



ÍNDICE

ANÁLISIS	2
Requerimientos	2
DISEÑO	3
Base de datos	3
Entidades detectadas	3
Puntos a tener en consideración	3
Diseño que cumple con las necesidades requeridas así como teniendo en cuenta posible escalabilidad (Entidad relación y Modelo lógico con notación de Barker)	4
TECNOLOGÍAS UTILIZADAS	5
Frontend	5
Backend	5
Base de datos	5
Índices seleccionados	5



ANÁLISIS

Requerimientos

- Almacenar nombres, apellidos, teléfono, dirección, fecha de nacimiento y otro dato personal de relevancia (Datos de una persona)
 - Estas personas pueden tener o no uno o varios logros académicos de los cuales pueden ser licenciatura, maestría, doctorado o también diplomados, certificaciones entre otros tipos de logros académicos, el logro académico debe de tener registrado año, título del programa, institución de la que lo obtuvo y otra información que sea necesaria y pertinente.
 - Estas personas pueden tener o no una o varias áreas de desempeño, por ejemplo: ciencias de la computación, psicología, ciencias económicas, entre otros.
 - Cada persona deberá tener un usuario y contraseña para ingresar al sistema
- El sistema deberá contar con las siguientes funcionalidades
 - Login
 - Actualización de datos personales (¿usuario autenticado?)
 - Actualización de datos académicos (¿usuario autenticado?)
 - Actualizar área de desempeño (¿usuario autenticado?)
 - Agregar o quitar logros académicos (¿usuario autenticado?)
 - Agregar o quitar área de desempeño (¿usuario autenticado?)
 - Una vista (no se necesita un login para acceder) en la cual se pueda visualizar las personas registradas, así como el último grado académico que han conseguido con su respectiva información de dicho grado académico.
 - Una vista (no se necesita un login para acceder) en la cual se pueda visualizar las áreas de desempeño registradas, así como el número de personas que tienen esa área de desempeño.



DISEÑO

Base de datos

Entidades detectadas

Se utilizaron nombres en inglés para las tablas debido al uso de la letra 'ñ' para la entidad area de desempeño, de esta manera logramos tener una mejor práctica en el diseño de la misma así como se decidió usar snake case para los identificadores.

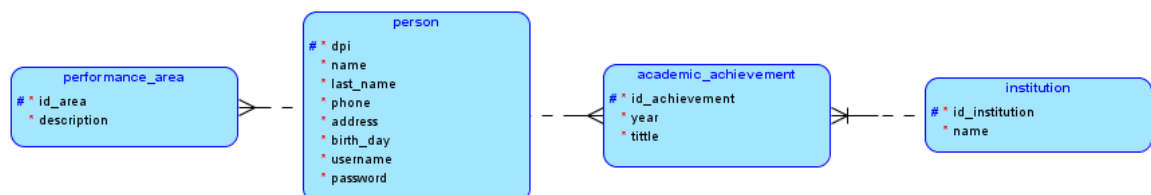
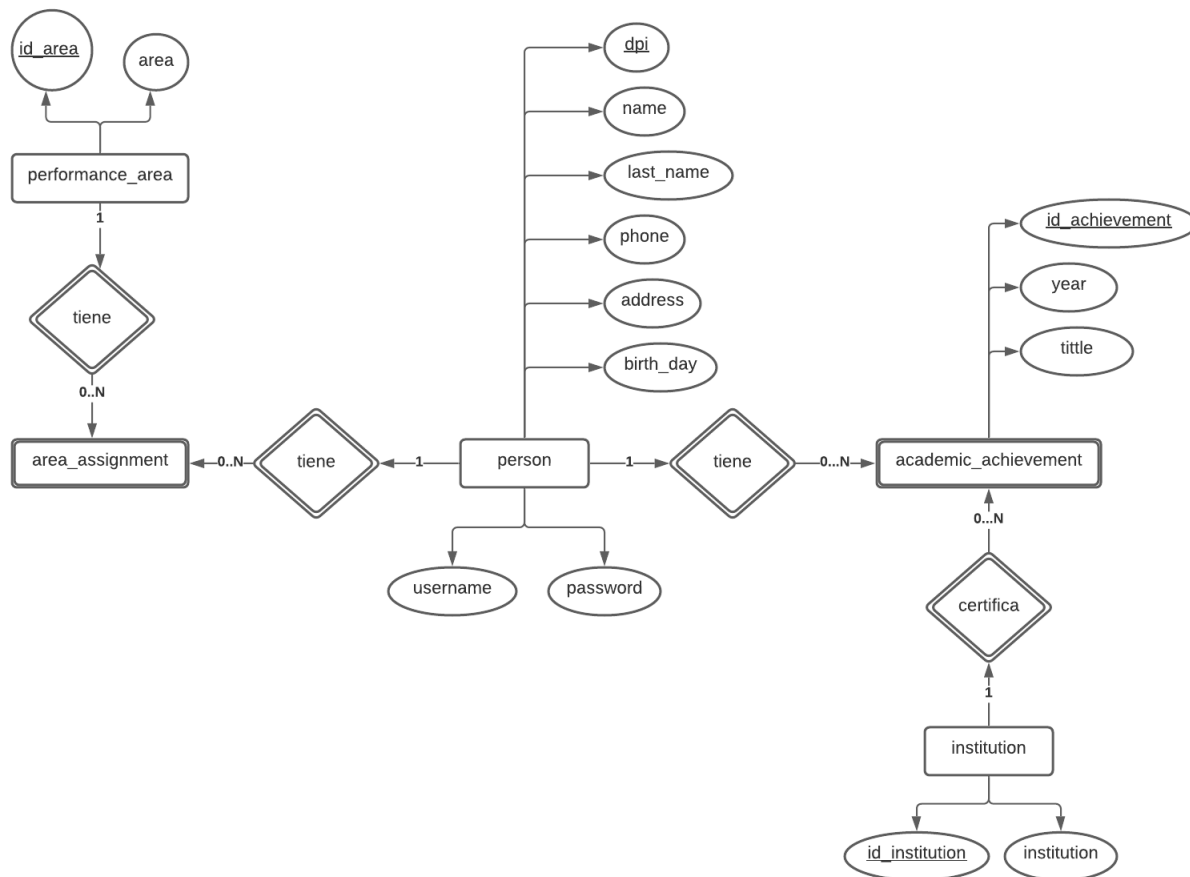
- Persona -> Tabla person
- Logro académico -> Tabla academic_achievement
- Área de desempeño -> Tabla performance_area
- Usuario -> Tabla user
- ¿Institución?

Puntos a tener en consideración

- Dado que no existen jerarquías entre usuarios, por lo que no es necesario tener una tabla específicamente para usuarios dado que una persona tendrá uno y solo un usuario.
- La institución se coloca como un campo de la tabla academic_achievement debido a que no se considera en el enunciado que podrían existir títulos otorgados por la misma institución, dándonos redundancia de datos innecesaria.



Diseño que cumple con las necesidades requeridas así como teniendo en cuenta posible escalabilidad (Entidad relación y Modelo lógico con notación de Barker)





TECNOLOGÍAS UTILIZADAS

Frontend

- Librería bootstrap
- Consumo de API utilizando php
- Xampp
 - php
 - apache

Backend

- Módulos de NodeJs
 - Sequelize (ORM), para manipular la data de manera más segura y simple.
 - mysql, para realizar la conexión con la base de datos
 - express, para el manejo de las rutas
 - cors, para el manejo del intercambio de recursos de origen cruzado
 - bcrypt, para encriptación de contraseñas
 - jsonwebtoken, para la autenticación de usuarios

Base de datos

- MySQL

Índices seleccionados

Debido a la 'naturaleza' de nuestro software, la mayoría de búsquedas se realizan mediante valores enteros, por lo que la carga de cpu en estas búsquedas no se ve afectada en gran medida, por otro lado una consulta que se podría ver afectada ocurre en la tabla persona, esto debido a que en determinado momento se realiza una búsqueda en base a la columna username, la cual es una columna de texto plano, teniendo en cuenta que esta tabla es la que tendrá la mayor cantidad de información, es un lugar clave para agregar un índice.