



FACULTAD DE INGENIERÍA

Carrera de Ingeniería de Sistemas Computacionales

“IMPLEMENTACIÓN Y DESARROLLO WEB DE LOS MODULOS DE GESTION COMERCIAL PARA LA EMPRESA UNILENE SAC., LIMA, 2025”

**Trabajo de suficiencia profesional para optar al título
profesional de:**

Ingeniera de Sistemas Computacionales

Autor:

Katherine Yessenia Santamaria Mendoza

Asesor:

Mg. Ing. Guido Trujillo Valdivieso

<https://orcid.org/0000-0002-3019-6599>



Lima - Perú

2025

Informe de Similitud

KATHERINE YESSENIA SANTAMARIA MENDOZA

**KatherineSantamaria_N00144462_COR-F-REC-VAC-05.17_8
_Fotmato Documentacion Titulacion.docx**

-  Suficiencia Profesional
-  17 Taller de tesis
-  Asesores

Detalles del documento

Identificador de la entrega

trn:oid::1:3302349271

Fecha de entrega

25 jul 2025, 7:47 a.m. GMT-5

Fecha de descarga

25 jul 2025, 7:51 a.m. GMT-5

Nombre de archivo

KatherineSantamaria_N00144462_COR-F-REC-VAC-05.17_8_Fotmato_Documentacion_Titulacion.docx

Tamaño de archivo

6.9 MB

78 Páginas

9714 Palabras

56.772 Caracteres



Página 1 of 81 - Portada

Identificador de la entrega trn:oid::1:3302349271



Página 2 of 81 - Descripción general de integridad

Identificador de la entrega trn:oid::1:3302349271

1% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- ▶ Bibliografía

Tabla de contenido

Índice de tablas	6
Índice de Figuras.....	7
RESUMEN EJECUTIVO.....	12
CAPÍTULO I. INTRODUCCIÓN.....	13
CAPÍTULO II. MARCO TEÓRICO	15
CAPÍTULO III. DESCRIPCIÓN DE LA EXPERIENCIA	22
CAPÍTULO IV. RESULTADOS	57
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....	73
REFERENCIAS	75
ANEXOS	77

Índice de tablas

Tabla 1: Requerimientos Funcionales y No Funcionales	25
Tabla 2: Procedimientos almacenados desarrollados en el módulo de Gestión de Pedidos	29
Tabla 3 Comandos Git utilizados en el flujo de trabajo por ticket. Fuente: Elaboración propia.....	54

Índice de Figuras

Figura 1: Organigrama Institucional.....	14
Figura 2: Diagrama To-Be del proceso de Gestión de Pedidos en el sistema Satélite.	26
Figura 3: Relación entre las tablas CO_Documento y CO_DocumentoDetalle. Fuente: Elaboración propia.....	28
Figura 4: Diagrama de Secuencia: Flujo desde el Frontend hasta la Ejecución del Store Procedure	30
Figura 5 : Sentencias SQL para el registro de permiso y opción de menú del módulo “Gestión de Pedidos” en la base de datos SateliteCore. Fuente: Elaboración propia.	32
. Figura 6: Código de los métodos en el controlador del módulo Gestión de Pedidos. Fuente: Elaboración propia	34
Figura 7 : Código de los métodos en el IService del módulo Gestión de Pedidos. Fuente: Elaboración propia.....	35
Figura 8 : Código de los métodos en el Service del módulo Gestión de Pedidos. Fuente: Elaboración propia.....	36
Figura 9Código de los métodos en el IRepository del módulo Gestión de Pedidos.	36
Figura 10 : Código de los métodos en el IRepository del módulo Gestión de Pedidos. Fuente: Elaboración propia.	37
Figura 11Código para generar el componente dentro de la carpeta comercial pedido. Fuente: Elaboración propia.....	38
Figura 12 : Componentes creados dentro de la carpeta comercial pedido.	38

Figura 13 : Rutas incorporadas manualmente para reconocer el componente. Fuente: Elaboración propia.....	39
Figura 14 : Rutas configuradas manualmente para reconocer la ruta del servidor.	40
Figura 15 : HTML en el cual se muestra el uso del fromControlName. Fuente: Elaboración propia.....	41
Figura 16 : TS en el cual se muestra la creación de los controles para el HTML. Fuente: Elaboración propia.....	42
Figura 17 : HTML en el cual se muestra la llamada al método en el TS. Fuente: Elaboración propia.....	43
Figura 18 : Implementación de la tabla dinámica con ngx-datatable para la visualización de pedidos en el módulo Gestión Comercial. Fuente: Elaboración propia	45
Figura 19 Código utilizado para llamar a diversas funcionalidades dentro de la tabla. Fuente: Elaboración propia.....	46
Figura 20 Código utilizado los iconos que se utilizan dentro de la tabla. Fuente: Elaboración propia.....	46
Figura 21 : Código utilizado para llamar el método que se encuentra en el TS. Fuente: Elaboración propia.....	47
Figura 22 : Método filtrar() en el componente lista-pedido.component.ts, encargado de obtener los filtros, invocar el servicio y asignar los resultados a la tabla. Fuente: Elaboración propia.....	47
Figura 23 : Método obtenerPedidos() en pedido.service.ts, que realiza la llamada HTTP POST al endpoint del backend con los filtros definidos.	48
Figura 24 : Endpoint FiltrarPedidos() en PedidoController.cs, encargado de recibir la solicitud desde el frontend y delegar la lógica al servicio. Fuente: Elaboración	

propia	49
Figura 25 : Método FiltrarPedidos() en la interfaz IPedidoService. Fuente: Elaboración propia.....	50
Figura 26 : Implementación del método FiltrarPedidos() en la clase PedidoService. Fuente: Elaboración propia.	50
Figura 27 : Declaración del método FiltrarPedidos() en la interfaz IPedidoRepository. Fuente: Elaboración propia.....	50
Figura 28 : Implementación del método FiltrarPedidos() en la clase PedidoRepository, donde se ejecuta el procedimiento almacenado. Fuente: Elaboración propia.....	51
Figura 29 : Procedimiento almacenado usp_satelite_listar_clientes que filtra los pedidos según los parámetros recibidos. Fuente: Elaboración propia.....	52
Figura 30 : Códigos para ingresar a la carpeta raíz de las fuentes.....	54
Figura 31 : Códigos para compilar el proyecto y subir las fuentes al servidor...	54
Figura 32 : Listar Pedido	57
Figura 33 : Buscar Pedido	58
Figura 34 : Visualizar Pedido	58
Figura 35 : Nuevo Pedido	59
Figura 36 : Nuevo Pedido desde una Cotización.....	59
Figura 37 : Elegir Cotización	59
Figura 38 : Eliminar y/o Agregar Items al Pedido.....	60
Figura 39: Nuevo Comentario	61
Figura 40 : Guardar Comentario.....	61

Figura 41 : Guardar Pedido con Cotización	61
Figura 42 : Crear Pedido Sin Cotización	62
Figura 43 : Elegir Cliente	63
Figura 44 : Agregar Item	63
Figura 45 : Elegir Item.....	64
Figura 46 : Eliminar Item y/o Agregar Comentario	64
Figura 47 : Guardar Pedido Sin Cotización.....	65
Figura 48 : Aprobar Pedido	65
Figura 49 : Pedido Con Lotes Asignados	66
Figura 50 : Seleccionar Lote.....	66
Figura 51 : Guardar Aprobar Pedido	66
Figura 52 : Pantalla para ingresar a modificar Pedido.....	67
Figura 53 : Guardar Modificación Pedido	67
Figura 54 : Documento Pedido	68
Figura 55 : Documento Factura Electrónica.....	68
Figura 56 : Dashboard Seguimiento de Pedido página 1.....	69
Figura 57 : Dashboard Seguimiento de Pedido página 2.....	69
Figura 58 : Dashboard Seguimiento de Pedido página 3.....	70
Figura 59 : Dashboard Seguimiento de Pedido página 4.....	70
Figura 60 : Dashboard Seguimiento de Pedido página 5.....	71
Figura 61 : Dashboard Seguimiento de Pedido página 6.....	71

Figura 62 : Dashboard Seguimiento de Pedido página 7..... 72

Figura 63 : Dashboard Seguimiento de Pedido página 8..... 72

RESUMEN EJECUTIVO

La presente experiencia profesional se desarrolló en la empresa UNILENE S.A.C., en el área de Tecnología de la Información, donde asumí el reto de diseñar e implementar el módulo web de **Gestión de Pedidos** como parte del sistema interno Satélite. Este módulo reemplazó al sistema SPRING, una solución de escritorio limitada y dependiente de un proveedor externo. El principal objetivo fue modernizar el proceso comercial, brindando mayor trazabilidad, seguridad, flexibilidad y autonomía al área usuaria.

Para lograrlo, se aplicó el modelo de análisis de procesos To-Be y se desarrolló una arquitectura basada en capas, utilizando tecnologías como Angular, .NET Core, SQL Server, GitLab y Jira. La lógica de negocio fue centralizada en procedimientos almacenados, lo que permitió reutilizar componentes del sistema legado y optimizar los tiempos de desarrollo. Además, se implementaron buenas prácticas de control de versiones y despliegue progresivo por ticket funcional.

Como resultado, se obtuvo una solución adaptable, segura y completamente integrada a los procesos internos de la empresa. Esta experiencia permitió aplicar competencias clave como análisis de sistemas, programación full stack, documentación técnica, trabajo colaborativo y gestión de proyectos tecnológicos, consolidando mi formación como **ingeniera de sistemas** especializada en el desarrollo de soluciones empresariales.

NOTA

El contenido de la investigación no se encuentra disponible en **acceso abierto** por determinación de los propios autores, en concordancia con en el Texto Integrado del Reglamento RENATI (artículo 12), la Directiva N°048-2020-CONCYTEC-P que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto (ALICIA) administrado por el pliego Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica - CONCYTEC y la Ley N° 29733, Ley de Protección de Datos Personales.

REFERENCIAS

Bizagi. (2023). *BPMN for business process modeling*. Recuperado de <https://www.bizagi.com>

Chen, X., Zhang, Y., & Huang, L. (2023). Challenges and solutions for IT infrastructure scalability in healthcare systems. *Journal of Healthcare Engineering*, 2023, 1-12. <https://doi.org/10.1155/2023/9856421>

Google. (2021). *Angular 11 documentation*. Recuperado de <https://angular.io/docs>

ISO. (2015). *ISO 9001:2015 – Quality management systems — Requirements*. International Organization for Standardization.

ISO. (2016). *ISO 13485:2016 – Medical devices — Quality management systems — Requirements for regulatory purposes*. International Organization for Standardization.

Kim, D., & Solomon, M. G. (2022). *Fundamentals of information systems security* (3rd ed.). Jones & Bartlett Learning.

Martin, R. C. (2020). *Clean architecture: A craftsman's guide to software structure and design* (2nd ed.). Prentice Hall.

Microsoft. (2021). *Building web APIs with ASP.NET Core*. Recuperado de <https://learn.microsoft.com/en-us/aspnet/core/web-api/>

Microsoft SQL Docs. (2022). *SQL Server security and auditing*. Recuperado de <https://learn.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit>

Node.js Foundation. (2023). *Node.js v16.x documentation*. Recuperado de <https://nodejs.org/en/docs/>

Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley.

Kotler, P., & Keller, K. L. (2016). *Dirección de marketing* (15.a ed.). Pearson Educación.

Microsoft. (2021). *C# documentation*. Microsoft Docs.
<https://learn.microsoft.com/en-us/dotnet/csharp/>

Microsoft. (2021). *REST APIs - Best practices for designing RESTful APIs*. Microsoft Learn.
<https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>

Richardson, L., & Ruby, S. (2007). *RESTful Web Services*. O'Reilly Media.

Richards, M. (2015). *Software architecture patterns*. O'Reilly Media.
<https://www.oreilly.com/library/view/software-architecture-patterns/9781491924242/>

Welling, L., & Thomson, L. (2017). *PHP and MySQL Web Development* (5th ed.). Addison-Wesley.