



UCA

Universidad
de Cádiz

PINFBET

Antonio Morales Fernández
Alejandro Serrano Fernández
Pedro Antonio Navas Luque
Juan Carlos Fernández Gallardo

October 2020

Contents

1	Introducción	3
1.1	Motivación	3
1.2	Descripción del sistema actual	3
1.3	Objetivos y Alcance	3
1.3.1	Objetivo General	3
1.3.2	Objetivos Específicos	3
2	Planificación	5
2.1	Planificación del Proyecto	5
2.2	Gestión de riesgos	6
2.3	Inventario de riesgos	6
2.4	Actividades a ejecutar	7
2.5	Costes	7
2.6	Organización	7
3	Análisis de Requisitos	8
3.1	Catálogo de actores	8
3.2	Requisitos funcionales	8
3.3	Requisitos de Información	8
3.4	Requisitos no funcionales	9
3.5	Reglas de negocio	9
3.6	Alternativas tecnológicas	9
3.7	Casos de uso	9
3.7.1	Registrar Usuario	9
3.7.2	Iniciar Sesión	10
3.7.3	Añadir Asignatura Matriculada	10
3.7.4	Enviar Solicitud de Amigo	10
3.7.5	Aceptar Solicitud de Amigo	11
3.7.6	Realizar Apuesta	11
3.7.7	Apostar sobre un Amigo	12
3.7.8	Aprobar Asignatura	12
3.7.9	Ver Perfil	13
3.7.10	Editar Perfil	13
3.7.11	Ver Términos y condiciones	13
3.7.12	Ver Ayuda	13
3.7.13	Ver Contacto	13
3.7.14	Cerrar Sesión	14
4	Diseño de Sistema	15
4.1	Diseño de la arquitectura	15
4.1.1	Arquitectura física	15
4.1.2	Arquitectura lógica	15
4.1.3	Arquitectura de diseño	15
4.2	Diseño de la interfaz de usuario	16
5	Implementación del Sistema	16
5.1	Entorno tecnológico	16
5.2	Código fuente	17
5.3	Calidad del código	20
6	Pruebas del Sistema	21
6.1	Pruebas de aceptación	21

1 Introducción

1.1 Motivación

Nuestra motivación es conseguir que por medio de las apuestas los estudiantes mejoren sus notas académicas y se cree un ambiente competitivo donde obtener premios.

1.2 Descripción del sistema actual

Este proyecto es el primer paso hacia el mundo de las apuestas académicas. En este caso, se centra en buscar un sistema de apuestas simple, fiable, atractivo y de calidad profesional. Esto incluye:

- En primer lugar, encontramos un diseño adecuado y vistoso para nuestro sistema de apuestas. Esto implica:
 - Incluye un diseño adictivo para atraer a más usuarios y que estos realicen más apuestas.
 - El uso de colores llamativos para lograr un entorno atractivo.
 - Interfaz amigable y que facilite la transmisión de información, la navegabilidad y la interacción.
- En segundo lugar, elegiremos los lenguajes de programación más adecuados para realizar este sistema y facilitar sus tareas a los programadores. Ser adecuado incluye que sea sencillo, eficiente, abstracto y estructurado. Además hacemos uso de Wampserver para alojar el servidor en nuestro propio pc.
- Nuestro sistema nos permitirá opciones para apostar, realizar solicitudes de amistad, contactar con alguien si encontramos algún problema, apostar a favor, o en contra de tus amigos y editar nuestro perfil.

1.3 Objetivos y Alcance

1.3.1 Objetivo General

Desarrollar correctamente una página de apuestas académicas para la mejora académica de los estudiantes y su entretenimiento(poder aprobar la asignatura).

1.3.2 Objetivos Específicos

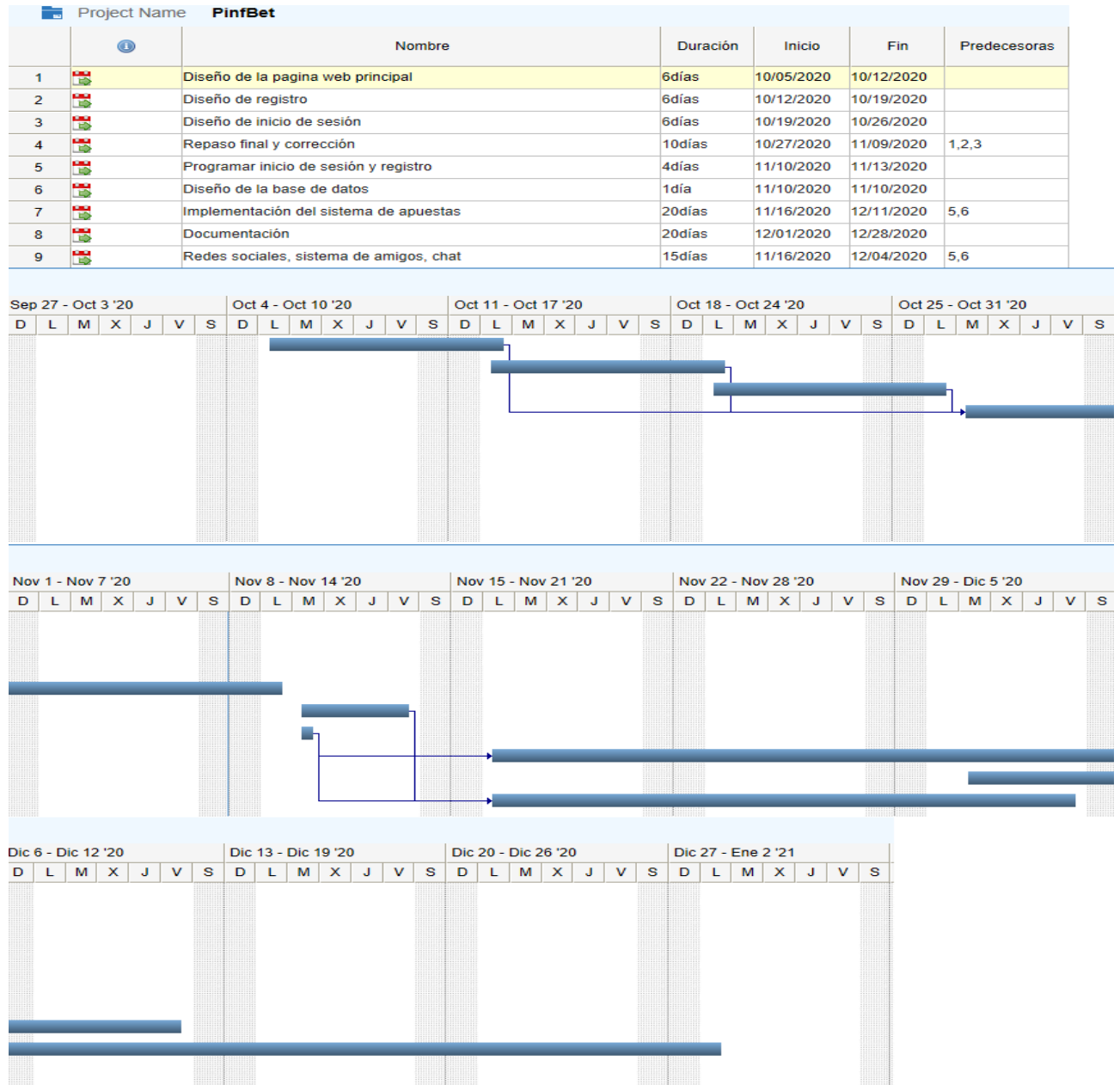
- Emplear correctamente el método MVC (Modelo-Vista-Controlador)
Grado de cumplimiento : 100%
- Diseñar correctamente una base de datos.
Grado de cumplimiento : 100%
- Diseñar correctamente la creación de usuarios, inicios de sesión y cierres de sesión.
Grado de cumplimiento : 100%
- Diseñar correctamente un sistema de apuestas basado en créditos de asignaturas universitarias.
Grado de cumplimiento : 75%
- Diseño atractivo al usuario.
Grado de cumplimiento : 95%
- Diseñar correctamente un sistema de amigos entre estudiantes universitarios.
Grado de cumplimiento : 70%
- Diseñar una interfaz con buena usabilidad para el estudiante.
Grado de cumplimiento : 100%

- Visualización de nuestra página en el localhost.
Grado de cumplimiento : 100%
- Implementación de un sistema de apuestas entre amigos
Grado de cumplimiento : 100%
- Personalización del perfil de usuario
Grado de cumplimiento : 70%
- Implementación de un sistema seguro
Grado de cumplimiento : 90%

2 Planificación

2.1 Planificación del Proyecto

Para la planificación del proyecto creamos un diagrama de Gant en el cual fijarnos, aunque hemos sido bastante flexibles a la hora de seguirlo.



2.2 Gestión de riesgos

2.3 Inventario de riesgos

En este punto del documento se detallarán todos los riesgos identificados y que han llevado a la elaboración del presente Plan de Acción. Para la descripción de los riesgos, hemos rellenado una tabla, adjuntando la siguiente información:

- Código del problema o riesgo detectado.
- Descripción del problema o riesgo detectado.
- Impacto del problema en la planificación del proyecto. Podrá ser Alto, Medio o Bajo, significando:
Impacto ALTO: Implica incumplimiento del calendario del proyecto.
Impacto MEDIO: Puede provocar un incumplimiento del calendario.
Impacto BAJO: No afecta a la planificación del proyecto.
- Probabilidad
- Fecha de detección del problema.

Riesgo / Problema				
Código	Descripción	Impacto	Probabilidad	Fecha Detección
01	Falta de conocimiento para el desarrollo del proyecto	Medio	Alta	5/10/2020
02	Pérdida parcial o total de la documentación debido a algún fallo del equipo	Alto	Baja	18/10/2020
03	Incompatibilidad de extensión de ficheros, lenguaje de programación... etc.	Alto	Baja	11/10/2020
04	Enfermedad de uno o varios integrantes	Medio	Media	15/11/2020
05	Ausencia temporal de un integrante por causas personales/familiares	Bajo	Baja	15/11/2020
06	Mala elección o falla en una herramienta de trabajo	Medio	Media	13/11/2020
07	Saturación en la red debido a un confinamiento general	Medio	Alto	17/11/2020
08	Daño o pérdida de archivos debido a un malware	Alto	Baja	17/11/2020
09	Interrupción del hardware debido a un apagón	Medio	Baja	20/11/2020

2.4 Actividades a ejecutar

Incluir en este punto una descripción de las actividades a realizar indicando, preferiblemente con una matriz de referencia cruzada, los riesgos que soluciona/palia cada una de las actividades descritas. Rellenar una tabla, adjuntando la siguiente información:

- Código del problema o riesgo que soluciona.
- Nombre corto de la actividad para identificarla en otros apartados.
- Descripción de la acción a realizar para la resolución del problema.

Actividades a Ejecutar		
Código	Actividad	Descripción
01	Investigación	Investigar sobre el conocimiento faltante y necesario para el desarrollo del proyecto
02	Copias de seguridad y duplicidad	Hacer copias de seguridad cada cierto tiempo y distribuirlas entre varios integrantes y no solo uno.
03	Planificación	Durante la planificación del proyecto los integrantes deben ponerse de acuerdo en la tecnología a usar, lo que evitaría estas incompatibilidades.
04 y 05	Repartición del trabajo	Los demás integrantes deberán repartirse las tareas del faltante durante su ausencia.
06	Uso de otra herramienta	Si tenemos problemas con una herramienta de trabajo ya sea debido a carencias de la misma, cuelgues, problemas. . . . Buscaríamos alguna herramienta alternativa similar.
07	Trabajo individual y puesta en común	Los integrantes deberán desarrollar sus partes correspondientes de manera individual y decidir la fecha en común cuando la red se recupere.
08	Prevención, detección y recuperación	Para prevenir la adquisición de un malware deberíamos usar un antivirus, además de poseer conocimientos inteligencia social. El antivirus también es útil en la detección en caso de ya poseerlo. Y para la recuperación de los archivos dañados o perdidos (o incluso del sistema), el uso de copias de seguridad.
09	Previsión	Si estamos haciendo algo importante, deberíamos ir guardando los progresos cada poco tiempo, si tenemos un equipo portátil es mejor tenerlo cargado y en casos puntuales podemos sustituir la red wifi por los datos del móvil. Esto nos ayudara a minimizar el impacto del apagón.

2.5 Costes

Para el desarrollo de este proyecto no hemos necesitado de ningún coste monetario, únicamente de tiempo y personal(los 4 integrantes del grupo).

2.6 Organización

Los 4 integrantes del grupo nos hemos organizado de la siguiente manera:

- Antonio Morales Fernández: Se ha encargado de la base de datos y desarrollo del back-end de la página, además de la creación de casos de uso.
- Alejandro Serrano Fernández: Lider del proyecto, se ha encargado del front-end, diseño y creación de imágenes de PINFBET y la página web.
- Pedro Antonio Navas Luque: Se ha encargado de la documentación, desarrollo del front-end y realización de las pruebas.

- Juan Carlos Fernández Gallardo: Se ha encargado de la documentación, términos y condiciones de la página, creación de casos de uso y realización de pruebas.

3 Análisis de Requisitos

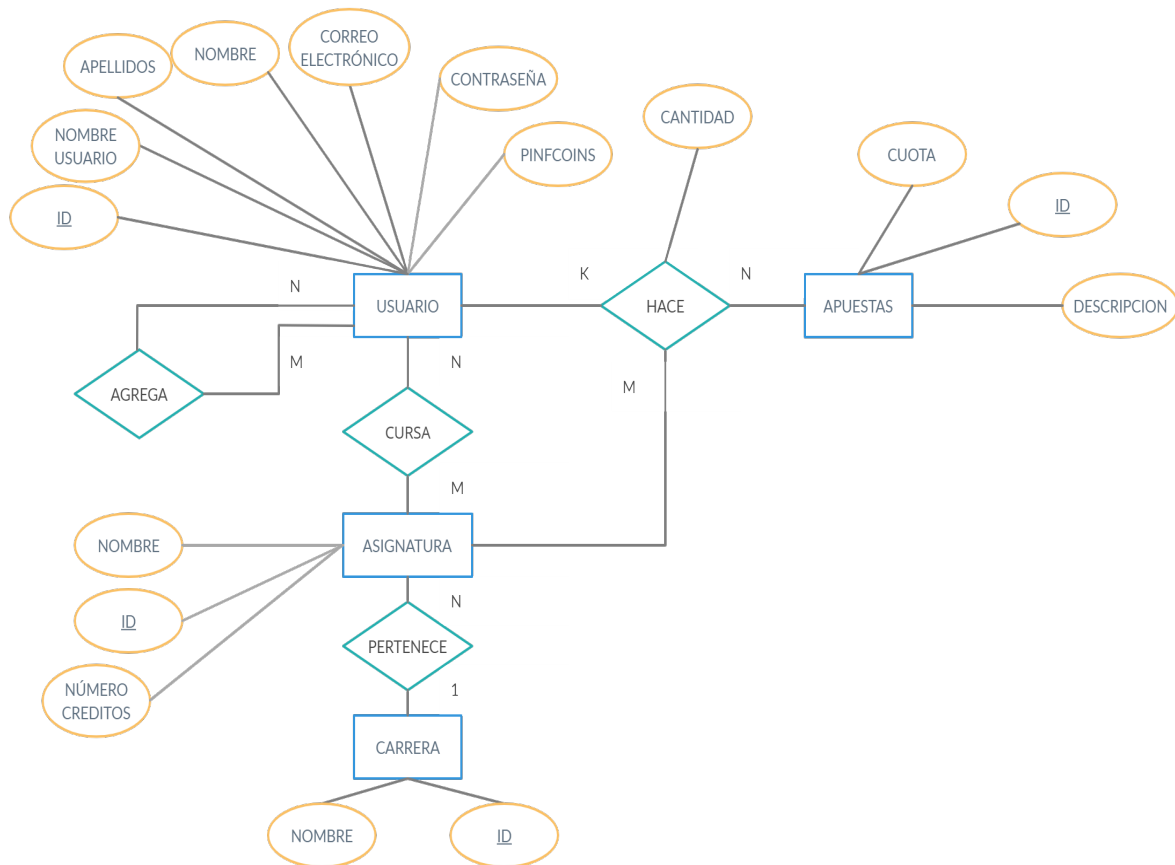
3.1 Catálogo de actores

1. Usuario: Todos los usuarios que se registren en la página pueden realizar apuestas.
2. Administrador: Los administradores tienen acceso a la base de datos.

3.2 Requisitos funcionales

Nuestra página web ofrece un sistema de apuestas académicas basado en las notas de los estudiantes. Además incluye una pequeña red social con la que interactuar con otros usuarios e incluso realizar apuestas en base a sus notas.

3.3 Requisitos de Información



3.4 Requisitos no funcionales

- Utilizaremos un diseño vistoso,colorido que llame la atención del usuario.
- Con respecto a la seguridad, las contraseñas de los usuarios se almacenaran encriptadas en la base de datos. Y los datos personales no serán visibles a otros usuarios a menos que el propio usuario lo tenga agregado en su lista de amigos.
- Exportabilidad: La página es accesible desde cualquier ordenador.
- Escalabilidad y Rendimiento: La base de datos es lo suficientemente grande para albergar un gran número de usuarios sin que el rendimiento de la página se vea afectado. En caso de un aumento significativo simplemente deberíamos aumentar de tamaño la base de datos y adquirir un servidor con mayores prestaciones.

3.5 Reglas de negocio

- Normas y Restricciones: Los usuarios deben tener al menos 17 años o la edad legal en su jurisdicción para poder hacer uso de la página web. La página web tiene derecho a suspender o desactivar sin previo aviso la cuenta de un usuario según el criterio de los administradores.
- Derechos de autor: Todas las imágenes, texto, logotipos, diseño, marcas propias...etc son propiedad exclusiva de PINFBET INC salvo que se indique explícitamente lo contrario.

3.6 Alternativas tecnológicas

Hemos utilizado MySQL para la base de datos, la hemos elegido sobre otras alternativas como EXCEL o Microsoft Access. Para la página web la hemos creado nosotros usando HTML, CSS Y PHP para conectarla a la base de datos. Una alternativa para la creación de la pagina web hubiese sido Node.js y Javascript.

3.7 Casos de uso

3.7.1 Registrar Usuario

Actor Principal: Usuario

Escenario principal:

1. El sistema muestra la pantalla de registro.
2. El usuario introduce el nombre.
3. El usuario introduce los apellidos.
4. El usuario introduce el correo electrónico.
5. El sistema comprueba que el correo electrónico existe y no se encuentra en la base de datos.
 - (a) Escenario alternativo: El sistema comprueba que el correo electrónico no existe, muestra un mensaje de error y vuelve al paso 4.
 - (b) Escenario alternativo: El sistema comprueba que el correo electrónico se encuentra en la base de datos, muestra un mensaje de error y vuelve al paso 4.
6. El usuario introduce su nombre de perfil.
7. El sistema comprueba que el nombre perfil es correcto y único.
 - (a) Escenario alternativo: El sistema comprueba que el nombre de perfil se encuentra en la base de datos, muestra un mensaje de error y vuelve al paso 6.
8. El usuario introduce la contraseña.

9. El sistema comprueba que la contraseña es correcta.

- (a) Escenario alternativo: El sistema comprueba que la contraseña es incorrecta, muestra un mensaje de error y vuelve al paso 8.

10. El usuario acepta los Términos y Condiciones y la Declaración de Responsabilidad

11. El sistema registra al usuario y muestra la pantalla de inicio de sesión.

3.7.2 Iniciar Sesión

Actor Principal: Usuario

Escenario principal:

1. El sistema muestra la pantalla de inicio de sesión.

2. El usuario introduce el correo electrónico y la contraseña.

3. El sistema comprueba que el correo electrónico se encuentra en la base de datos.

- (a) Escenario alternativo: El sistema comprueba que el email no se encuentra en la base de datos, muestra un mensaje de error y vuelve al paso 2.

4. El sistema comprueba que la contraseña coincide en la base de datos con el correo introducido.

- (a) Escenario alternativo: El sistema comprueba que la contraseña no coincide en la base de datos con el correo introducido, muestra un mensaje de error y vuelve al paso 2.

5. El sistema muestra la página de apuestas

3.7.3 Añadir Asignatura Matriculada

Actor Principal: Usuario

Escenario principal:

1. Incluye Iniciar Sesión.

2. El usuario selecciona la pestaña Asignaturas Matriculadas

3. El usuario selecciona la asignatura sobre la que está matriculado y desea añadir a la plataforma.

4. El usuario selecciona la opción enviar

5. El sistema añade la asignatura seleccionada a las demás del usuario.

6. El sistema vuelve a la pestaña de Asignaturas Matriculadas.

3.7.4 Enviar Solicitud de Amigo

Actores Principales: Usuario

Escenario principal:

1. Incluye Iniciar Sesión.

2. El usuario entra en la pestaña Amigos.

3. El usuario selecciona la opción Buscar Amigo.

4. El sistema redirige al usuario a la página de búsqueda de amigos.

5. El usuario introduce el nombre de usuario de el amigo al que quiere agregar.

6. El sistema comprueba que el nombre de usuario introducido es correcto y corresponde a un usuario de la página.
 - (a) Escenario alternativo: El sistema comprueba que el usuario introducido no existe, muestra un mensaje de error y vuelve al paso 5.
7. El sistema manda la petición de amistad al usuario seleccionado.

3.7.5 Aceptar Solicitud de Amigo

Actores Principales: Usuario

Escenario principal:

1. Include Iniciar Sesión.
2. El usuario entra en la pestaña Amigos.
3. El usuario selecciona la opción Ver Peticiones de Amigo.
4. El sistema redirecciona al usuario a la página de Peticiones de amistad.
5. El usuario ve sus peticiones de amistad.
6. El usuario selecciona una petición de amistad y la acepta.
 - (a) Escenario alternativo: El usuario selecciona una petición de amistad y la rechaza.
 - (b) El sistema redirecciona al usuario a la pestaña de amigos.
7. El sistema añade a la lista de amigos al usuario aceptado.
8. El sistema redirecciona al usuario a la pestaña de amigos.

3.7.6 Realizar Apuesta

Actor Principal: Usuario

Escenario principal:

1. Include Iniciar Sesión.
2. El usuario entra en la pestaña Apostar.
3. El usuario elige la asignatura en la que desea realizar la apuesta.
4. El usuario selecciona el tipo de apuesta que desea realizar (Aprueba/Suspende).
5. El usuario introduce el número de Pincoins que desea apostar.
6. El sistema comprueba que el usuario cuenta con un número superior o igual de Pincoins a las apostadas.
 - (a) Escenario alternativo: El sistema comprueba que el usuario no tiene las Pincoins necesarias para apostar. Muestra un mensaje de error y vuelve al paso 5.
7. El sistema resta al usuario las Pincoins apostadas de su cartera.
8. El sistema genera una nota aleatoria.
9. La apuesta es ganada. El sistema devuelve al usuario las Pincoins ganadas.
 - (a) Escenario alternativo: La apuesta es perdida y el usuario pierde las Pincoins apostadas.
10. El sistema comprueba que la nota es mayor que 5 y almacena la asignatura con las aprobadas por el usuario, además de sumarle las Pincoins correspondientes.
 - (a) Escenario alternativo: El sistema comprueba que la nota es menor que 5 y vuelve a la pestaña Apostar.

3.7.7 Apostar sobre un Amigo

Actor Principal: Usuario

Escenario principal:

1. Incluye Iniciar Sesión.
2. El usuario entra en la pestaña Amigos.
3. El usuario selecciona la opción Apostar Amigo en el amigo que quiere seleccionar.
4. El sistema redirecciona al usuario a la página de apuesta amigo.
5. El usuario selecciona la asignatura de su amigo sobre la que desea apostar.
6. El usuario selecciona el tipo de apuesta que desea realizar (Aprueba/Suspende).
7. El usuario introduce el número de Pinfcoins que desea apostar.
8. El sistema comprueba que el usuario cuenta con un número superior o igual de Pinfcoins a las apostadas.
 - (a) Escenario alternativo: El sistema comprueba que el usuario no tiene las Pinfcoins necesarias para apostar. Muestra un mensaje de error y vuelve al paso 7.
9. El sistema resta al usuario las Pinfcoins apostadas de su cartera.
10. El sistema genera una nota aleatoria.
11. La apuesta es ganada. El sistema devuelve al usuario las Pinfcoins ganadas.
 - (a) Escenario alternativo: La apuesta es perdida y el usuario pierde las Pinfcoins apostadas.
12. El sistema comprueba que la nota es mayor que 5 y almacena la asignatura con las aprobadas por el usuario sobre el que apostamos, además de sumarle las Pinfcoins correspondientes.
 - (a) Escenario alternativo: El sistema comprueba que la nota es menor que 5 y vuelve a la página de apuesta amigo.

3.7.8 Aprobar Asignatura

Actor Principal: Usuario

Escenario principal:

1. Incluye Inicar Sesión.
2. El usuario entra en la pestaña Asignaturas Aprobadas.
3. El usuario selecciona la asignatura que ha aprobado.
4. El usuario introduce la nota que tiene.
5. El sistema comprueba que la nota es mayor que 5, almacena la asignatura y suma las Pinfcoins correspondientes al usuario.
 - (a) Escenario alternativo: El sistema comprueba que la nota es inferior a 5 y muestra un mensaje de error.
 - (a) Escenario alternativo: El sistema comprueba que la nota es superior a 10 y muestra un mensaje de error.

3.7.9 Ver Perfil

Actor Principal: Usuario

Escenario principal:

1. Incluye Inicar Sesión.
2. El usuario accede a la pestaña Mi perfil.
3. El sistema muestra los datos del usuario.

3.7.10 Editar Perfil

Actor Principal: Usuario

Escenario principal:

1. Incluye Iniciar Sesión.
2. El usuario entra editar perfil.
3. El usuario edita el campo de su perfil que quiera(Nombre,Apellido,Contraseña,etc)
4. El sistema comprueba que el campo o los campos editados son correctos y almacena los cambios.
 - (a) Escenario alternativo: El sistema comprueba que algún campo esta vacío (exceptuando la contraseña que lo pueda estar), muestra un mensaje de error y vuelve al paso 3.

3.7.11 Ver Términos y condiciones

Actor Principal: Usuario

Escenario principal:

1. El usuario accede a la pestaña Términos y Condiciones.
2. El sistema redirecciona al usuario a la página de Términos y Condiciones.

3.7.12 Ver Ayuda

Actor Principal: Usuario

Escenario principal:

1. El usuario accede a la pestaña Ayuda.
2. El sistema redirecciona al usuario a la página Ayuda.

3.7.13 Ver Contacto

Actor Principal: Usuario

Escenario principal:

1. El usuario accede a la pestaña Contacto.
2. El sistema redirecciona al usuario a la página de Contacto.

3.7.14 Cerrar Sesión

Actor Principal: Usuario

Escenario principal:

1. Include Inicar Sesión.
2. El usuario accede selecciona la opción Cerrar Sesión.
3. El sistema cierra la sesión del usuario y vuelve a la pantalla principal de PINFBET.

4 Diseño de Sistema

4.1 Diseño de la arquitectura

4.1.1 Arquitectura física

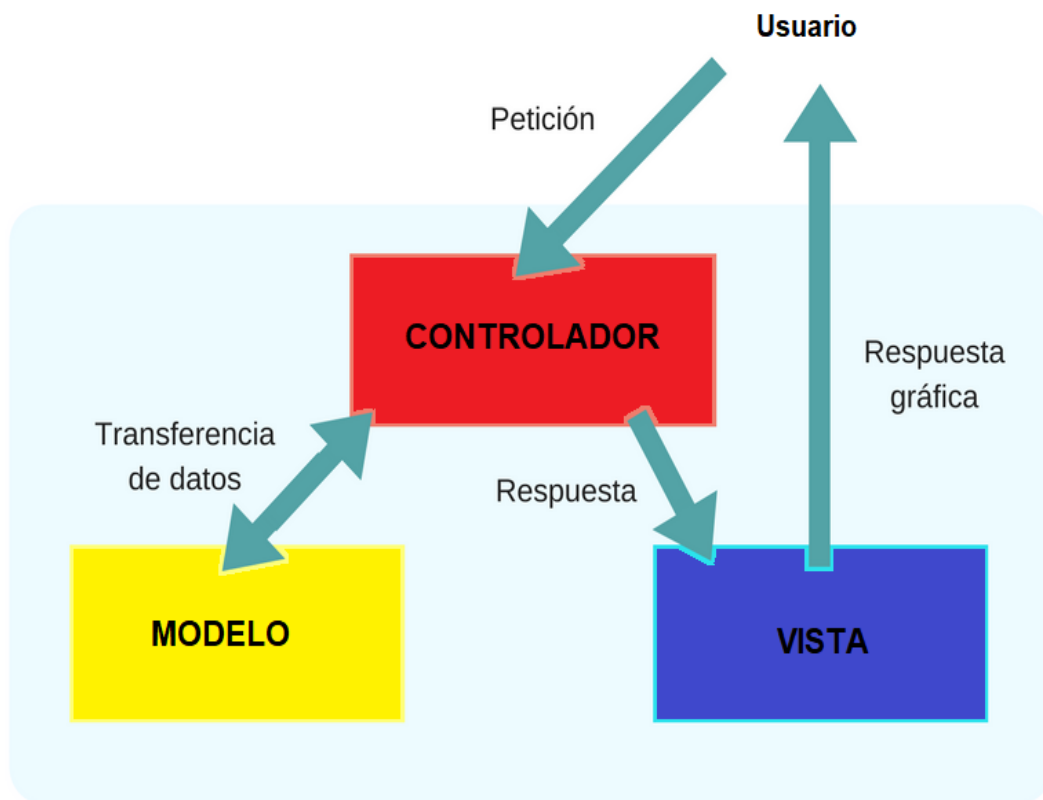
Hacemos uso de nuestro ordenador personal para alojar el servidor y la base de datos.

4.1.2 Arquitectura lógica

Los programas que utilizamos son MySQL(base de datos), WampServer(Servidor), GitHub(almacenamiento y compartición de archivos), Photoshop (creación y edición de imágenes), OverLeaf(procesador de texto).

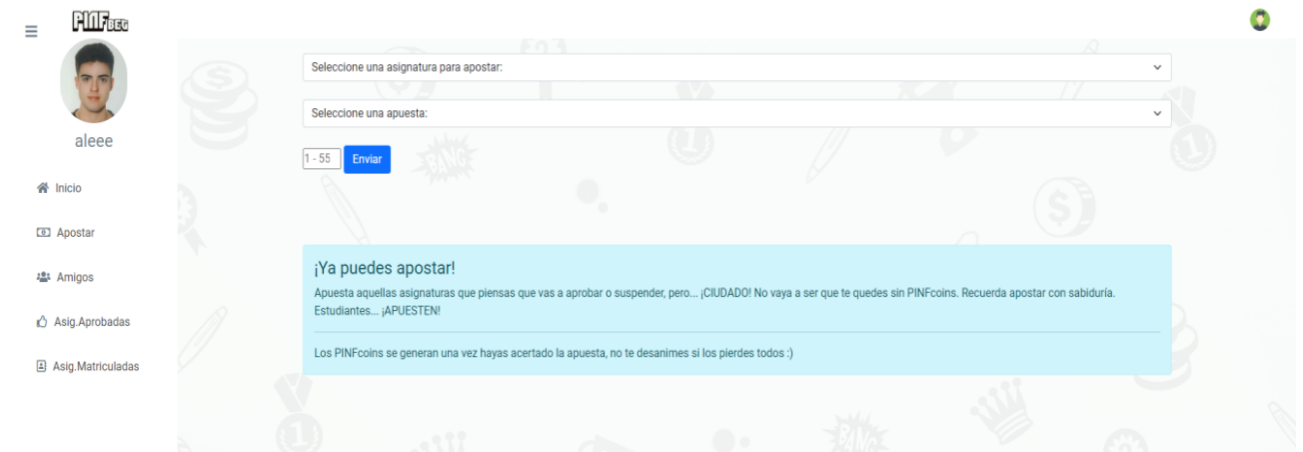
4.1.3 Arquitectura de diseño

Hemos usado un modelo-vista-controlador para estructurar todo

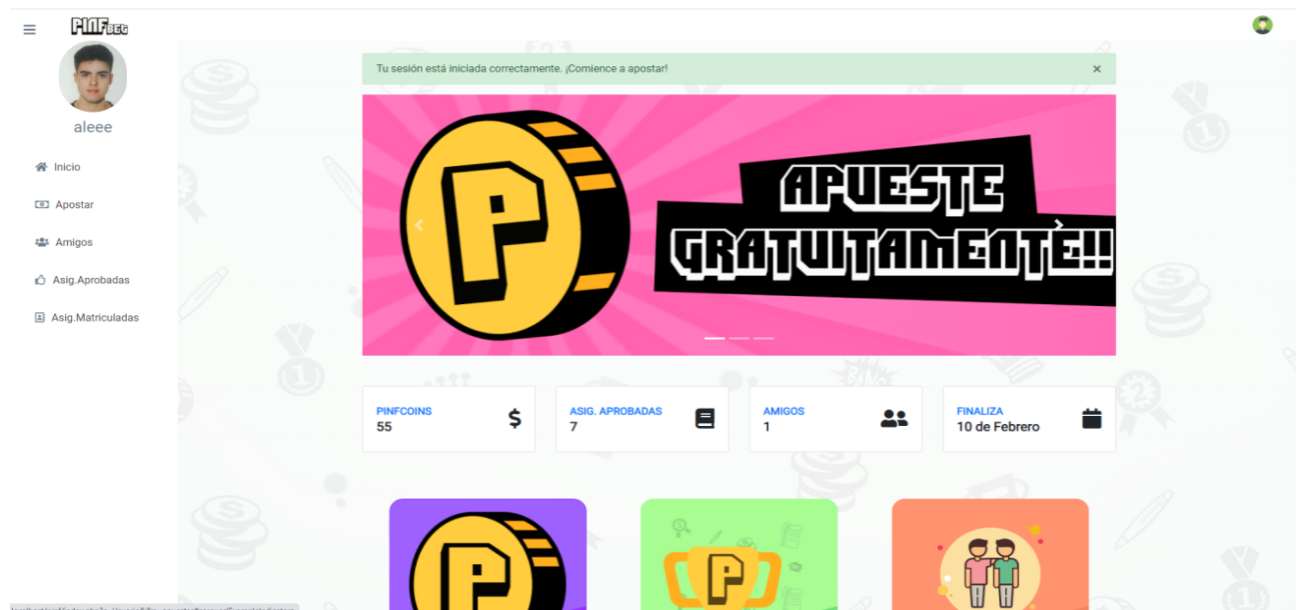


4.2 Diseño de la interfaz de usuario

Aquí mostramos algunas capturas de pantalla de nuestra interfaz :
Así se muestra nuestra apuestas.



Así se muestra el inicio una vez iniciado sesión.



5 Implementación del Sistema

5.1 Entorno tecnológico

El entorno tecnológico en el que desarrollamos el proyecto es uno con una gran variedad de opciones y competidores por lo que tendremos que ir actualizando nuestro proyecto para que no se quede obsoleto.

La tecnología que utilizamos es: Los programas que utilizamos son MySQL(base de datos), WampServer(Servidor), GitHub(almacenamiento y compartición de archivos), Photoshop (creación y edición de imagenes), Over-Leaf(procesador de texto).

5.2 Código fuente

A continuación se mostrará una de las partes del código fuente, esta corresponde a la página principal, concretamente su código html, es decir, lo que ve un usuario cualquiera al entrar en la página.

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Landing Page</title>
  <link rel="stylesheet" href="Estatico/Estilos/estilos-index.css">
  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,700,800&display=swap" rel="stylesheet">
  <link href='https://unpkg.com/boxicons@2.0.5/css/boxicons.min.css' rel='stylesheet'>
</head>

<body>
  <header>
    <nav>
      <a href="index.php?c=Usuario&a=iniciosesion">Iniciar Sesion</a>
    </nav>
    <div class="container">
      <div class="textos">
        
        <h2>Realiza tus apuestas de asignaturas totalmente GRATIS !</h2>
        <a href="index.php?c=Usuario&a=save">
          <span></span>
          <span></span>
          <span></span>
          <span></span>
          Reg strate
        </a>
      </div>
      
    </div>
  </header>
  <div class="wave">
    <div style="height: 150px; overflow: hidden;"><svg viewBox="0 0 500 150"
      preserveAspectRatio="none" style="height: 100%; width: 100%;"><path d="M0
      .00,49.98 C150.00,150.00 349.20,-50.00 500.00,49.98 L500.00,150.00 L0
      .00,150.00 Z" style="stroke: none; fill: #fff;"></path></svg></div>
  </div>

  <section class="contenedor" id="expertos">
    <h2 class="subtitulo"> Que puedes hacer?</h2>
    <section class="experts">
      <div class="cont-expert">
        
        <h3 class="n-expert">Compite con tus amigos</h3>
      </div>
      <div class="cont-expert">
        
        <h3 class="n-expert">Gana PINFCOINS</h3>
      </div>
      <div class="cont-expert">
        
        <h3 class="n-expert">Mot vate para estudiar</h3>
      </div>
    </section>
  </section>

  <footer id="footer">
    <div class="contenedor footer-content">
      <div class="contact-us">
        <h2 class="brand">S guenos en nuestras redes sociales</h2>
        <p>!Recuerda estar siempre atento a ellas!</p>
      </div>
    </div>
  </footer>
</body>
```

```

        <div class="social-media">
            <a href="/" class="social-media-icon">
                <i class='bx bxl-facebook'></i>
            </a>
            <a href="/" class="social-media-icon">
                <i class='bx bxl-twitter'></i>
            </a>
            <a href="/" class="social-media-icon">
                <i class='bx bxl-instagram'></i>
            </a>
        </div>
    </div>
    <div class="line"></div>
</footer>
</body>
</html>

```

El siguiente código corresponde a una parte de UsuarioControlador.php, que como su nombre indica es una parte importante de como funciona nuestro sistema modelo-vista-controlador.

En el podemos ver algunas funciones cuya función es simplemente obtener una "Vista", y algunas más importantes como "apostar" y "apostar_amigos".

```

<?php
require_once 'Modelos/UsuarioModelo.php';
require_once 'Modelos/AsignaturaModelo.php';
require_once 'Modelos/UsuApuesAsigModelo.php';
class UsuarioControlador {
    public function principal(){
        require_once 'Vistas/Principal.html';
    }

    public function cambiocontra(){
        require_once 'Vistas/CambioContra.html';
    }

    public function contacto(){
        require_once "Vistas/contacto.html";
    }

    public function terminos(){
        require_once "Vistas/Terminos.html";
    }

    public function apuestas(){
        $asig=new Asignatura(null,null,null,null,null);
        $asig->insertar_id_user($_SESSION['identidad']->id);
        $num=$asig->recuento_asignaturas_aprobadas();
        $num1=(int)$num->num;
        $amist=new Amistad(null,null);
        $amist->insertar_identif1($_SESSION['identidad']->id);
        $num2=$amist->recuento_amigos();
        $num3=(int)$num2->num4;
        require_once "Vistas/Apuestas.phtml";
    }

    public function miperfil(){
        $asig=new Asignatura(null,null,null,null,null);
        $asig->insertar_id_user($_SESSION['identidad']->id);
        $num=$asig->recuento_asignaturas_aprobadas();
        $num1=(int)$num->num;
        $asig1=new Asignatura(null,null,null,null,null);
        $asig1->insertar_id_user($_SESSION['identidad']->id);
        $num2=$asig1->recuento_asignaturas_matriculadas();
        $num3=(int)$num2->num;
        require_once "Vistas/MiPerfil.phtml";
    }
}

```

```

public function ayuda(){
    require_once "Vistas/ayuda.html";
}

public function asignaturas(){
    require_once "Vistas/asignatura.phtml";
}

public function apostar(){
    if(!isset($_SESSION['identidad'])){
        require_once 'Vistas/Registro.phtml';
        $_SESSION['error_registro']=false;
    }else{
        if ($_SESSION['identidad']->pinfcoins == 0) {
            (new Usuario(null,null,null,null,null,$_SESSION['identidad']->id
            ,40))->actualizar_pinfcoins_usuario();
            $_SESSION['identidad']->pinfcoins=40;
        }
        $_SESSION['error_apostar'] = false;
        $asig= new Asignatura();
        $asig->insertar_id_user($_SESSION['identidad']->id);
        $misasignaturas=$asig->asignaturas_matriculadas_usuario();
        $apuestas = (new Usuario())->mostrar_apuestas();
        if (isset($_POST['apuestas'])){
            $asigapos = $_POST['asignaturas']=='false' ? false : $_POST['asignaturas'];
            $apos = $_POST['apuesta']=='false' ? false : $_POST['apuesta'];
            $pinfc=isset($_POST['pinfcoin']) ? $_POST['pinfcoin'] : false;

            $asig->insertar_id_asignatura($asigapos);
            if($asigapos && $apos && $pinfc){
                $apuesta= new UsuApuesAsig ($_SESSION['identidad']->id,(int)
                $asigapos,(int)$apos);
                $aleatorio=rand(0,10);

                $cuota=($apuesta->obtener_cuota())->fetchObject();
                $cuota1=(float)$cuota->cuota;
                $pinfcoins_t = $_SESSION['identidad']->pinfcoins-$pinfc;
                if($apos==1 && $aleatorio<5){

                    $pinfcoins_t=$pinfcoins_t+($pinfc*$cuota1);
                    $pinfcoins_t=(int)$pinfcoins_t;
                }
                if($apos==2 && $aleatorio >= 5){
                    $pinfcoins_t=$pinfcoins_t+($pinfc*$cuota1);
                    $pinfcoins_t=(int)$pinfcoins_t;
                }
                if($aleatorio >=5){
                    $asig1=$asig->buscar_asignatura();
                    $pinfcoins_t=$pinfcoins_t+($aleatorio*$asig1->numero_creditos
                    );
                    $asig->insertar_asignatura_aprobada();
                    $asig->borrar_asignatura();
                }
                $_SESSION['identidad']->pinfcoins=$pinfcoins_t;
                $user=new Usuario(null,null,null,null,null,$_SESSION['identidad']->id,$_SESSION['identidad']->pinfcoins);
                $user->actualizar_pinfcoins_usuario();
                $apuesta->insertar_apuesta();
            } else {
                $_SESSION['error_apostar'] = "Rellena todos los campos";
            }
            require_once 'Vistas/apostar.phtml';
            //header("Location:index.php?c=Usuario&a=apostar");
            //$asignatura = isset($_POST[])
        }
        require_once 'Vistas/apostar.phtml';
    }
}

```

```

}

public function apostar_amigo(){
    if(!isset($_SESSION['identidad'])){
        require_once 'Vistas/Registro.phtml';
        $_SESSION['error_registro']=false;
    }else{
        $_SESSION['error_apostar_amigo'] = false;
        $asig= new Asignatura();
        $asig->insertar_id_user($_SESSION['id_amigo']);
        $misasignaturas=$asig->asignaturas_matriculadas_usuario();
        $apuestas = (new Usuario())->mostrar_apuestas();
        if (isset($_POST['apuestas'])){
            $asigapos = $_POST['asignaturas']=='false' ? false : $_POST['asignaturas'];
            $apos = $_POST['apuesta']=='false' ? false : $_POST['apuesta'];
            $pinfc=isset($_POST['pinfcoin']) ? $_POST['pinfcoin'] : false;

            $asig->insertar_id_asignatura($asigapos);
            if($asigapos && $apos && $pinfc){
                $apuesta= new UsuApuesAsig ($_SESSION['identidad']->id,$asigapos, $apos);
                $aleatorio=rand(0,10);
                $aleatorio=10;

                $cuota=($apuesta->obtener_cuota())->fetchObject();
                $cuota1=(float)$cuota->cuota;
                $pinfcoins_t = $_SESSION['identidad']->pinfcoins-$pinfc;
                if($apos==1 && $aleatorio<5){

                    $pinfcoins_t=$pinfcoins_t+($pinfc*$cuota1);
                    $pinfcoins_t=(int)$pinfcoins_t;
                }
                if($apos==2 && $aleatorio >= 5){
                    $pinfcoins_t=$pinfcoins_t+($pinfc*$cuota1);
                    $pinfcoins_t=(int)$pinfcoins_t;
                }
                if($aleatorio >= 5){
                    $user=new Usuario(null,null,null,null,null,null,$_SESSION['id_amigo']);
                    $asig1=$asig->buscar_asignatura();
                    $user->insertar_pinfcoins($user->obtener_pinfcoins()+ ($aleatorio*$asig1->numero_creditos));
                    $user->actualizar_pinfcoins_usuario();
                    $asig->insertar_asignatura_aprobada();
                    $asig->borrar_asignatura();
                }
                $_SESSION['identidad']->pinfcoins=$pinfcoins_t;
                $user=new Usuario(null,null,null,null,null,null,$_SESSION['identidad']->id,$_SESSION['identidad']->pinfcoins);
                $user->actualizar_pinfcoins_usuario();
                $apuesta->insertar_apuesta();
            }
            else {
                $_SESSION['error_apostar_amigo'] = "Rellena todos los campos";
            }
            require_once 'Vistas/apostar_amigo.phtml';
        }
        require_once 'Vistas/apostar_amigo.phtml';
    }
}
}

```

5.3 Calidad del código

Para comprobar la calidad del código hemos utilizado dos herramientas online. Estas son AChecker, que revisa tanto el código HTML como el CSS, y Nu Html la cual se centra en el HTML.

Después de haber pasado nuestro código html por las herramientas nos dan resultados aceptables, ya que no nos proporcionan ni porcentaje ni nota. Con esto hemos intentado mejorar nuestro código consiguiendo así, por ejemplo, en la página principal el siguiente resultado:

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for Principal.html

Checker Input

Show

☐ source

☐ outline

☐ image report

Options...

Check by

file upload

▼

Seleccionar archivo

Principal.html

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

Check

Document checking completed. No errors or warnings to show.

6 Pruebas del Sistema

6.1 Pruebas de aceptación

Para las pruebas de nuestro sistema hemos seguido la siguiente dinámica: Cada vez que implementábamos un subsistema, o añadíamos alguna subpágina a nuestro sistema, realizábamos pruebas de esa parte y además comprobábamos que no había conflictos entre lo hecho anteriormente y lo más reciente. Siguiendo esta metodología hemos comprobado que funcionan correctamente las siguientes funcionalidades.

- Registro de usuario.
- Inicio de sesión.
- Cerrar sesión.
- Realizar apuestas sobre nuestras asignaturas.
- Enviar solicitud de amistad a un amigo.
- Aceptar/Rechazar solicitudes de amistad.
- Realizar apuestas sobre un amigo.
- Ver nuestro perfil
- Editar nuestro perfil.
- Añadir asignaturas aprobadas.
- Matricularse de las asignaturas.
- Navegar por la página.

Además también hemos comprobado que funcionan correctamente las siguientes no funcionalidades.

- Correcta visualización de la página.
- Las contraseñas se guardan encriptadas.