PINFBET

GRUPO 22

Iván López Ceballos
Francisco Chanivet Sánchez
Antonio de los Reyes Pérez
Eliseo Fernández Monroy

14 de enero del 2021

ÍNDICE

INTRODUCCIÓN AL PROYECTO INTEGRADO: PINFBET

Logo de PINFBET

La temática de este proyecto consiste en una página de apuestas que se nos ha propuesto en la asignatura de Proyectos Informáticos. Para este proyecto trabajaremos en un grupo de cuatro personas donde a cada miembro se le asignará un rol o varios dependiendo de cómo nos hayamos repartido las tareas. El lenguaje de programación a utilizar será elegido por nosotros, pudiendo elegir desde C++ hasta HTML, llevándonos a la conclusión de que el resultado final puede ser tanto en terminal como gráfico. En nuestro caso, hemos elegido como framework Django, utilizando Python para el código, y HTML y CSS como lenguajes de marcado.

Hemos alojado la aplicación web en un servidor gratuito de Heroku, con la siguiente dirección: https://pinfbet.herokuapp.com.*

El código fuente de la aplicación puede encontrarse en https://github.com/Grupo22-PINF/PINFBET, y el repositorio del servidor de Heroku se encuentra en https://git.heroku.com/pinfbet.git.

* Los servidores de Heroku ponen sus aplicaciones a dormir tras un tiempo de inactividad. La primera conexión a la página será lenta, pero este comportamiento no se dará con un flujo de usuarios constante.

1. PLANIFICACIÓN DEL PROYECTO

Para realizar la planificación del proyecto, hemos utilizado el diagrama de Gantt que nos permite repartir la carga de trabajo de forma equitativa entre todos los miembros del grupo y así no saturarlos.

Para ello, hemos utilizado el programa ProjectLibre que nos permite realizar esta organización de una forma más sencilla.

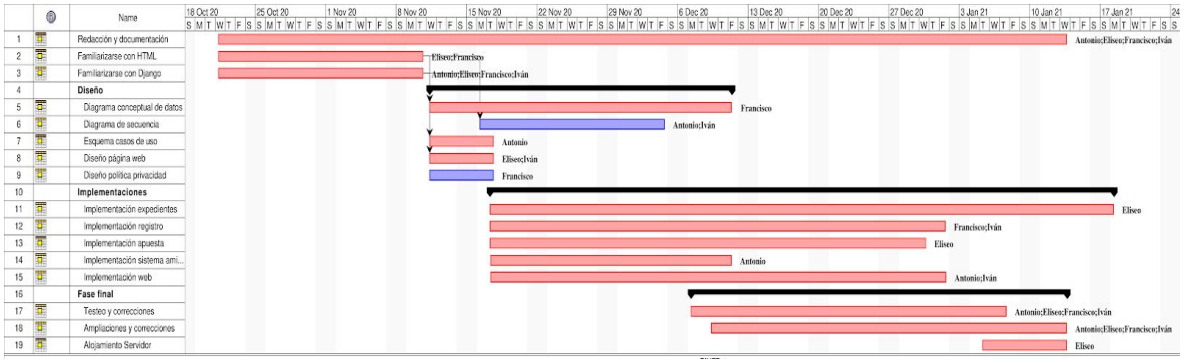


Diagrama de Gantt de nuestro proyecto

Como vemos en el diagrama, nuestro equipo se ha organizado de tal manera que todo el mundo tenga trabajo en cada instante. Como las herramientas que hemos escogido para el proyecto no las conocemos en profundidad, hemos establecido un intervalo de tiempo para estudiarlas. Por lo tanto, el primer intervalo de tiempo que vemos en el diagrama se corresponde con la familiarización del Framework y los lenguajes de programación. Antes de comenzar a programar la página web, tenemos que decidir las diferentes funcionalidades que le vamos a otorgar, además del diseño para que la página tenga una apariencia destacable y sea sencilla su navegación. Finalmente, ya tendríamos los requisitos necesarios para empezar con la implementación, la etapa a la que más tiempo le vamos a dedicar, debido a que es donde pueden surgir más problemas.

A medida que vamos realizando el trabajo, debemos rellenar la documentación con las mejoras y actualizaciones que vayamos incorporando.

2. ANÁLISIS DE LOS REQUISITOS DEL PROYECTO

Para poder diseñar la aplicación web con el principal objetivo de obtener buenos resultados, hemos realizado distintas tareas dentro de esta fase de desarrollo del proyecto.

2.1. Catálogo de actores

En nuestra página Web podemos encontrar los siguientes roles:

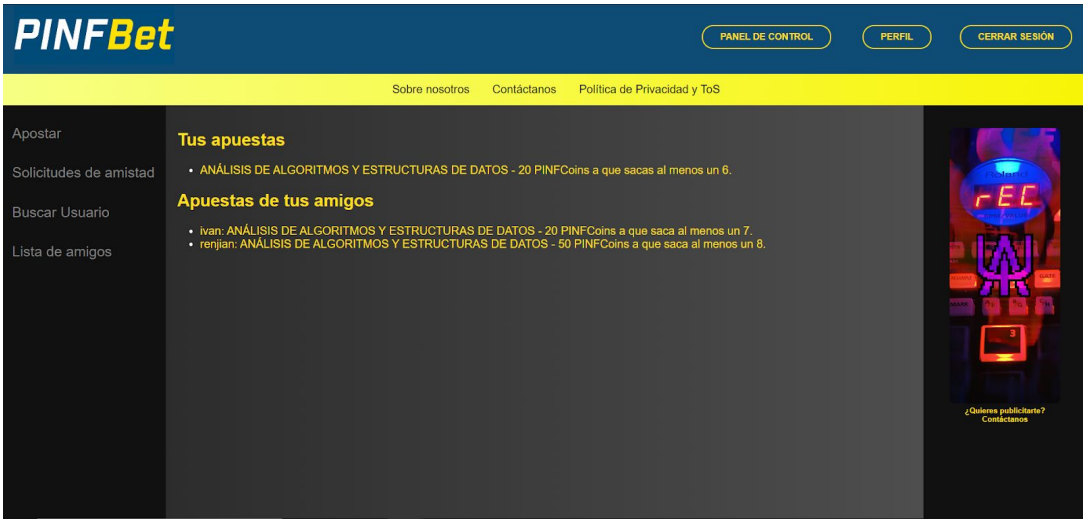
- **Usuario**, que puede interactuar con las distintas funcionalidades que ofrece la página web. Este, para un total acceso, debe registrarse, facilitando datos personales, como el nombre o el curso en el que se encuentra.
- **Administrador**, quien se encarga de gestionar los servicios si ocurre algún incidente o si algún usuario quiere modificar alguno de sus datos.

2.2. Requisitos funcionales

Como hemos citado en el punto anterior, los requisitos funcionales dependen del hecho de que estemos registrados o no. Supongamos el caso de que no estemos registrados, podríamos acceder a diferentes secciones: “Sobre nosotros”, donde presentamos a los miembros que conforman el Grupo 22 y un tutorial de cómo registrarse en la web, “Contáctanos”, que facilita a las personas que quieran publicitarse dentro de nuestra web con un contacto directo con nosotros, y “Política de Privacidad y Términos de Uso” donde explicamos cómo tratamos los datos de los usuarios y qué no deben hacer dentro de la web.

Página web cuando entramos sin iniciar sesión

Una vez que estemos registrados, podemos acceder a la función “Apostar”, que mediante ella apostamos monedas si obtenemos la nota que hemos apostado en la asignatura correspondiente. Además, podemos realizar diversas acciones a nuestro perfil, como subir un expediente, actualizar la matrícula o incluso eliminar nuestra cuenta. `Para las dos primeras funciones necesitaremos un archivo en formato PDF que sean nuestra matrícula de este año y el expediente respectivamente. También el usuario puede buscar a otros usuarios y añadirlos como amigos, así pudiendo ver las apuestas que han realizado.



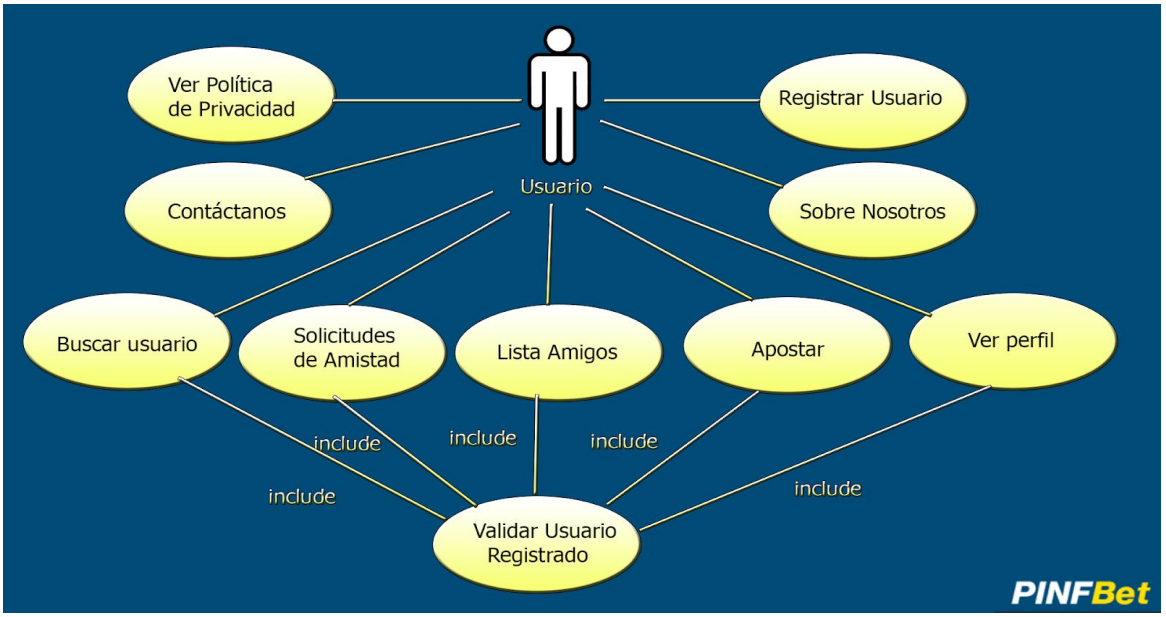
Página web cuando el usuario está registrado y es administrador

2.3. Estudio de alternativas tecnológicas

Para poder implementar este servicio, podemos utilizar otras alternativas que están disponibles en el mercado como una aplicación móvil hecha con el motor gráfico Unity, una página web hecha con el framework Laravel, una aplicación hecha con Android Studio, etc.

Como podemos observar, existen diversas posibilidades para llevar a cabo la implementación de este proyecto.

2.4. Modelo de caso de uso general



Modelo de caso de uso de un usuario

Para poder llevar a cabo correctamente el desarrollo de la aplicación web, hemos tenido que modelar un caso de uso general de la aplicación para poder implementar las distintas funciones.

Como podemos observar en la imagen adjunta, en el caso de que el usuario no esté registrado, las diferentes funciones que se van a poder realizar desde la aplicación web son las siguientes:

- **Registrarse**
- **Acceder a página de "Contáctanos"**
- **Acceder a página de "Sobre Nosotros"**
- **Ver Política de Privacidad y Términos de Uso.**
- **Buscar usuario**

En el caso opuesto, el usuario puede hacer uso de las funcionalidades principales:

- **Apostar**
- **Ver perfil**
- **Lista de Amigos**
- **Solicitudes de Amistad**

2.4.1. Registro

Para poder navegar a través de la web sin ninguna restricción, el usuario puede registrarse en ella. Sin embargo, sólo necesitamos ser mayores de edad para poder registrarnos debido a la legalidad vigente que lo contempla así.

Como citamos anteriormente, necesitamos estar registrados dentro de la página web para poder usar todas las funcionalidades de nuestros servicios. Para dicho registro necesitamos los siguientes datos: nuestro nombre y apellidos, carrera que estemos cursando y un nombre de usuario que debe ser exclusivo para cada usuario.

2.4.2. Buscar usuario

Para poder acceder a esta opción, necesitamos registrarnos en el sistema. Entonces, una vez que estemos registrados, podemos buscar a un usuario, no haciendo falta escribir su nombre completo, porque con poner una letra o varias se nos abrirá un listado que coincidan con dichas referencias. Luego, al hacer click en el usuario que deseamos agregar, veremos información suya como su nombre y apellidos y las asignaturas que esté cursando, además de un botón mediante el cual podemos enviarle una solicitud de amistad. Sin embargo, en el caso de que ya lo tenemos agregado, él a nosotros o que le hayamos enviado una solicitud de amistad, este botón no estará disponible.

2.4.3. Apostar

Una vez que hayamos subido nuestro expediente, podemos apostar por cualquiera de nuestras asignaturas. Las apuestas se realizan de la siguiente forma: Si sacamos la nota que hayamos apostado en una determinada asignatura, ganaremos la apuesta y obtendremos PINFCoins, de lo contrario, habremos perdido nuestros PINFCoins.

2.4.4. Ver perfil

Dentro de esta sección podemos visualizar nuestros datos, además de subir nuestro expediente y nuestra matrícula. Una vez realizado esto, podremos apostar sin problemas. Como última instancia, dentro de nuestro perfil tenemos la opción de eliminar nuestra cuenta.

En caso que un usuario deje de utilizar este servicio, lo más recomendable es que elimine su cuenta, para que sus datos se eliminen por completo.

2.4.5. Solicitudes de Amistad

Las solicitudes de amistad que nos hayan llegado de los diferentes usuarios registrados estarán visibles dentro de esta sección, que podemos ver en la barra lateral izquierda de nuestra página web. Una vez dentro, podemos aceptar o rechazar su petición.

2.4.6. Lista de Amigos

Gracias al sistema de amigos que proporcionamos, podemos comprobar algunos de los datos personales de nuestros amigos sin necesidad de buscarles, además de comprobar las apuestas que hayan realizado recientemente.

2.4.7. Política de privacidad

Para poder tratar los datos de los usuarios, necesitamos definir una política de privacidad donde se detalle cómo funcionan nuestros servicios gestionando sus datos y nuestro compromiso de proteger sus datos personales frente a terceros.

En esta sección, también se encuentran los términos de uso de la página web que le informamos al usuario de que no está permitido subir algún tipo de fichero que sea ilegal o intentar comprometer los datos de los demás usuarios.

2.4.8. Contáctanos

Cualquier empresa o usuario puede publicitarse si cumple con los requisitos necesarios establecidos por nosotros. Esta sección se puede encontrar haciendo click en “Contáctanos” ubicada en la barra superior de la web. Los anuncios de las empresas quedarán visibles en la barra lateral derecha.

2.4.8 Sobre nosotros

Los usuarios pueden saber acerca de nosotros mediante un pequeño texto de presentación y un tutorial sobre cómo registrarse.

2.5. Diseño de la aplicación web

Una vez realizado los modelos correspondientes para saber los requisitos principales del proyecto, vamos a diseñar la página web en el Front-End. Para ello, hemos hecho un primer diseño de la página principal donde podemos observar algunos elementos que hemos citado anteriormente.

La interfaz de la página web se ha pensado para un uso intuitivo y directo, centrándonos en que los usuarios se sientan cómodos navegando por ella.

Tenemos una barra en la parte superior en la cual se nos muestra la opción de “Registrarse” e “Iniciar Sesión”, que, una vez iniciemos sesión, se convierte en “Cerrar Sesión”.

3. DISEÑO DEL SISTEMA.

3.1. Arquitectura física

Los elementos tangibles utilizados para la realización del trabajo han sido nuestros propios equipos, además de todos los periféricos necesarios para su uso. No ha sido necesario adquirir material adicional para el desarrollo del proyecto, todos los elementos utilizados fueron adquiridos con anterioridad.

3.2. Arquitectura lógica

Para poder llevar a cabo el desarrollo del proyecto, hemos utilizado el lenguaje de programación Python, el lenguaje de marcado HTML, el lenguaje de hojas de estilos CSS, y el framework django, que es el que nos ha posibilitado implementar muchas funciones gracias a que ya estaban implementadas dentro de ellas.

Respecto a los datos, Django mantiene su propia base de datos y permite encriptar las contraseñas directamente. La base de datos que utilizamos al inicializar el servidor de forma local es la de SQLite, mientras que el servidor Heroku que hemos habilitado para alojar la web utiliza PostgreSQL.

3.3. Arquitectura de diseño

3.3.1. Capa de presentación

Cuando el usuario entré en la web, se va a encontrar dentro de la sección “Sobre Nosotros”. Lo primero que tendrá que hacer este usuario será registrarse si quiere usar todas las funcionalidades disponibles, en el caso de que no lo haya hecho anteriormente.

En caso contrario, solo tendrá que iniciar sesión. Ya con acceso a nuestra cuenta, podremos apostar en la página, además de poder agregar amigos. También tendrá la opción de eliminar su cuenta.

3.3.2. Capa lógica

Para poder realizar todas las operaciones disponibles, Django se comunicará de tal forma que se enviarán todos los datos y peticiones dependiendo de cada función. Cuando el usuario envía sus datos, django los recibirá y los guardará en la base de datos.

3.3.3. Capa de datos

En el servicio desarrollado por nosotros necesitaremos una estructura diseñada para poder almacenar diferentes datos. Para ello, hemos utilizado una base de datos que almacena varias tablas:

-**Alum asig**: En esta tabla almacenamos las asignaturas de la que está matriculado el alumno.

-**Alumno**: Aquí almacenamos los datos de los usuarios que se registran en nuestros servicios.

-**Amistad**: En esta tabla almacenamos los amigos que tiene cada usuario.

-**Asignatura**: En esta tabla almacenamos las asignaturas que se imparten en las distintas titulaciones de la UCA.

-**Usuario**: Esta tabla sirve para iniciar sesión dentro de la web. Se almacena el nick, el correo electrónico y la contraseña.

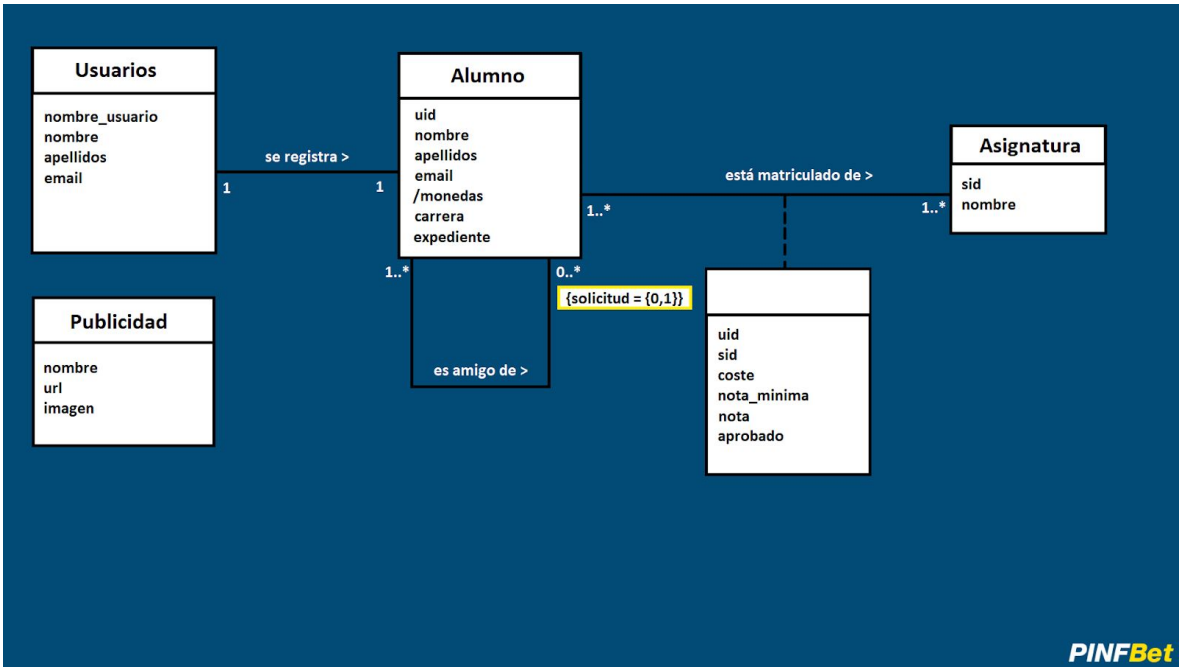
3.4. Diseño de la interfaz de usuario

Como sabemos es esencial tener una interfaz de usuario clara e intuitiva para que la página web sea legible y accesible para cualquier usuario.

La interfaz diseñada por nuestro grupo consta de una barra superior que muestra los botones de inicio de sesión y registro, además de una barra que muestra la opción para buscar amigos y para apostar. Además, dentro de nuestro perfil, podemos ver nuestros amigos y notificaciones, subir la matrícula, actualizar el expediente, e incluso borrar nuestra cuenta si así deseamos.

Todo esto está en un solo menú para que sea directo y que el usuario no tenga que navegar por menús enrevesados para encontrar estas opciones.

3.5. Diseño de datos.



Modelo conceptual de datos de la página web

Las **claves primarias** de cada clase son las siguientes:

- **Alumno: uid**
- **Asignatura: sid**

Para obtener el número de monedas que disponemos actualmente, el sistema aplica la siguiente fórmula:

$$PINFCoins = (Cantidad\ apostada * (Nota\ obtenida/10) + Cantidad\ apostada)$$

Cuando el usuario se registra, el sistema le regala 500 monedas y cuando sube la matrícula, este valor se actualiza al valor correspondiente.

Para que haya un mayor entendimiento de este esquema, vamos a aclarar los siguientes casos:

- La solicitud tiene valor 0 ó 1 según corresponda, en el caso de que no hayas aceptado aún la solicitud, este tendrá valor 0 y en el caso opuesto, tendrá valor 1.
- El usuario que está registrado en la tabla Users, se registra en la plataforma con la tabla Alumnos. Esta tabla sirve para que el sistema identifique al usuario previamente.

- El valor de las monedas se obtiene sumando el número de monedas que tengamos actualmente más las conseguidas por cada apuesta correspondiente. También nos encontramos con el caso de que las perdamos.

Como citamos anteriormente, para los alumnos empleamos dos tablas distintas: Users, creada por Django, es la que utilizamos para crear y almacenar la información relevante al inicio de sesión de los usuarios, aprovechando el cifrado automático de las contraseñas. La segunda tabla, Alumnos, almacena la información relevante a cada alumno, como por ejemplo su nombre de usuario, que empleamos para relacionar las dos tablas entre sí,el número de PINFCoins que posee o su carrera universitaria.

Las asignaturas quedan almacenadas en la tabla Asignaturas con su nombre y su código de ocho dígitos. En la tabla figuran todas las asignaturas de las ocho carreras a las que damos soporte, de modo que los alumnos no pueden actualizar notas o apostar sobre asignaturas que no figuren en nuestra base de datos, puesto que pertenecen a una carrera sin soporte.

La tabla AlumAsig recoge la relación entre una Asignatura y un Alumno que haya estado en algún momento matriculado de la misma. En ella figura si la asignatura está aprobada o no y con qué nota, y el número de monedas apostadas y la nota mínima a sacar en caso de que el alumno haya apostado sobre dicha asignatura.

La tabla Amistad incluye todas las relaciones de amistad entre usuarios de la página, relacionando a dos miembros de la clase Alumno entre sí y recogiendo en la variable solicitud 0 si la solicitud está pendiente de aceptar o 1 si ya se ha aceptado la solicitud de amistad. Si una solicitud es rechazada, el miembro de la clase se elimina.

La tabla Publicidad es en la cual se almacenan los banners de publicidad de la página junto con el enlace que publicitan.

3.6. Diagrama de Secuencia.

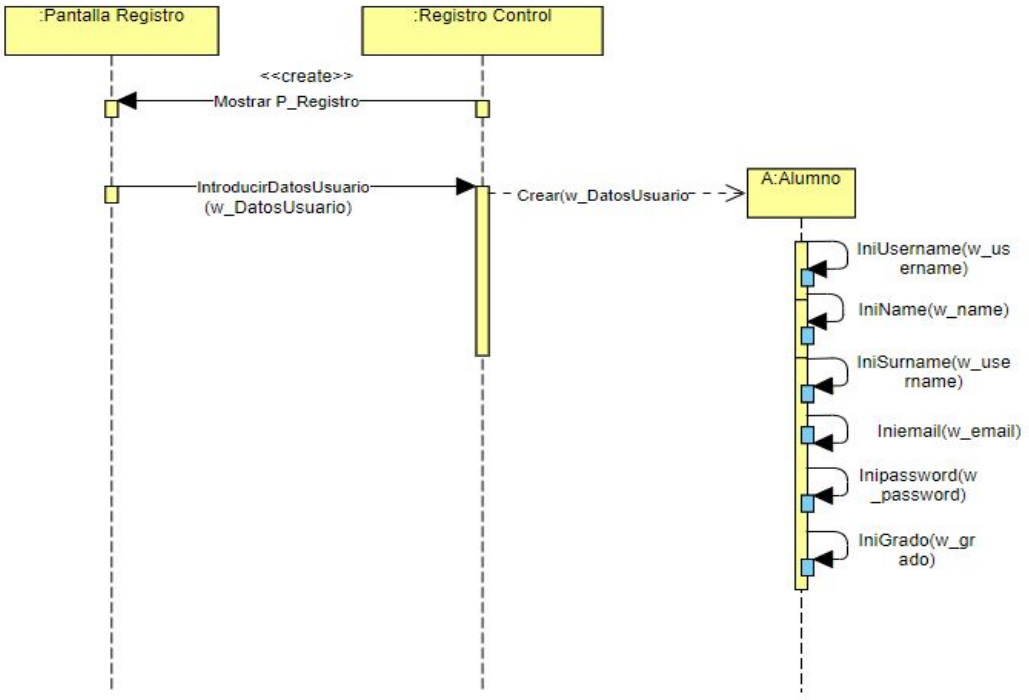


Diagrama de secuencia

Para el diagrama de secuencia hemos utilizado de ejemplo el caso de uso "Registro Usuario". Una vez que hacemos click en el botón "Registrarse", la interfaz de usuario cambiará, mostrándonos un formulario que tenemos que rellenar para poder ser usuarios de PINFBET. Una vez introducido todos los datos, serán enviados a la base de datos creándose así un objeto Alumno que contiene de atributos todos los datos que nos hayan pedido previamente.

3.7. Parametrización del software libre.

Para configurar correctamente el software que hemos desarrollado hemos utilizado funciones que el framework nos ha facilitado, una de ellas es dividir el uso de la página web en dos modos: administrador y usuario. Con el primer modo, podemos gestionar la base de datos de dicha web, además de poder dar o quitar permisos a ciertos usuarios. También se puede configurar varios parámetros que afectasen al funcionamiento del producto. Y el modo usuario es el modo con el que el usuario final puede usar la página, por ejemplo apostar, agregar amigos, etc.

4. IMPLEMENTACIÓN DEL SISTEMA.

4.1. Entorno tecnológico.

Para poder utilizar el sistema, necesitamos ejecutar el servidor de forma local y, haciendo uso del framework Django, lo ejecutamos para administrarlo y mantenerlo. Esto se haría desde nuestros propios equipos informáticos.

Y para usar este sistema, en referencia a su uso de cara al público/administrador, accederemos a la aplicación web a través un navegador de Internet.

La página web está alojada en un servidor web llamado Heroku.

4.2. Implementación.

La aplicación está implementada en su totalidad mediante el framework web de Django 3.1.5.

Django nos permite diseñar y mantener la base de datos de forma rápida y sencilla, implementándola en lenguaje Python en el fichero models.py. La base de datos es el pilar fundamental del funcionamiento de la página. Si ejecutamos la página web de forma local, utilizamos SQLite, mientras que si ejecutamos la página web desde el servidor de Heroku que tenemos habilitado se utiliza PostgreSQL.

Todas las funcionalidades de la página se realizan mediante vistas, cada una de ellas situadas en el fichero views.py. Cada vista se encarga de acceder a la base de datos y de obtener la información necesaria para una tarea concreta, y renderiza la plantilla html correspondiente (almacenadas en la carpeta templates), enviándole a las plantillas las variables que necesiten. Cada vista tiene asociada una url, definidas en el fichero urls.py.

4.3. Calidad de código.

En principio, nuestro código tenía partes duplicadas, así que, para mantener la calidad del código, hemos eliminado ese código creando un solo archivo (nav.html) que se extiende en los demás .html.

Además, hemos revisado el código muerto y lo hemos eliminado, por lo que nuestros archivos no contienen código que no se use.

5. PRUEBAS DEL SISTEMA.

5.1. Pruebas unitarias.

Dado que necesitamos hacer uso de Django y la base de datos para usar nuestras funciones, no podemos ejecutar el código por separado para hacer pruebas, así que probaremos las funciones de nuestra web sin aislar el código.

Realizaremos, como ejemplo, una prueba de nuestra función subir expediente. Como el nombre indica, debemos subir un expediente de la Universidad de Cádiz y la prueba consistirá en los siguientes casos:

Caso 1. No subir ningún archivo:

Si pulsamos el botón de subir sin adjuntar ningún archivo, obtenemos un mensaje de error: "No has subido ningún archivo". Esto es una excepción que hemos añadido a raíz de hacer la prueba y funciona correctamente.

Caso 2. Subir un archivo que no esté en formato PDF:

Al subir una imagen, la web nos muestra un mensaje de error (Expediente no válido) pero no se cuelga. Eso es porque hemos añadido la excepción y funcionará para cualquier archivo que no pueda ser transformado a formato excel.

Caso 3. Subir un archivo que esté en formato pdf, pero no sea un expediente de la UCA:

Si subimos un archivo que tenga formato pdf, nos aparece el mismo error que si subimos un archivo con otro formato, así que funciona correctamente, dado que se trata del mismo error.

Caso 4: Subir un archivo que esté en formato pdf y sea un expediente de la UCA:

Si subimos un expediente correcto, vemos como no nos da ningún mensaje de error y nuestro perfil se habrá actualizado añadiendo PINFCoins en función de las asignaturas que hayamos aprobado según el documento

Con la siguiente fórmula, calculamos las PINFCoins:

$$PINFCoins = ((Nota\ Mínima) * (Nota\ obtenida/10) * Cant\_Apostada + Cant\_Apostada)$$

5.2. Pruebas de integración.

Una vez comprobada cada función implementada, las probamos en su conjunto. Esta prueba consistirá en probar cada una de las opciones que nos aparecen en la página web, habiéndose probado por separado anteriormente.

Cuando ejecutamos la web, todo funciona correctamente. Se nos muestra la interfaz y cada botón responde adecuadamente sin problemas.

En definitiva, podemos acceder a cada una de las funciones que ya hemos mencionado, ya que funcionan todas en conjunto.

5.3. Pruebas de sistemas.

5.3.1. Pruebas funcionales.

Esta prueba es un tipo de caja negra que nos permite evaluar las funciones de modo que debemos planificar que tipo de posibles errores vamos a cometer a propósito para evaluar si está correctamente implementada o no la web.

En este caso, vamos a utilizar la funcionalidad Apostar. Para dicha función, podemos introducir distintos tipos de valores pero solo se debe admitir un tipo de valor que son monedas en formato de números naturales. Entonces para ello, hemos diseñado las siguientes pruebas:

Prueba 1. Introducir un número negativo en el número de monedas que apuesto:

Sea el caso en el que introducimos un número negativo a apostar para obtener más PINFCoins a base de aprovechar fallos del sistema. La primera vez que hicimos la prueba, nos añadía el número de monedas que introducimos pero en valor positivo. Entonces para ello

Prueba 2. Introducir una cadena de caracteres:

En este caso, si introducimos una cadena de caracteres en esta opción nos saltaba un error de django que nos mostraba que no se podía convertir ese dato en base 10 (es un error que salta en python concretamente).

Entonces para ello lo que hemos hecho es que la función compruebe antes de procesar el dato si el tipo de dato es el correcto y en el caso de que no lo sea, nos muestre un mensaje de error diciéndonos que no es válido

Prueba 3. Introducir 0:

En este caso probamos que apostamos 0 monedas para conseguir más monedas y no perder nuestras monedas.

Entonces, intentamos introducir la cantidad de 0 monedas la web nos mostrará un mensaje de error informándonos de que el número de monedas que debemos introducir debe ser superior a 0.

Prueba 4. Introducir un número superior al número de monedas que tenemos:

En este caso si fuésemos un usuario malicioso y directo, probaremos si pudiésemos apostar más PINFCoins de los que tenemos.

Entonces probamos si es posible y nos muestra un aviso de que no disponemos de la cantidad de PINFCoins suficientes para apostar.

Prueba 5: Introducir números con separación entre ellos.

Si queremos introducir un número con espacios entre ellos, la web nos detecta que hemos introducido una cadena de caracteres por lo tanto debemos comprobar si el tipo de dato es el correcto. Entonces, cuando introducimos una cadena de caracteres, la web nos muestra un mensaje de error informándonos de que el tipo de dato que hemos introducido no es válido.

5.3.2. Pruebas no funcionales.

Este tipo de pruebas recoge todos aquellos aspectos de gran importancia para nuestro proyecto, pero que no se tratan de sus funcionalidades. Es decir, se trata de todo aquello que no podemos englobar en un caso de uso. En este tipo de pruebas podemos encontrar: Prueba de carga, rendimiento y estrés. Al no disponer del personal suficiente para comprobar los límites que puede soportar la página, realizaremos solo las pruebas de carga.

Pruebas de carga

Mediante las pruebas de carga comprobamos el funcionamiento de nuestra página cuanto está sometida a un cierto número de usuarios.

En nuestro caso, al ser un grupo de 4 personas, solo podemos comprobarlo con ese máximo número de personas. La manera de comprobar esto es a través de la herramienta Live Share que proporciona Visual Studio Code, mediante la cual un usuario puede hostear la web.

Aún con 4 personas conectadas de PINFBET simultáneamente, la web funciona con normalidad.

Prueba escalabilidad

Añadir nuevas funcionalidades a nuestro proyecto es bastante sencillo. En primer lugar, tenemos que crear su vista correspondiente en “views.py”. Para que la vista correspondiente a la nueva funcionalidad funcione correctamente, tenemos que crear su HTML correspondiente, en caso de que tengamos que crear una nueva página que no se corresponde con ninguno de los HTML anteriormente creados en la carpeta “templates”. Además, tenemos que crear una nueva URL en “urls.py”. Finalmente, debemos actualizar la “nav” para permitir al usuario la navegación con la nueva funcionalidad que le hemos otorgado a nuestro proyecto. Por último, si es necesario crear una nueva base de datos o modificar una ya existente, podemos hacerlo modificando “models.py”.

5.4. Pruebas de aceptación.

En esta prueba lo que se analiza es si el producto resultante, la página web, cumple las necesidades del cliente y los usuarios. Como sabemos, el requerimiento inicial es poder realizar un sistema de apuestas para asignaturas en la cual los usuarios pudieran apostar las notas de sus asignaturas. Este sistema está implementado y es completamente funcional, además de la gestión de usuarios, expedientes y matrículas (obviamente reparando en la seguridad del sistema).

A partir de ahí, hemos pensado e implementado funcionalidades añadidas a la principal que complementen la experiencia, como, por ejemplo, un sistema amigos y un espacio para que nuestros usuarios, generalmente estudiantes, puedan publicitarse con el objetivo de tener un impacto positivo en la comunidad universitaria.

6. ANEXO - GUÍA DE USO

6.1 Acceso a la plataforma

6.1.1. Acceso a la plataforma desde un servidor local

En primera instancia, para ejecutar la web, debemos instalar el framework Django (https://www.djangoproject.com/download) que nos indican las instrucciones que debemos seguir.

En segunda instancia, debemos introducir el siguiente comando para instalar las librerías que son necesarias para poder ejecutar la web:

```
pip install -r requirements.txt
```

También es necesario tener instalado Java para poder utilizar todas las funciones del sistema.

Una vez instaladas las librerías y Django, podemos proceder a ejecutar el servidor local utilizando un comando que requiere Python desde el directorio raíz de la aplicación:

```
python manage.py runserver
```

Después de ejecutar el comando en la terminal y recibir la respuesta en la cual se nos informa de que servidor está en funcionamiento, basta con ir a nuestro navegador e introducir la ip local y añadirle **'/polls'**, dónde se encuentra la página principal de nuestra web.

6.1.2. Acceso a la página desde servidor web

El servidor web en el que alojamos la página debe tener instaladas todas las librerías mencionadas en el apartado 6.1.1, así como tener instalado Java. Entonces se puede acceder sin instalar nada, y se puede usar la página web sin ningún problema.

El servidor ya está alojado en un servidor web de Heroku, en la siguiente dirección: https://pinfbet.herokuapp.com. El servidor actual es completamente gratuito, por lo que está limitado a 18 horas de uso cada 24 horas y tiene acceso a memoria reducida, además de poner la aplicación en reposo si no hay actividad. Se recomienda actualizar a un plan de pago para mejorar los tiempos de respuesta de la página.

6.2 Registro de la cuenta de usuario y matriculación en asignaturas

Para comenzar a utilizar la plataforma, el usuario debe registrarse utilizando el botón situado en la esquina superior derecha de la página, introduciendo sus datos y aceptando los términos del servicio. Una vez registrado, inicia sesión y accede a su perfil, una vez más desde el panel superior.

Para darnos matricularse de asignaturas debemos utilizar el botón de "Subir nueva matrícula" y subir desde nuestro equipo un archivo .pdf de la matrícula del usuario, que habrá obtenido en su Portal Uca. Tras unos segundos el sistema importará la última matrícula del alumno a la base de datos y eliminará el fichero.

Para obtener las PINFCoins de las asignaturas aprobadas en cursos anteriores, se usará el botón "Subir nuevo expediente", de la misma manera que la matrícula. Nótese que este también debe ser obtenido desde el Portal Uca.

6.3 Uso del servicio de apuestas

6.3.1. Realizar apuestas

Para realizar apuestas referentes a las asignaturas matriculadas del usuario, éste deberá acceder al menú de apuestas situado en el panel izquierdo de la pantalla y seleccionar la asignatura sobre la cual desea apostar.

Acto seguido el usuario introducirá la nota mínima a obtener en la apuesta y las PINFCoins a apostar. Nótese que estos valores no podrán ser reducidos más adelante, ni la apuesta podrá ser cancelada.

6.3.2. Modificar apuestas realizadas

Las apuestas del usuario, así como las de sus amigos, podrán ser visualizadas desde el menú principal una vez que el usuario inicie sesión. Al clicar sobre una de sus apuestas, el usuario accede a una pantalla resumen en la cual se le da la opción de modificar la apuesta.

El usuario volverá a introducir una nota mínima y una cantidad de PINFCoins a apostar, que deberán ser superiores o iguales a las introducidas con anterioridad para que el sistema las acepte.

6.3.3. Comprobar resultado de la apuesta

Para comprobar si el usuario ha superado o no la apuesta y recibir su recompensa, se deberá hacer uso del botón de "Subir nuevo expediente" situado en el perfil del usuario.

Si en el expediente subido figura la asignatura sobre la que se ha apostado, se cierra la apuesta y se asignan las PINFCoins correspondientes al usuario según su nota.

La cantidad de PINFCoins recibida es directamente proporcional a la nota obtenida en la asignatura, en la cantidad apostada, y en la nota mínima objetivo de la apuesta.

6.4 Uso del servicio de amistad

6.4.1 Buscar usuarios y enviar solicitudes de amistad

Para hacer uso del sistema de amistad, tendremos que buscar un usuario. Para esto, accedemos a la opción que nos aparece en el panel lateral izquierdo. Buscar amigos funciona con el nombre de usuario, y basta con introducir una letra o parte del nombre de usuario para que nos liste todos los que la contengan. Una vez buscado, podemos acceder al usuario haciendo clic encima de su nombre, mostrando así su perfil y una opción para enviar una solicitud de amistad. Si enviamos dicha solicitud, será el otro usuario quien tenga que aceptar o rechazar la misma.

6.4.2 Aceptar solicitudes de amistad

Para ver aquellas solicitudes que le han enviado otros usuarios, el usuario debe acceder a la opción “Solicitudes de amistad” que aparece en la barra lateral izquierda. Desde ahí verá si tiene alguna solicitud pendiente y podrá aceptarla o rechazarla mediante los botones que aparecen junto al nombre del usuario que la envió.

6.4.3 Acceder a la lista de amigos

Por último, estos usuarios aceptados, o bien, aquellos que acepten al usuario, aparecerán en lista de amigos, otra opción situada en el menú lateral. Desde ahí el usuario puede acceder a los perfiles de sus amigos igual que en la función de búsqueda, con la diferencia de que ya no aparecerá el botón de enviar solicitud dentro de sus perfiles puesto que ya hay una relación de amistad establecida.

6.5 Uso del Panel de Control (Exclusivo para Administradores)

En caso de ser administrador del sistema, el usuario dispondrá de una nueva funcionalidad en la web. En la barra superior de PINFBET aparecerá un nuevo botón, al lado de “Cerrar Sesión” y “Ver Perfil”, mediante al cual podemos acceder al panel de administración de bases de datos que nos brinda Django. Aquí dentro, el administrador tendrá la capacidad de añadir, modificar o eliminar objetos de las tablas de la base de datos ya existentes en el sistema.

La aplicación ofrece unas credenciales de inicio de sesión para un administrador genérico con las siguientes credenciales:

Username: admin

password: pinfbetadmin

Esta cuenta no tiene ningún perfil de usuario asociado, y deberá usarse exclusivamente para crear cuentas de superusuario nuevas para los administradores del sistema durante la configuración inicial, siendo eliminada justo después o modificando la contraseña desde el propio panel.