

Documentación Front-End

Índice

- 1. Introduccion**
- 2. Template**
 - a. Archivos de proposito general.**
 - b. layout.**
 - c. student.**
 - d. teacher.**
 - e. admin**
- 3. Static**
- 4. Atribuciones**

1 Introducción: el presente documento tiene la funcionalidad de resumir la funcionalidad e interconexión entre los archivos que se usan en el front end de la plataforma épsilon, los cuales están divididos en dos carpetas principales templates y static, estas a su vez se dividen en más carpetas, pero la idea general es que template contiene todos los html y javascript, y static contiene todas las imágenes y css.

Toca tener en cuenta lo siguiente, sintaxis de jinja está presente en todos los html en particular se usa para desplegar todos los listados o tablas, y los javascript están todos dentro de un elemento script en sus respectivo html, es decir que no hay una selección distinta de archivos donde se guarden los javascripts, además se usa el sistema de herencia de flask para poder organizar mejor el código.

2 Template: esta carpeta contiene varios archivos, no vamos a entrar en detalle sobre la función de cada archivo a excepción de unos cuantos de especial importancia.

La estructura de esta carpeta es bastante simple, hay 4 archivos en esta carpeta y 3 carpetas dentro de esta carpeta, cada carpeta destinada a cada uno de los tipos de usuario final de la plataforma.

Durante esta sección solo vamos a hacer recencia a archivos de html por lo tanto vamos a omitir la terminación “.html”, todos los archivos usan el stylesheet “styletemplate.css” a excepción de unas pocas excepciones, en dichas excepciones se hará explicito este hecho en la descripción.

a) archivos de propósito general: vamos a hablar de los más relevantes.

1. login: este archivo es el que se usa para renderizar el login de la aplicación, es uno de los pocos html en el que se usan stylesheet externas, posee un elemento form para usuario y contraseña, esta información es después usada para realizar la validación de usuario, además en la parte inferior donde dice “olvido su contraseña” hay una conexión con forget_passwd.
2. forget_passwd: es un formulario que se invoca cuando el usuario da click en “olvido su contraseña” requiere completar un campo en el cual se pide el correo electrónico, es uno de los archivos que usa stylesheet externas.
3. change_passwd: es un formulario el cual todo usuario que se logea por primera vez debe rellenar, se pide completar dos campos, la contraseña nueva y la confirmación de la contraseña nueva, al igual que el ítem anterior este archivo usa stylesheets externas.
4. admin_teacher: cuando alguien que tiene tanto rol de profesor como rol de administrador se logea en la plataforma, este html se renderiza mostrando en pantalla la opción de ingresar como administrador o como profesor, este archivo se encuentra dentro de la carpeta admin.

b) Layout: en esta carpeta están los html que se usan de template en toda la pagina, contiene solo 3 archivos uno por cada tipo de usuario, nombrados de la forma “estructura”+tipo_usuario.html donde tipo_usuario varia entre estudiante, profesor y administrador.

1. estructura_estudiante: contiene la estructura básica para todas las paginas a la acceden los estudiantes, es decir, un botón para ver y editar los datos personales, un botón para ver el historial académico, un botón de logout y una campana de notificaciones.
2. estructura_profesor: es un template para todas las páginas que usan los profesores, contiene un botón para acceder a los cursos que el profesor está dictando actualmente, un botón para poder los cursos que se han dictado en el pasado y un botón de logout.
3. estructura_administrador: determina la forma de todos los html para el administrador contiene, un botón para ver los estudiantes, un botón para los profesores, un botón para ver las distintas asignaturas, un botón de administración de la plataforma, un botón de logout y la campana de notificaciones.

Cabe resaltar que todas las páginas que sean accesibles desde algún botón de estos archivos es accesible desde cualquier página que herede el botón de alguno de estos archivos

c) Student: carpeta que contiene todos los html para el tipo de usuario estudiante, en esta carpeta todos los html estan conectados entre sí al dar click en alguno de sus botones, todo está basado en el template estructura_estudiante.

1. main_student: es lo primero que ve un estudiante al entrar a la página, está conectado a traves de botones y tags "<a>" con los archivos student_alert, student_data y academic_history.
2. student_alert: se accede a él dando click en la campana de notificaciones, le muestra al estudiante cuales son las alertas que se le han generado en forma de tabla, está conectada con main_student y todas las paginas a las que se conectan con main_student.
3. student_data: le muestra al estudiante su informacion personal, también la posibilidad de actualizarla, tiene la misma conectividad que main_student.
4. academic_history: muestra los distintos periodos académicos en que un estudiante ha tomado clases, al dar click en alguno de estos periodos se redirige a period_classes en donde están las notas de ese periodo académico.

5. `period_classes`: tiene un contenido casi idéntico al de `main_student` con la misma conectividad, pero la tabla de notas corresponde al periodo que se seleccionó en `academic_history`.

d) Teacher: aquí podemos encontrar todos los html que se usan para el tipo de usuario profesor, cabe resaltar que cada una de estos html al estar basados en `estructura_profesor`,

1. `main_teacher`: es lo primero que el profesor después de hacer login, y le muestra al profesor una lista con los cursos que está dictando, al seleccionar uno de estos cursos se redirecciona a `class.html`, está conectado con `historic_teacher`.
2. `class`: se renderiza cuando un profesor selecciona uno de sus cursos en `main_teacher`, muestra una tabla con los estudiantes y las notas del curso seleccionado, permite subir un archivo tipo Excel al sistema, además al darle click al botón de editar se redirecciona hacia `class_edit` para editar las notas en la tabla.
3. `class_edit`: tiene un contenido casi idéntico a `class`, la diferencia principal es que los inputs en las casillas de las tablas son editables.
4. `class_history`: muestra una lista de los distintos periodos en que el profesor ha dado clase, se accede a través del botón que dice “historial de cursos”, al seleccionar alguno de estos periodos se redirecciona hacia `historic_teacher`.
5. `historic_teacher`: tras haber seleccionado un periodo en `class_history`, `historic_teacher` despliega la lista de asignaturas que el profesor dictó en ese periodo, al seleccionar alguna de ellas, se redirecciona hacia `historic_class`.
6. `historic_class`: muestra una tabla con los estudiantes y las notas de la asignatura que se seleccionó en `historic_teacher` en el periodo que se clickeo en `class_history`, es posible descargar esta tabla en formato Excel, sin embargo, no es posible editarla.

e) Admin: contiene todos los html para generar las vistas de los administradores, todo está basado en el template `estructura_administrador`, esta es la carpeta con más elementos dentro de la carpeta `templates`, contiene 2 subcarpetas `student` y `teacher` (no confundir con las carpetas `student` y `teacher` que están al mismo nivel que la carpeta `admin`) con el objetivo de poder mostrarle al administrador las cosas que necesita saber sobre profesores o estudiantes, de la gran cantidad de elementos que tiene esta carpeta hay muchos en los que es suficiente con una corta explicación, pues son elementos bastante sencillos o funcionan de manera idéntica a elementos ya mencionados.

1. `main_admin`: es lo primero que ve un administrador tras hacer login e ingresar como administrador.

2. `admin_students`: se accede dando click en el botón de estudiantes, muestra un listado de los estudiantes y aparece una barra de búsqueda en la parte superior de la pantalla, esta barra de búsqueda solo permite buscar estudiantes, al escribir en ella se hace un autocompletado con fuzzy search mostrando diversas opciones, al dar click en el botón de buscar, el listado de estudiantes se va a modificar y va a mostrar únicamente las recomendaciones del fuzzy search, al dar click en un estudiante el admin va a pasar a una interfaz muy parecida a la del estudiante, los html de dicha interfaz se guardan en la subcarpeta `student`, la interacción entre ellos es esencialmente igual a los html en la carpeta `student` (la que está por fuera de la carpeta `admin`) que ya fue explicada.
3. `admin_teachers`: se accede al dar click en el botón de profesores, muestra un listado de los profesores, al seleccionar alguno de los profesores se hace un llamado a `main_teacher`, que se ubica en la subcarpeta `teacher`.
4. `history_classes`: se renderiza cuando el administrador da click en el botón de materias, muestra un listado de los distintos periodos académicos, al seleccionar alguno de los periodos redirecciona hacia `admin_classes`.
5. `admin_classes`: tras haber seleccionado un periodo académico en el ítem anterior, se muestra un Listado de todas las materias dictadas en dicho periodo, además se añade una barra de búsqueda con autocompletado, este autocompletado está basado únicamente en hacer match con el prefijo del contenido de la barra de búsqueda, además toda la funcionalidad de dicha barra de búsqueda está completamente limitada a materias, en cuanto a los elementos de la lista, estos tienen dos funcionalidades, si se hace click en el nombre de la materia, entonces se hace redirección hacia `admin_groups`, si se hace click en el icono de grafica de líneas entonces se carga con la respectiva asignatura como parametro `groups_report`.
6. `admin_groups`: se muestra tras haber hecho click en el nombre de una asignatura en el ítem anterior, pone en pantalla una tabla que muestra el grupo y el profesor a cargo en la asignatura seleccionada (en el periodo académico seleccionado en el ítem 5), si se hace click en el nombre de algún profesor entonces se carga el documento `admin_show_class`.
7. `admin_show_class`: muestra una tabla de estudiantes y notas en el grupo a cargo del profesor seleccionado en el ítem anterior de acuerdo a las selecciones de materia y periodo académico necesarias para llegar a este punto desde el ítem 5, aparece un pequeño título describiendo la tabla y un botón para descargar la tabla en formato Excel.

8. `groups_report`: se accede dando click en el icono de graficas de línea en el ítem 5, nos muestra una gráfica de distribución de las notas, con un código de colores para los distintos grupos de la asignatura, también aparece un botón para descargar esta grafica.
9. `admin_admin_functions`: aparece en pantalla al clicar el botón de administración, muestra las 3 funciones de manejo de la plataforma que el admin posee, es decir, importar información de profesores, importar información de un periodo académico y la creación de un nuevo usuario, las primera dos opciones redireccionan a `import_data_from_file` pero con url y argumentos de jinja distintos, se hace 3 llamados distintos a este documento, el primero cuando se va a importar información de profesores, en ese caso el documento agarra un `if` (de jinja) que le permite mostrar en pantalla un texto describiendo la acción a realizar un botón para seleccionar el archivo a subir y un botón para subir el archivo una vez seleccionado, si esto se hace correctamente se considera un proceso de importación de datos exitoso y se hace un llamado a `import_sucess`, la otra instancia de este documento ocurre al importar datos de un periodo académico, en este caso un condicional nos permite mostrar en pantalla un texto describiendo la acción a realizar y dos barras para especificar el periodo académico, además de los botones de seleccionar y subir archivo, si este proceso se hace exitosamente entonces ocurre la última instancia de este documento donde se vuelve a usar para importar un archivo con la información de materias y de ahí se hace el llamado a `import_sucess` en caso de que el proceso sea exitoso.
10. `Import_sucess`: se muestra tras cada proceso de importación de datos exitoso, solo contiene un aviso que dice que los datos se subieron exitosamente.
11. `admin_alert`: se muestra en pantalla al dar click en la campana de notificaciones, nos deja ver una tabla con las alertas que hay actualmente en el sistema, en la parte inferior tenemos un botón para que los administradores puedan publicar alertas personalizadas al clicar en él se redirección a `admin_create_alert`.
12. `create_user`: aparece cuando al admin selecciona crear usuario en `admin_admin_functions`, nos muestra un dropdown donde se puede seleccionar el tipo de usuario a crear y botones para seleccionar y subir un archivo del usuario a crear, si el proceso es exitoso se invoca a `create_user_success`.
13. `main_teacher`: se encuentra en la subcarpeta teacher, se accede como se indica en el ítem 3, nos muestra un listado con los periodos académicos en que el profesor en cuestión ha dictado clase al seleccionar alguno de estos periodos se renderiza el archivo `historic_teacher` de la subcarpeta teacher.
14. `historic_teacher`: se encuentra en la subcarpeta teacher y se accede como se menciona en el ítem anterior, nos muestra una lista con las materias que el profesor dicto en el periodo

seleccionado, en dicho listado cada elemento tiene dos elementos clickeables, el nombre de la asignatura y el icono de grafica de líneas, al hacer click en el nombre de la asignatura se muestra `historic_class` de la subcarpeta `teacher`, si se da click en el icono de grafica de líneas entonces se renderiza `class_report`.

15. `historic_class`: se accede como se muestra en el item anterior, contiene una tabla con los estudiantes y las notas respectivas al grupo, periodo y asignatura seleccionados previamente.
16. `class_report`: se accede como se indica en el item 14 y muestra una gráfica de distribución de las notas para el grupo, periodo y asignatura seleccionados previamente, además existe un botón para poder descargar dicha grafica.
17. `create_user_success`: es renderizado en el momento en que una operación de creación de usuario es exitosa, contiene un aviso indicando esto.
18. `admin_create_alert`: formulario para la creación de una alerta, contiene una barra para ingresar el destinatario, un dropdown para seleccionar el nivel de gravedad de la alerta y una descripción de la misma, si los datos son ingresados de manera correcta nos envía a `alert_success`.
19. `alert_success`: es invocado cuando un proceso de publicación de alerta personalizada es exitoso, contiene un aviso indicando eso.

3 Static

Dentro de esta carpeta se encuentran las imágenes y los css, al abrir la carpeta vemos que contiene dos archivos y dos carpetas, a continuación describimos los archivos.

1. `fondo.jpg`: esta imagen es la que se usa de fondo en toda la plataforma sin excepciones.
2. `styletemplate.css`: es el stylesheet de la mayor parte de documentos html

Prosiguiendo con las carpetas

1. `login`: contiene todo lo necesario para poder mostrar los archivos `login.html`, `forget_passwd.html` y `change_passwd.html`, es decir imágenes, archivos `.css`, `javascript` y demás.
2. `template_general`: contiene las carpetas de imágenes que se usan a lo largo de todo el proyecto, esto sin contar gráficos de barras o de líneas pues estos son generados a partir de estadísticas.

