



INSTITUTO TECNOLÓGICO DE IZTAPALAPA

LENGUAJES Y AUTOMATAS 2

REPORTE DEL PROYECTO FINAL

FRONT-END

OZALDE ALDAY ELIZABETH VIRIDIANA

ZENDEJAS MENDEZ EDWIN ANTONIO

ROSALES JUAN DE DIOS ISAAC

MARTINEZ CORTES CRISTOPHER HERNAN



RESUMEN

El desarrollo del nuevo front-end se inició debido a la necesidad de un compilador el cual permita mejores diagnósticos, mejor integración con IDE, una licencia que sea compatible con productos comerciales y un compilador ágil que sea fácil de desarrollar y mantener. Todos estos fueron motivos para comenzar a trabajar en una nueva interfaz que pudiera satisfacer estas necesidades.

Clang proporciona una interfaz de lenguaje, en esto viene incluido varias herramientas para los lenguajes C, C++, RenderScript y toda su familia.

Sus características principales son las compilaciones rápidas y con poco uso de memoria, también un diagnóstico expresivo en el cual podemos encontrar en dónde están nuestros errores y la compatibilidad que tiene con Clang.

Se trata de un compilador reciente cuyo desarrollo comenzó en 2005 y su primera versión con licencia de código abierto se produjo en 2007. La versión actual de Clang es la 3.7, lanzada en septiembre de 2015, aunque el desarrollo es bastante activo y salen nuevas versiones del producto con cierta frecuencia.

Clang está escrito en C++ y puede ser utilizado en sistemas operativos basados en Unix. El compilador más conocido y usado para estos lenguajes ha sido y sigue siendo GCC (GNU Compiler Collection), desarrollado por el proyecto GNU para dar soporte a diversos lenguajes de programación. Sin embargo, existe una gran variedad de compiladores para estos y otros lenguajes.

Clang a día de hoy puede ser usado como una alternativa directa al compilador GCC, ofreciéndonos ciertas ventajas, entre las que destaca su diseño modular para el análisis estático del código.

Para facilitar su uso a usuarios acostumbrados a GCC, el proyecto Clang ha buscado que el frontend sea altamente compatible con este compilador. De esta manera, podemos encontrar que el uso por línea de comandos es muy similar y que muchas de las opciones incluidas son compartidas.

Para entender de Clang es necesario conocer acerca de LLVM

LLVM es un proyecto de código abierto que busca la creación de compiladores para cualquier lenguaje de programación, proporcionando la infraestructura necesaria para su desarrollo. No obstante, hay que destacar que, inicialmente, la idea del proyecto era centrarse en los lenguajes C y C++.



Dentro de un compilador, podemos distinguir entre frontend y backend: el frontend traduce el código fuente en una representación intermedia, la cual es entendible por el backend, que convierte finalmente esta representación en código máquina. LLVM, como backend, provee de las capas intermedias para conformar un sistema de compilación completo, tomando la representación interna generada por el frontend creado para un determinado lenguaje de programación.

INSTITUTO TECNOLÓGICO DE IZTAPALAPA CRONOGRAMA DE RESIDENCIAS PROFESIONALES																	
EMPRESA:		TELÉFONO:				FAX:				FOTO							
DIRECCIÓN:																	
ALUMNO:						No. DE CONTROL											
CARRERA:																	
NOMBRE DEL PROYECTO:						ASESOR EXTERNO:											
ASESOR INTERNO:						FECHA DE TERMINACIÓN											
FECHA DE INICIO:																	
OBJETIVO DEL PROYECTO:																	
ACTIVIDAD		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	P																
	R																
	P																
	R																
	P																
	R																
	P																
	R																
OBSERVACIONES:																	
ENTREGA DE REPORTES		ASESOR INTERNO															
		ESTUDIANTE															
		JEFE DEPTO.															

Av. Telcelmexicocolor 578, Col. Chivumpoc de Juárez, C.P. 03260,
Alcaldía de Iztapalapa, Ciudad de México Tel. 5773-8210, e-mail: division@iztapalapa.tecnm.mx
www.tecnm.mx | www.iztapalapa.tecnm.mx

ANÁLISIS DE RIESGOS

LISTA DE RIESGOS.

SOLUCIONES PROPUESTAS.

Falta de inversionistas: La falta de capital es un problema para el proyecto ya que se necesitan fondos para los recursos tangibles que se ocupan durante el proyecto.

Realizar un crowdfunding con personas interesadas en el tema del proyecto así mismo el objetivo es compartir el conocimiento y obtener recursos económicos a lo largo del desarrollo del proyecto.

Ataques externos: En front-end la seguridad no es la mejor y corre riesgos. Los ciberdelincuentes siempre tienen en su punto de mira a las empresas y sus sistemas, con el objetivo de robar información, tirar sus sistemas o utilizar sus recursos.

Ejercer una serie de métodos de seguridad para fortalecer la seguridad de la información y proteger el DDos ya que es lo que más atacan.

Desastres naturales: Es posible que se den situaciones que pongan en peligro los activos informáticos de la empresa como inundaciones o sobrecargas en la red eléctrica.

Una posible solución es que en todo momento estemos haciendo respaldos de activos, con el fin de no perder la información.

Conocimiento sobre el tema: no todos tienen el conocimiento sobre el tema y esto puede provocar un entorpecimiento en la comprensión.

Se puede impartir el conocimiento sobre el tema para comprender el front-end.

Falta de integrantes: la falta de integrantes puede ser un riesgo para el proyecto ya que si uno sufre un accidente o está de incapacidad, no habrá quien lo pueda suplir

Se puede contratar a alguien en caso de que esta situación pase, entonces el integrante extra lo podrá cubrir y podrá terminar el trabajo que se le encargó



JUSTIFICACIÓN

Los front-end son importantes para poder tener una fácil interacción con los programas, es decir simplificar el proceso usuario-máquina. El objetivo es minimizar la cantidad de errores humanos posibles y así evitar pérdidas de información o bien económicas.

debe caracterizarse por ser simple de entender, que visualmente sea agradable e intuitivo ya que de esta manera podremos minimizar la cantidad de dudas en cuanto a su manejo. Una de las ventajas es que el lenguaje en el que será creado es muy amplio y fácilmente podemos encontrar información sobre como utilizar sus librerías al igual que algunos ejemplos.



METODOLOGÍA

Para la metodología, escogimos una forma tradicional para hacer la interfaz utilizando programas para la elaboración de esta interfaz.

en NetBeans que fue lo que se usó para hacer la interfaz se hizo la programación necesaria para cada una de las partes que lleva este mismo.

se hizo de esta manera ya que ya se tiene conocimiento de netbeans y por eso se hizo más fácil el procedimiento para su creación.



REQUERIMIENTOS

FUNCIONALES Y NO FUNCIONALES

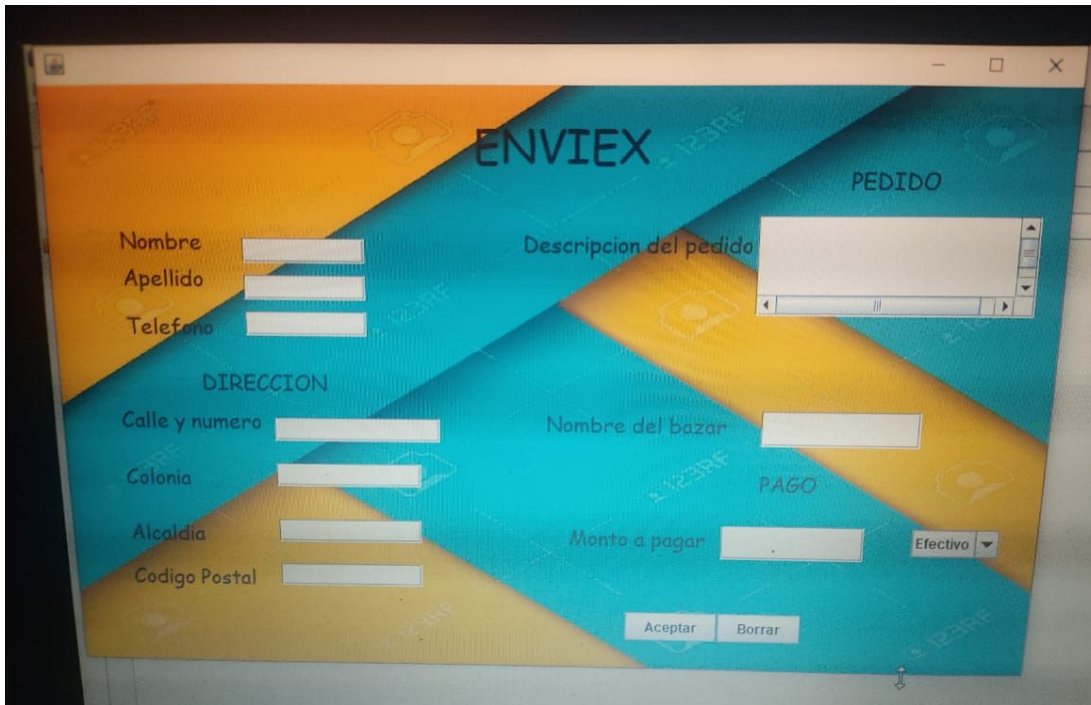
funcionales:

- puede ser manejado con facilidad
- la interfaz es segura
- registrar datos de usuarios
- imagen gráfica de la interfaz
- creación de usuarios

no funcionales:

- demasiados apartados en una sola interfaz
- no tener anuncios (en caso de ser necesario)

DISEÑO Y DESARROLLO



La imagen muestra una interfaz web titulada "ENVIEX PEDIDO" con un fondo de colores amarillos y azules. El formulario está dividido en varias secciones:

- Nombre:** Campos para "Nombre" y "Apellido".
- Telefono:** Campo para ingresar el número telefónico.
- DIRECCION:** Campos para "Calle y numero", "Colonia", "Alcaldia" y "Codigo Postal".
- Descripcion del pedido:** Campo de texto grande para describir el pedido.
- Nombre del bazar:** Campo para el nombre del bazar.
- PAGO:** Campo para "Monto a pagar" y un menú desplegable para seleccionar el método de pago (actualmente muestra "Efectivo").
- Botones:** "Aceptar" y "Borrar" al final del formulario.

para este proyecto de front-end hicimos una interfaz sencilla como ejemplo de este mismo.

Esta imagen es un da a conocer como se hizo la interfaz qué apartados tiene para su interacción.

las herramientas que se utilizaron para la interfaz, son:

- NetBeans

nota: esta interfaz está conectada a una base de datos hecha con otras herramientas. Esto se hizo para el correcto funcionamiento de la página web.

herramientas de la base de datos: Mysql



```
package clases;

import com.mysql.jdbc.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.swing.JOptionPane;

public class conectar {

    Connection conectar=null;

    public conectar(){

        try {
            Class.forName("com.mysql.jdbc.Driver");
            conectar= (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/");
            if (conectar !=null){
                JOptionPane.showMessageDialog(null, "La conexión ha sido exitosa");
            }
        } catch (Exception e){
            System.out.println("Error al conectar: "+ e);
        }
    }

    public Connection getConnection () {
        return conectar;
    }
}
```

```
public void setApellido(String apellido) {
    this.apellido = apellido;
}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

public String getCalle() {
    return calle;
}

public void setCalle(String calle) {
    this.calle = calle;
}

public String getColonia() {
    return colonia;
}

public void setColonia(String colonia) {
    this.colonia = colonia;
}

public String getAlcaldia() {
    return alcaldia;
}
```

```
public class pedidosDAO {

    public static String registrarpedidos (pedidosVO ped) throws SQLException{

        String resultado=null, last=null;
        conectar cc= new conectar ();
        Connection cn= cc.getConnection();
        PreparedStatement pst = null;
        String sql = "INSERT INTO pedidos VALUES (null, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

        try{
            pst=cn.prepareStatement(sql);
            pst.setString(1, ped.getNombre());
            pst.setString(2, ped.getApellido());
            pst.setString(3, ped.getTelefono());
            pst.setString(4, ped.getCalle());
            pst.setString(5, ped.getColonia());
            pst.setString(6, ped.getAlcaldia());
            pst.setString(7, ped.getCodigo_postal());
            pst.setString(8, ped.getDescripcion_pedido());
            pst.setString(9, ped.getNombre_bazar());
            pst.setString(10, ped.getMonto_pagar());
            pst.setString(11, ped.getTipo_pago());
            pst.execute();
            pst=cn.prepareStatement("SELECT MAX (id_pedido) AS id_ped FROM pedidos");
            ResultSet rs= pst.executeQuery();

            if(rs.next()){
                last = rs.getString(1);
            }

            resultado= "El pedido ha sido añadido id"+ last;
        }
    }
}
```

```
package clases;

import java.sql.PreparedStatement;
import javax.swing.JOptionPane;

public class conectar {

    Connection conectar=null;

    public conectar(){

        try {
            Class.forName("com.mysql.jdbc.Driver");
            conectar= (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/");
            if (conectar !=null){
                JOptionPane.showMessageDialog(null, "La conexión ha sido exitosa");
            }
        } catch (Exception e){
            System.out.println("Error al conectar: "+ e);
        }
    }

    public Connection getConnection () {
        return conectar;
    }

    public void desconectar(){
        conectar=null;
    }
}
```



```
Source History
1
2 package classVO;
3
4 import javax.accessibility.AccessibleContext;
5
6
7 public class pedidosVO {
8
9     private int id_pedido;
10    private String nombre;
11    private String apellido;
12    private String telefono;
13    private String calle;
14    private String colonia ;
15    private String alcaldia;
16    private String codigo_postal;
17    private String descripcion_pedido;
18    private String Nombre_bazar;
19    private String monto_pagar;
20    private String tipo_pago;
21    private String resultado;
22
23    public String getResultado() {
24        return resultado;
25    }
26
27    public void setResultado(String resultado) {
28        this.resultado = resultado;
29    }
30
31    public int getId_pedido() {
32        return id_pedido;
33    }
34}
```