# Informe técnico

Aylín Álvarez Santos C-312
Grettel Hernandez Garbey C-311
Carlos Toledo Silva C-311

## Índice

# 1. Aplicación main

Esta aplicación es la que permite el mostrado de los objetos que se aprecian en la página principal.

# 2. Aplicación admins

Esta aplicación contempla el registro de administradores. Para registrar un administrador se utiliza un formulario que hereda del formulario que trae Django para el registro de usuarios. Los campos que se utilizan son los de nombre de usuario, nombre, apellidos, email y contraseña. El modelo que se utiliza es similar al modelo de usuario de Django. Se utiliza la vista basada en clase `AdminCreate`, la cual pinta el formulario en el template `registrar\_user.html` y después guarda la información que recibe en la base de datos y otorga a dicho usuario los permisos correspondientes. También está la vista basada en clase `AdminList` con la cual se obtiene la lista de administradores y se refleja en el template `admins.html`. Otra vista basada en clase que se utiliza es `AdminDelete` la cual se utiliza parra eliminar a un administrador del sistema. Por último está la vista basada en función `list` que obtiene de la base de datos a los profesores y los estudiantes de una determinada asignatura y pinta estas relaciones en el template `subject_admin.html`.

# 3. Aplicación professors

Esta aplicación contempla el registro de profesores. Para registrar un profesor se utiliza un formulario que hereda del formulario que trae Django para el registro de usuarios. Los campos que se utilizan son los de nombre de usuario, nombre, apellidos, email y contraseña. El modelo que se utiliza es similar al modelo de usuario de Django. Se utiliza la vista basada en clase `ProfessorCreate`, la cual pinta el formulario en el template `registrar\_user.html` y después guarda la información que recibe en la base de datos y otorga a dicho usuario los permisos correspondientes. También está la vista basada en clase `ProfessorList` con la cual se obtiene la lista de profesores y se refleja en el template `professors.html`. Otra vista basada en clase que se utiliza es `ProfessorDelete` la cual se utiliza parra eliminar a un profesor del sistema.

# 4. Aplicación students

Esta aplicación contempla el registro de estudiantes. Para registrar un estudiante se utiliza un formulario que hereda del formulario que trae Django para el registro de usuarios. Los campos que se utilizan son los de nombre de usuario, nombre, apellidos, email y contraseña. El modelo que se utiliza es similar al modelo de usuario de Django, pero además se le agega un campo que indica la cantidad de puntos(créditos) obtenidos por el estudiante. Se utiliza la vista basada en clase `StudentCreate`, la cual pinta el formulario en el template `registrar\_user.html` y después guarda la información que recibe en la base de datos y otorga a dicho usuario los permisos correspondientes. También está la vista basada en clase `StudentrList` con la cual se obtiene la lista de estudiantes y se refleja en el template `students.html`. Otra vista basada en clase que se utiliza es `StudentDelete` la cual se utiliza parra eliminar a un estudiante del sistema.

# 5. Aplicación users

Esta aplicación solo contempla la vista basada en clase `UserUpdate` cuya funcionalidad es que cualquier tipo de usuario pueda editar su información personal.

# 6. Aplicación elements

En esta aplicación se define el modelo `Element`.

# 7. Aplicación basic_elements

Esta aplicación contempla las operaciones relacionadas con los elementos básicos.

Se define el modelo `BasicElement` que hereda de `Element` y el formulario `BasicElementCreateForm`.

Primero se define la vista basada en función `basic_element_list` que obtiene de la base de datos todos los elementos básicos de una determinada asignatura y muestra esta información el template `basic_elements.html`. Luego se tiene la vista basada en clase `BasicElementCreate` que pinta el formulario en el template `create_basic_element.html` y luego de llenarse los campos del formulario guarda la información en la base de datos. La vista basada en función `make\_visible` permite hacer visble un elemento determinado para los estudiantes, o sea que lo puedan utilizar para hacer mezclas. Por último se tienen la vista basada en clase `BasicElementEdit` y la vista basade en función `delete_basic_element`, cuya funcionalidades son editar y eliminar un elemento básico que no esté visible respectivamente.

# 8. Aplicación non_basic_elements

Esta aplicación contempla las operaciones relacionadas con los elementos no básicos.

Se define el modelo `NonBasicElement` que hereda de `Element`.

Primero se tiene la vista basada en función `accepted_list` que dado un estudiante y una asignatura obtiene los elementos creados por el estudiante en la asignatura y refleja esa información el template `accepted_list.html`. La siguiente vista basada en función hace algo similar, lo que para los elementos que están pendientes de aceptación o rechazo y la información se muestra en el template `pending_list.html`.

Luego se tiene la vista basada en función `create_non_basic_element` la cual utiliza la función `create_form` que devuelve un formulario `nonBasicElementCreateForm`. La función de esta vista es la de crear elementos no básicos por parte de los estudiantes. El formulario obtenido se pinta en el template `create_non_basic_element.html` y la información introducida se guarda en la base de datos.

Luego tenemos las vistas basadas en funciones `reject_non_basic_element` y `accept_non_basic_element` cuya funcionalidad es permitir a los profesores rechazar y aceptar las solicitudes de mezclas hechas por los estudiantes.

# 9. Aplicación subjects

Esta aplicación contempla las operaciones relacionadas con las asignaturas. En esta aplicación se define el modelo `Subject` y el formulario `SubjectCreateForm`.

Primero aparece definida la vista basada en clase `SubjectList` cuya funcionalidad es obtener de la base de datos todas las asignaturas si el usario actual es un administrador o un estudiante. Si es un profesor solo se recuperan las asignaturas que dicho profesor imparte. Esta información se pinta en el template `subjects.html`. Luego aparecen las vistas basadas en clases `SubjectCreate`, `SubjectUpdate` y `SubjectDelete` las cuales se utilizan para crear, editar y eliminar una asignatura respectivamente. En caso de la creación y edición se utiliza el formulario `SubjectCreateForm`. Por último la vista basada en función `professor_or_student_or_admin` es la que se encarga, cuando se quiere acceder a la página de una asignatura, según sea el usuario redireccionarlo a la página correcta. Para el caso de un estudiante que trate de acceder a la página de una asignatura, si este no está matriculado lo lleva a la página de matrícula y en caso de estarlo entra a la página de la asignatura.

# 10. Aplicación imparts

Esta aplicación contempla las operaciones relacionadas con las relaciones de imaprtir una asignatura por parte de algún profesor. En esta se definen el modelo `Imparts` y el formulario `ImpartsCreateForm`.

Las primeras vistas basadas en función que se pueden apreciar son `create_imparts` y `delete_imparts` la cuales se utilizan para crear y eliminar estas relaciones respectivamente. Los templates utilizados por las mismas son `create_imparts.html` y `delete_imparts.html`. La última vista basada en función `request_list` obtiene las solicitudes de mezclas que hacen los estudiantes en una asignatura y las muestra en el template `subject_professor.html`.

# 11. Aplicación study

Esta aplicación contempla las operaciones relacionadas con las relaciones de estudiar/cursar una asignatura por parte de algún estudiante.

La vista basada en función `subject_student` permite el acceso de un estudiante a la página de una determinada asignatura cargando sus respectivos datos y mostrándolos en el template. `subject_student.html`. La vista basada en función `enroll` tiene como funcionalidad permitir a un estudiante matricularse en una determinada asignatura. Por su parte la vista basada en función `delete_study` tiene como funcionalidad eliminar la relación de que un estudiante cursa una determinada asignatura.

# 12. Aplicación queries

Esta aplicación contempla todo lo relacionado con las consultas, desde su implementación a nivel de modelos, las vistas y los templates. Para el desarrollo de la aplicación, se tiene una clase abstracta Query que representa una consulta que se efectúa a través de su método execute(context), el cual devuelve una instancia de context que es un diccionario, donde están los elementos para enviar al template. Se tiene una clase concreta para cada consulta del sistema. Además se aplicó el patrón de diseño Factory para la construcción de las instancias de las clases que heredan de Query, de forma dinámica. Luego se tiene un formulario para cada consulta del sistema. Los formularios se envián al servidor utilizando el método POST, y se toma el nombre del formulario de un campo oculto en el template. Se construye una instancia de la clase Query correspondiente utilizando

las bondades de Factory. Luego se ejecuta la consulta y se devuelve la tabla respuesta, la cual es almacenada en el diccionario context, y se renderiza hacia el template "queries_result.html".

# 13. Diseño visual de la aplicación

Para el frontend de la aplicación se utilizó el propio framework Django. Se agregaron estilos a las vistas para un manejo agradable de la aplicación. Los archivos estáticos como imágenes, CSS y fuentes podrán ser encontrados en la carpeta `static`.