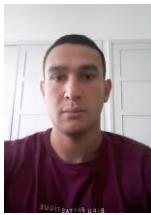


Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACION 1
DOCUMENTO DE ANÁLISIS



Pons Martínez, Oscar Diego (264383)



Eduard Junior Suarez Alvez (252860)



Hirigoyen Sierra, Leandro (272402)

[Grupo 7]

Docente: [Santiago Fagnoni]

[AP/ATI]

[Fecha de entrega del documento (27-05-2021)]

INDICE

DOCUMENTO DE ANALISIS	PAGINA	3
INFORME DE TESTING	PAGINA	19
CODIGO JS	PAGINA	20
LIBRERÍA	PAGINA	44
CODIGO HTML	PAGINA	49
INFORMACION PRECARGA	PAGINA	55

Índice

1.	Descripción general del problema a resolver.....	6
1.1	Tipos de usuario del sistema	6
1.2	Listado de funcionalidades.....	6
2.	Detalle de Funcionalidades.....	7
2.1	F01 – Registrarse.....	7
2.1.1	Acceso.....	7
2.1.2	Descripción	7
2.1.3	Validaciones.....	8
2.2	F02 – Ingresar en la Aplicación (Logueo)	8
2.2.1	Acceso.....	8
2.2.2	Descripción:	8
2.2.3	Interfaz de usuarios:.....	8
2.2.4	Validaciones:.....	8
2.3	F03 Asignar alumnos un nivel	9
2.3.1	Acceso : Docente	9
2.3.2	Descripción:	9
2.3.3	Interfaz:	9
2.3.4	Validaciones:.....	9
2.4	F04 Plantear ejercicios:	9
2.4.1	Acceso: Docente y Alumno	9
2.4.2	Descripción :	9
2.4.3	Interfaz :	10
2.4.4	Validaciones (Planteo Docente):.....	11
2.5	F05 Redactar devoluciones para las tareas entregadas.	11
2.5.1	Acceso : Docente	11

2.5.2 Descripción :	11
2.5.3 Interfaz:	11
2.5.4 Validaciones:	12
2.6 F01 Registrarse en la Aplicación:	12
2.6.1 Interfaz:	12
2.6.2 Validaciones:	12
2.7 F02 Ingresar en la Aplicación(ver funcionalidad ingresada en perfil docente)	13
2.8 F03 Ver ejercicios planteados :	13
2.8.1 Acceso : Alumno.....	13
2.8.2 Descripción :	13
2.8.3 Interfaz:	13
2.8.4 Validaciones:	14
2.9 F04 Realizar las entregas :	14
2.9.1 Acceso:	14
2.9.2 Descripción:	14
2.9.3 interfaz :	15
2.9.4 Validaciones:	16
2.10 F05 Ver ejercicios:	16
2.10.1 Acceso:	16
2.10.2 Descripción:	16
2.10.3 Interfaz :	16
2.10.4 Validaciones:	16
2.11 F06 Ver Estadísticas:	16
2.11.1 Acceso:	16
2.11.2 Descripción:	17
2.11.3 Interfaz :	17
2.11.4 Validaciones:	17

2.12.1 Acceso:.....	17
2.12.2 Descripción:	18
2.12.3 Interfaz :	18
2.12.4 Validaciones:.....	18

1. Descripción general del problema a resolver

Un profesor de guitarra desea disponer de una aplicación para poder gestionar las clases que brinda a distancia, de manera de poder tener un vínculo cercano con sus alumnos. Para que la aplicación pueda ser de utilidad, la crearemos para que múltiples docentes puedan hacer uso de ella.

1.1 Tipos de usuario del sistema

1 – Docente

2_ Alumno

1.2 Listado de funcionalidades

Docente:

F01 – Registrarse.

F02 – Ingresar en la aplicación.

F03 – Asignar a sus alumnos un nivel (“inicial” por defecto, “intermedio” o “avanzado”).

F04 – Plantear ejercicios para sus alumnos de un determinado nivel .

F05 – Redactar devoluciones para las tareas entregadas por los alumnos.

F06 – Visualizar información estadística.

Alumno:

F01 – Registrarse en la aplicación.

F02 – Ingresar en la aplicación (ver funcionalidad indicada en perfil docente).

F03 – Ingresar en la aplicación (ver funcionalidad indicada en perfil docente).

F04 – Realizar la entrega de un ejercicio.

F05 – Ver todos los ejercicios que ha resuelto.

F06 – Visualizar información estadística.

2. 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.

2.1 F01 – Registrarse

REGISTRO DOCENTE	REGISTRO ALUMNO
Tipo Usuario- Docente :	
Nombre : <input type="text"/>	Nombre : <input type="text"/>
Usuario: <input type="text"/>	Usuario: <input type="text"/>
Contraseña: <input type="password"/>	Contraseña: <input type="password"/>
	Docente: <input (carlos)"="" type="text" value="Carlos "/>
<input type="button" value="Registrarse"/>	<input type="button" value="Registrarse"/>

2.1.1 Acceso

Podrá acceder Docente y Alumno.

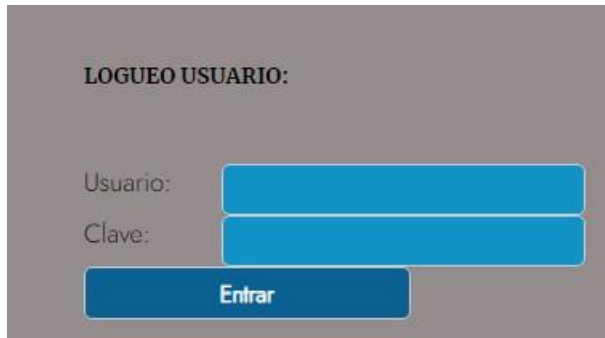
2.1.2 Descripción

Docente: deberá ingresar nombre , nombre de usuario y contraseña.

2.1.3 Validaciones

Docente : El nombre de usuario no se puede repetir en dos usuarios diferentes y la contraseña debe tener un mínimo de 4 caracteres con una mayúscula una minúscula y un numero. *Se agregó que Nombre y usuario deban de tener mínimo de 2 caracteres.*

2.2 F02 – Ingresar en la Aplicación (Logueo)



LOGUEO USUARIO:

Usuario:

Clave:

2.2.1 Acceso

Podrá acceder Docente y Alumno.

2.2.2 Descripción:

para ingresar a la aplicación ya debe contar con usuario y contraseña.

2.2.3 Interfaz de usuarios:

2.2.4 Validaciones:

El nombre de usuario podrá estar escrito con mayúsculas o minúsculas pero la contraseña debe ser case sensitive.

2.3 F03 Asignar alumnos un nivel

2.3.1 Acceso : Docente

2.3.2 Descripción:

Asignar a sus alumnos un nivel (“inicial” por defecto, “intermedio” o “avanzado”).

2.3.3 Interfaz:



CAMBIAR NIVEL ALUMNO

Alumnos : quike Nivel: Inicial ▼

Asignar Nivel: Intermedio ▼

Cambiar

2.3.4 Validaciones:

Deberá poder cambiar el nivel en cualquier momento, pero solamente desde inicial a intermedio y de intermedio a avanzado, nunca al revés.

2.4 F04 Plantear ejercicios:

2.4.1 Acceso: Docente y Alumno

2.4.2 Descripción :

Plantear ejercicios para sus alumnos de un determinado nivel. En el perfil alumno, solo podrán ver esos ejercicios los alumnos del docente que correspondan a ese nivel.

2.4.3 Interfaz :

Docente :

PLANTEAR EJERCICIO

Título del ejercicio:

Foto Ejercicio: Ningún archivo seleccionado

Descripción:

Nivel :

Alumno:

LISTA DE SUS EJERCICIOS A ENTREGAR					
Id de Ejercicio	Título	Descripción	Foto	Subir Entrega	Enviar
29	grabar en acustico	Ehacerlo para el martes		<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	<input type="button" value="Enviar"/>
30	grabar en elctro	Ehacerlo para el martes		<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	<input type="button" value="Enviar"/>

2.4.4 Validaciones (Planteo Docente):

El ejercicio constará de un título, una imagen y una descripción en la que se plantea el objetivo a resolver. La suma total de caracteres entre título y descripción no puede superar los 200 caracteres ni ser menor a 20 caracteres, no importa la distribución mientras entre ambos campos no se superen los 200 caracteres y haya al menos 20 caracteres, además ninguno de los campos puede quedar vacío. Las imágenes estarán precargadas en una carpeta imágenes, de ahí se seleccionará el archivo correspondiente.

2.5 F05 Redactar devoluciones para las tareas entregadas.

2.5.1 Acceso : Docente

2.5.2 Descripción :

Redactar devoluciones para las tareas entregadas por los alumnos.

2.5.3 Interfaz:

REDACTAR EJERCICIO				
Alumno	ID de Ejercicio	Solucion	Redactar	Enviar Devolucion
lopp	1	<div> <div>▶</div> <div>0:00 / 0:25</div> <div> <div></div> </div> <div>🔊</div> <div>⋮</div> </div>	<div></div>	<div>Enviar</div>

2.5.4 Validaciones:

Solo se listarán en la aplicación las tareas que correspondan a ese docente y no hayan recibido ya una devolución.

Alumno

2.6 F01 Registrarse en la Aplicación:

El nombre de usuario no se puede repetir en dos usuarios diferentes y la contraseña debe tener un mínimo de 4 caracteres con una mayúscula una minúscula y un numero.

2.6.1 Interfaz:



REGISTRO ALUMNO

Nombre :

Usuario:

Contraseña:

Docente:

2.6.2 Validaciones:

El nombre de usuario no se puede repetir en dos usuarios diferentes y la contraseña debe tener un mínimo de 4 caracteres con una mayúscula una minúscula y un numero. *Se agregó que Nombre y usuario deban de tener mínimo de 2 caracteres.*

2.7 F02 Ingresar en la Aplicación(ver funcionalidad ingresada en perfil docente)

2.8 F03 Ver ejercicios planteados :



2.8.1 Acceso : Alumno

2.8.2 Descripción :

Ver todos los ejercicios que ha planteado su docente para su nivel.

En esta interfaz se deberá agregar un buscador para que el usuario pueda buscar ejercicios según el criterio de búsqueda que escriba. La búsqueda se hará en los títulos de los ejercicios, en caso de no obtener resultados, se buscará si hay resultados que coincidan en la descripción. No se buscará en título y descripción simultáneamente, solo si no hay resultados en la búsqueda por título se pasará al segundo criterio. En caso de no haber resultados para la búsqueda, se mostrará un texto que diga “No hay resultados que coincidan con su búsqueda”.

2.8.3 Interfaz:

LISTA DE SUS EJERCICIOS A ENTREGAR					
Id de Ejercicio	Titulo	Descripcion	Foto	Subir Entrega	Enviar
29	grabar en acustico	Ehacerlo para el martes		<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	<input type="button" value="Enviar"/>
30	grabar en elctro	Ehacerlo para el martes		<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	<input type="button" value="Enviar"/>

BUSCAR EJERCICIO

Buscar

Buscar

Ejercicios

Id de Ejercicio	Titulo	Descripcion	Foto	Subir Entrega	Enviar
31	Ritmo distorcionado	Ehacerlo para el martes		<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado	<input type="button" value="Enviar"/>
			La marcha de los santos		

2.8.4 Validaciones:

ingresar titulo o parte del mismo.

2.9 F04 Realizar las entregas :

2.9.1 Acceso:

Alumno:

2.9.2 Descripción:

Realizar la entrega de un ejercicio, adjuntando un audio correspondiente a dicha tarea, quedando la tarea pendiente de revisión por parte del docente. Los audios estarán precargados en la aplicación en una carpeta que se llame audios, de ahí se seleccionará el archivo correspondiente. La entrega la podrá realizar tanto desde La lista de Ejercicios a entregar como de la lista que surja en la funcionalidad de Búsqueda.

2.9.3 interfaz :

LISTA DE SUS EJERCICIOS A ENTREGAR

Id de Ejercicio	Titulo	Descripcion	Foto	Subir Entrega	Enviar
29	grabar en acústico	Ehacerlo para el martes		<div> <div>Seleccionar archivo</div> <div>Ningún archivo seleccionado</div> </div>	Enviar
30	grabar en elctro	Ehacerlo para el martes		<div> <div>Seleccionar archivo</div> <div>Ningún archivo seleccionado</div> </div>	Enviar

BUSCAR EJERCICIO

Buscar

Ritmo

Buscar

Ejercicios

Id de Ejercicio	Titulo	Descripcion	Foto	Subir Entrega	Enviar
31	Ritmo distorcionado	Ehacerlo para el martes		<div> <div>Seleccionar archivo</div> <div>Ningún archivo seleccionado</div> </div>	Enviar
					

2.9.4 Validaciones:

El alumno no puede entregar dos veces el mismo ejercicio, por lo que no se le deberá permitir al usuario iniciar la entrega de un ejercicio ya entregado anteriormente.

2.10 F05 Ver ejercicios:



2.10.1 Acceso:

Alumno

2.10.2 Descripción:

ver todos los ejercicios resueltos.

2.10.3 Interfaz :

LISTA DE SUS EJERCICIOS RESUELTOS			
Id de Ejercicio	Titulo	Descripcion	Foto
25	ejercicio 3 en do mayor	locar con notas negativas	
26	Ritmo del agua	grabarlo y enviarlo en un zip	

2.10.4 Validaciones:

Ninguna.

2.11 F06 Ver Estadísticas:

2.11.1 Acceso:

Docente.

2.11.2 Descripción:

- * Mostrar el nombre del alumno que ha resuelto más ejercicios. Si son varios mostrar todos los nombres. Los alumnos deberán corresponder a ese docente, sin discriminar el nivel.
- * Mostrar la cantidad de ejercicios que se han entregado en total para ese docente.
- * Seleccionar de un desplegable a uno de sus alumnos y mostrar el total de ejercicios que se habían planteado (por el docente logueado) para su nivel y cuantos se han resuelto. *NOTA: En el desplegable se agregó el Nivel del alumno para uso meramente informativo.*

2.11.3 Interfaz :

Estadísticas

Alumnos con mas Ejercicios Resueltos

Ver lista de alumnos con mas Ejercicios Resueltos

Listar

Cantidad total de ejercicios resueltos

Ver cantidad total de ejercicios resueltos

Listar

Total Ejercicios por Nivel y resueltos por el alumno

Alumnos : quike Nivel: Inicial ▼

Listar

2.11.4 Validaciones:

Ninguna

2.12.1 Acceso:

Alumno.

2.12.2 Descripción:

Ver porcentaje de ejercicios resueltos, respecto a los planteados para su nivel por su docente

Ver cuantos ejercicios que resolvió tuvieron devoluciones y cuantos no.

2.12.3 Interfaz :



2.12.4 Validaciones:

ninguna

#	¿Qué estoy probando?	Usuario	¿Con qué datos?	¿Qué espero como resultado?	¿Qué Obtuve?
1	Registro	usuarioD (Docente)	Anita - anita - Ab123	Registrarse como docente	Mensaje de registro con éxito y se agrega a lista de docentes (reflejado en Registro Alumno desplegable)
2	Error en registro	usuarioD (Docente)	Anita - anita - ab123	Registrarse como docente	Aviso de formato para clave.
3	Error en registro	usuarioD (Docente)	A - a - ab123	Registrarse como docente	Mensaje de mínimo de 2 caracteres para Nombre y Usuario
4	Error en registro	usuarioD (Docente)	Carlos - carlos - Ab123 (uso un usuario ya precargado) También se probó con un registro recién realizado	Registrarse como docente	no se puede repetir los usuarios
5	Registro	usuarioA (Alumno)	Nando - nando - Ab123	Registrarse como alumno	Mensaje de registro con éxito
6	Error en registro - Contraseña	usuarioA (Alumno)	Nando - nando - ab123	Registrarse como alumno	Aviso de formato para clave.
7	Error en registro	usuarioA (Alumno)	N - n - Ab123	Registrarse como alumno	Mensaje de mínimo de 2 caracteres para Nombre y Usuario
8	Error en registro	usuarioA (Alumno)	Nego - nego - Ab123 (uso un usuario ya precargado) También se probó con un registro recién realizado	Registrarse como alumno	no se puede repetir los usuarios
9	Login	usuarioD (Docente)	Carlos y Ab123 (Precargado)	Logearse como Docente	Muestra Interfaz para Docente
10	Error en login	usuarioD (Docente)	Mauricio y Ab123 - Carlos (Docente precargado) - ab123	Mensajes de error al ingresar datos	Mensaje de Usuario o Clave incorrectos
11	Login	usuarioA (Alumno)	negó y Ab123 y con Negro Ab123 (Alumno precargado)	Logearse como Alumno	Muestra Interfaz para Alumno
12	Error en login	usuarioA (Alumno)	German y Ab123 - Negro (Alumno precargado) - ab123	Mensajes de error al ingresar datos	Mensaje de Usuario o Clave incorrectos
13	Cambiar Nivel Alumno	usuarioD (Docente)	Se ingresa con el Docente Carlos quike Nivel: Inicial pasarlo a otro nivel	le cambia el Nivel al Alumno	Lo hace correctamente y también cambia el Nivel del Alumno en el desplegable de la funcionalidad para Estadísticas - total Ejercicios por Nivel y resueltos por el Alumno
14	Error en Cambiar Nivel Alumno	usuarioD (Docente)	quike Nivel: Avanzado Intento pasarlo a Nivel Inicial	que no deje realizar el cambio - No se puede bajar de Nivel a un alumno	Muestra mensaje "No se puede cambiar a un nivel menor"
15	Plantear Ejercicio	usuarioD (Docente)	Se ingresa: Título. Se carga archivo. Se ingresa Descripción y se elige Nivel en desplegable.	Que cargue el ejercicio	Mensaje "El ejercicio se cargo" Se chequea en consola que el ejercicio fue agregado
16	Error Plantear Ejercicio	usuarioD (Docente)	Se ingresaron vacíos - Se probaron las diferentes combinaciones. Se prueba sin seleccionar archivo	Que de error	Da error en cuando Título y Mensaje no cumple con la petición. Para el caso del archivo vacío el mensaje es "El archivo está vacío"
17	Redactar devolución y Entregar	usuarioD (Docente)	Se cargan datos en el área Redactar de una resolución entregada por quike	Que la entrega se exitosa	Muestra "ENVIADO QUIKE"
18	Estadísticas - Alumnos con mas Ejercicios Resueltos	usuarioD (Docente)	Precargados	Que muestre el nombre del/los alumnos que mas ejercicios resolvieron	Despliega Quike
19	Estadísticas - Cantidad total de ejercicios resueltos/entregados que se han entregado	usuarioD (Docente)	Precargados	Muestra cantidad de ejercicios resueltos/entregados por sus alumnos	Carga 11 que es lo correcto.
20	Estadísticas - Total Ejercicios por Nivel y resueltos por el alumno que se escoge	usuarioD (Docente)	Se elige un alumno del desplegable	Total Ejercicios que se plantearon para el nivel del Alumno. ejercicios resolvió el alumno seleccionado en el desplegable	Mensajes correctos
21	Lista de sus Ejercicios a Entregar y Enviar resolución	usuarioA (Alumno)	Se carga archivo de audio (tomar nota de Id para corroborar que se cargue en la otra lista) Negro - Id de Ejercicio 30	Enviar la resolución y que lo agregue a la lista de ejercicios resueltos	Mensaje "Enviado" y se agrega a lista de ejercicios resueltos
22	Lista de sus Ejercicios Resueltos	usuarioA (Alumno)	Carga los archivos que tiene resueltos. Negro Probamos con Dario que no tiene ejercicios resueltos	Muestra lista de ejercicios resueltos- O da mensaje de que no tiene ejercicios resueltos	Mensaje "Carga Ejercicios Resueltos y si no los tiene muestra mensaje: "No tiene ejercicios resueltos"
23	Buscar Ejercicios y enviar resolución	usuarioA (Alumno)	Busqueda - Martes - CARGA - Luego se carga archivo de audio (tomar nota de Id para corroborar que se cargue en la otra lista) Dario - Id de Ejercicio 33	Enviar la resolución y que lo agregue a la lista de ejercicios resueltos	Busqueda exitosa, despliega los ejercicios que contienen lo buscado se agrega a lista de ejercicios resueltos
24	Estadísticas - Porcentaje ejercicios resueltos	usuarioA (Alumno)	Dario. Se cargaron dos ejercicios	Porcentaje de ejercicios resueltos contra ejercicios planteados por su docente por su nivel. TotalEjercicios para el nivel = 7 Total Total Resoluciones 2	28%
25	Estadísticas - Ejercicios resueltos con devolución y cuanto sin devolución	usuarioA (Alumno)	Nego	Cantidad de Ejercicios resueltos CON devolución Cantidad de ejercicios resueltos SIN devolución	Muestra cuantos resueltos CON DEVOLUCION Y MUESTRA CUANTOS EJERCICIOS NO HAN RECIBIDO DEVOLUCION

```

//VARIABLES Y ARREGLOS GLOBALES DE TODO EL PROGRAMA
let nom;//nombre
let usu;//usuario
let contra;//contraseña
let prof;//nombre profesor
let errorA;//error
let muestra;//mostrar
let usuariosA=[];//lista alumnos
let usuariosD=[];//lista docentes
let ejercicios=[];//lista ejercicios
let resoluciones=[]//lista de resoluciones
let unUsuarioD;//undocente
let unUsuarioA;//un alumno
let tipUsu;//tipo de usuario nque se loguea
let unBotonHTML;//boton
let elEjercicio;
let mens;//mensaje para usuario
let nombreUsuDoc;//nombre del usuario docente logueado
let errorD="";//mensaje de error
let muestraD="";//mostrar mensaje
let ejerciciosAlum;//ejercicios de alumno
let mensaje;//mensaje
let resDeEjercicio;//resoucion de ejercicio
let mensajesinoHay;//mensaje al usuario
let resoDeAlumno=[];//resoluciones de alumno
let resolution;//resolucion
let unResolucion=[];//resoluciones
let unRe;//una resolucion
let unE;//ejercicio
let ejD;//ejercicio de docente
let ejerDelDocente=[];//ejercicios del docente
let sms; //mensaje al usuario
let er;//emensaje de erros
let env=0;
let totalResoluciones=0;//total de resoluciones de ejercicio por nivel
let totalEjerciciosNivel=0;//total de ejercicios de un nivel
let nivelUsu; //nivel del alumno
let unAlumno;//cada uno de los que tiene resoluciones
let alumnoConMasRes=[];//el con mas resoluciones
let max = 0;//maximo
let susAlumnos = [];//alumnos por docente
let totalRes = 0;//total resoluciones
let a=[];
let xAlumnos=[]; //alumnos
let cadaAlumno;//un alumno
let todosLosAlumnosConRes=[];//alumnos con resolucion
let b=[];
let misAlumnos=[];//alumnos de un docente

```

```

let logReg;//logueo registro
let idEjerciciosEnviados=[];//paraguardar el id del ejercicio enviado
let ejerciciosDocenteLogueado=[];//ejercicios docente logueado
let resXAlumno=0;//resolucion por alumno
let nInicial=0;//nivel inicial
let nIntermedio=0;//nivel intermedio
let nAvanzado=0;//nivel avanzado
let doc;

//-----CLASES-----
class UsuarioD//DOCENTES
{
    constructor(pNombre, pUsuario, Pcontrasenia){
        this.nombre = pNombre;
        this.usuario = pUsuario;
        this.contrasenia = Pcontrasenia;
    }
}
class UsuarioA //ALUMNOS
{
    constructor(pNombre, pUsuario, pcontrasenia, profesor,pnivel){
        this.nombre = pNombre;
        this.usuario = pUsuario;
        this.contrasenia = pcontrasenia;
        this.docent = profesor;
        this.Nivel=pnivel;
    }
    cambiarNivel(nombreusu,nuevoNivel)
    {
        if(this.usuario===nombreusu){
            this.Nivel=nuevoNivel;
            return true;
        }
        return false;
    }
}
class Ejercicio//EJERCICIOS
{
    constructor(pid,ptitulo, pimagen, pdescripcion, pnivel,pdocente){
        this.id=pid;
        this.Titulo=ptitulo;
        this.Imagen=pimagen;
        this.Descripcion=pdescripcion;
        this.Nivel=pnivel;
        this.Docente=pdocente;
    }
}

```

```

}
class Resolucion{//RESOLUCIONES
    constructor (unid, paudio, pdevolucion, alumno){
this.id=unid;//vinculacion con la tarea
this.audio=paudio;//parte del alumno
this.devolucion=pdevolucion;//parte del docente
this.Alumno=alumno;//vinculacion con el alumno
    }
}
    precargaDocente();
    precargaAlumno();
    precargaResolucion();
    precargaEje();
//-----EVENTOS DE CLICK DE TODO EL PROGRAMA-----
//-----
//REGISTRO Y LOGUEO
    document.querySelector("#btnRegistroD").addEventListener("click", RegistroD);//re-
gistro docente
document.querySelector("#btnRegistroA").addEventListener("click", RegistroA);//re-
gistro alumno
document.querySelector("#btnEntrar").addEventListener("click", Loguearse);//Lo-
guearse
//-----
//DOCENTE

document.querySelector("#btnNivel").addEventListener("click", asignarNivel);//asig-
nar nivel a alumno
document.querySelector("#btnAgregarEjer").addEventListener("click", agregarEjerci-
cio);//agregar un ejercicio
//-----
//ESTADISTICAS DOCENTE
document.querySelector("#btnBuscaAlumno").addEventListener("click", listaAlumnosMa-
sEjercicios);//listado de alumno con mas ejercicios
document.querySelector("#btnBuscaCantidad").addEventListener("click", cantidadTotal-
DeEjerciciosEntregados);
document.querySelector("#btnListarEstadisticas").addEventListener("click", cantida-
dEjerciciosNivel);
//-----
//ALUMNO

document.querySelector("#btnBuscarEjer").addEventListener("click", buscarListarEjer-
cio);//buscar y mostrar el ejercicio
//-----
//ESTADISTICAS ALUMNO

```

```

document.querySelector("#btnEstadisticaUno").addEventListener("click", porcentaje );
document.querySelector("#btnEstadisticaDos").addEventListener("click", cantidadEjer-
ConDevolucion);
//-----

document.querySelector("#btnVolver").addEventListener("click", volverLogueoRegis-
tro);//volver al inicio

//----- MANEJO INTERFAZ -----
-----

let botonesMenu = document.querySelectorAll(".btnSeccion");
for (let i = 0; i < botonesMenu.length; i++) {
    const boton = botonesMenu[i];
    boton.addEventListener("click", cambiarSeccionClick);
}
function cambiarSeccionClick() {//cambia el id del boton transformarlo
//en el id de la seccion que quiero mostrar, luego llamo a cambiarSeccion
    let idBoton = this.getAttribute("id"); //"btnSeccionAgregar"
    let idSeccionMostrar = idBoton.charAt(3).toLowerCase() + idBoton.subs-
tring(4); //"seccionAgregar"
    cambiarSeccion(idSeccionMostrar);
}
function cambiarSeccion(idSeccion) {//llega id de seccion, oculta el
    //resto y muestra la que necesito
    let secciones = document.querySelectorAll(".seccion");
    for (let i = 0; i < secciones.length; i++) {
        const seccion = secciones[i];
        seccion.style.display = "none";//NO SE VE
    }
    document.querySelector("#" + idSeccion).style.display = "block";//SE VE
}
//mostrarBotones("docente");
function mostrarBotones(tipoUsuario) {
    let botones = document.querySelectorAll(".btnSeccion");
    for (let i = 0; i < botones.length; i++) {
        const boton = botones[i];
        boton.style.display = "none";
    }
    let botonesMostrar = document.querySelectorAll("." + tipoUsuario);
    for (let i = 0; i < botonesMostrar.length; i++) {
        const unBoton = botonesMostrar[i];
        unBoton.style.display = "block";
    }
}

```

```
//-----PRECARGAS -----
//-----PRECARGAS      DOCENTES-----

function precargaDocente() {
  agregarDocente("Carlos", "carlos", "Ab123");
  agregarDocente("Juan", "juan", "Ab123");
  agregarDocente("Diego", "diego", "Ab123");
  agregarDocente("Santiago", "santiago", "Ab123");
}
function agregarDocente(elNombre, elUsuario, laContrasenia) {
  let valNombre = textoObligatorioNom(elNombre);
  let valUsuario = textoObligatorioUsu(elUsuario);
  let valContrasenia = validar_clave(laContrasenia);
  if (valNombre && valUsuario && valContrasenia) {
    let docenteCarga = new UsuarioD();
    docenteCarga.nombre = elNombre;
    docenteCarga.usuario = elUsuario;
    docenteCarga.contrasenia = laContrasenia;
    usuariosD.push(docenteCarga);
  }
}
//-----PRECARGAS      ALUMNOS-----

function precargaAlumno() {
  agregarAlumno("Quike", "quike", "Ab123", "carlos", "Inicial");
  agregarAlumno("Jorge", "Jorje", "Ab123", "juan", "Inicial");
  agregarAlumno("Nego", "nego", "Ab123", "diego", "Inicial");
  agregarAlumno("maria", "maria", "Ab123", "diego", "Inicial");
  agregarAlumno("dario", "dario", "Ab123", "diego", "Inicial");
  agregarAlumno("Peca", "peca", "Ab123", "juan", "Inicial");
  agregarAlumno("bruno", "bruno", "Ab123", "juan", "Inicial");
  agregarAlumno("santi", "santi", "Ab123", "carlos", "Inicial");
  agregarAlumno("glass", "glass", "Ab123", "carlos", "Inicial");
  agregarAlumno("lopp", "lopp", "Ab123", "carlos", "Inicial");
}
function agregarAlumno(elNombre, elUsuario, laContrasenia, elDocente, elNivel) {
  let valNombre = textoObligatorioNom(elNombre);
  let valUsuario = textoObligatorioUsu(elUsuario);
  let valContrasenia = validar_clave(laContrasenia);
  if (valNombre && valUsuario && valContrasenia) {
    let alumnoCarga = new UsuarioA();
    alumnoCarga.nombre = elNombre;
    alumnoCarga.usuario = elUsuario;
    alumnoCarga.contrasenia = laContrasenia;
    alumnoCarga.docent = elDocente;
  }
}
```



```

        alumnoCarga.Nivel = elNivel;
        usuariosA.push(alumnoCarga);
    }
}

//-----PRECARGAS -- EJERCICIOS-----
-----

function precargaEje() {
    //carlos
    agregarEje(1,"Ejecutar en Do menor", "ej1.png","El ejercicio debe contemplar op-
ciones","Inicial", "carlos");//CAMBIÉ A INICIAL!!!!
    agregarEje(2,"Establecer ritmos de ocho en ocho", "ej2.png","Se debe ejecu-
tar solo indicado","Inicial", "carlos");
    agregarEje(3,"Ritmo latino", "ej3.png","Puede hacer variantes","Inicial", "car-
los");
    agregarEje(4,"Introduccion a ritmos", "ej4.png","Se deben realizar varian-
tes","Inicial", "carlos");
    agregarEje(5,"Primeras notas", "ej5.png","La ejecucion es libre","Ini-
cial", "carlos");
    agregarEje(6,"segundas notas", "ej6.png","La ejecucion es libre","Ini-
cial", "carlos");
    agregarEje(7,"terceras notas", "ej7.png","La ejecucion es libre","Avan-
zado", "carlos");
    agregarEje(8,"cuartas notas", "ej8.png","La ejecucion es libre","Avan-
zado", "carlos");
    //juan
    agregarEje(9,"Introduccion a ritmos", "ej7.png","Se deben realizar varian-
tes","Inicial", "juan");
    agregarEje(10,"Ritmo dominicano", "ej8.png","Ejercicio de calecita libre","Ini-
cial", "juan");
    agregarEje(11,"Ritmo colombiano", "ej1.png","hacerlo para el lu-
nes sin falta ","Intermedio", "juan");
    agregarEje(12,"Ritmo africano", "ej2.png","Ejercicio de ejecucion libre","Avan-
zado", "juan");
    agregarEje(13,"Ritmo paraguayo", "ej3.png","Ejercicio de ejecucion lim-
pia ","Avanzado", "juan");
    agregarEje(14,"Ritmo uruguayo", "ej4.png","Ejercicio de ejecucion ce-
rrada en sol","Inicial", "juan");
    agregarEje(15,"Ritmo brasilero", "ej5.png","Ejercicio de ejecucion ce-
rrada en do","Inicial", "juan");
    agregarEje(16,"Ritmo argentino", "ej6.png","Ejercicio de ejecucion libre","Ini-
cial", "juan");
    //santiago
    agregarEje(17,"Ritmo electronico", "ej5.png","tocar y rasguear","Ini-
cial", "santiago");
    agregarEje(18,"ejercicio complemetario", "ej5.png","to-
car solo con una mano","Inicial", "santiago");
}

```

```

    agregarEje(19,"parcial numero 1", "ej9.png","tocar con sol,do y re","Ini-
cial", "santiago");
    agregarEje(20,"ejercicio 4", "ej1.png","tocar con sol,do y re","Inicial", "san-
tiago");
    agregarEje(21,"ejercicio 2do semestre", "ej2.png","tocar con sol,do y re","Ini-
cial", "santiago");
    agregarEje(22,"ejercicio 3", "ej3.png","tocar con sol,do y re","Interme-
dio", "santiago");
    agregarEje(23,"ejercicio 4", "ej4.png","tocar con sol,do y re","Interme-
dio", "santiago");
    agregarEje(24,"ejercicio 5", "ej5.png","tocar con sol,do y re","Interme-
dio", "santiago");
    //diego
    agregarEje(25,"ejercicio 3 en do mayor", "ej5.png","tocar con notas negati-
vas","Inicial", "diego");//entregado
    agregarEje(26,"Ritmo del agua", "ej4.png","grabarlo y enviarlo en un zip","Ini-
cial", "diego");//entregado
    agregarEje(27,"cantar y tocar", "ej1.png","este ejerccio lleva 10 puntos","In-
termedio", "diego");
    agregarEje(28,"grabar con pedalera distorcionada", "ej2.png","el mierco-
les 23 pronto","Intermedio", "diego");
    agregarEje(29,"grabar en acustico", "ej3.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(30,"grabar en elctro", "ej8.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(31,"Ritmo distorcionado", "ej9.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(32,"Ritmo en sol negativo", "ej8.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(33,"Ritmo a discrecion menos la nota &", "ej6.png","Eha-
cerlo para el martes","Inicial", "diego");
}
sumaId=33;//id
function agregarEje(elId,elTitulo, laImagen, laDescri, elNivel,elDocente,elRe-
suelto) {
    //if(sumaCaracteres(elTitulo,laDescri)){
    //let valTitulo = elTitulo;
    //let valDescri = laDescri;}
    let valTitulo = elTitulo;
    let valDescri = laDescri;
    if (valTitulo && valDescri) {

        let ejeCarga = new Ejercicio();
        ejeCarga.id = elId;
        ejeCarga.Titulo = valTitulo;
        ejeCarga.Imagen = quitarFakepath(laImagen);
        ejeCarga.Descripcion = laDescri;
        ejeCarga.Nivel = elNivel;
    }
}

```

```

    ejeCarga.Docente = elDocente;
    ejercicios.push(ejeCarga);
  }
}

//-----PRECARGAS RESOLUCIONES-----
function precargaResolucion(){
  agregarResolu(1,"audios/ej9.m4a","", "lopp");//sin

  agregarResolu(1,"audios/ej6.m4a","", "glass");//sin
  agregarResolu(2,"audios/ej4.m4a","", "glass");//sin
  agregarResolu(3,"audios/ej5.m4a","", "glass");//sin

  agregarResolu(9,"audios/ej5.m4a","", "bruno");//sin
  agregarResolu(10,"audios/ej2.m4a","", "bruno");//sin

  agregarResolu(26,"audios/ej7.m4a","", "nego");//sin
  agregarResolu(25,"audios/ej6.m4a","", "nego");//sin

  agregarResolu(9,"audios/ej9.m4a","", "peca");//sin
  agregarResolu(10,"audios/ej8.m4a","", "peca");//sin

  agregarResolu(1,"audios/ej1.m4a","bien", "santi");//con

  agregarResolu(26,"audios/ej3.m4a","", "maria");//sin
  agregarResolu(25,"audios/ej7.m4a","", "maria");//sin

  agregarResolu(1,"audios/ej4.m4a","", "quike");//sin
  agregarResolu(2,"audios/ej3.m4a","", "quike");//sin
  agregarResolu(3,"audios/ej2.m4a","", "quike");//sin
  agregarResolu(4,"audios/ej1.m4a","", "quike");//sin
  agregarResolu(5,"audios/ej8.m4a","", "quike");//sin
  agregarResolu(6,"audios/ej5.m4a","", "quike");//sin
}
function agregarResolu (unId,elAudio, laDevolu, elAlumno ){

  let cargaResolu = new Resolucion();
  cargaResolu.id = unId;
  cargaResolu.audio = elAudio;
  cargaResolu.devolucion = laDevolu;
  cargaResolu.Alumno = elAlumno;
  resoluciones.push(cargaResolu);
}

```

```
// -----LOGUEO USUARIO -----
-----

cambiarSeccion("seccionLogueo");
function Loguearse()
{
    //llamar();
    document.querySelector("#errorLog").innerHTML = "";
    document.querySelector("#mostrarLog").innerHTML = "";
    usu = document.querySelector("#txtUsuario").value;
    contra = document.querySelector("#txtContraseña").value;
    usu = usu.toLowerCase();
    if(logueoUsuario(usuariosD, usu, contra))
    {
        tipUsu = "seccionDocente";
        ListarEjercicios();
        document.querySelector("#mostrarLog").innerHTML = "se a lo-
gueado con éxito!!";
        nombreUsuDoc= buscarUsuDevolver(usuariosD,usu);
        listarAlumnosXdocente(usu);
        cambiarSeccion(tipUsu);
        ocultarLogueoRegistro();
        listaComparativa(usu);
        vaciar();
    }
    else if(logueoUsuario(usuariosA, usu, contra))
    {
        recargarEjerciciosResueltos(usu);
        tipUsu = "seccionAlumno";
        listaEjerciciosDeAlumno(usuariosA, usu);
        document.querySelector("#mostrarLog").innerHTML = "se a lo-
gueado con éxito!!";
        cambiarSeccion(tipUsu);
        ocultarLogueoRegistro();
        listaEjerciciosResueltos();
        vaciar();
    }
    else
    {
        errorLog = "Usuario o Clave incorrectos";
        document.querySelector("#errorLog").innerHTML =errorLog;
        vaciar();
        return-1;
    }
}
}

//----- OCULTAR LOGUEO -----
```

```

function ocultarLogueoRegistro()
{
    logReg = document.querySelector(".logueoRegistro");
    logReg.style.display = "none";
}
//VOLVER INICIO
function volverLogueoRegistro()
{
    document.querySelector(".logueoRegistro").style.display = "block";
    cambiarSeccion("seccionLogueo");
}
//carga docentes disponibles para selection
listarDocentesXnombre();

//-----REGISTRO DOCENTE -----
-----

function RegistroD(){
    nom = document.querySelector("#txtNom").value;
    usu = document.querySelector("#txtUsu").value;
    contra = document.querySelector("#txtCon").value;
    //let buscar = usu;//guardo en otra variable para buscar si ya existe
    //let encon = usuarios.indexOf(buscar);//guardo poscion del usua-
rio en caso de encontrarlo
    ClearD();
    //if (encon > -1)//si encuentro (pos =0 o pos >=1 etc)
    if(buscarUsuario(usuariosD,usu)=== true){
        ClearD();
        errorD = "no se pueden repetir los usuarios";//mensaje al usuario
        mostrarD();
        vaciarRegistroDocente();
        return -1;//no guardo nada espero que usuario vuelva ingresar
    }
    if (textoObligatorioNom(nom) === false){
        ClearD();
        errorD = " debe contener minimo 2 caracteres ";
        mostrarD();
        vaciarRegistroDocente();
        return -1;
    }
    }else if(textoObligatorioUsu(usu) === false){
        ClearD();
        errorD = " debe contener minimo 4 caracteres ";
        mostrarD();
        vaciarRegistroDocente();
        return -1;
    }
}

```

```

    }
    else if (validar_clave(contra)=== false){
        ClearD();
        errorD = "su clave debe tener al menos una mayuscula una minuscula y un nu-
mero";
        mostrarD();
        vaciarRegistroDocente();
        return -1;
    }
    else{
        unUsuarioD = new UsuarioD(nom, usu, contra);
        usuariosD.push(unUsuarioD);
        ClearD();
        muestraD = " se ha registrado con exito!!";
        mostrarD();
        vaciarRegistroDocente();
    }
    listarDocentesXnombre();
}

//-----REGISTRO ALUMNO-----

function listarDocentesXnombre(){//
    document.querySelector("#slcDocente").innerHTML= "";
    for (let n = 0; n < usuariosD.length; n++) {//<option value=""></option>
    document.querySelector("#slcDocente").innerHTML +=
` <option value="${usuariosD[n].usuario}">${usuariosD[n].nombre} "(${usua-
riosD[n].usuario})"</option><br>`
    }
}

function RegistroA(){
    nom = document.querySelector("#txtNomA").value;
    usu = document.querySelector("#txtUsuA").value;
    contra = document.querySelector("#txtConA").value;
    prof = document.querySelector("#slcDocente").value;
    errorA="";
    if(buscarUsuario(usuariosA,usu)){
        ClearA();
        errorA = "no se pueden repetir los usuarios";//mensaje al usuario
        mostrarA();
        vaciarRegistroAlumno();
        return -1;//no guardo nada espero que usuario vuelva a ingresar
    }
    if (textoObligatorioNom(nom) === false){
        ClearA();
        errorA = " debe contener minimo 2 caracteres el nombre";
        mostrarA();
    }
}

```

```

        vaciarRegistroAlumno();
        return -1;

    }else if(textoObligatorioUsu(usu) === false){
        ClearA();
        errorA = " debe contener minimo 2 caracteres el Usuario";
        mostrarA();
        vaciarRegistroAlumno();
        return -1;
    }
    else if (validar_clave(contra)=== false){
        ClearA();
        errorA = "su clave debe tener al menos una mayuscula una minuscula y un numero";
        mostrarA();
        vaciarRegistroAlumno();
        return -1;
    }
    else{
        unUsuarioA = new UsuarioA(nom, usu, contra, prof,"Inicial");
        usuariosA.push(unUsuarioA);
        ClearA();
        vaciarRegistroAlumno();
        muestra= "se ha registrado con exito!!";
        mostrarA();
    }
}

//----- DOCENTE DOCENTE DOCENTE DOCENTE DOCENTE DOCENTE DOCENTE DO-
CENTE DOCENTE DOCENTE DOCENTE -----

//----- PERFIL DOCENTE - CAMBIAR NIVEL ALUMNOS -----
-----
function asignarNivel(){
    let niv = document.querySelector("#txtNivel").value;
    let usuA = document.querySelector("#txtVerAlumnos").value;
    for (let i = 0; i < usuariosA.length; i++) {
        if(usuariosA[i].usuario === usuA){
            let uS = usuariosA[i];
            if((uS.Nivel === "Inicial")&&(niv === "Intermedio") ||
            (uS.Nivel === "Intermedio") && (niv === "Avanzado") ||
            (uS.Nivel === "Inicial")&&(niv === "Avanzado")){
                uS.cambiarNivel(usuA, niv);
                listarAlumnosXdocente(usu);
                listaComparativa(usu);
            }
            else{

```

```

        document.querySelector("#errordeAsignacion").innerHTML = " No se puede cambiar a un nivel menor";
        return -1;
    }
}

function listarAlumnosXdocente(nomUsuDocente){
    document.querySelector("#txtVerAlumnos").innerHTML = "";
    for (let s = 0; s < usuariosA.length; s++) {
        if(usuariosA[s].docent=== nomUsuDocente)
        {
            document.querySelector("#txtVerAlumnos").innerHTML +=
                ` <option value="${usuariosA[s].usuario}">${usuariosA[s].usuario}  + " Nivel: "+usuariosA[s].Nivel}</option><br>`;
        }
    }
}

//----- PERFIL DOCENTE - PLANTEAR (AGREGAR) EJERCICIO-----
-----

function agregarEjercicio(){
    document.querySelector("#agregado").innerHTML = "";
    let elTitulo = document.querySelector("#txtTituloEjer").value;
    let laImagen = document.querySelector("#fileFotoEjer").value;
    let laDescri = document.querySelector("#txtDescripcion").value;
    let resu = false;
    let niv = document.querySelector("#txtNiv").value;
    if(laImagen === ""){
        document.querySelector("#agregado").innerHTML = "El archivo esta vacio"
        return -1;
    }
    else{
        if(sumaCaracteres(elTitulo,laDescri)){
            let cargaEjer = new Ejercicio();
            sumaId++;
            cargaEjer.id = sumaId;
            cargaEjer.Titulo = elTitulo;
            cargaEjer.Imagen = quitarFakepath(laImagen);
            cargaEjer.Descripcion = laDescri;
            cargaEjer.Nivel = niv;
            cargaEjer.Docente=usu;
            cargaEjer.resuelto = resu;
            ejercicios.push(cargaEjer);
            mensaje = "El ejercicio se agregó!!";
        }
        else{

```



```

        mensaje = "Revise titulo y descripcion <br> notas : <br> minimo 20 caracte-
res maximo 200 <br> no pueden ser vacios !!";
    }
}

document.querySelector("#agregado").innerHTML = mensaje;
document.querySelector("#txtTituloEjer").value="";
document.querySelector("#fileFotoEjer").value="";
document.querySelector("#txtDescripcion").value="";
}

//FUNCION PARA VALIDAR TITULO Y DESCRIPCION DE LOS EJERCICIOS-----
-----
function sumaCaracteres(tex1, tex2)//validar ejercicio, titulo y descripcion
{
    //sumados menor que 200 pero mayor de 20 y cada uno DIFERENTE de 0
    if(((tex1.length + tex2.length) <= 200) && ((tex1.length + tex2.len-
gth) >= 20) && (tex1.length !=0) & (tex2.length !=0))
    {
        return true;
    }
    return false;
}

function quitarFakepath(pfoto){//para usar en la interfaz del Docente imagen ejerci-
cio
    let nombreFoto = pfoto.substring(pfoto.lastIndexOf("\\") + 1);
    return nombreFoto
}

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

//-----DOCENTE LISTAR Y REDACTAR-----
-----
//-----DOCENTE LISTAR Y REDACTAR-----
-----

let Aid;
let Reid;
function ListarEjercicios(){//parametro usuario de logueo
    unResolucion=[];
    ejerDelDocente=[];
    document.querySelector("#tdListarEje").innerHTML =``;
    sms ="";
    for (let y = 0; y < ejercicios.length; y++) {
        //recorro los ejercicios
        unE = ejercicios[y];
        //unE.Docente = unE.Docente.toLowerCase();
        if(unE.Docente=== usu )//comparo si encuentro un ejercicio que el do-
cente sea igual el docente logueado
        {ejerDelDocente.push(unE);}
    }
}

```

```

    //unResolucion = arregloSolucionesdeUnEjercicio(resoluciones, unE.id); //guardo el arreglo de resoluciones del ejercicio con el id del mismo
    for (let i = 0; i < ejerDelDocente.length; i++) { //recorro las resoluciones
        //unaRe = unResolucion[i];
        ejD = ejerDelDocente[i];
        unResolucion = arregloSolucionesdeUnEjercicio(resoluciones, ejD.id);
        for (let l = 0; l < unResolucion.length; l++) {
            unRe = unResolucion[l];
            if(unRe.devolucion === "") { //si la devolucion es igual a vacia se listan
                Aid= unRe.Alumno;
                Reid=unRe.id;
                Aid+=Reid;
                document.querySelector("#tdListarEje").innerHTML +=
                    `<tr>
                    <td>${unRe.Alumno}</td>
                    <td>${unRe.id}</td>
                    <td><audio src="${unRe.audio}" controls ></audio></td>
                    <td><textarea type="text" cols="30" rows="5" id="-
                    ${Aid}" class="txtRedactar"></textarea></td>
                    <td><input type="button" value="Enviar" class="btnRedactar" id="${unRe.id}" data-idEjercicio="${unRe.Alumno}"></td>
                    </tr>`;
            }
        }
    }
    let botones = document.querySelectorAll(".btnRedactar"); //genero un boton por cada click
    for (let i = 0; i < botones.length; i++) {
        unBotonHTML = botones[i];
        unBotonHTML.addEventListener("click", Redactar);
    }
}

function Redactar() { //el evento de click activa el guardado de lo que esta redactado
    let elId = this.getAttribute("data-idEjercicio"); //tomo el id de la fila que se leccione
    let ide = this.getAttribute("id");
    let ideN = elId+ide;
    let devo = document.querySelector("#-"+ ideN).value;
    if(textoObligatorioUsu(devo)) {
        ide = Number(ide);
        modificarDevoucion(devo, resoluciones, ide, elId); //capturo la resolucion con el id y nombre de esa fila
        ejerDelDocente = [];
        ListarEjercicios();
        sms = `* Enviado ${elId} <br>`; //guardo en la propiedad devlucion lo que escriba el docente
        er="";
    }
}

```

```

    }
    document.querySelector("#sms").innerHTML = sms;
}

function modificarDevoucion(devol, arregloResoluciones, id, alumno){
    let unRes;
    for (let i = 0; i < arregloResoluciones.length; i++) {
        unRes = arregloResoluciones[i];
        if(unRes.Alumno === alumno && unRes.id === id){
            unRes.devolucion = devol;
            break;
        }
    }
}

// -----
- ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO -----
-

//PERFIL ALUMNO - -----
function listaEjerciciosDeAlumno(usuariosA, usu){//LISTA SUS EJERCICIOS A RESOL-
VER Y ENTREGAR - ALUMNO
    document.querySelector("#tbllistaEjercicios").innerHTML = "";
    let elAlum = buscarUsuarioDevolver(usuariosA, usu);
    for (let i = 0; i < ejercicios.length; i++) {
        ejerciciosAlum = ejercicios[i];
        if(ejerciciosAlum.Docente === elAlum.docent && ejerciciosAlum.Ni-
vel === elAlum.Nivel && idEjerciciosEnviados.indexOf(ejerciciosAlum.id) < 0)
        {
            document.querySelector("#tbllistaEjercicios").innerHTML +=
            `<tr>
            <td>${ejerciciosAlum.id}</td>
            <td>${ejerciciosAlum.Titulo}</td>
            <td>${ejerciciosAlum.Descripcion}</td>
            <td></td>
            <td><input type="file" id="archivoA${ejercicio-
sAlum.id}" class="arch"></td>
            <td><input type="button" value="Enviar" class="btnEn-
viar" id="A${ejerciciosAlum.id}" data-idEjercicioAlumo="${ejerciciosAlum.id}"></td>
            </tr>`;
        }
    }
    let botones = document.querySelectorAll(".btnEnviar");//genero un bo-
ton por cada click

```

```

        for (let i = 0; i < botones.length; i++) {
            const BotonHTML = botones[i];
            BotonHTML.addEventListener("click", Enviar);
        }
        mensaje="";
        document.querySelector("#noEncont").innerHTML = mensaje;
    }
function Enviar(){//el evento de click activa el guardado de lo que esta subido
    document.querySelector("#noEncont").innerHTML = "";
    let elEjeAEnv = Number(this.getAttribute("data-idEjercicioAlumno")); //devuelve el numero de id del ejercicio en el que si hizo CLICK
    let audio = document.querySelector("#archivoA" + elEjeAEnv).value;
    audio = "audios/" + audio.substring(12);
    if(audio === ""){
        mensaje = "* No hay archivo cargado";
    }else{
        let solu = new Resolucion(elEjeAEnv, audio, "",usu );
        for (let i = 0; i < ejercicios.length; i++) {
            let elEjercicioEnviar = ejercicios[i];
            if(elEjeAEnv === elEjercicioEnviar.id){
                idEjerciciosEnviados.push(elEjercicioEnviar.id);
                break;
            }
        }
        //ejerciciosAlum.resuelto = true;
        resoluciones.push(solu);
        mensaje = "* Enviado";
        //document.querySelector("#A" + elEjeAEnv).setAttribute("disabled", "disabled");
        listaEjerciciosResueltos();
        listaEjerciciosDeAlumno(usuariosA, usu);
    }
    document.querySelector("#noEncont").innerHTML = mensaje;
}

//PERFIL ALUMNO - BUSCAR LISTAR-----
-----

function buscarListarEjercicio(){//BUSCAR DENTRO DE LOS EJERCICIOS A RESOLVER Y ENTREGAR
    document.querySelector("#tblEjercicios").innerHTML = "";

    let busqueda = document.querySelector("#txtTextoBusqueda").value;

    let alum = buscarUsuarioDevolver(usuariosA, usu);
    mensajesinohay = "";
    mens="";
    for (let i = 0; i < ejercicios.length; i++) {

```

```

    elEjercicio = ejercicios[i];
    if(elEjercicio.Docente === alum.docent && elEjercicio.Nivel === alum.Ni-
vel && idEjerciciosEnviados.indexOf(elEjercicio.id) < 0){
        if (verificarSubCadena(elEjercicio.Titulo, busqueda))
        {
            document.querySelector("#tblEjercicios").innerHTML +=
            `<tr>
            <td>${elEjercicio.id}</td>
            <td>${elEjercicio.Titulo}</td>
            <td>${elEjercicio.Descripcion}</td>
            <td></td>
            <td><input type="file" id="archivoB${elEjercicio.id}" class="ar-
chivo"></td>
            <td><input type="button" value="Enviar" class="btnEnviarBuscar" data-
idEjercicioAlum="${elEjercicio.id}"></td>
            </tr>`;
        }else if(verificarSubCadena(elEjercicio.Descripcion, busqueda)){
            document.querySelector("#tblEjercicios").innerHTML +=
            `<tr>
            <td>${elEjercicio.id}</td>
            <td>${elEjercicio.Titulo}</td>
            <td>${elEjercicio.Descripcion}</td>
            <td></td>
            <td><input type="file" id="archivoB${elEjercicio.id}" class="ar-
chivo"></td>
            <td><input type="button" value="Enviar" class="btnEnviarBuscar" data-
idEjercicioAlum="${elEjercicio.id}"></td>
            </tr>`;
        }
    }
}

if(idEjerciciosEnviados.length === 0){
    mensajesinohay = "No hay resultados que coincidan con su busqueda";
}

let botones = document.querySelectorAll(".btnEnviarBuscar");//genero un bo-
ton por cada click
for (let i = 0; i < botones.length; i++) {
    const BotonHTML = botones[i];
    BotonHTML.addEventListener("click", EnviarBuscar);
}
document.querySelector("#noEncontrado").innerHTML = mens;
}

function EnviarBuscar(){//el evento de click activa el guar-
dado de lo que esta subido

```

```

    let elEjeAEnvBus = Number(this.getAttribute("data-idEjercicioAlum")); //devuelve el numero de id del ejercicio en el que si hizo CLICK
    //let audio = document.querySelector(".archivo").value;
    let audio = document.querySelector("#archivoB" + elEjeAEnvBus).value;
    audio = "audios/" + audio.substring(12);
    if(audio === ""){
        mens = "* No hay archivo cargado";
    }else{

        let solu = new Resolucion(elEjeAEnvBus, audio, "",usu );

        for (let i = 0; i < ejercicios.length; i++) {
            let elEjercicioEnviar = ejercicios[i];
            if(elEjeAEnvBus === elEjercicioEnviar.id){
                idEjerciciosEnviados.push(elEjercicioEnviar.id);
                break;
            }
        }
        //elEjercicio.resuelto = true;
        resoluciones.push(solu);
        mens = "* Enviado";
        listaEjerciciosResueltos();
        listaEjerciciosDeAlumno(usuariosA, usu);
        buscarListarEjercicio();
    }

document.querySelector("#noEncontrado").innerHTML = mens;
}
//-----

//PERFIL ALUMNO - LISTA LOS EJERCICIOS QUE YA RESOLVIÓ-----

function listaEjerciciosResueltos(){// LISTA LOS EJERCICIOS QUE YA RESOLVIÓ
    document.querySelector("#listaEjercicios").innerHTML = "";
    let elAlum = buscarUsuarioDevolver(usuariosA, usu); //busco usuraio

    for (let i = 0; i < ejercicios.length; i++) {
        const ejerciciosAlu = ejercicios[i]; //comparo quye sean ejercicios del su docente que tenga su nivel y que ya esten resultados
        if((ejerciciosAlu.Docente === elAlum.docent) && (ejerciciosAlu.Nivel === elAlum.Nivel) && (idEjerciciosEnviados.indexOf(ejerciciosAlu.id) > -1))
        {
            document.querySelector("#listaEjercicios").innerHTML +=
                `<tr><td>${ejerciciosAlu.id}</td><td>${ejerciciosAlu.Titulo}</td><td>${ejerciciosAlu.Descripcion}</td>`
        }
    }
}

```

```

        <td></td>
    </tr>`;
    mensajesinohay="";

    }

    }

    if(idEjerciciosEnviados.length === 0){
        mensajesinohay = "No tiene ejercicios Resueltos";
    }

    document.querySelector("#smsSinoHay").innerHTML = mensajesinohay;

}

//-----ESTADISTICAS DOCENTE -----
-----

//----- LISTA DE ALUMNOS CON MAS EJERCICIOS RESUELTOS -----
-----

function listaAlumnosMasEjercicios(){
    alumnoConMasRes=[];
    document.querySelector("#txtlosAlumnos").innerHTML = "";

    losAlumnos = alumnosDocenteLogueado();
    for (let j = 0; j < losAlumnos.length; j++) { //reco-
rro array de los alumnos del docente logueado

        for (let i = 0; i < resoluciones.length; i++) { //reco-
rro el array resoluciones

            if (losAlumnos[j] === resoluciones[i].Alumno){ //COMPARO ELE-
MENTO DE LOS DOS ARRAY
                unAlumno = resoluciones[i].Alumno; //TOMO EL ELEMENTO QUE COINCIDE
                totalRes++
                if(totalRes>=max){ //COMPARA PARA "MAYOR" NO ESTA QUE-
DANDO PARA LOS QUE TIENE IGUAL CANTIDAD DE RESOLUCIONES
                    alumnoConMasRes.push(unAlumno);
                    max=totalRes;
                }
            }
        }
    }

    totalRes =0;

```

```

    }
    a = BorrarRepetidos(alumnoConMasRes);
    for (let k = 0; k < a.length; k++) {
        document.querySelector("#txtlosAlumnos").innerHTML += `<br>* ${a[k]}<br>`;
    }
}
// ----- MOSTRAR CANTIDAD DE EJERCICIOS ENTREGADOS -----
-----
function cantidadTotalDeEjerciciosEntregados(){

    todosLosAlumnosConRes.splice(0);
    xAlumnos.splice(0);
    xAlumnos = alumnosDocenteLogueado();
    for (let j = 0; j < xAlumnos.length; j++) { //recorro array de los alumnos del docente logueado

        for (let i = 0; i < resoluciones.length; i++) { //recorro el array resoluciones

            if (xAlumnos[j] === resoluciones[i].Alumno) { //COMPARO ELEMENTO DE LOS DOS ARRAY
                cadaAlumno = resoluciones[i].Alumno; //TOMO EL ELEMENTO QUE COINCIDE

                todosLosAlumnosConRes.push(cadaAlumno);
            }
        }
    }

    document.querySelector("#txtCantidadEjerciciosRes").innerHTML = `<br>* ${todosLosAlumnosConRes.length}<br>`;

}
//-----LISTA COMPARATIVA POR ALUMNO- TOTAL EJERCICIOS NIVEL - ENTREGADOS -----

function listaComparativa(nomUsuDocente){ //ARMA DESPLEGABLE CON SUS ALUMNOS Y EL NIVEL DE CADA UNO
    document.querySelector("#txtSusAlumnos").innerHTML = "";
    document.querySelector("#totalEjerciciosNivel").innerHTML = "";
    for (let s = 0; s < usuariosA.length; s++) {

        if(usuariosA[s].docent=== nomUsuDocente){
            //misAlumnos.push(usuariosA[s]); //array de sus alumnos
            document.querySelector("#txtSusAlumnos").innerHTML +=
            ` <option value="${usuariosA[s].usuario}">${usuariosA[s].usuario} + " Nivel: "+usuariosA[s].Nivel}</option><br>`;
        }
    }
}

```



```

}
function cantidadEjerciciosNivel(){
  ejerciciosDocenteLogueado=[];
  resXAlumno=0;
  nInicial=0;
  nIntermedio=0;
  nAvanzado=0;
  //VALOR variable unAlumno sale del desplegable
  let unAlumno = document.querySelector("#txtSusAlumnos").value;
  let alumni = buscarUsuarioDevolver(usuariosA, unAlumno);
  let nivelAlumno = alumni.Nivel;

  //SEGUNDA PARTE cantidad de ejercicios del nivel del alumno
  for (let i = 0; i < ejercicios.length; i++) {
    let nivelEje= ejercicios[i].Nivel;
    if(ejercicios[i].Docente === usu){
      if(nivelAlumno === "Inicial" && nivelEje === "Inicial"){
        nInicial++;
        document.querySelector("#totalEjerciciosNivel").innerHTML = `<br>* Cantidad ejercicios Iniciales planteados : ${nInicial}<br>`;
      }else if(nivelAlumno === "Intermedio" && nivelEje === "Intermedio"){
        nIntermedio++;
        document.querySelector("#totalEjerciciosNivel").inner-
HTML = `<br>* Cantidad ejercicios Intermedios planteados : ${nIntermedio}<br>`;
      }
      else if(nivelAlumno === "Avanzado" && nivelEje === "Avanzado"){
        nAvanzado++;
        document.querySelector("#totalEjerciciosNivel").inner-
HTML = `<br>* Cantidad ejercicios Avanzados planteados : ${nAvanzado}<br>`;
      }
    }
  }
  if(nInicial === 0 && nIntermedio === 0 && nAvanzado === 0){
    document.querySelector("#totalEjerciciosNivel").innerHTML = `* No hay ejercicios planteados para nivel : ${nivelAlumno}`;
  }
  for (let j = 0; j < resoluciones.length; j++) {
    let cAlumno = resoluciones[j].Alumno;

    if(cAlumno === unAlumno ){
      resXAlumno++
      document.querySelector("#totalResueltosAlumno").inner-
HTML = `<br>* El alumno ${unAlumno} resolvió ${resXAlumno} ejercicios <br>`;
    }
  }
}
//-----
-----

```

```

//ESTADISTICAS ALUMNO ESTADISTICAS ALUMNO ESTADISTICAS ALUMNO ESTADISTI-
CAS ALUMNO ESTADISTICAS ALUMNO

function porcentaje(){// porcentaje de ejercicios que ha resueltos segun su ni-
vel planteados por su docente
  for (let i = 0; i < resoluciones.length; i++) {
    //let pResol = resoluciones[i].Alumno;
    if(resoluciones[i].Alumno === usu) {
      totalResoluciones++//cuento las resouciones
    }
  }

  for (let r = 0; r < usuariosA.length; r++) {
    if (usuariosA[r].usuario === usu){
      nivelUsu = usuariosA[r].Nivel;
      doc = usuariosA[r].docent;
      break;
    }
  }
  for (let z = 0; z < ejercicios.length; z++) {

    if(ejercicios[z].Nivel === nivelUsu && ejercicios[z].Docente === doc )
      totalEjerciciosNivel++

  }

  //totalEjerciciosNivel= BorrarRepetidos(totalEjerciciosNivel);

  let porcentaje = Math.floor((totalResoluciones * 100) / totalEjerciciosNivel);

  //Math redondeo para abajo
  document.querySelector("#porcentaje").innerHTML = `<br>* El alumno ${usu} resol-
vi  ${porcentaje} % de los ejercicios <br>`;
}

function cantidadEjerConDevolucion(){// Ejercicios resueltos CON devolucion -- Ejer-
cicios resueltos SIN devolucion.
  document.querySelector("#pSinDevolucion").innerHTML = "";
  document.querySelector("#pConDevolucion").innerHTML = "";
  let cantConDevolucion=0;
  let cantSinDevolucion=0;
  let esa = "";
  for (let i = 0; i < resoluciones.length; i++) {
    //let pResol = resoluciones[i].Alumno;
    if(resoluciones[i].Alumno === usu) {
      if(resoluciones[i].devolucion !== ""){
        cantConDevolucion ++;
      }
    }
  }

```

```

    }
    if(resoluciones[i].devolucion === ""){
        cantSinDevolucion ++;
        esa += resoluciones[i].id + ",";
        document.querySelector("#pSinDevolucion").inner-
HTML = `* ${usu} no ha recibido devolucion en ${cantSinDevolucion} ejerci-
cios <br> * los cuales sus identificadores son: ${esa}<br>`;
    }

}

}

    if (cantConDevolucion === 0){
        document.querySelector("#pConDevolucion").inner-
HTML = `* ${usu} ha recibido ${cantConDevolucion} devoluciones `;
    }

}

function recargarEjerciciosResueltos(usu){
    idEjerciciosEnviados=[];
    for (let s = 0; s < resoluciones.length; s++) {
        resolution = resoluciones[s]; //recorro resouicione spara ver cua-
les son de ese alumno
        if(resolution.Alumno === usu){
            //resoDeAlumno.push(resolution.id); //guardo las que son de el
            idEjerciciosEnviados.push(resolution.id);
        }
    }
}

//-----
function vaciar(){
    document.querySelector("#txtUsuario").value="";
    document.querySelector("#txtContrasenia").value="";
}

function vaciarRegistroAlumno(){
    document.querySelector("#txtNomA").value="";
    document.querySelector("#txtUsuA").value="";
    document.querySelector("#txtConA").value="";
}

function vaciarRegistroDocente(){
    document.querySelector("#txtNom").value="";
    document.querySelector("#txtUsu").value="";
    document.querySelector("#txtCon").value="";
}

```

LIBRERÍA

```
//BUSQUEDAS
function Buscar(busqueda, arreglo){
    let encuentre =false;
    if(arreglo.indexOf(busqueda) > -1){
        encuentre =true;
    }
    return encuentre;
}
function BorrarRepetidos(a)
{
    let newArr = [];
    for(let i=0;i<a.length;i++)
    {
        let elem= a[i];
        if(a.indexOf(elem)=== i)
        {
            //Primera impresión
            newArr.push(elem);
        }
    }
    return newArr;
}
function buscarUsuario(usuarios, usu)
{
    for(let i = 0; i < usuarios.length; i++){
        let unUS = usuarios[i];

        if(unUS.usuario === usu ){
            return true;
        }
    }
    return false;
}
//busca y devuelve un usuario
function buscarUsuDevolver(usuarios, usu)
{
    let unUS;
    let retorno;
    for(let i = 0; i < usuarios.length; i++){
        unUS = usuarios[i];

        if(unUS.usuario === usu ){
            retorno = unUS.usuario;
        }
    }
}
```

```

        break;
    }
}
return retorno;
}
function buscarUsuarioDevolver(usuarios, usu)
{
    let unUS;
    let retorno;
    usu=usu.toLowerCase();
    for(let i = 0; i < usuarios.length; i++){
        unUS = usuarios[i];
        unUS.usuario=unUS.usuario.toLowerCase();
        if(unUS.usuario === usu ){
            retorno = unUS;
            break;
        }
    }
    return retorno;
}
//funcion que sirve para buscar ejercicios y resoluciones por su arreglo y su identificador
function buscarUsuDevolverEjercicio(ejercicios, id)
{
    let unEj;
    let retorno;
    for(let i = 0; i < ejercicios.length; i++){
        unEj = ejercicios[i];

        if(unEj.id === id ){
            retorno = unEj;
            break;
        }
    }
    return retorno;
}
function buscarPorId(arreglo, id,usuarioAlumno){//esta funcion toma el arreglo de resoluciones y busca la resolucio
n que tenga el nombre y id puestos
    let unArreglo;//crea un arreglo
    for (let i = 0; i < arreglo.length; i++) {
        //recorre
        unArreglo = arreglo[i];
        //guarda
        if(unArreglo.id === id && unArreglo.Alumno === usuarioAlumno){
            //compara id y nombre de alumno
            return unArreglo;
            //retorna cuando encuentre
        }
    }
}
//buscar resoluciones por el id y el usuario del alumno
//VERIFICACIONES
function verificarSubCadena(texto, subcadena)

```

```

{
  texto = texto.toLowerCase();
  subcadena = subcadena.toLowerCase();
  if(texto.indexOf(subcadena) === -1){
    return false;
  }
  return true;
}
function PrimeraMayusTodasMinus(tx){
  txNuevo = "";
  tx = tx.toLowerCase();
  let letra = tx.charAt(0);
  let may= letra.toUpperCase();
  for (let i = 1; i < tx.length;i++){
    txNuevo += tx.charAt(i);
    txNuevo.toLowerCase();
  }
  txNuevo = may + txNuevo;
  return txNuevo;
}
function textoObligatorioUsu(tex)
{
  let esValido = false;
  //largo es igual a o o menor a 4 retorna falso
  if(tex.length >= 2)
  {
    esValido = true;
  }
  return esValido;
}
function textoObligatorioNom(tex)
{
  let esValido = false;
  //largo es mayor o igualretorna true
  if(tex.length >= 2)
  {
    esValido = true;
  }
  return esValido;
}
function validar_clave(contrasenia){
  if(contrasenia.length >= 4)//largo de la clave minimo 4
  {
    let mayuscula = false;
    let minuscula = false;
    let numero = false;

    for(let i = 0;i<contrasenia.length;i++)//recorro la misma

```

```

    {
        if(contrasenia.charCodeAt(i) >= 65 && contrasenia.charCodeAt(i) <= 90)
        {
            mayuscula = true;//uso los charcode para saber si tiene mayusculas
        }
        else if(contrasenia.charCodeAt(i) >= 97 && contrasenia.charCo-
deAt(i) <= 122)
        {
            minuscula = true;//uso los charcode para saber si tiene minusculas
        }
        else if(contrasenia.charCodeAt(i) >= 48 && contrasenia.charCo-
deAt(i) <= 57)
        {
            numero = true;////uso los charcode para saber si tiene numeros
        }
    }
    if(mayuscula == true && minuscula == true && numero == true)//si tiene todo
    {
        return true;//es verdadero
    }
}
return false;//si no falso
}
//LOGUEO
function logueoUsuario(usuarios, usu, contr){
    for(let i = 0; i < usuarios.length; i++){
        let unUS = usuarios[i];
        unUS.usuario = unUS.usuario.toLowerCase();
        usu = usu.toLowerCase();
        if(unUS.usuario === usu && unUS.contrasenia === contr){
            return true;
        }
    }
    return false;
}
function alumnosDocenteLogueado(){//ARMA LISTA DE ALUMNOS DEL DOCENTE LOGUEADO (su-
susAlumnos) *esta funcion solo sirve para eso
    susAlumnos=[];
    for (let s = 0; s < usuariosA.length; s++) {
        let pAlumnos = usuariosA[s];
        if(pAlumnos.docent=== usu) {
            susAlumnos.push(pAlumnos.usuario);//los alumnos del docente logueado
        }
    }
    return susAlumnos;
}
//ARREGLOS
function arregloSolucionesdeUnEjercicio(arreglodesoluciones, id)

```

```

{
  nuevoAr=[];
  for (let i = 0; i < arreglodesoluciones.length; i++) {
    const unS = arreglodesoluciones[i];
    if(unS.id === id){
      nuevoAr.push(unS);
    }
  }
  return nuevoAr;
}
//LIMPIAR Y MOSTRAR REGISTRO
function mostrarA(){
  document.querySelector("#errorA").innerHTML = errorA;
  document.querySelector("#mostrarA").innerHTML = muestra;
}
function ClearA(){
  muestra="";
  errorA="";
  document.querySelector("#errorA").innerHTML = errorA;
  document.querySelector("#mostrarA").innerHTML = muestra;
}
function mostrarD(){
  document.querySelector("#errorD").innerHTML = errorD;
  document.querySelector("#mostrarD").innerHTML = muestraD;
}
function ClearD(){
  errorD="";
  muestraD="";
  document.querySelector("#errorD").innerHTML = errorD;
  document.querySelector("#mostrarD").innerHTML = muestraD;
}

function ejerciciosEntregados(docente){
  let ejerciciosDeElDocente=[];
  let ejerciciosRes=[];
  let un;
  for (let i = 0; i < ejercicios.length; i++) {
    if(docente === ejercicios[i].Docente){
      ejerciciosDeElDocente.push(ejercicios[i]);
    }
  }
  for (let x = 0; x < resoluciones.length; x++) {
    un = resoluciones[x];
    for(let y =0; y< ejerciciosDeElDocente.length; y++){
      if(un.id === ejerciciosDeElDocente[y].id){
        ejerciciosRes.push(ejerciciosDeElDocente[y]);
      }
    }
  }
}

```



```

    }
    ejerciciosRes = BorrarRepetidos(ejerciciosRes)
    return ejerciciosRes;
}
// ejerciciosDocenteLogueado = ejerciciosEntregados(usu);

```

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">
    <title>Document</title>
</head>
<body>
    <!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" href="css/estilos.css">
    <title>OBLIGATORIO GRUPO 7</title>
</head>
<body>

    <div id="contenido">
        <input type="button" value="Volver Logueo/Registro " id="btnVolver">
        <!--LOGUEO -->
        <div class="logueoRegistro" >

            <!--REGISTRO AMBOS -->
            <!--REGISTRO DOCENTE -->

            <div id="seccionRegistro">
                <div id="seccionRegistroD">
                    <h5>REGISTRO DOCENTE</h5>
                    <p>Tipo Usuario- Docente :</p>
                    <label for="txtNom">Nombre :</label>
                    <input type="text" id="txtNom"><br><br>

                    <label for="txtUsu">Usuario:</label>

```

```

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

        <label for="txtCon" >Contraseña:</label>
        <input type="password" name="" id="txtCon"><br><br>

        <input type="button" value="Registrarse" id="btnRegistroD"><br>
        <p id="errorD"></p>
        <p id="mostrarD"></p>
    </div>

    <!--REGISTRO ALUMNO-->
    <div id="seccionRegistroA">
        <h5>REGISTRO ALUMNO</h5>
        <label for="txtNomA">Nombre :</label>
        <input type="text" id="txtNomA"><br>

        <label for="txtUsuA">Usuario:</label>
        <input type="text" name="" id="txtUsuA"><br>

        <label for="txtConA" >Contraseña:</label>
        <input type="password" name="" id="txtConA"><br>

        <label for="slcDocente">Docente: </label>
        <select id="slcDocente">
        </select>
        <br><br>

        <input type="button" value="Registrarse" id="btnRegistroA">
        <p id="errorA"></p>
        <p id="mostrarA"></p>
    </div>
</div>
<div id="seccionLogueo">
    <h5> LOGUEO USUARIO:</h5><br>
    <label for="txtUsuario"> Usuario:</label>
    <input type="text" name="" id="txtUsuario"><br>
    <label for="txtContraseña">Clave:</label>
    <input type="password" id="txtContraseña"><br>
    <input type="button" value="Entrar" id="btnEntrar">
    <p id="mostrarLog"></p>
    <p id="errorLog"></p><br>
    <p id="pParrafo"></p>
</div>
</div>
    <!--DOCENTE DOCENTE DOCENTE DOCENTE DOCENTE DOCENTE DOCENTE DOCENTE -->
<!-- CONTENEDOR TODO DOCENTE-->
    <!--PERFIL DOCENTE - GESTION DE ALUMNOS-->
    <div class="seccion" id="seccionDocente">

```

```

<div>
  <br><br>
  <h5>CAMBIAR NIVEL ALUMNO</h5>
  <label for="txtVerAlumnos">Alumnos : </label>
  <select id="txtVerAlumnos">
  </select><br>
  <label for="txtNivel">Asignar Nivel:</label>
  <select name="" id="txtNivel">
    <option value="Intermedio">Intermedio</option>
    <option value="Avanzado">Avanzado</option>
  </select> <br><br>
  <input type="button" id="btnNivel" value="Cambiar"><br>
  <p id="errordeAsignacion"></p><br>
  <br>
  <hr><hr>
</div>

<!--PERFIL DOCENTES - PLANTEAR EJERCICIO-->
<div>
  <h5>PLANTEAR EJERCICIO</h5>
  <label for ="txtTituloEjer"> Titulo del ejercicio</label><br>
  <input type ="text" id="txtTituloEjer"> <br> <br>

  <label for="fileFotoEjer">Foto Ejercicio:</label> <!-- FOTO -->
  <input type="file" id="fileFotoEjer"><br> <br>

  <label for ="txtDescripcion">Descripcion</label> <br>
  <textarea id="txtDescripcion" cols="30" rows="5"></textarea><br> <!--
- AREA PARA TEXTO -->

  <label for="txtNiv">Nivel :</label>
  <select id="txtNiv">
    <option value="Inicial">Inicial</option>
    <option value="Intermedio">Intermedio</option>
    <option value="Avanzado">Avanzado</option>
  </select><br><br>

  <input type ="button" id="btnAgregarEjer" value ="Agregar"><br>

  <p id="agregado"></p>
  <hr><hr>
</div>

<!--PERFIL DOCENETE - REDACTAR EJERCICIO-->
<div>
  <h5>REDACTAR EJERCICIO</h5>
  <table border="1" >
    <thead>

```

```

        <tr>
            <th>Alumno</th>
            <th>ID de Ejercicio</th>
            <th>Solucion</th>
            <th>Redactar</th>
            <th>Enviar Devolucion</th>
        </tr>
    </thead>
    <tbody id="tdListarEje"></tbody>
</table>
<p id="sms"></p>
<hr><hr>
</div>
<!-------ESTADISTICAS DOCENTE----->
----->
<h2>Estadisticas</h2>

<h5>Alumnos con mas Ejercicios Resueltos</h5>

<label for ="txtunAlumno"> Ver lista de alumnos con mas Ejerci-
cios Resueltos</label><br>

<input type ="button" id="btnBuscaAlumno" value ="Listar"><br>
<p id="txtlosAlumnos"></p>

<h5>Cantidad total de ejercicios resueltos</h5>

<label for ="txtunAlumno"> Ver cantidad total de ejercicios resuel-
tos</label><br>

<input type ="button" id="btnBuscaCantidad" value ="Listar"><br>
<p id="txtCantidadEjerciciosRes"></p>

<h5>Total Ejercicios por Nivel y resueltos por el alumno</h5>
<label for="txtSusAlumnos">Alumnos : </label>
<select id="txtSusAlumnos">
</select><br>

<input type="button" id="btnListarEstadisticas" value="Listar"><br>
<p id="totalEjerciciosNivel"></p>
<p id="totalResueltosAlumno"></p>

</div>      <!-- fIIIIINNNNNAAALLL   CONTENEDOR TODO DOCENTE-->

<!-- ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO ALUMNO-->

    <div class="seccion" id="seccionAlumno" ><!-- CONTENE-
DOR TODO ALUMNO-->

```

```

<!--PERFIL ALUMNO - BUSCAR EJERCICIO-->
<div>

    <h5>LISTA DE SUS EJERCICIOS A ENTREGAR</h5>

    <!--lista de ejercicios qe aun no entrego-->
>

    <table border="1">
        <thead>
            <tr>
                <th>Id de Ejercicio</th>
                <th>Titulo</th>
                <th>Descripcion</th>
                <th>Foto</th>
                <th>Subir Entrega</th>
                <th>Enviar</th>

            </tr>
        </thead>
        <tbody id="tbllistaEjercicios"></tbody>
    </table>
    <br>
    <p id="noEncont"></p><br>

</div>
<div>
    <h5>BUSCAR EJERCICIO</h5>

    <label for='txtTextoBusqueda'>Buscar</label>
    <input type='text' id='txtTextoBusqueda'><br><br>
    <input type="button" id="btnBuscarEjer" value="Buscar">
    <h3>Ejercicios</h3>

    <table border="1">
        <thead>
            <tr>
                <th>Id de Ejercicio</th>
                <th>Titulo</th>
                <th>Descripcion</th>
                <th>Foto</th>
                <th>Subir Entrega</th>
                <th>Enviar</th>

            </tr>
        </thead>
        <tbody id="tblEjercicios"></tbody>
    </table>
    <br>

```

```

        <p id="noEncontrado"></p><br>

    </div>
    <hr><hr>
    <div>
        <h5>LISTA DE SUS EJERCICIOS RESUELTOS</h5>
        <table border="1">
            <thead>
                <tr>
                    <th>Id de Ejercicio</th>
                    <th>Titulo</th>
                    <th>Descripcion</th>
                    <th>Foto</th>
                </tr>
            </thead>
            <tbody id="listaEjercicios"></tbody>
        </table>
        <br>
        <p id="smsSinoHay"></p>
    </div>
    <!-------ESTADISTICAS ALUMNO----->
    ----->

    <h2>Estadisticas</h2>

    <label for='txtEstadistica1'>Porcentaje ejercicios resueltos:</label>

    <input type="button" id="btnEstadisticaUno" value="Porcentaje">
    <p id="porcentaje"></p><br>

    <label for='txtEstadistica2'>Ejercicios con devolucion:</label>

    <input type="button" id="btnEstadisticaDos" value="ConDevolucion">
    <p id="pConDevolucion"></p>
    <br><p id="pSinDevolucion"></p>
</div>
</div>
    <!-- fIIIIINNNNNAAALLL   CONTENEDOR TODO ALUMNO-->
    <script src="js.js/libreria.js"></script>
    <script src="js.js/codigo.js"></script>
</body>
</html>

```

PRECARGA

```
//-----PRECARGAS -----
//-----PRECARGAS      DOCENTES-----

function precargaDocente() {
  agregarDocente("Carlos", "carlos", "Ab123");
  agregarDocente("Juan", "juan", "Ab123");
  agregarDocente("Diego", "diego", "Ab123");
  agregarDocente("Santiago", "santiago", "Ab123");
}
function agregarDocente(elNombre, elUsuario, laContrasenia) {
  let valNombre = textoObligatorioNom(elNombre);
  let valUsuario = textoObligatorioUsu(elUsuario);
  let valContrasenia = validar_clave(laContrasenia);
  if (valNombre && valUsuario && valContrasenia) {
    let docenteCarga = new UsuarioD();
    docenteCarga.nombre = elNombre;
    docenteCarga.usuario = elUsuario;
    docenteCarga.contrasenia = laContrasenia;
    usuariosD.push(docenteCarga);
  }
}
//-----PRECARGAS      ALUMNOS-----

function precargaAlumno() {
  agregarAlumno("Quike", "quike", "Ab123", "carlos","Inicial");
  agregarAlumno("Jorge", "Jorje", "Ab123", "juan","Inicial");
  agregarAlumno("Nego", "nego", "Ab123", "diego","Inicial");
  agregarAlumno("maria", "maria", "Ab123", "diego","Inicial");
  agregarAlumno("dario", "dario", "Ab123", "diego","Inicial");
  agregarAlumno("Peca", "peca", "Ab123", "juan","Inicial");
  agregarAlumno("bruno", "bruno", "Ab123", "juan","Inicial");
  agregarAlumno("santi", "santi", "Ab123", "carlos","Inicial");
  agregarAlumno("glass", "glass", "Ab123", "carlos","Inicial");
  agregarAlumno("lopp", "lopp", "Ab123", "carlos","Inicial");
}
function agregarAlumno(elNombre, elUsuario, laContrasenia, elDocente, elNivel) {
  let valNombre = textoObligatorioNom(elNombre);
  let valUsuario = textoObligatorioUsu(elUsuario);
  let valContrasenia = validar_clave(laContrasenia);
  if (valNombre && valUsuario && valContrasenia) {
    let alumnoCarga = new UsuarioA();
    alumnoCarga.nombre = elNombre;
```

```

    alumnoCarga.usuario = elUsuario;
    alumnoCarga.contrasenia = laContrasenia;
    alumnoCarga.docent = elDocente;
    alumnoCarga.Nivel = elNivel;
    usuariosA.push(alumnoCarga);
  }
}

//-----PRECARGAS  -- EJERCICIOS-----
-----

function precargaEje() {
  //carlos
  agregarEje(1,"Ejecutar en Do menor", "ej1.png","El ejercicio debe contemplar opciones","Inicial", "carlos");//CAMBIÉ A INICIAL!!!!
  agregarEje(2,"Establecer ritmos de ocho en ocho", "ej2.png","Se debe ejecutar solo indicado","Inicial", "carlos");
  agregarEje(3,"Ritmo latino", "ej3.png","Puede hacer variantes","Inicial", "carlos");
  agregarEje(4,"Introduccion a ritmos", "ej4.png","Se deben realizar variantes","Inicial", "carlos");
  agregarEje(5,"Primeras notas", "ej5.png","La ejecucion es libre","Inicial", "carlos");
  agregarEje(6,"segundas notas", "ej6.png","La ejecucion es libre","Inicial", "carlos");
  agregarEje(7,"terceras notas", "ej7.png","La ejecucion es libre","Avanzado", "carlos");
  agregarEje(8,"cuartas notas", "ej8.png","La ejecucion es libre","Avanzado", "carlos");
  //juan
  agregarEje(9,"Introduccion a ritmos", "ej7.png","Se deben realizar variantes","Inicial", "juan");
  agregarEje(10,"Ritmo dominicano", "ej8.png","Ejercicio de calecita libre","Inicial", "juan");
  agregarEje(11,"Ritmo colombiano", "ej1.png","hacerlo para el lunes sin falta ","Intermedio", "juan");
  agregarEje(12,"Ritmo africano", "ej2.png","Ejercicio de ejecucion libre","Avanzado", "juan");
  agregarEje(13,"Ritmo paraguayo", "ej3.png","Ejercicio de ejecucion limpia ","Avanzado", "juan");
  agregarEje(14,"Ritmo uruguayo", "ej4.png","Ejercicio de ejecucion cerrada en sol","Inicial", "juan");
  agregarEje(15,"Ritmo brasilero", "ej5.png","Ejercicio de ejecucion cerrada en do","Inicial", "juan");
  agregarEje(16,"Ritmo argentino", "ej6.png","Ejercicio de ejecucion libre","Inicial", "juan");
  //santiago

```



```

    agregarEje(17,"Ritmo electronico", "ej5.png","tocar y rasguear","Ini-
cial", "santiago");
    agregarEje(18,"ejercicio complemetario", "ej5.png","to-
car solo con una mano","Inicial", "santiago");
    agregarEje(19,"parcial numero 1", "ej9.png","tocar con sol,do y re","Ini-
cial", "santiago");
    agregarEje(20,"ejrcicio 4", "ej1.png","tocar con sol,do y re","Inicial", "san-
tiago");
    agregarEje(21,"ejercicio 2do semestre", "ej2.png","tocar con sol,do y re","Ini-
cial", "santiago");
    agregarEje(22,"ejercicio 3", "ej3.png","tocar con sol,do y re","Interme-
dio", "santiago");
    agregarEje(23,"ejercicio 4", "ej4.png","tocar con sol,do y re","Interme-
dio", "santiago");
    agregarEje(24,"ejercicio 5", "ej5.png","tocar con sol,do y re","Interme-
dio", "santiago");
    //diego
    agregarEje(25,"ejercicio 3 en do mayor", "ej5.png","tocar con notas negati-
vas","Inicial", "diego");//entregado
    agregarEje(26,"Ritmo del agua", "ej4.png","grabarlo y enviarlo en un zip","Ini-
cial", "diego");//entregado
    agregarEje(27,"cantar y tocar", "ej1.png","este ejerccio lleva 10 puntos","In-
termedio", "diego");
    agregarEje(28,"grabar con pedalera distorcionada", "ej2.png","el mierco-
les 23 pronto","Intermedio", "diego");
    agregarEje(29,"grabar en acustico", "ej3.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(30,"grabar en elctro", "ej8.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(31,"Ritmo distorcionado", "ej9.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(32,"Ritmo en sol negativo", "ej8.png","Ehacerlo para el martes","Ini-
cial", "diego");
    agregarEje(33,"Ritmo a discrecion menos la nota &", "ej6.png","Eha-
cerlo para el martes","Inicial", "diego");
}
sumaId=33;//id
function agregarEje(elId,elTitulo, laImagen, laDescri, elNivel,elDocente,elRe-
suelto) {
    //if(sumaCaracteres(elTitulo,laDescri)){
    //let valTitulo = elTitulo;
    //let valDescri = laDescri;}
    let valTitulo = elTitulo;
    let valDescri = laDescri;
    if (valTitulo && valDescri) {

        let ejeCarga = new Ejercicio();
        ejeCarga.id = elId;

```

```

    ejeCarga.Titulo = valTitulo;
    ejeCarga.Imagen = quitarFakepath(laImagen);
    ejeCarga.Descripcion = laDescri;
    ejeCarga.Nivel = elNivel;
    ejeCarga.Docente = elDocente;
    ejercicios.push(ejeCarga);
  }
}

//-----PRECARGAS RESOLUCIONES-----
function precargaResolucion(){
  agregarResolu(1,"audios/ej9.m4a","", "lopp");//sin

  agregarResolu(1,"audios/ej6.m4a","", "glass");//sin
  agregarResolu(2,"audios/ej4.m4a","", "glass");//sin
  agregarResolu(3,"audios/ej5.m4a","", "glass");//sin

  agregarResolu(9,"audios/ej5.m4a","", "bruno");//sin
  agregarResolu(10,"audios/ej2.m4a","", "bruno");//sin

  agregarResolu(26,"audios/ej7.m4a","", "nego");//sin
  agregarResolu(25,"audios/ej6.m4a","", "nego");//sin

  agregarResolu(9,"audios/ej9.m4a","", "peca");//sin
  agregarResolu(10,"audios/ej8.m4a","", "peca");//sin

  agregarResolu(1,"audios/ej1.m4a", "bien", "santi");//con

  agregarResolu(26,"audios/ej3.m4a","", "maria");//sin
  agregarResolu(25,"audios/ej7.m4a","", "maria");//sin

  agregarResolu(1,"audios/ej4.m4a","", "quike");//sin
  agregarResolu(2,"audios/ej3.m4a","", "quike");//sin
  agregarResolu(3,"audios/ej2.m4a","", "quike");//sin
  agregarResolu(4,"audios/ej1.m4a","", "quike");//sin
  agregarResolu(5,"audios/ej8.m4a","", "quike");//sin
  agregarResolu(6,"audios/ej5.m4a","", "quike");//sin
}
function agregarResolu (unId,elAudio, laDevolu, elAlumno ){

  let cargaResolu = new Resolucion();
  cargaResolu.id = unId;
  cargaResolu.audio = elAudio;
  cargaResolu.devolucion = laDevolu;
  cargaResolu.Alumno = elAlumno;

```

```
    resoluciones.push(cargaResolu);  
}
```