

INDICE

1.Miembros del trabajo.....	
2.Explicación del trabajo realizado.....	
3.Estructura utilizada.....	
4.GitHub.....	
5.Mejoras.....	
6.Pruebas realizadas.....	
7.Archivo ejecutable.....	
8.Problemas encontrados.....	
9.Bibliografia.....	

RECURSOS UTILIZADOS:

- NetBeans IDE 8.1
- SceneBuilder

Lenguajes de programación utilizados:

- JavaFX
- XML
- CSS

Descripción del proyecto NetBeans:

- Proyecto principal:
 - Nombre del Proyecto: Gestor inventario de Componentes Informáticos

1.MIEMBROS DEL TRABAJO:

En el proyecto realizado hemos participado 5 personas:

- Jorge Melero Muñoz
- Jorge Diaz Megias
- Julián Lorenzo Puente
- Rodrigo Muñoz López
- Sergio Cabello Serrano

De la clase de Desarrollo de Aplicaciones Multiplataforma del Centro San Pablo CEU

2. Explicación del trabajo realizado:

Nuestro grupo ha realizado un programa java que consiste en un gestor inventario/stock de almacén. El tema escogido para nuestro trabajo ha sido los" Componentes Informáticos" (graficas, placas base, periféricos...).

En nuestro proyecto hemos tenido que añadir de cada componente una foto, el nombre, una pequeña descripción del producto, el precio, el stock, generar un código de barras, y las Fechas de los productos.



```
public ObservableList getDatosComponente() {
    return datosComponentes;
}

@Override
public void start(Stage escenarioPrincipal) {
    this.escenarioPrincipal = escenarioPrincipal;
    //Pongo el titulo
    this.escenarioPrincipal.setTitle("Gestor Inventario");
    //Inicializa el layout principal
    initLayoutPrincipal();
    //muestra la vista persona
    muestraInventario();
}

public void initLayoutPrincipal() {
    //cargar el layout principal mediante la vista principal
    FXMLLoader loader = new FXMLLoader();
    URL location = GestorInventario.class.getResource("/view/VistaPrincipal.fxml");
    loader.setLocation(location);
    try {
        VistaPrincipalController = loader.load();
    } catch (IOException ex) {
    }
    //Doy el controlador acceso a la aplicación principal
    VistaPrincipalController controller = loader.getController();
    controller.setGestorInventario(this);

    Scene escena = new Scene(VistaPrincipal);
    escenarioPrincipal.setScene(escena);
    escenarioPrincipal.show();

    //Muestro la escena
    escenarioPrincipal.show();

    //Intento cargar el ultimo archivo abierto
    File archivo = getRutaArchivoComponentes();
    if (archivo != null) {
        cargaComponentes(archivo);
    }
}
```

```

public void muestraInventario() {
    FXMLoader loader = new FXMLoader();
    URL location = GestorInventario.class.getResource("/view/VistaInventario.fxml");
    loader.setLocation(location);
    try {
        vistaInventario = loader.load();
    } catch (IOException ex) {
    }

    //añado a centro la vista inventario
    layoutPrincipal.setCenter(vistaInventario);

    VistaComponenteController controller = loader.getController();
    controller.setGestorInventario(this);
}

public void muestraDetalle(Componente componente) {
    FXMLoader loader = new FXMLoader();
    URL location = GestorInventario.class.getResource("/view/VistaDetalle.fxml");
    loader.setLocation(location);
    try {
        vistaDetalle = loader.load();
    } catch (IOException ex) {
    }

    Stage escenarioDetalle = new Stage();
    escenarioDetalle.setTitle("Detalles");
    escenarioDetalle.initModality(Modality.WINDOW_MODAL);
    escenarioDetalle.initOwner(escenarioPrincipal);
    Scene escena = new Scene(vistaDetalle);
    escenarioDetalle.setScene(escena);

    VistaDetalleController controller = loader.getController();
    controller.setEscenarioDetalle(escenarioDetalle);
    controller.setComponente(componente);
    escenarioDetalle.showAndWait();
}
}

```

```

public boolean muestraNuevo(Componente componente) {
    FXMLoader loader = new FXMLoader();
    URL location = GestorInventario.class.getResource("/view/VistaNuevo.fxml");
    loader.setLocation(location);
    try {
        vistaNuevo = loader.load();
    } catch (IOException ex) {
    }

    Stage escenarioNuevo = new Stage();
    escenarioNuevo.setTitle("Nuevo Componente");
    escenarioNuevo.initModality(Modality.WINDOW_MODAL);
    escenarioNuevo.initOwner(escenarioPrincipal);
    Scene escena = new Scene(vistaNuevo);
    escenarioNuevo.setScene(escena);

    VistaNuevoController controller = loader.getController();
    controller.setEscenarioNuevo(escenarioNuevo);
    controller.setComponente(componente);
    //controller.isGuardarClickado();
    escenarioNuevo.showAndWait();

    return controller.isGuardarClickado();
}

```

```

//Vista editarComponente
public boolean muestraEditarComponente(Componente componente) {
    //Cargo la vista nuevo a partir de VistaNuevo.fxml
    FXMLoader loader = new FXMLoader();
    URL location = GestorInventario.class.getResource("/view/VistaNuevo.fxml");
    loader.setLocation(location);
    try {
        vistaComponente = loader.load();
    } catch (IOException ex) {
        Logger.getLogger(GestorInventario.class.getName()).log(Level.SEVERE, null, ex);
        return false;
    }

    //Creo el escenario de edición (con modal) y establezco la escena
    Stage escenarioEdicion = new Stage();
    escenarioEdicion.setTitle("Editar Componente");
    escenarioEdicion.initModality(Modality.WINDOW_MODAL);
    escenarioEdicion.initOwner(escenarioPrincipal);
    Scene escena = new Scene(vistaComponente);
    escenarioEdicion.setScene(escena);

    //Asigno el escenario de edición y el componente seleccionado al controlador
    VistaNuevoController controller = loader.getController();
    controller.setEscenarioNuevo(escenarioEdicion);
    controller.setComponente(componente);

    //Muestro el dialogo objeto que el usuario lo cierre
    escenarioEdicion.showAndWait();

    //devuelvo el botón pulsado
    return controller.isGuardarClickado();
}

```

```
//Carga componentes de un fichero
public void cargaComponentes(File archivo) {

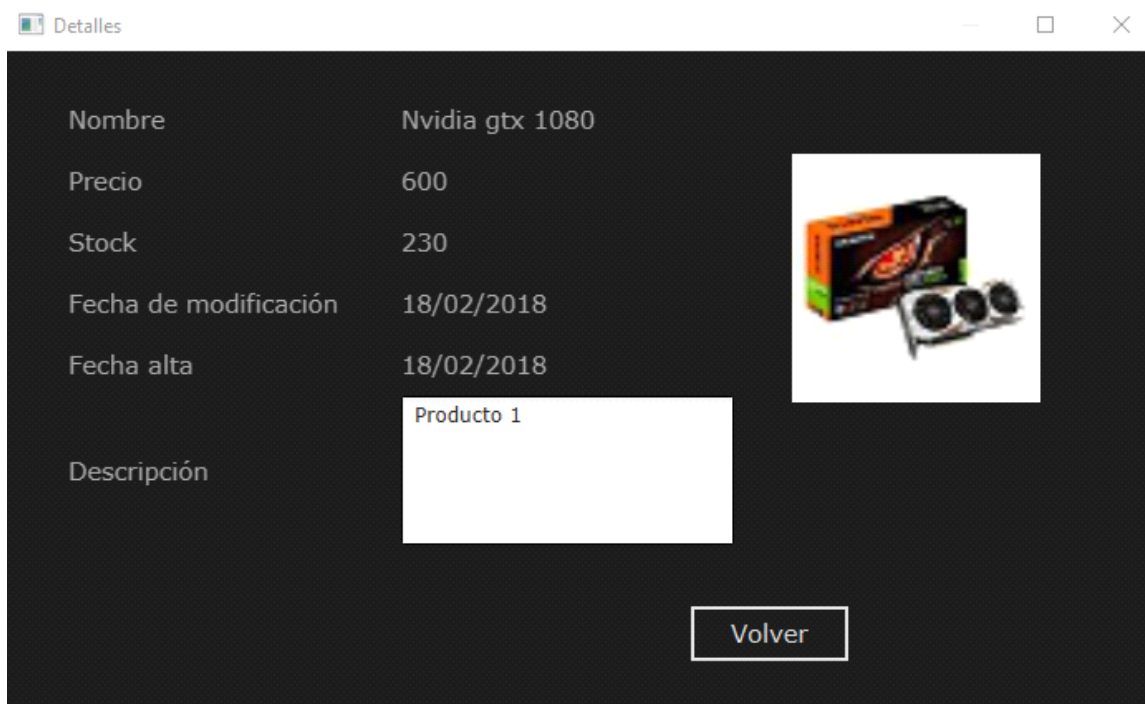
    try {
        //Contexto
        JAXBContext context = JAXBContext.newInstance(Empaquetador.class);
        Unmarshaller um = context.createUnmarshaller();

        //Leo XML del archivo y hago unmarshal
        Empaquetador empaquetador = (Empaquetador) um.unmarshal(archivo);

        //Borro los anteriores
        datosComponente.clear();
        datosComponente.addAll(empaquetador.getComponentes());

        //Guardo la ruta del archivo al registro de preferencias
        setRutaArchivoComponentes(archivo);
    } catch (Exception e) {
        //Muestro alerta
        Alert alerta = new Alert(Alert.AlertType.ERROR);
        alerta.setTitle("Error");
        alerta.setHeaderText("No se pueden cargar datos de la ruta " + archivo.getPath());
        alerta.setContentText(e.toString());
        alerta.showAndWait();
    }
}
}
```

En la imagen podemos ver una pequeña visualización del trabajo realizado por el grupo en el que muestra todos los componentes implementados en la lista. Nosotros podremos ver los detalles de cada componente.



```

    }

    @FXML
    public void initialize(){
        // nombreLabel.setText(datos.getNombre(componente));
        // precioLabel.setText(componente.getPrecio());
        // stockLabel.setText(componente.getStock());
        // gestorInventario.muestraDetalle();
    }

    public void setGestorInventario(GestorInventario gestorInventario) {
        this.gestorInventario = gestorInventario;
    }

    public void setEscenarioDetalle(Stage escenarioDetalle){
        this.escenarioDetalle = escenarioDetalle;
    }

    public void setComponente(Componente componente){
        this.componente = componente;

        foto.setImage(new Image(componente.getRuta()));
        nombreLabel.setText(componente.getNombre());
        precioLabel.setText(componente.getPrecio());
        stockLabel.setText(componente.getStock());
        fechaAltaLabel.setText(Fechas.formato(componente.getFechaAlta()));
        fechaModificacionLabel.setText(Fechas.formato(componente.getFechaMod()));
        descripcionLabel.setText(componente.getDescripcion());
    }

    @FXML
    public void borrar(){
        escenarioDetalle.close();
    }

    @FXML
    public void volver(){
        escenarioDetalle.close();
    }
}

```

También podremos añadir un producto nuevo:

Nuevo Componente

Foto

Elegir foto

Nombre

Precio

Stock

Descripción

Añadir

Cancelar

```

@FXML
private void initialize() {
}

//Es llamado por la aplicación principal para tener una referencia de vuelta de si mismo
public void setGestorInventario(GestorInventario gestorInventario) {
    this.gestorInventario = gestorInventario;
}

//Establece el escenario de edición
public void setEscenarioNuevo(Stage escenarioNuevo) {
    this.escenarioNuevo = escenarioNuevo;
}

public boolean isGuardarClickado() {
    return guardarClickado;
}

@FXML
private void anadir() {
    // System.out.println(ruta + " " + fotoTextField.getText());
    if (datosValidos()) {
        int numero = (int) (Math.random() * 10000 + 5000);
        //Asigno datos a propiedades de persona
        LocalDate now = LocalDate.now();
        //String valor = ruta();
        //System.out.println(fotoTextField.getText());
        if (archivo == null) {
            archivo = new File(componente.getRuta());
        }
        System.out.println(archivo.getAbsolutePath());
        componente.setRuta("file:" + archivo.getPath());
        componente.setFoto(new ImageView("file:" + archivo.getPath()));
        componente.setNombre(nombreTextField.getText());
        componente.setPrecio(precioTextField.getText());
        componente.setStock(stockTextField.getText());
        componente.setFechaAlta(now);
        componente.setFechaMod(now);
        componente.setCodigo(numero);
        componente.setDescripcion(descripcion.getText());

        guardarClickado = true; //Cambio valor booleano
        escenarioNuevo.close(); //Cierro el escenario de edición
    }
}

```

```

//Llamado cuando se pulsa Cancelar
@FXML
private void cancelar() {
    ventanaInventario.close();
}

//Validación de datos
private boolean datosValidos() {
    //Inicializo string para mensajes
    String mensajeError = "";

    //Compruebo los campos
    if (nombreTextField.getText() == null || nombreTextField.getText().length() == 0) {
        mensajeError += "Nombre no válido.\n";
    }
    //if (fotoTextField.getText() == null || fotoTextField.getText().length() == 0) {
    //    mensajeError += "Foto no válida.\n";
    //}
    //if (direccionTextField.getText() == null || direccionTextField.getText().length() == 0) {
    //    mensajeError += "Dirección no válida.\n";
    //}

    if (stockTextField.getText() == null || stockTextField.getText().length() == 0) {
        mensajeError += "Código postal no válido.\n";
    }

    if (mensajeError.length() == 0) {
        //System.out.println(ruta + " " + fotoTextField.getText());
        return true;
    } else {
        //Muestro alerta y devuelvo false
        Alert alerta = new Alert(Alert.AlertType.ERROR);
        alerta.setTitle("Error");
        alerta.setHeaderText("Datos no válidos");
        alerta.setContentText("Por favor, corrige los errores");
        alerta.showAndWait();
        System.out.println(mensajeError);
        return false;
    }
}

```

```

//Establece la componente a editar
public void setComponente(Componente componente) {
    this.componente = componente;

    nombreTextField.setText(componente.getNombre());
    precioTextField.setText(componente.getPrecio());
    stockTextField.setText(componente.getStock());
    descripcionTextField.setText(componente.getDescripcion());
    //codigoPostalTextField.setText(Integer.toString(persona.getCodigoPostal()));
    //ciudadTextField.setText(persona.getCiudad());
    //fechaDeNacimientoTextField.setText(UtilidadDeFechas.formato(persona.getFechaDeNacimiento()));
    //fechaDeNacimientoTextField.setPromptText("dd/mm/yyyy");
}

@FXML
public void ruta() {
    FileChooser fileChooser = new FileChooser();
    //Muestro el diálogo de guardar
    ventanaInventario.showOpenDialog(ventanaInventario);
    //System.out.println(archivo.getNombre());
    FileChooser.ExtensionFilter extFilter = new FileChooser.ExtensionFilter(
        "JPG files (*.jpg)", "*.jpg");
    fileChooser.getExtensionFilters().add(extFilter);

    byte[] contenido = null;
    try {
        contenido = Files.readAllBytes(Paths.get(archivo.getParent(), archivo.getName()));
    } catch (IOException ex) {
        System.out.println("error al leer");
    }

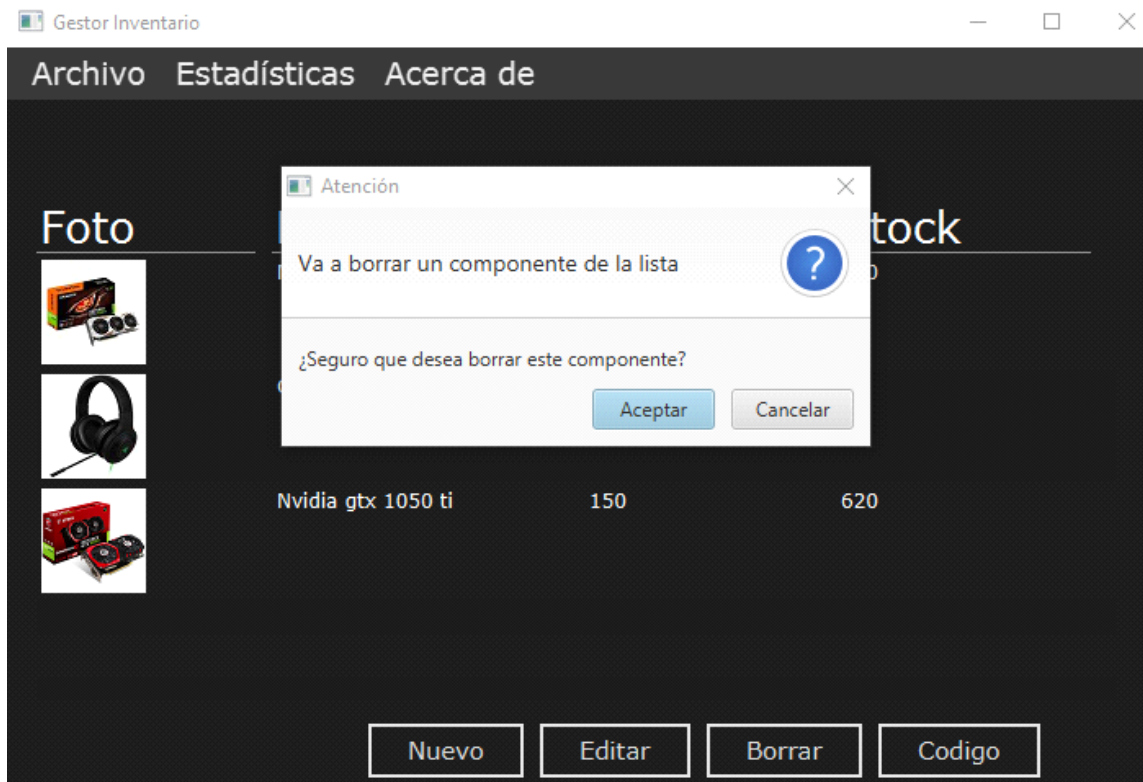
    URL location = GestorInventario.class.getResource("../img");

    File destino = new File(location.getFile());
    System.out.println(destino.getParent());
    //System.out.println(contenido);
    try {
        Files.write(Paths.get(destino.toString(), archivo.getName()), contenido);
    } catch (IOException ex) {
        System.out.println("error al escribir");
    }

    ruta = "../img/" + archivo.getName();
    //fotoTextField.setText(ruta);
    //return ruta;
}

```

Y podremos eliminar componentes de la lista:



Por último hemos creado un archivo ejecutable (.exe) donde el usuario podrá instalar la aplicación realizada en su computador

3.Estructura utilizada

La estructura que hemos realizad es el Modelo-Vista-Controlador



La estructura MVC ("Model-View-Controller") es un paradigma utilizado en diversos desarrollos de software, a través de este "Framework" se logra una división de las diferentes partes que conforman una aplicación, siendo su principal razón de ser: *manutención del código fuente*.


4. GitHub

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones.

Hemos utilizado esta plataforma para subir nuestras versiones, cada vez que uno de los miembros del proyecto realizaba un cambio o implementaba algo subía dicha versión a la cuenta de GitHub

Link del Repositorio:

<https://github.com/jorgemelero/GestorJorgeM1JorgeD2Sergio3Julian4Rodrigo5>

 [Features](#) [Business](#) [Explore](#) [Marketplace](#) [Pricing](#)

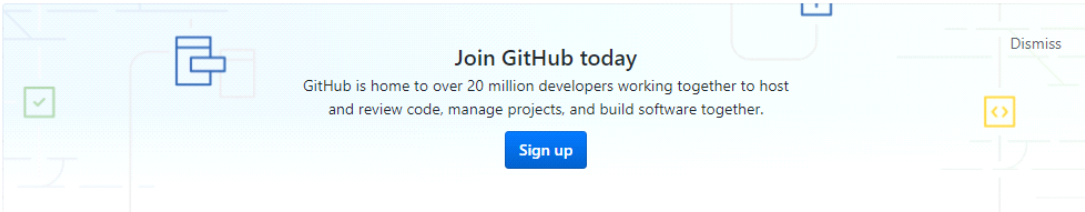
This repository

[Sign in](#) or [Sign up](#)

[jorgemelero](#) / [GestorJorgeM1JorgeD2Sergio3Julian4Rodrigo5](#)

[Watch](#) 0 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues](#) 0 [Pull requests](#) 0 [Projects](#) 0 [Insights](#)



No description, website, or topics provided.

40 commits1 branch0 releases5 contributors

Branch: masterNew pull requestFind fileClone or download


jorgedmegias

Add files via upload

Latest commit a377cc6 11 days ago

nbproject	Proyecto	26 days ago
src	Add files via upload	11 days ago
.gitignore	Proyecto	26 days ago
build.xml	Proyecto	26 days ago
manifest.mf	Proyecto	26 days ago

© 2018 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)



[Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

5. Mejoras

Después de realizar el trabajo vimos que podríamos haber implementado muchas más cosas, para que quedara un proyecto más completo, pero por problemas de tiempo no pudimos implementar todo lo que quisimos.

Por ejemplo nos hubiera gustado poner más componentes, también implementar algún método mas o por ejemplo mas botones para poder editar cada componente de la lista

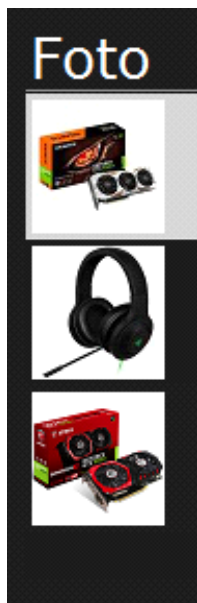
Aun así el trabajo realizado por el grupo ha sido bueno y estamos satisfechos con los resultados y el trabajo en grupo

6.Pruebas realizadas

Las pruebas realizadas fueron muchas ya que cada vez que implementábamos algún método o cambiamos algo en alguna de las vistas lo ejecutábamos.

Al principio lo que hicimos fue hacer un pequeño diseño en un papel para ver como seria la pre-visualización. Cuando nos pusimos a programar tuvimos que hacer algunos cambios para la mejora de la aplicación, como por ejemplo, la ventana detalles, y fuera mas fácil su acceso.

También hicimos distintas pruebas al meter las imágenes de los componentes, por la dimensión de las imágenes, tuvimos que realizar distintas pruebas para que tuviera las dimensiones ideales



Una de las pruebas que tuvimos que realizar en un proyecto externos fue la creación del código de barras. Al final conseguimos sacarlos con "Barcode" en la que nos genero el código en los pdfs correspondientes.

Nvidia gtx 1080



123

Nvidia gtx 1080



123

Y por ultimo en la implementación de CSS tuvimos que realizar muchas pruebas para elegir el color adecuado para una mejor visualización del proyecto, probamos varios colores para nuestro proyecto

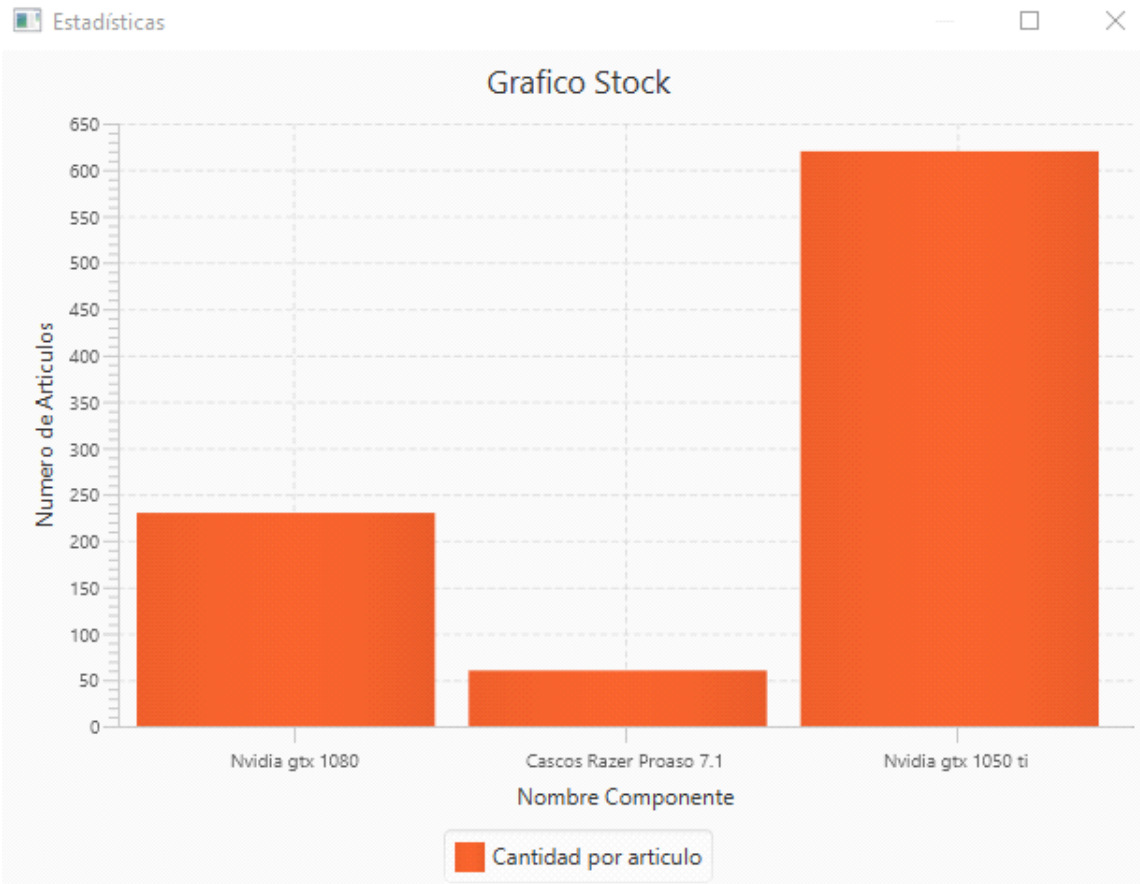
Gestor Inventario

Archivo Estadísticas Acerca de

Foto	Nombre	Precio	Stock
	Nvidia gtx 1080	600	230
	Cascos Razer Proaso 7.1	150	60
	Nvidia gtx 1050 ti	150	620

Nuevo Editar Borrar Codigo

Tambien tenemos hecho la grafica con el metodo de barras:



```
public void crearGrafico() {  
    //Cargo la vista estadísticas  
    FXMLLoader loader = new FXMLLoader();  
    URL location = GestorInventario.class.getResource("/view/VistaEstadistica.fxml");  
    loader.setLocation(location);  
    try {  
        vistaEstadisticas = loader.load();  
    } catch (IOException ex) {  
        Logger.getLogger(GestorInventario.class.getName()).log(Level.SEVERE, null, ex);  
    }  
  
    //Inicializo un nuevo escenario y asigno el principal  
    Stage escenarioEstadisticas = new Stage();  
    escenarioEstadisticas.setTitle("Estadísticas");  
    escenarioEstadisticas.initModality(Modality.WINDOW_MODAL);  
    escenarioEstadisticas.initOwner(escenarioPrincipal);  
  
    //Cargo la escena que contiene ese layout de estadísticas  
    Scene escena = new Scene(vistaEstadisticas);  
    escenarioEstadisticas.setScene(escena);  
  
    //Asigno el controlador  
    VistaEstadisticasController controller = loader.getController();  
    controller.setDatosComponente(datosComponente);  
  
    //Muestro el escenario  
    escenarioEstadisticas.show();  
}
```

7. Archivo ejecutable

Este apartado tratara de como desplegar (empaquetar publicar) nuestra aplicación, siguiendo para ello el tutorial de esta página: https://netbeans.org/kb/docs/java/native_pkg.html

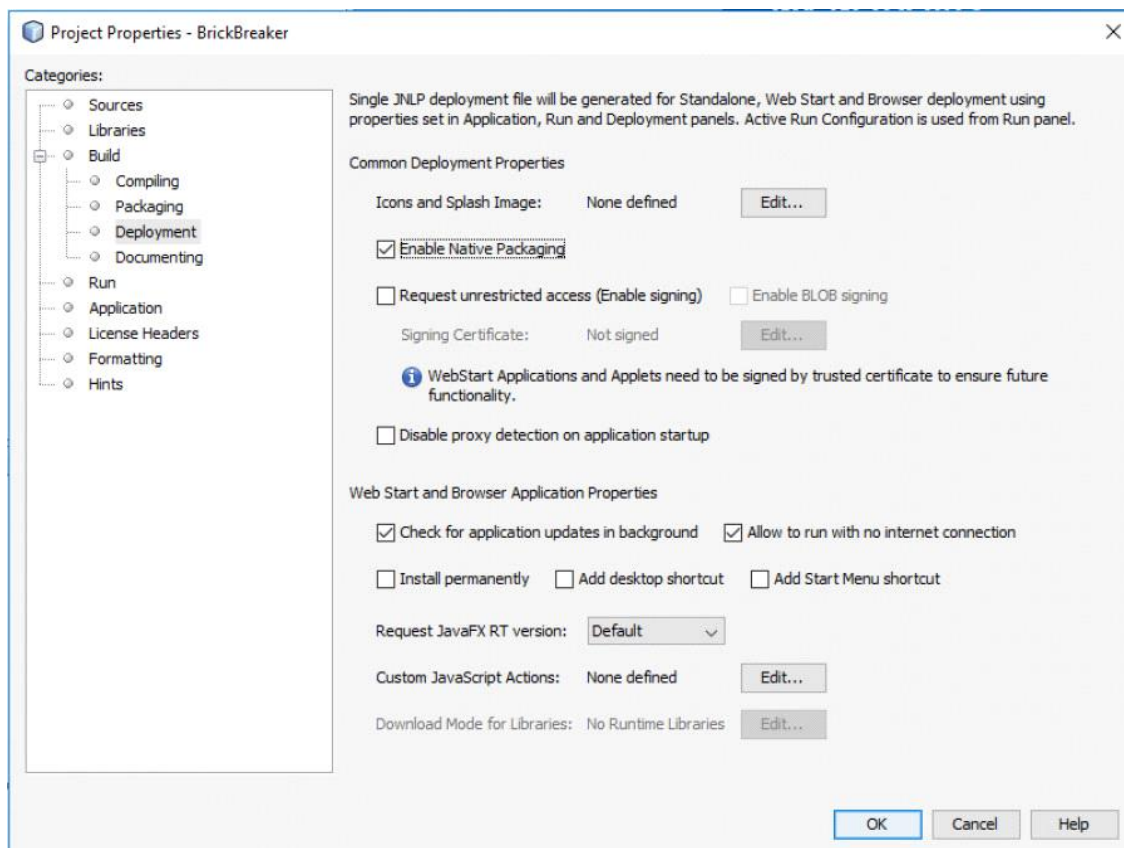
Utilizaremos el Native Packaging o empaquetamiento nativo de NetBeans para JavaFX.

Las herramientas utilizadas para poder utilizar el empaquetamiento nativo son:

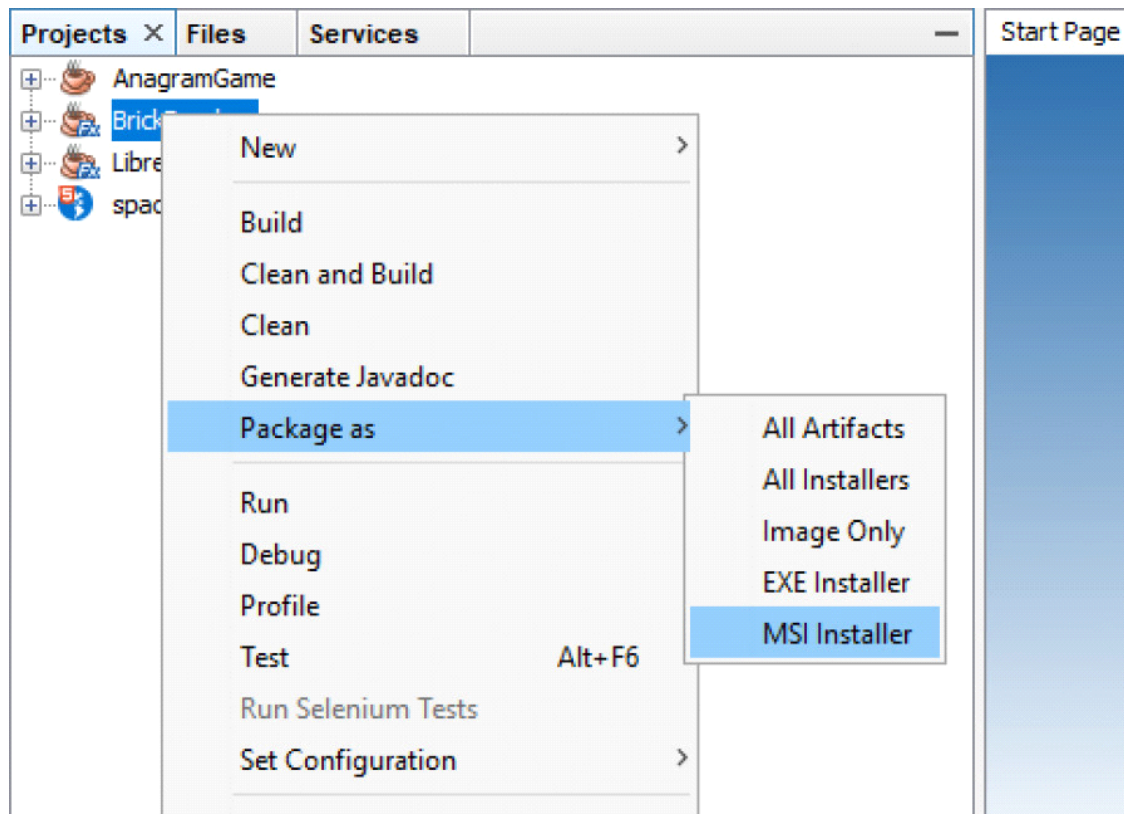
- Inno Setup Stable Release para generar archivos .exe
- Wix Toolset para generar archivos *.msi

El proceso que hemos utilizado para empaquetar nuestra aplicación sido el siguiente:

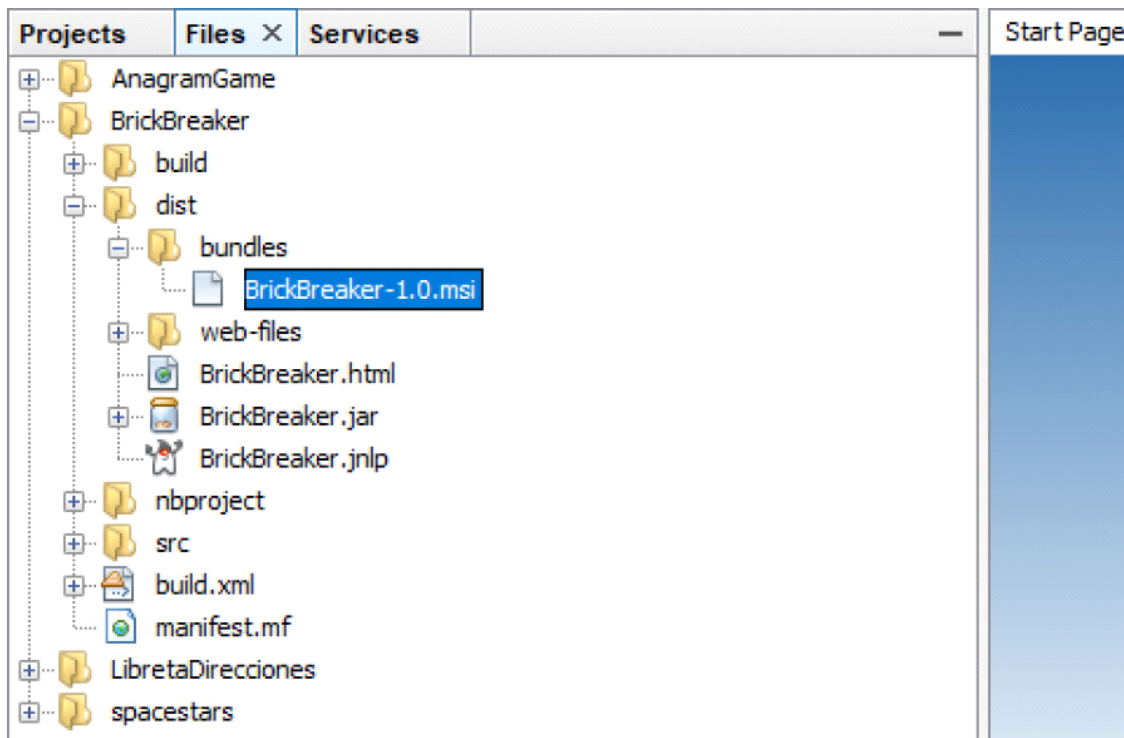
1. Habilitamos las acciones de empaquetado para el proyecto



2. Creamos un instalador MSI:

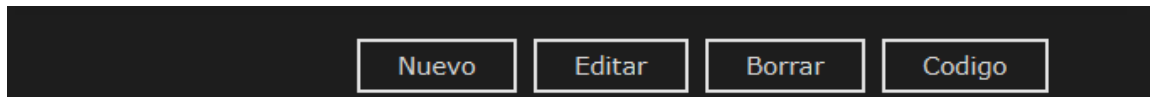


3. Y por ultimo podremos encontrar un archivo ejecutable dentro de la carpeta dist/bundles:



8.Problemas encontrados

Primer problema: Tuvimos que rehacer el diseño de la aplicación ya que no pudimos implementar los botones como queríamos desde un principio hecho en el diseño. Pero al final los pusimos con una utilidad mas simple y accesible



Segundo problema: al actualizar un valor, no se reflejaba en la lista, pero si que se realizaba el cambio en vista detalle

Precio	Stock
600	230
150	60
150	620

Tercer Problema: Al generar el xml no supimos como subir una imagen, ya que solo acepta string el xml, pero conseguimos llevar la ruta como string y conseguimos generar el xml

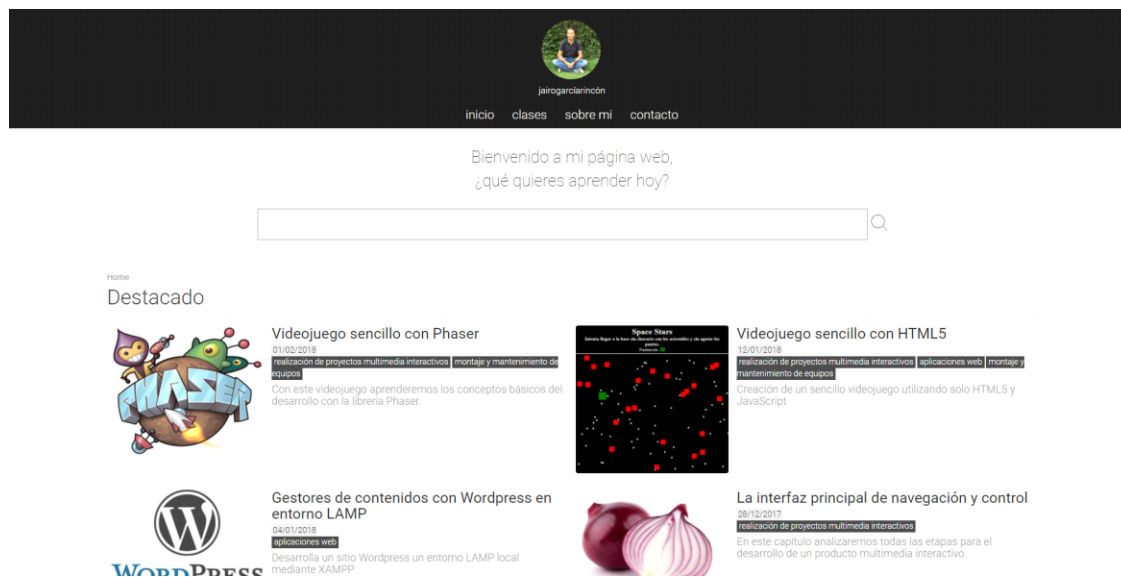
Cuarto problema: Una de las cosas que mas dio guerra fue los graficos ya que no se generaban y tuvimos que pedir ayuda a compañeros de otros grupos para poder solucionarlo

Quinto problema: No hemos tenido suficiente tiempo para pulir cosas que nos gustaria que tuviera nuestra aplicacion


9.Bibliografia

Las paginas utilizadas han sido:

<http://www.jairogarciarincon.com/>



<https://es.stackoverflow.com/>


 Preguntas Etiquetas Usuarios

[Iniciar sesión](#) [Regístrate](#)


Stack Overflow en español es un sitio de preguntas y respuestas para programadores y profesionales de la informática. Únete a ellos, toma menos de un minuto.

[Regístrate](#)


Así es como funciona:



Cualquiera puede formular una pregunta



Cualquiera puede responder



Se vota a favor de las mejores respuestas, y éstas suben a los primeros puestos

Explora nuestras preguntas

php javascript java android c# html jquery mysql python sql más etiquetas

1 pregunta con nueva actividad

0 votos 0 respuestas 15 vistas

¿Por qué no funciona con css el siguiente código?

jquery css bootstrap laravel-5 php

modificada hace 1 min Mara Rosa Cambero 43

1 voto 1 respuesta 8 vistas

¿Como crear la relacion de dos tablas en laravel lumen?

laravel

modificada hace 3 minutos Dev Joel 14.5k

0 votos 0 respuestas 2 vistas

como hacer que en un listView se mantengan los elementos seleccionados luego de cambiar el adaptador

listview

formulada hace 8 minutos Zanav Lobartsi 13

0 votos 1 respuesta 13 vistas

Error en Android desde un Login (MainActivity) a un AppCompatActivity implements NavigationView.OnNavigationItemSelectedListener, IFragments

android android-studio

modificada hace 11 minutos Matias Nicolas Núñez Rojas 6

0 votos 0 respuestas 5 vistas

Problema al escanear un caracter y guardarlo en una estructura

c

formulada hace 26 minutos Leonard 1

0 votos 0 respuestas 5 vistas

Como puedo integrar las siguientes funciones en mi calculadora científica,

c#

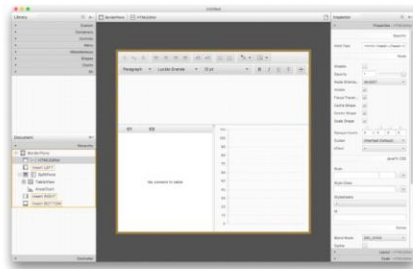
formulada hace 26 minutos Aweel Duarte 6

Preguntas populares en la red

- Are violent video games a better explanation of school shootings than access to guns?
- What is the fastest the Earth has ever spun?
- First call to the moon 1969
- How to remove duplicate lines in a CSV based on first field, and 1st n chars of 2nd field?
- What is this type of scene transition called?
- What prevents certain grammatical forms to be analysed as one word?
- What are the standard genre characteristics of contemporary women's fantasy
- How can I stop acquaintances from reposting my photos on social media without credit to me?
- Why didn't the Hogwarts faculty find the Chamber of Secrets?
- Are the terms "girlfriend" and "boyfriend" limited to some age in the spoken language?
- How do I handle strangers who want to "borrow" a cigarette and give nothing in return?
- Is it socially acceptable to directly contact renowned academics as a student?
- I am an EU resident being contacted about stock I own in a defunct company by a supposedly US based lawyer, is this a scam?
- Readability versus maintainability: special case of writing mental function calls
- Do these pictures depict the same man after two separate crises?

<http://gluonhq.com/products/scene-builder/>

Scene Builder



Drag & Drop,
Rapid Application
Development.

[Download Now](#)


Integrated

Scene Builder works with the JavaFX ecosystem – official controls, community projects, and Gluon offerings including [Gluon Mobile](#), [Gluon Desktop](#),



Simple

Drag & Drop user interface design allows for rapid iteration. Separation of design and logic files allows for team members to quickly and easily focus on



Supported

Scene Builder is free and open source, but is backed by Gluon. Commercial support offerings are available, including [training](#) and custom consultancy