**FASE DE PLANEACIÓN:**

**Planear el proyecto usando la metodología scrum.**

i: Roles

PO: Samuel Luna

Scrum Master: Jesus Peña

Equipo Scrum: Samuel Luna y Jesus Peña.

ii: Metologia

Usaremos trello para poder planear las tareas, con sus correspondiente criterios de evaluación e historias de usuario y poder plantear un product backlog y un sprint backlog correspondientes para cada sprint planeado

**Determinar RQF y RQNF.**

**ANEXO 1. REQUISITOS FUNCIONALES PARA LA CLASE PRODUCTO**

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF001 | Nombre: Registrar Producto en la Base de Datos |
| Descripción: El sistema permitirá a los empleados registrar un nuevo producto en la base de datos, ingresando detalles como el nombre del producto, categoría, precio, proveedor y ciudad de origen. Esta información será almacenada de manera estructurada para facilitar su consulta y análisis posterior. |
| Usuarios: Administrador, Personal de Compras |

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF002 | Nombre: Actualizar Producto en la Base de Datos |
| Descripción: Los usuarios podrán modificar los datos de los productos en la base de datos, Actualizando, sus correspondientes atributos |
| Usuarios: Administrador, Personal de compras |

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF003 | Nombre: Eliminar Producto de la Base de Datos |
| Descripción: El sistema proporcionará una funcionalidad para que los empleados puedan eliminar un producto de la base de datos en caso de que este ya no se encuentre la cantidad necesaria o no sea suministrado por los proveedores. |
| Usuarios: Administrador, Personal de Compras |

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF004 | Nombre: Consultar producto en la Base de Datos |
| Descripción: Los empleador podrán buscar un producto ya sea por: Nombre, Tipo, Categoría, Especificaciones Técnicas, Costo del producto |
| Usuarios: Administrador, Personal de Compras |

**ANEXO 2. REQUISITOS NO FUNCIONALES PARA LA CLASE PRODUCTO**

|  |  |
| --- | --- |
| CÓDIGO | requisitos NO funcionales |
| RQNF0XX | Nombre: |
|  |

**ANEXO 3. REQUISITOS FUNCIONALES PARA LA CLASE PROVEEDOR**

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF001 | Nombre: Registrar Proveedor en la Base de Datos |
| Descripción: El sistema permitirá a los empleados registrar un nuevo proveedor en la base de datos, ingresando detalles como el nombre del proveedor, ubicación proveedor, calidad producto, tiempo entrega, capacidad entrega, costo del producto. Esta información será almacenada de manera estructurada para facilitar su consulta. |
| Usuarios: Administrador, Personal de Compras |

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF002 | Nombre: Actualizar Proveedor en la Base de Datos |
| Descripción: Los usuarios podrán modificar los datos de los proveedores en la base de datos, Actualizando, sus correspondientes atributos |
| Usuarios: Administrador, Personal de compras |

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF003 | Nombre: Eliminar Proveedor de la Base de Datos |
| Descripción: El sistema proporcionará una funcionalidad para que los empleados puedan eliminar un proveedor de la base de datos en caso de que este ya no se encuentre en contacto con la empresa o no pueda suministrar los productos. |
| Usuarios: Administrador, Personal de Compras |

|  |  |
| --- | --- |
| CÓDIGO | requisitos funcionales |
| RQF004 | Nombre: Consultar Proveedor en la Base de Datos |
| Descripción: Los empleados podrán buscar un proveedor ya sea por: Nombre, Ubicación proveedor, Calidad producto, Capacidad entrega, Tiempos Entrega, Costo del producto |
| Usuarios: Administrador, Personal de Compras |

**ANEXO 4. REQUISITOS NO FUNCIONALES PARA LA CLASE PROVEEDOR**

|  |  |
| --- | --- |
| CÓDIGO | requisitos NO funcionales |
| RQNF0XX | Nombre: |
|  |

**Solución por Integrante**

**Solución Samuel:** Implementación de una Base de Datos Relacional

Crear una base de datos relacional que organice de manera estructurada la información de proveedores, productos, ciudades, precios, calidad de servicio y tiempos de entrega. Esto se logrará a través de un diseño de tablas bien definidas y relaciones claras entre ellas. La base de datos permitirá realizar consultas rápidas y precisas, facilitando el acceso a información relevante para la toma de decisiones.

**Solución Jesús:** Implementación de una Base de Datos NoSQL para Flexibilidad y Escalabilidad.

Optar por una base de datos NoSQL, como MongoDB, que permita almacenar los datos de proveedores, productos, ciudades, precios, calidad de servicio y tiempos de entrega en un formato de documentos flexibles. Este enfoque permitirá adaptarse a posibles cambios en los datos o la estructura sin necesidad de redefinir esquemas rígidos. Los datos se pueden organizar en documentos JSON, facilitando la inclusión de información adicional en el futuro sin reestructurar la base de datos.

**Diagrama de clases**

**Texto

Descripción generada automáticamente con confianza baja**

**Modelo entidad relación MER**

