MEMORIA DE LA PRÁCTICA 2

INTRODUCCIÓN

En este documento se describe con detalle los scripts y la estructura de la base de datos que conforman este proyecto.

LISTADOS DE SCRIPTS PARA LAS VISTAS

Se ha decidido utilizar un script PHP para cada vista, dando como resultado la siguiente lista clasificada según la funcionalidad o el tema.

Inicio de sesión y creación de cuenta.

- login.php: muestra una vista desde la que el usuario puede iniciar sesión. Acompañando a este script está logout.php que permite al usuario cerrar sesión, no es como tal un script de vista.
- register.php: muestra una vista desde la que el usuario puede completar un formulario para crearse una cuenta.

Subida de kernels.

- your_kernels.php: muestra una vista exclusiva para usuarios registrados en la que hay información sobre los kernels subidos por él y un botón para añadir uno nuevo.
- |kernel_form.php|: muestra un formulario para completar la información se un kernel que se quiere añadir. Está conectado con el script your_kernels.php a través de un botón.

Ejecución de kernels.

- kernel_marketplace.php: muestra una vista de mercado donde el usuario puede ver las ofertas de kernels.
- kernel_info.php: muestra toda la información referente a un kernel. Se conecta con kernel_marketplace.php a través de un botón.
- |kernel_execution.php|: es una vista auxiliar a kernel_info.php que se muestra cuando el usuario decide ejecutar un kernel.

Valoraciones.

- kernel_info.php: junto a la información mostrada del kernel hay una sección que permite hacer valoraciones.

Gestión de tokens.

- wallet.php: muestra una vista exclusiva para usuarios registrados que da información detallada sobre la cartera de tokens: historial de transacciones,

gráficas, ... Junto a la información hay dos secciones con botones para retirar tokens o añadir tokens.
- token_transaction.php: muestra una vista auxiliar para completar la transacción deseada (retirar o añadir tokens) a través de un formulario.

Ranking.

- ranking.php: muestra una vista que lista a los usuarios con mayor participación dentro de la aplicación. Si estás registrado, se mostrará tu posición, aunque no estés en los primeros

Gestión de administrador.

- admin_dashboard.php: muestra una vista exclusiva a usuario con rol de administrador que permite realizar todas las acciones de administrador.

Vistas complementarias.

- index.php: muestra la vista principal de la página, lo que primero se muestra a un usuario que no esté registrado.
- FAQ.php: muestra una vista para preguntar cuestiones sobre la página a través de un buscador o por una lista de preguntas dividida en secciones.
- contacto.php: muestra una vista desde la que se puede rellenar un formulario para sugerencias, quejas o cuestiones. Además, se muestran enlaces a las distintas redes sociales de la empresa.
- user_dashboard.php: muestra una vista exclusiva para usuarios registrados. Es la vista que primero se le muestra a un usuario registrado. En ella hay información resumida sobre la cuenta.
- settings.php: muestra una vista exclusiva para usuarios registrados donde se puede modificar datos de la cuenta o del perfil público.
- profile_view.php: muestra una vista que da detalles públicos sobre un usuario.

Componentes.

- nav_bar.php: genera una barra de navegación que permite acceder a las páginas públicas para cualquier usuario: index.php, kernel_marketplace.php, ranking.php, FAQ.php y contacto.php. Además, dependiendo de si se está registrado o no, se mostrarán botones para iniciaar sesión y crear cuenta, o se mostrará el nombre del usuario registrado y un acceso al user_dashboard.php.
- user_nav_bar.php: genera una segunda barra de navegación exclusiva para usuarios registrados que permite navegar por las vistas de usuario: user_dashboard.php, wallet.php, your_kernels.php y settings.php. Se muestra sólo en esas vistas.
- footer.php: genera un pie de página para dar mayor accesibilidad al usuario.
- token_big_info.php: genera una sección que da detalles sobre la cartera de tokens de un usuario.
- transaction_graph.php: genera una gráfica que resume las transacciones realizadas por el usuario.
- transaction_table.php: genera una tabla con las trasacciones realizadas por el usuario.

LISTADOS DE SCRIPTS ADICIONALES

Como la gestión de la base de datos es compleja, se ha decidido separar la funcionalidad en clases:

- Aplicacion.php: contiene la funcionalidad básica para mantener una única conexión con la base de datos.
- Usuario.php: contiene toda la funcionalidad directamente relacionada con el usuario.
- Kernel.php: contiene toda la funcionalidad directamente relacionada con los kernels.
- Comentario.php: contiene toda la funcionalidad directamente relacionada con los comentarios.
- Transaction.php: contiene toda la funcionalidad directamente relacionada con las transacciones (retirada o adición de tokens).

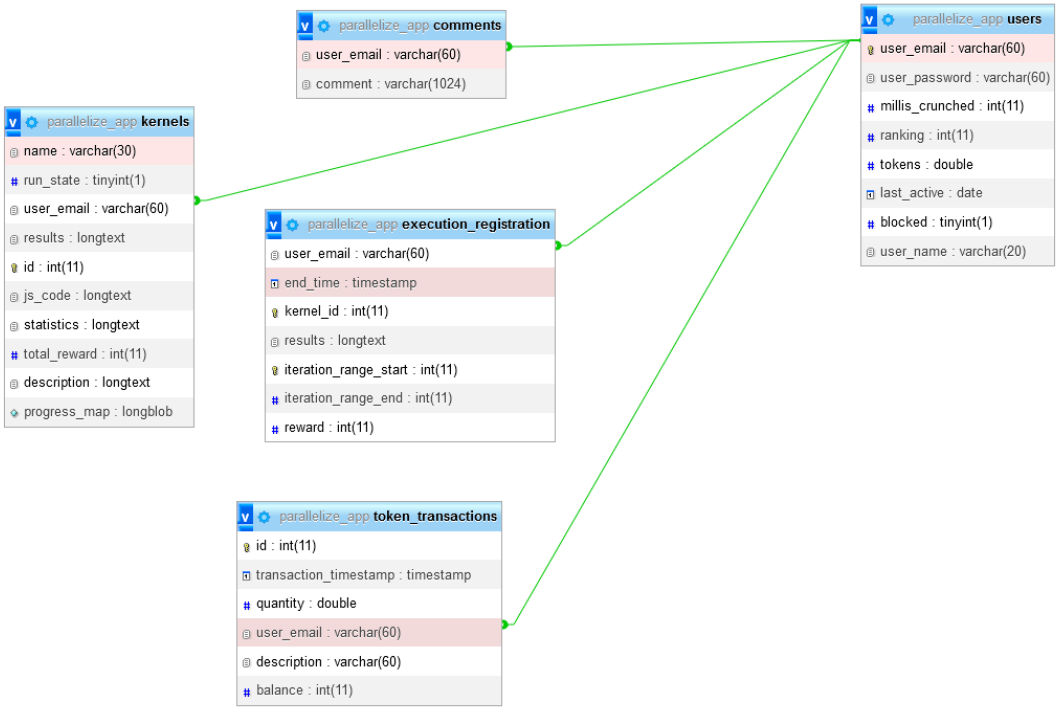
Además de estas clases, se ha decido crear una familia de clases que heredan de Formulario.php, lo que permite organizar y modificar fácilmente todos los tipos de formularios utilizados.

- Formulario.php: clase abtracta de la que heredan el resto de formularios. Con tiene la funcionalidad compartida entre todos los formularios.
- FormularioContacto.php: permite crear y gestionar un formulario para la página de contacto.
- FormularioLogin.php: permite crear y gestionar un formulario para la página de login.
- FormularioRegister.php: permite crear y gestionar un formulario para la página de creación de cuenta.
- FormularioTransaction.php: permite crear y gestionar un formulario para la página de token_transaction.php.

Por último, también se ha decidio utilizar un script PHP con funcionalidad axuliar.

- config.php: permite reducir la repetición de código común entre los script y que permite iniciarlizar la sesión y otros valores como constantes.

ESTRUCTURA DE LA BASE DE DATOS



Explicación de las tablas

- users
- comments: registra los comentarios enviados por la funcionalidad de contacto. La clase comentario maneja las lecturas y escrituras de esta tabla.
    - user_email: apunta al usuario autor
    - comment: el contenido del comentario
- kernels: guarda los kernels que han sido subidos por los usuarios y la información acerca de ellos. La clase Kernel maneja las lecturas y escrituras de esta tabla.
    - name: el nombre del kernel
    - run_state: determina si el kernel ha terminado del todo
    - user_email: apunta al usuario autor
    - results: los resultados intermedios de la ejecucion
    - id: identificador unico
    - js_code: el codigo a ejecutar
    - statistics: datos de la velocidad de cada iteracion
    - total_reward: el valor de recompensa total ofrecida
    - description: un texto que describa el funcionamiento del código
    - progress_map: mapa de bits que determina que iteraciones están ejecutadas y cuales no.
- token_transaction: registra cada transacion entrante o saliente de tokens para poder mostrar la evolución a lo largo del tiempo de tu balance. La clase Transaction maneja las lecturas y escrituras de esta tabla.
    - id: identificador unico
    - transaction_timestamp: segundo en el que se ejecutó la transación
    - quantity: cantidad de tokens introducida o extraida

    - user_email: usuario beneficiado
    - description: breve texto explicando la razón del movimiento
    - balance: balance resultante
- execution_registration: lleva un registro de todas las veces que un usuario ejecuta un kernel para poder repartid las recompensas, se guarda en el momento que el cliente devuelve los datos resueltos. La funcionalidad no está implementado.
    - user_email: el usuario responsable del calculo
    - end_time: el momento cuando se devolvieron los datos
    - kernel_id: el kernel ejecutado
    - iteration_range_start: por que iteración ha empezado a ejecutar
    - iteration_range_end: que iteración ha sido la última en ejecutar
    - reward: cuantos créditos han sido transferidos

SOBRE EL PROTOTIPO FUNCIONAL

Uso.

El prototipo tiene una carpeta 'sql' que contiene los archivos necesarios para importar la base de datos. Una vez importada se puede probar directamente las funcionalidades con las cuentas ya creadas: jaime2@gmail.com (contraseña: aaaaaaaa) y jaime@gmail.com (contraseña: aaaaaaaa). También es posible crear cuentas de usuario nuevas

Estructura

En la raíz de la carpeta donde se almacena el prototipo ('public') se pueden encontrar todos los scripts utilizados para la vistas y una serie de carpetas con recursos que usarán dichos scripts: 'css', 'img', 'includes', 'js' y 'svg'. La carpeta includes contiene los scripts auxiliares y las carpetas 'dao' (clases de la base de datos), 'formularios' y 'vistas' (secciones reutilizables).