



### **Evidencia de producto:**

GA7-220501096-AA3-EV01 codificación de módulos del software Stand alone, web y móvil de acuerdo con el proyecto a desarrollar

### **Presentado por:**

Luis Enel palacios córdoba

### **Instructor:**

Milton Iván Barbosa Gaona

Centro de la Productividad del Diseño y la Productividad Empresarial

Girardot, Cundinamarca

### **Análisis y Desarrollo de Software**

Ficha: 27583347

(SENA)

### **Tabla de contenido**





Introducción .....	3
Objetivo General .....	3
Objetivos Específicos .....	3
Desarrollo del Proyecto .....	4
1. Configuración de NetBeans para JavaFX .....	4
2. Desarrollo de los Módulos del Software .....	5
Ejemplo de Código en JavaFX: .....	5
3. Aplicación de Estándares de Codificación .....	7
4. Implementación de Herramientas de Versionamiento .....	8
Conclusión.....	8
Bibliografía.....	9





## Introducción

Este documento proporciona instrucciones detalladas sobre cómo configurar y desarrollar el software moderno en modalidad stand-alone utilizando NetBeans y JavaFX. El objetivo es garantizar una implementación eficiente y conforme a los estándares de codificación para el desarrollo de una aplicación robusta y funcional.

## Objetivo General

Configurar y codificar el software moderno en modalidad stand-alone utilizando NetBeans y JavaFX, siguiendo las mejores prácticas de desarrollo y asegurando que el sistema cumpla con los requisitos establecidos y estándares de codificación.

### Objetivos Específicos

1. **Configurar NetBeans para JavaFX:** Preparar el entorno de desarrollo en NetBeans para trabajar con JavaFX.
2. **Desarrollar los Módulos del Software:** Codificar los módulos del software moderno de acuerdo con los artefactos del ciclo del software.
3. **Aplicar Estándares de Codificación:** Implementar el código cumpliendo con los estándares de codificación y documentarlo adecuadamente.
4. **Implementar Herramientas de Versionamiento:** Utilizar herramientas de control de versiones para gestionar el desarrollo del proyecto.





## Desarrollo del Proyecto

### 1. Configuración de NetBeans para JavaFX

#### 1. Instalar NetBeans:

- Asegúrate de tener la versión más reciente de NetBeans instalada en tu sistema. Puedes descargarlo desde [el sitio oficial de NetBeans](#).

#### 2. Instalar el JDK y JavaFX:

- Descarga e instala el [JDK](#) si aún no lo has hecho.
- Descarga el SDK de JavaFX desde el sitio de Gluon.

#### 3. Configurar NetBeans para JavaFX:

- Abre NetBeans y crea un nuevo proyecto seleccionando File > New Project.
- En el diálogo de creación de proyectos, selecciona Java y luego JavaFX como tipo de proyecto.
- Si JavaFX no está visible en las opciones, asegúrate de que el plugin de JavaFX esté instalado en NetBeans. Puedes instalarlo desde el Plugins en el menú Tools.

#### 4. Agregar Librerías de JavaFX:

- Ve a las propiedades del proyecto (Right-click on project > Properties).
- En la sección Libraries, añade las librerías de JavaFX que has descargado. Asegúrate de incluir los módulos de JavaFX necesarios (e.g., javafxcontrols, javafx-fxml, etc.).





## 2. Desarrollo de los Módulos del Software

### 1. Crear la Estructura del Proyecto:

- Una vez configurado, comienza creando la estructura del proyecto según los requisitos.
- Define los paquetes para organizar el código, como model, view, controller, etc.

### 2. Codificar el Proyecto:

- **Interfaz de Usuario:** Usa JavaFX para diseñar la interfaz gráfica. Crea clases que extiendan Application y utilicen Scene, Stage, y otros componentes de JavaFX.
- **Lógica de Negocio:** Implementa la lógica contable en clases separadas, siguiendo los diagramas de clases y otros artefactos del ciclo de software.
- **Manejo de Eventos:** Agrega manejadores de eventos para los controles de la interfaz de usuario (botones, campos de texto, etc.).

### Ejemplo de Código en JavaFX:

```
package com.auto_quibdo.moderno;
```

```
import javafx.application.Application;
```

```
import javafx.scene.Scene; import
```

```
javafx.scene.control.Button; import
```





```
javafx.scene.layout.VBox; import  
  
javafx.stage.Stage;  
  
public class MainApp extends Application {  
  
    @Override    public void start(Stage primaryStage) {  
  
        // Configuración de la ventana principal  
  
        primaryStage.setTitle("Software Contable INTCOR");  
  
        // Crear un botón de ejemplo  
  
        Button btn = new Button("Click aquí");  
  
        btn.setOnAction(e -> System.out.println("Botón clickeado"));  
  
        // Crear el layout  
  
        VBox vbox = new VBox(btn);  
  
        // Crear la escena  
  
        Scene scene = new Scene(vbox, 300, 200);  
  
        primaryStage.setScene(scene);
```





```
// Mostrar la ventana

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}
```

### 3. Documentar el Código:

- Incluye comentarios descriptivos en el código para explicar su funcionalidad y lógica.
- Asegúrate de que las variables, métodos y clases tengan nombres claros y significativos.


### 3. Aplicación de Estándares de Codificación

#### 1. Seguir Convenciones de Codificación:

- Usa convenciones estándar de JAVA, como nombres en camelCase para métodos y variables, y PascalCase para nombres de clases.
- Mantén el código limpio y organizado, siguiendo principios de diseño como la separación de responsabilidades.

#### 2. Comentarios:



- 
- Añade comentarios en puntos clave del código, especialmente en métodos y clases complejas.
  - Usa comentarios de documentación (`/** ... */`) para describir la funcionalidad de métodos y clases.

#### **4. Implementación de Herramientas de Versionamiento**

##### **1. Configurar Git en NetBeans:**

- Si no tienes Git instalado, descárgalo e instálalo desde [Git](#).
- En NetBeans, abre tu proyecto y selecciona Team > Git > Initialize Repository para crear un repositorio local.
- Realiza commits regulares para guardar los cambios (Team > Git > Commit).

##### **2. Subir a un Repositorio Remoto:**

- Si utilizas un servicio como GitHub, crea un repositorio en el servicio.
- En NetBeans, usa Team > Git > Remote > Push para subir tu proyecto al repositorio remoto.

### **Conclusión**

La configuración y desarrollo del software contable INTCOR en modalidad standalone utilizando NetBeans y JavaFX implica la correcta configuración del entorno de desarrollo, la implementación meticulosa de los módulos del software, y el cumplimiento de estándares de codificación. Al seguir estos pasos y utilizar herramientas adecuadas de versionamiento, se asegurará el desarrollo de una aplicación contable sólida y eficiente.





## Bibliografía

1. **Dea, C., Heckler, M., Grunwald, G., & Pereda, J. (2014).** *JavaFX 8: Introduction by Example*. Apress.

Este libro ofrece ejemplos prácticos y explicaciones detalladas sobre la creación de aplicaciones con JavaFX, ideal para aprender a usar esta tecnología en proyectos de desarrollo.

2. **Dea, C., Heckler, M., Grunwald, G., & Pereda, J. (2018).** *JavaFX 11 by Example*.





Apress.

Un recurso actualizado que cubre las características más recientes de JavaFX con ejemplos prácticos para el desarrollo de aplicaciones ricas en interfaz gráfica.

3. **Schildt, H. (2020).** *Java: The Complete Reference*. McGraw-Hill.

Una referencia exhaustiva sobre el lenguaje de programación Java, que incluye secciones sobre JavaFX y sus aplicaciones.

4. **Chacon, S., & Straub, B. (2014).** *Pro Git*. Apress.

Un libro completo sobre el control de versiones con Git, disponible de forma gratuita en línea, que cubre desde conceptos básicos hasta técnicas avanzadas.

5. **Oracle.** *JavaFX Documentation*. Recuperado de <https://openjfx.io/>

Documentación oficial de JavaFX que incluye guías, tutoriales y referencias de API para el desarrollo con JavaFX.

6. **NetBeans.** *NetBeans JavaFX Tutorial*. Recuperado de

<https://netbeans.apache.org/kb/docs/java/javafx.html>

Tutoriales oficiales de NetBeans para configurar y desarrollar aplicaciones JavaFX.

7. **JavaFXpert.** *Configuring JavaFX in NetBeans*. Recuperado de

<https://www.javafxpert.com/2020/06/21/how-to-configure-javafx-in-netbeans/> Guía paso a paso para configurar JavaFX en NetBeans, incluyendo la instalación y configuración de bibliotecas.





8. **Oracle.** *JavaFX Tutorials*. Recuperado de [https://docs.oracle.com/javafx/2/get\\_started/jfxpub-get\\_started.htm](https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm) Tutoriales proporcionados por Oracle que cubren la creación de aplicaciones JavaFX desde el inicio hasta niveles avanzados.
9. **Git SCM.** *Git Documentation*. Recuperado de <https://git-scm.com/doc> Documentación oficial de Git que ofrece guías completas sobre el uso de Git para el control de versiones.
10. **GitHub.** *GitHub Docs*. Recuperado de <https://docs.github.com/en> Documentación y tutoriales para el uso de GitHub, incluyendo el manejo de repositorios y control de versiones colaborativo.
11. **Coursera.** *Java Programming and Software Engineering Fundamentals*.  
Recuperado de <https://www.coursera.org/specializations/java-programming>  
Un curso especializado en programación en Java, que incluye módulos sobre JavaFX y desarrollo de aplicaciones.
12. **Udemy.** *JavaFX Tutorial for Beginners*. Recuperado de <https://www.udemy.com/course/javafx-tutorial-for-beginners/>  
Curso en línea que proporciona una introducción completa a JavaFX, ideal para desarrolladores principiantes.
13. **Pluralsight.** *JavaFX Fundamentals*. Recuperado de <https://www.pluralsight.com/courses/javafx-fundamentals>



Curso detallado sobre los fundamentos de JavaFX, adecuado para el desarrollo de aplicaciones con esta tecnología.

