**Evaluación Individual Final: Gestión de Datos Académicos**

Cada estudiante de forma individual, debe resolver 1 de los siguientes 4 ejercicios relacionados con un sistema de datos académico y debe estar desplegado en en servidor web Apache Tomcat, cada ejercicio simula un sistema por separado.

Tiempo total de la evaluación: 2 horas.

**Ejercicio 1: Gestión de Materias (crear, buscar)**

**Objetivo**: Manejar un conjunto de materias usando objetos y arreglos en JavaScript, y mostrar los resultados en HTML.

Instrucciones:

1. Crear una carpeta llamada GestionMaterias\_[Nombre\_Alumno], y dentro de ella, generar los archivos gestionMaterias\_[Nombre\_Alumno].html y gestionMaterias\_[Nombre\_Alumno].js.

En el archivo js crear un arreglo de **materias**, inicialmente sin elementos.

Los valores de materia(txtNombreMateria, txtCodigo, txtDescripcion)

1. Implementa las siguientes funcionalidades en JavaScript:

**crearMateria**

Parámetros: no recibe parámetros

Crear un objeto materia tomando los valores de las cajas de la página.

Retorno:

Retorna el objeto creado

**agregarMateria**

Parámetros:

Invoca a la función crearMateria

Agrega el objeto materia, al arreglo de materias

Imprimir el arreglo en consola

Retorna:

No retorna nada

PÁGINA

Agregar tres inputs con los tres campos solicitados.

Agregar un botón GUARDAR en la página html, este botón invoca a agregarMateria.

**refrescarMaterias**

no recibe parámetros

Muestra todas las materias del arreglo en una tabla de la página html

no retorna nada

**PÁGINA HTML**

Agregar un botón REFRESCAR que invoca a refrescarMaterias

Se debe mostrar las materias al presionar el botón REFRESCAR (evento onload)

**buscarMateria**

Parámetros: recibe el Identificador único de la materia

Toma el valor de la caja txtBuscar

Busca la materia en el arreglo materias.

Retorno:

En un Alert “Materia encontrada”, en caso de no encontrar, retorna “Materia no encontrada”

**PÁGINA HTML**

Agregar un input para ingresar el identificador único de la materia

Agregar un botón BUSCAR que invoca a **buscarMateria**

Usa un formulario en HTML para interactuar con las funciones (botones para agregar, y buscar). Agregar estilos y diseño personalizado (uso de imágenes)

**Evaluación Individual Final: Gestión de Datos Académicos**

Cada estudiante de forma individual, debe resolver 1 de los siguientes 4 ejercicios relacionados con un sistema de datos académico y debe estar desplegado en en servidor web Apache Tomcat, cada ejercicio simula un sistema por separado.

Tiempo total de la evaluación: 2 horas.

**Ejercicio 1: Gestión de Estudiante (crear, buscar)**

**Objetivo**: Manejar un conjunto de estudiantes usando objetos y arreglos en JavaScript, y mostrar los resultados en HTML.

Instrucciones:

1. Crear una carpeta llamada GestionEstudiante\_[Nombre\_Alumno], y dentro de ella, generar los archivos gestionEstudiante\_[Nombre\_Alumno].html y gestionEstudiante\_[Nombre\_Alumno].js.

En el archivo js crear un arreglo de **estudiantes**, inicialmente sin elementos.

Los valores de estudiante: (txtNombre, txtCurso, txtMatricula)

1. Implementa las siguientes funcionalidades en JavaScript:

**crearEstudiante**

Parámetros: no recibe parámetros

Crear un objeto estudiante tomando los valores de las cajas de la página.

Retorno:

Retorna el objeto creado

**agregarEstudiante**

Parámetros:

Invoca a la función **crearEstudiante**

Agrega el objeto **estudiante**, al arreglo de estudiantes

Imprimir el arreglo en consola

Retorna:

No retorna nada

**PÁGINA**

Agregar tres inputs con los tres campos solicitados.

Agregar un botón GUARDAR en la página html, este botón invoca a **agregarEstudiante**.

**refrescarEstudiante**

no recibe parámetros

Muestra todos los estudiantes del arreglo en una tabla de la página html

no retorna nada

**PÁGINA HTML**

Agregar un botón REFRESCAR que invoca a **refrescarEstudiante**

Se debe mostrar los estudiantes al presionar el botón REFRESCAR (evento onload)

**buscarEstudiante**

Parámetros: recibe el Identificador único del estudiante

Toma el valor de la caja txtBuscar

Busca el estudiante en el arreglo.

Retorno:

En un Alert “Estudiante encontrado”, en caso de no encontrar, retorna “Estudiante no encontrado”

**PÁGINA HTML**

Agregar un input para ingresar el identificador único del estudiante

Agregar un botón BUSCAR que invoca a **buscarEstudiante**

Usa un formulario en HTML para interactuar con las funciones (botones para agregar, y buscar). Agregar estilos y diseño personalizado (uso de imágenes)

**Evaluación Individual Final: Gestión de Datos Académicos**

Cada estudiante de forma individual, debe resolver 1 de los siguientes 4 ejercicios relacionados con un sistema de datos académico y debe estar desplegado en en servidor web Apache Tomcat, cada ejercicio simula un sistema por separado.

Tiempo total de la evaluación: 2 horas.

**Ejercicio 1: Gestión de Profesor (crear, buscar)**

**Objetivo**: Manejar un conjunto de profesores usando objetos y arreglos en JavaScript, y mostrar los resultados en HTML.

Instrucciones:

1. Crear una carpeta llamada GestionProfesores\_[Nombre\_Alumno], y dentro de ella, generar los archivos gestionProfesores\_[Nombre\_Alumno].html y gestionProfesores\_[Nombre\_Alumno].js.

En el archivo js crear un arreglo de **profesores**, inicialmente sin elementos.

Los valores de profesor( txtNombre, txtEdad, txtEmail)

1. Implementa las siguientes funcionalidades en JavaScript:

**crearProfesor**

Parámetros: no recibe parámetros

Crear un objeto profesor tomando los valores de las cajas de la página.

Retorno:

Retorna el objeto creado

**agregarProfesor**

Parámetros:

Invoca a la función **crearProfesor**

Agrega el objeto **profesor**, al arreglo de profesores

Imprimir el arreglo en consola

Retorna:

No retorna nada

**PÁGINA**

Agregar tres inputs con los tres campos solicitados.

Agregar un botón GUARDAR en la página html, este botón invoca a **agregarProfesor**.

**refrescarProfesor**

no recibe parámetros

Muestra todos los profesores del arreglo en una tabla de la página html

no retorna nada

**PÁGINA HTML**

Agregar un botón REFRESCAR que invoca a **refrescarProfesor**

Se debe mostrar los profesores al presionar el botón REFRESCAR (evento onload)

**buscarProfesor**

Parámetros: recibe el Identificador único del profesor

Toma el valor de la caja txtBuscar

Busca profesor en el arreglo.

Retorno:

En un Alert “Profesor encontrado”, en caso de no encontrar, retorna “Profesor no encontrado”

**PÁGINA HTML**

Agregar un input para ingresar el identificador único del profesor

Agregar un botón BUSCAR que invoca a **buscarProfesor**

Usa un formulario en HTML para interactuar con las funciones (botones para agregar, y buscar). Agregar estilos y diseño personalizado (uso de imágenes)

**Evaluación Individual Final: Gestión de Datos Académicos**

Cada estudiante de forma individual, debe resolver 1 de los siguientes 4 ejercicios relacionados con un sistema de datos académico y debe estar desplegado en en servidor web Apache Tomcat, cada ejercicio simula un sistema por separado.

Tiempo total de la evaluación: 2 horas.

**Ejercicio 1: Gestión de Aula(crear, buscar)**

**Objetivo**: Manejar un conjunto de aulas usando objetos y arreglos en JavaScript, y mostrar los resultados en HTML.

Instrucciones:

1. Crear una carpeta llamada GestionAulas\_[Nombre\_Alumno], y dentro de ella, generar los archivos gestionAulas\_[Nombre\_Alumno].html y gestionAulas\_[Nombre\_Alumno].js.

En el archivo js crear un arreglo de **aulas**, inicialmente sin elementos.

Los valores de profesorAula (txtNumeroAula, txtHorario, txtCapacidadMáxima)

1. Implementa las siguientes funcionalidades en JavaScript:

**crearAula**

Parámetros: no recibe parámetros

Crear un objeto aula tomando los valores de las cajas de la página.

Retorno:

Retorna el objeto creado

**agregarAula**

Parámetros:

Invoca a la función **crearAula**

Agrega el objeto **aula**, al arreglo de aulas

Imprimir el arreglo en consola

Retorna:

No retorna nada

**PÁGINA**

Agregar tres inputs con los tres campos solicitados.

Agregar un botón GUARDAR en la página html, este botón invoca a **agregarAula**.

**refrescarAula**

no recibe parámetros

Muestra todas las aulas del arreglo en una tabla de la página html

no retorna nada

**PÁGINA HTML**

Agregar un botón REFRESCAR que invoca a **refrescarAula**

Se debe mostrar las aulas al presionar el botón REFRESCAR (evento onload)

**buscarAula**

Parámetros: recibe el Identificador único del aula

Toma el valor de la caja txtBuscar

Busca aula en el arreglo.

Retorno:

En un Alert “Aula encontrada”, en caso de no encontrar, retorna “Aula no encontrada”

**PÁGINA HTML**

Agregar un input para ingresar el identificador único del aula

Agregar un botón BUSCAR que invoca a **buscarAula**

Usa un formulario en HTML para interactuar con las funciones (botones para agregar, y buscar). Agregar estilos y diseño personalizado (uso de imágenes)