

Documentación y manual de usuario de software Control Días

Autor y desarrollador: Christian Matzner

Definición e información general	1
Idea y objetivos generales	2
Especificación de requerimientos del proyecto	2
Procedimientos:	3
Arquitectura del sistema:	4
Dependencias externas:	5
Descripción de procesos y servicios:	5
Manual de usuario:	8

1. Definición e información general

- a) **Descripción:** Control Días es un software que pretende dar información adicional al personal de portería, el cual monitorea un sistema de control de acceso de los usuarios que ingresan por medio del uso de tarjetas personales. El propósito del proyecto es evitar problemáticas relacionadas con el exceso de trabajo de los empleados en la empresa.
- b) **Procedimiento de instalación:** La obtención del software se hace por medio del siguiente [enlace](#), y no necesita instalación.

2. Idea y objetivos generales

- a) **Idea General:** El software monitorea los nuevos registros en la base de datos, para calcular y detectar los días trabajados de cada persona que ingresa por medio de una tarjeta de acceso, notificando y desplegando una alarma en el caso de que ingrese una persona al trabajo por siete o más días consecutivos, con el fin de evitar problemáticas al momento de registrar el ingreso al trabajo en el sistema oficial.
- b) **Objetivos:** Desarrollar un software con la capacidad de monitorear, desplegar alarma y generar reportes, para ser un verdadero apoyo a evitar problemas internos relacionados con la asistencia al trabajo.
- c) **Usuarios:** Las personas que usan el programa son exclusivamente los encargados de monitorear la información de salida/ingreso en portería.

3. Especificación de requerimientos del proyecto

- a) **Requisitos generales:**
 - Tiene que ser necesariamente visible para el usuario, pero no usar mucho espacio en pantalla, de manera que se pueda seguir monitoreando los ingresos mientras Control Días muestra información adicional.
 - Al haber un caso de exceso de días trabajados, aparte de aparecer una notificación visible, tiene que sonar una alarma para captar mayor atención del usuario.
 - Tiene que ser capaz de discriminar el departamento al que pertenece cada persona, para permitir el acceso a integrantes de departamentos que no tienen restricción de acceso.
 - Tiene que ser capaz de mostrar un reporte de los eventos ocurridos en diferentes intervalos de tiempo.

b) Requisitos funcionales:

- Desplegar en pantalla el nombre, apellido y días consecutivos de trabajo.
- Notificaciones visuales diferentes para los casos de exceder o no exceder el rango permitido
- Tener la opción de conectarse a diferentes bases de datos predefinidas o conexión manual por medio de ingreso de credenciales.
- Notificar cada problema referente a la conexión con el servidor de manera que el usuario sepa que está pasando.
- Tener un sistema de reporte visual de eventos, con filtros de fechas y nombre.

c) Información de auditoría:

- Control Días utiliza una base de datos previamente creada y configurada, perteneciente al sistema de registro de acceso propio de la empresa.

d) Alcance del sistema:

- Control Días tiene que tener permisos para escritura y lectura dentro de la base de datos

4. Procedimientos:

a) **Herramientas utilizadas:** Para llevar a cabo el proyecto, se utilizó el lenguaje de programación C# y su módulo de plantillas de WindowsForm. La base de datos utilizada es relacional y en lenguaje SQL. Se controlaron las versiones con el sistema de control de versiones .git y se utilizó el método de desarrollo incremental para la construcción del software.

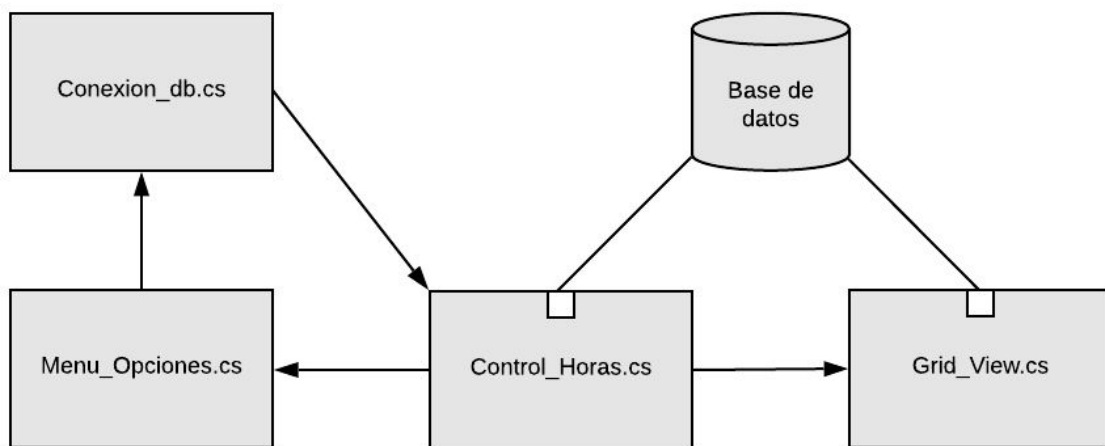
b) **Instalación y prueba:** La obtención del software se hace por medio del siguiente [enlace](#), y no necesita instalación. Para ejecutar el programa, ingresar a Control_Horas -> WindowsFormsApplication1 -> bin -> Release y hacer doble click en Control Dias.exe. Dentro del programa, puede probar la funcionalidad del sonido, ingresando a opciones. Para conectarse con la base de datos, es necesario desplegar la pestaña al lado derecho del texto “ Control Dias de Trabajo” y seleccionar la base de datos a la cual quiere conectar, para finalmente apretar el botón Seleccionar. En caso de tener conexión

exitosa, el programa notificará el estado de conexión y en caso contrario se desplegará una notificación explicando el problema.

5. Arquitectura del sistema:

- a) **Descripción jerárquica:** El código principal se aloja en el archivo “Control_Horas.cs”, el cual utilizará distintos archivos y componentes para el funcionamiento completo de la aplicación. Los archivos “Conexion_db.cs”, “Grid_View.cs”, “Menu_opciones.cs”, contienen las funcionalidades de las tres ventanas desplegables dentro del programa

- b) **Diagrama:**



- c) **Descripción individual:**

- **Control_Horas.cs:** Archivo principal, utiliza conexión directa con base de datos para quedar en funcionamiento constante. Responsable de extraer información de distintos campos de la base de datos y utilizarlos en el despliegue en pantalla. Contiene algoritmos de cálculo de días consecutivos de trabajo y toma decisiones sobre si mostrar o no las notificaciones al usuario. Los datos utilizados en dichos cálculos son subidos a la tabla **CAN_tblDayCount** para guardar los registros.
- **Grid_View.cs:** Interfaz que cuenta con conexión directa a base de datos y funciona como método de generación de reportes,

haciendo consultas a la tabla CAN_tblDayCount con distintos filtros opcionales.

- **Menu_Opciones.cs:** Menú desplegable a través del botón “Opciones” de la interfaz de Control_Horas y permite hacer pruebas de sonido y desplegar la interfaz de conexión manual por medio del botón “Conexión Manual”.
- **Conexion_db.cs:** Interfaz que permite al usuario ingresar credenciales de una conexión alternativa a las opciones predefinidas en la pestaña desplegable de la interfaz principal.
- **getSet.cs:** Almacén de variables globales utilizadas por todas las interfaces.

6. Dependencias externas:

El programa necesita conexión con una base de datos externa para poder comenzar su funcionamiento, necesitando permisos de conexión, lectura y escritura dentro de la misma.

Las librerías externas utilizadas dentro de C# fueron:

- **System.Windows.Forms**, para la generación de código automático al hacer uso de las plantillas de funciones entregadas por Visual Studio en conjunto con C#.
- **System.Data.SqlClient**, para las interacciones con la base de datos.
- **System.Media**, para la reproducción del sonido de alarma cuando se ejecuta la alerta.
- **System.Data**, para la manipulación de cadenas de caracteres.

7. Descripción de procesos y servicios:

a) **Control_Horas.cs:**

- RegisterForChanges():
Escucha los cambios en la base de datos y avisa a OnNotificationChange que hubo cambios para que se ejecute.
- OnNotificationChange(object caller, SqlNotificationEventArgs e):
Se encarga de recibir el evento entregado por RegisterForChanges() y activar las demás funciones de extracción de datos, además actualiza la interfaz principal con los datos de la persona que generó el evento.
- ExtractID():
Extrae el ID, nombre, apellido y departamento al cual pertenece el activador del evento y los almacena en variables globales alojadas en getSet.cs.
- ExtractDate():
Extrae las fechas de los eventos anteriores de la persona que activó el evento, calculando la diferencia que hay entre cada fecha para detectar cuando hubo una diferencia de más de 24 hrs para arrojar el número de días que pasaron hasta que ocurre ese caso. Guarda la fecha del evento y la fecha donde hay una diferencia de más de 24 hrs en variables globales almacenadas en getSet.cs y retorna el número total de días que han pasado desde el último intervalo de 24 hrs.
 - Versión 1.2: se añadió una columna a la tabla CAN_tblDoysControl, con el fin de discriminar si el registro es de un día repetido (2), o un cambio de día (1), con la finalidad de hacer un contador de 1. Esta nueva funcionalidad se agregó con la finalidad de resetear el contador en caso de evento de exceso de días y que no cuente si un día viene 5 minutos y se va. Para manipular un evento de exceso de días, se debe hacer UPDATE en la base de datos, cambiando el registro que se desea setear como nuevo comienzo, y modificar la columna ResetCount a 0, de esta forma, el contador de 1 avanza hasta que se topa con un 0 y compara el resultado con el contador de días original, en caso de que el

contador de 1 sea menor, el programa toma ese como los días de trabajo seguidos.

- ExtractRows():
Extrae el número de eventos que tiene la persona que generó el evento y lo guarda en una variable global en getSet.cs, para luego ser usada como largo máximo del arreglo que guarda las fechas en la función ExtractDate().
- Extract_cardID():
Extrae el número de tarjeta asociado al ID de la persona que activó el evento y lo guarda en una variable local en getSet.cs, para ser usada en el sistema de reporte.
- Extract_cardID():
Extrae el número de tarjeta asociado al ID de la persona que activó el evento y lo guarda en una variable local en getSet.cs, para ser usada por insert_info().
- insert_info(int card_id, int totalDays):
Recibe el id de tarjeta y el número de días retornado por ExtractDate() y los inserta en la base de datos, junto con otras variables globales almacenadas en getSet.cs
- validate_days(int days):
Recibe la cantidad de días retornada por ExtractDate() y muestra texto, imagen correspondiente al caso de que esté dentro o fuera de rango. Además, si está fuera de rango, muestra una ventana de notificación y reproduce un sonido hasta que el usuario presiona un botón dentro de la notificación. Después de todo el algoritmo, vuelve a RegisterForChanges() para escuchar nuevamente si hay nuevos cambios.
- pictureBox1_Click(int days):
Recibe como parámetro la cantidad de días retornada por ExtractDate() y dependiendo del valor muestra una imagen u otra.
- select_Server():
Capta el texto mostrado en el box correspondiente y en base a eso, decide cuál será el string de conexión, guardándolo en una variable global alojada en getSet.cs.

- Select_Button(object sender, EventArgs e):
Recibe como parámetro el evento de botón presionado, activando la función select_server(), para probar la conexión a la base de datos y permanecer en estado conectado o arrojando error, dependiendo de si la conexión fue exitosa o no.

b) Grid_View.cs:

- Search_Button_Click(object sender, EventArgs e):
Recibe como argumento el evento de botón presionado y capta todos los filtros ingresados por el usuario para aplicarlo a un string de consulta a base de datos.
- combo_Fill(ComboBox cb):
Llena el box del filtro por nombre con los nombres, apellidos e id, de las personas registradas en la base de datos.
- combo_Name_SelectedIndexChanged(object sender, EventArgs e):
Recibe como argumento el evento de seleccionar un nombre del box, recibiendo dicho texto por medio de una variable local, para extraer el id.

c) Conexion_db.cs:

- Aply_Button_Click(object sender, EventArgs e):
Recibe todos los datos ingresados en los box correspondientes, para construir el nuevo string de conexión manual y lo guarda como variable global en getSet.cs.

8. Manual de usuario:

Al abrir el programa, nos encontramos con la interfaz principal, a lo que acto seguido se abre la pestaña a la derecha del texto "Control Dias De Trabajo" y se selecciona la opción de servidor a la que se quiere conectar, en este caso, AXTRAX SERVER, y se presiona el botón "seleccionar". Si la conexión fue exitosa, lo notificará el mismo programa, si no, aparecerá una ventana notificando el programa. Una vez conectado, el programa está corriendo y escuchando a la base de datos.

Dentro de la interfaz principal, se puede acceder a diferentes opciones:

- 1) reporte: El botón reporte lleva a una interfaz que despliega una cuadrilla que, luego de aplicar los filtros deseados, presione el botón buscar para aplicar los filtros y realizar la búsqueda de los datos requeridos para mostrarlos en pantalla.
- 2) Opciones: El botón opciones abre una pequeña ventana, la que da acceso a las pruebas de sonido, apretando los botones “Prueba de sonido” y “Detener Prueba”. Por otro lado, esta interfaz nos da la posibilidad de establecer conexión manual.

2.1) Conexión Manual: Despliega una ventana con cuadros de ingreso de texto, en los que es necesario escribir las credenciales correspondientes para cada campo, para posteriormente presionar el botón “Aplicar”, para aplicar los cambios, y por último se presiona el botón aceptar para volver a la interfaz principal. Para probar la nueva conexión, en la pestaña desplegable seleccione la opción CONEXIÓN MANUAL y presione Seleccionar.
- 3) Por último, si el programa encuentra un evento que corresponde a una persona que ingresa al trabajo fuera del rango permitido, desplegará una ventana de notificación y una alarma sonora. Dentro del periodo de tiempo que la alarma está activa, el programa estará pausado, por lo que es importante atender la notificación lo antes posible.

IMPORTANTE:

- Siempre mantener el programa abierto y visible en pantalla.
- En caso de evento fuera del rango por un séptimo o más días, cerrar la notificación presionando el botón “Aceptar”, para proceder a detener a la persona en cuestión y notificar el estado en que se encuentra el contador de días de trabajo consecutivos. Después de hablar con la persona detonadora de la alarma, llamar a la persona encargada del sistema Chronos y notificar lo sucedido.
- En caso de cualquier problema con el programa, comunicarse con el departamento técnico.
- En caso de un corte de luz o algún evento que fuerce la desconexión del programa con la base de datos, abrir la aplicación desde el acceso directo “Control Días” ubicada en el escritorio y volver a conectar el sistema.