Universidad ORT Uruguay Facultad de Ingeniería

# Escuela de Tecnología

OBLIGATORIO PROGRAMACION 1 DOCUMENTO DE ANÁLISIS



**Fernando De Armas – 240970**



**Franco Borrelli – 269686**



**Guillermina Silva - 222048 N1F**

**Docente: Tatiana Rudomin Analista en Tecnologías de la Información**

**Índice**

Contenido

[Escuela de Tecnología 1](#_Toc75377764)

[1. Descripción general del problema a resolver 6](#_Toc75377765)

[1.1. Tipos de usuario del sistema 6](#_Toc75377766)

[1.2. Listado de funcionalidades 7](#_Toc75377767)

[2. Detalle de Funcionalidades 8](#_Toc75377768)

[2.1 F01 – Inicio y login en la aplicación Docente 8](#_Toc75377769)

[2.1.1 Acceso 8](#_Toc75377770)

[2.1.2 Descripción 8](#_Toc75377771)

[2.1.3 Interfaz de usuario 8](#_Toc75377772)

[2.1.4 Validaciones 9](#_Toc75377773)

[2.2 F02 – Registro en la aplicación Docente 10](#_Toc75377774)

[2.2.1 Acceso 10](#_Toc75377775)

[2.2.2 Descripción 10](#_Toc75377776)

[2.2.3 Interfaz de usuario 10](#_Toc75377777)

[2.2.4 Validaciones 10](#_Toc75377778)

[2.3 F03 – Asignación de niveles para alumnos 12](#_Toc75377779)

[2.3.1 Acceso 12](#_Toc75377780)

[2.3.2 Descripción 12](#_Toc75377781)

[2.3.3 Interfaz de usuario 12](#_Toc75377782)

[2.3.4 Validaciones 12](#_Toc75377783)

[2.4 F04 – Planteo de ejercicios según el nivel 13](#_Toc75377784)

[2.4.1 Acceso 13](#_Toc75377785)

[2.4.2 Descripción 13](#_Toc75377786)

[2.4.3 Interfaz de usuario 13](#_Toc75377787)

[2.4.4 Validaciones 14](#_Toc75377788)

[2.5 F05 – Redacción de devoluciones de tareas 15](#_Toc75377789)

[2.5.1 Acceso 15](#_Toc75377790)

[2.5.2 Descripción 15](#_Toc75377791)

[2.5.3 Interfaz de usuario 15](#_Toc75377792)

[2.5.4 Validaciones 15](#_Toc75377793)

[2.6 F06– Visualización de información estadística del docente 16](#_Toc75377794)

[2.6.1 Acceso 16](#_Toc75377795)

[2.6.2 Descripción 16](#_Toc75377796)

[2.6.3 Interfaz de usuario 16](#_Toc75377797)

[2.6.4 Validaciones 16](#_Toc75377798)

[2.7 F07– Inicio y login en la aplicación Alumno 17](#_Toc75377799)

[2.7.1 Acceso 17](#_Toc75377800)

[2.7.2 Descripción 17](#_Toc75377801)

[2.7.3 Interfaz de usuario 17](#_Toc75377802)

[2.7.4 Validaciones 18](#_Toc75377803)

[2.8 F08– Registro en la aplicación Alumno 19](#_Toc75377804)

[2.8.1 Acceso 19](#_Toc75377805)

[2.8.2 Descripción 19](#_Toc75377806)

[2.8.3 Interfaz de usuario 19](#_Toc75377807)

[2.8.4 Validaciones 20](#_Toc75377808)

[2.9 F09 – Visualización de ejercicios planteados 21](#_Toc75377809)

[2.9.1 Acceso 21](#_Toc75377810)

[2.9.2 Descripción 21](#_Toc75377811)

[2.9.3 Interfaz de usuario 21](#_Toc75377812)

[2.9.4 Validaciones 21](#_Toc75377813)

[2.10 F10 – Entrega de tareas 22](#_Toc75377814)

[2.10.1 Acceso 22](#_Toc75377815)

[2.10.2 Descripción 22](#_Toc75377816)

[2.10.3 Interfaz de usuario 22](#_Toc75377817)

[2.10.4 Validaciones 22](#_Toc75377818)

[2.11 F11 – Vista de ejercicios resueltos previamente 23](#_Toc75377819)

[2.11.1 Acceso 23](#_Toc75377820)

[2.11.2 Descripción 23](#_Toc75377821)

[2.11.3 Interfaz de usuario 23](#_Toc75377822)

[2.11.4 Validaciones 23](#_Toc75377823)

[2.12 F12 – Información estadística 24](#_Toc75377824)

[2.12.1 Acceso 24](#_Toc75377825)

[2.12.2 Descripción 24](#_Toc75377826)

[2.12.3 Interfaz de usuario 24](#_Toc75377827)

[2.12.4 Validaciones 24](#_Toc75377828)

[3 Testing 25](#_Toc75377829)

[3.1 Informe de testing (caja negra y valores limites) 25](#_Toc75377830)

[4 Código Completo (HTML y JS) 29](#_Toc75377831)

[5 Datos Precargados 53](#_Toc75377832)

# Descripción general del problema a resolver

La solución informática que se presenta a continuación, tiene como fin brindar un ambiente de trabajo en el cual un grupo de docentes podrán gestionar sus clases de manera virtual para que sus alumnos puedan acceder al material docente.

## Tipos de usuario del sistema

A continuación, se presenta el listado de usuarios del sistema:

* Usuario Docente
* Usuario Alumno

## Listado de funcionalidades

F01 - Inicio y login en la aplicación Docente – Usuario: Docente F02 – Registro en la aplicación – Usuario Docente

F03 - Asignación de niveles para alumnos – Usuario: Docente F04 – Planteo de ejercicios según el nivel – Usuario: Docente F05 – Redacción de devoluciones de tareas – Usuario. Docente

F06 – Visualización de información estadística del docente – Usuario: Docente F07 – Inicio y login en la aplicación Alumno – Usuario: Alumno

F08 – Registro en la aplicación Alumno – Usuario: Alumno

F09 – Visualización de ejercicios planteados – Usuario: Alumno F10 – Entrega de tareas – Usuario: Alumno

F11 - Vista de ejercicios resueltos previamente – Usuario: Alumno

F12– Visualización de información estadística del alumno – Usuario. Alumno

# Detalle de Funcionalidades

A continuación, se presenta el detalle de cada una de las funcionalidades a resolver en el obligatorio. Las funcionalidades están ordenadas por tipo de usuario.

## F01 – Inicio y login en la aplicación Docente

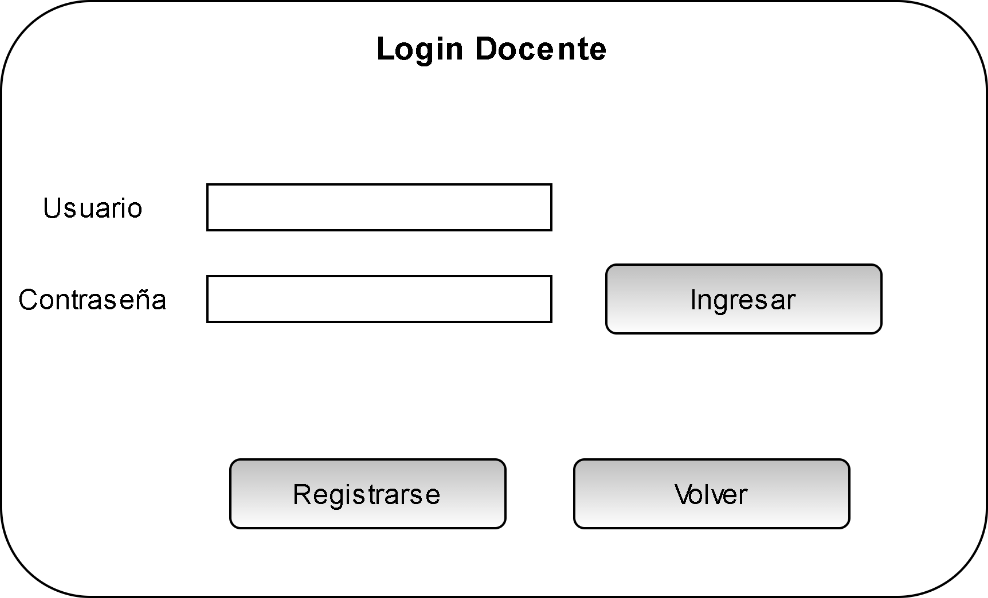
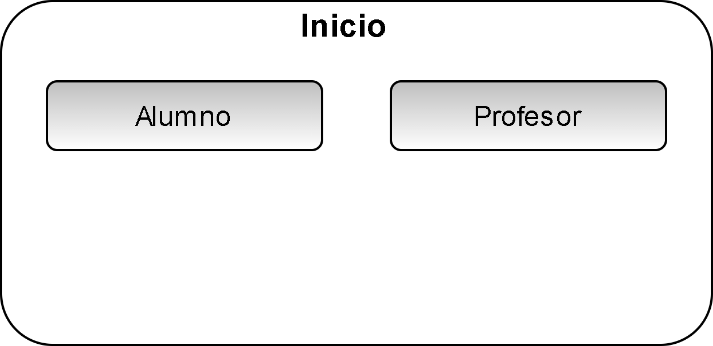
### Acceso

Usuario Docente

### Descripción

Esta funcionalidad consta del inicio de sesión por parte del docente en la aplicación. Se deberá ingresar el nombre del usuario y la contraseña, verificando así la existencia del usuario en el sistema.

### Interfaz de usuario



### Validaciones

TxtUsuario (obligatorio):

1 **-** No debe estar vacío. Mensaje de error: “Los campos están vacíos”

TxtContraseña (obligatorio):

1 **-** No debe estar vacío: Mensaje de error “Los campos están vacíos”

## F02 – Registro en la aplicación Docente

### Acceso

Usuario Docente

### Descripción

Esta funcionalidad consta del registro por parte del docente en la aplicación. Se deberá ingresar un nombre, nombre de usuario y contraseña.

### Interfaz de usuario



### Validaciones

txtNombre (obligatorio):

1 - No debe estar vacío. Mensaje de error: “Los campos están vacíos”

TxtUsuario (obligatorio):

1. - No debe estar vacío: Mensaje de error “Los campos están vacíos”
2. - No puede repetirse en otro usuario. Mensaje de error “El nombre de usuario no se encuentra disponible”

txtContraseña (obligatorio):

1. - No debe estar vacío: Mensaje de error “Los campos están vacíos”
2. – Debe tener un mínimo de 4 caracteres, contando con al menos una minúscula y un número: Mensaje de error “La contraseña no cumple con las condiciones”

btnCrearUsuario:

1 – Verificar que el usuario no exista previamente en el sistema. Mensaje de error: “El usuario ya existe”

## F03 – Asignación de niveles para alumnos

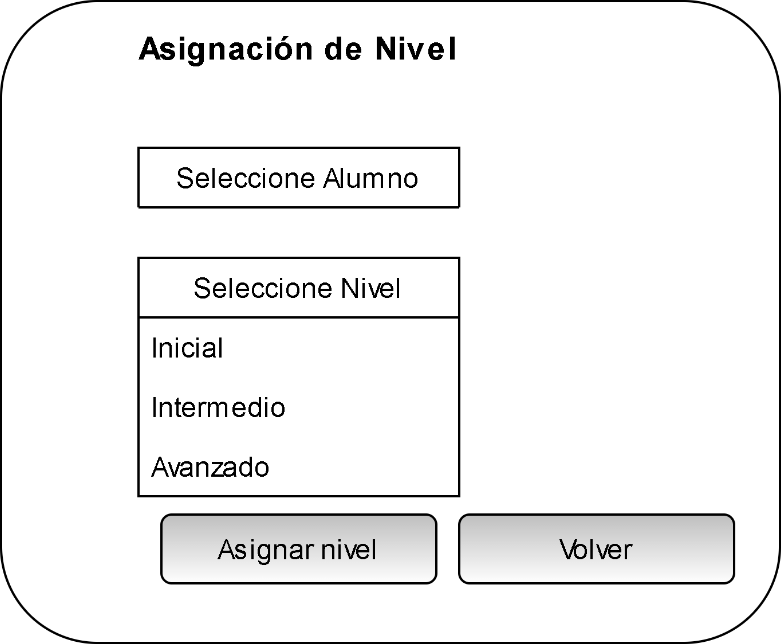
### Acceso

Usuario Docente

### Descripción

Esta funcionalidad consta de la asignación de niveles para los diversos alumnos del docente. Se deberá seleccionar el alumno en cuestión e indicar el nuevo nivel al cual pertenecerá.

### Interfaz de usuario



### Validaciones

btnAsignar:

1 – Deben estar seleccionado el alumno y el nivel. Mensaje de error:” Debe seleccionar el alumno y el nivel”

## F04 – Planteo de ejercicios según el nivel

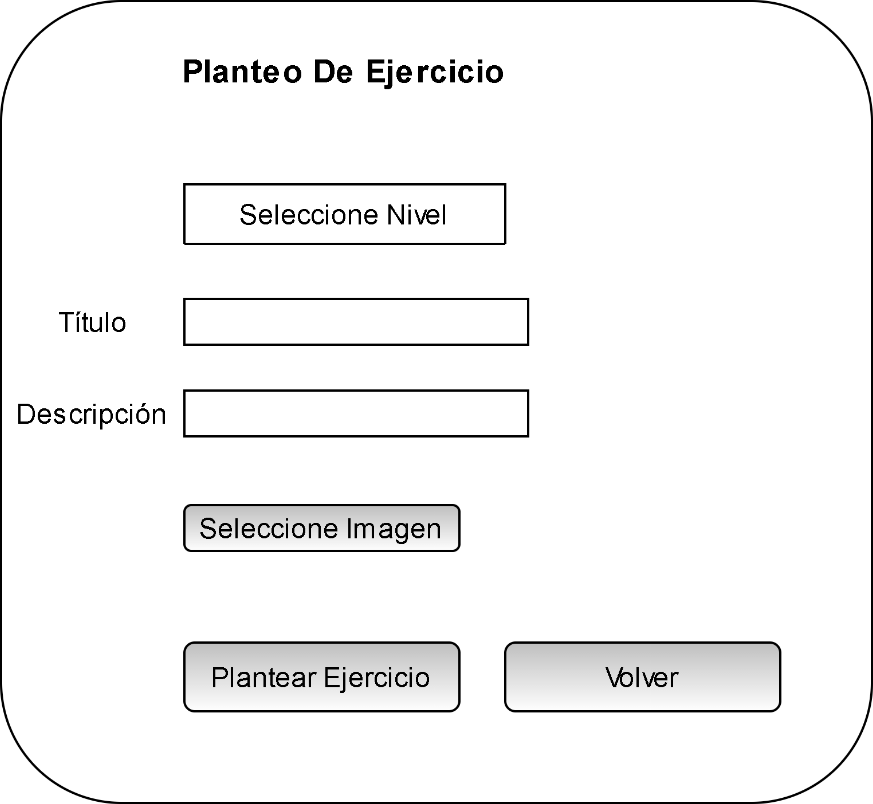
### Acceso

Usuario Docente

### Descripción

En esta funcionalidad el docente deberá plantear ejercicios para sus alumnos según el nivel, constando estos de un titula, imagen y descripción.

### Interfaz de usuario



### Validaciones

txtTitulo (obligatorio):

1 – No puede estar vacío: Mensaje de error: “El campo no puede estar vacío”

txtDescripcion (obligatorio):

1 – No puede estar vacío: Mensaje de error: “El campo no puede estar vacío”

btnPlantear

1 - La suma de caracteres del título y la descripción no pueden superar los 200 caracteres ni ser menor a 20: Mensaje de error:” La cantidad de caracteres debe estar en el rango admitido”

## F05 – Redacción de devoluciones de tareas

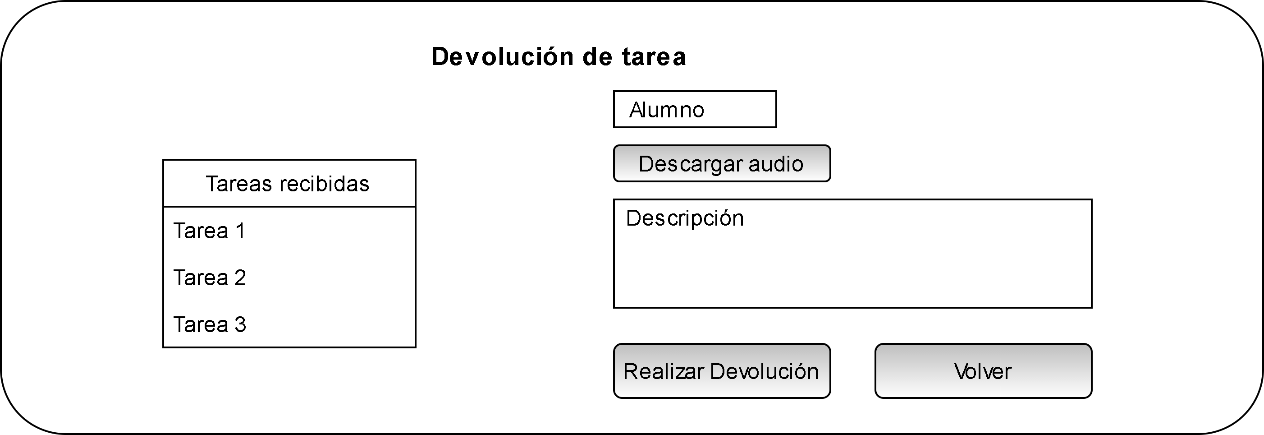
### Acceso

Usuario Docente

### Descripción

En esta funcionalidad el docente redactara devoluciones de las tareas entregadas por los alumnos

### Interfaz de usuario



### Validaciones

txtDescripción(obligatorio):

1- No puede estar vacío. Mensaje de error:” El campo “descripción” no puede estar vacío”

## F06– Visualización de información estadística del docente

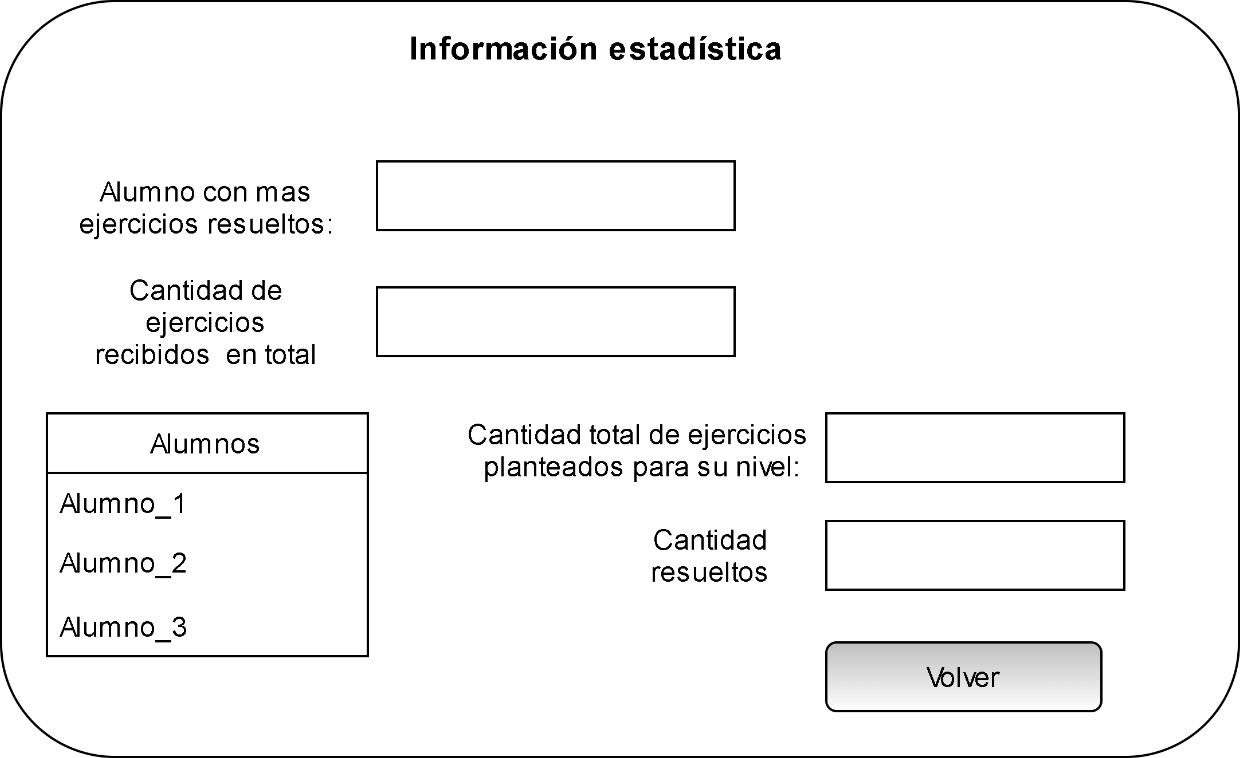
### Acceso

Usuario Docente

### Descripción

En esta funcionalidad el docente podrá visualizar la información estadística.

### Interfaz de usuario



### Validaciones

En este caso dicho formulario no cuenta con campos con validación de datos.

## F07– Inicio y login en la aplicación Alumno

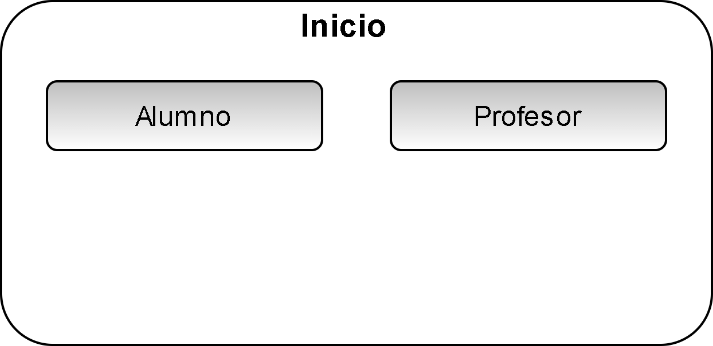
### Acceso

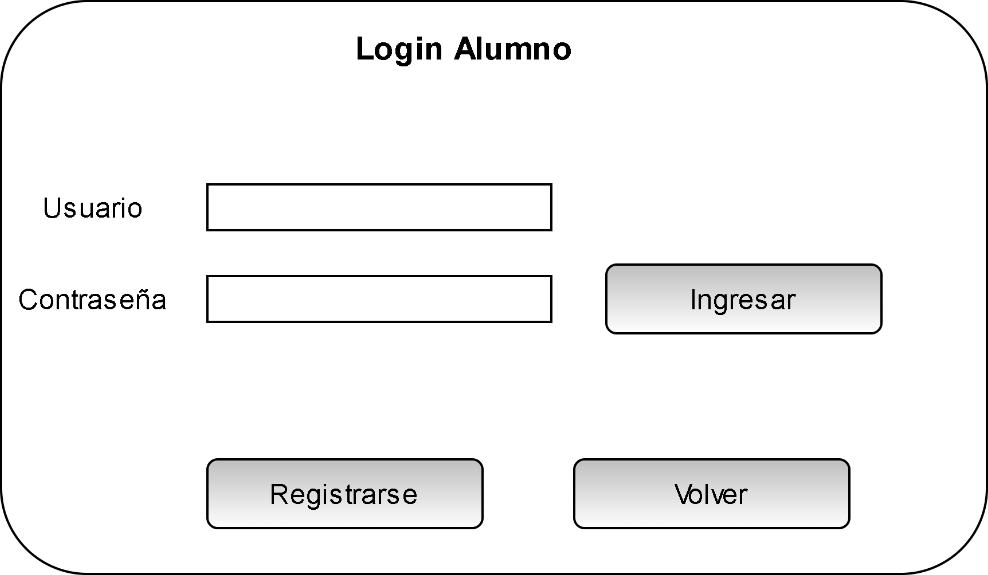
Usuario Alumno

### Descripción

En esta funcionalidad el alumno podrá iniciar sesión en el sistema con sus respectivos datos.

### Interfaz de usuario





### Validaciones

txtUsuario(obligatorio):

1- No puede estar vacío. Mensaje de error: “El campo “usuario” no puede estar vació”

TxtContraseña(obligatorio):

1 – No puede estar vacío. Mensaje de error: “El campo “contraseña” no puede estar vacío”.

## F08– Registro en la aplicación Alumno

### Acceso

Usuario Alumno

### Descripción

En esta funcionalidad el alumno podrá registrarse en el sistema en caso que no tenga un usuario previo.

### Interfaz de usuario



### Validaciones

txtNombre(obligatorio):

1- No puede estar vacío. Mensaje de error: “El campo “usuario” no puede estar vació”

txtUsuario(obligatorio):

No puede estar vacío. Mensaje de error: “El campo “usuario” no puede estar vació”

txtContraseña(obligatorio):

1 – No puede estar vacío. Mensaje de error: “El campo “contraseña” no puede estar vacío”.

2 - Debe tener al menos cuatro caracteres. Mensaje de error: “La contraseña debe tener al menos 4 caracteres”

3 - Debe tener al menos una mayúscula. Mensaje de error: “La contraseña debe tener al menos una mayúscula”

4 - Debe tener al menos una minúscula. Mensaje de error: “La contraseña debe tener al menos una minúscula”

5 - Debe tener al menos un número. Mensaje de error: “La contraseña debe tener al menos un número”

regDocentes(obligatorio):

1 – Debe seleccionar un Docente. Mensaje de error: “No se seleccionó ningún docente”

btnCrearUsuario:

1 – Verificar que el usuario no exista previamente en el sistema. Mensaje de error: “El usuario ya existe”

## F09 – Visualización de ejercicios planteados

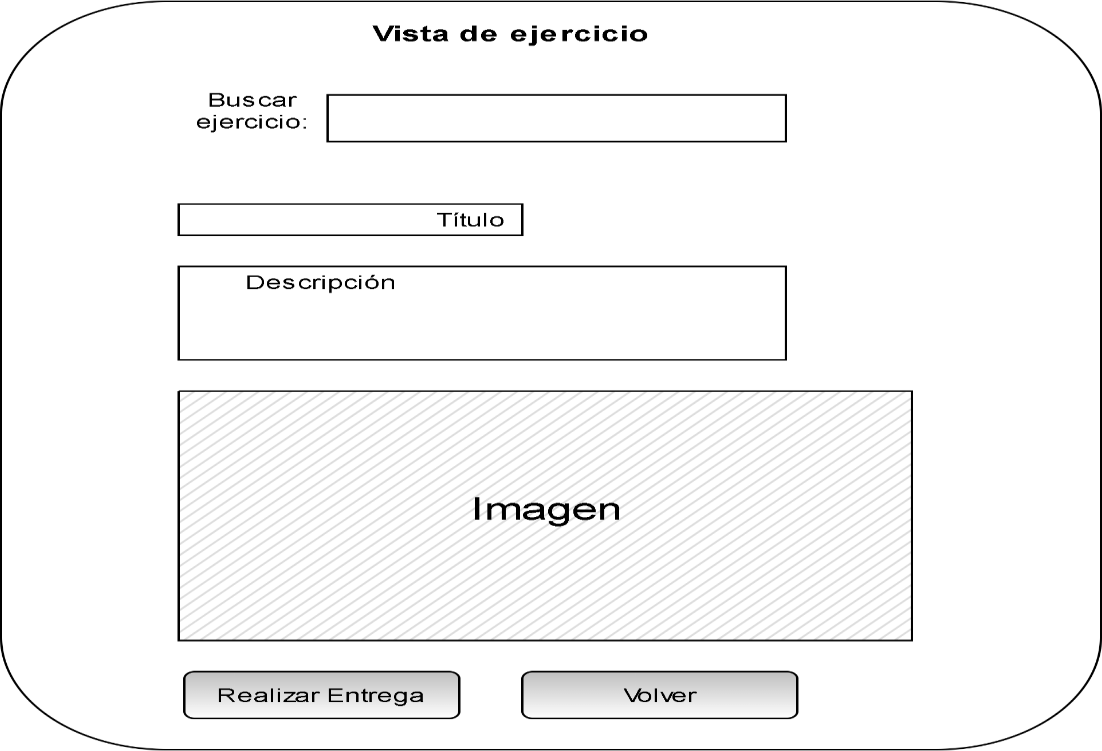
### Acceso

Usuario Alumno

### Descripción

En esta funcionalidad el alumno podrá visualizar los ejercicios planteados por el docente según su nivel.

### Interfaz de usuario



### Validaciones

btnRealizarEntrega

1 – Si ya se realizo la entrega anteriormente, no podrá volver a enviar otra. Mensaje de error: “No puede enviar otra tarea”

## F10 – Entrega de tareas

### 2.10.1 Acceso

Usuario Alumno

### Descripción

En esta funcionalidad el alumno podrá entregar la tarea seleccionada previamente, adjuntando un audio a la misma.

### Interfaz de usuario



### Validaciones

btnRealizarEntrega

1 – No se puede realizar la entrega sin adjuntar ningún audio. Mensaje de error: “Debe adjuntar un audio”

## F11 – Vista de ejercicios resueltos previamente

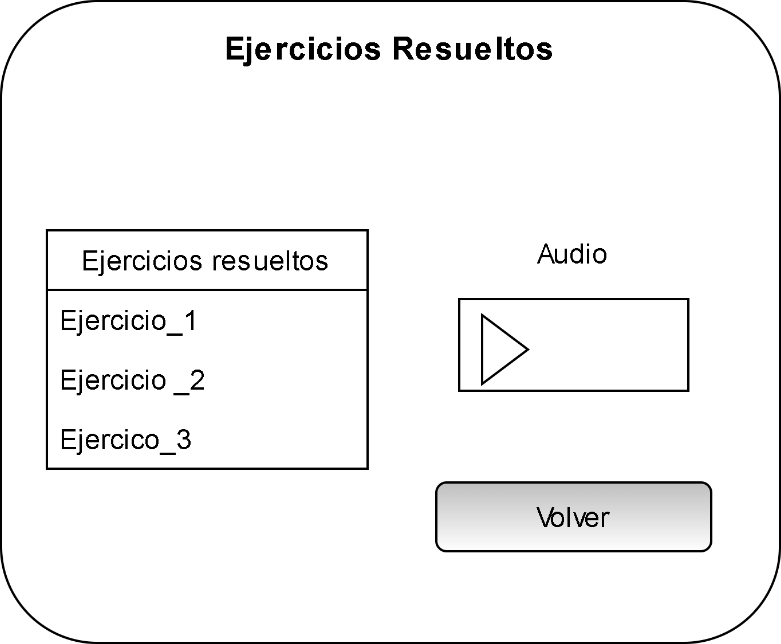
### Acceso

Usuario Alumno

### Descripción

En esta funcionalidad el alumno podrá visualizar los ejercicios entregados previamente y reproducir el audio adjunto al mismo

### Interfaz de usuario



### Validaciones

En esta funcionalidad los datos no se podrán editar, por lo que no existen validaciones.

## F12 – Información estadística

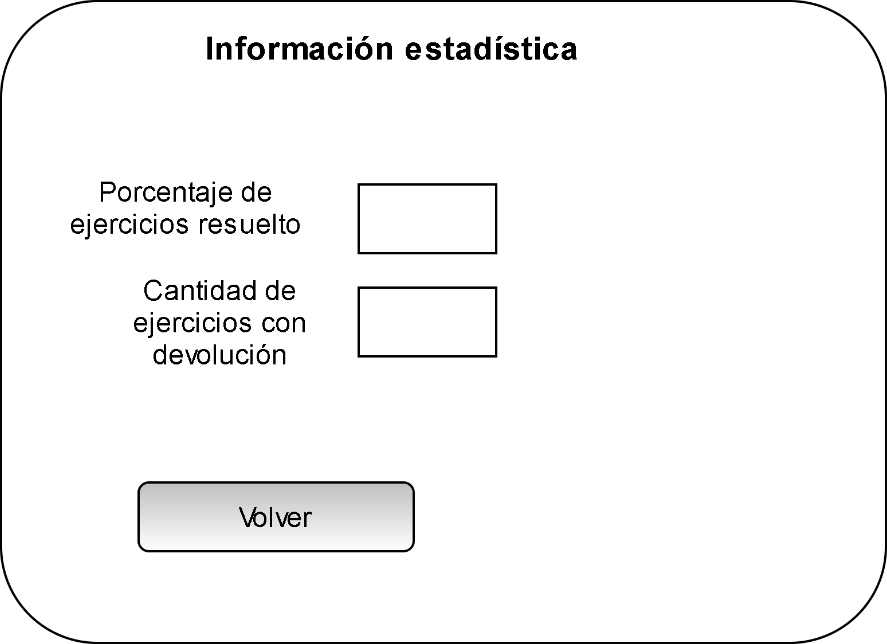
### 2.12.1 Acceso

Usuario Alumno

### Descripción

En esta funcionalidad el alumno podrá visualizar el porcentaje de ejercicios resueltos al igual que la cantidad de ejercicios con devolución del docente.

### Interfaz de usuario



### Validaciones

En esta funcionalidad los datos no se podrán editar, por lo que no existen validaciones.

# Testing

## 3.1 Informe de testing (caja negra y valores limites)

A continuación, se presenta la prueba de testing de caja negra y valores límites del obligatorio en cuestión. El mismo muestra las diferentes reacciones esperadas y finales del sistema ante diversos datos ingresados en funciones del mismo, indicando siempre los datos a pedir y formularios.

Tipos de Verificación:

-Aceptable: “El sistema responde acorde a lo esperado”

-Inaceptable: “El sistema responde de manera diferente a lo esperado, pero nunca finalizando la ejecución del programa (un mensaje diferente a lo que se debería, etc)”

-Error: Un fallo en el sistema que finalice la ejecución del mismo

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Acción | Ventana de Registro | Datos | Datos de entrada | Datos esperados | Datos de salida | Verificaciónón |
| El usuario desea registrarse en el sistema | divRegistro | -Nombre -Usuario  contraseña | Pepe;pepe12;1234aB | "Usuario Ingresado con éxito" | "Usuario Ingresado con éxito" | Aceptable |
|  | divRegistro | -Nombre -Usuario  contraseña | Pepe; “”; 1234aB | “No puede haber datos vacíos” | “No puede haber datos vacíos” | Aceptable |
|  | divRegistro | -Nombre -Usuario  contraseña | “”;Pepe12;1234aB | “No puede haber datos vacíos” | “No puede haber datos vacíos” | Aceptable |
|  | divRegistro | -Nombre -Usuario  contraseña | Pepe;pepe12;”” | “La contraseña debe tener mínimo 4 caracteres”  “La contraseña debe tener al menos una minúscula”  “La contraseña debe tener al menos una mayúscula”  “La contraseña debe tener al menos un número” | “La contraseña debe tener mínimo 4 caracteres”  “La contraseña debe tener al menos una minúscula”  “La contraseña debe tener al menos una mayúscula”  “La contraseña debe tener al menos un número” | Aceptable |
|  | divRegistro | -Nombre -Usuario  contraseña | Pepe;pepe12;1234 | “La contraseña debe tener al menos una minúscula”  “La contraseña debe tener al menos una mayúscula” | “La contraseña debe tener al menos una minúscula”  “La contraseña debe tener al menos una mayúscula” | Aceptable |
|  | divRegistro | -Nombre -Usuario  contraseña | Pepe;pepe12;1234a | “La contraseña debe tener al menos una mayúscula” | “La contraseña debe tener al menos una mayúscula” | Aceptable |
|  | divRegistro | -Nombre -Usuario  contraseña | Pepe;pepe12;1234B | “La contraseña debe tener al menos una minúscula” | “La contraseña debe tener al menos una minúscula” | Aceptable |
| El usuario desea ingresar al sistema. | divIngreso | -Usuario  -Contraseña | Pepe12;”” | “Usuario no registrado” | “Usuario no registrado” | Aceptable |
|  | divIngreso | -Usuario  -Contraseña | “”;1234aB | “Usuario no registrado” | “Usuario no registrado” | Aceptable |
|  | divIngreso | -Usuario  -Contraseña | “”,”” | “Usuario no registrado” | “Usuario no registrado” | Aceptable |
|  | divIngreso | -Usuario  -Contraseña | JuanPepe,”123a” | “Usuario no registrado” | “Usuario no registrado” | Aceptable |
|  | divIngreso | -Usuario  -Contraseña | Pepe12,1234aB | (Ingresa al sistema) | (Ingresa al sistema) | Aceptable |
| El usuario alumno desea realizar una entrega de ejercicios | divEntregaTarea | -btnEntregarAudio | (No se selecciona ningún audio) | “Debe seleccionar un audio” | “Debe seleccionar un audio” | Aceptable |
|  | divEntregaTarea | -btnEntregaAudio | (Audio seleccionado) | “Entrega Agregada” | “Entrega Agregada” | Aceptable |
| El usuario alumno desea visualizar su información estadística | divInformacionEstadisticaAlumno | btnInformacionEstadistica |  | “Se muestra la información estadística del alumno” | “Se muestra la información estadística del alumno” | Aceptable |
| El usuario docente desea asignar un nuevo nivel a sus alumnos | divAsignacionNivel | Select = regAlumnos | (Se selecciona al alumno de la lista) | (“Muestra los niveles disponibles”) | (“Muestra los niveles disponibles”) | Aceptable |
| El usuario elije un nuevo nivel para su alumno seleccionado | divMostrarNiveles | btnAsginarNivel | (No se selecciona ningún nivel) | (El sistema por defecto selecciona el primer nivel) | (El sistema por defecto selecciona el primer nivel) | Aceptable |
|  | divMostrarNiveles | btnAsginarNivel  Nivel actual del alumno: Básico | Avanzado | “Solo se puede subir un nivel” | “Solo se puede subir un nivel” | Aceptable |
|  | divMostrarNiveles | btnAsginarNivel  Nivel actual del alumno: Básico | Intermedio | “Se ha cambiado el nivel del alumno” | “Se ha cambiado el nivel del alumno” | Aceptable |
|  | divMostrarNiveles | (Se intenta cambiar el nivel del alumno el cual es avanzado) |  | “El nivel del alumno es avanzado, no se permite reducir el nivel del mismo” | “El nivel del alumno es avanzado, no se permite reducir el nivel del mismo” | Aceptable |
| El usuario docente desea plantear un ejercicio | divPlanteoEjercicio | -Nivel  -Titulo  -Descripción  -Imagen | Inicial;  Ejercicio 3;  En este ejercicio deberán hacer un solo de 10 minutos a un ritmo de 80 en el metrónomo;  IMG2.jpg | “Tarea agregada” | “Tarea agregada” | Aceptable |
|  | divPlanteoEjercicio | -Nivel  -Titulo  -Descripción  -Imagen | Inicial;  “”:  En este ejercicio deberán hacer un solo de 10 minutos a un ritmo de 80 en el metrónomo;  IMG2.jpg | “No pueden existir campos vacíos” | “No pueden existir campos vacíos” | Aceptable |
|  | divPlanteoEjercicio | -Nivel  -Titulo  -Descripción  -Imagen | Inicial;  “Ejercicio 12”;  “”;  IMG2.jpg | “No pueden existir campos vacíos” | “No pueden existir campos vacíos” | Aceptable |
|  | divPlanteoEjercicio | -Nivel  -Titulo  -Descripción  -Imagen | Inicial;  “Ejercicio 3”;  “En este ejercicio deberán hacer un solo de 10 minutos a un ritmo de 80 en el metrónomo;  “”; | “No pueden existir campos vacíos” | “No pueden existir campos vacíos” | Aceptable |
|  | divPlanteoEjercicio | -Nivel  -Titulo  -Descripción  -Imagen | (Se ingresan todos los campos, pero entre el título y la descripción no se llega a los 20 caracteres) | “El total de caracteres entre el título y la descripción debe estar entre 20 y 200” | “El total de caracteres entre el título y la descripción debe estar entre 20 y 200” | Aceptable |
|  | divPlanteoEjercicio | -Nivel  -Titulo  -Descripción  -Imagen | (Se ingresan todos los campos, pero entre el título y la descripción se superan los 200 caracteres) | “El total de caracteres entre el título y la descripción debe estar entre 20 y 200” | “El total de caracteres entre el título y la descripción debe estar entre 20 y 200” | Aceptable |
| El docente desea visualizar información estadística | divInformaciónEstadistica | btnMostrarTotalEjercicios | (El docente no selecciona el alumno) | (Se muestra la información estadística del mismo) | (Se muestra la información estadística del mismo) | Aceptable |
| El usuario docente desea realizar una devolución | divCorrecionEjerciciosEntregados | Devolución | “” | “Debe ingresar una devolución” | “Debe ingresar una devolución” | Aceptable |
|  | divCorrecionEjerciciosEntregados | Devolución | Excelente tarea, cumplio con todos los requisitos. | “Se asigno una devolución a la tarea” | “Se asigno una devolución a la tarea” | Aceptable |

# Código Completo (HTML y JS)

HTML:

index.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" type="text/css" href="css.css">

  <script src="js/clases.js"></script>

  <script src="js/obligatorio.js"></script>

  <title>Document</title>

</head>

<body>

  <div id="contenedor">

    <div id="cabezal" class="clearfix">

      <img id="logo" src="https://develotion.com/assets/img/logo.jpg" alt="logo">

      <ul id="navPrincipal">

      </ul>

      <div id="aux">

      </div><!-- FIN AUX -->

    </div><!-- FIN CABEZAL -->

    <div id="contenido">

      <div id="divIngreso">

        <h5>Ingreso</h5>

        <form action="#">

          <label for="loginUsuario">Usuario</label>

          <input type="text" id="loginUsuario" name="loginUsuario"><br>

          <label for="loginPass">Contraseña</label>

          <input type="password" id="loginPass" name="loginPass"><br>

          <input type="button" value="Ingresar" id="btnIngresar">

          <input type="button" value="Registrar" id="btnVentanaRegistrar">

        </form>

      </div>

      <div id="divRegistro" style="display: none;">

        <h5>Registro</h5>

        <div id="divSelectTipoUsuario">

          <label for="regTipoUsuario">Tipo de usuario</label>

          <select id="regTipoUsuario">

            <option value="A">Alumno</option>

            <option value="D">Docente</option>

          </select>

          <input type="button" value="Continuar" id="btnContinuar">

        </div>

        <div id="divDatosDeUsuario">

          <label for="regNombre">Nombre</label>

          <input type="text" id="regNombre" name="regNombre"><br>

          <label for="regUsuario">Usuario</label>

          <input type="text" id="regUsuario" name="regUsuario"><br>

          <label for="regPass">Contraseña</label>

          <input type="password" id="regPass" name="regPass"><br>

          <label for="regDocente" style="display: none;" id="labelRegDocente">Docente</label>

          <select name="regDocente" id="regDocente" style="display: none;">

          </select><br>

          <p id="errorRegistro" style="display: none;"></p>

          <input type="button" value="Registrar" id="btnRegistrar">

        </div>

        <input type="button" value="Salir" onclick="ocultar('#divRegistro') + mostrar('#divIngreso') ">

      </div>

      <hr>

      <div id="divFiltro">

        <label for="buscador">Ingrese el titulo o descripción del ejercicio a buscar: </label>

        <input type="text" id="buscador" name="buscador">

      </div>

      <div class="ejercicios" id="divEjercicios">

      </div>

      <!--FIN EJERCICIOS-->

      <div id="divEntregaTarea" style="display: none;">

        <h1 id="tituloEjercicio"></h1>

        <p id="descripcionEjercicio"></p>

        <div id="imagenEjercicio"></div>

        <label for="audio">Ingrese un audio</label>

        <input accept="audio/\*" type="file" name="audio" id="audio">

        <br>

        <input type="button" id="btnEntregarAudio" value="Entregar">

      </div>

      <!-- VER EJERCICIOS RESUELTOS -->

      <div id="divEjerciciosResueltos" style="display: none;">

      </div>

      <!-- Visualizar información estadística-->

      <div id="divInformacionEstadisticaAlumno">

        <p id="porcentajeEjResueltos"></p>

        <p id="ejerciciosConSinDevolucion"></p>

      </div>

      <div id="divInformacionEstadisticaDocente">

        <p id="alumnosConMayorEjercicios"></p>

        <br>

        <p id="cantidadDeEjercicios"></p>

        <br>

        <div id="divMostrarAlumnos" style="display: none;">

          <label for="regAlumnos">Seleccione el alumno</label>

          <select name="regAlumnosInformacion" id="regAlumnosInformacion" style="display: block;">

          </select><br>

          <input type="button" name="btnMostrarTotalEjercicios" id="btnMostrarTotalEjercicios"

            value="Mostrar Total Ejercicios" onclick="MostrarTotalEjercicios()">

          <p id="totalEjerciciosDelAlumno"></p>

        </div>

      </div>

      <!-- DEVOLUCIÓN DE TAREAS - DOCENTE -->

      <div id="divEjerciciosEntregados" style="display: none;">

      </div>

      <div id="divCorrecionEjerciciosEntregados" style="display: none;">

      </div>

      <div id="divPlanteoEjercicio" style="display: none;">

        <h5>Planteo de ejercicio</h5>

        <label for="planteoNivel">Para nivel</label>

        <select name="planteoNivel" id="planteoNivel">

          <option value="1">Inicial</option>

          <option value="2">Intermedio</option>

          <option value="3">Avanzado</option>

        </select><br>

        <label for="planteoTitulo">Título</label>

        <input type="text" id="planteoTitulo" name="planteoTitulo"><br>

        <label for="planteoDescripcion">Descripción</label>

        <textarea name="planteoDescripcion" id="planteoDescripcion" cols="30" rows="10"></textarea><br>

        <label for="planteoImagen">Cargar Imagen</label>

        <input accept="image/\*" type="file" name="planteoImagen" id="planteoImagen"><br>

        <input type="button" value="Plantear ejercicio" id="btnPlantearEjercicio">

      </div>

      <div id="divAsignacionNivel" style="display: none;">

        <h5>Asignacion de nivel</h5>

        <label for="regAlumnos">Seleccione el alumno</label>

        <select onchange="mostrarNiveles()" name="regAlumnos" id="regAlumnos" style="display: block;">

        </select>

        <br>

        <br>

      </div>

      <div id="divMostrarNiveles" style="display: none;">

        <label for="regNiveles">Seleccione el nivel</label>

        <select name="regNiveles" id="regNiveles" style="display: block;"></select><br>

        <input type="button" name="btnAsignarNivel" id="btnAsignarNivel" value="Asignar Nivel">

      </div>

    </div>

    <div id="pie">

      <a href="https://www.ort.edu.uy">Footer</a>

    </div><!-- FIN PIE -->

  </div><!-- FIN CONTENEDOR -->

</body>

</html>

JS

clases.js:

//Constructor Usuarios

class usuario{

    constructor(user, nombre,  pass, tipo, nivel, docente){

    this.id = user.toUpperCase();

    this.nombre = nombre

    this.pass = pass;

    this.tipo = tipo; //D:Docente - A:Alumno

    this.nivel = nivel;

    this.docente = docente.toUpperCase();

    }

}

class ejercicio{

    constructor(titulo, descripcion, imagen, docente, nivel){

        idEjercicio++;

        this.id = idEjercicio;

        this.titulo = titulo;

        this.descripcion = descripcion;

        this.imagen = imagen;

        this.docente = docente.toUpperCase();

        this.nivel = nivel;

    }

}

class EjerciciosEntregados{

    constructor(ejercicio, audio,usuario,devolucion,corregido){

        idEntrega++;

        this.id = idEntrega;

        this.ejercicio=ejercicio;

        this.audio=audio;

        this.usuario=usuario;

        this.devolucion=devolucion;

        this.corregido=corregido;

    }

}

obligatorio.js:

let usuarios = []; //Este es un array de usuarios, son los registrados en la aplicación.

let usuarioIngreso = ""; //Este es el usuario que ingresa

let ejercicios = []; //Estos son los ejercicios planteados.

let idEjercicio = 0; //Este es un id que es uníco para cada ejercicio que se plantea.

let idEntrega = 0;   //Este es el id unico de cada entrega

let ejercicioEntregado = ""; //este es el ejercicio que se está entregando

let ejerciciosEntregados = []; //estos son todos los ejercicios entregados

window.addEventListener("load", inicio);

function inicio() {

  datosPreCargados();

  ocultarTodo();

  document.querySelector("#btnRegistrar").addEventListener("click", registrarUsuario);

  document.querySelector("#btnContinuar").addEventListener("click", continuarRegistro);

  document.querySelector("#btnIngresar").addEventListener("click", ingresoDeUsuario);

  document.querySelector("#btnVentanaRegistrar").addEventListener("click", verVentanaRegistrar);

  document.querySelector("#btnAsignarNivel").addEventListener("click", asignarNivel);

  document.querySelector("#btnPlantearEjercicio").addEventListener("click", subirEjercicio);

  document.querySelector("#btnEntregarAudio").addEventListener("click", entregarEjercicio);

  document.querySelector("#buscador").addEventListener("keyup", cargarEjercicios);

}

/\*\*Esta función permite escoger ventanas a ocultar de acuerdo a lo que necesitamos mostrar\*/

function ocultarTodo() {

  ocultar("#divRegistro");

  ocultar("#divEjercicios");

  ocultar("#divPlanteoEjercicio");

  ocultar("#divAsignacionNivel");

  ocultar("#divEntregaTarea");

  ocultar("#divFiltro");

  ocultar("#divInformacionEstadisticaAlumno");

  ocultar("#divEjerciciosEntregados");

  ocultar("#divInformacionEstadisticaDocente")

  ocultar("#divEjerciciosResueltos");

  ocultar("#divCorrecionEjerciciosEntregados");

  ocultar("#divMostrarNiveles");

  ocultar("#divInformacionEstadisticaDocente");

  mostrar("#navPrincipal");

}

/\*\*Vaciar campos ingresados para el login\*/

function vaciarCampos() {

  document.querySelector("#loginUsuario").value = "";

  document.querySelector("#loginPass").value = "";

}

/\*\*Oculta unicamente los id que pasemos por parametro\*/

function ocultar(id) {

  document.querySelector(id).style.display = "none";

}

/\*\*Muestra unicamente los id que pasemos por parametro\*/

function mostrar(id) {

  document.querySelector(id).style.display = "block";

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//--------------------------------------------------REGISTRO DE USUARIOS--------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function continuarRegistro() {

  ocultar("#divSelectTipoUsuario");

  mostrar("#divDatosDeUsuario")

  ocultar("#errorRegistro");

  //vaciamos los campos por si se vuelve a ingresar

  vaciarCamposRegistro()

  let tipoDeUsuario = document.querySelector("#regTipoUsuario").value;

  if (tipoDeUsuario === "A") {

    hacerVisibleDocentes();

  } else {

    ocultar("#labelRegDocente");

    ocultar("#regDocente");

  }

}

/\*\*Si el usuario registrado es un alumno se muestra la lista de docentes registrados para seleccionar\*/

function hacerVisibleDocentes() {

  let docentes = "";

  mostrar("#labelRegDocente");

  mostrar("#regDocente");

  /\*\*Agrego docentes a la lista de selección\*/

  for (let elemento of usuarios) {

    if (elemento.tipo === "D") {

      docentes += '<option value="' + elemento.id + '">' + elemento.nombre + "(" + elemento.id + ")</option>";

    }

  }

  document.querySelector("#regDocente").innerHTML = docentes;

}

/\*\*Permite el registro de usuarios docente y alumno verificando que se cumplan las condiciones establecidas.

No pueden existir dos usuarios iguales La contraseña debe contener al menos 4 caracteres, 1 minúscula, 1 mayúscula y 1 número

Si el tipo de usuario es alumno deberá asignarse un docente\*/

function registrarUsuario() {

  let usuarioReg = document.querySelector("#regUsuario").value.toUpperCase();

  let contraseñaReg = document.querySelector("#regPass").value;

  let nombreReg = document.querySelector("#regNombre").value;

  let tipo = "D";

  let nivel = "";

  let docente = "";

  let error = "";

  if (comprobarSiUsuarioExiste(usuarioReg)) {

    error = "El usuario ya existe<br>";

  }

  if (usuarioReg.length === 0) {

    error += "Debe ingresar un usuario<br>";

  }

  error += comprobarPass(contraseñaReg);

  /\*\*Si seleccionó que es alumno debe tener un docente asignado\*/

  docente = document.querySelector("#regDocente").value;

  tipo = document.querySelector("#regTipoUsuario").value;

  if (tipo == "A" && docente == "") {

    error += "Debe seleccionar un docente<br>"

  }

  if (error != "") {

    document.querySelector("#errorRegistro").innerHTML = error;

    mostrar("#errorRegistro");

  } else {

    document.querySelector("#errorRegistro").innerHTML = "";

    ocultar("#errorRegistro")

    if (tipo == "A") { //Datos que son solo del alumno

      nivel = "1";

    } else {

      docente = "";

      nivel = "";

    }

    crearUsuario(usuarioReg, nombreReg, contraseñaReg, tipo, nivel, docente);

    mostrar("#divIngreso");

    ocultar("#divRegistro");

    vaciarCamposRegistro()

  }

}

function vaciarCamposRegistro() {

  usuarioReg = document.querySelector("#regUsuario").value = "";

  contraseñaReg = document.querySelector("#regPass").value = "";

  nombreReg = document.querySelector("#regNombre").value = "";

}

/\*\*Comprueba que el usuario que se esté creando no haya sido creado anteriormente\*/

function comprobarSiUsuarioExiste(user) {

  /\*Acá hay que recorre el vector de usuarios  para ver si ya existe\*/

  let yaExiste = false;

  for (let i = 0; i < usuarios.length && !yaExiste; i++) {

    if (usuarios[i].id === user) {

      yaExiste = true;

    }

  }

  return yaExiste;

}

/\*\*Comprueba que la contraseña contenga un mínimo de 4 caracteres, 1 mayúscula, 1 minúscula y 1 número\*/

function comprobarPass(pass) {

  //Se comprueba que la contraseña cumpla con los parámetros:

  let resultado = ""; //Correcta

  let codigo = 0;

  let tieneMin = "N";

  let tieneMay = "N";

  let tieneNum = "N";

  let tiene4Letras = "";

  if (pass.length < 4) {

    tiene4Letras = "N"; //No cumple largo de 4

  }

  for (let i = 0; i < pass.length; i++) {

    codigo = pass.charCodeAt(i);

    if (codigo >= 97 && codigo <= 118) {

      tieneMin = "S";

    }

    if (codigo >= 65 && codigo <= 90) {

      tieneMay = "S";

    }

    if (codigo >= 48 && codigo <= 57) {

      tieneNum = "S";

    }

  }

  if (tiene4Letras === "N") {

    resultado = "La contraseña debe tener mínimo 4 caracteres<br>";

  }

  if (tieneMin === "N") {

    resultado += "La contraseña debe tener al menos una minúscula<br>";

  }

  if (tieneMay === "N") {

    resultado += "La contraseña debe tener al menos una mayúscula<br>";

  }

  if (tieneNum === "N") {

    resultado += "La contraseña debe tener al menos un número<br>";

  }

  return resultado;

}

function verVentanaRegistrar() {

  document.querySelector("#loginUsuario").value = "";

  document.querySelector("#loginPass").value = "";

  ocultar("#divIngreso");

  ocultar("#divDatosDeUsuario");

  mostrar("#divRegistro");

  mostrar("#divSelectTipoUsuario")

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//--------------------------------------------------INGRESO DE USUARIOS--------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

/\*\*Comprueba que el usuario a logearse exista y permite el ingreso al menú que corresponda según tipo de usuario

Oculta ventana de ingreso \*/

function ingresoDeUsuario() {

  let loginUsuario = document.querySelector("#loginUsuario").value.toUpperCase();

  let loginPass = document.querySelector("#loginPass").value;

  let usuarioRegistrado = false;

  let tipo = "";

  for (let elemento of usuarios) {

    if (elemento.id === loginUsuario && elemento.pass === loginPass) {

      usuarioRegistrado = true;

      tipo = elemento.tipo;

      usuarioIngreso = new usuario(

        elemento.id,

        elemento.nombre,

        elemento.pass,

        elemento.tipo,

        elemento.nivel,

        elemento.docente

      );

      ocultar("#divIngreso");

      cargarMenu(tipo);

      mostrar("#contenedor");

    }

  }

  if (!usuarioRegistrado) {

    alert("Usuario no registrado");

  }

}

//Retorna a ventana ingreso de usuario

function salir() {

  ocultarTodo();

  document.querySelector("#navPrincipal").innerHTML = "";

  mostrar("#divIngreso");

  vaciarCampos();

}

//Carga menú según tipo de usuario

function cargarMenu(tipoUsuario) {

  let menuAmostrar = "";

  ocultarTodo();

  if (tipoUsuario === "A") {

    menuAmostrar = '<li onclick="cargarEjercicios()" > <a>' + "Ver ejercicios planteados y entregar" + "</a> </li>";

    menuAmostrar += '<li onclick="verVentanaEjerciciosResueltos()"> <a>' + "Ver ejercicios resueltos" + "</a> </li>";

    menuAmostrar += '<li onclick="infromacionEstedisticaAlumno()"> <a>' + "Informacion estadistica" + "</a> </li>";

    cargarEjercicios();//Carga ejercicios pendientes de realizar para el alumno en pantalla principal

  } else {

    menuAmostrar = '<li onclick="verVentanaAsignarNivel()"> <a id="btnAsignarNivel">' + "Asignar nivel alumno" + "</a> </li>";

    menuAmostrar += '<li onclick="verVentanaPlanteoEjercicio()"> <a>' + "Plantear ejercicios" + "</a> </li>";

    menuAmostrar += '<li onclick="cargarEjerciciosEntregados()"> <a>' + "Realizar devoluciones" + "</a> </li>";

    menuAmostrar += '<li onclick="informacionEstadisticaDocente()"> <a>' + "Informacion estadistica" + "</a> </li>";

    cargarEjerciciosEntregados();//Carga ejercicios entregados para el docente en pantalla principal

  }

  menuAmostrar += '<li onclick="salir()" > <a>' + "Salir." + "</a> </li>";

  document.querySelector("#navPrincipal").innerHTML = menuAmostrar;

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//--------------------------------------------------MENU DOCENTE /  Asignacion Nivel -------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function verVentanaAsignarNivel() {

  ocultarTodo();

  mostrar("#divAsignacionNivel");

  let alumnos = '<option value="0">Seleccione</option>';

  for (let elemento of usuarios) {

    if (elemento.tipo === "A" && elemento.docente === usuarioIngreso.id) {

      alumnos +=

        '<option value="' +

        elemento.id +

        '">' +

        elemento.nombre +

        "(" +

        elemento.id +

        ")</option>";

    }

  }

  document.querySelector("#regAlumnos").innerHTML = alumnos;

}

function mostrarNiveles() {

  if (document.querySelector("#regAlumnos").value == "0") {

    ocultar("#divMostrarNiveles");

  } else {

    mostrar("#divMostrarNiveles");

    let id\_alumno\_seleccionado = document.getElementById("regAlumnos").value;

    let niveles\_a\_mostrar = "";

    for (let elemento of usuarios) {

      if (id\_alumno\_seleccionado === elemento.id) {

        let nivel\_alumno = elemento.nivel;

        switch (nivel\_alumno) {

          case "1":

            niveles\_a\_mostrar += "<option value=2> Intermedio </option>";

            niveles\_a\_mostrar += "<option value=3> Avanzado </option>";

            break;

          case "2":

            niveles\_a\_mostrar = "<option value=3> Avanzado </option>";

            break;

          case "3":

            ocultar("#divMostrarNiveles");

            alert("El nivel del alumno es Avanzado, no se permite reducir el nivel del mismo");

            break;

          default:

            break;

        }

        document.querySelector("#regNiveles").innerHTML = niveles\_a\_mostrar;

      }

    }

  }

}

function asignarNivel() {

  let id\_alumno\_seleccionado = document.getElementById("regAlumnos").value;

  let nivel\_nuevo = document.getElementById("regNiveles").value;

  for (let elemento of usuarios) {

    if (id\_alumno\_seleccionado === elemento.id) {

      if (elemento.nivel === "1" && nivel\_nuevo === "3") {

        alert("Solo se puede subir un nivel");

      } else {

        elemento.nivel = nivel\_nuevo;

        alert("Se ha cambiado el nivel del alumno: " + id\_alumno\_seleccionado);

        ocultarTodo();

        cargarEjerciciosEntregados();

      }

    }

  }

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//----------------------------------------------VISTA- DE EJERCICIOS Planteados -Alumno-----------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function cargarEjercicios() {

  document.querySelector("#divEjercicios").innerHTML = "";

  ocultarTodo();

  mostrar("#divFiltro");

  mostrar("#divEjercicios");

  document.querySelector("#divEjercicios").innerHTML = '<p id="noHayEjercicios"></p>';

  let nombre\_a\_buscar = document.getElementById("buscador").value.toUpperCase();

  let listaAMostrar = buscarEjercicio(nombre\_a\_buscar);

  if (listaAMostrar.length == 0) {

    let mensaje = "No hay resultados que coincidan con su busqueda";

    if (nombre\_a\_buscar == "") {

      mensaje = "No hay ejercicios planteados."

    }

    document.querySelector("#noHayEjercicios").innerHTML = mensaje;

  } else {

    document.querySelector("#noHayEjercicios").innerHTML = "";

    for (let elemento of listaAMostrar) {

      if (elemento.docente === usuarioIngreso.docente && elemento.nivel === usuarioIngreso.nivel) {

        agregarEjercicioAPantalla(

          elemento.id,

          elemento.titulo,

          elemento.imagen,

          elemento.descripcion

        );

      }

    }

  }

}

function buscarEjercicio(nombre\_a\_buscar) {

  //Devuelve los ejercicios que  cumplen con el filtro.

  let listaAMostrar = [];

  encontro = false;

  for (let elemento of ejercicios) {

    //alert("elemento.docente " +elemento.docente + "\_" + usuarioIngreso.docente + " niv= " + elemento.nivel +" \_ = " + usuarioIngreso.nivel);

    if (elemento.docente === usuarioIngreso.docente && elemento.nivel === usuarioIngreso.nivel && elemento.titulo.toUpperCase().includes(nombre\_a\_buscar)) {

      encontro = true;

      listaAMostrar.push(elemento);

    }

  }

  if (encontro == false) {

    for (let elemento of ejercicios) {

      if (

        elemento.docente === usuarioIngreso.docente &&

        elemento.nivel === usuarioIngreso.nivel &&

        elemento.descripcion.toUpperCase().includes(nombre\_a\_buscar)

      ) {

        listaAMostrar.push(elemento);

      }

    }

  }

  return listaAMostrar;

}

function mostrarSubirEntrega(id) {

  ocultarTodo();

  mostrar("#divEntregaTarea");

  let encontre = false;

  let titulo = "";

  let docente = "";

  let descripcion = "";

  let imagen = "";

  for (let i = 0; i < ejercicios.length && !encontre; i++) {

    if (ejercicios[i].id == id) {

      ejercicioEntregado = ejercicios[i];

      encontre = true;

      titulo = ejercicioEntregado.titulo;

      docente = ejercicioEntregado.docente;

      descripcion = ejercicioEntregado.descripcion;

      imagen = ejercicioEntregado.imagen;

    }

  }

  document.querySelector("#tituloEjercicio").innerHTML = titulo;

  document.querySelector("#descripcionEjercicio").innerHTML = descripcion;

  let divImagen = '<img src="img/';

  divImagen += imagen;

  divImagen += '" alt="imagen"></img>';

  document.querySelector("#imagenEjercicio").innerHTML = divImagen;

}

function agregarEjercicioAPantalla(id, titulo, imagen, descripcion) {

  /\*Esta funcion recibe un ejercicio y lo agrega al div vistaEjercicio para verlo en la pantalla,

    Debe recibir:

    -Titulo

    -Descripción

    -Ruta de imagen

    -id del ejercicio para luego hacer la entrega \*/

  let yaEntrego = false;

  let descripcionBoton = "Realizar entrega"

  let habilitado = ""

  //Verifico si el alumno ya hizo una entrega para esa tarea:

  for (let i = 0; i < ejerciciosEntregados.length && !yaEntrego; i++) {

    if (ejerciciosEntregados[i].usuario.id == usuarioIngreso.id && ejerciciosEntregados[i].ejercicio.id == id) {

      habilitado = "disabled";

      descripcionBoton = "Ya se realizó entrega."

      yaEntrego = true;

    }

  }

  let htmlEjercicio = "<div id=ejercicio" + id;

  htmlEjercicio += ' > <h5 id="titEjercicio">' + titulo;

  htmlEjercicio += '</h5><p id="ejercicioDescripcion">' + descripcion;

  htmlEjercicio += '</p><img src="img/' + imagen;

  htmlEjercicio += '" alt="" id="ejercicioImagen"><br><input type="button" value="' + descripcionBoton + '" id="btnRealizarEntrega" onclick="mostrarSubirEntrega(' + id;

  htmlEjercicio += ')" ' + habilitado + ' ></div>';

  document.querySelector("#divEjercicios").innerHTML += htmlEjercicio;

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//----------------------------------------------VER EJERCICIOS RESUELTOS - ALUMNO---------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function verVentanaEjerciciosResueltos() {

  if (ejerciciosEntregados.length == 0) {

    alert("No se entregó ningun ejercicio.");

  } else {

    ocultarTodo();

    mostrar("#divEjerciciosResueltos");

    let id = "";

    let titulo = "";

    let descripcion = "";

    let imagen = "";

    let docente = "";

    let nivel = "";

    let audio = "";

    let usuario = usuarioIngreso.nombre;

    let devolucion = "";

    let corregido = "";

    //Recorro el arreglo de Ejercicios Entregados

    document.querySelector("#divEjerciciosResueltos").innerHTML = "";

    for (let i = 0; i < ejerciciosEntregados.length; i++) {

      //Si el ejercicio entrega es del alumno ingresado, entonces llamo a la funcion que muestra los ejercicios entregados del mismo

      if ((ejerciciosEntregados[i].usuario.id == usuarioIngreso.id)) {

        id = ejerciciosEntregados[i].ejercicio.id;

        titulo = ejerciciosEntregados[i].ejercicio.titulo;

        descripcion = ejerciciosEntregados[i].ejercicio.descripcion;

        imagen = ejerciciosEntregados[i].ejercicio.imagen;

        docente = ejerciciosEntregados[i].ejercicio.docente;

        nivel = ejerciciosEntregados[i].ejercicio.nivel;

        audio = ejerciciosEntregados[i].audio;

        devolucion = ejerciciosEntregados[i].devolucion;

        corregido = ejerciciosEntregados[i].corregido;

        cargarEjerciciosResueltoAPantalla(id, titulo, descripcion, imagen, docente, nivel, audio, usuario, devolucion, corregido);

      }

    }

  }

}

//Funcion para mostrar en el HTML un ejercicio entregado

function cargarEjerciciosResueltoAPantalla(id, titulo, descripcion, imagen, docente, nivel, audio, \_usuario, devolucion, corregid) {

  let htmlEjercicio = "<div id=ejercicio" + id + ">";

  htmlEjercicio += '<h5 id="titEjercicio">' + titulo + "</h5>";

  htmlEjercicio += '<p id="ejercicioDescripcion">' + descripcion + "</p>";

  htmlEjercicio +=

    '<img src="img/' + imagen + '" alt="imagen" id="ejercicioImagen">';

  htmlEjercicio += '<h5 id="docente">Docente: ' + docente + "</h5>";

  htmlEjercicio += '<h5 id="ejercicioNivel">Nivel: ' + nivel + "</h5>";

  htmlEjercicio +=

    '<audio controls><source src="audio/' +

    audio +

    '" type="audio/mpeg">Su navegador no permite el control de audio</audio>';

  htmlEjercicio += '<p id="ejercicioDevolucion">Devolución: ' + devolucion + "</p></div>";

  htmlEjercicio += "<hr>";

  document.querySelector("#divEjerciciosResueltos").innerHTML += htmlEjercicio;

}

function infromacionEstedisticaAlumno() {

  ocultarTodo();

  mostrar("#divInformacionEstadisticaAlumno");

  //Obtengo los ejercicios planteados para su nivel:

  let cantidadEjerciciosPlanteados = 0;

  let cantidadEjerciciosResueltos = 0;

  let cantidadConDevolucion = 0;

  let porcentajeResuelto = 0;

  let cantidadSinDevolucion = 0;

  for (let elemento of ejercicios) {

    if (elemento.nivel === usuarioIngreso.nivel && elemento.docente == usuarioIngreso.docente) {

      cantidadEjerciciosPlanteados += 1;

    }

  }

  //Obtengo la cantidad de ejercicios resueltos:

  for (let elemento of ejerciciosEntregados) {

    if (elemento.usuario.id === usuarioIngreso.id) {

      //Si el ejercicio resuelto es de este usuario

      cantidadEjerciciosResueltos += 1;

      //veo si tiene devolución:

      if (elemento.devolucion != "") {

        cantidadConDevolucion += 1;

      } else {

        cantidadSinDevolucion += 1

      }

    }

  }

  //regla de 3 para sacar el porcentaje

  if (cantidadEjerciciosPlanteados != 0) {

    porcentajeResuelto = (cantidadEjerciciosResueltos \* 100) / cantidadEjerciciosPlanteados;

  }

  //Agrego resultados a pantalla:

  document.querySelector("#porcentajeEjResueltos").innerHTML = "El porcentaje de ejercicios resueltos es %" + porcentajeResuelto;

  document.querySelector("#ejerciciosConSinDevolucion").innerHTML = "De un total de " + cantidadEjerciciosResueltos + " resueltos, " + cantidadConDevolucion +

    " recibieron devolución y " + cantidadSinDevolucion + " no recibieron devolución";

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//----------------------------------------------PLANTEO DE EJERCICIOS - DOCENTE-----------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function verVentanaPlanteoEjercicio() {

  ocultarTodo();

  document.querySelector("#planteoTitulo").value = "";

  document.querySelector("#planteoDescripcion").value = "";

  document.querySelector("#planteoImagen").value = "";

  mostrar("#divPlanteoEjercicio");

}

function subirEjercicio() {

  let max = 200;

  let min = 20;

  let nivel = document.querySelector("#planteoNivel").value;

  let titulo = document.querySelector("#planteoTitulo").value;

  let descripcion = document.querySelector("#planteoDescripcion").value;

  let imagen = document.querySelector("#planteoImagen").value;

  imagen = nombreDeArchivo(imagen); //imagen.replace('C:\\fakepath\\','')

  let id\_del\_usuario = usuarioIngreso.id;

  let cantcaracteres = titulo.length + descripcion.length;

  if (titulo == "" || descripcion == "" || imagen == "") {//Veo si los campos están vacíos

    alert("No pueden existir campos vacios");

  } else if (cantcaracteres < min || cantcaracteres > max) {

    alert("El total de caracteres entre titulo y descripcion debe estar entre 20 y 200");

  } else {

    crearEjercicio(titulo, descripcion, imagen, id\_del\_usuario, nivel);

    alert("Tarea agregada.")

    document.querySelector("#planteoTitulo").value = "";

    document.querySelector("#planteoDescripcion").value = "";

    document.querySelector("#planteoImagen").value = "";

    cargarEjerciciosEntregados();

  }

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//----------------------------------------------Entrega DE EJERCICIOS - Alumno--------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function entregarEjercicio() {

  let audio = nombreDeArchivo(document.querySelector("#audio").value);

  audio = nombreDeArchivo(audio);

  if (audio == "") {

    alert("Debe seleccionar un audio.");

  } else {

    crearEntregaDeEjercicio(ejercicioEntregado, audio, usuarioIngreso, "", "N");

    document.querySelector("#audio").value = "";

    alert("Entrega agregada");

    ocultarTodo();

    cargarEjercicios()

  }

}

function nombreDeArchivo(ruta) {

  let ultimaBarra = ruta.lastIndexOf("\\");

  return ruta.substring(ultimaBarra + 1);

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//----------------------------------------------INFO ESTADISTICA - DOCENTE-----------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function informacionEstadisticaDocente() {

  document.querySelector("#totalEjerciciosDelAlumno").innerHTML = "";

  ocultarTodo();

  mostrar("#divInformacionEstadisticaDocente");

  let nombreMayor = "";

  let cantidadEjerciciosMayor = 0;

  let idUsuarioActual="";

  //Obtengo cada nombre de usuario de los ejercicios entregados y los comparo con el mismo arreglo para obtener la cantidad de ejercicios resueltos.

  for(i = 0 ; i<usuarios.length; i++){

    if(usuarios[i].docente===usuarioIngreso.id && usuarios[i].tipo == "A"){

      let contador=0;

      idUsuarioActual=usuarios[i].id;

      for(y=0;y<ejerciciosEntregados.length;y++){

        if(ejerciciosEntregados[y].usuario.id===idUsuarioActual){

          contador++;

        }

      }

      if(cantidadEjerciciosMayor<contador){

        nombreMayor=usuarios[i].nombre;

        cantidadEjerciciosMayor=contador;

      }

    }

  }

  document.querySelector("#alumnosConMayorEjercicios").innerHTML = "El nombre del usuario con más ejercicios entregados es: " + nombreMayor + " con una cantidad de ejercicios de: " + cantidadEjerciciosMayor;

  //Obtengo la cantidad de ejercicios entregados para el docente

  cantidadEjerciciosParaDocente = 0;

  for (elemento of ejerciciosEntregados) {

    if ((elemento.ejercicio.docente == usuarioIngreso.id)) {

      cantidadEjerciciosParaDocente++;

    }

  }

  document.querySelector("#cantidadDeEjercicios").innerHTML = "La cantidad de ejercicios entregados para el docente es: " + cantidadEjerciciosParaDocente;

  mostrar("#divMostrarAlumnos")

  let alumnos = "";

  for (let elemento of usuarios) {

    if (elemento.tipo === "A" && elemento.docente === usuarioIngreso.id) {

      alumnos += '<option value="' + elemento.id + '">' + elemento.nombre + '(' + elemento.id + ')</option>';

    }

    document.querySelector("#regAlumnosInformacion").innerHTML = alumnos;

  }

}

function MostrarTotalEjercicios() {

  let id\_alumno\_seleccionado = document.querySelector("#regAlumnosInformacion").value;

  let nivel\_alumno\_seleccionado = "";

  //Obtengo el nivel del alumno seleccionado

  for (elemento of usuarios) {

    if (id\_alumno\_seleccionado === elemento.id) {

      nivel\_alumno\_seleccionado = elemento.nivel;

    }

  }

  //Muestro la cantidad de ejercicios planteados para el nivel del alumno

  cantidad\_de\_ejercicios\_planteados = 0;

  for (elemento of ejercicios) {

    if (elemento.nivel == nivel\_alumno\_seleccionado && elemento.docente===usuarioIngreso.id) {

      cantidad\_de\_ejercicios\_planteados++;

    }

  }

  //Muestro cantidad de ejercicios entregados

  let cantidad\_de\_ejercicios\_entregados = 0;

  for (elemento of ejerciciosEntregados) {

    if (elemento.usuario.id == id\_alumno\_seleccionado && elemento.ejercicio.nivel == nivel\_alumno\_seleccionado) {

      cantidad\_de\_ejercicios\_entregados++;

    }

  }

  document.querySelector("#totalEjerciciosDelAlumno").innerHTML = "De un total de " + cantidad\_de\_ejercicios\_planteados + " ejercicios planteados para su nivel, el alumno entrego un total de " + cantidad\_de\_ejercicios\_entregados;

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//----------------------------------------------DEVOLUCION DE TAREAS- DOCENTE-----------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function cargarEjerciciosEntregados() {

  //Esta función carga todos los ejercicios que entregaron los alumnos del docente con el que se ingresó:

  document.querySelector("#divEjerciciosEntregados").innerHTML = '<p id = "noHayEjercicioEntregado"><p>';

  ocultarTodo();

  mostrar("#divEjerciciosEntregados");

  if (ejerciciosEntregados.length == 0) {

    document.querySelector("#noHayEjercicioEntregado").innerHTML = "No hay ejercicios entregados para dar devolución.";

  } else {

    document.querySelector("#noHayEjercicioEntregado").innerHTML = "";

    let ejerciciosCargadosAPantalla = false;

    for (elemento of ejerciciosEntregados) {

      if (elemento.ejercicio.docente === usuarioIngreso.id && elemento.corregido != "S") {

        agregarEjercicioEntregadoAPantalla(elemento.id, elemento.ejercicio.titulo, elemento.audio, elemento.usuario, elemento.devolucion, elemento.corregido);

        ejerciciosCargadosAPantalla = true;

      }

    }

    if (!ejerciciosCargadosAPantalla) {

      document.querySelector("#noHayEjercicioEntregado").innerHTML = "No hay ejercicios entregados para dar devolución.";

    }

  }

}

function agregarEjercicioEntregadoAPantalla(ejercicioID, titulo, audio, usuario, devolucion, corregido) {

  let corregir = "Corregir";

  let htmlEjercicio = "<div id=ejercicio" + ejercicioID + ">";

  htmlEjercicio = '<h5 id="ejercicioTitulo">' + titulo + '</h5>';

  htmlEjercicio += '<audio controls><source src="audio/' + audio + '" type="audio/mpeg">Su navegador no permite el control de audio</audio>';

  htmlEjercicio += '<h5 id="usuarioEjercicio">Nombre de Usuario: ' + usuario.nombre + '</h5>';

  htmlEjercicio += '</h5 id="devolucionEjercicio"> Devolucion: ' + devolucion + '</h5>';

  htmlEjercicio += '<h5 id="corregidoEjercicio"> Corregido: ' + corregido + '</h5><br>';

  //Por si en algun momento se permite cambiar la devolución.

  /\*if(corregido==="S"){

    corregir="Modificar devolución"

  }\*/

  htmlEjercicio += '<input type="button" value="' + corregir + '" id="btnCorregirEntrega" onclick="corregirEntrega(' + ejercicioID;

  htmlEjercicio += ')"><hr></div>';

  document.querySelector("#divEjerciciosEntregados").innerHTML += htmlEjercicio

}

function corregirEntrega(idEntrega) {

  //Esta función muestra un ejercicio en específico que se quiere corregir, se recibe el id de la entrega

  ocultarTodo();

  document.querySelector("#divCorrecionEjerciciosEntregados").innerHTML = "";

  mostrar("#divCorrecionEjerciciosEntregados");

  let htmlCorrecion = "";

  for (entrega of ejerciciosEntregados) {

    if (entrega.id == idEntrega) {

      htmlCorrecion = '<div id="ejercicioAEntregar' + entrega.id + '">';

      htmlCorrecion += '<h5> <p>' + entrega.ejercicio.titulo + '</p></h5>';

      htmlCorrecion += '<audio controls><source src="audio/' + entrega.audio + '" type="audio/mpeg">Su navegador no permite el control de audio</audio>';

      htmlCorrecion += '</h5><p id="usuarioEjercicio">Nombre de Usuario: ' + entrega.usuario.nombre;

      htmlCorrecion += '<hr></div>';

    }

  }

  htmlCorrecion += '<label for="calificacion">Ingrese devolución:</label>';

  htmlCorrecion += '<textarea id="calificacion" name="calificacion"></textarea><br>';

  htmlCorrecion += '<input type="button" value="Calificar" id="btnCalificar" name="btnCalificar" onclick="calificarEntrega(';

  htmlCorrecion += idEntrega + ')">'

  document.querySelector("#divCorrecionEjerciciosEntregados").innerHTML += htmlCorrecion;

}

function calificarEntrega(idEntrega) {

  let devolucion = document.querySelector("#calificacion").value;

  if (devolucion == "") {

    alert("Ingrese una devolución.")

  } else {

    for (entrega of ejerciciosEntregados) {

      if (entrega.id === idEntrega) {

        agregaDevolucionATarea(entrega, devolucion);

        alert("Se asigno una devolución a la tarea");

        cargarEjerciciosEntregados();

      }

    }

  }

}

function agregaDevolucionATarea(entrega, devolucion) {

  entrega.corregido = "S"

  entrega.devolucion = devolucion;

  ocultar("#divCorrecionEjerciciosEntregados")

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//-----------------------------------Se crean los objetos y se agregan al vector------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function crearUsuario(user, nombre, pass, tipo, nivel, docente) {

  usuarios.push(new usuario(user, nombre, pass, tipo, nivel, docente));

}

function crearEjercicio(titulo, descripcion, imagen, docente, nivel) {

  ejercicios.push(new ejercicio(titulo, descripcion, imagen, docente, nivel));

}

function crearEntregaDeEjercicio(ejercicio, audio, usuario, devolucion, corregido) {

  ejerciciosEntregados.push(new EjerciciosEntregados(ejercicio, audio, usuario, devolucion, corregido));

}

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//--------------------------------------------------DATOS PRE-CARGADOS----------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

//------------------------------------------------------------------------------------------------------------------------------//

function datosPreCargados() {

  //Cargo 2 docentes:

  crearUsuario("doc1", "Docente 1", "1234aB", "D", "", "");

  crearUsuario("doc2", "Docente 2", "1234aB", "D", "", "");

  crearUsuario("doc3", "Docente 3", "1234aB", "D", "", "");

  //Cargo 2 alumnos:

  crearUsuario("alum1", "Alumno 1", "1234aB", "A", "1", "doc1");

  crearUsuario("alum2", "Alumno 2", "1234aB", "A", "1", "doc1");

  crearUsuario("alum3", "Alumno 3", "1234aB", "A", "3", "doc3");

  //Cargo ejercicios

  crearEjercicio("Este es el titulo del Ejercicio 1", "esta es la descripción del ejercicio", "ej1.png", "doc1", "1");

  crearEjercicio("Este es el titulo del Ejercicio 2", "esta es la descripción del ejercicio", "ej2.png", "doc1", "1");

  crearEjercicio("Este es el titulo del Ejercicio 3", "esta es la descripción del ejercicio", "ej3.png", "doc1", "1");

  crearEjercicio("Este es el titulo del Ejercicio 4", "esta es la descripción del ejercicio", "ej4.png", "doc1", "1");

  crearEjercicio("Este es el titulo del Ejercicio 5", "esta es la descripción del ejercicio", "ej5.png", "doc1", "2");

  crearEjercicio("Este es el titulo del Ejercicio 6", "esta es la descripción del ejercicio", "ej6.png", "doc1", "1");

  //Agrego etregas:

  crearEntregaDeEjercicio(ejercicios[0], "ej1.m4a", usuarios[3], "", "N");

  crearEntregaDeEjercicio(ejercicios[0], "ej2.m4a", usuarios[4], "", "N");

  //Agrego devolución:

  agregaDevolucionATarea(ejerciciosEntregados[0], "Excelente tarea, cumplio con los requisitos y esta listo para cumplir con el resto de tareas");

}

# Datos Precargados

Usuarios Creados:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nombre | Usuario | Contraseña | Tipo de Usuario | Nivel | Docente |
| Docente 1 | doc1 | 1234aB | Docente | “” | “” |
| Docente 2 | doc2 | 1234aB | Docente | “” | “” |
| Docente 3 | doc3 | 1234aB | Docente | “” | “” |
| Alumno 1 | alum1 | 1234aB | Alumno | Básico | doc1 |
| Alumno 2 | alum2 | 1234aB | Alumno | Básico | doc1 |
| Alumno 3 | alum3 | 1234aB | Alumno | Avanzado | doc3 |

Ejercicios Cargados:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Titulo | Descripción | Imagen | Docente | Nivel | ID |
| Este es el título del Ejercicio 1 | Esta es la descripción del ejercicio 1 | Ej1.png | Doc1 | Básico | 1 |
| Este es el título del Ejercicio 2 | Esta es la descripción del ejercicio 2 | Ej2.png | Doc1 | Básico | 2 |
| Este es el título del Ejercicio 3 | Esta es la descripción del ejercicio 3 | Ej3.png | Doc1 | Básico | 3 |
| Este es el título del Ejercicio 4 | Esta es la descripción del ejercicio 4 | Ej4.png | Doc1 | Básico | 4 |
| Este es el título del Ejercicio 5 | Esta es la descripción del ejercicio 5 | Ej5.png | Doc1 | Intermedio | 5 |
| Este es el título del Ejercicio 6 | Esta es la descripción del ejercicio 6 | Ej6.png | Doc1 | Básico | 6 |

Ejercicios Entregados:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Alumno | Audio | Docente | Devolución | ID |
| Alumno 1 | Ej1.m4a | Doc1 | “Excelente tarea, cumplió con los requisitos y está listo para seguir con el resto de tareas” | 1 |
| Alumno 2 | Ej2.m4a | Doc2 | “” | 2 |