



Tecnológico de Monterrey

Ciclo final. Prototipo funcional

Emiliano Vivas Rodríguez . A01424732

Axel Daniel Corona Ibarra . A01425010

Axel Espinoza Sebastián . A01425004

Daniel De Luna Díaz . A01423940

24 de abril del 2023

Construcción de software y toma de decisiones.

TC2005B.400

Ing. Sergio Hernández Alamilla .

Índice

| | |
|---|-----------|
| Índice..... | 1 |
| Product Backlog. Historias de usuario de Aulify Black Hole..... | 2 |
| Acceder a la página web..... | 2 |
| Iniciar sesión en Aulify Black Hole..... | 3 |
| Seleccionar alguna opción del menú principal..... | 4 |
| Jugar el videojuego de forma individual o con ayuda asistida..... | 5 |
| Explorar libremente el mapa del videojuego..... | 6 |
| Resolver dinámicas y retos del videojuego..... | 7 |
| Merecer insignias y logros a través de partidas jugadas..... | 8 |
| Acceder al historial de logros..... | 9 |
| Cerrar sesión de usuario..... | 10 |
| Requerimientos no funcionales..... | 11 |
| Diagrama de casos de uso..... | 12 |
| Modelo ERD y relacional de la Base de Datos de Aulify BlackHole..... | 13 |
| Código SQL para implementación y carga de información..... | 14 |
| Comprobación de carga de datos..... | 26 |
| Comandos requeridos para realizar la manipulación de la información..... | 28 |
| Vistas de la Arquitectura 4+1 de Krutchen..... | 30 |
| Escenario. Diagrama de casos de uso..... | 31 |
| Lógica. Diagrama de clases..... | 32 |
| Despliegue. Diagrama de componentes..... | 33 |
| De procesos. Diagrama de actividades..... | 34 |
| Física. Diagrama de despliegue..... | 36 |
| Pruebas dinámicas, caja blanca..... | 37 |
| Prueba de Cobertura de Decisión 1..... | 37 |
| Prueba de Cobertura de Decisión 2..... | 38 |
| Prueba de Cobertura de Decisión 3..... | 38 |
| Prueba de Cobertura de Decisión 4..... | 39 |
| Prueba de Cobertura de Decisión 5..... | 39 |
| Pruebas dinámicas, caja negra..... | 40 |
| KPI. Logros obtenidos durante las partidas jugadas..... | 44 |
| KPI. Número de partidas jugadas..... | 46 |
| Referencias de información..... | 48 |

Product Backlog. Historias de usuario de Aulify Black Hole.

Acceder a la página web.

Como internauta

Puedo acceder a la página web del videojuego en un navegador

Para conocer sobre el y jugarlo.

Prioridad: Alta.

Criterios de aceptación.

Dado que se está en el sitio oficial de Aulify

Cuando se despliegue el contenido textual e imágenes

Entonces se podrá jugar el videojuego desde el navegador.

Detalles.

· La experiencia de usuario deberá ser minimalista, pero satisfactoria.

· Los diseños y experiencia visual del software deben contar con los diseños de Aulify.

- La página web de Black Hole será una creación tipo onepage en donde se exponga la información fundamental de introducción a cualquier usuario.

Iniciar sesión en Aulify Black Hole.

Como estudiante de la plataforma Aulify

Puedo iniciar sesión dentro del videojuego

Para conseguir jugarlo, aprender y divertirme.

Prioridad: Alta.

Criterios de aceptación.

Dado que estoy dentro del videojuego

Cuando se presente la pantalla de inicio de sesión con los campos de correo electrónico y contraseña y se autentique con éxito

Detalles.

- El usuario será autenticado en el sistema cuando su cuenta pertenezca a alguna de las cuentas de usuario pertenecientes a Aulify.
- La autenticación de usuarios debería conllevar el correo electrónico y contraseña de la cuenta de usuario.
- Al terminar de utilizar el sistema, el usuario debe cerrar su sesión.

Seleccionar alguna opción del menú principal.

Como usuario registrado a Aulify

Puedo iniciar sesión en el videojuego y acceder al menú principal

Para seleccionar alguna opción que se ofrece en el mismo.

Prioridad: Media.

Criterios de aceptación.

Dado que se accede al videojuego y se ha iniciado sesión previamente

Cuando culmine el proceso de autenticación

Entonces se desplegará el menú principal del videojuego.

Detalles.

· El software debe presentar el menú del videojuego en la pantalla principal.

· El videojuego debería mostrar las opciones: visualizar estadísticas del usuario, jugar una partida y cerrar sesión.

Jugar el videojuego de forma individual o con ayuda asistida.

Como estudiante de la plataforma Aulify

Puedo jugar el videojuego Black Hole con un solo jugador o dos jugadores

Para experimentar la experiencia de juego y ayuda asistida con las dinámicas incluidas.

Prioridad: Alta.

Criterios de aceptación.

Dado que se deseé jugar de forma individual o con ayuda asistida de una persona

Cuando se presente la opción de juego

Entonces será posible elegir uno de los modos de juego, según corresponda.

Detalles.

- La interfaz de usuario del sistema debe presentar la vista del jugador principal en todo momento.
- La interfaz designada para el asistente del jugador deberá desplegarse cuando se le indique al videojuego.
- En caso de no contar con algún asistente para el jugador, tiene que ser posible desplegar la ayuda asistida simulada.

- Si el jugador principal resuelve los cuatro retos presentes en las cuatro estaciones del mapa del juego antes de que el cronómetro de fin de partida llegue a su fin, el jugador ganará la partida. Por otro lado, el jugador fracasará en su intento por contener el hoyo negro del laboratorio.

Explorar libremente el mapa del videojuego.

Como estudiante de la plataforma Aulify

Puedo explorar libremente el mapa del videojuego en primera persona

Para acceder a las estaciones de trabajo en donde se encuentran las dinámicas y retos matemáticos.

Prioridad: Baja.

Criterios de aceptación.

Dado que se haya iniciado sesión en el videojuego

Cuando se haya presionado el botón de jugar una nueva partida

Entonces el jugador, representado como un personaje en primera persona del videojuego, podrá desplazarse libremente y experimentar el entorno en tercera dimensión.

Detalles.

- El personaje del jugador principal solamente se desplazará en una sola velocidad en el mapa del videojuego.

- Dicho personaje no será capaz de hacer otro tipo de acción, más que caminar y seleccionar estaciones de trabajo para resolver sus retos.

- El personaje que controla el jugador principal siempre comenzará la partida en el pasillo principal del laboratorio para, posteriormente, adentrarse en la sala de estaciones del juego.

Resolver dinámicas y retos del videojuego.

Como jugador principal del videojuego

Puedo resolver cada una de las dinámicas presentadas en el mismo

Para evitar que el cronómetro cese su conteo, termine satisfactoriamente la partida y fortalezca los conocimientos en matemáticas básicas.

Prioridad: Alta.

Criterios de aceptación.

Dado que estoy registrado en Aulify y he iniciado sesión en el sistema con anterioridad

Cuando me tope con cada uno de los cuatro dispositivos ubicados en el mapa

Entonces podré acceder al reto o dinámica presente en ellos y resolverlos mediante entrada de caracteres en la consola del dispositivo.

Detalles.

- El videojuego cuenta con cuatro estaciones de trabajo repartidas en distintas posiciones del mapa en donde se presentan las dinámicas y retos matemáticos.
- Los retos del videojuego están relacionados a temas indicados en el temario de los cursos de matemáticas impartidos por Aulify para infantes de primaria.
- Al resolver cada reto, dependiendo de la ponderación y dificultad de este, se verá reflejado de manera positiva en el tiempo restante para que el cronómetro termine su conteo. De otro modo, el tiempo disminuirá si la respuesta es incorrecta.
- Las entradas permitidas en cada estación del videojuego solamente son caracteres.

Merecer insignias y logros a través de partidas jugadas.

Como jugador principal del videojuego

Puedo recibir insignias y logros de mis acciones en el videojuego

Para consultarlas posteriormente en mi historial de logros.

Prioridad: Media.

Criterios de aceptación.

Dado que soy el jugador principal y he jugado diversas partidas en el videojuego

Cuando realice algunas acciones que representen un hecho no común al jugar

Entonces podré recibir una insignia o logro que aparezca en mi historial de logros y me identifique con otros jugadores.

Detalles.

- Las insignias o logros pueden ser ganadas no solo por un jugador en el sistema, sino que pueden ser designadas a más de uno de ellos.
- Las insignias contienen un título, una descripción y una clasificación de la categoría a la que pertenecen.

Acceder al historial de logros.

Como jugador principal del videojuego

Puedo consultar el número e información de las insignias que he recibido por mis partidas

Para conocer las estadísticas que me identifican como jugador.

Prioridad: Media.

Criterios de aceptación.

Dado que soy el jugador principal y he jugado diversas partidas en el videojuego

Cuando acceda al botón que me redirige a la sección del historial de insignias

Entonces podré visualizar las estadísticas pertenecientes a mí a través de insignias y logros obtenidos.

Detalles.

- El historial de logros se despliega en forma de tabla para mantener un mayor orden en el despliegue de los datos.
- El historial de logros también contiene información general del usuario y sus partidas jugadas.
- Esta sección solamente despliega información del jugador principal que ha iniciado sesión en el videojuego.

Cerrar sesión de usuario.

Como jugador principal del videojuego

Puedo cerrar mi sesión de usuario

Para proteger mi información de jugador.

Prioridad: Baja.

Criterios de aceptación.

Dado que he iniciado sesión en el sistema

Cuando acceda al botón que cierra la sesión de los usuarios

Entonces podré abandonar el dispositivo en el que se acaba de jugar el videojuego y desplegar la página web.

Detalles.

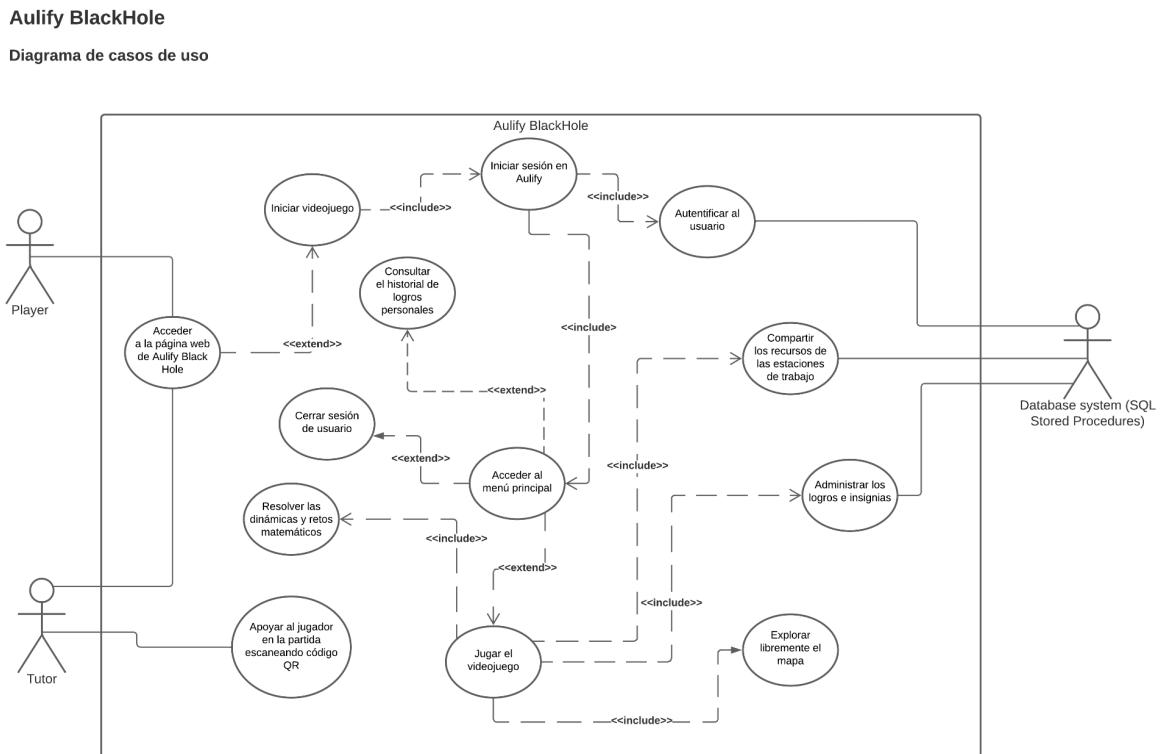
- El botón de cierre de sesión se encontrará en el menú principal del videojuego.

Requerimientos no funcionales.

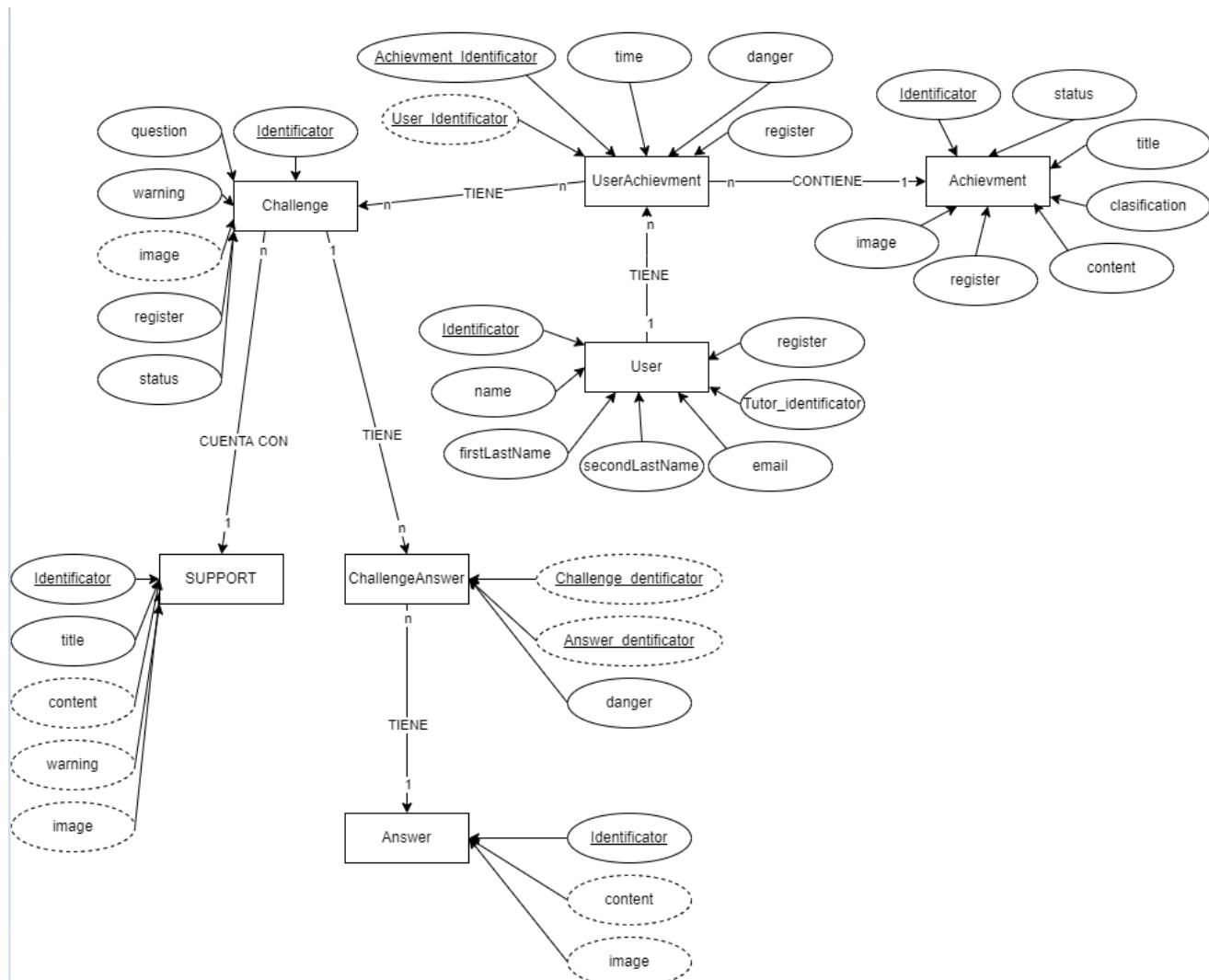
1. La página web debe registrar en la base de datos un usuario.
2. La página web debe verificar que se trate de un usuario registrado al iniciar sesión.
3. La página web debe almacenar la información del usuario (en caso que sea nuevo).
4. La página web debe contar con seguridad de sesiones de usuario.
5. La página web debe contener el juego antes desarrollado.
6. La página web debe permitir a los administradores acceder a las estadísticas de los jugadores.
7. El sistema del juego debe verificar que se trate de un usuario registrado al iniciar sesión.
8. El sistema del juego debe poder ejecutarse en distintas PC.
9. El sistema el juego debe contar con un tutorial que ayude al jugador.
10. El sistema del juego debe almacenar logros del jugador.

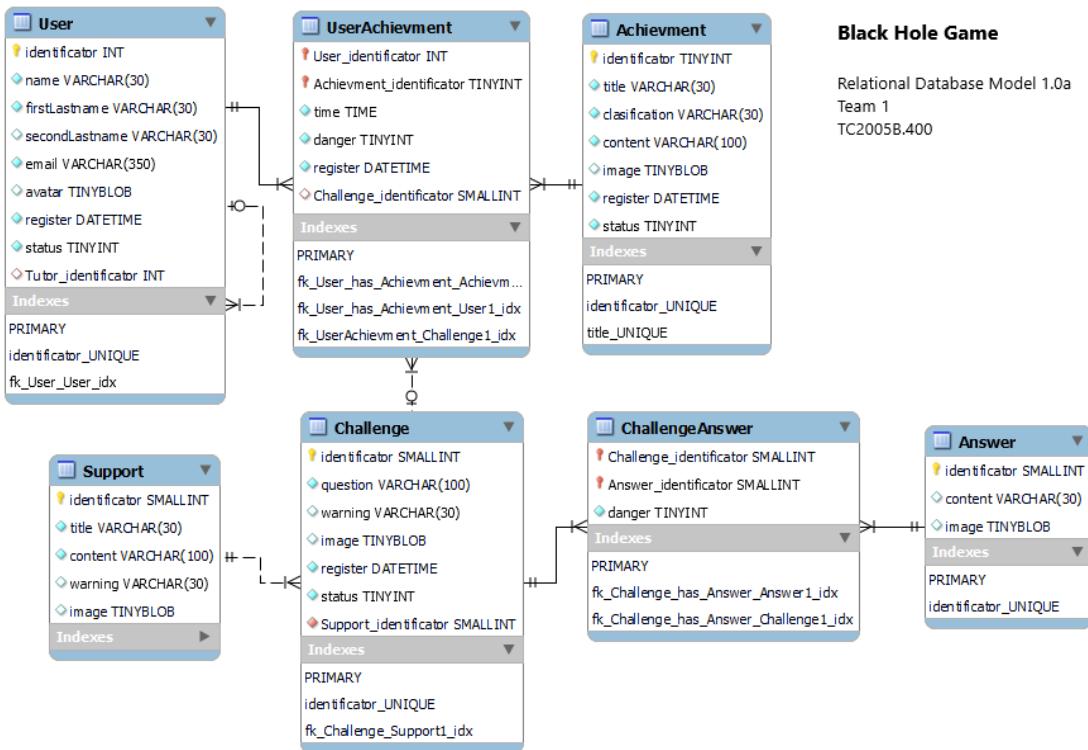
11. El sistema del juego debe identificar las respuestas correctas y errores del jugador.

Diagrama de casos de uso.



Modelo ERD y relacional de la Base de Datos de Aulify BlackHole.





Código SQL para implementación y carga de información.

```
-- Team 1, BlackHole
-- MySQL Workbench Forward Engineering
```

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERR
OR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
  
```

```
-- Schema blackhole
```

```
-- Schema blackhole
```

```
CREATE SCHEMA IF NOT EXISTS `blackhole` DEFAULT CHARACTER SET utf8 ;
USE `blackhole` ;
```

```
-- Table `blackhole`.`User`
```

```
CREATE TABLE IF NOT EXISTS `blackhole`.`User` (
  `identificador` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(30) NOT NULL,
  `firstLastname` VARCHAR(30) NOT NULL,
  `secondLastname` VARCHAR(30) NULL,
  `email` VARCHAR(350) NOT NULL,
  `avatar` TINYBLOB NULL,
  `register` DATETIME NOT NULL DEFAULT NOW(),
  `status` TINYINT UNSIGNED NOT NULL DEFAULT 1,
  `Tutor_identificador` INT UNSIGNED NULL,
  PRIMARY KEY (`identificador`),
  UNIQUE INDEX `identificador_UNIQUE` (`identificador` ASC) VISIBLE,
  INDEX `fk_User_User_idx` (`Tutor_identificador` ASC) VISIBLE,
  CONSTRAINT `fk_User_User`
    FOREIGN KEY (`Tutor_identificador`)
    REFERENCES `blackhole`.`User` (`identificador`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- Table `blackhole`.`Support`
```

```
CREATE TABLE IF NOT EXISTS `blackhole`.`Support` (
  `identificador` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
```



```
`title` VARCHAR(30) NOT NULL,  
`content` VARCHAR(100) NOT NULL,  
`warning` VARCHAR(30) NULL,  
`image` TINYBLOB NULL,  
PRIMARY KEY (`identificador`),  
UNIQUE INDEX `identificador_UNIQUE` (`identificador` ASC) VISIBLE)  
ENGINE = InnoDB;
```

```
-- Table `blackhole`.`Challenge`
```

```
CREATE TABLE IF NOT EXISTS `blackhole`.`Challenge` (  
`identificador` SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
`question` VARCHAR(100) NOT NULL,  
`warning` VARCHAR(30) NULL,  
`image` TINYBLOB NULL,  
`register` DATETIME NOT NULL DEFAULT NOW(),  
`status` TINYINT UNSIGNED NOT NULL DEFAULT 1,  
`Support_identificador` SMALLINT UNSIGNED NOT NULL,  
PRIMARY KEY (`identificador`),  
UNIQUE INDEX `identificador_UNIQUE` (`identificador` ASC) VISIBLE,  
INDEX `fk_Challenge_Support1_idx` (`Support_identificador` ASC) VISIBLE,  
CONSTRAINT `fk_Challenge_Support1`  
    FOREIGN KEY (`Support_identificador`)  
    REFERENCES `blackhole`.`Support` (`identificador`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- Table `blackhole`.`Answer`
```

```
CREATE TABLE IF NOT EXISTS `blackhole`.`Answer` (  
`identificador` SMALLINT NOT NULL AUTO_INCREMENT,  
`content` VARCHAR(30) NULL,  
`image` TINYBLOB NULL,1  
PRIMARY KEY (`identificador`),
```

```
UNIQUE INDEX `identifier_UNIQUE` (`identifier` ASC) VISIBLE
ENGINE = InnoDB;
```

-- Table `blackhole`.`Achievement`

```
CREATE TABLE IF NOT EXISTS `blackhole`.`Achievement` (
  `identifier` TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,
  `title` VARCHAR(30) NOT NULL,
  `clasification` VARCHAR(30) NOT NULL,
  `content` VARCHAR(100) NOT NULL,
  `image` TINYBLOB NULL,
  `register` DATETIME NOT NULL DEFAULT NOW(),
  `status` TINYINT UNSIGNED NOT NULL DEFAULT 1,
  PRIMARY KEY (`identifier`),
  UNIQUE INDEX `identifier_UNIQUE` (`identifier` ASC) VISIBLE,
  UNIQUE INDEX `title_UNIQUE` (`title` ASC) VISIBLE
)
ENGINE = InnoDB;
```

-- Table `blackhole`.`ChallengeAnswer`

```
CREATE TABLE IF NOT EXISTS `blackhole`.`ChallengeAnswer` (
  `Challenge_identifier` SMALLINT UNSIGNED NOT NULL,
  `Answer_identifier` SMALLINT NOT NULL,
  `danger` TINYINT UNSIGNED NOT NULL DEFAULT 0,
  PRIMARY KEY (`Challenge_identifier`, `Answer_identifier`),
  INDEX `fk_Challenge_has_Answer_Answer1_idx` (`Answer_identifier` ASC)
VISIBLE,
  INDEX `fk_Challenge_has_Answer_Challenge1_idx` (`Challenge_identifier` ASC)
VISIBLE,
  CONSTRAINT `fk_Challenge_has_Answer_Challenge1`
  FOREIGN KEY (`Challenge_identifier`)
  REFERENCES `blackhole`.`Challenge` (`identifier`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_Challenge_has_Answer_Answer1`
```

```

    FOREIGN KEY (`Answer_identificator`)
    REFERENCES `blackhole`.`Answer` (`identificator`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-- Table `blackhole`.`UserAchievement`

CREATE TABLE IF NOT EXISTS `blackhole`.`UserAchievement` (
    `User_identificator` INT UNSIGNED NOT NULL,
    `Achievement_identificator` TINYINT UNSIGNED NOT NULL,
    `time` TIME NOT NULL,
    `danger` TINYINT UNSIGNED NOT NULL,
    `register` DATETIME NOT NULL DEFAULT NOW(),
    `Challenge_identificator` SMALLINT UNSIGNED NULL,
    PRIMARY KEY (`User_identificator`, `Achievement_identificator`),
    INDEX `fk_User_has_Achievement_Achievement1_idx` (`Achievement_identificator` ASC)
VISIBLE,
    INDEX `fk_User_has_Achievement_User1_idx` (`User_identificator` ASC) VISIBLE,
    INDEX `fk_UserAchievement_Challenge1_idx` (`Challenge_identificator` ASC)
VISIBLE,
    CONSTRAINT `fk_User_has_Achievement_User1`
        FOREIGN KEY (`User_identificator`)
        REFERENCES `blackhole`.`User` (`identificator`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_User_has_Achievement_Achievement1`
        FOREIGN KEY (`Achievement_identificator`)
        REFERENCES `blackhole`.`Achievement` (`identificator`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_UserAchievement_Challenge1`
        FOREIGN KEY (`Challenge_identificator`)
        REFERENCES `blackhole`.`Challenge` (`identificator`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```
SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

-- -----
-- Data for table `blackhole`.`User`
-- -----


START TRANSACTION;
USE `blackhole`;
INSERT INTO `blackhole`.`User` (`identificador`, `name`, `firstname`, `secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identificador`)
VALUES (1, 'Emiliano', 'Vivas', 'Rodríguez', 'vivasrguez@outlook.com', NULL,
'2023-03-18 06:52:32', 1, NULL);
INSERT INTO `blackhole`.`User` (`identificador`, `name`, `firstname`, `secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identificador`)
VALUES (2, 'Axel', 'Corona', 'Ibarra', 'elaxel@yahoo.com', NULL, '2023-03-18
07:05:12', 1, NULL);
INSERT INTO `blackhole`.`User` (`identificador`, `name`, `firstname`, `secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identificador`)
VALUES (3, 'Axel', 'Espinoza', NULL, 'espinoza@gmail.com', NULL, '2023-03-18
08:09:12', 1, NULL);
INSERT INTO `blackhole`.`User` (`identificador`, `name`, `firstname`, `secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identificador`)
VALUES (4, 'Daniel', 'De Turner', 'Islas', 'danny@yahoo.com', NULL, '2023-02-15
14:09:34', 1, NULL);
INSERT INTO `blackhole`.`User` (`identificador`, `name`, `firstname`, `secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identificador`)
VALUES (5, 'Jimena', 'Vivas', 'Rodríguez', 'jimenajauregui@gmail.com', NULL,
'2023-01-23 07:15:34', 1, NULL);
INSERT INTO `blackhole`.`User` (`identificador`, `name`, `firstname`, `secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identificador`)
VALUES (6, 'Ademir', 'Erazo', 'Rodríguez', 'erazoade@gmail.com', NULL,
'2022-05-12 09:11:10', 1, NULL);
INSERT INTO `blackhole`.`User` (`identificador`, `name`, `firstname`, `secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identificador`)
VALUES (7, 'Adriana', 'Rodríguez', 'Ocampo', 'adrirodriguez@gmail.com', NULL,
'2023-06-12 10:04:56', 1, NULL);
```

```

INSERT INTO `blackhole`.`User` (`identifier`, `name`, `firstname`,
`secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identifier`)
VALUES (8, 'Luna', 'Vivas', 'Holguín', 'lunita@outlook.com', NULL,
'2023-03-18 07:00:00', 1, NULL);
INSERT INTO `blackhole`.`User` (`identifier`, `name`, `firstname`,
`secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identifier`)
VALUES (9, 'Sergio Aldahir', 'Lome', 'Villegas', 'villegas123@gmail.com', NULL,
'2022-12-12 07:12:13', 1, NULL);
INSERT INTO `blackhole`.`User` (`identifier`, `name`, `firstname`,
`secondlastname`, `email`, `avatar`, `register`, `status`, `Tutor_identifier`)
VALUES (10, 'Jacobo', 'Zabludowski', NULL, 'zabomafu@gmail.com', NULL,
'2023-12-01 09:29:36', 1, NULL);

COMMIT;

```

```

-- -----
-- Data for table `blackhole`.`Support`

-- -----
START TRANSACTION;
USE `blackhole`;
INSERT INTO `blackhole`.`Support` (`identifier`, `title`, `content`,
`warning`, `image`) VALUES (1, 'Secretos de figuras', 'Las figuras vistas desde
otra perspectiva contienen ciertos patrones...', 'Observa bien el panel', NULL);

COMMIT;

```

```

-- -----
-- Data for table `blackhole`.`Challenge`

-- -----
START TRANSACTION;
USE `blackhole`;
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (1, '5 x 5', NULL,
NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (2, '6 x 6', NULL,
NULL, '2022-11-01 08:10:00', 1, 1);

```

```

INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (3, '7 x 7', NULL,
NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (4, '8 x 8', NULL,
NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (5, '9 x 9', NULL,
NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (6, '10 x 10',
NULL, NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (7, '11 + 11',
NULL, NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (8, '12 + 12',
NULL, NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (9, '13 - 12',
NULL, NULL, '2022-11-01 08:10:00', 1, 1);
INSERT INTO `blackhole`.`Challenge` (`identifier`, `question`, `warning`,
`image`, `register`, `status`, `Support_identifier`) VALUES (10, '14 / 2',
NULL, NULL, '2022-11-01 08:10:00', 1, 1);

COMMIT;

```

```

-- Data for table `blackhole`.`Answer`

START TRANSACTION;
USE `blackhole`;
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (1,
'4', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (2,
'65', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (3,
'9', NULL);

```

```

INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (4,
'25', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (5,
'90', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (6,
'100', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (7,
'64', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (8,
'12', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES (9,
'18', NULL);
INSERT INTO `blackhole`.`Answer` (`identifier`, `content`, `image`) VALUES
(10, '34', NULL);

COMMIT;

```

```

-- Data for table `blackhole`.`Achievement`

START TRANSACTION;
USE `blackhole`;
INSERT INTO `blackhole`.`Achievement` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (1, 'Tiempo reloj', 'Tiempo',
'Terminar una partida en menos de 10 minutos', NULL, '2022-10-01 03:10:00', 1);
INSERT INTO `blackhole`.`Achievement` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (2, 'Modo relámpago',
'Pregunta', 'Contestar correctamente una pregunta en menos de 5 segundos', NULL,
'2022-10-01 03:10:00', 1);
INSERT INTO `blackhole`.`Achievement` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (3, 'Equipo dinamita',
'Pregunta', 'Terminar una partida asistida correctamente en menos de 5 minutos',
NULL, '2022-10-01 03:10:00', 1);
INSERT INTO `blackhole`.`Achievement` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (4, 'Nado sincronizado',
'Estrategia', 'Responder las preguntas de cada estación de forma cronológica',
NULL, '2022-10-01 03:10:00', 0);

```

```

INSERT INTO `blackhole`.`Achievment` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (5, 'Cerebrito', 'Pregunta',
'Ganar una partida sin comentar ninguna equivocación', NULL, '2022-10-01
03:10:00', 1);
INSERT INTO `blackhole`.`Achievment` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (6, 'Rey de los jeroglíficos',
'Pregunta', 'Contestar correctamente todas las preguntas de formas y figuras de
la partida', NULL, '2022-10-01 03:10:00', 1);
INSERT INTO `blackhole`.`Achievment` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (7, 'Ilustre matemático',
'Habilidad', 'Completar 10 partidas de forma exitosa', NULL, '2022-10-01
03:10:00', 1);
INSERT INTO `blackhole`.`Achievment` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (8, 'Llamada de emergencia',
'Tiempo', 'Contener al hoyo negro en menos de 5 minutos', NULL, '2022-10-01
03:10:00', 1);
INSERT INTO `blackhole`.`Achievment` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (9, 'Genio de la lámpara',
'Pregunta', 'Contestar las preguntas de álgebra de la partida correctamente',
NULL, '2022-10-01 03:10:00', 1);
INSERT INTO `blackhole`.`Achievment` (`identifier`, `title`, `clasification`,
`content`, `image`, `register`, `status`) VALUES (10, 'Acephador', 'Habilidad',
'Contener 10 hoyos negros consecutivamente', NULL, '2022-10-01 03:10:00', 1);

COMMIT;

```

```

-- Data for table `blackhole`.`ChallengeAnswer`

START TRANSACTION;
USE `blackhole`;
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (1, 5, 4);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (2, 3, 4);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (3, 10, 3);

```

```

INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (4, 4, 7);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (5, 7, 9);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (6, 1, 1);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (7, 7, 2);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (8, 6, 6);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (9, 2, 3);
INSERT INTO `blackhole`.`ChallengeAnswer` (`Challenge_identifier`,
`Answer_identifier`, `danger`) VALUES (10, 8, 5);

COMMIT;

```

```

-- Data for table `blackhole`.`UserAchievement`

START TRANSACTION;
USE `blackhole`;
INSERT INTO `blackhole`.`UserAchievement` (`User_identifier`,
`Achievement_identifier`, `time`, `danger`, `register`,
`Challenge_identifier`) VALUES (1, 9, '00:05:12', 7, '2023-10-01 03:10:00',
NULL);
INSERT INTO `blackhole`.`UserAchievement` (`User_identifier`,
`Achievement_identifier`, `time`, `danger`, `register`,
`Challenge_identifier`) VALUES (4, 10, '00:10:34', 9, '2023-10-01 03:10:00',
NULL);
INSERT INTO `blackhole`.`UserAchievement` (`User_identifier`,
`Achievement_identifier`, `time`, `danger`, `register`,
`Challenge_identifier`) VALUES (8, 4, '00:08:23', 1, '2022-10-01 03:10:00',
NULL);
INSERT INTO `blackhole`.`UserAchievement` (`User_identifier`,
`Achievement_identifier`, `time`, `danger`, `register`,
`Challenge_identifier`) VALUES (5, 3, '00:05:23', 4, '2022-10-01 03:10:00',
NULL);

```

```
INSERT INTO `blackhole`.`UserAchievement` (`User_identificator`,  
`Achievment_identificator`, `time`, `danger`, `register`,  
`Challenge_identificator`) VALUES (6, 9, '00:06:12', 5, '2022-10-01 03:10:00',  
NULL);  
INSERT INTO `blackhole`.`UserAchievement` (`User_identificator`,  
`Achievment_identificator`, `time`, `danger`, `register`,  
`Challenge_identificator`) VALUES (7, 7, '00:12:12', 12, '2022-10-01 03:10:00',  
NULL);  
INSERT INTO `blackhole`.`UserAchievement` (`User_identificator`,  
`Achievment_identificator`, `time`, `danger`, `register`,  
`Challenge_identificator`) VALUES (10, 5, '00:13:23', 4, '2022-10-01 03:10:00',  
NULL);  
INSERT INTO `blackhole`.`UserAchievement` (`User_identificator`,  
`Achievment_identificator`, `time`, `danger`, `register`,  
`Challenge_identificator`) VALUES (3, 8, '00:06:12', 5, '2022-10-01 03:10:00',  
NULL);  
INSERT INTO `blackhole`.`UserAchievement` (`User_identificator`,  
`Achievment_identificator`, `time`, `danger`, `register`,  
`Challenge_identificator`) VALUES (4, 4, '00:07:15', 8, '2022-10-01 03:10:00',  
NULL);  
INSERT INTO `blackhole`.`UserAchievement` (`User_identificator`,  
`Achievment_identificator`, `time`, `danger`, `register`,  
`Challenge_identificator`) VALUES (5, 4, '00:15:56', 14, '2022-10-01 03:10:00',  
NULL);  
  
COMMIT;
```

Comprobación de carga de datos.

```

1 • use blackhole;
2
3 • select * from User;

```

Result Grid

| identificador | name | firstname | secondlastname | email | avatar | register | status | Tutor_id |
|---------------|----------------|-------------|----------------|-------------------------|--------|---------------------|--------|----------|
| 1 | Emiliano | Vivas | Rodríguez | vivarsrguez@outlook.com | NULL | 2023-03-18 06:52:32 | 1 | NULL |
| 2 | Axel | Corona | Ibarra | elaxel@yahoo.com | NULL | 2023-03-18 07:05:12 | 1 | NULL |
| 3 | Axel | Espinosa | NULL | espinosa@gmail.com | NULL | 2023-03-18 08:09:12 | 1 | NULL |
| 4 | Daniel | De Turner | Islas | danny@yahoo.com | NULL | 2023-02-15 14:09:34 | 1 | NULL |
| 5 | Jimena | Vivas | Rodríguez | jimenajuregu@gmail.com | NULL | 2023-01-23 07:15:34 | 1 | NULL |
| 6 | Ademir | Erazo | Rodríguez | erazaade@gmail.com | NULL | 2022-05-12 09:11:10 | 1 | NULL |
| 7 | Adriana | Rodríguez | Ocampo | adrirodriguez@gmail.com | NULL | 2023-06-12 10:04:56 | 1 | NULL |
| 8 | Luna | Vivas | Holgún | lunita@outlook.com | NULL | 2023-03-18 07:00:00 | 1 | NULL |
| 9 | Sergio Aldahir | Lome | Vilegas | vilegas123@gmail.com | NULL | 2022-12-12 07:12:13 | 1 | NULL |
| 10 | Jacobo | Zabłudowski | NULL | zabomafu@gmail.com | NULL | 2023-12-01 09:29:36 | 1 | NULL |
| • | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

```

1 • use blackhole;
2
3 • select * from Achievement;

```

Result Grid

| identificador | title | clasification | content | image | register | status |
|---------------|-------------------------|---------------|---|-------|---------------------|--------|
| 1 | Tiempo reloj | Tiempo | Terminar una partida en menos de 10 minutos | NULL | 2022-10-01 03:10:00 | 1 |
| 2 | Modo relámpago | Pregunta | Contestar correctamente una pregunta en menos de 10 segundos | NULL | 2022-10-01 03:10:00 | 0 |
| 3 | Equipo dinamita | Pregunta | Terminar una partida asistida correctamente en menos de 10 segundos | NULL | 2022-10-01 03:10:00 | 1 |
| 4 | Nodo sincronizado | Estrategia | Responder las preguntas de cada estación de forma correcta | NULL | 2022-10-01 03:10:00 | 0 |
| 5 | Cerebrito | Pregunta | Ganar una partida sin comentar ninguna equivocación | NULL | 2022-10-01 03:10:00 | 1 |
| 6 | Rey de los jeroglíficos | Pregunta | Contestar correctamente todas las preguntas de la categoría | NULL | 2022-10-01 03:10:00 | 1 |
| 7 | Ilustre matemático | Habilidad | Completar 10 partidas de forma exitosa | NULL | 2022-10-01 03:10:00 | 1 |
| 8 | Llamada de emergencia | Tiempo | Contener al hoyo negro en menos de 5 minutos | NULL | 2022-10-01 03:10:00 | 1 |
| 9 | Genio de la lámpara | Pregunta | Contestar las preguntas de álgebra de la partida | NULL | 2022-10-01 03:10:00 | 1 |
| 10 | Acechador | Habilidad | Contener 10 hoyos negros consecutivamente | NULL | 2022-10-01 03:10:00 | 1 |
| • | NULL | NULL | NULL | NULL | NULL | NULL |

```

1 • use blackhole;
2
3 • select * from UserAchievement;

```

Result Grid

| User_Identifier | Achievement_Identifier | time | danger | register | Challenge_Identifier |
|-----------------|------------------------|----------|--------|---------------------|----------------------|
| 1 | 9 | 00:05:12 | 7 | 2023-10-01 03:10:00 | NULL |
| 3 | 8 | 00:06:12 | 5 | 2022-10-01 03:10:00 | NULL |
| 4 | 4 | 00:07:15 | 8 | 2022-10-01 03:10:00 | NULL |
| 4 | 10 | 00:10:34 | 9 | 2023-10-01 03:10:00 | NULL |
| 5 | 3 | 00:05:23 | 4 | 2022-10-01 03:10:00 | NULL |
| 5 | 4 | 00:15:56 | 14 | 2022-10-01 03:10:00 | NULL |
| 6 | 9 | 00:06:12 | 5 | 2022-10-01 03:10:00 | NULL |
| 7 | 7 | 00:12:11 | 12 | 2022-10-01 03:10:00 | NULL |
| 8 | 4 | 00:08:23 | 1 | 2022-10-01 03:10:00 | NULL |
| 10 | 5 | 00:13:23 | 4 | 2022-10-01 03:10:00 | NULL |
| • | NULL | NULL | NULL | NULL | NULL |

```
1 • use blackhole;
2
3 • select * from Challenges;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

| | identificador | question | warning | image | register | status | Support_identificador |
|----|---------------|----------|---------|-------|---------------------|--------|-----------------------|
| ▶ | 1 | 5 x 5 | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 2 | 6 x 6 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 3 | 7 x 7 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 4 | 8 x 8 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 5 | 9 x 9 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 6 | 10 x 10 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 7 | 11 + 11 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 8 | 12 + 12 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 9 | 13 - 12 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| 10 | 14 / 2 | NULL | NULL | NULL | 2022-11-01 08:10:00 | 1 | 1 |
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

```
1 • use blackhole;
2
3 • select * from Answers;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

| | identificador | content | image |
|----|---------------|---------|-------|
| ▶ | 1 | 4 | NULL |
| 2 | 65 | NULL | NULL |
| 3 | 9 | NULL | NULL |
| 4 | 25 | NULL | NULL |
| 5 | 90 | NULL | NULL |
| 6 | 100 | NULL | NULL |
| 7 | 64 | NULL | NULL |
| 8 | 12 | NULL | NULL |
| 9 | 18 | NULL | NULL |
| 10 | 34 | NULL | NULL |
| * | NULL | NULL | NULL |

```
1 • use blackhole;
2
3 • select * from ChallengeAnswer;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

| | Challenge_identificador | Answer_identificador | danger |
|----|-------------------------|----------------------|--------|
| ▶ | 1 | 5 | 4 |
| 2 | 3 | 4 | 3 |
| 3 | 10 | 3 | 3 |
| 4 | 4 | 7 | 7 |
| 5 | 7 | 9 | 9 |
| 6 | 1 | 1 | 1 |
| 7 | 7 | 2 | 2 |
| 8 | 6 | 6 | 6 |
| 9 | 2 | 3 | 3 |
| 10 | 8 | 5 | 5 |
| * | NULL | NULL | NULL |

```

1 • use blackhole;
2
3 • select * from Supports;

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

| identificador | title | content | warning | image |
|---------------|---------------------|--|-----------------------|-------|
| 1 | Secretos de figuras | Las figuras vistas desde otra perspectiva conte... | Observa bien el panel | |
| • | NULL | NULL | NULL | NULL |

Comandos requeridos para realizar la manipulación de la información.

```

3 -- Mostrar los usuarios con alguna insignia o logro registrado
4
5 • SELECT u.identificador, u.name, u.email FROM User AS u INNER JOIN UserAchievement AS ua
6 ON u.identificador = ua.User_identificador INNER JOIN Achievement AS ac
7 ON ua.Achievement_identificador = ac.identificador;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

| identificador | name | email |
|---------------|----------|--------------------------|
| 5 | Jimena | jimena.auregui@gmail.com |
| 4 | Daniel | danny@yahoo.com |
| 5 | Jimena | jimena.auregui@gmail.com |
| 8 | Luna | lunita@outlook.com |
| 10 | Jacobo | zabonafu@gmail.com |
| 7 | Adriana | adrirodriguez@gmail.com |
| 3 | Avel | espinoza@gmail.com |
| 1 | Emiliano | viverguez@outlook.com |
| 6 | Ademir | erazoode@gmail.com |
| 4 | Daniel | danny@yahoo.com |

```

9 -- Mostrar los logros del usuario no. 4
10
11 • SELECT ac.title FROM User AS u INNER JOIN UserAchievement AS ua
12 ON u.identificador = ua.User_identificador INNER JOIN Achievement AS ac
13 ON ua.Achievement_identificador = ac.identificador WHERE u.identificador = 4;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

| title |
|-------------------|
| Nodo sincronizado |
| Acechador |

```

15 -- Total de insignias del usuario 6
16
17 • SELECT COUNT(ac.identificador) AS "Total de insignias" from User AS u INNER JOIN UserAchievement AS ua
18 ON u.identificador = ua.User_identificador INNER JOIN Achievement AS ac
19 ON ua.Achievement_identificador = ac.identificador WHERE u.identificador = 6;
    
```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

| Total de insignias |
|--------------------|
| 1 |

Result Grid Form Editor Field Types

```

20
21 -- Identificador de soporte asociado a la pregunta no. 5
22
23 • SELECT s.identificador, s.title FROM Support AS s INNER JOIN Challenge AS c
24 ON s.identificador = c.Support_identificador WHERE c.identificador = 5;
    
```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

| identificador | title |
|---------------|---------------------|
| 1 | Secretos de figuras |

Result Grid Form Editor Field Types

```

26 -- Respuesta correcta de la pregunta no. 5
27
28 • SELECT a.identificador, a.content FROM Answer AS a INNER JOIN ChallengeAnswer AS ca
29 ON a.identificador = ca.Answer_identificador INNER JOIN Challenge AS c
30 ON c.identificador = ca.Challenge_identificador WHERE c.identificador = 5;
    
```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

| identificador | content |
|---------------|---------|
| 7 | 64 |

Result Grid Form Editor Field Types

```

32 -- Historial de logro detallado del usuario no. 4
33
34 • SELECT ac.title AS "Nombre de logro", ac.content AS "Descripción", ua.time AS "Tiempo necesario", ua.registered AS "Fecha"
35 ON u.identificador = ua.User_identificador INNER JOIN Achievement AS ac
36 ON ua.Achievement_identificador = ac.identificador WHERE u.identificador = 4;
    
```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

| Nombre de logro | Descripción | Tiempo necesario | Fecha |
|-------------------|--|------------------|---------------------|
| Nado sincronizado | Responder las preguntas de cada estación de f... | 00:07:15 | 2022-10-01 03:10:00 |
| Acechador | Contener 10 hoyos negros consecutivamente | 00:10:34 | 2023-10-01 03:10:00 |

Result Grid Form Editor Field Types

Vistas de la Arquitectura 4+1 de Kruchten.

La arquitectura 4+1 es un modelo de arquitectura de software que se utiliza para describir el diseño de sistemas de software. Fue propuesto por Philippe Kruchten en 1995 y consta de cinco vistas diferentes del sistema que representan diferentes perspectivas o preocupaciones de los stakeholders del proyecto.

Cada vista del modelo 4+1 va dirigida a diferentes grupos de stakeholders del proyecto.

1. La vista de casos de uso está dirigida principalmente a los usuarios finales del sistema, así como a los analistas de requisitos y a los expertos en dominio que deben entender cómo se utilizará el sistema y qué funcionalidades deben ser proporcionadas.
2. La vista lógica está dirigida principalmente a los arquitectos y diseñadores de software, así como a los desarrolladores de software que deben entender cómo se organiza el código y cómo se relacionan los componentes del sistema.
3. La vista de implementación está dirigida principalmente a los administradores de sistemas y a los operadores de redes que deben comprender cómo se despliega el software en el hardware y cómo se distribuyen los componentes en la red.
4. La vista de procesos está dirigida principalmente a los diseñadores y desarrolladores de software que deben entender cómo se ejecuta el software, cómo se gestionan los procesos y la concurrencia en el sistema.
5. Finalmente, la vista de escenarios está dirigida a todos los stakeholders del proyecto que necesitan entender cómo funcionará el sistema en situaciones específicas y cómo los casos de uso se realizan en el sistema.

El modelo 4+1 es una herramienta útil para los arquitectos de software, ya que les permite crear una representación completa del sistema desde diferentes perspectivas. También permite que diferentes stakeholders se centren en las vistas que les resulten más relevantes. Se utilizó la herramienta LucidChart para la realización de los diagramas.

Escenario. Diagrama de casos de uso.

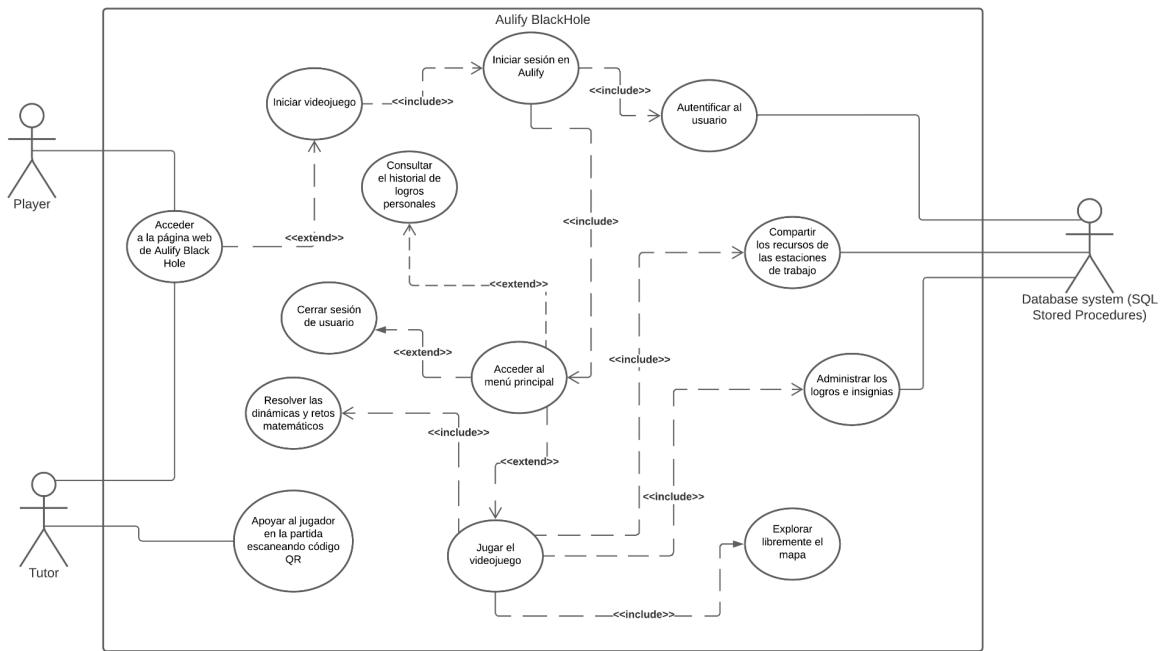
BLACK HOLE

Aulify BlackHole

Diagrama de casos de uso

EQUIPO 1

18/03/2023

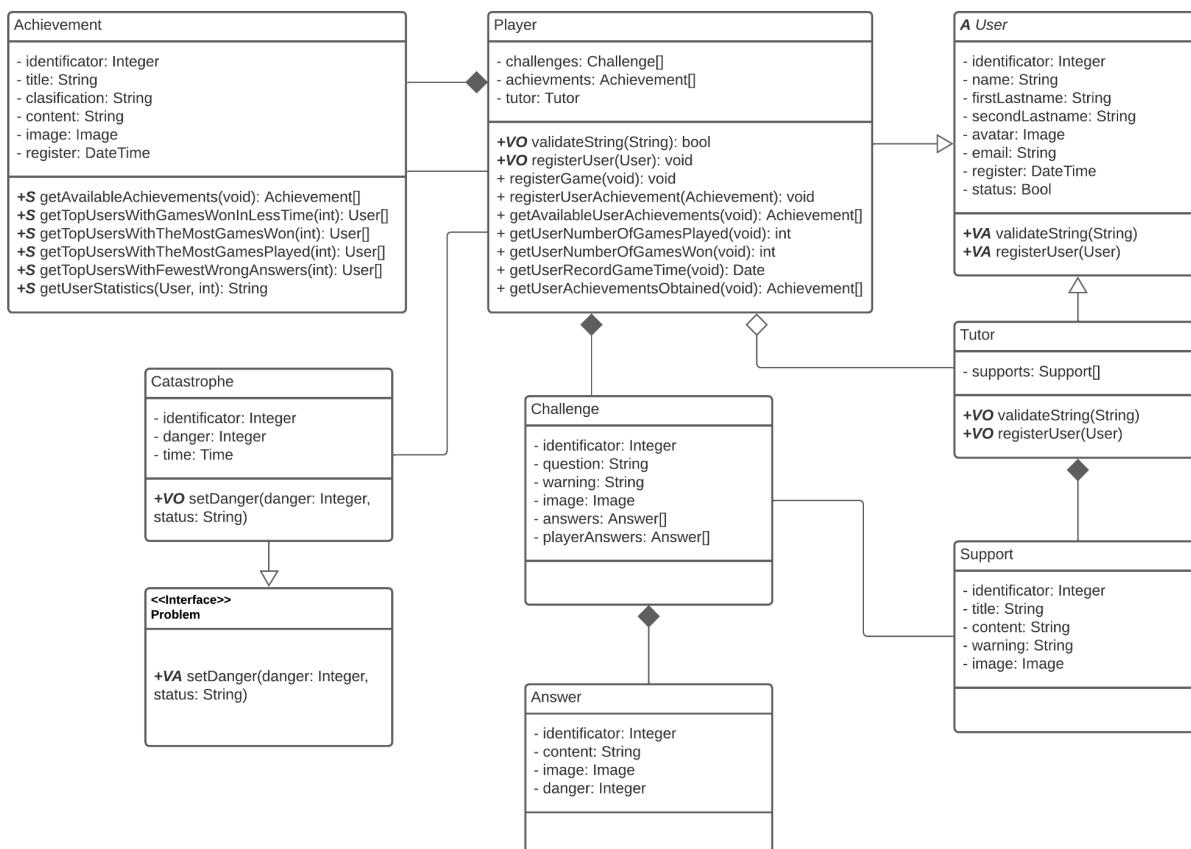


Lógica. Diagrama de clases.

Aulify BlackHole

Diagrama de clases

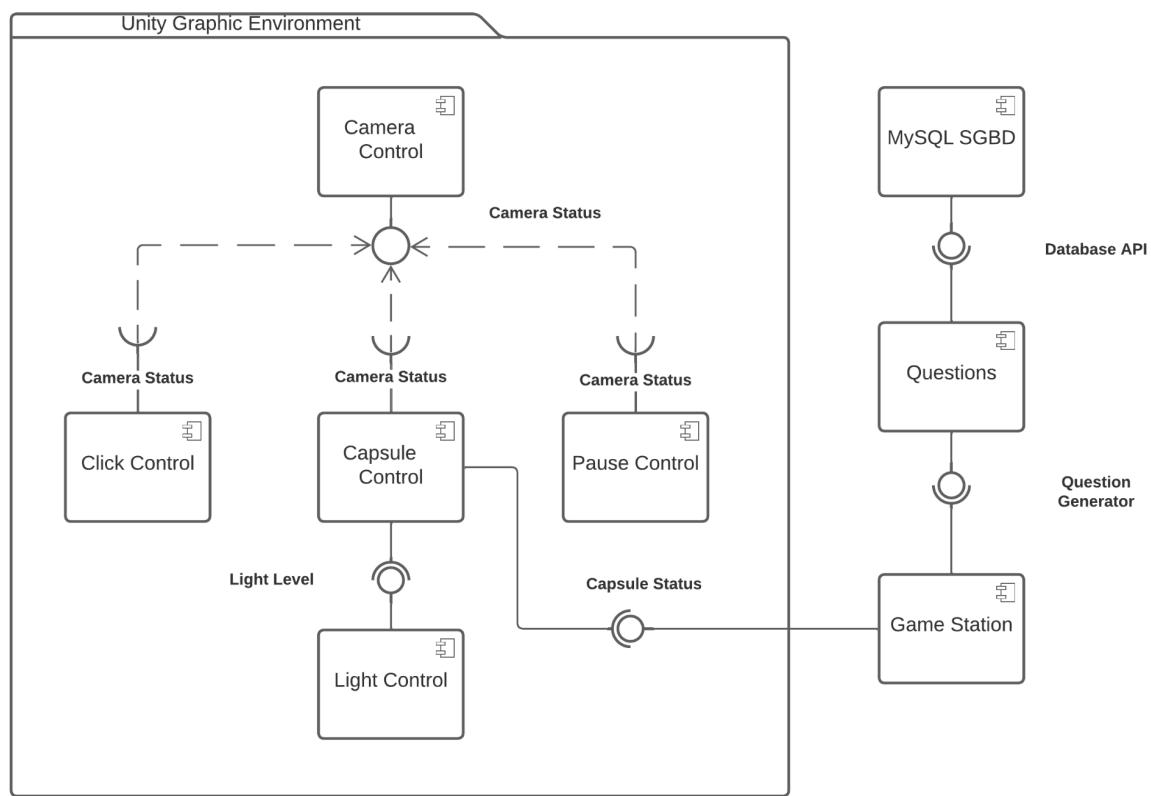
Note. All designed classes have get and set properties (**get; set;**) for each of their members. To simplify the diagram, it's worth noting their implicit existence in each class.



Despliegue. Diagrama de componentes.

Aulify BlackHole

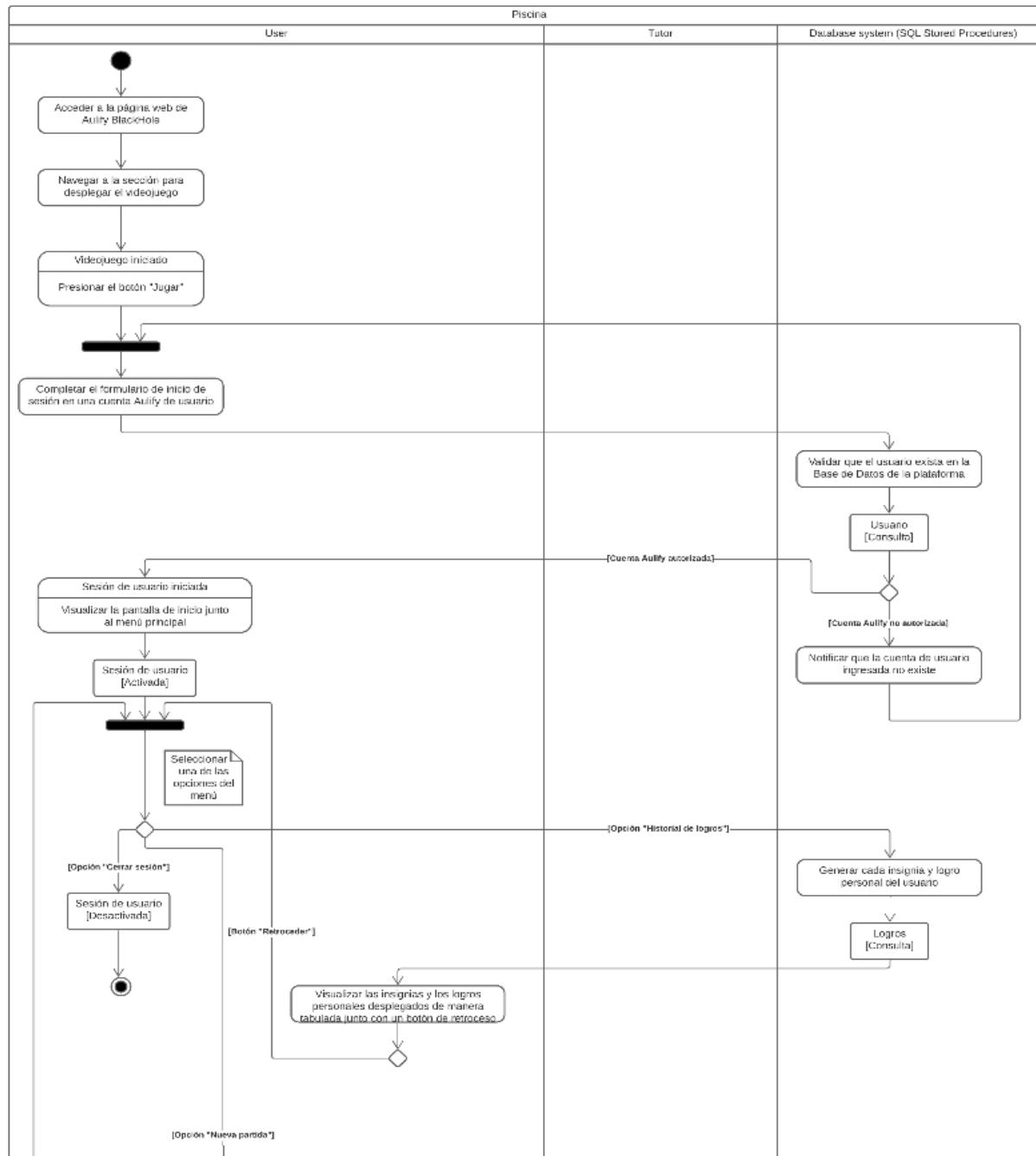
Diagrama de componentes de ejecución de la API Unity

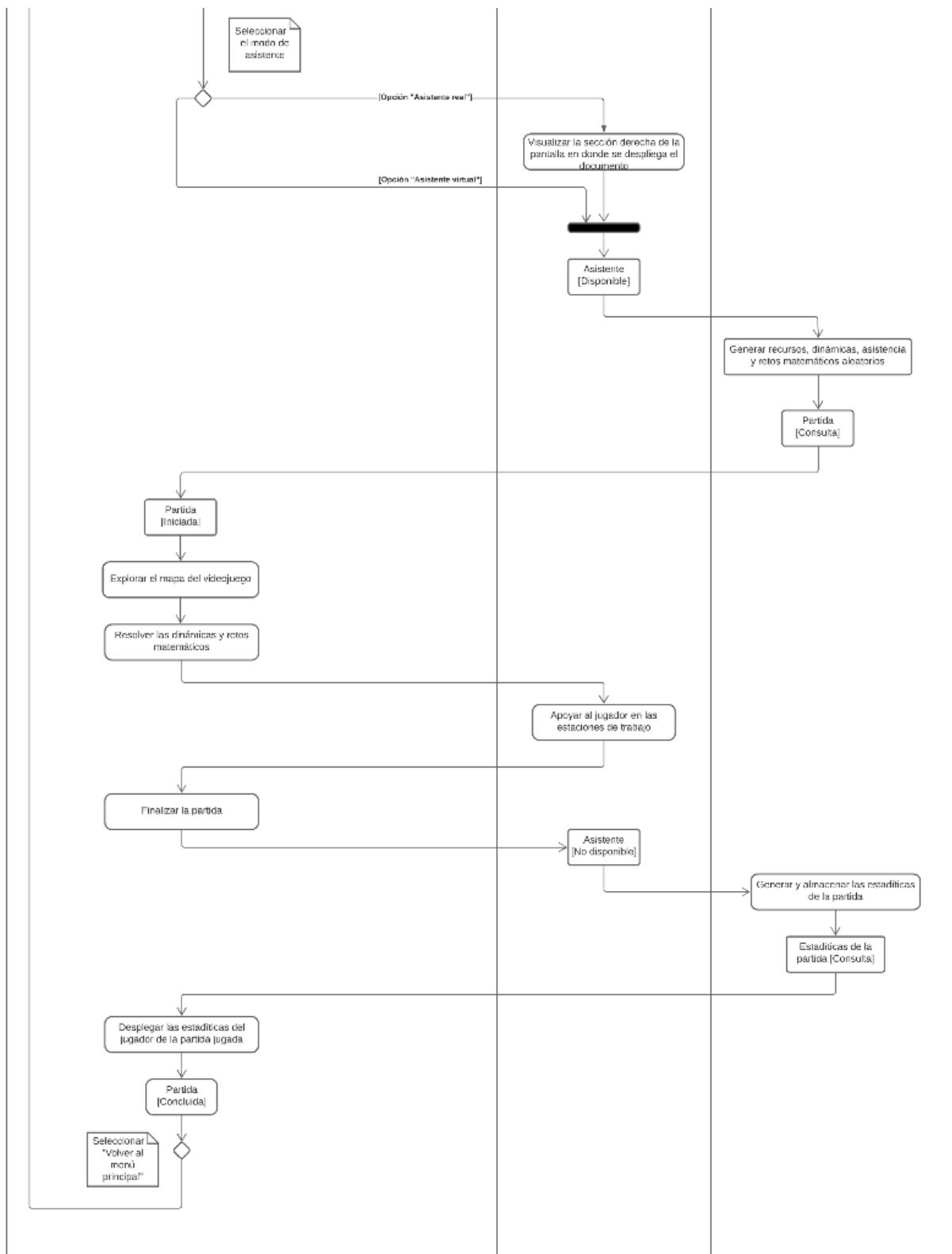


De procesos. Diagrama de actividades.

Aulify BlackHole

Diagrama de actividades

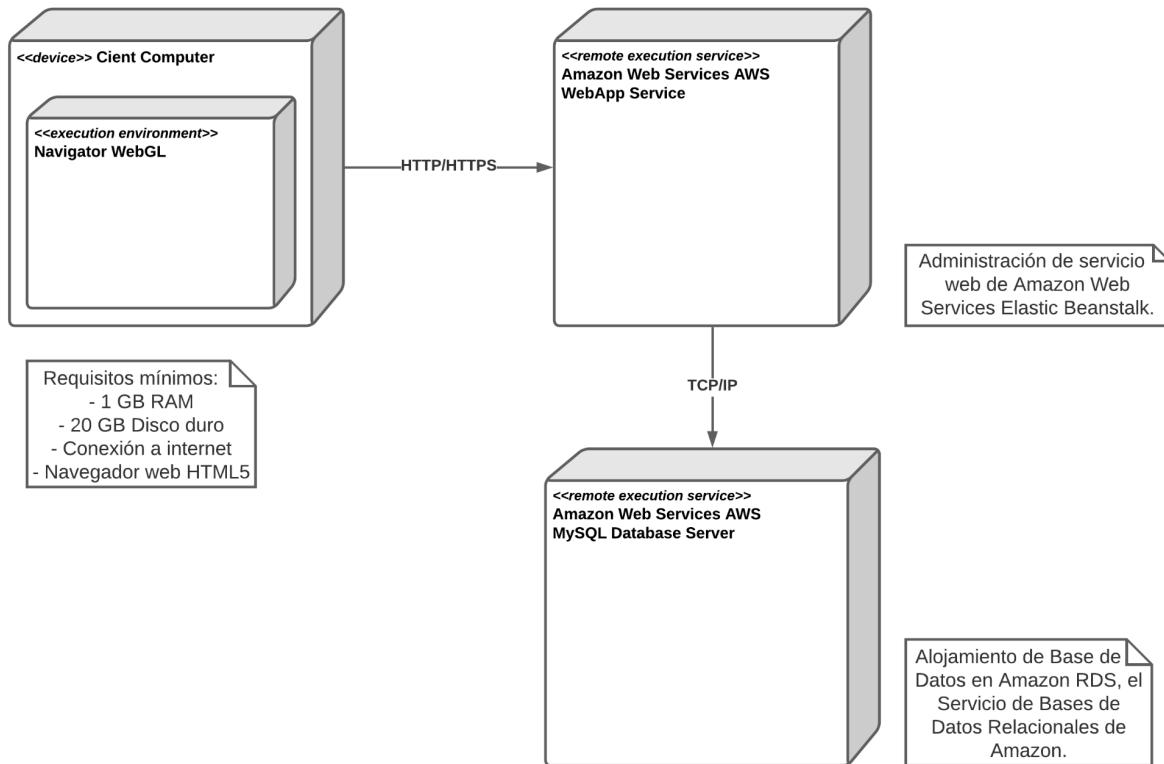




Física. Diagrama de despliegue.

Aulify BlackHole

Diagrama de despliegue

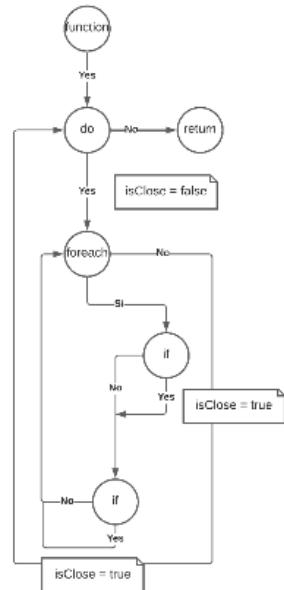


Pruebas dinámicas, caja blanca.

Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección. Son pequeños tests que validan el comportamiento de un objeto y la lógica. El sistema de caja blanca engloba este estilo de pruebas como los ejemplos que se muestran a continuación.

Prueba de Cobertura de Decisión 1.

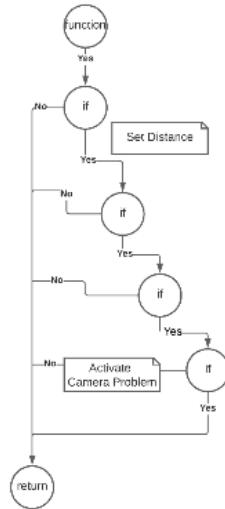
```
private Vector2 GenerateRandomPosition()
{
    Vector2 position;
    bool isClose;
    do
    {
        float x = UnityEngine.Random.Range(minX, maxX);
        float y = UnityEngine.Random.Range(minY, maxY);
        position = new Vector2(x, y);
        isClose = false;
        foreach (Vector2 pos in positions)
        {
            if (Vector2.Distance(pos, position) < spacing)
            {
                isClose = true;
                break;
            }
            if (Mathf.Abs(pos.x - position.x) < spacing && Mathf.Abs(pos.y - position.y) < spacing)
            {
                isClose = true;
                break;
            }
        }
    } while (isClose);
    return position;
}
```



En este primer caso se puede observar el grafo formado a partir de la generación aleatoria de una posición. La variable bandera lógica `isClose` determina el trayecto del flujo de ejecución de esta función y genera nuevas posibilidades de recorrido en posteriores ejecuciones de funciones.

Prueba de Cobertura de Decisión 2.

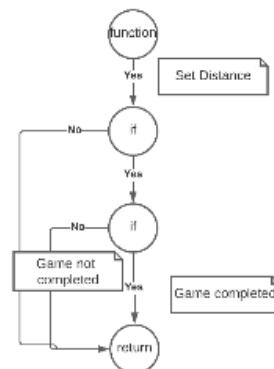
```
// Update is called once per frame
private void OnMouseDown()
{
    if(Time.timeSinceLevelLoad > 12){
        distance = Vector3.Distance(player.transform.position, this.gameObject.transform.position);
        if(completed == false){
            if(distance < minDistance){
                player.SetActive(false);
                if(cameraAnim != null)
                    Destroy(cameraAnim);
                cameraProblem.SetActive(true);
                StartCoroutine(OpenPanel());
            }
        }
    }
}
```



Este segundo caso está relacionado con el evento del ratón al momento de presionar alguna de las estaciones disponibles en el videojuego. En caso de que alguna situación ocurra, la función retornará excepciones que son manejadas por otras funciones del programa. Esto es de gran importancia porque es como se programa el post comportamiento del mouse después de su interacción con el software.

Prueba de Cobertura de Decisión 3.

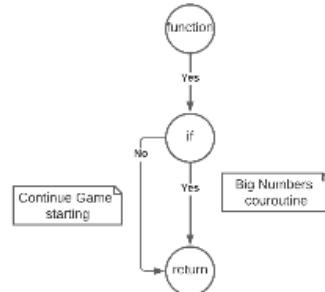
```
private void OnMouseDown()
{
    distance = Vector3.Distance(player.transform.position, transform.position);
    if(distance < minDistance){
        completed = counterScript.ReturnCounter();
        if(completed == 4){
            Debug.Log("You have completed the game");
            statsScript.GetStats();
            GoodEnding();
        }
        else{
            Debug.Log("You have not completed the game");
            errorSound.Play();
        }
    }
}
```



Del mismo modo, en la programación de nuestras clases se implementaron funciones que también sobrescriben el comportamiento de los eventos del ratón de computadora. En esta función, validamos la estación de trabajo final al momento de presionar el botón que aparece en la interfaz del videojuego.

Prueba de Cobertura de Decisión 4.

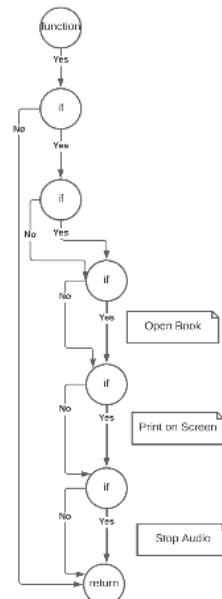
```
public void WrongAnswer(){
    if(textoTimer.rectTransform.localScale.x < 1.5f){
        StartCoroutine(BigNumbers());
    }
    wrongAnswers++;
    timer -= 20;
    BlackHoleGrows();
}
```



En el caso número 4 se puede observar el proceso requerido para solucionar la estación de trabajo no. 3 relacionada con la estructura y jerarquía de los dígitos de un número. En esta estación hacemos uso de otra función denominada BigNumbers() que permite utilizar números de tal envergadura.

Prueba de Cobertura de Decisión 5.

```
void Update()
{
    if(timerIsRunning == true){
        if(isOnePlayer == "True")
            clickControllerScript.openBook();
        time -= Time.deltaTime;
        minutes = (int)time / 60;
        _seconds = time % 60;
        milliseconds = ((int)(timer * 100) % 100);
        textoTimer.text = string.Format("{0}:{1}:{2}:{3}", _minutes, _seconds, _milliseconds);
        blackHoleGrowing();
        if(timer < 30)
            textoTimer.FontSharedMaterial.EnableGlow(ShaderUtilities.Keyboard_Glow);
    }
    if(timer < 13)
        if(timer % 1 < 0.1f)
            beepsound.Play();
    }
    if(timer < 0)
        statsScript.GetBadStats();
    timer = 0;
    if(backgroundMusic != null)
        backgroundMusic.GetComponent().Stop();
    }
    textoTimer.text = "00:00:00";
    cursorVisible = false;
    cursorLockState = CursorLockMode.Locked;
    finalStepsScript.Ending();
}
}
```



Por último, esta prueba resalta el papel del algoritmo para desplegar el libro dedicado al soporte de cada una de las estaciones. Hay instrucciones que se deben de realizar a la hora de consultarla, estas están relacionadas con otros elementos gráficos como el reloj. El grafo generado es el que se muestra en la parte superior.

Pruebas dinámicas, caja negra.

Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias y lo que prueban es que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupo. Un ejemplo de ellas es el sistema de caja negra.

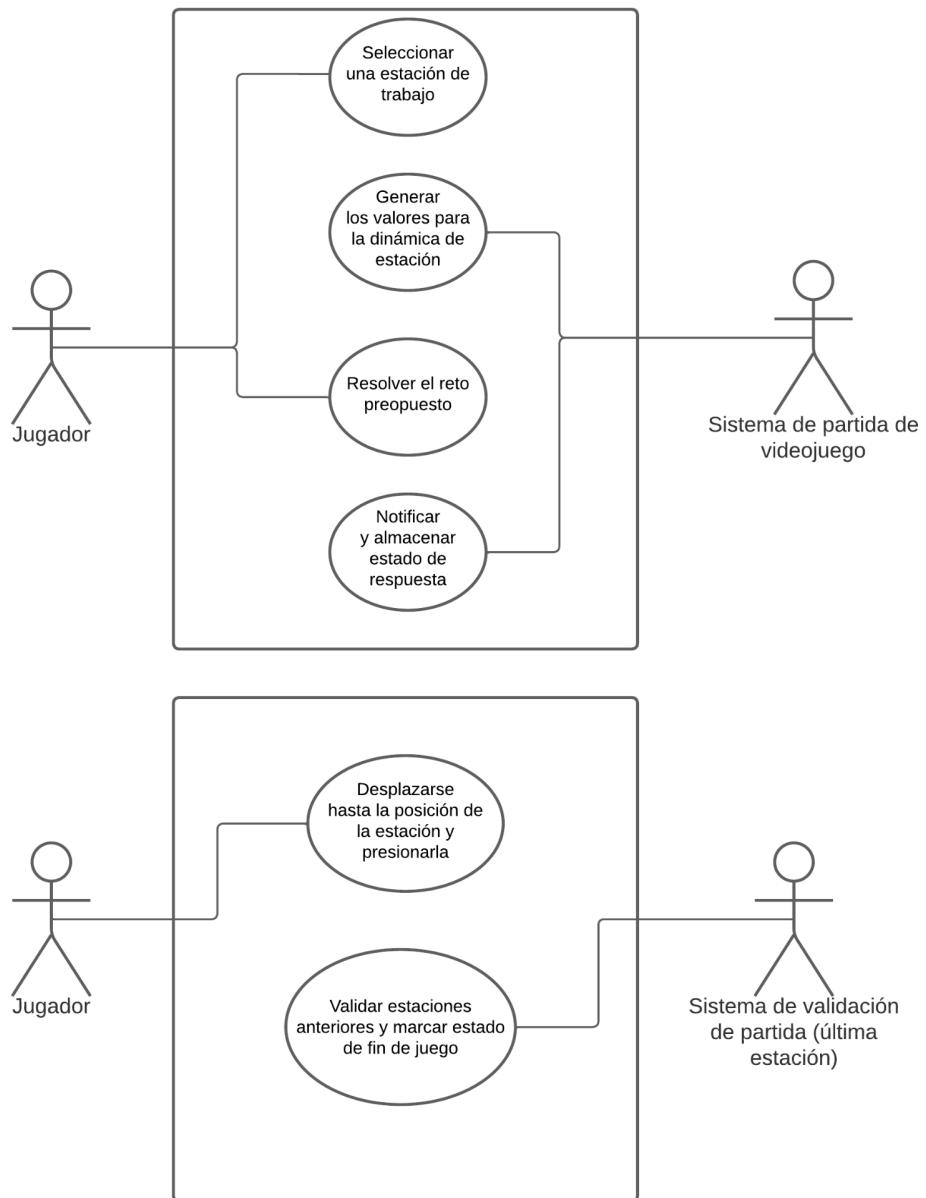
Cabe mencionar que el videojuego abarca cinco estaciones de retos en donde se aplican las matemáticas y la lógica con el fin de desarrollar dichas habilidades. De esta manera, presentamos **Prueba Partición de Equivalencia** con cada uno de las estaciones de trabajo presentadas en el software. Esta prueba permite validar los tipos de datos de entradas proporcionadas al videojuego con el objetivo de corroborar su ejecución.

| Variable | Clase de Equivalencia | Estado | Representante | Evidencia |
|----------------|---|---|-----------------|---|
| Estación no. 1 | EC1: $x \geq 0$ EC2: $x < 0$ EC3: no numérico | Válido No válido No válido | 9 |  |
| Estación no. 2 | EC4: $x \geq 0$ EC5: $x < 0$ EC6: no numérico | Válido No válido No válido | 8 5 |  |
| Estación no. 3 | EC7: $x \geq 0$ EC8: $x < 0$ EC9: no numérico | No válido No válido Válido | “Fraccion” |  |
| Estación no. 4 | EC10: $x \geq 0$ EC11: $x < 0$ EC12: no numérico | Válido No válido No válido | 12 |  |
| Estación no. 5 | EC13: $x \geq 0$ EC14: $x < 0$ EC15: no numérico EC15: evento de ratón | No válido No válido No válido Válido | Presionar botón |  |

Las estaciones de trabajo presentes en el videojuego son dinámicas totalmente independientes entre sí, esto significa que el usuario final puede resolver cada una de ellas libremente ya que no se encuentran relacionadas a nivel de construcción del videojuego, es decir, no hay ninguna jerarquía o requisito impuesto en ellas. Partiendo de lo dicho anteriormente, esta es la **Prueba Tabla de Decisión** para cada uno de las estaciones.

| | | | TC1 | TC2 | TC3 | TC4 |
|---------------------|--------------------------------|--|-----------------------|-----------------------|-----------------------|---------------------------|
| Precondición | Entrada a respuesta de usuario | Estación 1 Estación 2 Estación 3 Estación 4 Estación final | T T F T F | F F F F F | F F T F F | NA NA NA NA T |
| Actividad | Notificación en el videojuego | NA | T | T | T | T |

Por último, pero no menos importante, la implementación de pruebas de caso de uso nos permite conocer a nivel general cómo se supondría que el sistema debería reaccionar de forma normal frente alguna entrada o petición del usuario final. Debido a que nuestras estaciones de trabajo son muy similares en cuanto a la interacción del jugador, se agruparán bajo una **Prueba de Casos de Uso** que permita demostrar su comportamiento. Asimismo, la estación final requiere que todas las anteriores estaciones hayan sido resueltas por lo que también presentaremos su comportamiento en una prueba del mismo tipo.



KPI. Logros obtenidos durante las partidas jugadas.

Objetivo.

Resaltar las habilidades o áreas de oportunidad del jugador al momento de solucionar cada uno de los retos de las estaciones de trabajo en el videojuego Aulify BlackHole. Asimismo, compartir sus insignias obtenidas con los demás jugadores de la plataforma. Esta métrica permite conocer las habilidades y destrezas de los jugadores al momento de jugar el software.

Valor actual, punto de referencia y objetivos.

Desde la creación de su cuenta, el usuario comienza con una lista de logros vacía que, a partir del crecimiento del historial de juego, puede ir expandiendo para ser identificado en el panel de estadísticas. El panel de estadísticas presentará la selección de los mejores jugadores en distintas métricas. Esto tiene el objetivo de reconocer a los mejores jugadores involucrados en el videojuego.

Recopilación de datos.

Es posible recopilar datos de este KPI debido a la escritura de información que hace el sistema a nuestra base de datos.

El juego realiza dos peticiones y conexiones a la base de datos de Aulify BlackHole en cada partida, peticiones claves para conseguir la mencionada métrica. La primera petición HTTP retorna los logros disponibles para el usuario, mientras que la segunda petición se encarga de informar a la plataforma acerca de las estadísticas logradas durante el curso de las partidas, dependiendo de los parámetros incluidos (tiempo restante para fin de partida, respuestas incorrectas, modo de juego y módulos completados) se determina si el usuario es merecedor de alguna insignia para determinar algún logro.

Fórmula PERT.

1. Cálculo optimista (O): 10 logros
2. Cálculo más probable (M): 8 logros
3. Cálculo pesimista (P): 3 logros
4. Cálculo esperado (E): $(10 + 4(8) + 3)/6 = 7.5 \approx 8$ logros

KPI. Número de partidas jugadas.

Objetivo.

Conocer la frecuencia de jugabilidad de partidas de los jugadores más interesados en el videojuego Aulify BlackHole a partir del número de partidas jugadas con apoyo o sin apoyo de algún compañero.

Valor actual, punto de referencia y objetivos.

El conteo de las partidas jugadas por cada usuario es calculado automáticamente, esto significa que tras cada partida jugada el número final incrementará. El objetivo primordial de medir el número de partidas jugadas es también fomentar la competitividad de los jugadores más experimentados e incluidos en las respectivas tablas de líderes en el panel web de estadísticas del videojuego. Asimismo, remarcar sus demás variantes, como lo son: número total de partidas jugadas, número total de partidas ganadas, así como, menor número de equivocaciones entre los jugadores con mayores partidas.

Recopilación de datos.

Se considera como una partida ganada a aquella que se pueda completar en su totalidad junto con sus cuatro estaciones de trabajo que incluye, otro requisito es que el cronómetro que marca el tiempo restante no cese su operación con 00:00.

Es importante mencionar el papel que tienen las equivocaciones en el videojuego, ya que según el tipo de error que se cometa implicará en la disminución del tiempo restante, aunque hay dinámicas extra y ocultas a simple vista que pueden aumentar tiempo en el cronómetro para la partida actual del jugador.

Fórmula PERT.

1. Cálculo optimista (O): 20 partidas
2. Cálculo más probable (M): 10 partidas
3. Cálculo pesimista (P): 5 partidas
4. Cálculo esperado (E): $(20 + 4(10) + 5)/6 = 10.83 \approx 11$ partidas

Referencias de información

colaboradores de Wikipedia. (2021, 23 diciembre). *Modelo de Vistas de Arquitectura 4+1*. Wikipedia, la enciclopedia libre.

https://es.wikipedia.org/wiki/Modelo_de_Vistas_de_Arquitectura_4+1

Maze Found | Read the Docs. (s. f.).

<https://proyecto-semestral.readthedocs.io/en/latest/6+-+Design.html>

Team, K. (2023, April 11). ¿Qué son las pruebas de caja blanca? KeepCoding Bootcamps.

<https://keepcoding.io/blog/que-son-pruebas-de-caja-blanca/>

Latam, D. (2021, April 15). 15 métodos de prueba que todos los desarrolladores deben conocer - DevOps Latam. DevOps Latam.

<https://devopslatam.com/15-metodos-de-prueba-que-todos-los-desarrolladores-deben-conocer/>

Team, K. (2022, June 3). ¿Qué son las pruebas de caja negra? | KeepCoding Bootcamps. *KeepCoding Bootcamps*. <https://keepcoding.io/blog/que-son-las-pruebas-de-caja-negra/>

Mkt. (2020). En el Software Testing: Qué Son Las Pruebas De Caja Negra. *Trans-TI*.

<https://trans-ti.com/2020/12/30/en-el-software-testing-que-son-las-pruebas-de-caja-negra/>