S.I.G.P.D.

Programación Full Stack

CodeFlow

Rol	Apellido	Nombre	C.I	Email
Coordinador	Manzinalli	Felipe	5.714.301-9	felipe.manzinalli@estudiante.ceibal.edu.uy
Sub-Coordinador	Suarez	Pablo	5.656.677-9	psuarez@estudiante.ceibal.edu.uy
Integrante 1	Maldonado	Agustin	5.617.059-8	ezequiel.maldonado@estudiante.ceibal.edu.uy
Integrante 2	Diez	Daniel	5.547.065-6	danieldiez05@gmail.com

Docente: Pérez, Joaquín

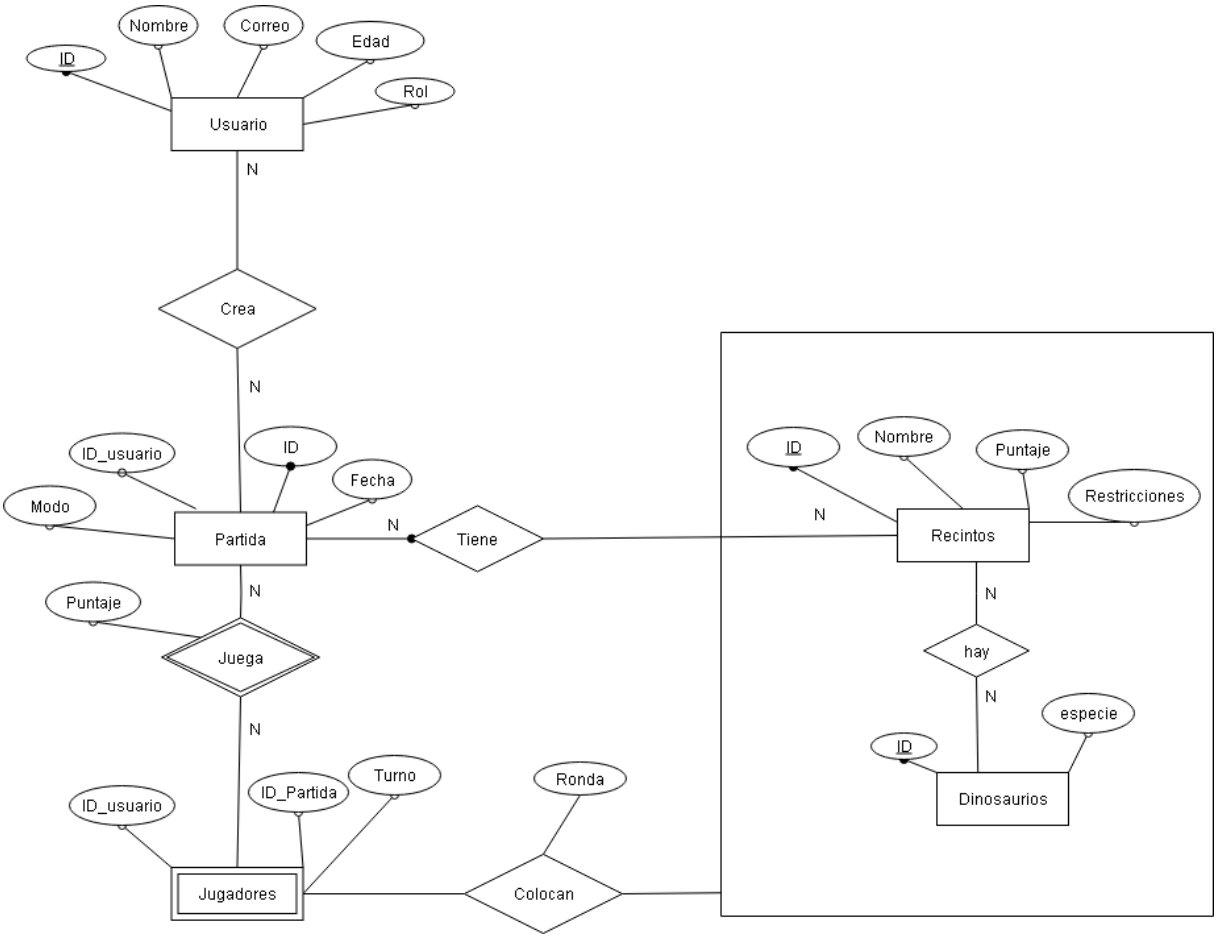
Fecha de culminación

15/9/2025

SEGUNDA ENTREGA

ÍNDICE

Diagrama Entidad Relación



Restricciones de Negocio y de Integridad (RNE):

Restricciones de los Recintos:

- Bosque de la semejanza: sólo dinosaurios de la misma especie, llenar de izquierda a derecha sin huecos.
- Prado de la Diferencia: sólo especies distintas, llenar de izquierda a derecha sin huecos.
- Pradera del Amor: cualquier especie, 5 puntos por pareja de la misma especie.
- Trío Frondoso: máximo 3 dinosaurios, 7 puntos si hay exactamente 3.
- Rey de la Selva: solo 1 dinosaurio, 7 puntos si nadie tiene más de esa especie.
- Isla Solitaria: solo 1 dinosaurio, 7 puntos si es único de su especie en el parque.

Pasaje a tablas del diagrama

Entidades:

Usuario (ID, Nombre, Correo, Edad)

Partida (ID, ID_Usuario, Modo, Fecha)

Recintos (ID, Nombre, Puntaje, Restricciones)

Dinosaurios (ID, Especie)

Entidades Débiles:

Jugadores (ID_usuario, ID_partida, turno)

Relaciones:

Crea (ID_usuario, ID,partida)

Colocan (Iid_usuariom, id_partida, turno, id_recinto, id_dino, ronda)

Hay (ID_recinto, ID_dinosaurio)

Tiene (ID_partida, ID_recinto)

PK FK

Identificación de Claves Foráneas (FK):

Crea FK (ID_usuario) referencia a usuario (ID)

Crea FK (ID_partida) referencia a Partida (ID)

Colocan FK (ID_usuario) referencia a Usuario (ID)

Colocan FK (ID_partida) referencia a Partida (ID)

Colocan FK (Turno) referencia a Jugadores (turno)

Colocan FK (id_recinto) referencia Recinto (ID)

Colocan FK (id_dino) referencia a Dinosaurio (ID)

Hay FK (ID_recinto) referencia a Recinto (ID)

Hay FK (ID_dinosaurio) referencia a Dinosaurio (ID)

Tiene FK (ID_partida) referencia a Partida (ID)

Tiene FK (ID_recinto) referencia a Recinto (ID)

Primera forma normal 1FN

No se permiten atributos multivaluados ni compuestos. Todo debe ser atómico

Todos los atributos son atómicos (ej: se guarda “especies = TRex, Velociraptor” en un mismo campo). Cumple 1FN.

Segunda Forma normal 2FN

Estar en 1FN y no tener dependencias parciales de la calve primara

No hay dependencias parciales

Tercera forma normal 3FN

Estar en 2FN y no tener dependencias transitivas (atributos no clave que dependan de otros atributos no clave).

No hay dependencias transitivas.

DDL de la base de datos en MySQL

Crear la base de datos:

```
Create DATABASE draftosaurus_bd;

USE draftosaurus_bd;


CREATE TABLE Usuario

(

	ID INT AUTO_INCREMENT PRIMARY KEY,

	Nombre VARCHAR(100) UNIQUE NOT NULL,

	Password VARCHAR(255) NOT NULL,

	Edad INT NOT NULL

);

CREATE TABLE Partida

(

	ID INT AUTO_INCREMENT PRIMARY KEY,

	ID_usuario INT NOT NULL,

	Modo VARCHAR(50) NOT NULL,

	Fecha DATETIME NOT NULL,

	FOREIGN KEY (ID_usuario) REFERENCES Usuario(ID)

);
```

```
CREATE TABLE Recintos (
        ID INT AUTO_INCREMENT PRIMARY KEY,
        Nombre VARCHAR(100) NOT NULL,
        Puntaje INT DEFAULT 0,
        Restricciones TEXT
);
CREATE TABLE Dinosaurios
(
        ID INT AUTO_INCREMENT PRIMARY KEY,
        Especie VARCHAR(100) NOT NULL
);

CREATE TABLE Jugadores
(
        ID_usuario INT NOT NULL,
        ID_partida INT NOT NULL,
        Turno INT NOT NULL,
        PRIMARY KEY (ID_usuario, ID_partida),
        FOREIGN KEY (ID_usuario) REFERENCES Usuario(ID),
        FOREIGN KEY (ID_partida) REFERENCES Partida(ID)
);
CREATE TABLE Crea
```

```
(
    ID_usuario INT NOT NULL,
    ID_partida INT NOT NULL,
    PRIMARY KEY (ID_usuario, ID_partida),
    FOREIGN KEY (ID_usuario) REFERENCES Usuario(ID),
    FOREIGN KEY (ID_partida) REFERENCES Partida(ID)
);
CREATE TABLE Colocan
(
    ID_usuario INT NOT NULL,
    ID_partida INT NOT NULL,
    Turno INT NOT NULL,
    ID_recinto INT NOT NULL,
    ID_dino INT NOT NULL,
    Ronda INT NOT NULL,
    PRIMARY KEY (ID_usuario, ID_partida, ID_recinto, ID_dino, Ronda),
    FOREIGN KEY (ID_usuario) REFERENCES Usuario(ID),
    FOREIGN KEY (ID_partida) REFERENCES Partida(ID),
    FOREIGN KEY (ID_recinto) REFERENCES Recintos(ID)

);
```

```
CREATE TABLE Hay (
        ID_recinto INT NOT NULL,
        ID_dinosaurio INT NOT NULL,
        PRIMARY KEY (ID_recinto, ID_dinosaurio),
        FOREIGN KEY (ID_recinto) REFERENCES Recintos (ID),
        FOREIGN KEY (ID_dinosaurio) REFERENCES Dinosaurios(ID)
);
CREATE TABLE Tiene (
        ID_partida INT NOT NULL,
        ID_recinto INT NOT NULL,
        PRIMARY KEY (ID_partida, ID_recinto),
        FOREIGN KEY (ID_partida) REFERENCES Partida(ID),
        FOREIGN KEY (ID_recinto) REFERENCES Recintos(ID)
);
```

Implementación de TODOS los recintos en PHP

Aclaraciones:

- Todas las funciones que implementan la lógica de los recintos en PHP están diseñadas para recibir como parámetros la cantidad de dinosaurios que permite a cada recinto alcanzar la máxima cantidad de puntos posibles, según las reglas del juego..

Existen excepciones en los recintos donde no hay un límite de dinosaurios (por ejemplo, Pradera del Amor), en cuyo caso se puede colocar cualquier cantidad de dinosaurios y la función calcula los puntos de acuerdo a las parejas que se formen, respetando la regla de puntaje del recinto.

- En todos los demás recintos, los parámetros reflejan la cantidad exacta de dinosaurios que permite obtener el puntaje máximo, asegurando que la simulación muestre correctamente la lógica de cada recinto.
- La regla que dice que "Al final de la partida, cada recinto que contenga al menos un T-Rex debería otorgar 1 punto de victoria adicional" **no está implementada** en las funciones actuales. Por lo tanto, aunque un recinto contenga uno o más T-Rex, la lógica actual **no suma ningún punto adicional** por esta condición; únicamente se calculan los puntos según la mecánica normal de cada recinto.

Trío Frondoso:

```php
private function TrioFrondoso($dino1 = "amarillo", $dino2 = "celeste", $dino3 = "rosado") {
    $puntos = 0;
    $dinosaurios = array_filter([$dino1, $dino2, $dino3], fn($d) => !empty($d));
    if (count($dinosaurios) > 3) return 0;
    if (count($dinosaurios) === 3) $puntos = 7;
    return $puntos;
}
```

Esta función verifica los dinosaurios colocados en el recinto (hasta 3). Si hay exactamente 3 dinosaurios, sin importar su especie, se otorgan 7 puntos. Si hay menos de 3 o si se intentara colocar más de 3, no se otorga ningún punto. En otras palabras, el recinto solo premia cuando está completo con 3 dinosaurios y no más.

Bosque de la Semejanza:

```php
private function BosqueSemejanza($dinos = ["rojo", "rojo", "rojo", "rojo", "rojo", "rojo"]) {
    $puntos = 0;
    if (count($dinos) > 6) $dinos = array_slice($dinos, 0, 6);
    if (empty($dinos)) return $puntos;
    foreach ($dinos as $dino) if (empty($dino)) return 0;
    $primeraEspecie = $dinos[0];
    foreach ($dinos as $dino) if ($dino !== $primeraEspecie) return 0;
    switch (count($dinos)) {
        case 1: $puntos = 2; break;
        case 2: $puntos = 4; break;
        case 3: $puntos = 8; break;
        case 4: $puntos = 12; break;
        case 5: $puntos = 18; break;
        case 6: $puntos = 24; break;
    }
    return $puntos;
}
```

Esta función recibe hasta 6 dinosaurios colocados de izquierda a derecha. Todos deben ser de la misma especie (mismo color) y no puede haber huecos entre ellos. Según la cantidad de dinosaurios correctamente colocados, se otorgan puntos: 2 puntos si hay 1, 4 si hay 2, 8 si hay 3, 12 si hay 4, 18 si hay 5 y 24 si hay 6. Si hay un hueco entre medio, especies diferentes o más de 6 dinosaurios, el recinto no suma puntos.

Prado de la Diferencia:

```php
private function PradoDiferencia($dinos = ["rojo", "amarillo", "celeste", "verde", "rosado",
"naranja"]) {
    $puntos = 0;
    if (count($dinos) > 6) $dinos = array_slice($dinos, 0, 6);
    if (empty($dinos)) return $puntos;
    foreach($dinos as $dino) if (empty($dino)) return 0;
    $especiesVistas = [];
    foreach ($dinos as $dino) {
        if (in_array($dino, $especiesVistas)) return 0;
        $especiesVistas[] = $dino;
    }
    switch (count($dinos)) {
        case 1: $puntos = 1; break;
        case 2: $puntos = 3; break;
        case 3: $puntos = 6; break;
        case 4: $puntos = 10; break;
        case 5: $puntos = 15; break;
        case 6: $puntos = 21; break;
    }
    return $puntos;
}
```

Esta función también recibe hasta 6 dinosaurios colocados de izquierda a derecha. Cada dinosaurio debe ser de una especie distinta y no puede haber huecos intermedios. Según la cantidad de dinosaurios válidos, se otorgan puntos: 1 punto si hay 1, 3 si hay 2, 6 si hay 3, 10 si hay 4, 15 si hay

5 y 21 si hay 6. Si se repite alguna especie, hay huecos entre medio o se colocan más de 6, el recinto no suma puntos.

La Pradera del Amor:

```php
private function PraderaAmor($dinos = ["rojo", "rojo"]) {
    $puntos = 0;
    if (empty($dinos)) return $puntos;
    $conteoEspecies = array_count_values($dinos);
    foreach ($conteoEspecies as $cantidad) {
        $parejas = intdiv($cantidad, 2);
        $puntos += $parejas * 5;
    }
    return $puntos;
}
```

La función recibe un array con los dinosaurios colocados. Luego cuenta cuántos hay de cada especie y calcula cuántas parejas completas se pueden formar. Por cada pareja se suman 5 puntos. Los dinosaurios que no forman pareja no suman puntos, y no importa cuántos dinosaurios haya en total.

La Isla Solitaria:

```php
private function IslaSolitaria($dino = "rojo", $parque = []) {
    if (empty($dino)) return 0;
    $conteo = 0;
    foreach ($parque as $d) if ($d === $dino) $conteo++;
    return ($conteo === 1) ? 7 : 0;
}
```

Esta función verifica si un dinosaurio colocado en el recinto Isla Solitaria es el único de su especie dentro de todo el parque del jugador. Por eso, además del dinosaurio colocado en el recinto, recibe un array $parque que representa todos los dinosaurios que el jugador ya tiene en su parque. Primero, si no se colocó ningún dinosaurio en el recinto, se devuelve 0 puntos. Luego recorre el parque contando cuántos dinosaurios hay de la misma especie que el colocado en el recinto. Finalmente, si solo hay uno de esa especie (es decir, el colocado en el recinto es único), otorga 7 puntos; si hay más de uno, devuelve 0 puntos.

El Rey de la Selva:

```php
private function ReyDeLaSelva($dino = "", $parques = []) {
    if (empty($dino)) return 0;
    $miConteo = 0;
    foreach ($parques[0] as $d) if ($d === $dino) $miConteo++;
    for ($i = 1; $i < count($parques); $i++) {
        $conteoOponente = 0;
        foreach ($parques[$i] as $d) if ($d === $dino) $conteoOponente++;
        if ($conteoOponente > $miConteo) return 0;
    }
    return 7;
}
```

Esta función verifica si el dinosaurio colocado en el recinto “El Rey de la Selva” es el que más tenés de su especie en comparación con los demás jugadores. Para eso recibe como parámetro un arreglo llamado “parques”, donde cada elemento representa el parque de cada jugador (siendo el primer elemento el parque tuyo). Primero cuenta cuántos dinosaurios de esa especie tiene vos, después recorre los parques de los demás jugadores comparando cuántos tienen de la misma especie. Si algún jugador tiene más, la función devuelve 0 puntos, si nadie tiene más o hay empate, devuelve 7 puntos. En otras palabras, $parques sirve para comprobar la cantidad de dinosaurios de la misma especie de todos los jugadores y determinar si eres el que más tiene o no.

El río:

```php
private function Rio($dinos = ["rojo"]) {
    if (empty($dinos)) return 0;
    return count($dinos);
}
```

La función recibe un array con todos los dinosaurios que están en la zona del Río. Cada dinosaurio suma 1 punto, sin importar su especie. Si no hay dinosaurios, devuelve 0 puntos. Simplemente cuenta cuántos dinosaurios hay y devuelve ese número como puntaje.

Puntaje Total:

```php
    private function PuntajeTotal(
        $trioFrondoso = ["", "", ""],
        $bosqueSemejanza = ["", "", "", "", "", ""],
        $pradoDiferencia = ["", "", "", "", "", ""],
        $praderaAmor = [],
        $islaSolitaria = "",
        $parqueIslaSolitaria = [],
        $reySelva = "",
        $parquesReySelva = [],
        $rio = []
    ) {
        $total = 0;
        $total += $this->TrioFrondoso(...$trioFrondoso);
        $total += $this->BosqueSemejanza($bosqueSemejanza);
        $total += $this->PradoDiferencia($pradoDiferencia);
        $total += $this->PraderaAmor($praderaAmor);
        $total += $this->IslaSolitaria($islaSolitaria, $parqueIslaSolitaria);
        $total += $this->ReyDeLaSelva($reySelva, $parquesReySelva);
        $total += $this->Rio($rio);
        return $total;
    }
}
?>
```

Esta función privada se encarga de sumar todos los puntos obtenidos por los distintos recintos y zonas del juego para un jugador en una partida. Recibe como parámetros los valores correspondientes a cada recinto o zona: los dinosaurios colocados en Trío Frondoso, Bosque de la Semejanza, Prado de la Diferencia, Pradera del Amor, Isla Solitaria, Rey de la Selva y Río, así como los parques necesarios para calcular la puntuación de ciertos recintos. Para cada recinto, llama a la función específica que calcula sus puntos y acumula el total. Al final, devuelve la suma total de puntos obtenidos por el jugador en todos los recintos y zonas.

Definición de todos los modelos en PHP

Se crearon los siguientes modelos:

- ColocanModel.php
- Database.php
- DinosaurioModel.php
- JuegaModel.php
- JugadorModel.php
- PartidaModel.php
- RecintoModel.php
- Usermodel.php

Cada uno con sus respectivos atributos, get y set, y constructores.
Para ver el contenido de cada uno se puede acceder a través de la carpeta "Proyecto" y a través del enlace al repositorio de GitHub.

Controladores para cada modelo

Por otra parte se crearon los controladores para cada modelo a los cuales se puede acceder a cada uno también a través de la carpeta "Proyecto" y a través del enlace al repositorio de GitHub.

Creación de vistas y su navegabilidad

Se crearon las siguientes vistas:

- index.html: es la página principal del sitio la cual explica en que consiste el juego y le permite al usuario elegir entre jugar al juego completo o modo seguimiento.
- login.html: Formulario para que el usuario ingrese su nombre y contraseña para iniciar sesión.
- signup.html: Formulario para que un nuevo usuario se registre, indicando nombre, edad y contraseña
- contacto.html: página de contacto para ver elementos vinculados a la empresa CodeFlow

- cuantosjugadores.html: Pantalla para seleccionar o indicar la cantidad de jugadores antes de iniciar una partida
- incorrecta.html: Mensaje que muestra cuando el usuario ingresa credenciales incorrectas al iniciar sesión
- inicioexitoso.html: Pantalla de confirmación que indica que el usuario ha iniciado sesión correctamente
- juego.html: Interfaz principal del juego digital, donde se desarollan las partidas y se colocan dinosaurios en los recintos
- noencontrado.html: Mensaje que aparece si el usuario intenta iniciar sesión con un usuario inexistente
- registroexitoso.html: Pantalla de confirmación que indica que el usuario ha iniciado sesión correctamente.
- usuarioyaexiste.html: Mensaje que muestra cuando se intenta registrar un usuario con un nombre que ya existe en la base de datos.

A todas las vistas se puede acceder a través de la carpeta del proyecto y a través del repositorio de GitHUB.

Repositorio de Github:

Puedes acceder al repositorio a través del siguiente enlace, el mismo está organizado por una carpeta por cada entrega. https://github.com/codeflow838/CodeFlow