Servicios de Salud.

## **11. Estándares y criterios de habilitación**

### **11. 1. 3. Estándar de dotación**

2.1. Programa de mantenimiento preventivo de los equipos biomédicos, que incluya el cumplimiento de las recomendaciones establecidas por el fabricante o de acuerdo con el

protocolo de mantenimiento que tenga definido el prestador, éste último cuando no esté definido por el fabricante.

2.2. Hoja(s) de vida del(los) equipo(s) biomédicos(s), con los registros de los mantenimientos preventivos y correctivos, según corresponda. (MINISTERIO DE SALUD Y PROTECCION SOCIAL, 2019).

## **METODOLOGIA**

### **DISEÑO**

La recolección de datos es de los métodos más relevantes al momento de iniciar un proceso investigativo. Con el transcurrir del proceso se pueden presentar situaciones con diferentes requerimientos y por ende variabilidad en los resultados. La metodología no solo se basa en teoría sino, también, en encontrarle sentido y justificación por medio de la comprobación y del análisis de datos (Strauss & Corbin, 2002).

La metodología de investigación será de tipo mixto, esto debido a que al final del proyecto las conclusiones podrían plantearse de forma cualitativa y cuantitativa pues no se puede limitar la información que se contendrá en el aplicativo. Del mismo modo, la investigación será de carácter experimental pues se requiere de la comprobación para verificar el funcionamiento del aplicativo. Esta investigación como no está definida para una institución en particular, pero puede constituir la base de una, no tendrá un margen temporal para desarrollarla.

## **TIPO DE INVESTIGACIÓN**

La investigación es de tipo exploratoria y descriptiva ya que a partir de la información y datos recolectados se pueden describir los pasos que permiten generar una completa base de datos de dispositivos y equipos biomédicos y con ello la implementación del aplicativo. También a través de la descripción, se identifican los parámetros e información necesaria para el manejo del inventario y, finalmente detallar el paso a paso para el diseño de un aplicativo utilizando el lenguaje de programación Python que permita alimentar la base de datos, inventario y programa de mantenimiento preventivo.

## **PROCEDIMIENTO**

### **RECOLECCIÓN DE DATOS PARA EL INVENTARIO**

Para poder crear un inventario completo se requiere la creación de una base de datos robusta. Se deben tener en cuenta los datos técnicos y funcionales de los equipos y dispositivos médicos. Estos datos permiten la plena identificación de cada equipo y se deben tener en cuenta los cambios y/o modificaciones pues estas cualidades pueden ser modificadas de acuerdo con las necesidades y frecuencias de actividad. (Granados, 2016).

Después de realizar una extensiva revisión bibliográfica, se sugiere que la información que contenga el inventario sea:

- Nombre del equipo.
- Marca.
- Modelo.
- Serie.
- Código de activo fijo.
- Tipo de equipo: fijo – móvil.
- Clasificación biomédica.
- Registro Sanitario.

- Clasificación de riesgo (Res 4725 de 2005)
- Vida útil.
- Fecha de adquisición.
- Fecha de instalación.
- Accesorios.
- Riesgos y seguridad eléctrica: tipo de protección contra descargas eléctricas, grado de protección contra descargas eléctricas.
- Ubicación: sede, servicio, ubicación específica.
- Fabricante
- Proveedor: nombre, dirección, teléfono, correo, ciudad, personal de mantenimiento.
- Proveedor de mantenimiento (si lo requiera).
- Datos técnicos: fuente de alimentación, voltaje máximo y mínimo, corriente máxima y mínima, potencia, frecuencia, velocidad, condiciones ambientales de operación.
- Frecuencia de mantenimiento.
- Cronograma de mantenimiento.
- Cronograma de calibración (si lo requiere).

## **DISEÑO DEL SOFTWARE DE INVENTARIO Y MANTENIMIENTO**

Para el cumplimiento de este proyecto se diseñó un software que contenga el inventario y, por ende, el mantenimiento de los equipos médicos. Se utilizó el lenguaje de programación Python por su facilidad en el lenguaje, visualmente amigable con la interfaz a proyectar y por ser de acceso gratuito. En el Apéndice A se muestra el código creado para este aplicativo.

El aplicativo consta de tres partes muy importantes:

- Proveedores: En este ítem se requiere de información relevante del proveedor que suministra el equipo biomédico. En caso de que el mantenimiento sea realizado por un externo a la institución también se puede identificar en este ítem. Por tanto, consta de dos partes para completar la información.

**Tabla 6.** Información solicitada en el ítem Proveedores

<b>Datos del Proveedor</b>	<b>Datos del personal de mantenimiento</b>
Nombre del proveedor	Proveedor de mantenimiento
Teléfono	Nombre del personal de mantenimiento
Ciudad	Teléfono
Correo	Ciudad
Dirección	Correo
Fax	Dirección
Nombre de fabricante	Fax
País	

- Datos: Después de crear el Proveedor, se procede a completar este ítem. Ahí se recopila la información técnica de los equipos.

**Tabla 7.** Información solicitada en el ítem Datos

Equipo	Marca
Modelo	Registro sanitario
Clasificación de riesgo (Res 4725 de 2005)	Vida útil
Verificación metrológica	Calibración metrológica
Clasificación biomédica	Fuente de alimentación

Tensión máxima	Tensión mínima
Corriente máxima	Corriente mínima
Potencia	Presión
Frecuencia	Temperatura
Humedad	Peso
Velocidad	Tiempo de mantenimiento

- Inventario: Una vez creado el equipo, se procede al ítem de Inventario en el cual se darán datos específicos del equipo creado.

**Tabla 8.** Información solicitada en el ítem Inventario

Empresa	Sede
Servicio	Ubicación específica
Serie	Código de inventario
Modalidad de mantenimiento	Año último mantenimiento
Periodicidad mantenimiento	Cronograma de mantenimiento preventivo
Costo adquisición	Forma de adquisición
Fecha adquisición	Fecha instalación
Vencimiento garantía	Accesorios

## RESULTADOS

La interfaz del software cuenta con tres segmentos importantes: Proveedores, Datos e Inventario. Cada uno corresponde a una pestaña individual para validar la información que ahí se contiene.

Para poder disponer de los equipos en el Inventario es necesario crear primero el Proveedor que lo surte, posterior se pasa a Datos en la que se proyecta toda la información técnica del equipo y por último se llega a la pestaña de Inventario que contiene información única de los equipos y a su vez permite ver los mantenimientos de esto con el total de equipos disponibles. Una vez completada la información en los segmentos de Proveedor, Datos e Inventario, se generarán unas tablas que muestran el contenido anteriormente descrito.

**Figura 4.** Creación del proveedor y contratista de mantenimiento (si lo requiere)

**Agregar Proveedor**

**Datos del Proveedor**

\*Nombre  Telefono  Ciudad  Correo  Dirección

Fax

**Datos del Personal de mantenimiento**

\*Proveedor mantenimiento  Nombre  Telefono  Ciudad  Correo

Dirección  Fax

**AGREGAR**

PROYECTO DE GRADO  
Copyright © 2022  
Design by Angelly R

**Figura 5.** Información técnica de los equipos y dispositivos médicos dispuestos en la Institución.

**Agregar dato**

\*Equipo  \*Marca  \*Modelo  \*INVIMA  \*Riesgo

\*Vida útil  \*Requiere verificación metrologica  \*Periodicidad  \*Requiere calibración metrologica  \*Periodicidad

\*Clasificación Biomedica  \*Manuales  \*Alimentación  \*Presión  \*Temperatura

\*Humedad  \*Corriente Máxima  \*Frecuencia  \*Velocidad  \*Tensión Máxima

\*Corriente Mínima  \*Peso  \*Tensión Mínima  \*Potencia  \*Plan Metrologico

\*Equipo Critico  \*Equipo predominante  Nombre Fabricante  País  \*Guías rápidas:

\*Tiempo mantenimiento  \*Proveedor

**AGREGAR**

PROYECTO DE GRADO  
Copyright © 2022  
Design by Angelly R

**Figura 6.** Creación de cada equipo que contiene información única.

The screenshot shows a web form titled "Agregar al inventario" with a purple header. The form is organized into several sections:

- Header:** "Agregar al inventario" with sub-headers "Inventario", "Datos", and "Proveedores".
- Form Fields:**
  - Empresa, Sede, Servicio, Ubicación Específica, Equipo (dropdown)
  - Marca, Modelo, Serie, Numero de inventario, Modalidad de mantenimiento (dropdown)
  - Valor mantenimiento, Año último mantenimiento, Periodicidad Mantenimiento (dropdown), Plan aseguramiento metrológico (dropdown), Costo de adquisición
  - Forma de adquisición (dropdown), Fecha de adquisición (calendar), Fecha de instalación (calendar), Tiempo de garantía, Vencimiento de garantía (calendar)
- ACCESORIOS Table:**

NOMBRE	MARCA	MODELO	SERIE
- MANTENIMIENTOS Calendar:**

ENE	FEB	MAR	ABR	MAY	JUN	JUL	AGO	SEP	OCT	NOV	DIC
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
- Buttons:** "AGREGAR" (purple) and "PRODUCTO DE GRADO" (small text at the bottom).

## DISCUSION (ANALISIS DE RESULTADOS)

Con este escrito se logra evidenciar el desarrollo de un aplicativo enfocado en el manejo y control de inventario de dispositivos y equipos biomédicos al interior de una institución de salud. La finalidad de un inventario es que el ente tenga pleno conocimiento de los equipos con los que cuenta, el estado de estos, ubicación, entre otros. Del mismo modo, se pretende tener cierto control al identificar el proveedor de la tecnología y los datos técnicos de esta. Por ello fue necesario manejar tres apartados para el desarrollo.

En el Apéndice A se logra ver el código desarrollado con el lenguaje de programación Python. Cuenta con 696 líneas de código que compilan todo el programa ejecutado. Para cada apartado también se tiene un código que realiza la función asignada, es decir la recopilación de los datos que posteriormente se visualizan en una tabla cuyas columnas constituyen la información solicitada. Para los tres apartados dispuestos en este aplicativo se desarrolla una tabla de datos.

Figura 7. Tabla de datos generada en el apartado de Proveedores

Inventario    Datos    Proveedores

Agregar    Exportar   

PROVEEDOR						PROVEEDORES					CONTRATISTA	
NOMBRE	TELEFONO	CIUDAD	CORREO	DIRECCION	FAX	NOMBRE DEL FABRICANTE	PAIS	PROVEEDOR DE MANTENIMIENTO	NOMBRE DEL PERSONAL	TELEF		
0												

PROYECTO DE GRADO  
Copyright © 2022  
Design by Angelly R

Figura 8. Tabla de datos generada en el apartado de Datos

Inventario    Datos    Proveedores

Agregar    Exportar   

OPCIONES	NOMBRE	REGISTRO INVIMA	RIESGO	VIDA UTIL	REQUIERE VERIFICACIÓN METROLOGICA	PERIODICIDAD DE VERIFICACIÓN	REQUIERE CALIBRACIÓN METROLOGICA	PERIODICIDAD DE CALIBRACIÓN	CLASIFICACION BIOMEDICA	MANUALES	FUE ALIMI
Total	0										

PROYECTO DE GRADO  
Copyright © 2022  
Design by Angelly R

**Figura 9.** Tabla de datos generada en el apartado de Inventario

OPCIONES	CÓDIGO	EMPRESA	SEDE	ÁREA	UBICACIÓN ESPECÍFICA	EQUIPO	MARCA	MODELO	SERIE	# INVENTARIO	MODALIDAD MANTENIMIENTO	VALOR
Total	0											

PROYECTO DE GRADO  
Copyright © 2022  
Design by Angelly R

Al ser extenso el contenido de las tablas no se logran apreciar, en las imágenes, todos los apartados de esta, pero la información que ahí se muestra será la solicitada al ingresar al botón *Agregar*. También se tiene la opción de filtrar por área, ubicación, nombre e, incluso, marca de los equipos. Si se desea estas tablas pueden ser exportadas y serán descargadas en formato .xls que es el compatible para hojas de cálculo como Excel.

## CONCLUSIONES Y RECOMENDACIONES

Teniendo en cuenta que este aplicativo puede utilizarse para controlar inventario, con una base de datos robusta en cuanto a la cantidad de información que almacena, y, posteriormente, permite el control del programa de mantenimiento. Puede constituir una herramienta importante, e indispensable, para un departamento de ingeniería biomédica.

La finalidad de la creación de este aplicativo es que sea de fácil manejo, amigable visualmente y cuyo contenido sea de fácil acceso. Un aspecto para tener en cuenta es que se facilite el acceso a la información allí contenida y que esta sea entendible a cualquier persona del

departamento. Con todos estos datos se puede realizar una caracterización completa y adecuada de los equipos permitiendo que su búsqueda sea más rápida y eficaz, además se pueden identificar las necesidades de la entidad en cuanto a renovación o reposición de tecnología.

Con el cronograma de mantenimiento, que se puede observar desde el apartado de Inventario, se busca que se controlen los tiempos de respuesta en cuanto a los preventivos. Del mismo modo, es importante para el personal de la institución en que sea aplicado que se conozcan las fechas (meses) en que estén programados los mantenimientos para tener en cuenta la disponibilidad y estado de los equipos.

## REFERENCIAS

- Aula21. (s.f.). *AULA21*. Obtenido de <https://www.cursosaula21.com/que-es-python/>
- aws amazon. (s.f.). *Amazon Web Services, Inc.* Obtenido de <https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes.>
- Campodónico Chávez, S. (2019). *Implementación de un Sistema Web de Gestión de Inventario Interno de Activos, aplicado a la empresa Open Office S.A.* Trabajo de grado, Universidad Católica de Santiago de Guayaquil, Facultad de Ingeniería, Guayaquil. Obtenido de <http://201.159.223.180/bitstream/3317/13037/1/T-UCSG-PRE-ING-CIS-233.pdf>
- Granados, H. (2016). Ingeniería clínica y gestión tecnológica hospitalaria. En G. Malagón-Laverde, G. Pontón Laverde, & J. Reynales Londoño, *Gestión Hospitalaria* (Cuarta ed., págs. 209-290). Bogotá, D.C., Colombia: Editorial Médica Panamericana.
- MINISTERIO DE LA PROTECCION SOCIAL. (2011). *RESOLUCION 2981 DE 2011*. Resolución, Bogotá, D.C. Obtenido de [https://www.icbf.gov.co/cargues/avance/docs/resolucion\\_minproteccion\\_2981\\_2011.htm](https://www.icbf.gov.co/cargues/avance/docs/resolucion_minproteccion_2981_2011.htm)
- MINISTERIO DE PROTECCION SOCIAL. (2005). *DECRETO 4725 DE 2005*. Decreto, Bogotá, D.C. Obtenido de [https://www.icbf.gov.co/cargues/avance/docs/decreto\\_4725\\_2005.htm](https://www.icbf.gov.co/cargues/avance/docs/decreto_4725_2005.htm)
- MINISTERIO DE SALUD Y PROTECCION SOCIAL. (2019). *RESOLUCIÓN 3100 DE 2019*. Resolución, Bogotá, D.C. Obtenido de [https://www.icbf.gov.co/cargues/avance/docs/resolucion\\_minsaludps\\_3100\\_2019.htm](https://www.icbf.gov.co/cargues/avance/docs/resolucion_minsaludps_3100_2019.htm)

Muñoz Salazar, K. (2008). *Manual de protocolos de mantenimiento de equipos biomédicos para el Hospital Susana López de Valencia E.S.E.* Tesis de grado, Universidad Autónoma de Occidente, Santiago de Cali. Obtenido de

<https://red.uao.edu.co/bitstream/handle/10614/6063/T04059.pdf?sequence=1&isAllowed=y>

Organización Mundial de la Salud (OMS). (2012). *Introducción a la gestión de inventarios de equipo médico.* Ginebra, Suiza: Ediciones de la OMS. Obtenido de

[http://apps.who.int/iris/bitstream/handle/10665/44817/9789243501390\\_spa.pdf;sequence=1](http://apps.who.int/iris/bitstream/handle/10665/44817/9789243501390_spa.pdf;sequence=1)

Organización Mundial de la Salud (OMS). (2012). *Introducción al programa de mantenimiento de equipos médicos.* Ginebra, Suiza: Ediciones de la OMS. Obtenido de

[https://apps.who.int/iris/bitstream/handle/10665/44830/9789243501536\\_spa.pdf](https://apps.who.int/iris/bitstream/handle/10665/44830/9789243501536_spa.pdf)

PyPI - Python Software Foundation. (27 de febrero de 2022). *PyPI - Python Software Foundation.* Obtenido de <https://pypi.org/project/XlsxWriter/>

Strauss, A., & Corbin, J. (2002). *Basics of qualitative research. Techniques and procedures for developing grounded theory. (Spanish version)* (Segunda ed.). (E. Zimmerman, Trad.)

Medellín, Colombia: Editorial Universidad de Antioquia. Obtenido de

<https://diversidadlocal.files.wordpress.com/2012/09/bases-investigacion-cualitativa.pdf>

Todo Mantenimiento. (s.f.). *mantenimiento.win.* Obtenido de <https://mantenimiento.win/>

## APENDICE A - Python

```

import os
import json
import xlswriter
from datetime import datetime

from django.db.models import Q
from django.http import HttpResponse
from django.shortcuts import redirect, render
from django.core.serializers import serialize
from django.core.serializers.json import DjangoJSONEncoder

from django.views.generic import TemplateView

from .forms import DatoForm, InventarioForm, ProveedorForm
from .models import Dato, Inventario, Proveedor
from proyecto.settings import BASE_DIR

class HomeView(TemplateView):
    template_name = 'inventario/home.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['inventario'] = Inventario.objects.filter(visible = True)

        return context

    def post(self, request, *args, **kwargs):
        is_ajax = request.headers.get('X-Requested-With') == 'XMLHttpRequest'

        if is_ajax:
            data = json.load(request)
            consulta = data.get('valor_filtro')
            query = list(Inventario.objects.filter(Q(equipo__startswith =
consulta) | Q(marca__startswith = consulta) | Q(modelo__startswith =
consulta) | Q(dato__proveedor__proveedor__startswith = consulta) |
Q(serie__startswith = consulta), visible = True).values())

            return HttpResponse(json.dumps(query,
sort_keys=True, indent=1, cls=DjangoJSONEncoder))

class AgregarInventarioView(TemplateView):
    template_name = 'inventario/agregar_inventario.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['items'] =
list(Dato.objects.all().values('equipo').distinct())

```

```

return context

def post(self, request, *args, **kwargs):
    context = {}
    is_ajax = request.headers.get('X-Requested-With') == 'XMLHttpRequest'

    if is_ajax:
        data = json.load(request)

        if data.get('equipo'):
            equipo = data.get('equipo')
            query = list(Dato.objects.filter(equipo =
equipo).values('marca').distinct())
        else:
            marca = data.get('marca')
            query = list(Dato.objects.filter(marca =
marca).values('modelo').distinct())

        return HttpResponse(json.dumps(query))
    else:
        if request.POST['numero_inventario'] == "" or not
(Inventario.objects.filter(numero_inventario =
request.POST['numero_inventario']).exists()):
            post = request.POST.copy()
            number = Inventario.objects.all().count()+1
            codigo_generado = "B"+str("{:04d}".format(number))
            equipo = post['equipo']
            marca = post['marca']
            modelo = post['modelo']
            id_dato = Dato.objects.filter(equipo = equipo, marca = marca,
modelo = modelo).get().id
            post['dato'] = id_dato
            post['codigo'] = codigo_generado
            post['visible'] = True
            request.POST = post

            form = InventarioForm(data=request.POST)

            if form.is_valid():
                form.save()
                return redirect('Inventario:home')
            else:
                context['items'] =
list(Dato.objects.all().values('equipo').distinct())
                context['errors'] = form.errors
                context['values'] = form.cleaned_data

                return render(request, self.template_name, {'form':
context})

        if Inventario.objects.filter(numero_inventario =
request.POST['numero_inventario']).exists():
            form = InventarioForm(data=request.POST)

```

```

        form.add_error('numero_inventario', 'This field is unique')
        context['items'] =
list(Dato.objects.all().values('equipo').distinct())
        context['errors'] = form.errors
        context['values'] = form.cleaned_data

        return render(request, self.template_name, {'form': context})

class EditarInventarioView(TemplateView):
    template_name = 'inventario/editar_inventario.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['item'] = list(Inventario.objects.filter(id =
kwargs['pk']).values())[0]
        context['equipos'] =
list(Dato.objects.all().values('equipo').distinct())
        context['marcas'] = list(Dato.objects.filter(equipo =
context['item']['equipo']).values('marca').distinct())
        context['modelos'] = list(Dato.objects.filter(equipo =
context['item']['equipo'], marca =
context['item']['marca']).values('modelo').distinct())

        return context

    def post(self, request, *args, **kwargs):

        context = {}
        is_ajax = request.headers.get('X-Requested-With') == 'XMLHttpRequest'

        if is_ajax:
            data = json.load(request)

            if data.get('equipo'):
                equipo = data.get('equipo')
                query = list(Dato.objects.filter(equipo =
equipo).values('marca').distinct())
            else:
                marca = data.get('marca')
                query = list(Dato.objects.filter(marca =
marca).values('modelo').distinct())

            return HttpResponse(json.dumps(query))
        else:
            if request.POST['numero_inventario'] == "" or not
(Inventario.objects.filter(numero_inventario =
request.POST['numero_inventario']).exists()):
                instance = Inventario.objects.get(id=kwargs['pk'])

                post = request.POST.copy()
                equipo = post['equipo']

```

```

        marca = post['marca']
        modelo = post['modelo']
        id_dato = Dato.objects.filter(equipo = equipo, marca = marca,
modelo = modelo).get().id
        post['dato'] = id_dato
        post['codigo'] = instance.codigo
        post['visible'] = True
        request.POST = post

        form = InventarioForm(request.POST or None,
instance=instance)

        if form.is_valid():
            form.save()
            return redirect('Inventario:home')
        else:
            context['items'] =
list(Dato.objects.all().values('equipo').distinct())
            context['errors'] = form.errors
            context['values'] = form.cleaned_data

            return render(request, self.template_name, {'form':
context})

        if Inventario.objects.filter(numero_inventario =
request.POST['numero_inventario']).exists():
            form = InventarioForm(data=request.POST)
            form.add_error('numero_inventario', 'This field is unique')
            context['items'] =
list(Dato.objects.all().values('equipo').distinct())
            context['errors'] = form.errors
            context['values'] = form.cleaned_data

            return render(request, self.template_name, {'form': context})

class EliminarInventarioView(TemplateView):
    template_name = 'inventario/eliminar_inventario.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['item'] = list(Inventario.objects.filter(id =
kwargs['pk']).values())[0]
        context['equipos'] =
list(Dato.objects.all().values('equipo').distinct())
        context['marcas'] = list(Dato.objects.filter(equipo =
context['item']['equipo']).values('marca').distinct())
        context['modelos'] = list(Dato.objects.filter(equipo =
context['item']['equipo'], marca =
context['item']['marca']).values('modelo').distinct())

        return context

```

```

def post(self, request, *args, **kwargs):
    update_list = []
    model_qs= Inventario.objects.filter(id__gt=kwargs['pk'])
    Inventario.objects.filter(id = kwargs['pk']).delete()

    for model_obj in model_qs:
        model_obj.codigo =
"B"+str("{:04d}").format(int(model_obj.codigo.split("B")[1])-1)) # Or what
ever the value is for simplicitly im providing foo only
        update_list.append(model_obj)

    Inventario.objects.bulk_update(update_list,['codigo'])

    return redirect('Inventario:home')

class DatoView(TemplateView):
    template_name = 'inventario/datos.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['datos'] = Dato.objects.filter(visible = True).values()

    return context

    def post(self, request, *args, **kwargs):
        is_ajax = request.headers.get('X-Requested-With') == 'XMLHttpRequest'

        if is_ajax:
            data = json.load(request)
            consulta = data.get('valor_filtro')
            query = list(Dato.objects.filter(Q(equipo__startswith = consulta)
| Q(marca__startswith = consulta) | Q(modelo__startswith = consulta), visible
= True).values())

            return HttpResponse(json.dumps(query))

class AgregarDatoView(TemplateView):
    template_name = 'inventario/agregar_dato.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['proveedores'] = Proveedor.objects.filter(visible = True)

    return context

    def post(self, request, *args, **kwargs):
        post = request.POST.copy()
        post['visible'] = True
        request.POST = post

        form = DatoForm(data=request.POST)

```

```

    if form.is_valid():
        form.save()
        return redirect('Inventario:datos')
    else:
        context = super().get_context_data(**kwargs)
        context['proveedores'] = list(Proveedor.objects.filter(visible =
True).values('id', 'proveedor'))
        context['errors'] = form.errors
        context['values'] = form.cleaned_data

        return render(request, self.template_name, {'form': context})

class EditarDatoView(TemplateView):
    template_name = 'inventario/editar_dato.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['item'] = list(Dato.objects.filter(id = kwargs['pk']).values(
            'equipo',
            'marca',
            'modelo',
            'registro_invima',
            'riesgo',
            'vida_util',
            'verificacion',
            'periodicidad_verificacion',
            'calibracion',
            'periodicidad_calibracion',
            'clasificacion',
            'manuales',
            'alimentacion',
            'corriente_maxima',
            'frecuencia',
            'velocidad',
            'tension_maxima',
            'corriente_minima',
            'peso',
            'presion',
            'tension_minima',
            'potencia',
            'temperatura',
            'humedad',
            'plan',
            'critico',
            'predominante',
            'guias_rapidas'))[0]

        return context

    def post(self, request, *args, **kwargs):
        context = {}

```

```

instance = Dato.objects.get(id=kwargs['pk'])
post = request.POST.copy()
post['visible'] = True
request.POST = post
form = DatoForm(request.POST or None, instance=instance)

if form.is_valid():
    form.save()
    return redirect('Inventario:datos')
else:
    context['errors'] = form.errors
    context['values'] = form.cleaned_data
    return render(request, self.template_name, {'form': context})

class EliminarDatoView(TemplateView):
    template_name = 'inventario/eliminar_dato.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['item'] = list(Dato.objects.filter(id = kwargs['pk']).values(
            'equipo',
            'marca',
            'modelo',
            'registro_invima',
            'riesgo',
            'vida_util',
            'verificacion',
            'periodicidad_verificacion',
            'calibracion',
            'periodicidad_calibracion',
            'clasificacion',
            'manuales',
            'alimentacion',
            'corriente_maxima',
            'frecuencia',
            'velocidad',
            'tension_maxima',
            'corriente_minima',
            'peso',
            'presion',
            'tension_minima',
            'potencia',
            'temperatura',
            'humedad',
            'plan',
            'critico',
            'predominante',
            'guias_rapidas'))[0]

        return context

```

```

def post(self, request, *args, **kwargs):
    Dato.objects.filter(id = kwargs['pk']).update(visible = False)
    return redirect('Inventario:datos')

class ProveedorView(TemplateView):
    template_name = 'inventario/proveedores.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['proveedores'] = Proveedor.objects.filter(visible =
True).values()

        return context

    def post(self, request, *args, **kwargs):
        is_ajax = request.headers.get('X-Requested-With') == 'XMLHttpRequest'

        if is_ajax:
            data = json.load(request)
            consulta = data.get('valor_filtro')
            query = list(Dato.objects.filter(Q(equipo__startswith = consulta)
| Q(marca__startswith = consulta) | Q(modelo__startswith = consulta), visible
= True).values())

            return HttpResponse(json.dumps(query))

class AgregarProveedorView(TemplateView):
    template_name = 'inventario/agregar_proveedor.html'

    def post(self, request, *args, **kwargs):
        context = {}
        post = request.POST.copy()
        post['visible'] = True
        request.POST = post

        form = ProveedorForm(data=request.POST)

        if form.is_valid():
            form.save()

            return redirect('Inventario:proveedores')
        else:
            context['errors'] = form.errors
            context['values'] = form.cleaned_data
            return render(request, self.template_name, {'form': context})

class EditarProveedorView(TemplateView):
    template_name = 'inventario/editar_proveedor.html'

    def get_context_data(self, **kwargs):

```

```

        context = super().get_context_data(**kwargs)
        context['item'] = list(Proveedor.objects.filter(id =
kwargs['pk']).values(
            'proveedor',
            'telefono_proveedor',
            'ciudad_proveedor',
            'correo_proveedor',
            'direccion_proveedor',
            'fax_proveedor',
            'nombre_fabricante',
            'pais',
            'nombre_contratista',
            'telefono_contratista',
            'ciudad_contratista',
            'correo_contratista',
            'direccion_contratista',
            'fax_contratista'))[0]

    return context

def post(self, request, *args, **kwargs):
    context = {}
    instance = Proveedor.objects.get(id=kwargs['pk'])
    post = request.POST.copy()
    post['visible'] = True
    request.POST = post
    form = ProveedorForm(request.POST or None, instance=instance)

    if form.is_valid():
        form.save()
        return redirect('Inventario:proveedores')
    else:
        context['errors'] = form.errors
        context['values'] = form.cleaned_data
        return render(request, self.template_name, {'form': context})

class EliminarProveedorView(TemplateView):
    template_name = 'inventario/eliminar_proveedor.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['item'] = list(Proveedor.objects.filter(id =
kwargs['pk']).values(
            'proveedor',
            'telefono_proveedor',
            'ciudad_proveedor',
            'correo_proveedor',
            'direccion_proveedor',
            'fax_proveedor',
            'nombre_fabricante',
            'pais',
            'nombre_contratista',

```

```

        'telefono_contratista',
        'ciudad_contratista',
        'correo_contratista',
        'direccion_contratista',
        'fax_contratista'))[0]

    return context

def post(self, request, *args, **kwargs):
    Proveedor.objects.filter(id = kwargs['pk']).update(visible = False)
    return redirect('Inventario:proveedores')

class ImprimirView(TemplateView):
    template_name = 'inventario/imprimir.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['id'] = kwargs['pk']

    return context

class HojaDeVidaView(TemplateView):
    template_name = 'inventario/hoja_de_vida.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        item = Inventario.objects.get(id = kwargs['pk'])
        context['item'] = item
        context['dato'] = Dato.objects.filter(equipo = item.equipo, marca =
item.marca, modelo = item.modelo).get()

    return context

class ReciboTecnologiaView(TemplateView):
    template_name = 'inventario/recibo_tecnologia.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        item = Inventario.objects.get(id = kwargs['pk'])
        context['item'] = item
        context['dato'] = Dato.objects.filter(equipo = item.equipo, marca =
item.marca, modelo = item.modelo).get()

    return context

class ReciboServicioView(TemplateView):
    template_name = 'inventario/recibo_servicio.html'

```

```

def get_context_data(self, **kwargs):
    context = super().get_context_data(**kwargs)
    item = Inventario.objects.get(id = kwargs['pk'])
    context['item'] = item
    context['dato'] = Dato.objects.filter(equipo = item.equipo, marca =
item.marca, modelo = item.modelo).get()

    return context

def export_excel_inventario(request):

    now = datetime.now().strftime("%d-%m-%Y-%HH-%MM-%SS")
    folder_path = os.path.join(BASE_DIR, 'media', 'INVENTARIO CIA', '')
    os.makedirs(folder_path, exist_ok=True)

    # Create a workbook and add a worksheet.
    workbook = xlsxwriter.Workbook(folder_path + now + '.xlsx')
    worksheet_inventario = workbook.add_worksheet("INVENTARIOS")
    worksheet_datos = workbook.add_worksheet("Datos")

    # Add a bold format to use to highlight cells.
    bold = workbook.add_format({'bold': True})

    # Add a number format for cells with money.
    money = workbook.add_format({'num_format': '$#,##0'})

    # Write some data headers.
    worksheet_inventario.write('A1', 'CÓDIGO', bold)
    worksheet_inventario.write('B1', 'EMPRESA', bold)
    worksheet_inventario.write('C1', 'SEDE', bold)
    worksheet_inventario.write('D1', 'ÁREA', bold)
    worksheet_inventario.write('E1', 'UBICACIÓN ESPECIFICA', bold)
    worksheet_inventario.write('F1', 'EQUIPO', bold)
    worksheet_inventario.write('G1', 'MARCA', bold)
    worksheet_inventario.write('H1', 'MODELO', bold)
    worksheet_inventario.write('I1', 'SERIE', bold)

    worksheet_inventario.set_column('A1:A1', len("CÓDIGO") + 2)
    worksheet_inventario.set_column('B1:B1', len("EMPRESA") + 2)
    worksheet_inventario.set_column('C1:C1', len("SEDE") + 2)
    worksheet_inventario.set_column('D1:D1', len("ÁREA") + 2)
    worksheet_inventario.set_column('E1:E1', len("UBICACIÓN ESPECIFICA") + 2)
    worksheet_inventario.set_column('F1:F1', len("EQUIPO") + 2)
    worksheet_inventario.set_column('G1:G1', len("MARCA") + 2)
    worksheet_inventario.set_column('H1:H1', len("MODELO") + 2)
    worksheet_inventario.set_column('I1:I1', len("SERIE") + 2)

    # Some data we want to write to the worksheet_inventario.
    expenses = list(Inventario.objects.filter(visible =
True).values_list('codigo', 'empresa', 'sede', 'servicio', 'ubicacion', 'equipo', '
marca', 'modelo', 'serie'))

```

```

# Start from the first cell below the headers.
row = 1
col = 0
length_codigo = len("CÓDIGO") + 2
length_empresa = len("EMPRESA") + 2
length_sede = len("SEDE") + 2
length_area = len("ÁREA") + 2
length_ubicacion = len("UBICACIÓN ESPECIFICA") + 2
length_equipo = len("EQUIPO") + 2
length_marca = len("MARCA") + 2
length_modelo = len("MODELO") + 2
length_serie = len("SERIE") + 2

# Iterate over the data and write it out row by row.

for codigo, empresa, sede, area, ubicacion, equipo, marca, modelo, serie
in (expenses):
    worksheet_inventario.write(row, col,      codigo)
    worksheet_inventario.write(row, col + 1, empresa)
    worksheet_inventario.write(row, col + 2, sede)
    worksheet_inventario.write(row, col + 3, area)
    worksheet_inventario.write(row, col + 4, ubicacion)
    worksheet_inventario.write(row, col + 5, equipo)
    worksheet_inventario.write(row, col + 6, marca)
    worksheet_inventario.write(row, col + 7, modelo)
    worksheet_inventario.write(row, col + 8, serie)

    if length_codigo < len(codigo) + 2:
        worksheet_inventario.set_column('A1:A1', len(codigo) + 2)
        length_area = len(codigo) + 2
    if length_empresa < len(empresa) + 2:
        worksheet_inventario.set_column('B1:B1', len(empresa) + 2)
        length_empresa = len(empresa) + 2
    if length_sede < len(sede) + 2:
        worksheet_inventario.set_column('C1:C1', len(sede) + 2)
        length_sede = len(sede) + 2
    if length_area < len(area) + 2:
        worksheet_inventario.set_column('D1:D1', len(area) + 2)
        length_area = len(area) + 2
    if length_ubicacion < len(ubicacion) + 2:
        worksheet_inventario.set_column('E1:E1', len(ubicacion) + 2)
        length_ubicacion = len(ubicacion) + 2
    if length_equipo < len(equipo) + 2:
        worksheet_inventario.set_column('F1:F1', len(equipo) + 2)
        length_equipo = len(equipo) + 2
    if length_marca < len(marca) + 2:
        worksheet_inventario.set_column('G1:G1', len(marca) + 2)
        length_marca = len(marca) + 2
    if length_modelo < len(modelo) + 2:
        worksheet_inventario.set_column('H1:H1', len(modelo) + 2)
        length_modelo = len(modelo) + 2
    if length_serie < len(serie) + 2:
        worksheet_inventario.set_column('I1:I1', len(serie) + 2)

```

```

length_serie = len(serie) + 2

row += 1

# Write a total using a formula.
# worksheet.write(row, 0, 'Total', bold)
# worksheet.write(row, 1, '=SUM(B2:B5)', money)

worksheet_datos.freeze_panes(0, 1)
worksheet_datos.freeze_panes(0, 2)

worksheet_datos.write('A1', 'NOMBRE', bold)
worksheet_datos.write('B1', 'REGISTRO INVIMA', bold)
worksheet_datos.write('C1', 'VIDA UTIL', bold)
worksheet_datos.write('D1', 'REQUIERE VERIFICACIÓN METROLOGICA', bold)
worksheet_datos.write('E1', 'PERIODICIDAD', bold)
worksheet_datos.write('F1', 'REQUIERE CALIBRACIÓN METROLOGICA', bold)
worksheet_datos.write('G1', 'PERIODICIDAD', bold)
worksheet_datos.write('H1', 'CLASIFICACION BIOMEDICA', bold)
worksheet_datos.write('I1', 'MANUALES', bold)
worksheet_datos.write('J1', 'ALIMENTACION', bold)
worksheet_datos.write('K1', 'CORRIENTE MAX', bold)
worksheet_datos.write('L1', 'FRECUENCIA', bold)
worksheet_datos.write('M1', 'VELOCIDAD', bold)
worksheet_datos.write('N1', 'TENSIÓN MAX', bold)
worksheet_datos.write('O1', 'CORRIENTE MIN', bold)
worksheet_datos.write('P1', 'PESO', bold)
worksheet_datos.write('Q1', 'PRESIÓN', bold)
worksheet_datos.write('R1', 'TENSIÓN MIN', bold)
worksheet_datos.write('S1', 'POTENCIA', bold)
worksheet_datos.write('T1', 'TEMPERATURA', bold)
worksheet_datos.write('U1', 'HUMEDAD', bold)
worksheet_datos.write('V1', 'PLAN METROLOGICO', bold)
worksheet_datos.write('W1', 'EQUIPO CRITICO', bold)
worksheet_datos.write('X1', 'EQUIPO PREDOMINANTE', bold)
worksheet_datos.write('Y1', 'GUIAS RÁPIDAS', bold)

worksheet_datos.set_column('A1:A1', len("NOMBRE") + 2)
worksheet_datos.set_column('B1:B1', len("REGISTRO INVIMA") + 2)
worksheet_datos.set_column('C1:C1', len("VIDA UTIL") + 2)
worksheet_datos.set_column('D1:D1', len("REQUIERE VERIFICACIÓN
METROLOGICA") + 2)
worksheet_datos.set_column('E1:E1', len("PERIODICIDAD") + 2)
worksheet_datos.set_column('F1:F1', len("REQUIERE CALIBRACIÓN
METROLOGICA") + 2)
worksheet_datos.set_column('G1:G1', len("PERIODICIDAD") + 2)
worksheet_datos.set_column('H1:H1', len("CLASIFICACION BIOMEDICA") + 2)
worksheet_datos.set_column('I1:I1', len("MANUALES") + 2)
worksheet_datos.set_column('J1:J1', len("ALIMENTACION") + 2)
worksheet_datos.set_column('K1:K1', len("CORRIENTE MAX") + 2)
worksheet_datos.set_column('L1:L1', len("FRECUENCIA") + 2)
worksheet_datos.set_column('M1:M1', len("VELOCIDAD") + 2)
worksheet_datos.set_column('N1:N1', len("TENSIÓN MAX") + 2)

```

```

worksheet_datos.set_column('O1:O1',len("CORRIENTE MIN") + 2)
worksheet_datos.set_column('P1:P1',len("PESO") + 2)
worksheet_datos.set_column('Q1:Q1',len("PRESIÓN") + 2)
worksheet_datos.set_column('R1:R1',len("TENSIÓN MIN") + 2)
worksheet_datos.set_column('S1:S1',len("POTENCIA") + 2)
worksheet_datos.set_column('T1:T1',len("TEMPERATURA") + 2)
worksheet_datos.set_column('U1:U1',len("HUMEDAD") + 2)
worksheet_datos.set_column('V1:V1',len("PLAN METROLOGICO") + 2)
worksheet_datos.set_column('W1:W1',len("EQUIPO CRITICO") + 2)
worksheet_datos.set_column('X1:X1',len("EQUIPO PREDOMINANTE") + 2)
worksheet_datos.set_column('Y1:Y1',len("GUIAS RÁPIDAS") + 2)

expenses = list(Dato.objects.filter(visible =
True).values_list('equipo', 'marca', 'modelo', 'registro_invima', 'vida_util', 've
rificacion', 'periodicidad_verificacion', 'calibracion', 'periodicidad_calibraci
on',

'clasificacion', 'manuales', 'alimentacion', 'corriente_maxima', 'frecuencia', 've
locidad', 'tension_maxima', 'corriente_minima', 'peso', 'presion', 'tension_minima
', 'potencia', 'temperatura', 'humedad',
    'plan', 'critico', 'predominante', 'guias_rapidas'))

row = 1
col = 0
length_nombre = len("NOMBRE") + 2
length_invima = len("REGISTRO INVIMA") + 2

for
(equipo,marca,modelo,registro_invima,vida_util,verificacion,periodicidad_verif
icacion,calibracion,periodicidad_calibracion,clasificacion,manuales,alimentac
ion,corriente_max,frecuencia,

velocidad,tension_max,corriente_min,peso,presion,tension_min,potencia,tempera
tura,humedad,plan,critico,predominante,guia_rapida) in (expenses):
    worksheet_datos.write(row, col,      equipo+" "+marca+" "+modelo)
    worksheet_datos.write(row, col + 1, registro_invima)
    worksheet_datos.write(row, col + 2, vida_util)
    worksheet_datos.write(row, col + 3, verificacion)
    worksheet_datos.write(row, col + 4, periodicidad_verificacion)
    worksheet_datos.write(row, col + 5, calibracion)
    worksheet_datos.write(row, col + 6, periodicidad_calibracion)
    worksheet_datos.write(row, col + 7, clasificacion)
    worksheet_datos.write(row, col + 8, manuales)
    worksheet_datos.write(row, col + 9, alimentacion)
    worksheet_datos.write(row, col + 10, corriente_max)
    worksheet_datos.write(row, col + 11, frecuencia)
    worksheet_datos.write(row, col + 12, velocidad)
    worksheet_datos.write(row, col + 13, tension_max)
    worksheet_datos.write(row, col + 14, corriente_min)
    worksheet_datos.write(row, col + 15, peso)
    worksheet_datos.write(row, col + 16, presion)
    worksheet_datos.write(row, col + 17, tension_min)
    worksheet_datos.write(row, col + 18, potencia)

```

```
worksheet_datos.write(row, col + 19, temperatura)
worksheet_datos.write(row, col + 20, humedad)
worksheet_datos.write(row, col + 21, plan)
worksheet_datos.write(row, col + 22, critico)
worksheet_datos.write(row, col + 23, predominante)
worksheet_datos.write(row, col + 24, guia_rapida)

if length_nombre < len(equipo+" "+marca+" "+modelo) + 2:
    worksheet_datos.set_column('A1:A1',len(equipo+" "+marca+"
"+modelo) + 2)
    length_area = len(equipo+" "+marca+" "+modelo) + 2
if length_invima < len(registro_invima) + 2:
    worksheet_datos.set_column('B1:B1',len(registro_invima) + 2)
    length_empresa = len(registro_invima) + 2

row += 1

workbook.close()

return HttpResponse(json.dumps({"msg": "Archivo descargado
 exitosadamente"}), content_type="application/json")
```