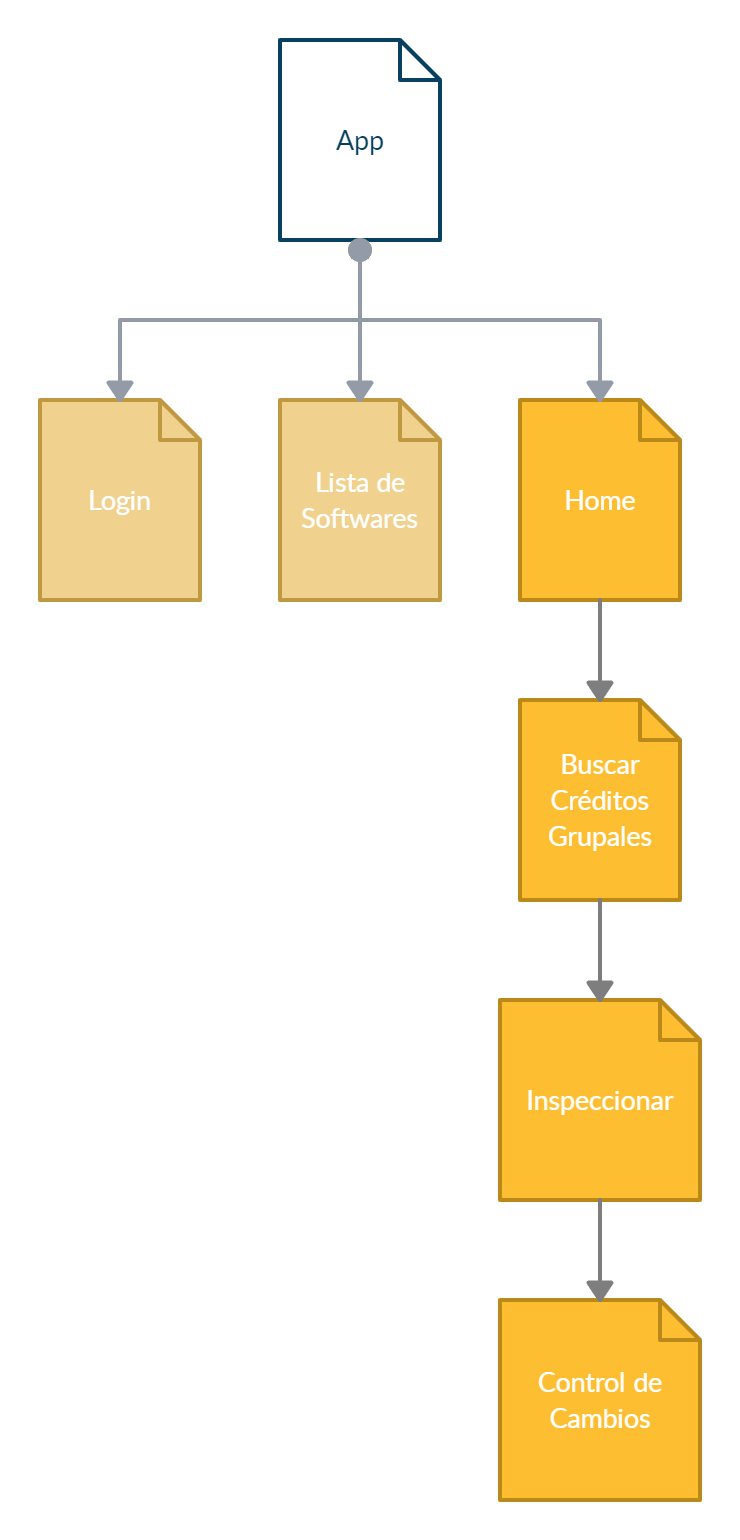
**PRUEBA DE PROGRAMACIÓN FRONTEND**

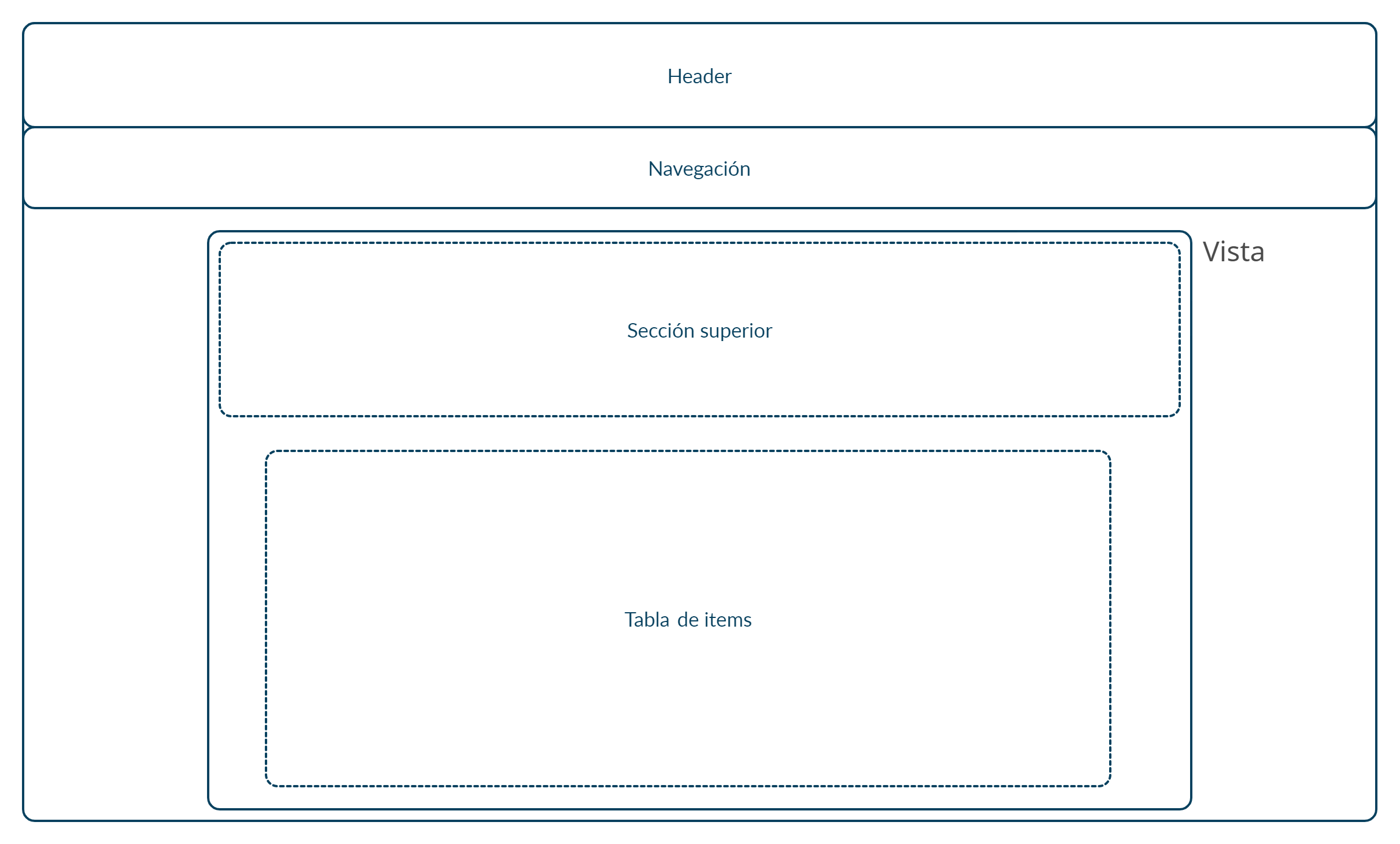
De Canhui Huang

La aplicación es desarrolada en Vue.js, no se utilizó ninguna librería o preprocesadores de estilos. Se usó font-awesome para para los íconos. Y se utilizaron estas librerías: vue-router para el routing, axios para http requests. No se utilizó ninguna herramienta de manejo de estados.

La aplicación tiene 6 vistas, organizadas así:



Este es el layout/diseño de la aplicación:



Al navegar por diferentes vistas, lo único que cambiaría en la aplicación es la sección de vista, donde todas nuestras vistas tienen un diseño similar. Todas las vistas tienen una sección superior que contiene: título y/o botones, barras de búsqueda, etc. Debajo de sección superior está la tabla de ítems, donde cada ítem es un componente reutilizable llamado **RowItem** para todas las vistas.

Por lo tanto, el componente de las vistas tienen un template más o menos así:

<div class="vista">

    <div class="topbar">

      …

    </div>

    <table class="items">

      <RowItem

        v-for="(item) in mis\_items "

        :data="item"

        :fractions="[1, 3, 14, 1]"

      />

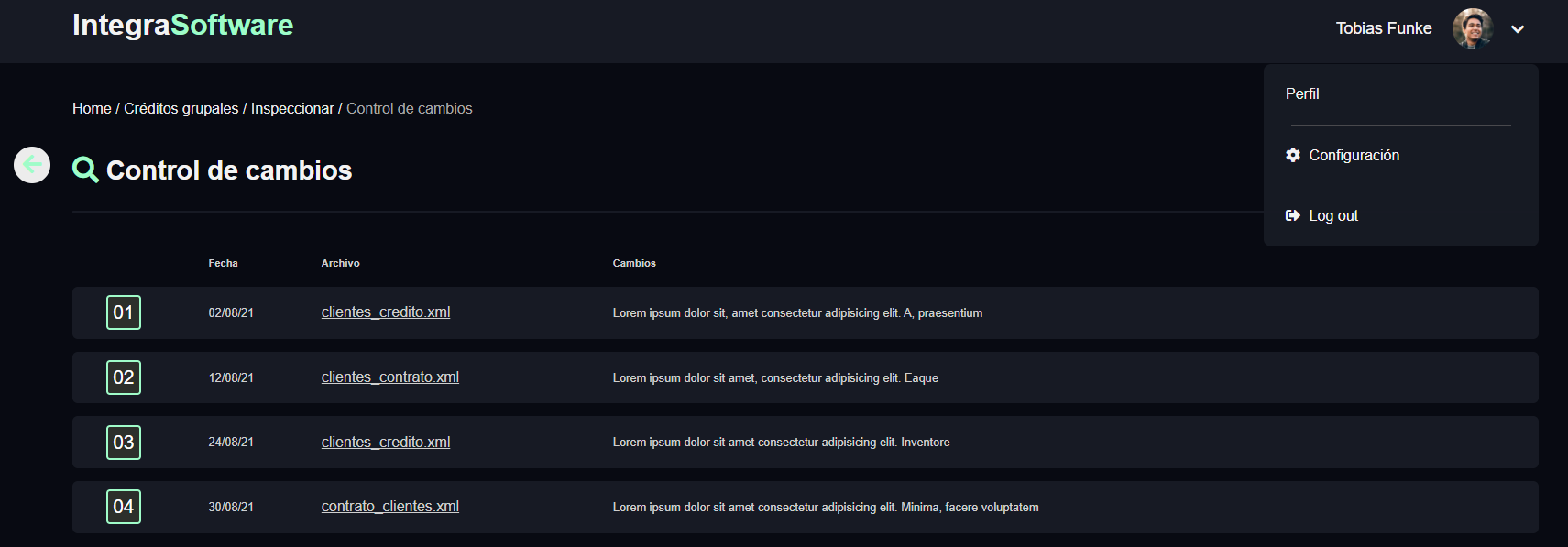
    </table>

</div>

Login es la única vista que no tiene este diseño, que consite en un simple form que toma input, y tiene la funcionalidad de loguear.

Los componentes vistas de Crédito Grupales, Control de cambios, y Listado de Softwares crean sus items con los datos consumiendo de un backend, en este caso, es un restful API falso(<https://my-json-server.typicode.com/canhuiHuang/vuetask/>) que tiene los datos mostrado en las imágenes del documento de **PRUEBA DE PROGRAMACIÓN FRONTEND** para hacer GET requests y simular POST requests. Se usó la librería axios para hacer los http calls.

Se puede simular el logueo con el correo de cualquier usuario de <https://reqres.in/> con cualquier contraseña. La sesión es persistente con la información guardada en localStorage.  
Al estar logueado, se podrá desloguearse con el menú desplegable el component de perfil en la parte superior derecha.



Se usó vue-utils, vue-jest y se realizó pruebas unitarias simples del UI

