

Proyecto Final Programación

Pensamiento Computacional

Abstracción

- ¿Qué se solicita finalmente? (problema)
 - sistema de gestión para la administración de un Hospital

- ¿Qué información es relevante dado el problema anterior?

gestión de historiales médicos, programación de consultas y asignación de médicos, proporcionando una interfaz segura e intuitiva.

- ¿Cómo se agrupa la información relevante?
 - Hospital: nombre, pacientes, médicos, administradores, salas, citas, historias
 - Paciente: Datos personales, Historial clínico, datos de persona
 - Medico: Horario, datos de persona
 - Administradores: nombre, apellido, id, teléfono
- ¿Qué funcionalidades se solicitan?
 - Funcionalidades Generales de Pacientes
 - Registro y actualización de datos personales.
 - Solicitud y cancelación de citas médicas.
 - Consulta de historial médico.
 - Recepción de notificaciones sobre citas programadas.
 - Funcionalidades Generales de Médicos
 - Acceso a los historiales médicos de sus pacientes.
 - Registro de diagnósticos y tratamientos.
 - Administración de horarios de consulta.
 - Notificación de cambios en las citas.
 - Administradores
 - Registro, modificación y eliminación de médicos y pacientes.
 - Gestión de salas y horarios de atención.
 - Monitoreo de disponibilidad de médicos y asignación de pacientes.
 - Generación de reportes sobre citas médicas y ocupación del hospital.
 - Funcionalidad Extendida (Opcional)
 - Manejo de archivos médicos: Adjuntar documentos como resultados de exámenes y recetas médicas.
 - Sistema de notificaciones: Envío de recordatorios sobre citas y tratamientos por correo electrónico.
 - Integración con farmacias: Conexión con farmacias para solicitar medicamentos recetados.

Descomposición

¿Cómo se distribuyen las funcionalidades?

¿Qué debo hacer para probar las funcionalidades?

Reconocimiento de patrones

¿Qué puedo reutilizar de la solución de otros problemas?

Estructura básica del programa consultorio odontológico

Codificación

¿Cómo pruebo la solución en java?

¿Cómo escribo la solución en java?

Código adjunto

Otros

Requisitos de entrega:

Mínimo 15 commits (cambios relevantes, modificaciones al código) por integrante

Máximo 3 integrantes, solo un integrante entrega.

Cada commit bien documentado.

Utilizar JavaFx para las interfaces, herencia, polimorfismo, encapsulamiento y abstracción.

Commits

Cada commit tiene su identificador.

Cada commit dice que se hizo en el código .

Investigar cosas y eso sobre GitHub .

Código guardado:

```
public void openHospitalMenuInicio(){
    try {
        FXMLLoader loader = new FXMLLoader();
        loader.setLocation(App.class.getResource("MenuInicio.fxml"));
        AnchorPane rootLayout = (AnchorPane) loader.load();
        MenuInicioViewController menuInicioViewController = loader.getController();
        menuInicioViewController.setApp(this);

        Scene scene = new Scene(rootLayout);
        primaryStage.setScene(scene);
        primaryStage.show();
    } catch (IOException e){
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

```
public void onOpenMenuInicio() {
    app.openHospitalMenuInicio();
}
```

```
loader.setLocation(App.class.getResource("co/edu/uniquindio/proyecto_ejemplo_javafx/MenuInicio.fxml"));
```

```
public void openViewMenuInicio(){
    inicializarData();
    try{
        FXMLLoader loader = new FXMLLoader();
        URL fxmUrl = getClass().getClassLoader().getResource("co/edu/uniquindio/proyecto_ejemplo_javafx/MenuInicio.fxml");
        if (fxmUrl == null) {
            throw new RuntimeException("¡Archivo FXML no encontrado! Buscando en: "
                + new File("src/main/resources/co/edu/uniquindio/proyecto_ejemplo_javafx/MenuInicio.fxml").getAbsolutePath());
        }
        loader.setLocation(fxmUrl);

        AnchorPane anchorPane = loader.load();
        MenuInicioController menuInicioController = loader.getController();
        menuInicioController.setApp(this);
        menuInicioController.setHospital(hospital);

        Scene scene = new Scene(anchorPane);
        primaryStage.setScene(scene);
        primaryStage.show();
    } catch (IOException e){
        // TODO Auto-generated catch block
        e.printStackTrace();

        Alert alert = new Alert(Alert.AlertType.ERROR);
        alert.setTitle("Error");
        alert.setHeaderText("No se pudo cargar la interfaz");
        alert.setContentText(e.getMessage());
        alert.showAndWait();
    }
}
```

```
public void openViewMenuInicio() {
    try {
        // Debug: Verifica la ruta física del archivo
        Path fxmlPath = Paths.get("src/main/resources/co/edu/uniquindio/proyecto_ejemplo_javafx/MenuInicio.fxml");
        System.out.println("Buscando FXML en: " + fxmlPath.toAbsolutePath());
        System.out.println("El archivo existe: " + Files.exists(fxmlPath));

        // Carga el FXML
        FXMLLoader loader = new FXMLLoader();
        InputStream fxmlStream =
getClass().getClassLoader().getResourceAsStream("co/edu/uniquindio/proyecto_ejemplo_javafx/MenuInicio.fxml");
        if (fxmlStream == null) {
            throw new RuntimeException("No se pudo abrir el stream del FXML");
        }
        AnchorPane root = loader.load(fxmlStream);

        // Configura el controller
        MenuInicioController controller = loader.getController();
        controller.setApp(this);

        // Muestra la escena
        primaryStage.setScene(new Scene(root));
        primaryStage.show();
    } catch (Exception e) {
        System.err.println("Error crítico al cargar FXML:");
        e.printStackTrace();

        // Muestra un diálogo de error
    }
}
```

```
Alert alert = new Alert(Alert.AlertType.ERROR);
alert.setTitle("Error Fatal");
alert.setHeaderText("No se pudo iniciar la aplicación");
alert.setContentText(e.toString());
alert.showAndWait();

Platform.exit();
}
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>co.edu.uniquindio</groupId>
  <artifactId>Hospital_con_javafx</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>21</maven.compiler.source>
    <maven.compiler.target>21</maven.compiler.target>
    <junit.version>5.10.2</junit.version>
    <javafx.version>21.0.2</javafx.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-controls</artifactId>
      <version>${javafx.version}</version>
    </dependency>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-fxml</artifactId>
      <version>${javafx.version}</version>
    </dependency>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-api</artifactId>
      <version>${junit.version}</version>
      <scope>test</scope>
    </dependency>
  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.13.0</version>
      </plugin>
      <plugin>
        <groupId>org.openjfx</groupId>
        <artifactId>javafx-maven-plugin</artifactId>
        <version>0.0.8</version>
        <configuration>
          <mainClass>co.edu.uniquindio.proyecto_ejemplo_javfx.App</mainClass>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>co.edu.uniquindio</groupId>
  <artifactId>Hospital_con_javafx</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>proyecto_ejemplo_javafx</name>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <junit.version>5.10.2</junit.version> </properties>

  <dependencies>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-controls</artifactId>
      <version>17.0.6</version>
    </dependency>
    <dependency>
      <groupId>org.openjfx</groupId>
      <artifactId>javafx-fxml</artifactId>
      <version>17.0.6</version>
    </dependency>

    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-api</artifactId>
      <version>${junit.version}</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>${junit.version}</version>
      <scope>test</scope>
    </dependency> </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.13.0</version>
        <configuration>
          <source>23</source>
          <target>23</target>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.openjfx</groupId>
        <artifactId>javafx-maven-plugin</artifactId>
        <version>0.0.8</version>
        <executions>
          <execution>
            <!-- Default configuration for running with: mvn clean javafx:run -->
            <id>default-cli</id>
            <configuration>
              <mainClass>co.edu.uniquindio.proyecto_ejemplo_javafx/co.edu.uniquindio.proyecto_ejemplo_javafx.HelloApplication</mainClass>
              <launcher>app</launcher>
              <jlinkZipName>app</jlinkZipName>
              <jlinkImageName>app</jlinkImageName>
              <noManPages>true</noManPages>
              <stripDebug>true</stripDebug>
              <noHeaderFiles>true</noHeaderFiles>
            </configuration>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</project>

```

```
</plugin>  
</plugins>  
</build>  
</project>
```