**DASHBOARD PARA SEGUIMIENTO Y ANÁLISIS DE SERVICIOS DE RED**

**PEDRO JAVIER BOHÓRQUEZ CAMELO**

**PAULA ANDREA CAICEDO ESCOBAR**

**LUISA FERNANDA MONSALVE SERRANO**

**BASES DE DATOS II**

**INGENIERÍA DE SOFTWARE I**

**REDES DE DATOS**

**ELKIN ALFEDRO ALBARRACÍN NAVAS**

**LENIN JAVIER SERRANO GIL**

**UNIVERSIDAD PONTIFICIA BOLIVARIANA**

**ESCUELA DE INGENIERÍA**  
 **FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**BUCARAMANGA**

**2021**

1. **SITUACIÓN PROBLEMA**

A - La situación sanitaria actual ha modificado los sistemas educativos y empresariales, involucrando al teletrabajo y la modalidad virtual o tele-presencialidad, afectando directamente el consumo de canal de datos o Internet. Por ende, en los hogares se ha visto la tarea de optimizar el uso del ancho de banda, donde diariamente ingresan dispositivos móviles, host no identificados y protocolos innecesarios que afectan negativamente el trabajo que se está realizando, ya sea de uso educativo o profesional. Por tal motivo, se visualiza en detalle las diferentes solicitudes hacia Internet o servicios consumidos en cada hogar en un rango de tiempo determinado, con el fin de encontrar soluciones que eviten el consumo innecesario de datos.

B – ¿Como visualizar la información de un segmento de red para conocer qué tipo de tráfico se consume?

1. **MARCO CONCEPTUAL**

En este capítulo se describirán conceptos generales para el proyecto, empezando con la arquitectura tecnología y arquitectura de software que se divide en: Redes Informáticas, que compone: tipos de redes informática, protocolos, rastreador de paquetes, modelo OSI, entre otros subtemas. Y luego, la Bases de Datos, que compone: características de bases de datos, sistema gestor de bases relacionales, entre otros subtemas. Finalmente, la Ingeniería de Software que compone: metodologías agiles, características del software, proceso del software, entre otros subtemas.

**2.1 ARQUITECTURA DE TECNOLOGICA DE TRES NIVELES.**

Un modelo arquitectónico trabaja en capas, para conseguir independencia y robustez de cada una de ellas centradose en sus objetivos específicos.

Las arquitecturas de tres niveles son modelos de programación que permiten la distribución de la funcionalidad de la aplicación en tres sistemas independientes, normalmente:

* Componentes de cliente que se ejecutan en estaciones de trabajo locales (nivel uno)
* Procesos que se ejecutan en servidores remotos (nivel dos)
* Una colección discreta de bases de datos, gestores de recursos y aplicaciones de sistema principal (nivel tres)

Por la definición del proyecto, se define una arquitectura de tres niveles que especifican las siguientes características, de acuerdo a lo explicado anteriormente:

* El primer nivel es la capa de presentación que interactua con el usuario y permite realizar consultas a procesos de segundo nivel de forma segura.
* El segundo nivel es la capa lógica de aplicación, este proceso accede y hace las conexiones al tercer nivel. Varios componentes de cliente pueden acceder simultáneamente a los procesos de segundo nivel, por lo que esta capa lógica de aplicación debe gestionar sus propias transacciones.
* El tercer nivel la capa que maneja la información del sistema a nivel de base de datos, los servicios de tercer nivel están protegidos del acceso directo por parte de los componentes del cliente que residen en una red segura. La interacción debe producirse a través de los procesos de segundo nivel.

**2.2 ARQUITECURA DE SOFTWARE**

Desarrollar un proyecto implica definir y planear que herramientas son necesarias para realizar la aplicación. El proceso se enfoca principalmente en entender los requerimientos que el cliente necesita para automatizar, mejorar y definir los estructuras que permite modelar el sistema.

Mediante patrones o lineamientos que ayudan a la construcción del programa, se empiezan a definir las tecnologías, servidores, bases de datos, entre otros, con el fin de crear una estructura de la aplicación que sea fácilmente escalable y que no esté fuertemente acoplada [1].

La arquitectura del proyecto se divide en dos partes, las cuales son:

* Redes Informáticas
* Bases de Datos

A continuación, se realiza una investigación de los temas más relevantes para cada parte.

**2.3 REDES INFORMÁTICAS**

Es un conjunto de sistemas informáticos (ordenadores, móviles, dispositivos IoT) que se conectan entre sí, a través de un enlace que puede ser alámbrico o inalámbrico donde constantemente se comparte información mediante paquetes de datos, transmitidos por impulsos eléctricos, ondas electromagnéticas o cualquier otro medio [2].

**2.3.1** **Tipos De Redes Informáticas**

Las redes pueden clasificarse según su alcance. Estas son algunas de las más conocidas [3]:

Redes **LAN** (Local Area Network): redes de tamaño menor, que pertenecen a una sola localización de una red. Por ejemplo, se puede encontrar en: un cuarto, un edificio, un departamento, una residencia, entre otros.

Redes **MAN** (Metropolitan Area Network): redes de tamaño intermedio que abarca una cobertura en un área geográfica más extensa que un campus, por lo cual su ancho de banda es de mayor capacidad y de mayor velocidad dentro del área delimitada.

Redes **WAN** (Wide Area Network): son redes informáticas que une varias redes locales (LAN) y alcanzan a convertirse en redes globales utilizadas para los satélites, el internet, fibras ópticas públicas, entre otros.

**2.3.2 RASTREADOR DE PAQUETES**

Los analizadores capturan distintos paquetes que se encuentran en circulación a través de una red informática. La captura de dichos datos permite analizar el contenido y la topología de la red para determinar el comportamiento del servidor. Mediante análisis e informes se puede obtener: información determinada, remitente del caso, destinatario de la misma, servidor que ocupa el proceso y tipo de paquete que se está transmitiendo, de tal manera que se pueda administrar la red y monitorizar el tráfico que entra y sale de la organización [4].

Existen diversos rastreadores de paquetes, a continuación, se explican tres:

* Wireshark: antes conocido como el Etheral, es un analizador de protocolos para solucionar y analizar problemas de comunicación de redes, desarrollo de software y protocolos, además de ser herramienta didáctica. Cuenta con una interfaz y opciones de filtrado y organización de información.
* Tcpdump: herramienta de analizar el tráfico que circula por la red, se ejecuta por medio de línea de comando y permite al usuario capturar y mostrar en tiempo real los paquetes que se transmiten y se reciben en la red del computador conectado.
* Microsoft Message Analyzer: aplicación orientada a los expertos de redes y administrador de sistemas para que los usuarios capturen los paquetes de su red local a tiempo real (incluye forma remota) de tal manera que los puedan listar y filtrar por protocolos de red.

**2.3.3 PROTOCOLOS**

Son sistemas de reglas conformadas por estándares, políticas formales y restricciones que definen el intercambio de paquetes de información para comunicar dos servidores o más dispositivos a través de una red. Mediante mecanismos, los dispositivos se identifican y establecen una conexión, de tal manera que los paquetes de datos se transmiten a través de un formato que permite la comprensión de la información de mensajes tanto de entra como de salida, creando una red de comunicación confiable de alto rendimiento [5].

Por capas de acuerdo al modelo TCP/IP, se explican algunos protocolos:

Protocolo de la **capa de acceso** al medio: **ARP** (Adress Resolution Protocolo), este protocolo trabaja con la dirección MAC (físicas-hardware) utilizadas para mapear las direcciones IP en los protocolos de enlace de datos. Este protocolo se aplica cuando se aplica cuando se utiliza el protocolo IP sobre Ethernet.

Protocolo de la **capa de red**: **IP** (Internet Protocol), estándar de especificaciones sobre el funcionamiento de dispositivos conectados que se encuentran en internet. Se conforma en dos partes el direccionamiento, que asegura que cualquier dispositivo de una dirección IP única, y el routing, que determina el camino del trafico de red.

**ICMP** (Internet Control Message Protocol): este protocol reporta los errores y consultas de gestión. Es un protocolo utilizado por dispositivos como router para enviar mensajes de error e información relacionada a las operaciones.

Protocolo de la **capa de transporte**: **TCP** (Transmission Control Protocol) garantiza que los datos se transmitan a través de internet. El protocolo asegura que el trafico llegue a destino de una manera confiable. Otras de sus funciones también son:

* Que no se pierdan los paquetes de datos.
* Control del orden de los paquetes de datos.
* Control de una posible saturación que se llegue a experimentar.
* Prevención de duplicado de paquetes.

Protocolo de **capa de aplicación**: **HTTP** (Hypertext Transfer Protocol), permite que los navegadores y servidores web se comuniquen, los usuarios pueden interactuar con archivos mediante la visualización de las páginas web que cuentan con imágenes, música, videos, texto, etc. El protocolo HTTPS, el cual nos proporciona seguridad punto a punto (entre el cliente y el servidor web). El protocolo HTTPS utiliza el protocolo **TLS** (Transport Layer Security) que también utiliza TCP por encima.

**DNS** (Domain Name System): encargado de interpretar nombres de dominio a direcciones IP. Los DNS, reciben la información de los nombres de domino, de tal manera que las reduce y permite la visualizan de la página web deseada.

**2.3.4 MODELO OSI (International Organization for Standarization)**

Estándar que permite interconectar sistemas de procedencia distinta para intercambiar información. Conformado por 7 capas o niveles de abstracción, cada uno con su propia función, hacen posible la intercomunicación de protocolos distintos al concentrar funciones específicas en cada nivel de operación [6].

Para explicar el modelo OSI, se deben de conocer las siguientes terminologías:

**Sistema:** elemento físico donde se aplica el modelo. El conjunto de maquinas que lo conforman debe de ser capaz de transferir información.

**Modelo:** define la estructura junto a una serie de funciones que realiza el sistema, define el procedimiento normalizado para intercambiar información.

**Nivel**: es el conjunto de funciones especificas que permiten la comunicación agrupada en una entidad que esta relacionada a su vez a otro nivel.

A continuación, se explica cada una de las etapas del modelo:

**Capa 1 Física**: se encarga directamente de los elementos físicos de la conexión. Gestiona los procedimientos a nivel electrónico para que la cadena de bits de información viaje desde el trasmisor al receptor sin alteración alguna.

**Capa 2 Enlace de datos**: proporciona los medios funcionales para establecer la comunicación de los elementos físicos. El direccionamiento físico de los datos, el acceso al medio y especialmente de la detección de errores en la trasmisión.

**Capa 3 Red**: se encarga de la identificación del enrutamiento entre dos o mas redes conectadas. Los datos puedan llegar desde el trasmisor al receptor siendo capaz de hacer las comunicaciones necesarias para que el mensaje llegue.

**Capa 4 Trasporte:** se encarga realizar el transporte de los datos que se encuentran dentro del paquete de transmisión desde el origen al destino. Esto se realiza de forma independiente al tipo de red que haya detectado el nivel inferior. La unidad de información o PDU antes vista, también le llamamos Datagrama si trabaja con el protocolo UDP orientado al envío sin conexión, o Segmento, si trabaja con el protocolo TCP orientado a la conexión.

**Capa 5 Sesión**: Se puede controlar y mantener activo el enlace entre las maquinas que están transmitiendo información. También realiza el mapeo de la dirección de sesión que introduce el usuario para pasarlas a direcciones de transporte con las que trabajan los niveles inferiores.

**Capa 6 Presentación:** Se encarga de la representación de la información transmitida. Traduce las cadenas de caracteres para los usuarios entiendan los datos que tienen en destino los diferentes protocolos.

**Capa 7 Aplicación:** Permiten a los usuarios ejecutar acciones y comandos de sus propias aplicaciones. Permite también la comunicación entre el resto de capas inferiores.

**2.4 SISTEMA DE BASES DE DATOS**

Un sistema de bases de datos permite un tener un gran repositorio de datos con sistemas centralizados que puede ser utilizados simultáneamente por varios usuarios. Con el fin eliminar al máximo de duplicaciones, el sistema se compone de datos integrados, que se vuelven un recurso compartido de la organización y que se pude usar gracias a los diccionarios de datos que tienen la descripción de la información que se almacena [7].

**2.4.1 BASES DE DATOS**

Permite gestionar grandes cantidades de información, a través de estructuras almacena los datos y garantiza la fiabilidad, seguridad y disponibilidad de la misma en una empresa [8].

**2.4.1.1 Características**

Una base de datos permite [9]:

* Integración de toda la información de la organización.
* Persistencia de datos
* Accesibilidad simultanea de distintos usuarios.
* Descripción unificada de los datos e independiente de las ampliaciones.
* Independencia de las aplicaciones respecto a la representación física de los datos.
* Mecanismos para asegurar la integridad y la seguridad de datos.

**2.4.2 SISTEMA GESTOR DE BASES DE DATOS**

Un SGBD (Sistema Gestor de Base de Datos) es un software o programa que sigue un modelo de sistema de bases de datos. El sistema define la estructura, el orden, forma de acceso, permisos, usuarios, objetos, de la base de datos de tal manera que permite al usuario la utilización de lenguajes de definición de datos y lenguaje de manipulación de datos. El más conocido de estos lenguajes es SQL (Structured Query Language, lenguaje de consulta estructurad) [10].

**Ventajas de los SGBD:**

* Gestión estructura de grandes conjuntos de datos.
* Acceso sencillo y eficaz a los datos almacenados.
* Gran Flexibilidad
* Integridad y consistencia de la disponibilidad.

Los **SGBD** se clasifican en base a la forma que administran los datos, en dos tipos [11]:

**Sistemas Gestores de Bases de Datos Relacionales (SQL)**: Establece relaciones o vínculos entre datos, donde cada relación es un conjunto de entidades con mismas propiedades y se compone de registros, representados mayormente en tabla, cuyos valores dependen de ciertos atributos.

**Sistemas Gestores de Bases de Datos No Relacionales (NoSQL):** No requiere estructuras de datos fijas (Tablas). Estos tipos de bases de datos están optimizados específicamente para aplicaciones que requieren grandes volúmenes de datos, baja latencia y modelos de datos flexibles.

**2.5** **INGENIERIA DE SOFTWARE**

La ingeniería de software estudia todo lo relacionado a la orientación metódica, ejecución y conservación del software. Compone la estructura del software que utiliza sistemas, instrumentos y técnicas que se emplean para desarrollar un programa, y a través análisis de antecedentes comprueba el funcionamiento de sistema [12].

**2.5.1** **Características del Software**

Algunas características del software son:

* El Software es un elemento lógico y no físico
* Se puede construir a medida.
* No existen “Piezas de Repuesto” y se puede deteriorar.
* Maximiza los recursos del sistema y debe de estar en constante evolución.

**2.5.2 PROCESO DEL SOFTWARE**:

Un software debe de seguir cuatro fases principales de desarrollo.

* **Especificación**: Que debe de hacer el software y sus especificaciones de desarrollo
* **Desarrollo:** Producción del sistema de software
* **Validación:** Verificar que el software hace lo que el cliente pide.
* **Evolución:** Adaptar el software a las demandas (evolución en el tiempo).

**2.5.3 METODOLOGIAS DE DESARROLLO DE SOFTWARE**

Son un conjunto de técnicas que se aplican para diseñar una estrategia de trabajo en equipo. Una metodología de desarrollo permite organizar el trabajo en etapas, sub-equipos y actividades con el fin de reducir el nivel dificultad en la etapa de desarrollo y agilizar el proceso de las aplicaciones a desarrollar [13].

A continuación, se explican dos tipos de metodologías de desarrollo software [14]:

**Metodologías de desarrollo de Software tradicionales:** Definen en su totalidad los requisitos al inicio de los proyectos de ingeniería de software. Estas metodologías son poco flexibles y no permiten realizar cambios.

Algunas son:

* Waterfall (cascada)
* Prototipado
* Espiral
* Incremental
* Diseño rápido de aplicaciones (RAD)

**Marco de trabajo que ejecutan metodologías de desarrollo de Software ágiles:** La metodología permite adaptar el software a las necesidades que surgen en el desarrollo, siendo flexible y ágil a la hora de realizar cambios, lo que permite entrega de trabajo mucho más eficientes y productos.

[23] Oracle, «docs. Oracle,» 2010. [En línea]. Available: https://docs.oracle.com/cd/E19957-01/820-2981/ipov-10/. [Último acceso: 3 agosto 2021].

Algunas son:

* **Kanbam:** Hace uso de tarjetas para gestionar, de manera visual, la realización de determinados procesos, de tal manera que el proceso se hace mediante tarjetas que indican: Backlog, que toca hacer, que se está haciendo, hecho.
* **Scrum:** Crear tareas que se dividen de acuerdo a requisitos y va iterando en tiempos cortos y fijos para conseguir un resultado en cada iteración.
* **Lean:** está configurado para que pequeños equipos de desarrollo muy capacitados elaboren cualquier tarea en poco tiempo.
* **Programación extrema (XP)**: Crea un fedback constante del cliente, dividiendo el trabajo en 12 conceptos: diseño sencillo, testing, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación.

1. **OBJETIVOS**

**3.1 Objetivo General**

Desarrollar una aplicación web, que permita visualizar gráficos sobre el consumo de ancho de banda en un periodo de tiempo, mediante un analizar de red de tal manera que el cliente tenga un sustento sobre el comportamiento de las solicitudes de su ancho de banda para tomar las decisiones que crea pertinentes.

**3.2 Objetivos Específicos**

* Analizar acerca de aplicaciones o artículos científicos con el tema asociado del proyecto, a través del estado de arte, para el modelamiento del software.
* Diseñar el modelo tecnológico y arquitectónico de la aplicación web mediante el uso de modelos entidad-relación, diagramas de casos uso, estructuras de niveles y grafos de red para estructurar la aplicación.
* Implementar el sistema rastreador de paquetes en aplicación web para tener un seguimiento de las solicitudes que llegan o salen de la red.
* Desarrollar un backend y frontend que conecte el sistema rastreador de paquetes y el sistema de bases de datos al entorno web en el que se desarrolla la aplicación.
* Comprobar el correcto funcionamiento de la aplicación, mediante la aplicación de un conjunto de casos de prueba.

1. **JUSTIFICACIÓN**

Los sniffers son aplicados para capturar la información de la red e indagar en detalle los paquetes de datos que transitan. De tal manera que el usuario pueda analizar qué tipo de información circula dentro de su canal (host-no identificados, aplicaciones de alto rendimiento, protocolos no necesarios, etc.…) y el consumo que genera en el ancho de banda. Con un control total, el usuario que utilice el analizador podrá realizar auditorías, comprobar tráfico que entra y sale de una entidad y monitorizar en base a ello su comportamiento y manejar cada caso que se presente.

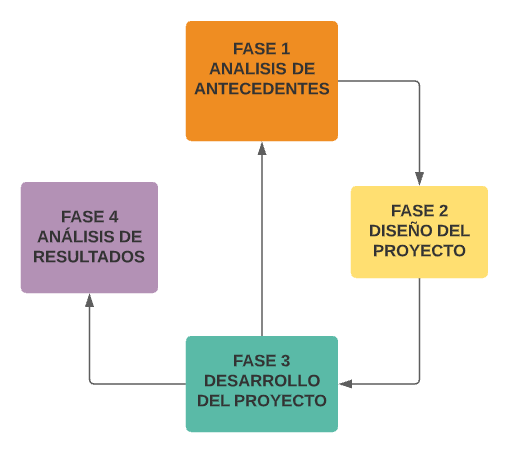
El este proyecto, se buscar analizar los diferentes servicios que se visualizan en el trafico de una red local, de tal manera que sirve como base, para realizar investigaciones mas profundas en el campo de las comunicaciones e implementar más configuraciones al tráfico que solo realizar consultas por parte del usuario. Con una aplicación directa en la seguridad de la red, detectar que dispositivos transitan en el ancho de banda, identifica que comportamientos inusuales presenta el segmento de red y el usuario en cuestión puede tomar decisiones en base a los análisis que se pueden realizar.

1. **METODOLOGÍA**

En este capítulo se explican la metodología del proyecto y la metodología de desarrollo del software.

**5.1 Metodología del Proyecto**

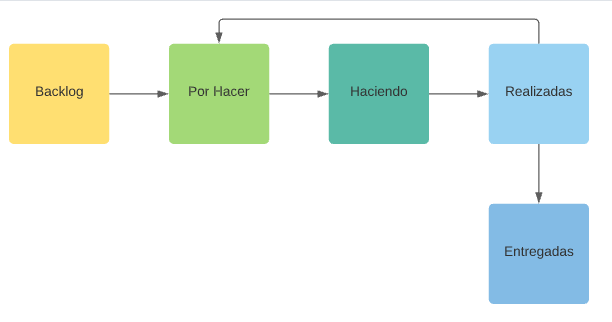
La metodología del proyecto, basada en el ciclo del software, se divide en 4 etapas o fases (ver figura 1), cada fase representa el proceso de desarrollo del proyecto. La primera fase, **exploración de antecedentes,** hace un análisis de los proyectos similares a la propuesta, en artículos de revistas, tesis de grado que proporcionen conocimiento a lo que se va a realizar. La segunda fase, **diseño del proyecto**, define la estructura del software, mediante un modelamiento grafico de la composición del software, se realiza un análisis de la metodología de desarrollo de software que se va a usar para el desarrollo de la aplicación, además que incluye la documentación de los requerimientos del proyecto. La fase siguiente, la tercera, **desarrollo del proyecto**,ejecuta las decisiones que se toman en la etapa del diseño, a través de un entorno de desarrollo y un marco de trabajo llamado **Kanban**, que indica las actividades que abarcan el desarrollo de la aplicación web. Finalmente, la fase 4, **análisis de resultados**, se elabora un informe final, de las pruebas ejecutadas a la aplicación, y los resultados finales, de la entrega del producto final.



*Figura 1 Fases de la metodología del proyecto, base del ciclo del software.*

**5.2** **Metodología del desarrollo del Software**

En la segunda fase de la metodología del desarrollo del proyecto, realizamos el desarrollo del software utilizando el marco de trabajo **Kanbam.** El diseño de la metodología lo aplicamos a través de la aplicación Trello.



*Figura 2 Metodología de Des. De software, basada en el marco de trabajo Kanbam.*

Como se muestra en la aplicación (Ver figura 2), se divide las etapas de desarrollo en etapas que simulan listas, que indican que actividad se va a realizar. Se tienen cuatro columnas que especifican:

* **Backlog:** Compone el conjunto de todas las tareas que componen al proyecto y que se trabajan eventualmente en el “por hacer”.
* **Por Hacer:** Son las actividades que, en el trascurso de esa semana, se tiene que realizar y se fijan las fechas de entrega.
* **Haciendo:** cuando se mueven las tarjetas a esta lista significa que se está trabajando en ellas. Así, todo el equipo puede ver quién está trabajando en qué.
* **Realizadas:** Son las actividades que finaliza el equipo, y están listas para ser entregadas al cliente.
* **Entregadas:** Son las actividades que son finalmente revisadas y aprobadas por el cliente final.

El desarrollo del Backlog, se realiza de acuerdo al cronograma de actividades que se muestra en la siguiente sección, cada actividad se desarrolla de acuerdo a su fase, a continuación, se detalla el backlog, de acuerdo a sus fases.

* Fase de Análisis de Antecedentes
  + Investigación preliminar de aplicaciones similares y requisitos del proyecto.
  + Realizar propuesta del proyecto.
  + Realizar un estado del arte
* Fase de Diseño del Proyecto
  + Realizar requerimientos funcionales y no funcionales.
  + Realizar diagrama de casos de uso.
  + Realizar diagrama entidad-relación y diccionario de datos.
  + Realizar estructura de la aplicación web a nivel de arquitectura tecnológica.
  + Realizar un grafo de redes
* Fase de Desarrollo del Proyecto
  + Implementar un rastreador de paquetes
  + Realizar un backend y frontend que conecte el rastreador de paquetes y es sistema de base de datos al entorno web.
  + Generar reportes estadísticos.
  + Generar un protocolo de pruebas que se implementen en la etapa de desarrollo.
* Fase de Análisis de Resultados
  + Realizar un informe de las pruebas realizadas
  + Entrega del producto final.
  1. **Roles para el desarrollo del proyecto.**

Para el control cada una de las etapas de desarrollo del proyecto y definir un grupo de desarrollo del software, se asignaron las siguientes divisiones con sus respectivos módulos.

*Figura 3. Cuadro jerárquico de la definición de roles para el desarrollo del proyecto*

Como se muestra en la figura 3, se tiene las siguientes asignaciones:

**El Cliente:** encargado de revisar e identificar los requerimientos principales del producto. Los siguientes lideres deben de realizar entregadas de avances del proyecto que se especifican en el cronograma de actividades.

**El Líder de desarrollo del proyecto:** se encarga de revisar las entregas de cada una de las etapas del desarrollo del proyecto, así mismo revisa y controla las etapas de desarrollo del software y las actividades que se estipulas en el marco de trabajo.

**El Líder de control de backlog y lista de actividades:** La persona encargada de este aspecto es la encargada de registrar las actividades que se desarrollan en el marco de trabajo Kanbam a través de la aplicación Trello. Además, lleva los tiempos de cada fase del proyecto que son: El análisis de antecedentes, el diseño del proyecto, el desarrollo del proyecto y los análisis de resultados.

**El Líder de la etapa de desarrollo del Software:** Se encarga solo del desarrollo del software en sus partes que son: El Sniffer, el Backend, el Frontend y la conexión al sistema de base de datos.El equipo de trabajo conforma a:

* **Pedro Javier Bohórquez Camelo.**
* **Paula Andrea Caicedo Escobar.**
* **Luisa Fernanda Monsalve Serrano.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CRONOGRAMA DE ACTIVIDADES** | | | | | | | | | | | | | | | | | | | | | | | |
| **FASE** | **ACTIVIDAD** | **TIEMPO** | | | | | | | | | | | | | | | | | | | **RECURSO** | **INTEGRANTES** | **ENTREGABLE** |
| **JULIO** | | **AGOSTO** | | | | **SEPTIEMBRE** | | | | | | | **OCTUBRE** | | | | | |
| **1** | **2** | **1** | **2** | **3** | **4** | **1** | | **2** | | **3** | | **4** | **1** | **2** | **3** | | | **4** |  | | |
| **1** | Realizar propuesta |  | **26-30** |  |  |  |  |  |  | |  | |  | |  |  | |  |  | | Computador  Artículos  Videos | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Propuesta |
| **1** | Revisión de la propuesta |  |  | **2-6** |  |  |  |  |  | |  | |  | |  |  | |  |  | | Computador  Artículos  Videos | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Observaciones de los docentes |
| **1** | Sustentación propuesta |  |  |  | **9-13** |  |  |  |  | |  | |  | |  |  | |  |  | | Computador  Artículos  Videos | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Propuesta completa  Observaciones de los docentes |
| **1** | Realizar un estado del arte |  |  |  |  | **16-18** |  |  |  | |  | |  | |  |  | |  |  | | Computador  Artículos  Videos | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Estado del Arte |
| **2** | Realizar requerimientos funcionales y no funcionales |  |  |  |  | **19-21** |  |  |  | |  | |  | |  |  | |  |  | | Computador  Artículos  Videos | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Requerimientos no funcionales |
| **2** | Modelar diagrama de casos de uso |  |  |  |  |  | **27-28** |  |  | |  | |  | |  |  | |  |  | | StarUML  Workbench | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Diagrama de casos de uso |
| **2** | Modelar diagrama entidad-relación y diccionario de datos |  |  |  |  |  | **27-28** |  |  | |  | |  | |  |  | |  |  | | StarUML  Workbench | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Diagrama entidad-relación  Diccionario de datos |
| **2** | Modelar estructura de la aplicación web. |  |  |  |  |  | **29-31** |  |  | |  | |  | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Modelo de la aplicación web |
| **2** | Primer avance |  |  |  |  |  |  | **1-3** |  | |  | |  | |  |  | |  |  | | Lucidchart  Word  Computador | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | 30% del proyecto |
| **2** | Realizar grafo de redes |  |  |  |  |  |  | **6-10** |  | |  | |  | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Grafo de redes |
| **3** | Realizar el sniffer |  |  |  |  |  |  | **6-10** |  | |  | |  | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Software |
| **3** | Realizar informe avance proyecto |  |  |  |  |  |  |  | **13-17** | |  | |  | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | 50% del proyecto |
| **3** | Realizar el backend |  |  |  |  |  |  |  | **13-17** | | **20-24** | |  | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Software |
| **3** | Realizar el frontend |  |  |  |  |  |  |  | **13-17** | | **20-24** | |  | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Software |
| **3** | Generar graficas estadísticas |  |  |  |  |  |  |  |  | |  | | **27-30** | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Graficas estadísticas |
| **3** | Generar pruebas del sistema |  |  |  |  |  |  |  |  | |  | | **27-30** | |  |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Diseño de los casos de prueba |
| **4** | Realizar informe de pruebas |  |  |  |  |  |  |  |  | |  | |  | | **4-8** |  | |  |  | | Word  Recursos bibliográficos  Aplicativo realizado | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Informe |
| **4** | Entrega de producto final |  |  |  |  |  |  |  |  | |  | |  | | **4-8** |  | |  |  | |  | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | 100% del proyecto |
| **4** | Pre-sustentar el proyecto |  |  |  |  |  |  |  |  | |  | |  | |  | **11-15** | |  |  | | Diapositivas  Video | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Diapositivas  Aplicativo web |
| **4** | Realizar informe final |  |  |  |  |  |  |  |  | |  | |  | |  |  | | **18-21** |  | | Word  Recursos bibliográficos  Aplicativo realizado | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Informe |
| **4** | Realizar diapositivas para exponer |  |  |  |  |  |  |  |  | |  | |  | |  |  | | **18-21** |  | | PowerPoint | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Diapositivas |
| **4** | Realizar video explicativo |  |  |  |  |  |  |  |  | |  | |  | |  |  | |  | **23-24** | | Apowersoft  Filmora | Pedro Bohórquez  Paula Caicedo  Luisa Monsalve | Video  Diapositivas  Aplicativo web |

**REFERENCIAS BIBLIOGRÁFICAS**

[1] IBM, “Arquitectura de tres niveles”, 2021. [En Línea]. https://www.ibm.com/docs/es/was-nd/9.0.5?topic=overview-three-tier-architectures. [Accedido: 27-08-2021]

[2] F. Barraza, Modelado y Diseñado de Arquitectura de Software. [En línea]. Disponible en: http://cic.javerianacali.edu.co/wiki/lib/exe/fetch.phpmedia=materias:s2\_conceptosdemodelado.pdf [Accedido el: 27- Jul- 2021].

[3] R. Belloso, Introducción A La Gestión De Redes. 1 edición. 2010 [En línea]. Disponible en: https://www.urbe.edu/info-consultas/web-profesor/12697883/archivos/planificacion-gestion-red/Unidad-I.pdf [Accedido el: 27- Jul- 2021].

[4] L. Gorgona, Teoría de Redes de Computadoras. 2010 [En línea]. Disponible en: https://www.oas.org/juridico/spanish/cyber/cyb29\_computer\_int\_sp.pdf [Accedido el: 27- Jul- 2021].

[5] UnderCoder, “Las 5 mejores herramientas analizadores de red y sniffers”, 2016. [En Línea]. https://underc0de.org/foro/pentest/las-5-mejores-herramientas-analizadoras-de-red-y-sniffers!/. [Accedido: 27-08-2021].

[6] A. Interserver Protocolos de red comunes, 2016. [En línea]. Disponible en: https://www.interserver.net/tips/kb/common-network-protocols-ports/. [Accedido el: 27- Jul- 2020].

[7] J. Castillo Modelo OSI, 2018. [En línea]. Disponible en: https://www.profesionalreview.com/2018/11/22/modelo-osi/#Que\_es\_el\_modelo\_OSI. [Accedido el: 27- Jul- 2020].

[8] R. Margaret, «kionetworks,» 04 noviembre 2020. [En línea]. Available: https://www.kionetworks.com/blog/data-center/protocolos-de-comunicaci%C3%B3n-de-redes. [Accedido: 21 agosto 2021].

[9]Ranchal, «ComputerMc,» 23 marzo 2017. [En línea]. Available: https://www.muycomputer.com/2017/03/23/redes-en-windows/. [Accedido: 21 Agosto 2021].

[10] D. R. B. Chacín, Planificación y Gestión de Red, Maracaibo, Venezuela: 1ra. Edición, 2010.

[11] C. J. Date an Introduction to Database Systems, 1997. [En línea]. Disponible en: https://www.aiu.edu/cursos/base%20de%20datos/pdf%20leccion%201/lecci%C3%B3n%201.pdf I. [Accedido el: 27- Jul- 2020].

[12] D. Morato, Computer Networking, 3 Edición, 2004.

[13] J. Ticona, Protocolo ARP, Santa María: Universidad.

[14] Castillo, A. “Modelo OSI: para que se utiliza”. 2018. [En Línea] https://www.profesionalreview.com/2018/11/22/modelo-osi/#Que\_es\_el\_modelo\_OSI. [Acceded: 27/08/2021].

[15] Antoniou. A. “Networking Basics: TCP vs UDP, TCP/IP vs OSI Model & More”.2011. [En Línea] https://www.pluralsight.com/blog/it-ops/networking-basics-tcp-udp-tcpip-osi-models. [Accedido: 27/08/2021].

[16] Hernández, E. “Comparación de los modelos OSI y TCP/IP”. 2018. [En Línea] https://www.uaeh.edu.mx/scige/boletin/huejutla/n10/r1.html. [Accedido: 27/08/2021].

[17] C.J. Date (1993), Introducción a los Sistemas de Bases de Datos Volumen I, Quinta Edición Addison-Wesley Iberoamericana Sexta Edición en 1995 (en inglés, por Addison-Wesley).

[18] R. Elmasri, S.B. Navathe (1997), Sistemas de Bases de Datos. Conceptos fundamentales, Segunda Edición, AddisonWesley Iberoamericana, Tercera Edición en 1999 (en inglés, por Addison-Wesley.

[19] T. Connolly, C. Begg, A. Strachan (1996), Database Systems. A Practical Approach to Design, Implementation and Management, Addison-Wesley Segunda Edición en 1998.

[20] Pons Capote, Olga, Introducción a las bases de datos. El modelo relacional, Thomson Paraninfo, S.A., 2005.