

S.I.G.P.D.

Programación Full-Stack

CEITAROX

Rol	Apellido	Nombre	C.I	Email
Coordinador	Rodríguez	Piero	5.592.003-7	rpierina286@gmail.com
Sub-Coordinador	Gracés	Tatiana	5.713.652-9	tatianatareasutu@gmail.com
Integrante 1	Caballero	Juancruz	5.714.462-1	zuzuonetwothree@gmail.com
Integrante 2	Cardozo	Celeste	5.716.176-6	celestecardozo1803@gmail.com
Integrante 3	Vázquez	Ignacio	5.479.448-7	nachosfera14@gmail.com

Docente: Romero, Carlos

**Fecha de
culminación
14/07/2025**

PRIMERA ENTREGA



Índice

1. Desarrollo del proyecto
2. Justificación tecnológica
 - 2.1 Presentación y justificación de las tecnologías utilizadas en el proyecto
 - 2.2 Análisis de PHP 8.2 y ventajas frente a otras alternativas
3. Prototipo de interfaz de usuario
 - 3.1 Pantallas UI/UX
 - 3.2 Diseño desde Bootstrap
 - 3.3 Diseño desde CSS
4. Modelado de datos
 - 4.1 Diagrama Entidad-Relación (DE-R)
 - 4.2 Esquematización
 - 4.3 Normalización
5. Repositorio
6. Explicación del código



1. Desarrollo del proyecto

En esta carpeta se expondrá nuestro proyecto de egreso, con el nombre de CEITAROX presentamos un sistema de seguimiento y gestión para el juego de mesa de tipo "draft" de nombre Draftosaurus de manera online, el cual hemos bautizado como FlowerDraft, donde ganará el jugador con la mejor florería.

En esta carpeta se mostrará el primer avance del prototipo web de la aplicación, el modelado de la base de datos, las pantallas UI/UX, el acceso al repositorio y la justificación tecnológica de los recursos a utilizar para el resto del proyecto.



2. Justificación tecnológica

2.1 Presentación y justificación de las tecnologías utilizadas en el proyecto

Para nuestro proyecto utilizamos los siguientes recursos tecnológicos:

- HTML 5
- CSS 3
- MariaDB
- Bootstrap 5.3
- JavaScript
- PHP 8.2
- Figma
- Visual Studio Code

HTML 5: Elegimos esta versión ya que es la más optimizada y en funcionamiento en la actualidad, por ello todos los diseñadores y programadores utilizan esta versión de HTML. Permite diversas tecnologías que con versiones de HTML posteriores a 2014, pudiendo aplicar varios lenguajes y frameworks a nuestro trabajo (en nuestro caso Bootstrap, JavaScript y PHP).

Bootstrap 5.3: El framework permite adaptar el diseño del sitio web a dispositivos móviles y nos facilita el trabajo en su elaboración.

CSS 3: Bootstrap tiene limitaciones en su uso por lo que es necesario utilizar esta herramienta para solucionar las restricciones de diseño.

MariaDB: Utilizaremos este software para la gestión de la base de datos, utilizando el lenguaje SQL. Es un software libre, confiable y compatible con MySQL, además de que incluye características que este otro software no contiene. El rendimiento y la seguridad de MariaDB hemos encontrado que se considera mejor que el de MySQL.



JavaScript: Permite manipular y programar etiquetas HTML mediante la Programación Orientadas a Objetos (POO) y le atribuye mayor funcionalidad al sistema back-end y front-end de la plataforma.



Figma: Plataforma web que autoriza el trabajo colaborativo, maneja bocetos vectoriales y tiene herramientas que facilitan el maquetado del sitio a diferentes escalas de pantalla.

Visual Studio Code: Es un entorno muy familiarizado para nosotros, cómodo e intuitivo. Permite trabajar con varios lenguajes, como a su vez, contiene una terminal para trabajar con repositorios en Git.

2.2 Análisis de PHP 8.2 y ventajas frente a otras alternativas

Hemos optado por utilizar PHP para este proyecto al encontrar múltiples ventajas frente a otras alternativas, principalmente por su simplicidad y la facilidad de uso. Encontramos mejor debido también a su compatibilidad con la mayoría de proveedores de hosting, especialmente los económicos, junto con su compatibilidad con frameworks modernos, además de la no necesidad de configuraciones complejas, reiterando su facilidad de uso. Tiene constantes mejoras en su seguridad y calidad, y al ser una herramienta que lleva más de dos décadas en la web, consideramos que PHP es altamente confiable.

3. Prototipo de interfaz de usuario

Hemos realizado una maquetación utilizando HTML5, CSS3 y Bootstrap. Para esta primera entrega hemos omitido la interfaz para PC (1920px - 1080px) y solo nos hemos enfocado en la interfaz para dispositivos móviles (320px - 576px). Recomendamos que debido a esto, una vez ejecutado el archivo .html principal “Inicio”, se cambie el tipo de pantalla a una de iPhone o Android en la sección “Inspeccionar” para observar la maquetación.

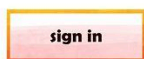
Para la UI del sistema, hemos optado por hacer un diseño basado en elementos del tablero y en la estética de este para generar una interfaz amigable y colorida. Cada elemento fue ilustrado digitalmente en Paint Tool Sai 2.



3.1 Pantallas UI/UX

Para esta primera entrega nos hemos centrado en realizar la interfaz para dispositivos móviles, sin embargo todos los assets han sido reutilizados y adaptados para PC.

Hemos organizado cada elemento mediante bosquejos desde Paint Tool Sai 2, luego pasamos todos los elementos en .png a Figma para el resultado final.



Pantalla de inicio

Elementos:

Botones:

- btn Sign in: para registrarse por primera vez
- Btn Cantidad de jugadores: En cada casilla verde se colocarán los números 2, 3, 4, 5 (Pantalla/s: Inicio jugadores)

Presentación del sistema con el título FlowerDraft, en la parte inferior se encontrará un footer con información sobre SIGPD y CEITAROX, junto con un link al correo de CEITAROX y al repositorio de Github.

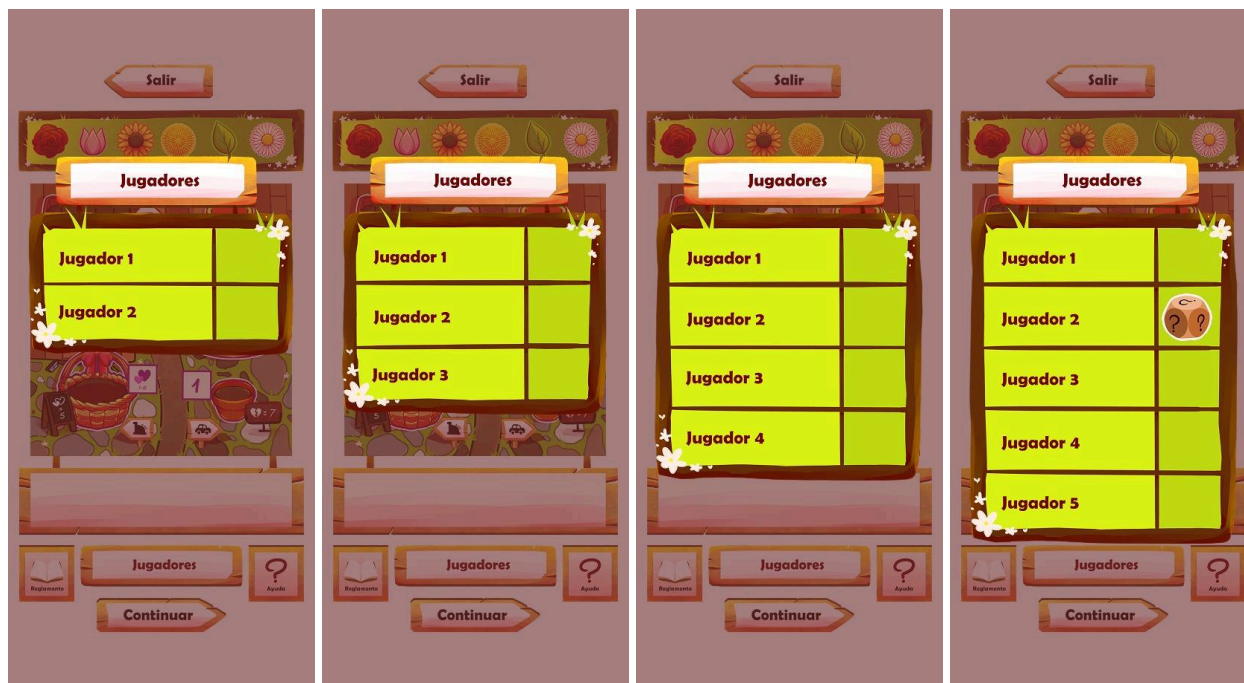


Pantalla de juego

Elementos:

- Btn salir: Para regresar al menu de inicio (Pantalla: Pantalla de Inicio)
- Inventario de fichas: Próximamente interactivo
- Tablero: Próximamente interactivo
- Cartel de restricciones: Indicará si se incumple una norma y se mostrará en el cartel.
- Btn Reglamento: Abrirá una ventana con el reglamento del juego
- Btn Ayuda: Abrirá una ventana con el manual de uso del sistema.
- Btn Jugadores: Abrirá la tabla con cada jugador (Pantalla: Tabla jugadores)
- Btn Continuar: Para realizar el calculo de puntos y terminar la partida(Pantalla: Pantalla final)

Tabla/s jugadores



Elementos:

- 1 Btn por cada jugador: Al clickear se abrirá la pantalla de juego con el tablero del jugador seleccionado, con el objetivo de que coloque sus fichas en este.
- 1 Btn por cada dado: Casillas verdes oscuro. Se debe clickear la casilla del jugador que tiró el dado para seleccionar la cara (Pantalla: Pantalla dado).

Se realizó una pantalla para cada caso (Cantidad de jugadores).



Pantalla Dado

Elementos:

- Cada una de las caras del dado, siendo estas seleccionables.
- Btn Aceptar para regresar(Pantalla: Tabla jugadores)

Luego de seleccionar la cara, se activará la restricción de esta para el resto de jugadores.



**¡JUGADOR
X CANÓ!**



Pantalla Final

Elementos:

- Cartel indicando qué jugador ganó la partida por mayor cantidad de puntos totales
- Tabla de ranking global/local: Con las flechas(btn) el usuario puede cambiar de tabla para observar el ranking local(ranking con los jugadores de la partida recién terminada) o el global(ranking con los mejores puntajes de todas las partidas)
 - btn Volver a inicio: (Pantalla: Pantalla Inicio)
 - btn Reiniciar partida: crea una nueva partida con los mismos jugadores. (Pantalla: Pantalla de juego)



Inicio Jugadores

Elementos:

- btn Salir: (Pantalla: Pantalla inicio)
- btn sign in: Para registrarse por primera vez
- Textfield 1: Para ingresar nombre de usuario
- Textfield 2: Para ingresar contraseña
- btn Invitado: Para indicar que el jugador no iniciará sesión con contraseña, solo nombre temporal. NOTA: Si se selecciona esta opción el puntaje no se guardará en la base de datos.
- btn Ingresar: Para enviar los datos e iniciar sesión
- btn Iniciar Partida: (Pantalla: Pantalla de juego)

La interfaz y cantidad de div 's que hayan dependerá de la cantidad de jugadores que se hayan seleccionado, en este

caso se seleccionaron dos.

El diseño del programa se centra en crear una herramienta simple, clara y fácil de usar.

La interfaz es limpia y visualmente amigable, con colores llamativos y botones grandes para qué el usuario siempre tenga claro dónde apretar, los textos son legibles incluso en pantallas pequeñas para qué pueda ser usado cómodamente en cualquier dispositivo.

El diseño original de Draftosaurus fue cambiado de un parque de dinosaurios a una florería, todas las fichas de dinosaurios fueron cambiadas por fichas de flores de distintos tipos, y todos los recintos fueron cambiados por macetas en las qué plantar las flores.



3.2 Diseño desde Bootstrap

3.2.1 Pantalla de inicio:

Lo principal de esta pantalla es la selección de jugadores e inicio de sesión, teniendo 4 buttons superpuestos sobre una imagen para la cantidad de estos y debajo otro button para el registro (sin funcionamiento), al presionar uno de los 4 buttons, se llama a una función que hace visible un div, este es donde se ingresa la cuenta de los jugadores (o se eligen como invitados), dependiendo de que botón se haya presionado aparecerán más celdas de jugadores o menos. En este mismo div hay un button para volver atrás, volviendo a sacarle a visibilidad.

A parte de hacer visible ese div, la función también hace visible uno que se superpone a este, en el cual se encuentra un button con un href a la pantalla de juego y deshabilita el scroll default del body, así evitando errores al deslizar cuando se estos divs se hacen visibles.

```
<main class="d-block d-md-none">
  <div class="container ms-4">
    
  </div>
  
  <div class="my-3" style="position: relative; display: inline-block;">
    

    <button onclick="mostrarJugadores(2)" class="btn btn-sm" style="position: absolute; top: 70px; left: 25px; wi
    <button onclick="mostrarJugadores(3)" class="btn btn-sm" style="position: absolute; top: 70px; left: 175px; wi
    <button onclick="mostrarJugadores(4)" class="btn btn-sm" style="position: absolute; top: 170px; left: 25px; wi
    <button onclick="mostrarJugadores(5)" class="btn btn-sm" style="position: absolute; top: 170px; left: 175px; w
  </div>
  <div class="ms-5">
    <button class="btn btn-sm ms-3"></button>
  </div>
```



```
<script>
function mostrarJugadores(x){
    document.body.style.overflow = "hidden";

    document.getElementById("jugadores").style.display = "flex";
    document.getElementById("iniciar").style.display = "flex";

    switch (x){
        case 2: document.getElementById("2jugadores").style.display = "flex";
            break;

        case 3: document.getElementById("3jugadores").style.display = "flex";
            break;

        case 4: document.getElementById("4jugadores").style.display = "flex";
            break;

        case 5: document.getElementById("5jugadores").style.display = "flex";
            break;

        default: document.getElementById("jugadores").style.display = "flex";
    }
}
```

```
function ocultarJugadores(){
    document.getElementById("jugadores").style.display = "none";
    document.getElementById("2jugadores").style.display = "none";
    document.getElementById("3jugadores").style.display = "none";
    document.getElementById("4jugadores").style.display = "none";
    document.getElementById("5jugadores").style.display = "none";
    document.getElementById("iniciar").style.display = "none";
    document.body.style.overflow = "auto";
}
```

3.2.2 Pantalla de juego:

La primera parte de esta pantalla es el tablero y lo que está encima de este, esta hecha con un button que redirige a la pantalla de inicio e imágenes de las flores a seleccionar, el tablero y el cartel de restricciones abajo, sin funcionamiento alguno aún.

Debajo de esto hay 4 buttons, el principal siendo el de jugadores. Este llama a un método que hace visible un div donde muestra a los jugadores y junto a cada uno un button, con este se selecciona la restricción del dado y a su vez quién lo tiró, al presionar uno, se llama a otra función que hace visible un nuevo div, siendo este donde se selecciona la restricción en cuestión mediante buttons con las imágenes de estas. Al presionar uno, se cierra automáticamente el div de selección de restricción, volviendo al div de jugadores, esta vez con la imagen del button que fue presionado anteriormente cambiada por la restricción seleccionada. Si se trata de volver a hacer lo mismo con otro jugador, la imagen de la restricción anterior desaparecerá y mostrará la nueva.

Por último, un button para la ayuda sin funcionalidad, otro que abre el reglamento, haciendo visible un div y otro con un href a la pantalla de resultados.



```
<button class="btn btn-sm ms-5 ps-4 my-1"><a href="inicio.html"></a></button>



<div class="col">
  <button class="btn btn-sm mt-2" onclick="mostrarReglamento()"></button>
  <button class="btn btn-sm mt-2" onclick="mostrarJugadores()">
  <button class="btn btn-sm mt-2"></button>
</div>
<button class="btn btn-sm ms-5 ps-4 my-1"><a href="resultados.html"></a>
```

```
<div id="imagenjugadores" style="background-color: rgba(0,0,0,0.8); display: none; position: fixed; top: -50px; left: 0; right: 90px; bottom:

<button onclick="abrirDado(1)" class="btn btn-sm" style="position: absolute; top: 195px; left: 223px; width: 70px; height: 65px;"><img id="
<button onclick="abrirDado(2)" class="btn btn-sm" style="position: absolute; top: 263px; left: 223px; width: 70px; height: 65px;"><img id="
<button onclick="abrirDado(3)" class="btn btn-sm" style="position: absolute; top: 333px; left: 223px; width: 70px; height: 65px;"><img id="
<button onclick="abrirDado(4)" class="btn btn-sm" style="position: absolute; top: 403px; left: 223px; width: 70px; height: 65px;"><img id="
<button onclick="abrirDado(5)" class="btn btn-sm" style="position: absolute; top: 470px; left: 223px; width: 70px; height: 65px;"><img id="

<button class="btn btn-sm mt-3" onclick="cerrarJugadores()"></button>

<!--Elegir restriccion dado-->
<div id="dado" style="display: none; background-color: rgba(0,0,0,0.8); position: fixed; top: -50px; left: 0; right: 90px; bottom: 200px;
<div style="position: relative; display: inline-block;">
  
  <p style="position: absolute; top: 20px; left: 30px; font-size: 25px; font-family: 'Berlin Sans FB Demi'; color: #750708;">Jugad:
</div>
<div style="position: relative; display: inline-block;">
  

  <button onclick="cambiarDado(1)" class="btn btn-sm" style="position: absolute; top: 33px; left: 26px; width: 120px; height: 125px;"
  <button onclick="cambiarDado(2)" class="btn btn-sm" style="position: absolute; top: 33px; left: 151px; width: 120px; height: 125px;"
  <button onclick="cambiarDado(3)" class="btn btn-sm" style="position: absolute; top: 164px; left: 26px; width: 120px; height: 130px;"
  <button onclick="cambiarDado(4)" class="btn btn-sm" style="position: absolute; top: 164px; left: 151px; width: 120px; height: 130px;"
  <button onclick="cambiarDado(5)" class="btn btn-sm" style="position: absolute; top: 300px; left: 26px; width: 120px; height: 125px;"
  <button onclick="cambiarDado(6)" class="btn btn-sm" style="position: absolute; top: 300px; left: 151px; width: 120px; height: 125px;"
</div>
<button class="btn btn-sm mt-3" onclick="cerrarDado()"></button>
```

```
let jugador = 0;

function mostrarJugadores() {
  document.getElementById("imagenjugadores").style.display = "flex";
}

function cerrarJugadores() {
  document.getElementById("imagenjugadores").style.display = "none";
}

function abrirDado(x){
  document.getElementById("dado").style.display = "flex";
  jugador = x;
}

function cerrarDado(){
  document.getElementById("dado").style.display = "none";
}

function mostrarReglamento(){
  document.getElementById("reglamento").style.display = "flex";
}

function cerrarReglamento(){
  document.getElementById("reglamento").style.display = "none";
}
```



```
function cambiarDado(restriccion){
  document.getElementById("imgdado1").src = "../../public/assets/UI/dadodefault.png";
  document.getElementById("imgdado2").src = "../../public/assets/UI/dadodefault.png";
  document.getElementById("imgdado3").src = "../../public/assets/UI/dadodefault.png";
  document.getElementById("imgdado4").src = "../../public/assets/UI/dadodefault.png";
  document.getElementById("imgdado5").src = "../../public/assets/UI/dadodefault.png";

  switch (restriccion){
    case 1: document.getElementById(`imgdado${jugador}`).src = "../../public/assets/CARAS DADO/madera.png";
      break;
    case 2: document.getElementById(`imgdado${jugador}`).src = "../../public/assets/CARAS DADO/rocoso.png";
      break;
    case 3: document.getElementById(`imgdado${jugador}`).src = "../../public/assets/CARAS DADO/no planta.png";
      break;
    case 4: document.getElementById(`imgdado${jugador}`).src = "../../public/assets/CARAS DADO/rosa.png";
      break;
    case 5: document.getElementById(`imgdado${jugador}`).src = "../../public/assets/CARAS DADO/caja registradora.png";
      break;
    case 6: document.getElementById(`imgdado${jugador}`).src = "../../public/assets/CARAS DADO/estacionamiento.png";
      break;
    default: document.getElementById(`imgdado${jugador}`).src = "../../public/assets/CARAS DADO/dadodefault.png";
      break;
  }
  document.getElementById("dado").style.display = "none";
}
```

3.2.3 Pantalla de final:

Este cuenta con un tablero de ranking local, el cual es una imagen, a sus lados, buttons, los cuales intercambian entre el ranking local y el global, esto llamando a funciones que hacen visible a uno e invisible a otro, a la vez que cambia sus imágenes, pues dependiendo de en cual ranking se esté servirá uno u otro para cambiarlo.

Debajo de esto, dos buttons los cuales redirigen a la pantallas de inicio y juego.

```
<div class="ms-5 mt-3">
  <p class="ms-1" style="font-size: 30px; font-family: 'Berlin Sans FB Demi'; color: #750708;">Jugador x ganó!</p>
</div>
<!--Ranking-->
<div class="mt-1 pt-1 px-4" style="position: relative; display: inline-block;">
  <div id="local" style="display: flex"> <!--Ranking Local-->
    
  </div>
  <div id="global" style="display: none"> <!--Ranking Global-->
    
  </div>

  <!--Cambiar de ranking-->
  <button onclick="cambiarLocal()" class="btn btn-sm" style="position: absolute; top: 0; left: 0;"><img id="derecha" src="../../pub
</div>
```




```
function cambiaraGlobal(){
  if (derecha == 1){
    document.getElementById("global").style.display = "flex";
    document.getElementById("local").style.display = "none";

    document.getElementById("izquierda").src = "../../public/assets/UI/act_izq.png";
    document.getElementById("derecha").src = "../../public/assets/UI/inac_der.png";

    derecha = 0;
    izquierda = 1;
  }
}

function cambiaraLocal(){
  if (derecha == 0){
    document.getElementById("global").style.display = "none";
    document.getElementById("local").style.display = "flex";

    document.getElementById("izquierda").src = "../../public/assets/UI/ianc_izq.png";
    document.getElementById("derecha").src = "../../public/assets/UI/act_der.png";

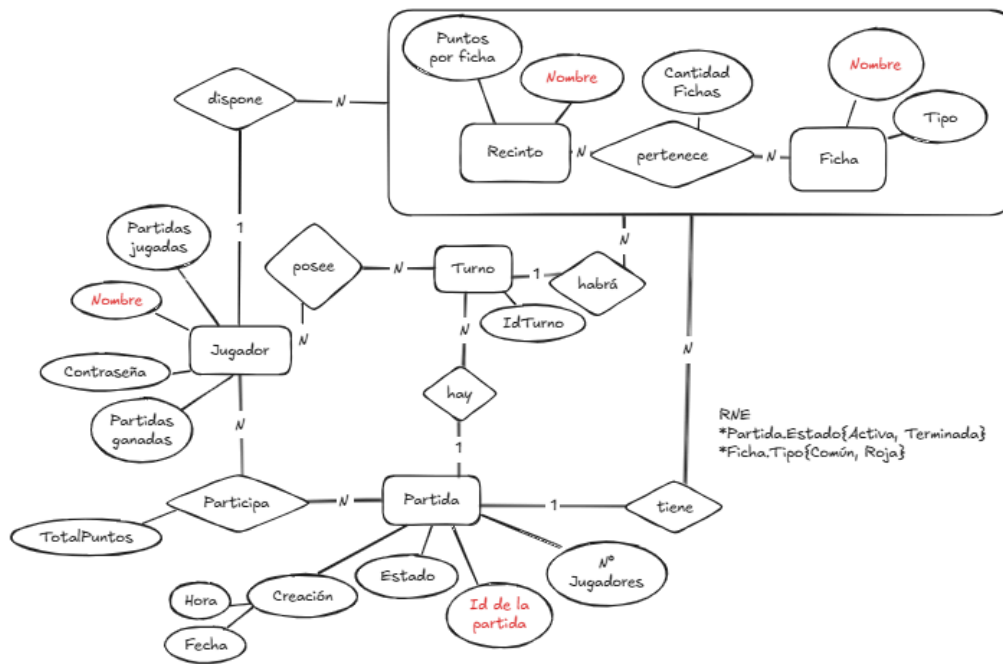
    derecha = 1;
  }
}
```

Todos los html tienen un style en el head, el cual hace referencia a una fuente para poder usarla en los textos de estos:

```
<style>
  @font-face {
    font-family: "Berlin Sans FB Demi";
    src: url("BerlinSansFBDemi.ttf") format("truetype");
  }
</style>
```

4. Modelado de datos

4.1 Diagrama Entidad-Relación (DE-R)





4.2 Pasaje a tablas

Jugador(Nombre, PartidasJugadas, Contraseña, PartidasGanadas)

Partida(IdPartida, NumJugadores, FechaHoraCreacion, Estado)

Turno(IdTurno)

Recinto(NombreRecinto, PuntosPorFicha)

Ficha(NombreFicha, Tipo)

Pertenece(NombreRecinto, NombreFicha, CantidadFichas)

FK(NombreRecinto)->Recinto(NombreRecinto)

FK(NombreFicha)->Ficha(NombreFicha)

Dispone(NombreRecinto,NombreFicha,NombreJugador)

FK(NombreRecinto)-> Recinto(NombreRecinto)

FK(NombreFicha)-> Ficha(NombreFicha)

FK(NombreJugador)-> Jugador(Nombre)

Participa(IdPartida, NombreJugador, TotalPuntos)

CEITAROX

I.S.B.O.

3MJ

Pág.



FK(NombreJugador)-> Jugador(Nombre)

FK(IdPartida)-> Partida(IdPartida)

Tiene(NombreRecinto,NombreFicha, IdPartida)

FK(NombreRecinto)-> Recinto(NombreRecinto)

FK(NombreFicha)-> Ficha(NombreFicha)

FK(IdPartida)-> Partida(IdPartida)

Hay(IdTurno, IdPartida)

FK(IdTurno)->Turno(IdTurno)

FK(IdPartida)->Partida(IdPartida)

Habr (NombreRecinto, NombreFicha, IdTurno)

FK(NombreRecinto)->Recinto(NombreRecinto)

FK(NombreFicha)->Ficha(NombreFicha)

FK(IdTurno)->Turno(IdTurno)

Posee(IdTurno,NombreJugador)

FK(IdTurno)->Turno(IdTurno)

FK(NombreJugador)-> Jugador(Nombre)



14/07/25

Restricciones No Estructurales

*Partida.Estado(Activa, Terminada)

*Ficha.Tipo(Común, Roja)

CEITAROX

I.S.B.O.

3MJ

Pág.



4.3 Normalización

4.3.1 Pasaje a 2da Forma Normal

Consideramos que no hace falta hacer el pasaje a 2da forma normal

4.3.2 Pasaje a 3ra Forma Normal

Jugador(Nombre, Contraseña)

EstadisticaJugador(Nombre, PartidasJugadas, PartidasGanadas)

FK(Nombre) -> Jugador(Nombre)

Partida(IdPartida, NumJugadores, FechaHoraCreacion, Estado)

Turno(IdTurno)

Recinto(NombreRecinto, PuntosPorFicha)

Ficha(NombreFicha, Tipo)



Pertenece(NombreRecinto, NombreFicha, CantidadFichas)

FK(NombreRecinto) -> Recinto(NombreRecinto)

FK(NombreFicha) -> Ficha(NombreFicha)

Dispone(NombreRecinto, NombreFicha, NombreJugador)

FK(NombreRecinto) -> Recinto(NombreRecinto)

FK(NombreFicha) -> Ficha(NombreFicha)

FK(NombreJugador) -> Jugador(Nombre)

Participa(IdPartida, NombreJugador, TotalPuntos)

FK(IdPartida) -> Partida(IdPartida)

FK(NombreJugador) -> Jugador(Nombre)

Tiene(NombreRecinto, NombreFicha, IdPartida)

FK(NombreRecinto) -> Recinto(NombreRecinto)

FK(NombreFicha) -> Ficha(NombreFicha)

FK(IdPartida) -> Partida(IdPartida)



Hay(IdTurno, IdPartida)

FK(IdTurno) -> Turno(IdTurno)

FK(IdPartida) -> Partida(IdPartida)

Habr (NombreRecinto, NombreFicha, IdTurno)

FK(NombreRecinto) -> Recinto(NombreRecinto)

FK(NombreFicha) -> Ficha(NombreFicha)

FK(IdTurno) -> Turno(IdTurno)

Posee(IdTurno, NombreJugador)

FK(IdTurno) -> Turno(IdTurno)

FK(NombreJugador) -> Jugador(Nombre)

Restricciones No Estructurales

*Partida.Estado(Activa, Terminada)

*Ficha.Tipo(Com n, Roja)



5. Repositorio

Enlace a repositorio: <https://github.com/PieroRod/SIGPD/tree/main>