

Codificación de módulos del software

Jorge Armando Toloza Carvajal

**Análisis y desarrollo de software
Ficha # 2627106**

Instructor: Carlos Ernesto Lizarazo Sierra

**Regional Santa fe de Antioquia
Septiembre 2023**

Informe Técnico: Software para Call Center

I. Resumen Ejecutivo

Descripción: En esta sección, proporcionamos una visión general de alto nivel del proyecto. Debe ser lo suficientemente claro como para que cualquier interesado no técnico pueda comprenderlo.

Ejemplo: "El proyecto consiste en desarrollar un sistema de software integral para un centro de llamadas con el objetivo de mejorar la eficiencia operativa y la satisfacción del cliente."

Conclusión: Se establece la importancia del proyecto y se destaca el valor que aportará a la organización.

II. Descripción del Proyecto

Descripción: Aquí se detalla el propósito y la necesidad del proyecto, así como la visión general del software. Se debe explicar por qué se necesita el software y cómo se alinea con los objetivos del negocio.

Ejemplo: "El software se desarrolla para gestionar las llamadas entrantes y salientes, almacenar información del cliente y agilizar los procesos de atención al cliente."

Conclusión: Se establece la base de la necesidad del proyecto y se vincula con los objetivos empresariales.

III. Análisis de Requisitos

Descripción: En esta sección, se enumeran y describen los requisitos específicos del software. Esto incluye tanto los requisitos funcionales como los no funcionales.

Ejemplo: "Los requisitos funcionales incluyen la capacidad de enrutar llamadas entrantes, almacenar registros de llamadas y generar informes. Los requisitos no funcionales incluyen la seguridad de los datos y la escalabilidad del sistema."

Conclusión: Establece la base para el diseño y desarrollo posterior del software.

IV. Diseño del Software

Descripción: Se describe cómo se estructurará el software, incluida la arquitectura del sistema, el diseño de la interfaz de usuario y la estructura de la base de datos.

Ejemplo: "El sistema se diseñará utilizando una arquitectura de tres capas con una base de datos relacional MySQL. La interfaz de usuario será intuitiva y fácil de usar para los agentes de llamadas."

Conclusión: Establece la base para el desarrollo y la implementación.

V. Desarrollo

Descripción: En esta sección, se explica cómo se llevará a cabo el desarrollo del software, incluida la elección de una metodología (por ejemplo, Agile o Scrum).

Ejemplo: "El desarrollo se llevará a cabo utilizando la metodología Scrum, con sprints de dos semanas y una revisión regular de avances. Se utilizarán tecnologías Java y Spring para la implementación."

Conclusión: Establece un marco para la ejecución del proyecto.

VI. Pruebas

Descripción: Se describe cómo se realizarán las pruebas, incluidas las pruebas unitarias, de integración y de aceptación.

Ejemplo: "Se realizarán pruebas unitarias después de cada sprint y pruebas de integración antes de la entrega de cada versión. Las pruebas de aceptación se llevarán a cabo con usuarios finales."

Conclusión: Destaca la importancia de garantizar la calidad del software.

VII. Implementación y Despliegue

Descripción: Se explica cómo se implementará y desplegará el software en un entorno de producción.

Ejemplo: "La implementación se realizará de manera gradual en varios departamentos, comenzando por el equipo de soporte al cliente. El despliegue se realizará en servidores internos."

Conclusión: Establece el proceso para llevar el software al mundo real.

VIII. Capacitación y Documentación

Descripción: Se explica cómo se capacitará a los usuarios y se documentará el software.

Ejemplo: "Se proporcionará capacitación en persona y recursos en línea para los usuarios finales. Se creará documentación técnica y manuales de usuario."

Conclusión: Garantiza que los usuarios puedan utilizar eficazmente el software.

IX. Mantenimiento y Soporte

Descripción: Se explica cómo se gestionará el mantenimiento continuo y el soporte para el software.

Ejemplo: "Se establecerá un equipo de soporte técnico para abordar problemas y mejoras. Se llevará un registro de problemas y solicitudes de cambios."

Conclusión: Establece la importancia de mantener el software en funcionamiento.

X. Presupuesto y Recursos

Descripción: Se detallan los costos y recursos necesarios para el proyecto.

Ejemplo: "Se estima un presupuesto de \$100,000, que incluye salarios del equipo, hardware y software. Se asignarán tres desarrolladores y un administrador de proyecto."

XI. Diseño de Software - Diagrama de Clases

Descripción: En esta sección, se incluye un diagrama de clases que representa la estructura y las relaciones entre las clases del software. Esto proporciona una vista clara de la organización del código.

Ejemplo: Adjunta un diagrama de clases que muestra las clases principales como Usuario, Cliente, Cita, etc., y cómo se relacionan entre sí.

Conclusión: El diagrama de clases ayuda a los desarrolladores a comprender la arquitectura del software.

XII. Diseño de Software - Diagramas de Casos de Uso

Descripción: Se presentan los diagramas de casos de uso que describen cómo interactúan los actores con el sistema y qué funcionalidades ofrece el software.

Ejemplo: Incluye diagramas de casos de uso que ilustran cómo un agente de llamadas crea una cita o cómo un administrador accede a los informes.

Conclusión: Los diagramas de casos de uso ayudan a comprender las interacciones del usuario con el software.

XIII. Historias de Usuario

Descripción: Se describen historias de usuario que detallan escenarios específicos de uso del software desde la perspectiva del usuario.

Ejemplo: Ejemplos de historias de usuario incluyen "Como agente de llamadas, quiero poder registrar los detalles de la llamada entrante" o "Como administrador, quiero generar informes mensuales de actividad".

Conclusión: Las historias de usuario guían el desarrollo y las pruebas al centrarse en las necesidades del usuario.

XIV. Diseños y Prototipos

Descripción: Se menciona cómo se crearán diseños de interfaz de usuario y prototipos para visualizar la apariencia y la navegación del software.

Ejemplo: "Utilizaremos herramientas de diseño como Adobe XD para crear prototipos interactivos que muestren cómo se verá y se sentirá el software".

Conclusión: Los diseños y prototipos ayudan a alinear las expectativas del usuario y el equipo de desarrollo.

XV. Plan de Trabajo

Descripción: Se presenta un plan detallado con fechas clave, hitos y asignación de tareas para llevar a cabo el proyecto de desarrollo de software para el centro de llamadas.

Ejemplo: Incluye un calendario con fechas de inicio y finalización para el diseño, desarrollo, pruebas y lanzamiento del software.

Conclusión: El plan de trabajo proporciona una guía clara para la ejecución exitosa del proyecto.

XVI. Conclusiones y Próximos Pasos

Descripción: Se resume el informe técnico y se destacan los próximos pasos, como el inicio del desarrollo, la capacitación del equipo y la revisión continua del progreso.

Ejemplo: "El informe técnico proporciona una base sólida para el desarrollo del software. El próximo paso es comenzar con la fase de diseño y desarrollo".

Conclusión: Se enfatiza la importancia de seguir el plan y se establece una visión clara del camino a seguir.

Este informe técnico ampliado y el plan de trabajo proporcionan una visión completa del proyecto de desarrollo de software para el centro de llamadas, desde la concepción hasta la implementación y el mantenimiento continuo. Estos elementos son fundamentales para una gestión eficaz del proyecto y para garantizar el éxito en la entrega del software.

XVII. Estándares de codificación en NetBeans:

- Nombramiento de variables
- Nombramiento de métodos
- Nombramiento de clases
- Nombramiento de paquetes
- Debe tener funcionalidades de inserción, consulta, actualización y eliminación

/*

* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
*/

```
package javaapplication1;
```

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.logging.Level;  
import java.util.logging.Logger;
```

```
public class JavaApplication1 {
```

```
    private static final String DB_URL = "jdbc:mysql://127.0.0.1:3306/Callbuc_Dial"; // Usar la  
    IP local
```

```
    private static final String DB_USER = "root";  
    private static final String DB_PASSWORD = "";
```

```
    public static void main(String[] args) {  
        JavaApplication1 app = new JavaApplication1();
```

```
        Connection conexion = null;  
        Statement statement = null;  
        ResultSet rs = null;
```

```
        try {  
            // Cargar el controlador JDBC  
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            // Establecer la conexión a la base de datos  
            conexion = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
```

```
            // Crear una declaración SQL  
            statement = conexion.createStatement();
```

```
            // Ejecutar una consulta (ejemplo)  
            String consultaSQL = "SELECT * FROM tabla_ejemplo";  
            rs = statement.executeQuery(consultaSQL);
```

```
            // Procesar los resultados (ejemplo)  
            while (rs.next()) {  
                int id = rs.getInt("id");  
                String nombre = rs.getString("nombre");  
                System.out.println("ID: " + id + ", Nombre: " + nombre);  
            }
```

```
            // Ejemplo de inserción de datos
```

```

        app.insertarDatos("NuevoNombre", 30);

    } catch (ClassNotFoundException ex) {
        Logger.getLogger(JavaApplication1.class.getName()).log(Level.SEVERE, "Error al
cargar el controlador JDBC", ex);
    } catch (SQLException ex) {
        Logger.getLogger(JavaApplication1.class.getName()).log(Level.SEVERE, "Error SQL:
" + ex.getMessage(), ex);
    } finally {
        // Cerrar recursos en un bloque finally
        try {
            if (rs != null) {
                rs.close();
            }
            if (statement != null) {
                statement.close();
            }
            if (conexion != null) {
                conexion.close();
            }
        } catch (SQLException ex) {
            Logger.getLogger(JavaApplication1.class.getName()).log(Level.SEVERE, "Error al
cerrar recursos", ex);
        }
    }
}

```

```

public void insertarDatos(String nombre, int edad) {
    Connection conexion = null;
    PreparedStatement preparedStatement = null;

    try {
        conexion = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
        String insercionSQL = "INSERT INTO tabla_ejemplo (nombre, edad) VALUES (?, ?)";
        preparedStatement = conexion.prepareStatement(insercionSQL);
        preparedStatement.setString(1, nombre);
        preparedStatement.setInt(2, edad);
        preparedStatement.executeUpdate();
        System.out.println("Datos insertados con éxito.");
    } catch (SQLException ex) {
        Logger.getLogger(JavaApplication1.class.getName()).log(Level.SEVERE, "Error al
insertar datos: " + ex.getMessage(), ex);
    } finally {
        // Cerrar recursos en un bloque finally
        try {
            if (preparedStatement != null) {
                preparedStatement.close();
            }
            if (conexion != null) {

```

```
        conexion.close();
    }
} catch (SQLException ex) {
    Logger.getLogger(JavaApplication1.class.getName()).log(Level.SEVERE, "Error al
cerrar recursos", ex);
}
}
}
```