

1. Se te ha pedido implementar funcionalidad para manejar usuarios y sus direcciones. Para ello, considera que un usuario puede tener muchas direcciones. Estas direcciones deberían estar en una tabla aparte, para poder manejar cada dato de la dirección por separado.
  - a. Implementa los modelos necesarios para vincular un usuario con sus direcciones.
  - b. Implementa rutas, controladores y HTML/JS en el frontend de modo que un usuario pueda ingresar su nombre de usuario y los detalles de su dirección y al hacer click en un botón, que la dirección se guarde en la base de datos.
  - c. Implementa rutas, controladores y HTML/JS en el frontend de modo que un usuario pueda ingresar su nombre de usuario y al hacer click en un botón, muestre todas las direcciones en un formato que tenga sentido visualmente.
  - d. Implementa HTML/JS en el frontend de modo que al hacer click en otro botón, no se muestren todas las direcciones, si no las que estén ubicadas en un distrito en particular.
2. Los mismos usuarios del ejercicio anterior también quieren hacer pedidos de órdenes por la página Web. Sin embargo, son ordenes caras, así que varios usuarios se pueden juntar para comprarlas.
  - a. Implementa los modelos necesarios para vincular un usuario con sus órdenes y viceversa. Las órdenes deberían tener, además, un campo para su monto y la fecha.
  - b. Implementa las rutas, controladores y HTML/JS en el frontend de modo que un usuario pueda listar todos los otros usuarios con los que quiere realizar una compra y el monto de la orden para agregarla a la base de datos.
  - c. Implementa las rutas, controladores y HTML/JS en el frontend de modo que se pueda ingresar un monto y se devuelva el id, fecha, y monto de las órdenes con costo mayor al ingresado.
  - d. Implementa las rutas, controladores y HTML/JS en el frontend de modo que se pueda ingresar el ID de una orden y se muestran todos los usuarios que la compraron, junto con su monto y la fecha.