Equipo Pagina Web

Hewlett-Packard | [Dirección de la compañía]

Requerimientos WEB Service y Base de Datos

Equipo 1, Pagina web

1. Cruz ponce eliot gabriel
2. luna fonseca javier neftali
3. manjarrez carrillo ander osvaldo
4. romero gonzalez roger

**Requerimientos**

**Requerimiento 1 Base de datos**

Necesito una base datos que contenga 8 tablas:

Tabla Alumno:

Campo Matricula:

* Tamaño máximo de caracteres 10

Campo Nombre:

* Tamaño máximo de caracteres 100

Campo Apellido\_Paterno:

* Tamaño máximo de caracteres 50

Campo Apellido\_Materno:

* Tamaño máximo de caracteres 50

Campo Edad:

* Tipo entero

Campo Estatus:

* True o False, tipo bit

Campo Id\_Carrera

* Tipo entero

Tabla Inscribir:

Campo Id\_Inscripcion:

* Auto generable

Campo Matricula\_Alumno:

* Tamaño máximo de caracteres 10

Campo Clave\_Grupo:

* Tamaño máximo de caracteres 10

Campo Fecha:

* Tipo date

Tabla Carrera:

Campo Id\_Carrera:

* Tipo entero, auto generable

Campo Nombre:

* Tamaño máximo de caracteres 50

Tabla Grupo:

Campo Clave:

* Tamaño máximo de caracteres 10

Campo Materia:

* Tamaño máximo de caracteres 50

Campo Id\_Carrera:

* Tipo Entero

Campo Matricula\_Docente:

* Tamaño máximo de caracteres 10

Tabla GrupoDetalle:

Campo Clave\_Grupo:

* Tamaño máximo de caracteres 10

Campo Día:

* Tipo Entero

HoraInicio:

* Tipo Entero

HoraFin:

* Tipo Entero

Id\_Salon:

* Tipo Entero

Tabla Docente:

Campo Matricula:

* Tamaño máximo de caracteres 10

Campo Nombre:

* Tamaño máximo de caracteres 100

Campo Apellido\_Paterno:

* Tamaño máximo de caracteres 100

Password:

* Tamaño máximo de caracteres 50

Tabla Salón:

Campo Id\_Salon

* Tipo Entero

Campo Clave

* Tamaño máximo de caracteres 5

Tabla Usuarios:

Campo Usuario

* Tamaño máximo de caracteres 50

Campo Password

* Tamaño máximo de caracteres 50

La tabla Alumno debe tener como llave primaria la Matricula y debe relacionarse con la tabla Inscribir.

La tabla Carrera debe tener como llave primaria el Id\_Carrera y relacionarse con un alumno. Esta tabla debe tener como mínimo 5 carreras ya agregadas.

La tabla Grupo debe tener como llave primaria la Clave y debe relacionarse con la tabla Inscribir, también debe contener una relación con la tabla Docente y la tabla GrupoDetalle.

La tabla Inscribir debe tener dos tablas relacionadas con la misma y una llave primaria que es el Id\_Inscripcion.

La tabla Docente debe tener una llave primaria Matricula y debe relacionarse con la tabla Grupo. La Tabla ya deberá tener maestros agregados en ella, al menos 10 maestros.

La tabla Salon debe tener como llave primaria Id\_Salon y debe relacionarse con la tabla GrupoDetalle.

La tabla Usuarios tendrá como llave primaria Usuarios y no se relacionara con nada.

**Requerimiento 2 Servicios WEB**

Métodos Alumno

Necesito un método InsertarAlumno():

* **Recibe:** siete strings, Matricula, Nombre, Apellido Paterno, Apellido Materno, Edad, Estatus y Carrera respectivamente. Estos campos se enviarán encriptados con json.
* **Regresa:** un true si los datos se insertaron correctamente y un false si los datos no se pudieron insertar.
* **Descripción:** Este método servirá para dar de alta un alumno en el sistema.

Necesito un método ActualizarEstatus():

* **Recibe:** Dos string, el primer string será la matricula del alumno y el segundo será el nuevo estado del estatus del Alumno y se enviará encriptado en json.
* **Regresa:** True si la matricula existe y se actualizo el estado por el nuevo o falsede lo contrario.
* **Descripción:** Servirá para cambiar el estatus de un alumno, para darlo de baja y que no se pueda volver a inscribir. Este método solo cambiará el estatus de un alumno.

Necesito un método ConsultarAlumno():

* **Recibe:** un string con el campo matricula y nombre de la carrera.
* **Regresa:** un arreglo de strings con los campos de la tabla.
* **Descripción:** Este método servirá para ver los datos de un alumno según el filtro de matrícula y nombre de la carrera.

Necesito un método BuscarAlumno()

* **Recibe:** Un string para el campo “MatriculaAlumno”.
* **Regresa:** Un true o un false indicando si el alumno existe o no.
* **Descripción:** Este método permitirá saber si existe un alumno.

Métodos Inscripción

Necesito un método InsertarInscripcion():

* **Recibe:** un string para cada campo de la tabla, excepto el id
* **Regresa:** true o false si se realizó correctamente la inserción de la inscripción
* **Descripción:** Este método realizar la inserción de los datos enviados en la tabla inscripción, agregar un alumno y un grupo servirá para registrar a un alumno en ese grupo.

Necesito un método InscribirAlumno()

* **Recibe:** Un string para el campo “MatriculaAlumno”, un string para el campo “ClaveGrupo” y un string para el campo “Fecha”.
* **Regresa:** Un true o un false indicando si se inscribió el alumno al grupo o no
* **Descripción:** Este método permitirá inscribir un alumno a un grupo.

**Método Docentes**

Necesito un método ConsultarDocente():

* **Recibe:** un string para el campo matricula
* **Regresa:** un arreglo de strings con los campos de la tabla
* **Descripción:** Este método servirá para ver los datos del Docente con un filtro de su matrícula.

**Método Carrera**

Necesito un método ConsultarCarrera():

* **Recibe:** un string para el campo nombre de carrera
* **Regresa:** un true o false, si existe esa carrera o no existe
* **Descripción:** Este método servirá para validar que no ingrese una carrera que no esté dentro de la base de datos.

**Método Login**

Necesito un método Login()

* **Recibe:** dos strings, el primero para Usuario y el segundo para la contraseña estos campos se enviarán encriptados con json.
* **Regresa:** un true si el usuario y contraseña son correcto y un false sino lo son.
* **Descripción:** Este método servirá para validar que el usuario y contraseña son correcto al momento de que un docente ingresa a la página web.

**Métodos Grupos**

Necesito un método InsertarGrupo():

* **Recibe:** 7 strings. Grupo: Materia, Id\_Carrera, Matricula Docente. GrupoDetalle: Dia, HoraInicio, HoraFin. Salon: Clave.
* **Regresa:** un dato de tipo entero el cual regresara uno de los siguientes valores correspondientes:

1. Se inserto
2. No se insertó porque el docente está ocupado a esa misma hora.
3. No se insertó porque el salón está ocupado a esa misma hora.
4. No se insertó porque el salón y el docente está ocupado a esa misma hora.
5. No se insertó.

* **Descripción:** Este método realizara la inserción para grupo. Deberá validar si el salón no está en uso; si el docente no está ocupado para la hora designada.

Necesito un método BorrarGrupo()

* + **Recibe:** Un string para el campo “Clave”
  + **Regresa:** Un true o un false indicando si el grupo se borró o no.
  + **Descripción:** Este método permitirá dar de baja un grupo.

Necesito un método ConsultarGrupo():

* + **Recibe:** un string con el campo clave
  + **Regresa:** un arreglo de strings con los campos de la tabla.
  + **Descripción:** Este método servirá para ver los datos de un grupo según el filtro de su clave.

Necesito un método ActualizarGrupo()

* + **Recibe:** Un string para el campo “Clave”, un string para el campo “Materia”, un string para el campo “IdCarrera” y un string para el campo “MatriculaDocente”
  + **Regresa:** Un true o un false indicando si el grupo se actualizo o no.
  + **Descripción:** Este método permitirá actualizar un grupo.

Necesito un método ExistenciaGrupo()

* + **Recibe:** Un string para el campo “Clave”
  + **Regresa:** Un “true” o un “false” indicando si existe o no el grupo.
  + **Descripción:** Este método permitirá saber si un grupo existe o no, previo a cualquier ABC a realizar a dicho grupo.

Necesito un método ObtenerGrupo()

* + **Recibe:** Un string para el campo “Clave”
  + **Regresa:** Un arreglo que contenga la clave del grupo, materia y nombre de la carrera.
  + **Descripción:** Este método permitirá obtener la información del grupo al que se le vaya a realizar algún ABC.