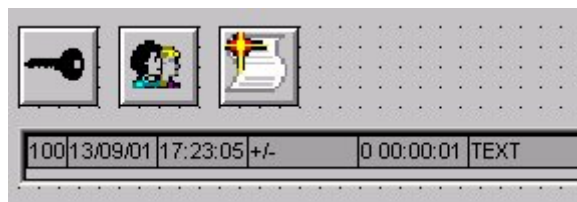


Capítulo 3

Histórico de alarmas.

La siguiente parte es bastante sencilla. Vamos a realizar un histórico de alarmas. Como ya hemos definido anteriormente prácticamente todo, solo tenemos que hacernos un botón en bottom.pdl que nos salte a la ventana de alarmas. Pero ¿a cual si no la hemos hecho?. Ciertamente, que malo es el alzheimer. Abrimos con graphics designer plantilla.pdl (o start.pdl, que aún son la misma cosa), y la guardamos bajo el nombre historico_alarmas.pdl.

Ahora ya podemos abrir bottom.pdl y crear un botón como la figura:

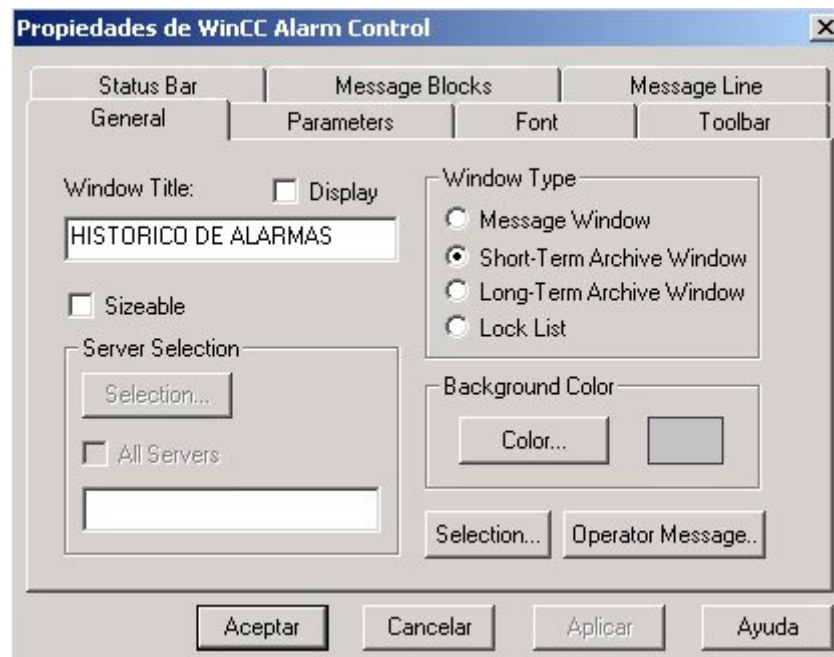


El código asociado al botón será lógicamente:

OpenPicture("historico_alarmas.PDL");

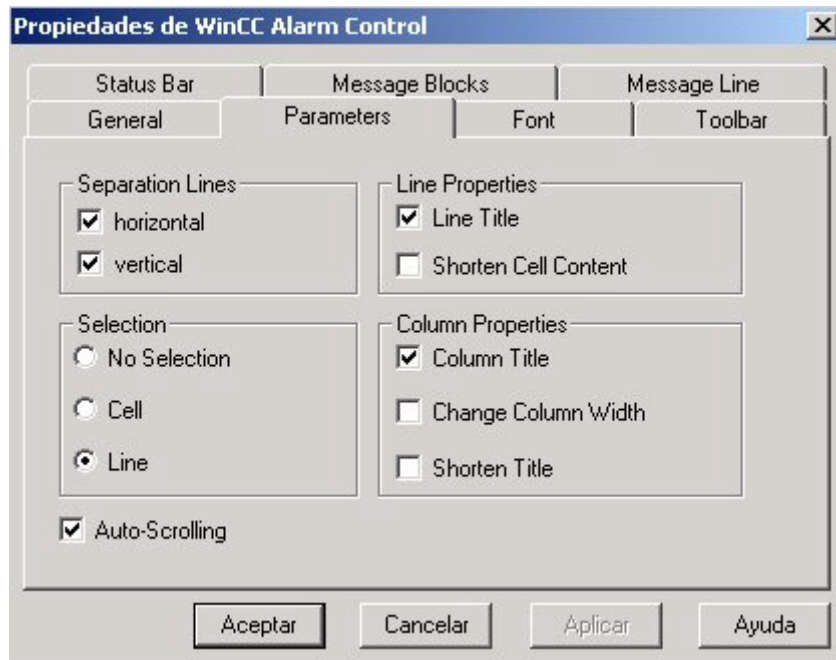
Ya podemos saltar a la ventana de historico_alarmas, para crear allí una ventana de alarmas, tal como vimos anteriormente (recuerda, lo de controls, a la derecha de la pantalla. ¿no?, pues vuelve a atrás en el libro).

Veamos como definimos esta ventana de alarmas (se va a parecer a la otra como un huevo a una gallina, veras:

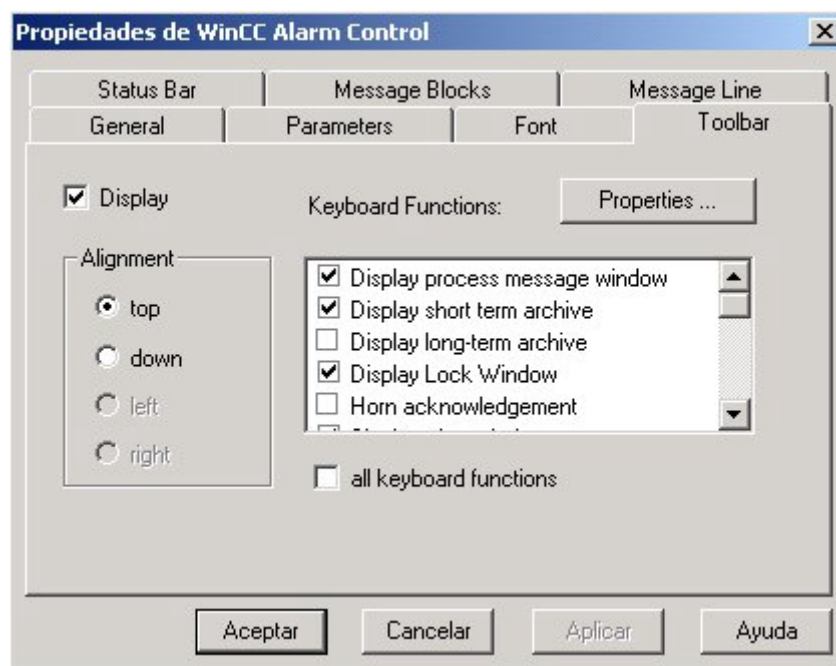


Un ejemplo para WinCC

El borde de la ventana lo quitamos. El Window Type debe de ser short-term, para que cuando entre a la pantalla nos muestre lo primero de todo el histórico de alarmas. Luego, a través de los botones del toolbar podremos visualizar las alarmas actuales, pero por defecto nos mostrará el tambor de alarmas guardado en el disco duro.



La selección que sea por línea, que no se pueda cambiar el ancho de las columnas y separación entre las mismas.

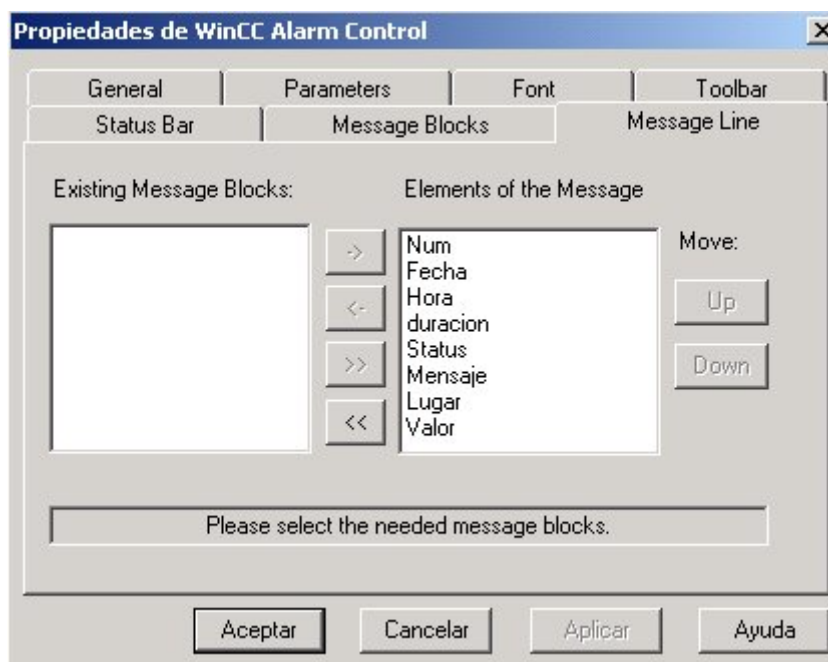


Llegamos a la barra de botones. Dejamos los siguientes:

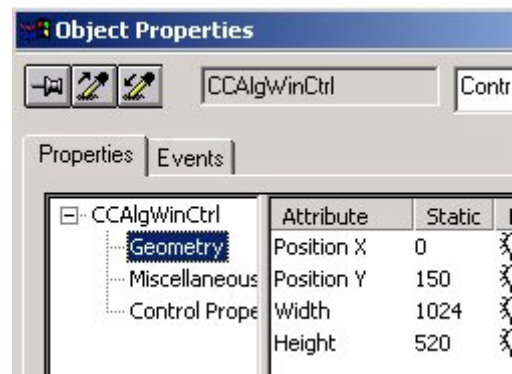
- Display process message: mostrar las alarmas actuales.
- Display short-term archive: mostrar el tambor de alarmas guardado.

- Display lock window: mostrar las alarmas bloqueadas desde wincc (p. Ej. Si queremos que aunque se active una alarma en el plc no nos esté saliendo en la pantalla del scada).
- Single acknowledgement: acusar una alarma.
- Group acknowledgement: acusar todas las alarmas a la vez.
- Auto scroll on-off: poder seleccionar una alarma en concreto en el histórico.
- Selection: realizar un filtrado de alarmas (ver solo las de la fecha x hasta la fecha y).
- Set lock: bloquear una alarma.
- Start of list: ir a la primera alarma.
- End of list: ir a la última alarma.
- Next message: siguiente alarma.
- Previous message: alarma anterior.
- Infotext: muestra la ayuda sobre dicha alarma.
- Loop in alarm: salta a una pdl que hayamos asociado a dicha alarma. Luego lo gastaremos, tranqui.
- Unlock/lock message: bloquear o desbloquear una alarma para que aparezca en el scada.

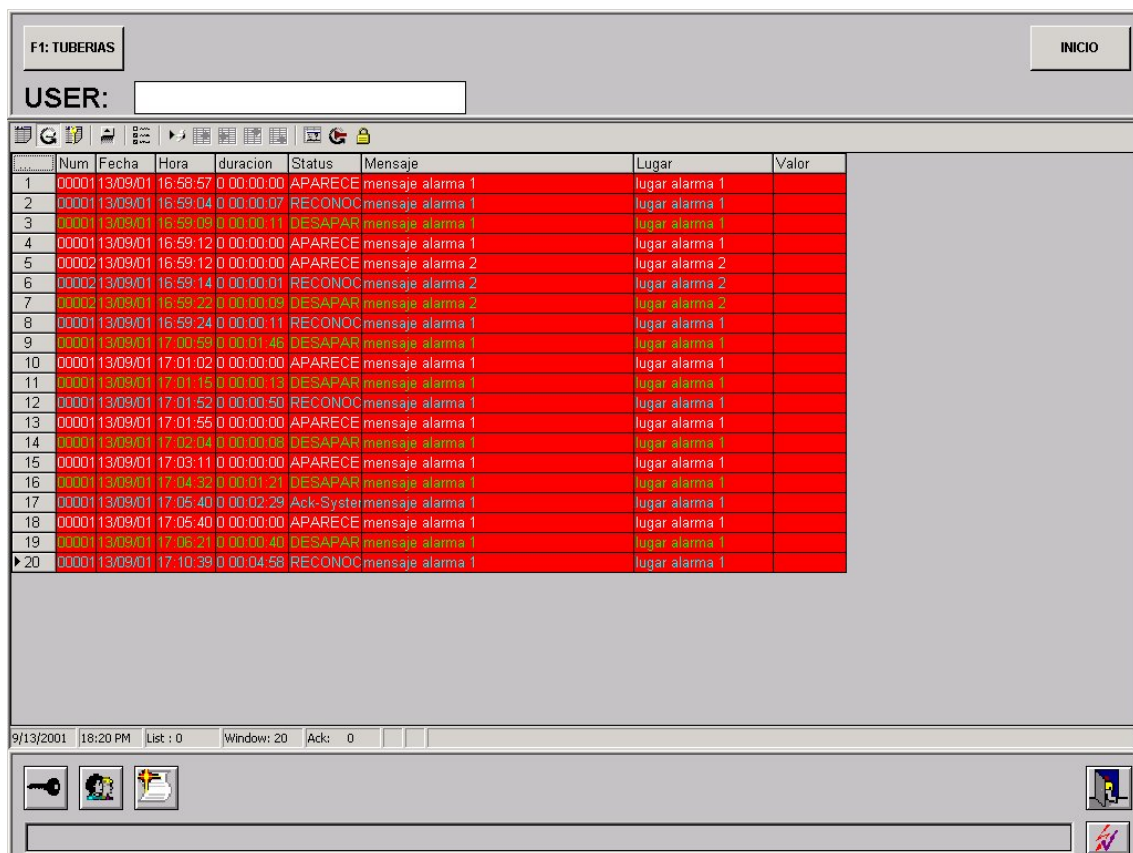
En status bar activamos todas las opciones. Y por último nos queda definir la línea de alarmas.



Ya tenemos configurada nuestra ventana de histórico de alarmas. Le damos un tamaño mas o menos como muestra la figura siguiente, y nombramos el objeto como ventana_historico_alarmas.

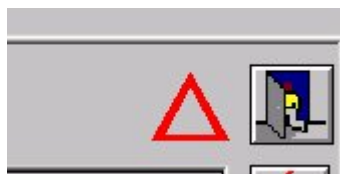


El resultado debe de ser el siguiente:



Un indicador de alarma general.

Vamos a poner un triangulo en la pantalla, que parpadeará mientras exista una alarma en la instalación. En la biblioteca podemos encontrar uno que nos va a servir para el cometido:

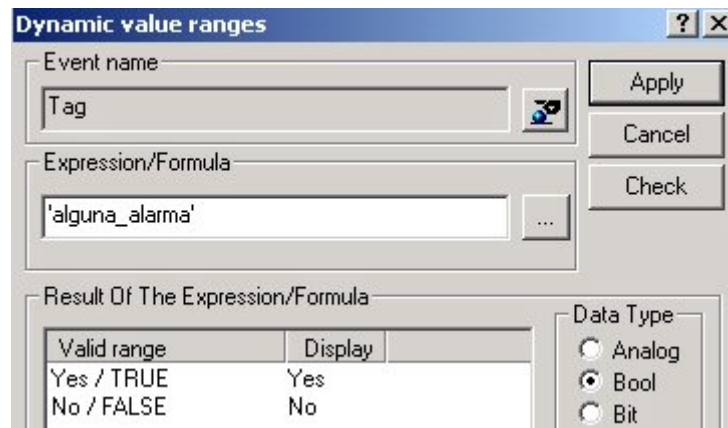


Nos creamos una variable de tipo bool. Esta variable, dentro del plc será una marca, que debe de activarse por programa de plc siempre que exista una alarma generada por el plc. Es decir, si alguna de las alarmas definidas en el plc está activa ese bit de marca deberá de ponerse a uno. Así nos quitamos tener que complicarnos la vida en el scada.

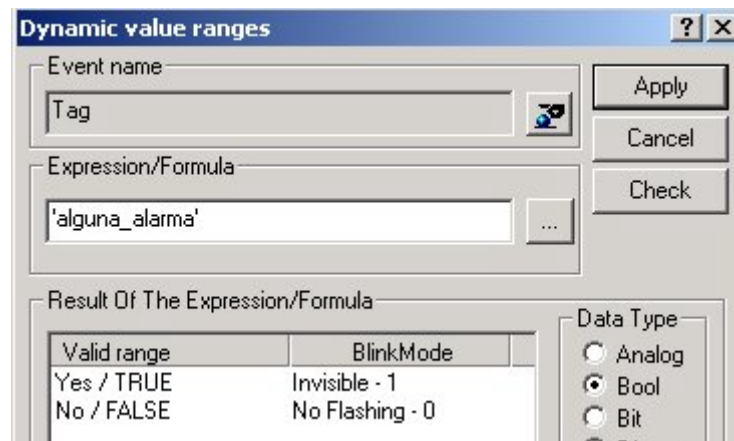
Teniendo ese bit activándose correctamente en el plc, ponemos nuestro triángulo en bottom.pdl, al lado del botón de salir, como se aprecia en la imagen.

Queda hacer que parpadee. Es sencillo. Nuestra variable la llamaremos alguna_alarma, asociada a la marca anteriormente descrita, y de tipo bool, lógicamente.

En la propiedad del triángulo display, asociamos el siguiente dynamic dialog: En static le ponemos no, para que la primera vez no se nos muestre.



En la propiedad blinkmo de le asociamos el siguiente dynamic:



Si queremos rizar el rizo, podemos en función del tipo de alarma, cambiar de color el triángulo, o incluso la velocidad de parpadeo, pero no lo voy a aplicar porque son chorradas.

Imprimir el histórico de alarmas.

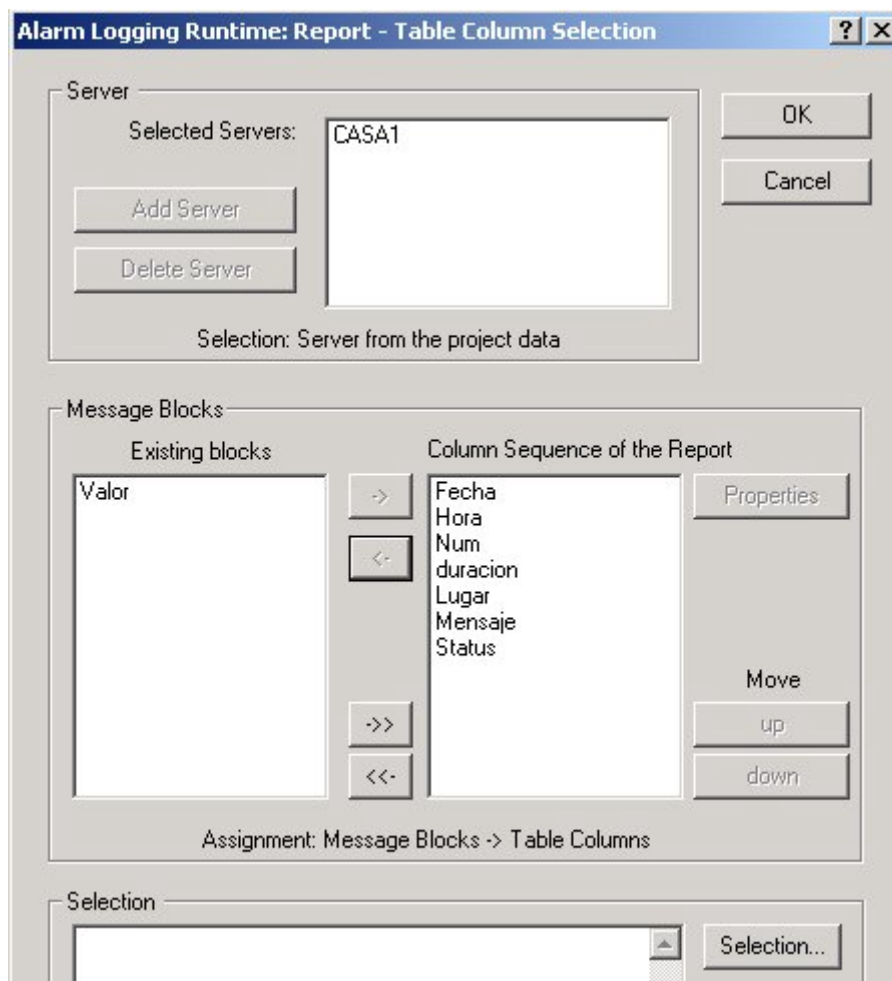
Vamos ya con la impresión de las alarmas. Si utilizamos el botón e la ventana del histórico de alarmas, automáticamente nos intenta imprimir el layout @ccalgrtshorttermarchive.rpl.

O sea, que si deseamos utilizar ese botón, por narices tendremos que modificar dicho template. Esto tiene la ventaja de que trabajas menos que el sastre de trazan, pero tiene varios inconvenientes:

- Si hacemos por lo que sea dos ventanas de alarmas, siempre saldrá el mismo formulario para las dos, y si en una mostramos un tipo de línea y en otra otro ya me dirás...
- No puedes controlar la situación, el wincc te controla a ti.
- La puedes cagar..

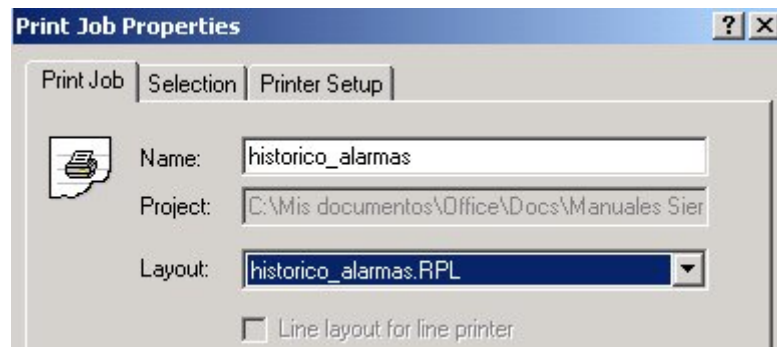
Así que pasando de esta solución, y vamos a hacernos un precioso botón que me imprima un histórico de alarmas, y además el que a mi me de la gana y como quiera yo. Como ya tenemos trabajo hecho, abrimos @ccalgrtshorttermarchive.rpl y lo guardamos con el nombre historico_alarmas.rpl.

Vamos a modificarlo. Entrando en el vemos que tiene una tabla dinámica, que en propiedades->connect->selection podemos definir como deseamos que sea nuestro estilo de línea de larmas (esta vez en la impresora, no en la pantalla).

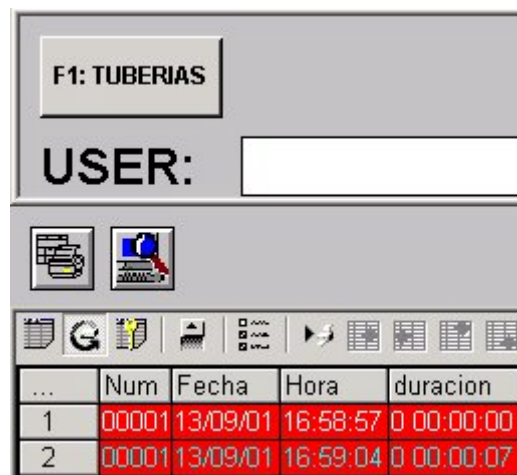


Aquí ya hemos terminado. Si quieres modificamos la cabecera y el pie de página para amoldarlo a nuestro proyecto.

Una vez finalizado el layout nos vamos a hacer un job que incluya este layout:



¿Qué nos queda?. Únicamente crear un botón que imprima dicho job. Nos ponemos dos botones de la librería, como muestra la figura.



Uno de los botones lo utilizaremos para visualizar el report antes de imprimirlo. El código asociado es el siguiente:

RPTJobPreview("historico_alarmas"); //Return-Type :BOOL

El otro nos imprimirá directamente el report:

RPTJobPrint("historico_alarmas"); //Return-Type :BOOL

