BRUNO MACIAS

- Las Base de Datos son muy utilizadas debido a las grandes garantías y bondades que proporcionan, el hecho de que permiten almacenar los datos en un solo sitio, conservando su integridad y consistencia es sin duda una de sus más importantes características

- Diseñar e implementar una Base de Datos Centralizada para administrar los datos capturados por Sistemas de Seguridad mediante una interfaz desarrollada en LabVIEW.

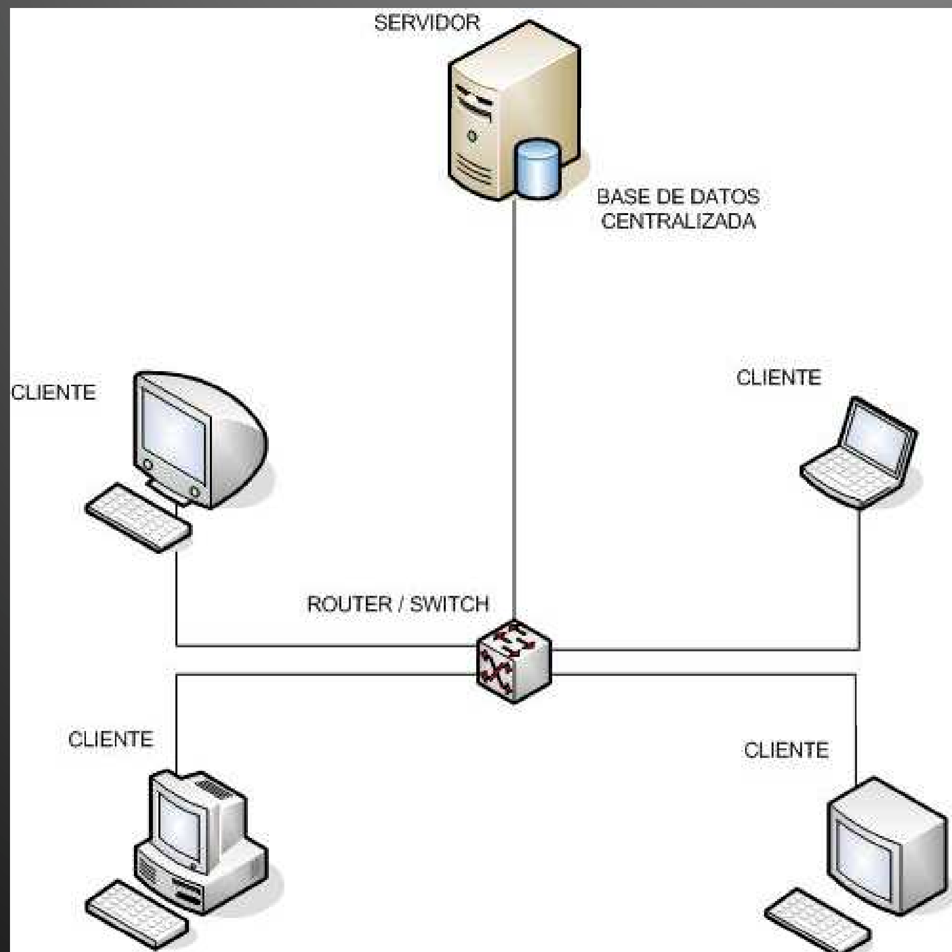
- Sistema de Seguridad Domiciliaria
- Sistema de Seguridad Industrial
- Sistema de Seguridad para un Laboratorio
- Sistema de Seguridad para equipos de Laboratorio
- Sistema de Seguridad para Vehículos
- Sistema para Control de Personal
- Sistema de Seguridad para entrada principal y corredores

- Utilizar LabVIEW, para el desarrollo de una interfaz gráfica que permita administrar una Base de Datos de un Sistema de Seguridades, debido a que la forma de programar en LabVIEW es mediante el uso de símbolos gráficos denominados "iconos" lo cual se conoce como programación en Lenguaje "G".

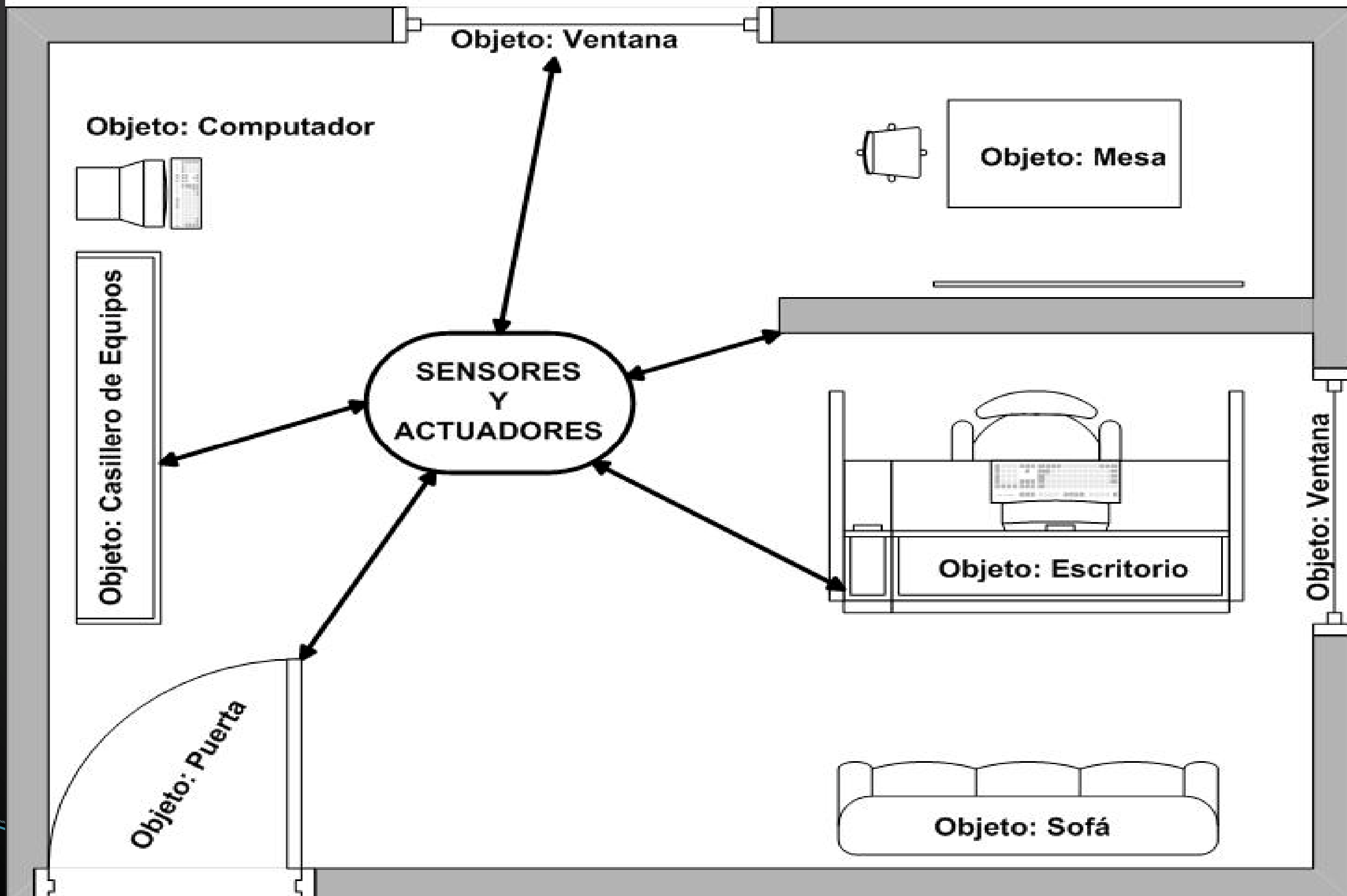
- Esta forma de programar es muy diferente frente a otros lenguajes de programación que se basan en líneas de texto, como C, VISUAL BASIC o C #.

- El diseño de la bases de datos es genérica capaz de soportar sistemas de seguridades futuros, ya que se base esencialmente en los sensores, actuadores y agentes.

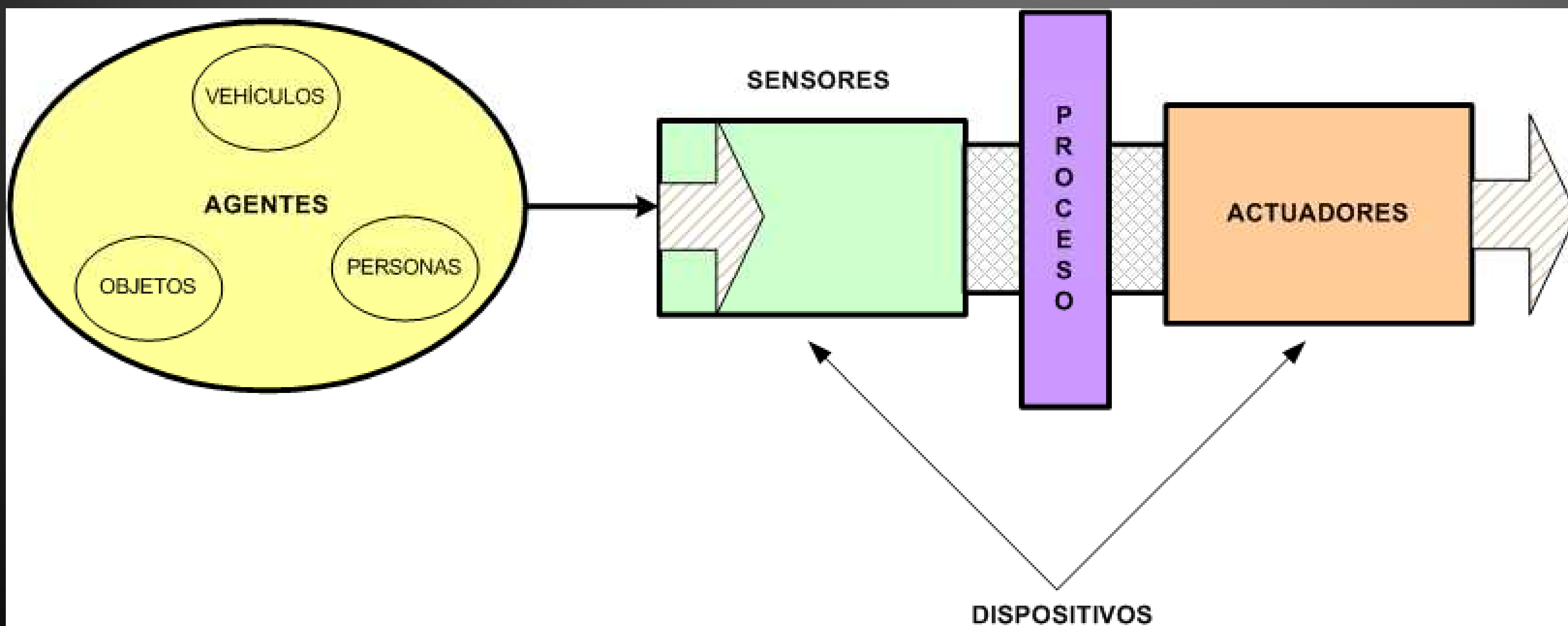
- A pesar de que estamos sujetos al uso de LabVIEW para el desarrollo de la interfaz gráfica que administrara una Base de Datos centralizada, cabe destacar que LabVIEW nos ofrece una gran gama de “iconos” para programar cualquier tipo de aplicación.

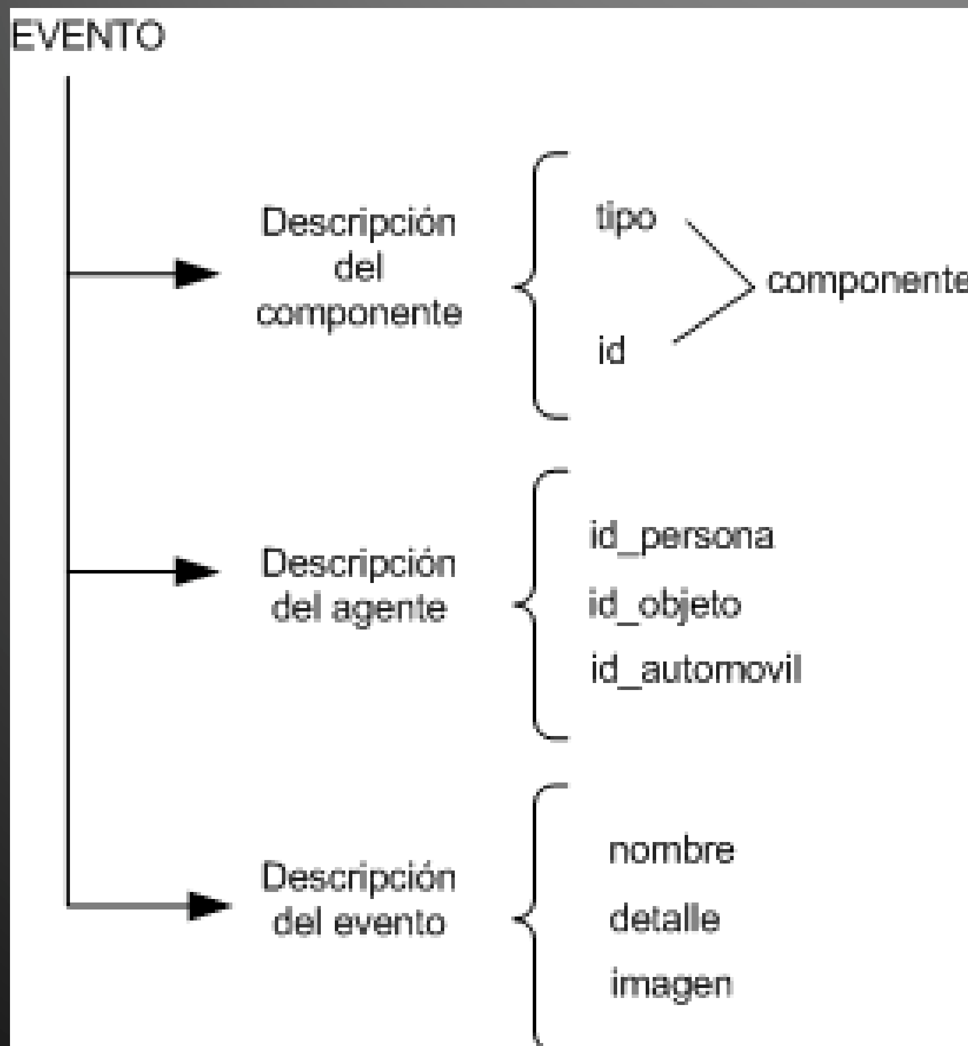

SERVIDOR
BASE DE DATOS
CENTRALIZADA
CLIENTE
CLIENTE
ROUTER / SWITCH
CLIENTE
CLIENTE

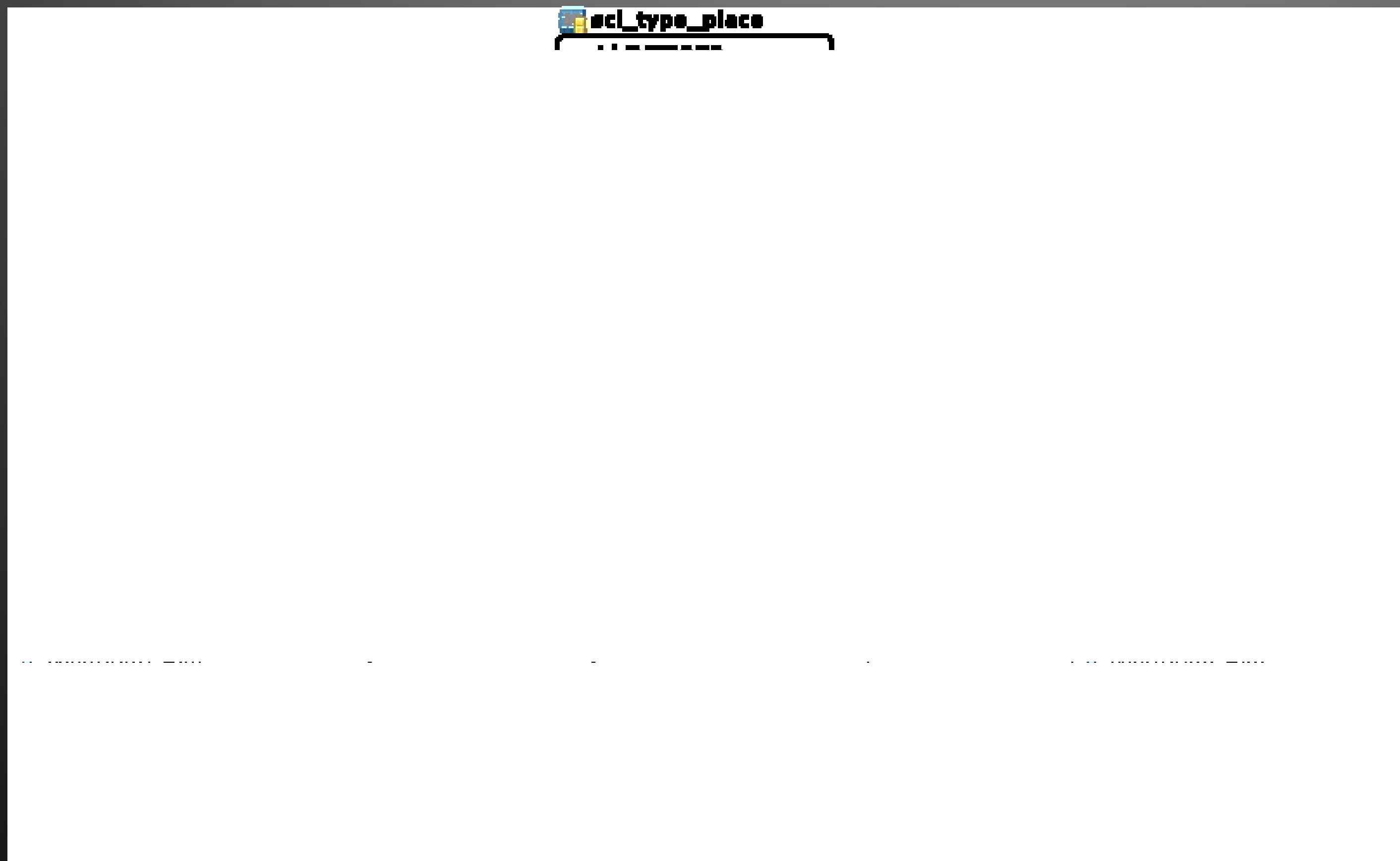
Identificación de Objetos, Sensores y Actuadores en una Localidad

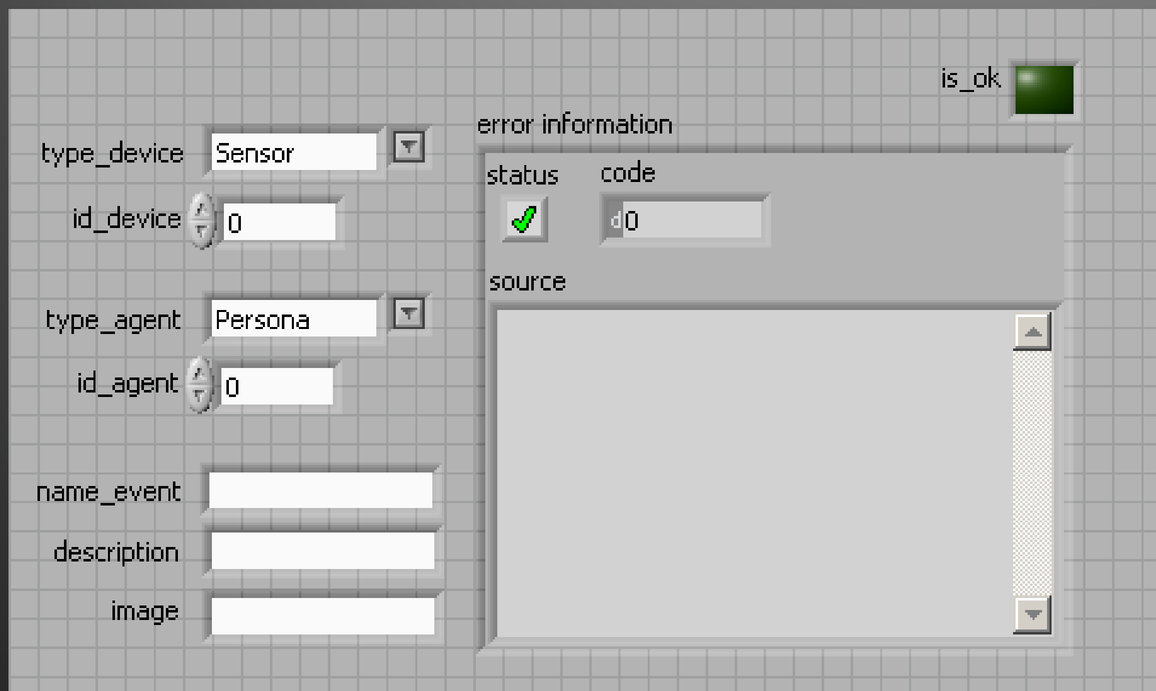


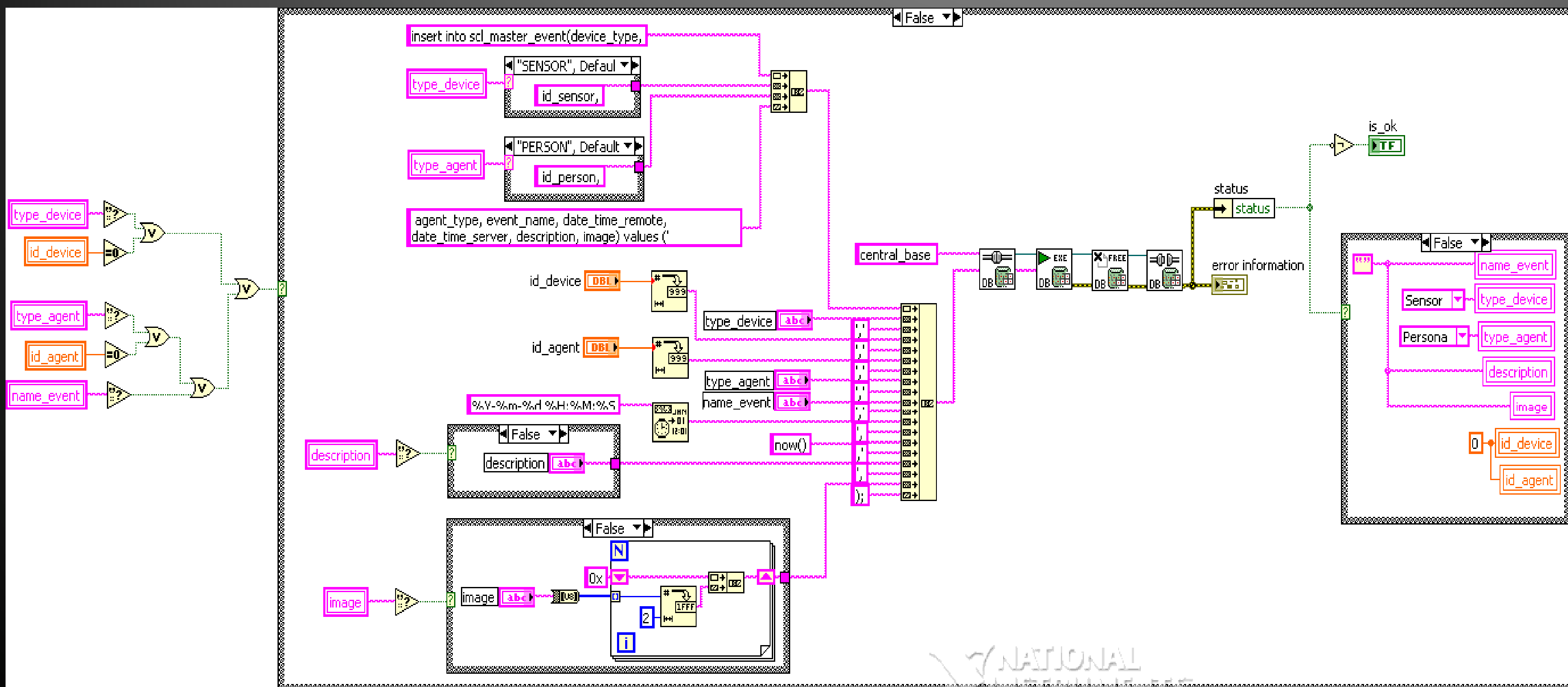
- Se define como un evento a las acciones originadas tanto por sensores y actuadores, esto quiere decir que una entrada produce una llamada a la función *send_event* y una salida también llamara a la misma función.


VEHÍCULOS
AGENTES
OBJETOS
PERSONAS
SENSORES
P
R
O
C
E
S
O
ACTUADORES
DISPOSITIVOS


EVENTO
Descripción del componente
tipo
id
componente
Descripción del agente
id_persona
id_objeto
id_automovil
Descripción del evento
nombre
detalle
imagen

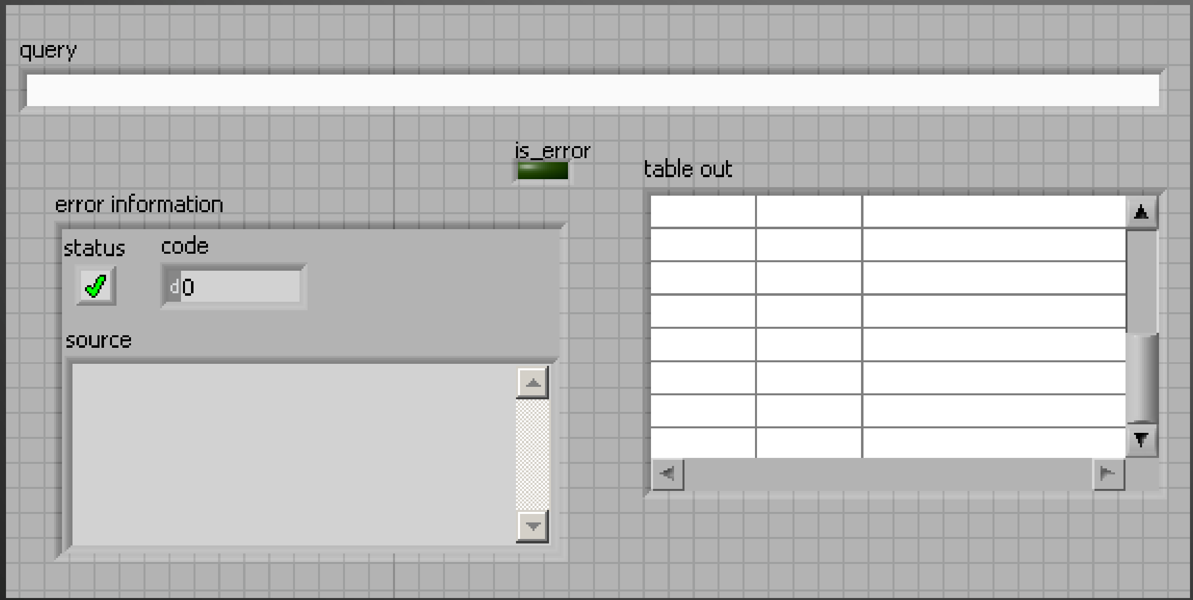

scl_type_place


is_ok
error information
type_device
Sensor
id_device
0
status
code
d0
type_agent
Persona
id_agent
0
source
name_event
description
image

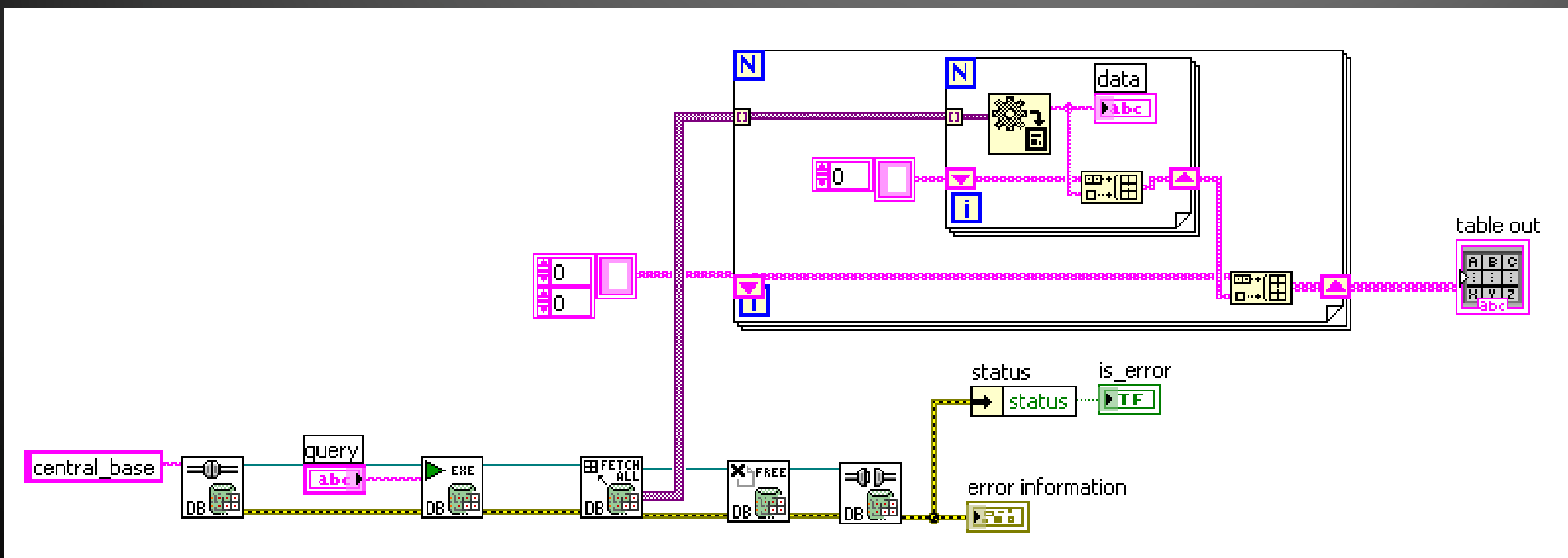


False
insert into scl_master_event(device_type,
"SENSOR", Defaul
type_device
id_sensor,
"PERSON", Default
type_agent
id_person,
agent_type, event_name, date_time_remote,
date_time_server, description, image) values ('
type_device
id_device
type_agent
id_agent
name_event
id_device
DBL
id_agent
DBL
type_device
type_agent
name_event
%Y-%m-%d %H:%M:%S
now()
description
False
description
image
False
image
0x
N
2
i
);
central_base
status
status
error information
is_ok
TF
False
name_event
Sensor
type_device
Persona
type_agent
description
image
0
id_device
id_agent

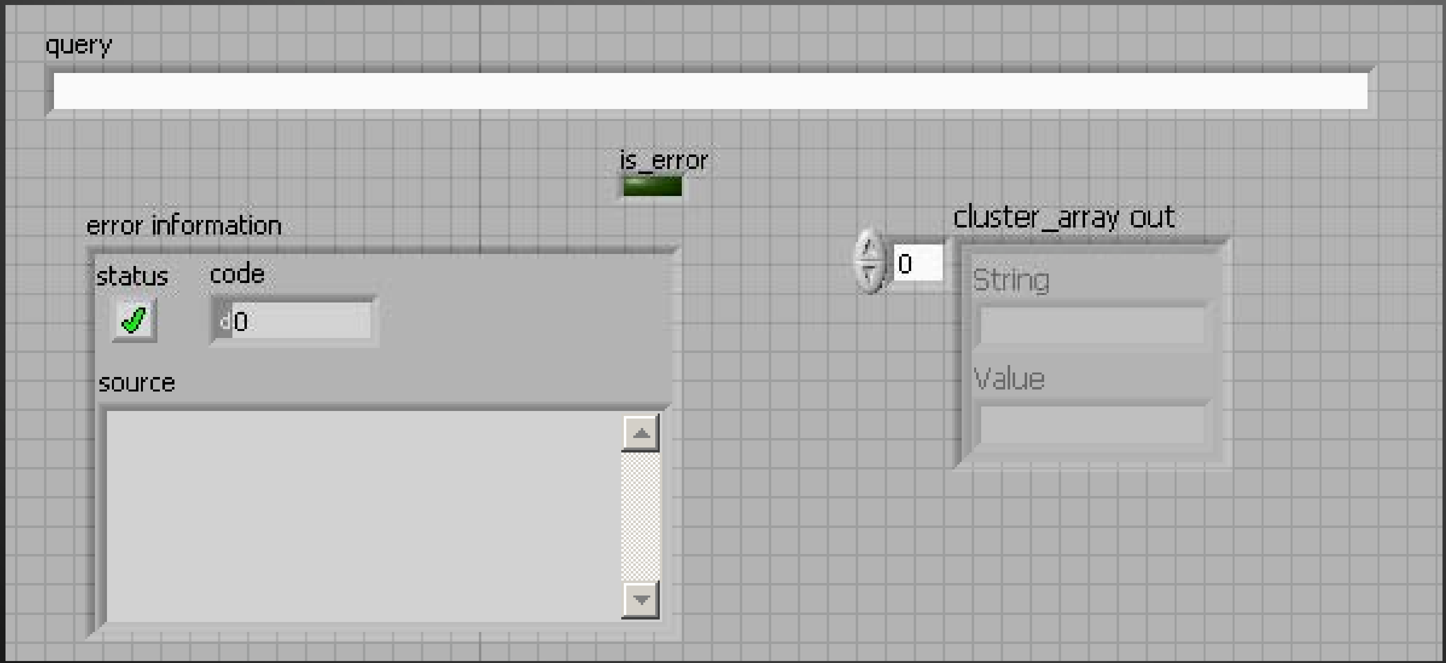
- db.vi (Front Panel)



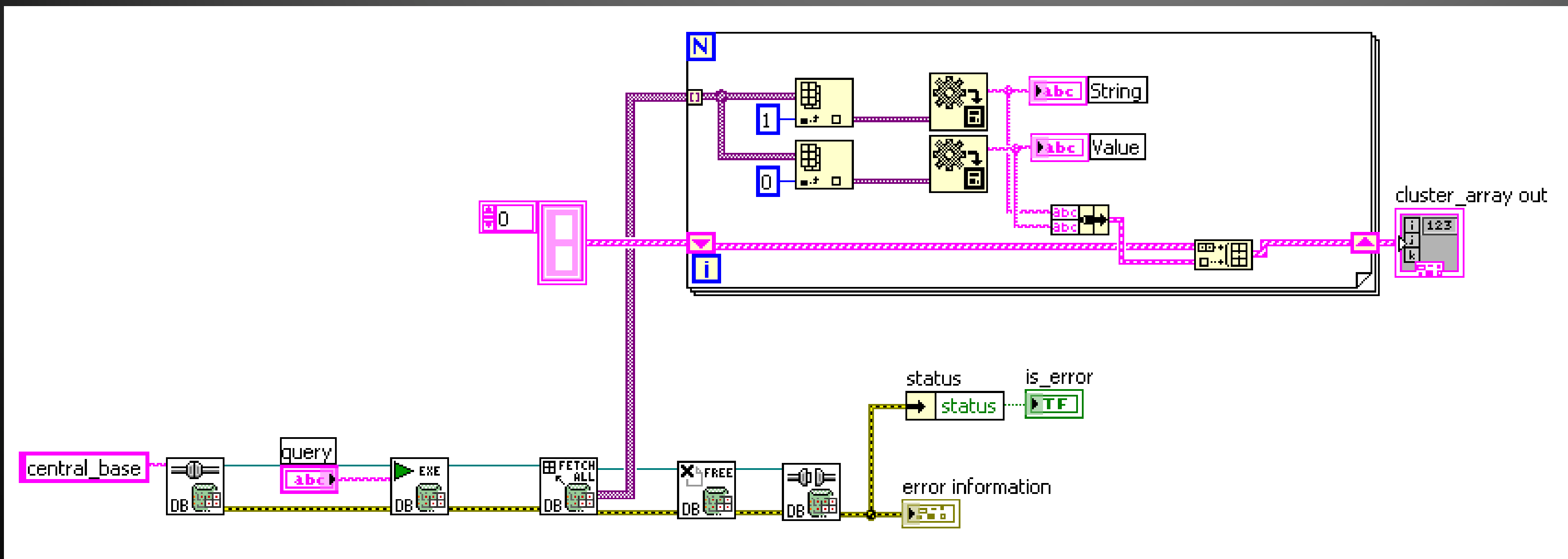
- db.vi (Block Diagram)



- db_cbx.vi (Front Panel)



- db_cbx.vi (Block Diagram)



- CREATE FUNCTION `getDataDevice`(type_device VARCHAR(8), id_sensor INTEGER(11), id_executer INTEGER(11)) RETURNS varchar(400)

- CREATE FUNCTION `getDataAgent`(type_agent VARCHAR(10), id_person INTEGER(11), id_object INTEGER(11), id_automovil INTEGER(11), type_device VARCHAR(8))
 RETURNS varchar(400)

- Podemos concluir que estableciendo el alcance (número de proyectos soportados) y elaborando una abstracción de la lógica de los Sistemas de Seguridades, hemos logrado obtener un diseño genérico de Base de Datos, capaz de soportar diferentes Sistemas de Seguridades.

- Algo de notar es que el diseño obtenido se basa en los equipos o terminales (sensores y actuadores) haciendo esto posible la adaptación de dicho diseño a un sistema de seguridad no soportado. Es decir el diseño esta abierto para otros Sistemas de Seguridades similares.

- De acuerdo a los resultados obtenidos, podemos concluir que en LabVIEW usando lenguaje G se pueden desarrollar aplicaciones de todo tipo y de gran calidad, las cuales no tienen nada que envidiar a las aplicaciones tradicionales que se desarrollan con líneas de código.