**Identificación del problema:**

**Descripción del contexto problemático (causas y síntomas)**

En el mundo existen muchos tipos de enfermedades unas más graves que otras entre las graves están las relacionadas con el corazón. Las enfermedades más comunes relacionadas a problemas en el corazón son la Angina, Infarto de miocardio y la Hipertensión (Henríquez, 2018), Estas enfermedades relacionadas al corazón también llamadas enfermedades coronarias, deben ser tratadas a tiempo de lo contrario esto puede llevar graves consecuencias incluso pueden terminar en la muerte. En la ciudad de Cleveland en los Estados Unidos un hospital se vio interesado en la problemática de identificar a tiempo estas enfermedades realizó un estudio a una muestra de 304 personas de esta ciudad para identificar si estas contaban con problemas en el corazón o no tomando una serie de datos y exámenes.

En un *dataset* la información se puede encontrar registrada de manera consistente y esta no es la excepción, se cuenta con información valiosa y representativa esta no se encuentra estructurada de la manera adecuada por lo cual si se quisiera realizar un *análisis a la antigua* sería necesario tener un amplio equipo encargado de leer y analizar cada uno de estos casos registrados de manera manual en los que se incluyen más de 13 parámetros (Columnas), en otras palabras ¡sería necesario analizar 4.256 datos uno por uno! Por si esto fuera poco, suponiendo que esta labor tiene éxito, a su vez es necesario realizar un análisis profundo de esta información no basta con solo saber el contenido de la muestra, la verdadera importancia radica en poder inferir si una persona tiene problemas cardiacos o no, sería muy *engorroso* para un médico buscar en el *dataset* un paciente que tenga las mismas características y mirar si este tendrá problemas en el corazón o no, es muy probable que simplemente no lo encuentre y que este se equivoque en la búsqueda. Estas y demás labores de cierto modo son inalcanzables si se hacen manualmente pero un *software* puede resolver en unos pocos segundos, la idea es que mediante la carga de datos y el Machine Learning se pueda facilitar la labor de los médicos a la hora de determinar si una persona tiene o puede tener problemas en el corazón, una labor que debe realizarse con sumo cuidado debido a que se trata de vidas humanas.

**Identificación de necesidades:**

**Definición del problema:**

Un hospital de Cleveland ha realizado una investigación sobre personas que tienen y no tienen problemas en el corazón, a cada persona de la muestra se le ha recopilado la edad, sexo, dolor de pecho, presión arterial en reposo, colesterol en la sangre, resultado de examen de azúcar en sangre en ayunas, resultado del electro cardiograma en reposo, frecuencia cardiaca máxima en reposo, si tiene angina inducida por ejercicio, ejercicio relativo al descanso, *slope*, numero de vasos sanguíneos principales coloreados por fluoroscopia, Talasemia y el resultado de si tiene problemas en el corazón o no. Se requiere ordenar, visualizar mediante tablas y gráficos de manera intuitiva la información además de poder inferir si una persona tendrá problemas en el corazón de manera concisa.

**Recopilación de la información necesaria**

El dolor de pecho es un síntoma común que se presenta en personas con un mal flujo sanguíneo al corazón y que puede significar un riesgo de ataque al corazón (Chen, s.f.) si se realiza un examen este pude dar como resultado 3 valores.

1. Angina
2. Asintomático
3. Anormal

El electrocardiograma es un examen que registra la actividad eléctrica del corazón y permite detectar si una persona ha tenido algún infarto o si tiene arritmias cardiacas este puede generar alguno de los siguientes resultados (Mañero, 2019)

1. Normal
2. Anormal
3. Hipertenso

La angina inducida por ejercicio es si una persona presenta dolor de pecho al someterse a alguna actividad física (Chen, s.f.)

*Slope*: Es una pendiente que representa una tolerancia al ejercicio físico (R S Finkelhor, 1986)

Un árbol de decisión aplicado en el Machine Learning es conjunto de datos que se forman (Valga la redundancia) en forma de árbol dependiendo a una función que el programador asigna, este conjunto de datos tiene muchas utilidades, sus dos principales son poder determinar si un set de datos de entrada pertenece a una determinada clase, esta puede ser clasificada por un resultado binario (Verdadero/Falso) o por más de un resultado. En ciertos casos en los que el árbol no sea muy grande puede permitir visualizar de manera intuitiva cómo está funcionando la clasificación en el *dataset*.

La regresión en el Machine Learning es una técnica en la cual el resultado es un valor numérico en un conjunto infinito de soluciones como estimar que cantidad de productos se va a vender en un supermercado (Heras, 2020)

**Estado del arte**

**HELPSALUD**

Es un proyecto con digitalizar el sector salud a través de soluciones de software aplicando Machine Learning e Inteligencia Artificial, su objetivo final es el siguiente “Proporcionar el mejor diagnóstico, pronóstico o estimación de riesgos, según sea el caso, que sirva de ayuda al personal clínico en la toma de decisiones acerca de sus pacientes permitiendo un tratamiento más personalizado y eficaz” (ITI, 2017)

**Impresión 3D**

Aunque a primera vista la impresión 3D no parezca tener relación con el sector salud y mucho menos con el Machine Learning esto no puede estar más alejado de la realidad, combinando el uso de *big data*, Machine Learning e inteligencia artificial fue posible la impresión 3D del primer cráneo completo en Titania que le fue implantado a una persona de 60 años de edad exitosamente (Aon, 2020)

**Búsqueda de soluciones creativas**

●**Ideas para la selección del lenguaje de programación**

{Java} Crear la aplicación en el lenguaje de programación Java, un lenguaje muy popular usado en más de 3 billones de dispositivos en el mundo y usar alguna librería que permite visualizar un mapa en la aplicación.

{Bloques} Crear la aplicación en un lenguaje de programación por bloques como lo puede ser App inventor y Scratch.

{Adaptar} Buscar una aplicación que ya alguien haya creado en cualquier tipo de lenguaje y adaptarla a las necesidades del cliente.

{Python} Crear nuestra aplicación en el lenguaje de programación Python, un lenguaje que en la última década ha tenido un crecimiento y un avance considerablemente alto.

{Excel} Crear la aplicación en Excel aplicación que muchas personas usan y conocen hoy en día.

{C#} Crear la aplicación desde 0 en el lenguaje de programación C# haciendo uso de las diferentes herramientas que este ofrece para la visualización de datos.

●**Ideas para la aplicación del Machine Learning**

{Clasificación Propia} Hacer uso de la técnica de clasificación por clases implementando un árbol de decisión propio asignándole determinadas funciones que también son propias

{Clasificación Papel} Implementar la técnica de clasificación con un árbol de decisión a papel y lápiz y clasificar cada línea del *dataset* de esta manera

{Regresión Propia} Hacer uso de la técnica de regresión implementando un árbol de decisión propio asignándole determinadas funciones que también son propias

{Clasificación Librería} Hacer uso de la técnica de clasificación por clases haciendo uso de una librería de machine learning que ofrezca el lenguaje de programación

{No Machine} El uso de machine learning no es necesario en esta ocasión por lo cual se podría omitir

{Regresión Librería} Hacer uso de la técnica de regresión haciendo uso de una librería de machine learning que ofrezca el lenguaje de programación

●**Ideas para la carga del DataSet**

{Métodos Propios} Hacer la carga del DataSet mediante el uso de métodos propios del lenguaje de programación escogido combinado con métodos creados por nosotros.

{Métodos en librerías} Hacer la carga del DataSet mediante el uso exclusivo de métodos propios del lenguaje de programación escogido.

{Manualmente} No hacer uso de un DataSet si no que por el contrario el usuario escriba. manualmente cada una de las ubicaciones que desee usar en el programa.

●**Ideas para la creación de la interfaz gráfica del usuario**

{JavaFx} Contemplar la posibilidad de realizar la interfaz gráfica mediante el uso de la librería gráfica de Javafx.

{Swing} Realizar toda la interfaz gráfica mediante el uso de la librería gráfica swing.

{AWT} Realizar toda la interfaz gráfica mediante el uso de la librería gráfica AWT.

{WPF} Realizar toda la interfaz gráfica del programa mediante el uso de las librerías gráficas WPF.

{Windows Forms} Realizar toda la interfaz gráfica del programa mediante el uso de las librerías gráficas de windows forms.

{Sin Interfaz} No hacer ninguna interfaz gráfica y solamente usar la consola de comandos.

# Referencias

Aon. (04 de Agosto de 2020). *NOA*. Obtenido de https://noa.aon.es/machine-learning-aplicaciones-salud/

Chen, M. A. (s.f.). *MedlinePlus*. Obtenido de https://medlineplus.gov/ency/patientinstructions/000775.htm

Henríquez, K. (30 de Septiembre de 2018). *elsalvador.com*. Obtenido de https://www.elsalvador.com/vida/salud/7-enfermedades-cardiacas-mas-comunes-a-nivel-mundial-2/523277/2018/

Heras, J. M. (29 de Septiembre de 2020). *IArtificial.net*. Obtenido de https://www.iartificial.net/clasificacion-o-regresion/#Regresion

ITI. (01 de Enero de 2017). *ITI Investigate To Innovate*. Obtenido de https://www.iti.es/proyectosidi/helpsalud-machine-learning-salud/

Mañero, M. R. (14 de Agosto de 2019). *Fundación Española del corazon* .

R S Finkelhor, K. E. (Agosto de 1986). *Pubmed.gov*. Obtenido de https://pubmed.ncbi.nlm.nih.gov/3739881/

Data set: <https://www.kaggle.com/ronitf/heart-disease-uci>