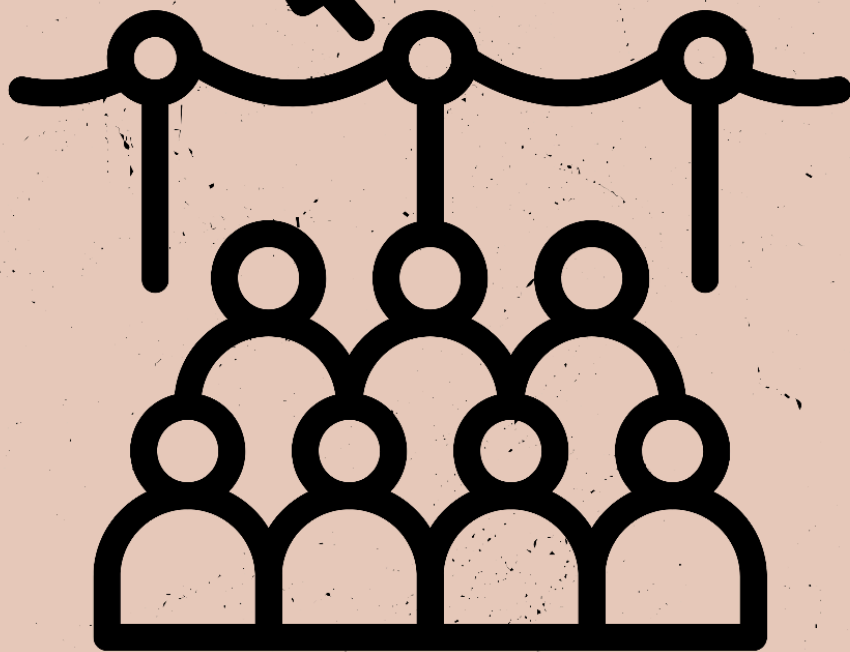


# Hyrule's Events

## Manual Técnico



Loading bugs



# MANUAL TÉCNICO

# HYRULE'S EVENTS

<b>Introducción</b>	<b>1</b>
<b>Presentación</b>	<b>2</b>
¿Qué incluye este manual?	2
¿Por qué es importante este manual?	2
<b>Resumen</b>	<b>2</b>
<b>Objetivo</b>	<b>2</b>
<b>Aspectos Técnicos</b>	<b>3</b>
- <b>Herramientas Utilizadas Para el Desarrollo del Programa:</b>	<b>3</b>
· Java	3
· Command Prompt	3
· NetBeans	3
· Sistema Operativo	3
- <b>Requisitos Mínimos del Sistema</b>	<b>3</b>
<b>Documentación</b>	<b>4</b>
- <b>Documentación Interna</b>	<b>4</b>
· Comentarios en Código	4
* Comentarios de clases:	4
* Comentarios de métodos:	4
* Algoritmos de métodos:	5
* Comentarios de variables:	5
* Estructura de una clase de backend:	5
- <b>Documentación Externa</b>	<b>6</b>
· Descripción de clases:	6
- Descripción de todas las clases:	7
· Diagrama de Clases:	8
· Repositorio en GitHub:	9
· Documentación en JavaDoc:	10
<b>Problemas del Programa</b>	<b>12</b>

## **Introducción**

Este manual técnico proporciona una guía detallada para comprender la lógica detrás del programa desarrollado por Triforce Software'n, utilizando la interfaz gráfica de java swing y el lenguaje de programación Java, el controlador de versiones Git, github y la base de datos relacional MySQL. Hyrule's Events ofrece a los usuarios la posibilidad de gestionar y administrar eventos del reino de Hyrule.

Este manual técnico proporciona una descripción exhaustiva de las características del programa, su funcionamiento interno y los requisitos de instalación y configuración. Además, se incluyen detalles sobre la responsabilidad del desarrollador y la compañía desarrolladora, Hyrule's Events, en cuanto a posibles riesgos asociados con el juego durante sesiones prolongadas o usos inadecuados, cambios en el código o el implemento de mejoras o cambio de sus funciones originales (mantenimiento).

Al momento de modificar un carácter o algo diferente a lo solicitado el juego o programa podría dejar de funcionar debidamente, por favor tómelo en cuenta al momento de realizar un cambio o implementar una mejora, por ello se adjunta toda la documentación de análisis y diseño necesaria para comprender el programa.

## **Presentación**

Bienvenidos desarrolladores al manual técnico de Hyrule's Events, el juego más complejo y entretenido que ha desarrollado Triforce Software, en este manual podrán encontrar lo necesario para poder entender la lógica y el proceso para poder realizar ajustes, cambios o posibles mejoras al programa.

### **¿Qué incluye este manual?**

En este manual, encontrará una descripción detallada de las características del desarrollo de Hyrule's Events, su funcionamiento interno y los requisitos necesarios para implementar o cambiar algo de Hyrule's Events desde tu sistema.

- Aspectos destacados:

- Herramientas utilizadas para el desarrollo del programa
- Desarrollo del programa
- Documentación Interna
- Documentación externa

### **¿Por qué es importante este manual?**

Este manual técnico es esencial para comprender completamente el funcionamiento de Hyrule's Events, así como para garantizar una experiencia de desarrollo sencilla, clara y concisa.

## **Resumen**

Este manual técnico facilitará la implementación de una mejora o el cambio de una funcionalidad para cualquier desarrollador, es decir el mantenimiento del programa.

## **Objetivo**

El objetivo de este manual es que como desarrolladores nos apoyemos facilitandonos el entendimiento y la comprensión de este programa, este programa nació con la finalidad de ser algo entretenido para nosotros, por ello se implementaron buenas prácticas de programación, por ello mismo se espera que este manual sea fácil de entender e interpretar por medio de las herramientas utilizadas.

## Aspectos Técnicos

### - Herramientas Utilizadas Para el Desarrollo del Programa:

#### · Java

Hyrule's Events es un juego desarrollado y creado exclusivamente con java. La versión usada de Java para desarrollar el juego Hyrule's Events es la versión “21.0.5” y la versión de JSE es la misma “21.0.5” y esta misma corresponde a la versión del JDK descargadas desde los sitios oficiales de java, con las variables de entorno configuradas.

#### · *Command Prompt*

Para observar los cambios realizados y las pruebas realizadas fue necesario ejecutar el archivo ejecutable .jar desde la terminal de windows 10.

#### · NetBeans

NetBeans fue el IDE en el cual se realizó el programa, donde se extrajo el archivo .jar, en el proceso de la creación del proyecto se utilizó “Java Maven” y “Java Application”, la versión de NetBeans es la versión “Apache NetBeans IDE 22” y todas las dependencias utilizadas por este yacen descritas en el pom.xml

#### · MySQL

El programa utiliza la base de datos relacional con la versión “9.1.0 for Win64 on x86\_64 (MySQL Community Server - GPL)”

#### · Sistema Operativo

El programa Java y Netbeans fue utilizado en la versión 10 de Windows, con una arquitectura de 64 bits de un procesador intel.

### - Requisitos Mínimos del Sistema

El programa está diseñado para que cualquier equipo que cuente con una consola (terminal “cmd”) pueda utilizar el programa, debido a que no es un programa que exige demasiados recursos, así que no hay requisitos mínimos (del hardware) propios del sistema para poder ejecutar el programa como tal, pero el programa podría llegarse a ver afectado por: La terminal del usuario (código cerrado o abierto con restricciones), una mala instalación de la JVM o de

MySQL, o una exageración (datos muy grandes) en los datos ingresados por el usuario en las entradas que requiere el programa para funcionar.

## Documentación

Como motivo de facilitar el trabajo en equipo y realizar programación abierta para todo el público se implementaron distintos métodos de documentación, estas las clasificamos en:

### - Documentación Interna

Definiendo la documentación interna como la herramienta en programación que está destinada a ser utilizada por los desarrolladores y miembros del equipo de desarrollo de software durante el proceso de creación de un programa o sistema. Esto porque esta documentación no está dirigida al usuario final, sino que está diseñada para facilitar el entendimiento y la colaboración entre los miembros del equipo de desarrollo.

#### · Comentarios en Código

Para explicar el código y tener buenas prácticas de programación, en cada clase se comentaron y enunciaron los métodos utilizados. Esto por si en algún momento se desee realizar un cambio en específico. El formato de la explicación de cada método es utilizando javadoc, donde se explica lo que realiza el método, qué necesita y que retorna en dado caso lo haga.

\* Comentarios de clases:

Se describe de forma breve la razón de existir de la clase, así como su autor, la versión de java que se utilizó para hacerla o las clases que pueden estar relacionadas a está y la fecha en la cual se creó (En las del frontend casi no están las fechas).

```
/**
 * Clase DBConnection es la clase encargada de realizar la conexión con MySQL en
 * java (junto con las dependencias en el POM) y crear las tablas que
 * representarán la base de datos la cual se utilizará para almacenar
 * información.
 *
 * @author YmCris
 * @since Aug 12, 2025
 */
public class DBConnection {
```

### \* Comentarios de métodos:

Se describe lo más detalladamente posible la acción que tiene el método, que representan los parámetros y que retorna (si aplica).

```
/**
 * Método encargado re realizar la conexión en la base de datos con java
 * através de JDBC con el driver manager, así inicializando connection que
 * será de utilidad para todo el programa.
 */
public void connect() {
    System.out.println("URL de conexión: " + URL);
    try {
        connection = DriverManager.getConnection(url:URL, user: USER_NAME, password: PASSWORD);
        System.out.println("Esquema: " + connection.getSchema());
        System.out.println("Catalogo: " + connection.getCatalog());
        query = new DBQuery(connection);
        insert = new DBInsert(connection, query);
        update = new DBUpdate(connection, query);
    } catch (SQLException e) {
        System.out.println("Ha ocurrido un error del tipo " + e.getClass().getName() + " al realizar la conexión");
        System.out.println("SQLState: " + e.getSQLState() + " Error code: " + e.getErrorCode() + "\n");
        e.printStackTrace();
    }
}
```

### \* Comentarios de variables:

En el comentario de variables no se utiliza el formato de javadoc porque representa un crecimiento del tamaño de una clase de manera exponencial, por lo tanto solo se hace referencia el tipo de variables que son, ya sean de referencia o primitivas.

```
// VARIABLES DE REFERENCIA -----
private DBQuery query;
private DBInsert insert;
private DBUpdate update;
private Connection connection;

// CONSTANTES -----
private static final String IP = "localhost";
private static final int PUERTO = 3306;
private static final String SCHEMA = "hyrule";
private static final String USER_NAME = "pruebas";
private static final String PASSWORD = "12345";
public static final String URL = "jdbc:mysql://" + IP + ":" + PUERTO + "/" + SCH
public static final String URL_FATAL = "jdbc:mysql://" + IP + ":" + PUERTO + "/"

// MÉTODOS CONCRETOS -----
```

### \* Estructura de una clase de backend:

En la creación de una clase de frontend se divide el contenido de la clase en base a la siguiente “Plantilla”, separando así todos los elementos más importantes de la clase, para que sea más fácil

de entender a todos los programadores, aunque claro que la estructura de esta planilla se ve modificada en muchas clases con métodos sobrescritos, funciones, etc.

```

/**
 * Clase HolaMundo
 *
 * @author YmCris
 * @since Mar 31, 2025
 */
public class HolaMundo {

    // VARIABLES DE REFERENCIA -----

    // VARIABLES PRIMITIVAS -----

    // INSTANCIAS -----

    // MÉTODO CONSTRUCTOR -----

    // MÉTODOS -----

    // GETTERS & SETTERS -----

}

```

## - Documentación Externa

Definiendo la documentación externa como la herramienta en programación que transmite información importante destinada a ser utilizada por usuarios finales, clientes, administradores de sistemas u otras partes interesadas que no forman parte del equipo de desarrollo de software. Esta documentación está diseñada para proporcionar orientación, instrucciones y recursos para el uso adecuado y efectivo del software desarrollado.

### · Descripción de clases:

Definiendo una **clase** como una planilla la cual sirve como molde para la creación de instancias o para realizar acciones (Basado en la SRP “Single Responsibility Principle”), comunicándose entre otras clases a través de acciones (Métodos), sin embargo ahora se implementó el concepto de Modelo, Vista y control, esto para tener mejor estructurado todo el código y facilitar el desarrollo y su mantenimiento, por lo que se describe de esa manera.



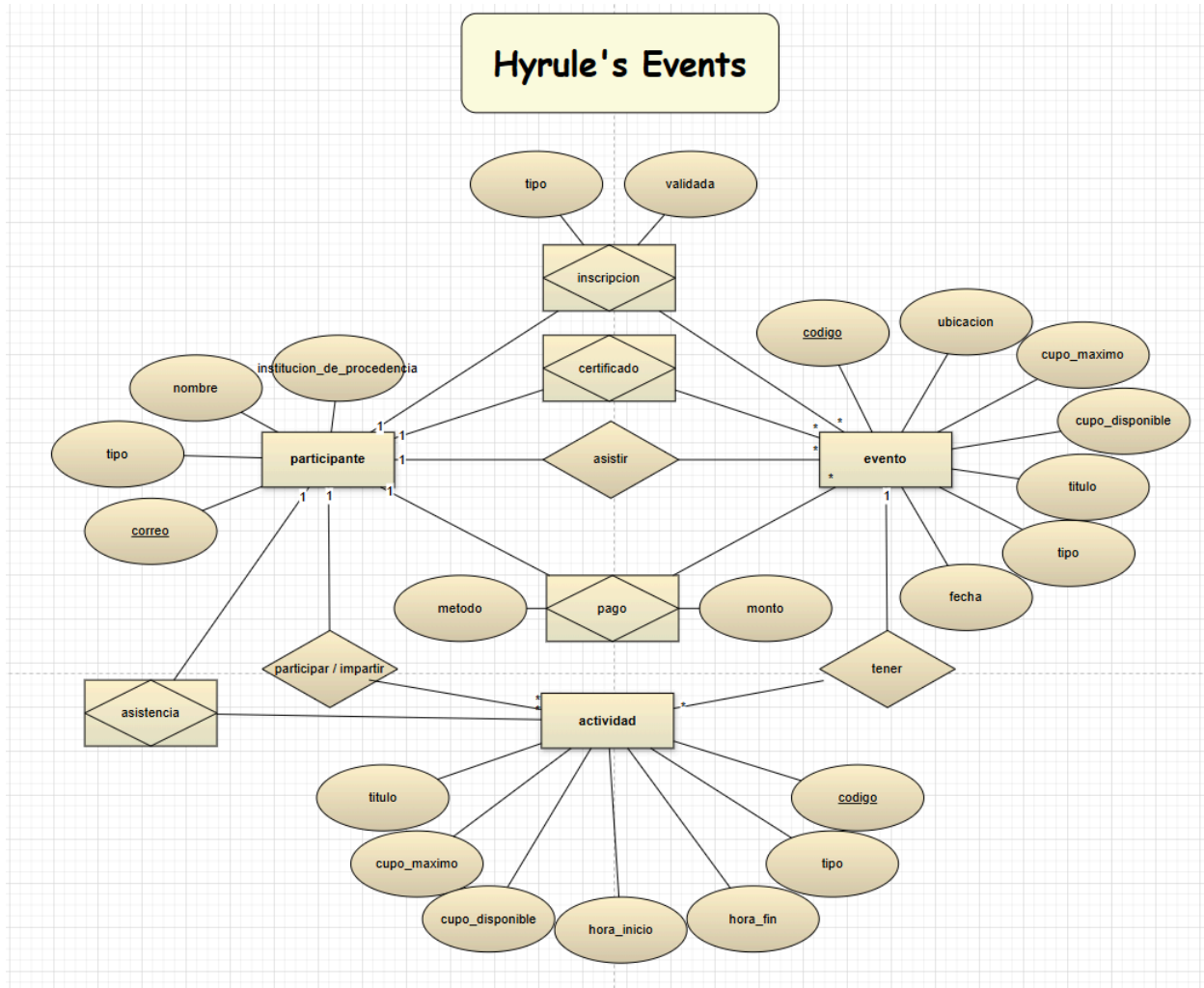
- Descripción de las clases más importantes:

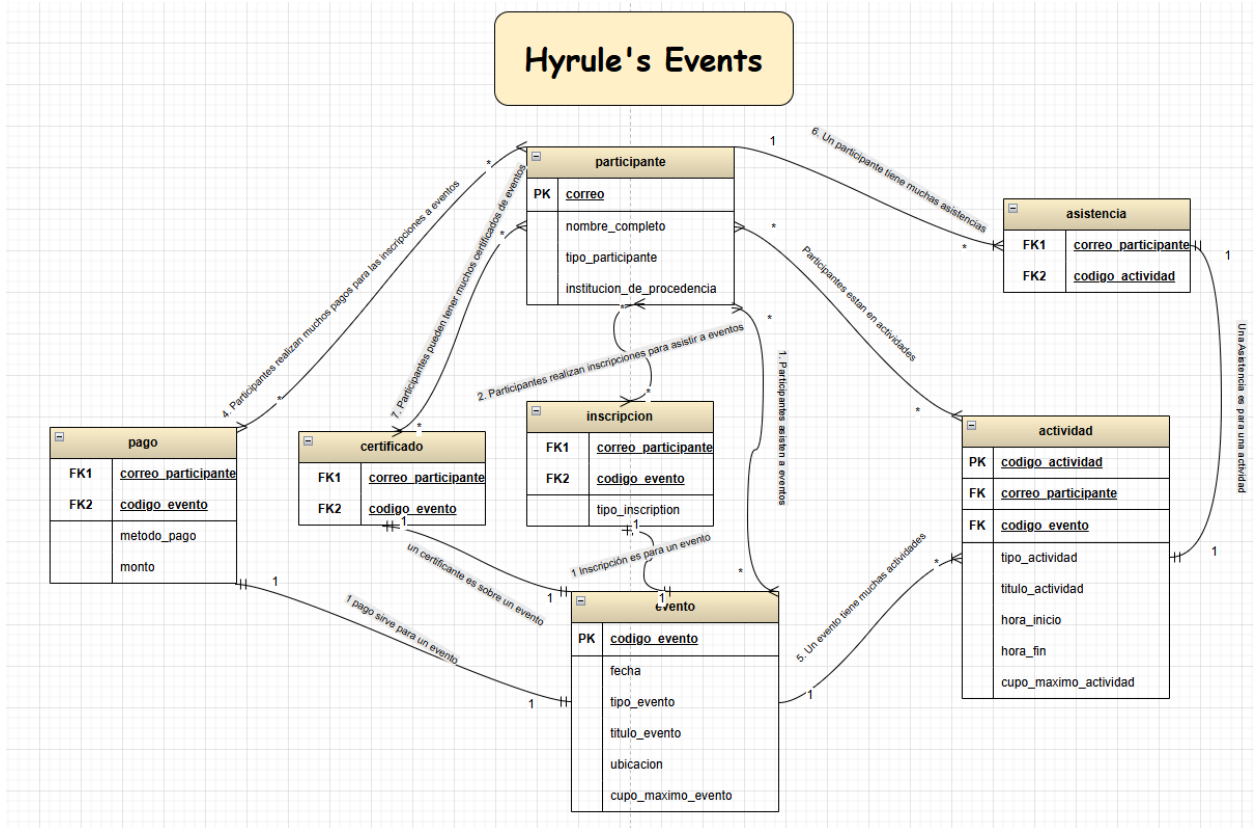
1. **JFMenuPrincipal:** JFMenuPrincipal es el frame encargado de mostrar el menú principal
2. **PanelPersonalizado:** Clase PanelPersonalizado es la clase encargada de crear el panel personalizado (ponerle una imagen a un panel)
3. **DBConnection:** Clase DBConnection es la clase encargada de realizar la conexión con MySQL en java (junto con las dependencias en el POM) y crear las tablas que representarán la base de datos la cual se utilizará para almacenar información.
4. **DBInsert:** Clase DBInsert es la clase encargada de introducir información a la base de datos, es decir que será usada por el backend del archivo y formulario, sin embargo aquí ya no deberán llegar datos erróneos.
5. **DBQuery:** Clase DBQuery es la clase encargada de realizar consultas para obtener información de la base de datos.
6. **DBUpdate:** Clase DBUpdate es la clase encargada de comunicarse con la base de datos para actualizar información ya existente en la DB.
7. **DBInsertion:** Clase DBInsertion es la clase encargada de recibir información de la clase validación formulario para poder comunicarse luego con la base de datos.

· Diagrama de Clases, Diagrama de Entidad - Relación, Diagrama de Tablas, Mapeo Físico :

Para representar gráficamente el programa se emplea el diagrama de UML diseñado en flowChart.

**Fase de Análisis:**





```

CREATE TABLE participante(
  correo VARCHAR(45) NOT NULL,
  nombre_completo VARCHAR(45) NOT NULL,
  tipo_participante VARCHAR(15) NOT NULL,
  institucion_de_procedencia VARCHAR(149) NOT NULL,
  CONSTRAINT pk_correo PRIMARY KEY (correo)
);

CREATE TABLE evento(
  codigo_evento VARCHAR(7) NOT NULL,
  fecha DATE NOT NULL,
  tipo_evento VARCHAR(10) NOT NULL,
  titulo_evento VARCHAR(45) NOT NULL,
  ubicacion VARCHAR(149) NOT NULL,
  cupo_maximo_evento INT NOT NULL,
  CONSTRAINT pk_codigo_evento PRIMARY KEY (codigo_evento)
);

CREATE TABLE actividad(
  codigo_actividad VARCHAR(7) NOT NULL,
  correo_participante VARCHAR(45) NOT NULL,
  codigo_evento VARCHAR(7) NOT NULL,
  tipo_actividad VARCHAR(8) NOT NULL,
  titulo_actividad VARCHAR(200) NOT NULL,
  hora_inicio TIME NOT NULL,
  hora_fin TIME NOT NULL,
  cupo_maximo_actividad INT NOT NULL,
  CONSTRAINT pk_codigo_actividad PRIMARY KEY (codigo_actividad),
  CONSTRAINT fk_correo_participante FOREIGN KEY (correo_participante) REFERENCES
participante(correo),
  CONSTRAINT fk_codigo_evento FOREIGN KEY (codigo_evento) REFERENCES
evento(codigo_evento)
);

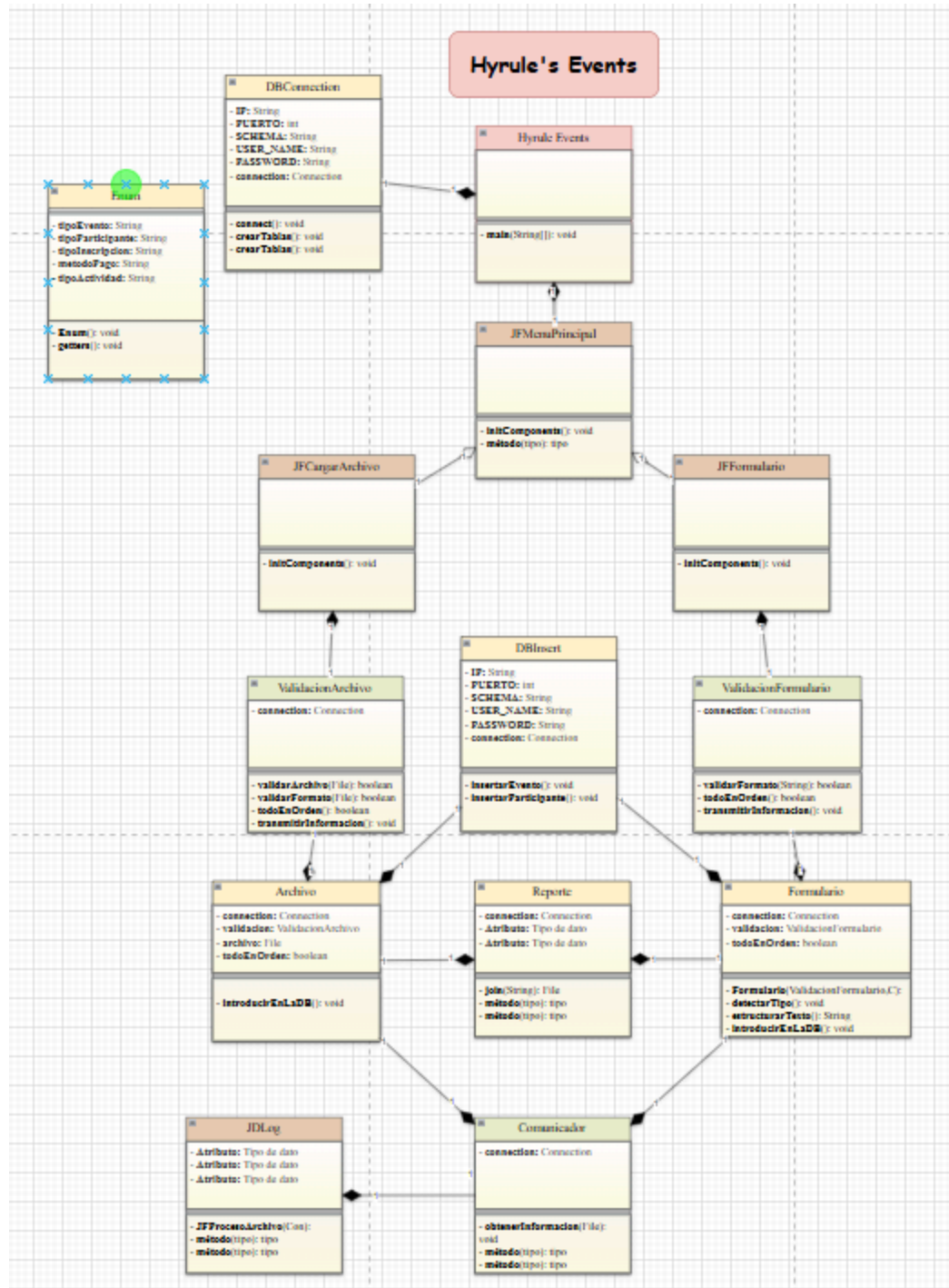
CREATE TABLE pago(
  correo_participante VARCHAR(45) NOT NULL,
  codigo_evento VARCHAR(7) NOT NULL,
  metodo_pago VARCHAR(15) NOT NULL,
  monto INT NOT NULL,
  CONSTRAINT fk_correo_participante FOREIGN KEY (correo_participante) REFERENCES
participante(correo_participante),
  CONSTRAINT fk_codigo_evento FOREIGN KEY (codigo_evento) REFERENCES
evento(codigo_evento)
);

CREATE TABLE certificado(
  correo_participante VARCHAR(45) NOT NULL,
  codigo_evento VARCHAR(7) NOT NULL,
  CONSTRAINT fk_correo_participante FOREIGN KEY (correo_participante) REFERENCES
participante(correo_participante),
  CONSTRAINT fk_codigo_evento FOREIGN KEY (codigo_evento) REFERENCES
evento(codigo_evento)
);

CREATE TABLE inscripcion(
  correo_participante VARCHAR(45) NOT NULL,
  codigo_evento VARCHAR(7) NOT NULL,
  tipo_inscripcion VARCHAR(15) NOT NULL,
  CONSTRAINT fk_correo_participante FOREIGN KEY (correo_participante) REFERENCES
participante(correo_participante),
  CONSTRAINT fk_codigo_evento FOREIGN KEY (codigo_evento) REFERENCES
evento(codigo_evento)
);

CREATE TABLE asistencia(
  correo_participante VARCHAR(45) NOT NULL,
  codigo_actividad VARCHAR(7) NOT NULL,
  CONSTRAINT fk_correo_participante FOREIGN KEY (correo_participante) REFERENCES
participante(correo_participante),
  CONSTRAINT fk_codigo_actividad FOREIGN KEY (codigo_actividad) REFERENCES
actividad(codigo_actividad)
);


```




· Fase de diseño:


Fase de Diseño::


Para una mejor descripción de los avances en el transcurso de la creación del programa se describen detalladamente los cambios en los commits del repositorio.

 IPC-2-1-PRACTICA-1 Private

[Watch](#) 0


 main


 7 Branches





 0 Tags


[Add file](#)


[Code](#)

 YmCris Merge pull request #9 from YmCris/DBInsertion new

6dd9b97 · 14 minutes ago  22 Commits

 HyruleEvents	Terminando el guardado de información en la base de datos...	15 minutes ago
 .gitignore	HyruleEvents	last week
 Mapeo-fisico.txt	Estableciendo la conexión en la DB y creación de las tablas d...	last week
 desktop.ini	Implementando todo el frontend	5 days ago

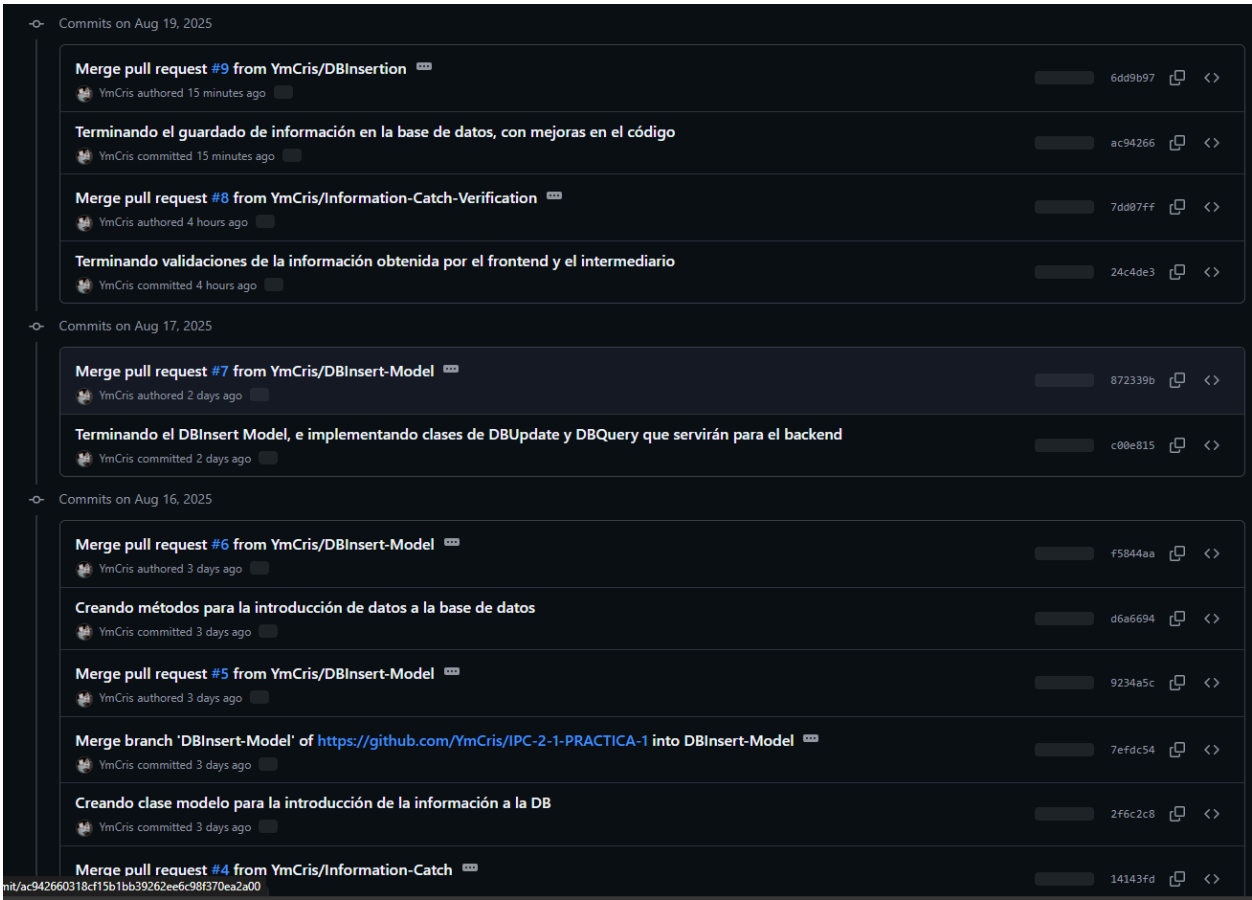
 README



## Add a README

Add a README with an overview of your project.

[Add a README](#)



· Documentación en JavaDoc:

Para representar los comentarios en código se incluye dentro de la carpeta target el index del Overview.

Overview (Hyrule's Events 1.0-SNAPSHOT API)

HyruleEvents 1.0-SNAPSHOT API

Packages	
Package	Description
ymcris.ipc2.practica1.hyruleevents	
ymcris.ipc2.practica1.hyruleevents.backend.archivos	
ymcris.ipc2.practica1.hyruleevents.backend.db	
ymcris.ipc2.practica1.hyruleevents.backend.formularios	
ymcris.ipc2.practica1.hyruleevents.frontend	
ymcris.ipc2.practica1.hyruleevents.frontend.archivos	
ymcris.ipc2.practica1.hyruleevents.frontend.formularios	
ymcris.ipc2.practica1.hyruleevents.frontend.log	
ymcris.ipc2.practica1.hyruleevents.intermediary	

## **Problemas del Programa**

En Triforce Software, entendemos que la satisfacción del cliente es nuestra prioridad número uno. Por eso, nos comprometemos a ofrecer un servicio de calidad que garantice una experiencia sin problemas al utilizar nuestros productos. En caso de que encuentres algún inconveniente durante la descarga, instalación o ejecución de nuestro programa, estamos aquí para ayudarte. Nos comprometemos a brindar asistencia técnica y, si es necesario, realizar devoluciones dentro de un plazo de 12 horas después de tu compra.

Es importante destacar que la mayoría de los problemas que puedas encontrar suelen ser el resultado de una ejecución incorrecta, pero no te preocupes, estamos aquí para guiarte en cada paso del proceso. Simplemente reiniciando el programa o volviendo a instalarlo, muchos de estos problemas se pueden resolver fácilmente.

Sin embargo, debemos señalar que cualquier modificación no autorizada al programa o descarga incorrecta podría limitar nuestra capacidad para abordar el problema. En tales casos, nos reservamos el derecho de no proporcionar asistencia.

Nuestro equipo de soporte técnico está siempre disponible para ayudarte a resolver cualquier inconveniente que puedas encontrar. Queremos que tengas la mejor experiencia posible con nuestros productos y haremos todo lo posible para asegurarnos de que así sea.