



Prueba Práctica de Programación

Prueba

Escribir un programa que resuelva los siguientes requerimientos:

Contexto:

La empresa istmocenter requiere almacenar la información de los usuarios de las diferentes compañías con las que trabaja, la información a almacenar es la siguiente:

Información de la compañía (Company):

Id (id)	int autoincrementable
Nombre (name)	string
Eslogan (catchphrase)	string
Negocio (bs)	string
Usuarios (users)	array<User>

Ejemplo:

```
"company": {  
  "id": 1,  
  "name": "Romaguera-Crona",  
  "catchPhrase": "Multi-layered client-server neural-net",  
  "bs": "harness real-time e-markets",  
  "users": []  
}
```

Información de los usuarios (User):

Id (id)	int autoincrementable
Nombre (name)	string
Nombre de Usuario (username)	string
Correo (email)	string
Dirección (address)	Address
Teléfono (phone)	string
Dirección web (website)	string

Ejemplo: "user":

```
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  "address": {
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-3874"
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org"
}
```

Información de dirección de usuario (Address):

Calle (street)	string
Apartamento o casa (suite)	string
Ciudad (city)	string
Código postal (zipcode)	string combo único no repetible en la tabla

Ejemplo:

```
"address": {
  "street": "Kulas Light",
  "suite": "Apt. 556",
  "city": "Gwenborough",
  "zipcode": "92998-3874"
},
```

Parte I

Requisito:

Crear la estructura en base de datos (Mysql o Postgres o Sqlserver) que permita almacenar dicha información relacionada de manera que una compañía pueda tener varios usuarios y cada usuario pueda tener solo una dirección.

Ingresar a la base de datos mínimo 2 compañías cada una con mínimo 5 usuarios.

Respetar los nombres en ingles al crear las tablas y campos.

Entregable:

Archivo sql con la base de datos (script de base de datos)

Nota:

El código postal siempre llevara el siguiente formato 34244-2342 cinco números seguido de un guion seguido de cuatro números más. No es necesaria su validación al insertarlo solo se avisa que ese formato siempre se respetara. La validación de que dicho campo (debe ser único) si se debe crear en la base de datos.

Parte II

Requisito:

Consumir dicha base de datos con cualquiera de los siguientes Frameworks (Koa o Express)

Con dicha información se debe de crear las siguientes vistas (interfaz gráfica en vuejs)

- a. Vista de listado de Compañías la cual podrá mostrar la vista de usuarios (b) a través de un botón o link.
- b. Vista de listado de Usuarios por compañía mostrar la información del usuario más la información de la dirección por cada usuario.

Entregable:

Carpeta del proyecto comprimida en zip

Nota:

No se califica el diseño de la interfaz gráfica solo el uso de css y vuejs.

Parte III

Requisito:

Se deberá crear una Api RestFull la cual deberá devolver la información de todos los usuarios en formato json respetando el formato descrito en el ejemplo de la respuesta del Api. La información del usuario debe estar ordenada por un método el cual hará que la información devuelta esta ordenada por el campo de zipcode de cada usuario de la siguiente manera:

Ordenara los usuarios por el zicode basándose en el último número de este de manera ascendente, en el caso de que varios usuarios tengan igual el último número del zipcode estos se ordenaran basándose en el número anterior y así subsecuentemente. Obviando el guion.

Ejemplo de orden de Usuarios.**Caso 1**

Maria zipcode 12344-1237, Juan zipcode 12113-1279, Marcos zipcode 12424-1238

Ultimo número de María es 7, el y ultimo de Juan es 9 y el ultimo de Marcos es 8 por ende el orden de estos en la lista seria Maria, Marcos, Juan

Caso 2

Maria zipcode 12344-1237, Juan zipcode 12113-1276, Marcos zipcode 12424-1247

Ultimo número de María es 7, el y ultimo de Juan es 6 y el ultimo de Marcos es 7 por ende el orden el primero en la lista será Juan por tener el menor, pero en el caso de Maria y Marcos nos fijaremos en su antepenúltimo numero el de Maria seria 3 y el de Marcos es 4 por ende el orden de estos en la lista seria Juan, Maria, Marcos

Ejemplo de la respuesta del Api:

```
[
  {
    "id": 3,
    "name": "Leanne Graham",
    "username": "Bret",
    "email": "Sincere@april.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough",
      "zipcode": "92998-3874"
    },
    "phone": "1-770-736-8031 x56442",
    "website": "hildegard.org",
    "company": {
      "id": 1,
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  },
  {
    "id": 2,
    "name": "Ervin Howell",
    "username": "Antonette",
    "email": "Shanna@melissa.tv",
    "address": {
      "street": "Victor Plains",
      "suite": "Suite 879",
      "city": "Wisokyburgh",
      "zipcode": "40566-7778"
    },
    "phone": "010-692-6593 x09125",
    "website": "anastasia.net",
    "company": {
```

```
    "id": 1,  
    "name": "Romaguera-Crona",  
    "catchPhrase": "Multi-layered client-server neural-net",  
    "bs": "harness real-time e-markets"  
  }  
}  
]
```

Entregable:

Carpeta del proyecto comprimida en zip (esta funcionabilidad puede ser un proyecto aparte o puede estar integrado en el proyecto creado en la parte II)

Nota:

Como se puede ver en el ejemplo de la respuesta de la Api cada usuario debe mostrar tanto la dirección (address) como la compañía a la cual pertenece (company).