S.I.G.P.D.

Programación FullStack

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: Perez, Joaquin

Fecha de culminación

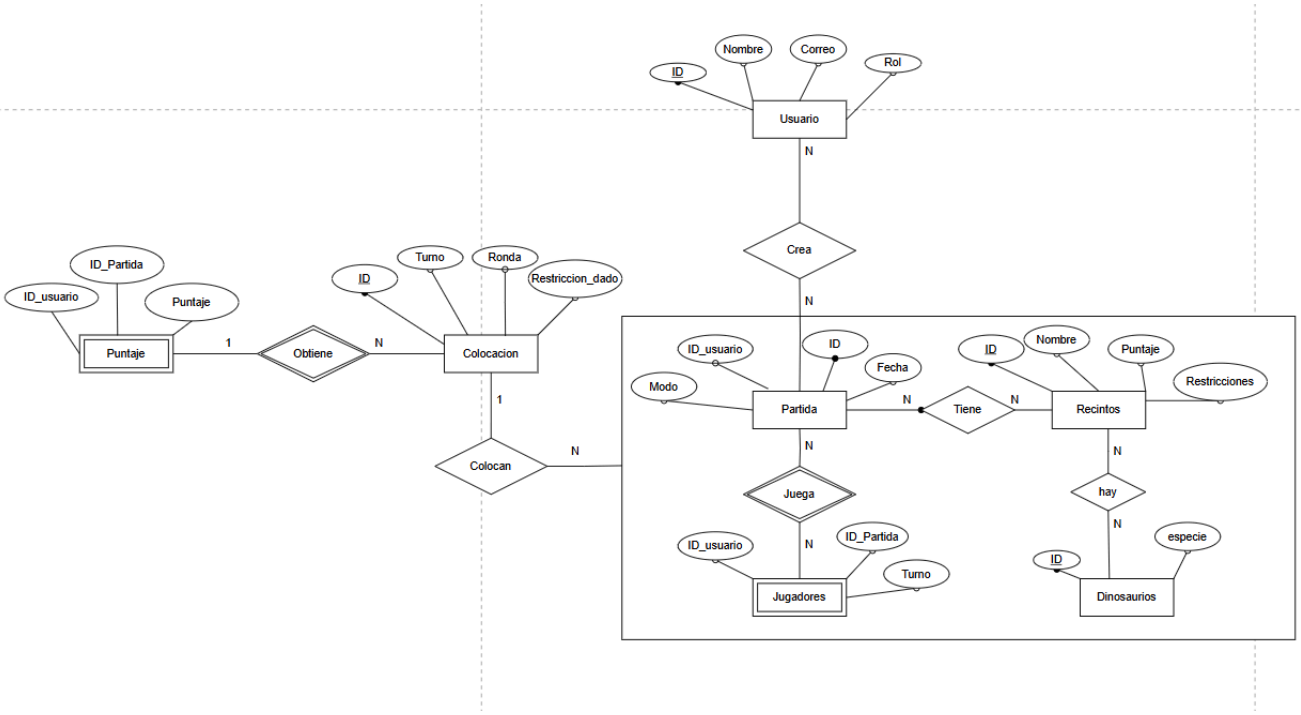
14/7/2025

PRIMERA ENTREGA

ITI CETP **3MD**

ÍNDICE

Diagrama Entidad-Relacion (MER)



Pasaje a tablas sin normalizar

Entidades fuertes

Usuario (ID, Nombre, Correo, Rol)

Partida (ID, ID_usuario, Modo, Fecha)

Recintos (ID, Nombre, Puntaje, Restricciones)

Dinosaurios (ID, Especie)

Colocacion (ID, Turno, Ronda, Restriccion_dado)

Entidades débiles

Jugadores (ID_usuario, ID_partida. turno)

Puntaje (Puntaje, ID_partida, ID_usuario)

Relaciones

Crea (ID_usuario, ID_partida)

Colocan (ID_Colocacion, ID_usuario, ID_recinto,ID_dinosaurio)

Hay (ID_recinto, ID_dinosaurio)

Tiene (ID_partida, ID_recinto)

PK FK

Identificación de claves Foráneas

Crea FK (ID_ususario) referencia a usuario (ID)

Crea FK (ID_partida) referencia a Partida (ID)

Colocan FK (ID_colocacion) referencia a Colocacion (ID)

Colocan FK (ID_usuario) referencia a Usuario (ID)

Colocan FK (ID_recinto) referencia a Recinto (ID)

Colocan FK (ID_dinosaurio) 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)

Definición de arquitectura MVC (Modelo Vista Controlador)

¿Qué es MVC?

MVC significa Modelo Vista Controlador, es una forma de organizar un sistema separando las partes que hacen cosas distintas

Modelo:

Es lo que maneja los datos, las reglas del juego y lo que se guarda en la base de datos.

Vista:

Es lo que ve el usuario, como las pantallas, botones y formularios.

Controlador:

Es lo que se encarga de recibir lo que hace el usuario y decidir qué tiene que pasar (por ejemplo, si puede colocar un dinosaurio o no).

¿Para qué es útil usar MVC en nuestro proyecto?

En nuestro proyecto usamos PHP, HTML y JavaScript para hacer una aplicación web. Como tiene muchas partes (login, juego, puntuación, configuración, etc.) usar MVC nos ayuda a tener todo ordenado.Separar las cosas en Modelo, Vista y Controlador hace que el código sea más fácil de comprender, de cambiar si es necesario y también permite que el diseño visual no se mezcle con la lógica del juego.

Aplicamos el patrón MVC

Modelo

Los modelos son responsables de interactuar con la base de datos en nuestro proyecto, incluye:

- Clases PHP que gestionan usuarios, partidas, dinosaurios y recintos.
- Funciones para registrar colocaciones y calcular puntuaciones.

Vista

Las vistas están conformadas por archivos del tipo .php, .html, .css y JavaScript que muestran la información al usuario:

- login
- Interfaz del juego (modo digital)
- Formulario de carga manual (modo seguimiento)
- Rankings y resultados
- Configuración de partida

Controlador

Los controladores accionan como un intermedio entre las vistas y los modelos:

- Reciben solicitudes (como colocar un dinosaurio o iniciar una partida).
- Ejecutan la lógica del juego y validan restricciones.
- Llaman a métodos del modelo para guardar o cargar datos.
- Determinan qué vista va a visualizar el usuario.

Definición de Modelos

En esta parte vamos a detallar los modelos que están involucrados en el sistema.Cada modelo representa una pieza clave del juego y de la gestión de los datos de usuarios, los modelos que utilizaremos serán:

Modelo usuario:

Descripción:Representa a una persona que usa el sistema. Puede ser un jugador o un administrador.

Atributos: ID (identificador único), Nombre, Correo, Contraseña y Rol (jugador o administrador)

Relaciones: un usuario puede crear varias partidas

Modelo: Partida

Descripción: Representa una sesión del juego Draftosaurus, ya sea en modo juego digital o modo seguimiento.

Atributos: ID, ID_usuario (creador de la partida), Modo (seguimiento o digital) y Fecha de creación.

Relaciones: Una partida puede tener varios jugadores y puede tener varios recintos

Modelo: Dinosaurio

Descripción: Representa un dinosaurio que puede colocarse en un recinto durante la partida.

Atributos: ID y Especie.

Relaciones: Un dinosaurio se coloca en un recinto

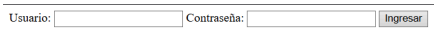
Modelo: Colocación

Descripción: Representa la acción de colocar un dinosaurio en un recinto durante una partida, por parte de un jugador.

Atributos: ID, Turno, Ronda, Restricción_dado, ID_usuario, ID_partida, ID_dino, ID_recinto.

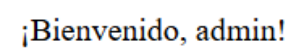
Relaciones: Está relacionada con Usuario, Partida, Dinosaurio y Recinto.

Log in funcional contra base de datos



En el login no diseñamos una interfaz pero es funcional contra la base de datos ya que genera una consulta hacia la base de datos para que sea necesario ingresar. Solo permite el ingreso a el

usuario “admin” con contraseña “admin123” en el caso de no poder validar estas credenciales no permitirá el ingreso avisando que o la contraseña o el usuario son incorrectos



LINK DEL REPOSITORIO DE GITHUB

https://github.com/codeflow838/CodeFlow.git