



**DKT WS  
SYSTEM**

# Documentación de arquitectura

Programación de Aplicaciones Visuales I  
3k1 - 2019

## Integrantes

- |   |                    |       |
|---|--------------------|-------|
| - | Avalle, Isaías     | 74973 |
| - | Maurino, Francisco | 77629 |
| - | Zapata, Mauricio   | 76034 |

# Índice

<b>Índice</b>	<b>2</b>
<b>Objetivos</b>	<b>3</b>
<b>Acuerdos de funcionalidad</b>	<b>3</b>
<b>Tecnologías utilizadas</b>	<b>3</b>
<b>Herramientas utilizadas</b>	<b>3</b>
<b>Modalidad de trabajo</b>	<b>4</b>
<b>Arquitectura</b>	<b>4</b>
Distribución en capas	4
Transacción	5
Validaciones	5
<b>Repositorio y base de datos</b>	<b>6</b>

## Objetivos

Gestionar y administrar órdenes de trabajo, llevar el control de stock de repuestos, administrar empleados, clientes y sus respectivos vehículos, para la operatoria de un taller mecánico automotriz, generando información resultante de dicha gestión.

## Acuerdos de funcionalidad

A lo largo del proceso de desarrollo del sistema, se han respetado los acuerdos pactados en cuanto a la funcionalidad solicitada, cumpliendo con la totalidad de los mismos, excepto por la gestión de presupuestos. De las dos transacciones declaradas en la documentación, sólo se implementó la primera, debido a la complejidad de la misma y la cantidad de tablas involucradas. La carga de clientes y sus respectivos vehículos se realiza por medio de dos operaciones separadas sobre las tablas "Clientes" y "Vehículos".

## Tecnologías utilizadas

Para la persistencia de datos, se optó por el motor de bases de datos de Microsoft, SQL Server 2008, debido a la facilidad para integrar el mismo con la herramienta Visual Studio 2019.

## Herramientas utilizadas

Para facilitar el desarrollo en conjunto se utilizó GitHub, que ofrece la posibilidad de clonar repositorios en la máquina local y compartir cambios de forma mucho más ágil.

Como entorno de desarrollo, se utilizó la última versión disponible de Visual Studio, 2019. Dicho entorno resultó de gran utilidad y brindó una experiencia más amigable con el lenguaje, facilitando en gran medida el diseño de la Interfaz Gráfica de Usuario (GUI), las conexiones con la base de datos y la creación de reportes. La herramienta de creación de reportes presentó varios inconvenientes en su configuración, pero una vez resueltos, dichos reportes se pudieron diseñar e implementar de manera muy ágil.

## Modalidad de trabajo

La división de las tareas llevadas a cabo a lo largo del proyecto, se realizó de manera equitativa, adaptando la delegación de las mismas a la disponibilidad de tiempo de cada integrante.

## Arquitectura

### Distribución en capas

Se ha decidido estructurar el producto haciendo uso de una arquitectura “en capas” o estratificada, compuesta por 4 niveles:

- Capa de Interfaz de Usuario: contiene todos los formularios que componen la interfaz.
- Capa de Servicios: actúa como intermediaria entre la capa de Acceso a Datos, y la capa de interfaz de usuario.
- Capa de Entidades: contiene la definición de las entidades que componen el dominio.
- Capa de acceso a datos: ofrece servicios de interacción con la base de datos, como ser: mapeo de objetos a registros de la base de datos, establecimiento y cierre de conexiones, además de inserciones y actualizaciones en las tablas de la base de datos.

La arquitectura seleccionada ofrece varios beneficios, entre ellos:

- Código bien organizado, clases con responsabilidades bien definidas, logrando un alto nivel de cohesión, lo que lleva a un código mucho más sencillo de mantener y adaptar en el futuro
- Dependencias mínimas entre capas, lo que deriva en un bajo nivel de acoplamiento entre las mismas, además permite obtener un producto más robusto y eficiente ante amenazas de seguridad.

## Transacción

La transacción corresponde a la creación de una Orden de Trabajo (OT), la cual es una clase con numerosos componentes internos. Al llevar a cabo la operación de creación de una OT se realizan cambios en tres tablas de la base de datos:

- Se realiza un INSERT en la tabla Ordenes con la nueva OT y los datos ingresados por el usuario.
- Se realiza un INSERT por cada Trabajo efectuado en la OT en la tabla DetallesOrdenTrabajo.
- Se realiza un INSERT por cada repuesto utilizado en cada trabajo en la tabla RepuestosxTrabajos.
- Por último, se realiza un UPDATE del stock de cada repuesto utilizado en los trabajos en la tabla Repuestos.

## Validaciones

Tras un exhaustivo proceso de testing, se han implementado todas las validaciones necesarias para evitar situaciones que lleven al sistema a un estado inestable o inconsistente, impidiendo de esta forma que acciones de los usuarios deriven en excepciones o errores.

Dichas validaciones incluyen: el llenado de campos obligatorios, validaciones de tipos de datos, y de cantidad máxima de caracteres.

## Repositorio y base de datos

Link a repositorio del proyecto: <https://github.com/Laikos38/TallerCordoba>

En el repositorio se pueden observar todos los commits realizados por parte de los alumnos, así como también la creación y cierre de distintos issues para la corrección de bugs o detalles a mejorar.

Dentro del repositorio se encuentra un archivo "script.sql" para la creación de la base de datos del sistema. El archivo debe ser ejecutado bajo el directorio de conexión "localhost/SQLEXPRESS" para que los reportes funcionen correctamente.

Bajo el directorio "Documentación" se puede encontrar los siguientes archivos:

- TPI Formulario, el cual incluye el formulario inicial del proyecto con su objetivo, alcances, casos de uso y normalización de tablas de base de datos.
- Manual DKT WS System, el cual provee una descripción paso a paso de las distintas funcionalidades del sistema.
- Documentación DKT WS System, incluyendo la descripción de arquitectura del sistema, las tecnologías utilizadas y la modalidad de trabajo.

Bajo el directorio "Logo" se encuentra un archivo de photoshop con el logo del sistema.

En el directorio "Tablas" se encuentra un archivo .pdf con las tablas de base de datos normalizadas del sistema.

Por último, bajo el directorio "TallerAutos" se encuentra el proyecto de Visual Studio.