

5.1.4.1 Manual técnico

Introducción

El manual técnico del sistema de soporte para el manejo de averías en Sitec Seguridad contiene información detallada sobre la estructura, los componentes y funcionalidades del sistema para su correcto uso.

Objetivos

Objetivo general

Proporcionar a los usuarios una comprensión exhaustiva de la estructura, componentes y funcionalidades del sistema de soporte para el manejo de averías, con el fin de facilitar la correcta implementación, operación y gestión eficiente del proceso.

Objetivos específicos

Comprender la arquitectura y diseño del sistema de soporte de averías, incluyendo su infraestructura tecnológica subyacente y las interconexiones entre módulos.

Familiarizarse con los componentes clave del sistema, tales como bases de datos, herramientas de seguimiento de clientes, generación de reportes y procesamiento de datos.

Dominar las funcionalidades del sistema que facilitan la gestión integral de averías, como la generación automatizada de creaciones de casos, seguimiento de leads, atención cercana al usuario que reporta fallas en los productos de la empresa para aperturas y cierres de casos, garantizando una experiencia fluida y efectiva para clientes y personal involucrado en el proceso de soporte.

Alcance

Este manual técnico abarca de manera detallada la implementación, funcionamiento y mantenimiento del sistema de soporte para gestión de averías, brindando información esencial para asegurar su correcta utilización y administración. Los entregables clave contemplados en este alcance son descritos a continuación.

Descripción de la aplicación: se proporcionará una descripción completa de la aplicación, incluyendo sus objetivos, contexto de uso y ventajas para la organización.

Arquitectura del sistema: se detallará la arquitectura de la aplicación, describiendo la distribución de módulos, componentes y sus interconexiones. Esto permitirá comprender cómo los diferentes elementos del sistema colaboran para lograr una gestión eficiente.

Requisitos del sistema: se especificarán los requisitos técnicos y de software necesarios para implementar y ejecutar la aplicación de manera adecuada, asegurando que el entorno de operación cumpla con los criterios necesarios.

Funcionalidades del sistema: se enumerarán y describirán en detalle las funcionalidades disponibles en la aplicación, tales como la gestión de datos importantes para la creación de casos, seguimiento de clientes y generación de reportes.

Interfaz de usuario: se proporcionará una guía exhaustiva sobre la interfaz de usuario de la aplicación, incluyendo capturas de pantalla y explicaciones para que los usuarios puedan navegar eficazmente a través de las distintas secciones y funciones.

Procedimientos operativos: se presentarán las instrucciones y procesos necesarios para llevar a cabo las tareas cotidianas utilizando la aplicación, desde agregar nuevos datos al sistema según la opción correspondiente hasta cerrar un caso de soporte y generar informes de seguimiento.

Mantenimiento y actualizaciones: se ofrecerán pautas sobre cómo mantener y actualizar la aplicación a lo largo del tiempo, incluyendo la resolución de problemas comunes y la implementación de mejoras y nuevas características.

Documentación técnica adicional: se proporcionará documentación técnica complementaria, como diagramas de flujo, esquemas de bases de datos y cualquier otro recurso que facilite la comprensión y el desarrollo continuo del sistema.

Ambiente de la aplicación

Para que la aplicación del sistema de soporte para manejo de averías funcione correctamente se ha definido los siguientes requerimientos técnicos.

Ambiente del servidor

El servidor donde se albergará la aplicación debe contar con las características de sistema operativo Windows server (última versión), dos procesadores Intel Core i5, RAM de 32 GB mínimo, tarjetas de red (contemplar las necesarias por eventuales situaciones inesperadas) y disco duro de mínimo 10 TB.

Para el servidor de base de datos se debe contar con lo siguiente: sistema operativo Windows server (última versión), base de datos SQL Server (última versión), cinco procesadores Intel Core i5, RAM de 32 GB mínimo (128 GB óptimo), tarjetas de red (contemplar las necesarias por eventuales situaciones inesperadas), conexión con servidor SAN (red de área de almacenamiento), disco duro de mínimo 10 TB y espacio mínimo disponible para almacenar base de datos de 1 TB.

Ambiente del cliente

Los equipos clientes deben contar con las siguientes características sistema operativo Windows 10 u 11 Professional de 64 bits en español, última versión, procesador Intel. El procesador debe ser mínimo de dos núcleos nativos con velocidad mínima de 2.93 GHz, mínimo 4 GB de memoria RAM DDR3 a 1333MHz expandible a 16 GB sin que se tenga que remover los 4 GB requeridos como mínimo, un disco duro de al menos 1 TB, interfaz SATA (interfaz de transmisión en serie más avanzada) con capacidad de transferencia de 3.0Gb/s y velocidad de rotación de 7200rpm (revoluciones por minuto), con 16MB de caché interno al disco, tarjeta de video SVGA color con un mínimo de 512 MB en video independientes de la memoria RAM del equipo y de arquitectura PCI Express, monitor de 19 pulgadas como mínimo tipo WIDE, VGA/DVI, EPG y navegador web Google Chrome (última versión).

Descripción general de los módulos

Para la descripción general de los módulos se hace referencia a la sección de **¡Error! Nose encuentra el origen de la referencia..**

Instalación

A continuación, se detallan los pasos para la instalación y ejecución de la aplicación de sistema de soporte para manejo de averías.

Paso 1: requisitos del sistema

Antes de comenzar, asegúrese de que su sistema cumpla con los requisitos técnicos especificados anteriormente. Esto puede incluir detalles sobre el sistema operativo, la versión de la base de datos, la capacidad de almacenamiento y otros componentes necesarios.

Paso 2: descarga de la aplicación

Descargue la última versión de la aplicación desde el repositorio designado o el recurso proporcionado en el manual. Asegúrese de obtener el paquete de instalación correcto y guarde el archivo en una ubicación accesible en su sistema.

Paso 3: instalación de la aplicación.

Ejecute el archivo de instalación descargado. Siga las instrucciones en pantalla para completar el proceso de instalación. Esto puede incluir la selección de la ubicación de instalación, la aceptación de los términos y condiciones, y la configuración de opciones personalizadas si es necesario.

Paso 4: Configuración inicial.

Una vez finalizada la instalación, se le pedirá que realice una configuración inicial. Esto puede implicar la configuración de la conexión a la base de datos, la creación de cuentas de usuario administrador y la personalización de preferencias iniciales. Siga las indicaciones proporcionadas en el proceso de configuración.

Paso 5: inicio de sesión

Inicie la aplicación e inicie sesión utilizando las credenciales de administrador o usuario proporcionadas durante la configuración inicial. Si es necesario, el manual técnico ofrecerá detalles sobre cómo acceder a la interfaz de inicio de sesión y cómo recuperar contraseñas olvidadas.

Paso 6: Navegación y uso

Una vez dentro de la aplicación, explore la interfaz de usuario y familiarícese con las distintas secciones y funciones disponibles. Consulte el apartado de "Interfaz de Usuario" en el manual técnico para obtener orientación sobre la navegación y cómo realizar tareas específicas.

Paso 7: implementación de tareas

Utilice la aplicación para llevar a cabo las tareas específicas relacionadas con el sistema de soporte para gestión de averías. Esto puede incluir agregar nuevos datos, registrar información de clientes, realizar aperturas o cierres de casos, entre otras tareas. Siga los procedimientos operativos detallados en el manual técnico para cada una de estas tareas.

Paso 8: Mantenimiento y actualización

Referencie la sección de "Mantenimiento y Actualizaciones" en el manual técnico para obtener orientación sobre cómo mantener la aplicación funcionando sin problemas. Aprenda cómo realizar copias de seguridad de los datos, resolver problemas comunes y aplicar actualizaciones periódicas.

Paso 9: Soporte técnico

Si en algún momento encuentra dificultades, consulte la sección de soporte técnico del manual para obtener información sobre cómo contactar al equipo de soporte o resolver problemas conocidos.

Paso 10: Cierre y finalización

Cuando haya completado su trabajo con la aplicación, asegúrese de cerrar sesión y, si es necesario, apagar o cerrar la aplicación según las indicaciones del manual técnico.

Aspectos generales de la arquitecturaMicrosoft

.NET Framework 4

El sistema debe ser desarrollado utilizando Visual Studio Community 2022 (última versión a la fecha).

Asegúrese de tener instalada la última versión en su computadora.

.NET Framework es un componente integral de Windows que admite la compilación y ejecución de aplicaciones y servicios Web XML. El diseño de .NET Framework está enfocado a cumplir los siguientes objetivos.

Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ya sea distribuida en Internet o de forma remota.

Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.

Ofrecer un entorno de ejecución de código que promueva la realización segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.

Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan scripts o intérpretes de comandos.

Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.

Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

ADO.NET Entity Framework 6.1

Entity Framework es un conjunto de tecnologías de ADO.NET que permiten el desarrollo de aplicaciones de software orientadas a datos, admitiendo código que es independiente de cualquier motor de almacenamiento de datos o esquema relacional determinado.

ADO.NET Entity Framework permite a los desarrolladores crear herramientas de acceso a datos programando con un modelo de aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y el mantenimiento necesario para las aplicaciones orientadas a datos. Las aplicaciones de Entity Framework ofrecen las siguientes ventajas.

Las aplicaciones pueden funcionar en términos de un modelo conceptual más centrado en la aplicación, que incluye tipos con herencia, miembros complejos y relaciones.

Las aplicaciones están libres de dependencias de codificación rígida de un motor de datos o de un esquema de almacenamiento.

Las asignaciones entre el modelo conceptual y el esquema específico de almacenamiento pueden cambiar sin tener que cambiar el código de la aplicación.

Los desarrolladores pueden trabajar con un modelo de objeto de aplicación coherente que se puede asignar a diversos esquemas de almacenamiento, posiblemente implementados en sistemas de administración de base de datos diferentes.

Se pueden asignar varios modelos conceptuales a un único esquema de almacenamiento.

La compatibilidad con Language Integrated Query (LINQ) proporciona validación de la sintaxis en el momento de la compilación para consultas en un modelo conceptual.

Fluent validation

Pequeña librería de validación de .NET que utiliza una interfaz y expresiones lambda para la construcción de reglas de validación para sus objetos de negocio.

Automaper

Es una pequeña librería construido para resolver un problema complejo, deshacerse de código que asigna un objeto a otro.

MvcReportViewer

Librería que permite la visualización de reportes de SQL Server Reporting Services en ASP.NET MVC.

Ninject 3.2.20

Librería utilizada para inyectar dependencias a los objetos, logrando que nuestros componentes trabajen de una manera genérica controlando la ejecución de aquellos específicos y maximizando la coherencia entre ellos, también para simplificar el diseño, la implementación y el mantenimiento de las aplicaciones.

La inyección de dependencias es un patrón de diseño orientado a objetos, en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto.

iTextSharp

Librería de código abierto para la generación de PDF para la plataforma .NET.

Bootstrap

Librería que agiliza el desarrollo de la interfaz web.

MvcProjectAwesome

Librería que contiene un amplio controles que se puede utilizar para construir aplicaciones web altamente interactivas y de respuesta AJAX.

MvcSiteMapProvider 4.6.3

Librería que provee menús flexibles para trabajar con ASP.NET MVC.

POCO

Clases independientes de tecnologías concretas de acceso a datos, con código completamente bajo nuestro control. En definitiva, con este diseño (Persistence Ignorance) lo que buscamos es que las capas superiores “no sepan nada” de las inferiores.

Pistas de auditoria (Entity Framework 6.1 DbContext API)

La mayoría de las aplicaciones tiene la necesidad de guardar la información de auditoría sobre cambios hechos a objetos en la base de datos. Tradicionalmente, esto sería hecho por procedimientos almacenados, trigger (o disparador); con Entity Framework 6.1 DbContext API, es bastante fácil poner en práctica una auditoría más robusta.

DbContext expone la propiedad de ChangeTracker, que contiene todos los cambios hechos desde SaveChanges más reciente. Esta propiedad puede ser usada para conseguir la información sobre los valores actuales y anteriores de cualquier entidad en el DbContext. Por lo que sobrescribiendo el método SaveChanges(), podemos poner en práctica una auditoría que registra cualquier cambio al mismo tiempo que son salvados en la base de datos.

El sistema cuenta con una bitácora donde se registran las actividades de los usuarios, por lo que se puede determinar la acción que fue realizada (inserción, modificación o borrado de datos). También cuenta con un reporte de la bitácora, el cual es exclusivo para las Jefaturas, donde se pueden obtener los datos en una fecha determinada.

Patrón de diseño ASP.NET MVC 5

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. Este patrón se encuentra dividido en 3 componentes.

Modelo: esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

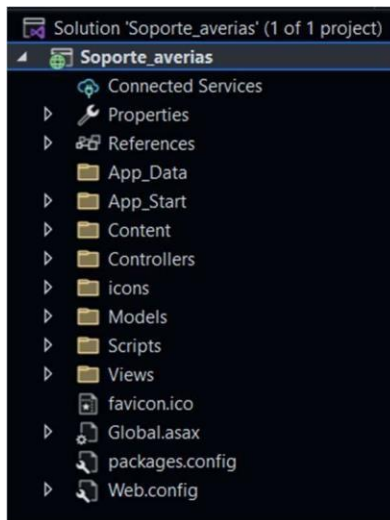
Vista: este presenta el modelo en un formato adecuado para interactuar, con la interfaz de usuario.

Controlador: este responde a eventos, acciones del usuario, e invoca peticiones al modelo y a la vista.

Proyecto Visual Studio

Para la comprensión de los segmentos que componen el proyecto sistema de soporte para manejo de averías. A continuación, se detallan cada una.

Ilustración 66. Segmentos del proyecto sistema de soporte para manejo de averías



Fuente: elaboración propia

App_Start

Contiene información para el inicio del sistema al ejecutarse.

Content

Contiene los archivos más importantes relacionados con Bootstrap para el diseño visual más atractivo de una página web

Controllers

Contiene todos los controladores necesarios para la funcionalidad correcta del sistema

Icons

Capa encargada de contener todos los iconos que se utilizan en la aplicación

Models

Capa encargada de la información de la base de datos y su conexión con las secciones importantes de la aplicación

Scripts

Sección que se encarga de ejecutar y validar los archivos para que en la aplicación funcionen correctamente esos archivos de la sección de 'Content'

Views

Capa encargada de mostrar la información en las pantallas de la aplicación.

Diccionario de datos

Con base en la Ilustración 1, se describe de manera completa los datos almacenados en la base de datos del sistema. Esta razón permite mejor comprensión y manejo de información en un entorno de tecnología de información.

Submódulos

El submódulo de períodos de garantía es el encargado de almacenar información relacionada con los períodos de garantía que tiene algún producto cuando se reporta alguna falla y se debe abrir un caso. La tabla TBL_PeriodoGarantia registra los detalles de los periodos de garantía como la cantidad y su respectiva descripción.

En el submódulo de fechas creación de caso es el que se encarga de almacenar información con relación a las fechas de creación de caso cuando se abre alguno por distintas razones que el usuario detalla. En la tabla TBL_FechaCreacionCaso se guardan los detalles de las fechas de creación y una breve descripción.

Para el submódulo de empleados se almacena información relacionada con los empleados con las características como: cédula, nombre, primer y segundo apellido. La tabla TBL_Empelador registra los detalles de las características mencionadas anteriormente.

Con el submódulo de municipalidades se almacena información relacionada con las municipalidades con sus características esenciales como nombre y descripción. La tabla TBL_Municipalidad registra los detalles de las características mencionadas anteriormente.

Dentro del submódulo de productos se almacena información relacionada con los productos ofrecidos por la empresa con sus características esenciales como nombre y descripción. La tabla TBL_Producto registra los detalles de las características mencionadas anteriormente.

Cada caso debe considerarse con alguna descripción por la cual existe, de manera que se realiza un submódulo de descripciones de caso en el cual se almacenan los datos como la descripción y algún comentario u observación relacionada con la descripción escrita. La tabla TBL_DescripcionCaso es la encargada de registrar esos detalles y guardarlos en la base de datos. Cada caso tiene su urgencia con la cual los del área de soporte ven cuál caso se atiende primero.

Por esa razón se realiza un submódulo de prioridades de caso en el cual se almacena datos como el nombre y la descripción de la prioridad. La tabla TBL_PrioridadCaso es la encargada de registrar esos detalles y guardarlos en la base de datos.

Cada caso tiene su estatus con el cual los del área de soporte van modificando según avanza la resolución caso que se atiende. Se realiza un submódulo de estados de caso en el cual se almacena datos como el nombre y la descripción del estado. La tabla TBL_EstadoCaso es la encargada de registrar esos detalles y guardarlos en la base de datos.

En el submódulo de fechas cierre de caso es el que se encarga de almacenar información con relación a las fechas de cierre de caso cuando se cierra por completo el caso que el usuario detalló. En la tabla TBL_FechaCierreCaso se guardan los detalles de las fechas de cierre y una breve descripción.

Para los dos últimos submódulos, roles y usuarios, cada uno se encarga de almacenar la información correspondiente. Para el caso del rol sería el nombre del rol y su descripción mientras que para el usuario, sus características son: al menos 1 rol de los creados en el submódulo de roles, su cédula, nombre, primer y segundo apellido, correo, teléfono y la clave. En las tablas TBL_Rol y TBL_Usuario se guardan los detalles de cada información, respectivamente.

Información detallada acerca del diccionario de datos

Tabla: TBL_PeriodoGarantia

Descripción: esta tabla almacena información sobre los períodos de garantía.

TN_IdPeriodoGarantia (PK): número de identificación único para cada periodo de garantía

TC_PeriodoGarantia: periodo de garantía

TC_Descripcion: descripción del periodo de garantía

Tabla: TBL_FechaCreacionCaso

Descripción: esta tabla almacena información sobre las fechas de creación de caso.

TN_IdFechaCreacionCaso (PK): número de identificación único para cada fecha de creación de caso

TD_FechaCreacionCaso: fecha de creación de caso (día, mes, año)

TC_Descripcion: descripción de la fecha de creación de caso Tabla:

TBL_Empleado

Descripción: esta tabla almacena información sobre los empleados. TN_IdEmpleado

(PK): número de identificación único para cada empleado TN_Cedula: número de cédula del empleado

TC_Nombre: nombre del empleado TC_PrimerApellido:

primer apellido del empleado TC_SegundoApellido: segundo

apellido del empleado Tabla: TBL_Municipalidad

Descripción: esta tabla almacena información sobre los períodos de garantía.

TN_IdMunicipalidad (PK): número de identificación único para cada municipalidad

TC_Nombre: nombre de la municipalidad

TC_Descripcion: descripción de la municipalidad Tabla:

TBL_Cliente

Descripción: esta tabla almacena información sobre los clientes. TN_IdCliente

(PK): número de identificación único para cada cliente TC_Nombre: nombre del cliente

TC_Descripcion: descripción del cliente Tabla:

TBL_Producto

Descripción: esta tabla almacena información sobre los productos.

TN_IdCliente (PK): número de identificación único para cada producto

TC_Nombre: nombre del producto

TC_Descripcion: descripción del producto

TBL_DescripcionCaso

Descripción: esta tabla almacena información sobre las descripciones de los casos.

TN_IdCliente (PK): número de identificación único para cada descripción de caso

TC_Descripcion: descripción del caso

TC_Observacion: comentario u observación con respecto a la descripción del caso

TBL_PrioridadCaso

Descripción: esta tabla almacena información sobre las prioridades del caso.

TN_IdCliente (PK): número de identificación único para cada prioridad de caso

TC_Nombre: nombre de la prioridad de caso

TC_Descripcion: descripción de la prioridad de caso

TBL_FechaCierreCaso

Descripción: esta tabla almacena información sobre las fechas de cierre de caso.

TN_IdFechaCierreCaso (PK): número de identificación para cada fecha de cierre de caso

TD_FechaCierreCaso: fecha de cierre de caso (día, mes, año)

TC_Descripcion: descripción de la fecha de cierre de caso

TBL_Rol

Descripción: esta tabla almacena información sobre los roles para los usuarios del sistema.

TN_IdRol (PK): número de identificación único para cada rol

TC_Nombre: nombre del rol

TC_Descripcion: descripción del rol

Tabla: TBL_Usuario

Descripción: esta tabla almacena información sobre los usuarios del sistema

TN_IdUsuario (PK): número de identificación único para cada usuario TN_IdRol

(FK): número de identificación único para el rol del usuario TN_Cedula: cédula del usuario

TC_Nombre: nombre del usuario TC_PrimerApellido:

primer apellido del usuario TC_SegundoApellido: segundo

apellido del usuario TC_Correo: correo del usuario

TN_Telefono: teléfono del usuario

TC_Clave: contraseña del usuario

Clave foránea TN_IdRol hace referencia a la tabla TBL_RolTabla:

TBL_Caso

Descripción: esta tabla almacena información sobre los casos TN_IdCaso

(PK): número de identificación único para cada caso

TN_IdPeriodoGarantia (FK): número de identificación único para cada período de garantía

TN_IdFechaCreacionCaso (FK): número único para cada fecha de creación de caso

TN_IdEmpleado (FK): número de identificación único para cada empleado TN_IdMunicipalidad

(FK): número de identificación único para cada municipalidad TN_IdCliente (FK): número de identificación único para cada cliente

TN_IdProducto (FK): número de identificación único para cada producto TN_IdDescripcionCaso

(FK): número de identificación único para cada descripción de

caso

TN_IdPrioridadCaso (FK): número de identificación único para cada prioridad de caso

TN_IdEstadoCaso (FK): número de identificación único para cada estado de caso

TN_IdFechaCierreCaso (FK): número de identificación único para cada fecha de cierre de caso

TN_IdUsuario (FK): número de identificación único para cada usuario

Clave foránea TN_IdPeriodoGarantia hace referencia a la tabla TBL_PeriodoGarantiaClave

foránea TN_IdFechaCreacionCaso hace referencia a TBL_FechaCreacionCasoClave foránea

TN_IdEmpelado hace referencia a la tabla TBL_Empleado

Clave foránea TN_IdMunicipalidad hace referencia a la tabla TBL_MunicipalidadClave

foránea TN_IdCliente hace referencia a la tabla TBL_Cliente

Clave foránea TN_IdProducto hace referencia a la tabla TBL_Producto

Clave foránea TN_IdDescripcionCaso hace referencia a la tabla TBL_DescripcionCasoClave

foránea TN_IdPrioridadCaso hace referencia a la tabla TBL_PrioridadCaso Clave foránea

TN_IdEstadoCaso hace referencia a la tabla TBL_EstadoCaso

Clave foránea TN_IdFechaCierreCaso hace referencia a la tabla TBL_FechaCierreCasoClave

foránea TN_IdUsuario hace referencia a la tabla TBL_Usuario

Uso del sistema

Se deben considerar ciertos aspectos a la hora de usar el sistema. Para lo que son registro e inicio de sesión, los usuarios pueden hacerlo en el sistema web. La autenticación ya está incorporada debido a que es proporcionada por ASP.NET MVC 5.

Para los filtros de búsqueda, debe verificarse que el campo es de texto o si es de seleccionar el dato. Después, se presiona el botón de buscar que tiene el icono de una lupa mientras que para borrar los filtros se utiliza el botón con el icono de un borrador. Los detalles cualquier usuario lo ve porque es una corroboración de los datos añadidos.

Para el proceso de creación de un caso, el usuario debe contactar a soporte para que le avisen al administrador para colaborar con la creación del caso con respecto a cierto producto con sus determinadas razones las cuales el usuario desea reportar para que le resuelvan.

Mantenimiento y actualización

Para agregar nuevos datos y crear nuevos registros en la base de datos, hay que asegurarse de que tanto la vista como los controladores correspondientes estén implementados y que el usuario sea el administrador del sistema de soporte.

Actualización de la información

La información es actualizada casi que inmediato en el sistema con la finalidad de que el usuario tenga una experiencia más agradable con el uso de la aplicación.

Modelo relacional base de datos

Considerando la Ilustración 1, las entidades principales son las tablas las cuales incluyen la información crucial sobre un caso. Además, Cada entidad representa lo que su nombre indica: periodo de garantía, fecha de creación de caso, empleados, municipalidades, clientes, productos, descripciones de caso, prioridades de caso, estados de caso, fecha de cierre de caso, roles para los usuarios y usuarios registrados en el sistema. Y como nota adicional, existen restricciones entre algunas relaciones. Para agregar un caso, obligatoriamente debe haber al menos 1 dato de cada entidad y en el caso de usuarios, es requisito que exista al menos un rol dentro de la tabla de roles.

Resolución de problemas

En caso de requerirse ayuda o asistencia técnica, debe contactarse con el equipo de desarrollo o al soporte técnico designado para que se le colabore.

Costos

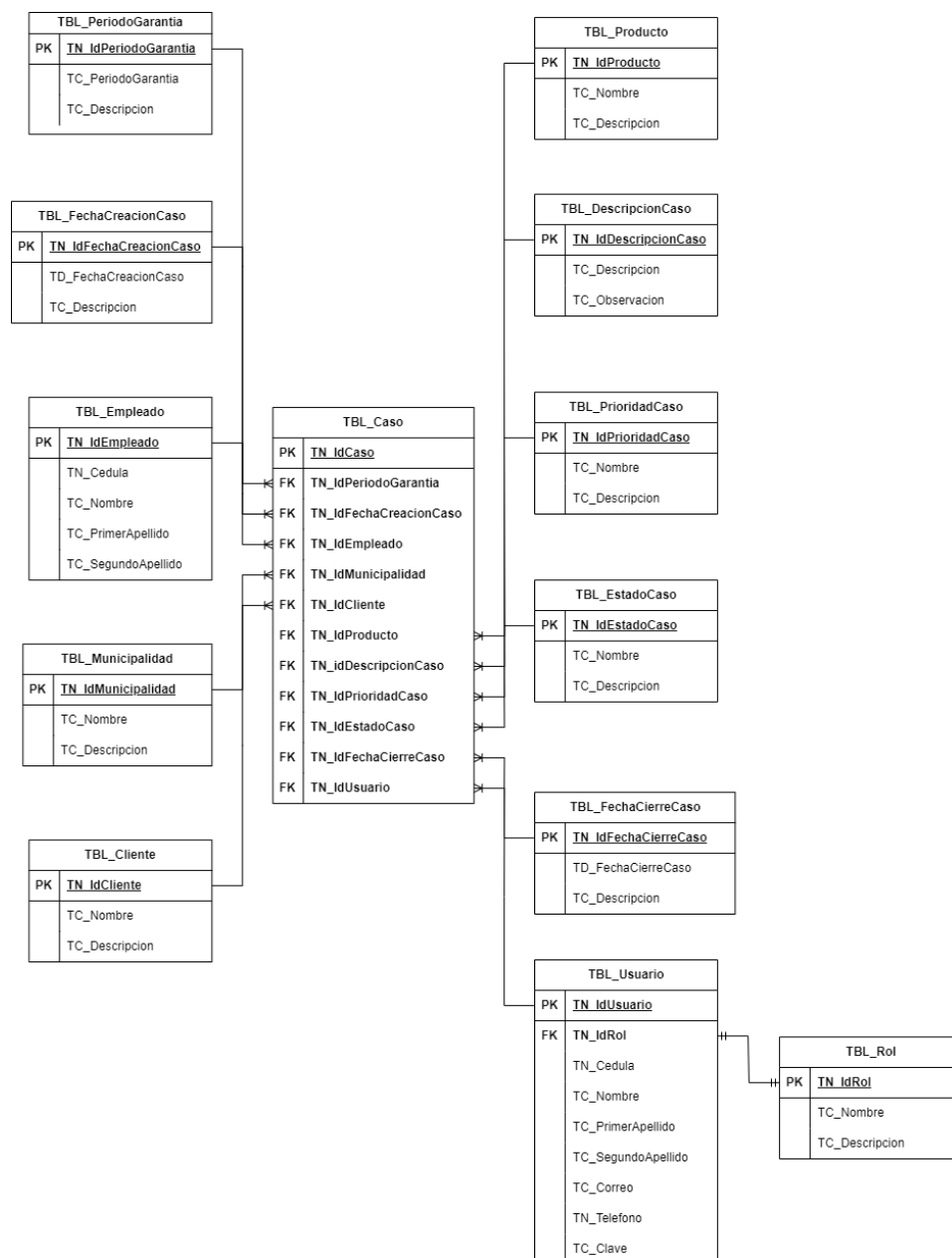
El proyecto podría evolucionar. Es indispensable considerarlo porque un sistema de soporte se mantiene en constante cambio por lo cual se realizarían los cambios en los manuales técnico y de usuario como en las actualizaciones de la aplicación.

Referencias del diccionario de datos

En este apartado del manual técnico se adjuntan dos referencias: la documentación ASP

.NET MVC se encuentra en el siguiente enlace <https://docs.microsoft.com/en-us/aspnet/mvc/overview> y la documentación de C# está disponible en el siguiente enlace <https://docs.microsoft.com/en-us/dotnet/csharp/>

Ilustración 1. Modelo relacional de la base de datos



Fuente: elaboración propia