




# Vue

Mantenimiento de Pacientes

## Requisitos

- Tener instalado Visual Studio Code
- Node JS








# Extensiones Vue

	<b>Vue VSCode Snippets</b> 1.6.0 Snippets that will supercharge your Vue workflow sarah.drasner	🔗 463K ★ 5 ⚙️
	<b>vue</b> 0.1.5 Syntax Highlight for Vue.js jcbuisson	🔗 124K ★ 4.5 ⚙️
	<b>Vue Peek</b> 1.0.2 Allows peek and goto definition for Vue single-file components. Dario Fuzinato	🔗 176K ★ 5 ⚙️

# Backend

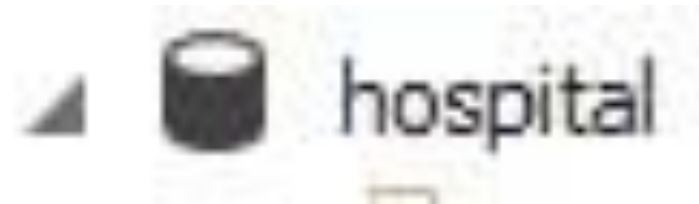
- Debes abrir el proyecto backend en Visual Studio Code

## BACKEND

- ▶  .vscode
- ▶  Hospital.Api
- ▶  Hospital.Entity
- ▶  Hospital.Repositoy
- ▶  Hospital.Service
- ▶  obj
-  Hospital.sln

# Backend

- Debes tener creado la base de datos hospital en SQL Server



# Backend

- Debes ejecutar la migración para crear la tabla paciente recuerda que debes ejecutar el comando dentro del proyecto Hospital.Repository.

```
\Backend\Hospital.Repositoy> dotnet ef --startup-project ..\Hospital.Api\ migrations add Init
```

```
\Backend\Hospital.Repositoy> dotnet ef --startup-project ..\Hospital.Api\ database update
```

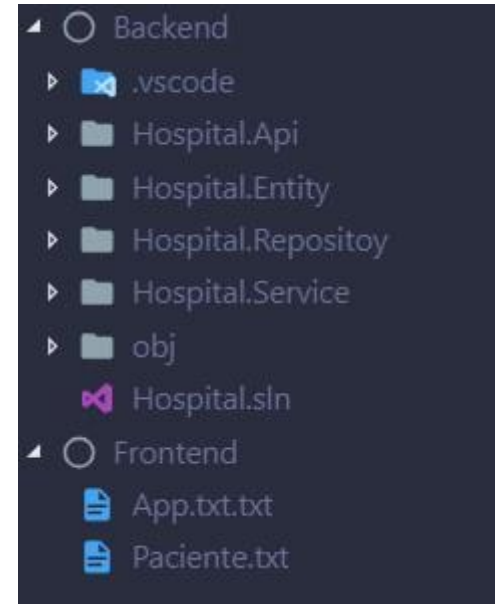
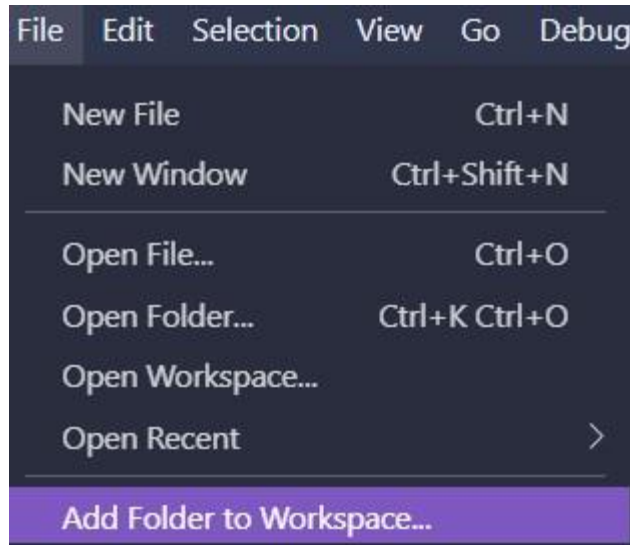
# Backend

- Verifica en la base de datos que se haya creado la tabla Paciente.



# Frontend

- Debes agregar carpeta Frontend al Visual Studio Code





# Frontend

- Instalamos Vue CLI para esto debes ubicarte dentro de la carpeta Frontend.

```
\Proyecto\Frontend> npm install -g @vue/cli
```

# Frontend

- Luego creamos proyecto con vue recuerda que debes ejecutar este comando dentro de la carpeta Frontend.

```
Proyecto\Frontend> vue create hospital
```

# Frontend

- Selecciona la opción Manually select features luego da ENTER

```
Vue CLI v3.8.2
? Please pick a preset:
  default (babel, eslint)
> Manually select features
```

# Frontend

- Elegir los siguientes features luego dar ENTER

```
? Check the features needed for your project:
(*) Babel
( ) TypeScript
( ) Progressive Web App (PWA) Support
(*) Router
(*) Vuex
( ) CSS Pre-processors
>( ) Linter / Formatter
( ) Unit Testing
( ) E2E Testing
```

# Frontend

- Escribe Y, luego da ENTER

```
? Use history mode for router? (Requires proper server setup for index fallback in production) (Y/n) Y
```

# Frontend

- Selecciona la segunda opcion, luego da ENTER

```
? Where do you prefer placing config for Babel, PostCSS, ESLint, etc.?
  In dedicated config files
> In package.json
```

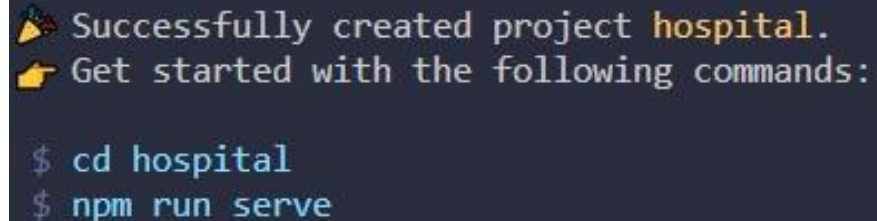
# Frontend

- Escribe N, luego da ENTER

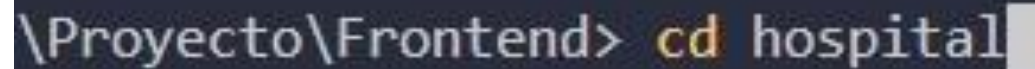
```
? Save this as a preset for future projects? (y/N) N
```

# Frontend

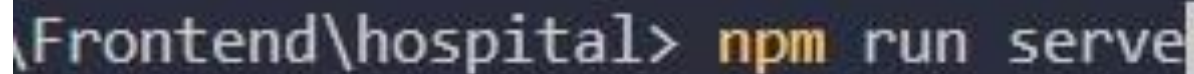
- Luego como indica el mensaje al finalizar la creación del proyecto debes ingresar a la carpeta del proyecto y escribir el siguiente comando para ejecutar el proyecto.



```
🎉 Successfully created project hospital.  
👉 Get started with the following commands:  
  
$ cd hospital  
$ npm run serve
```



```
\Proyecto\Frontend> cd hospital
```

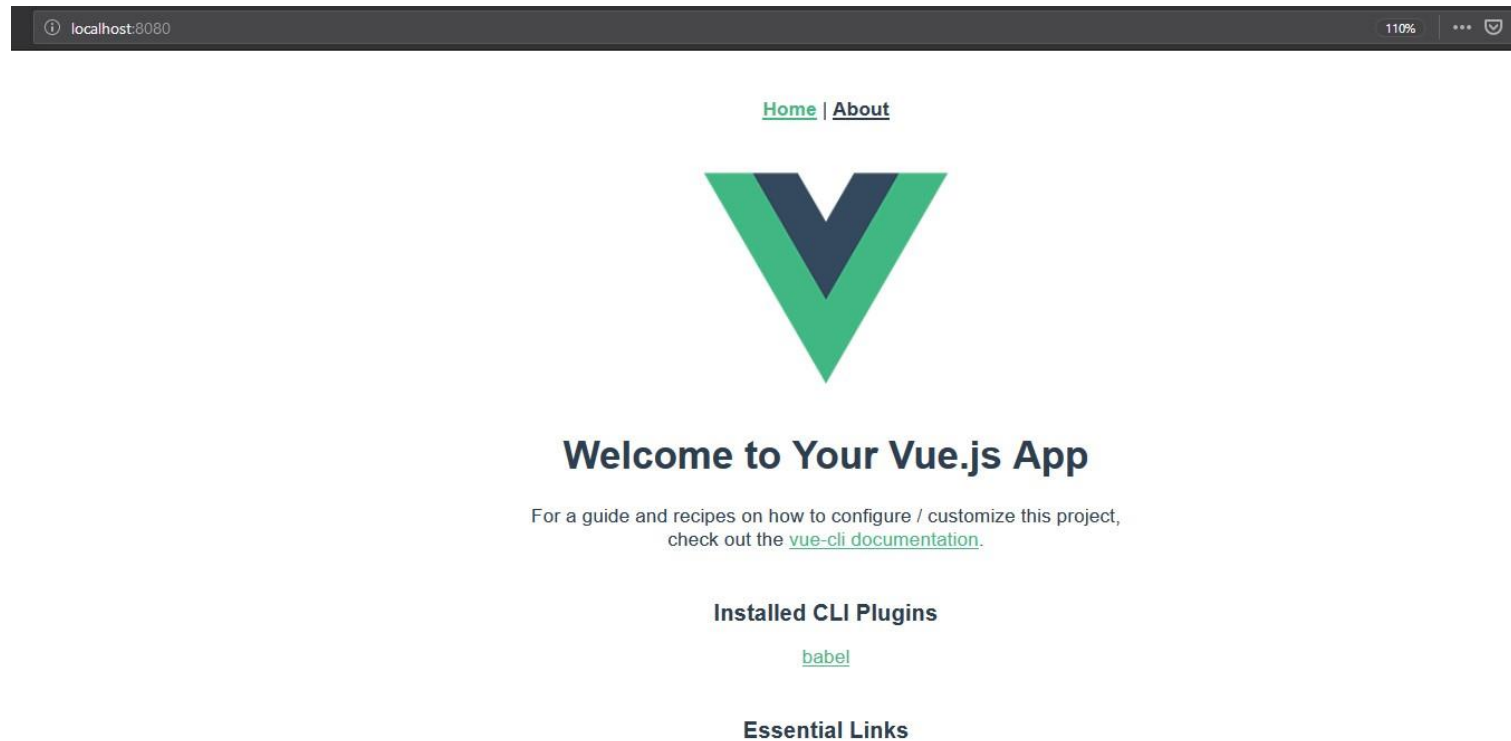


```
\Frontend\hospital> npm run serve
```



# Frontend

- Ingresa a <http://localhost:8080>



# Frontend

```
\Frontend\hospital> vue add vuetify
```

- Instalamos Vuetify, recuerda que para ejecutar el siguiente comando debes estar dentro de la carpeta hospital

# Frontend

- Selecciona la opción Configure (advanced) y luego da ENTER

```
? Choose a preset:  
Default (recommended)  
Prototype (rapid development)  
> Configure (advanced)
```

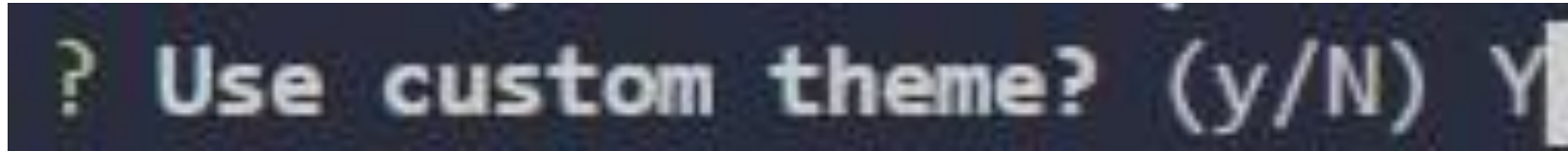
# Frontend

- Escribe Y, luego da ENTER

```
? Use a pre-made template? (will replace App.vue and HelloWorld.vue) (Y/n) Y
```

# Frontend

- Escribe Y, luego da ENTER

A screenshot of a terminal window with a dark background. The text "? Use custom theme? (y/N) Y" is displayed in a light-colored, monospaced font. A white cursor is positioned at the end of the line, after the letter 'Y'.

```
? Use custom theme? (y/N) Y
```

# Frontend

- Escribe Y, luego da ENTER

```
? Use custom properties (CSS variables)? (y/N) Y
```

# Frontend

- Selecciona Material Icons, luego da ENTER

```
? Select icon font (Use arrow keys)
> Material Icons
  Material Design Icons
  Font Awesome 5
  Font Awesome 4
```



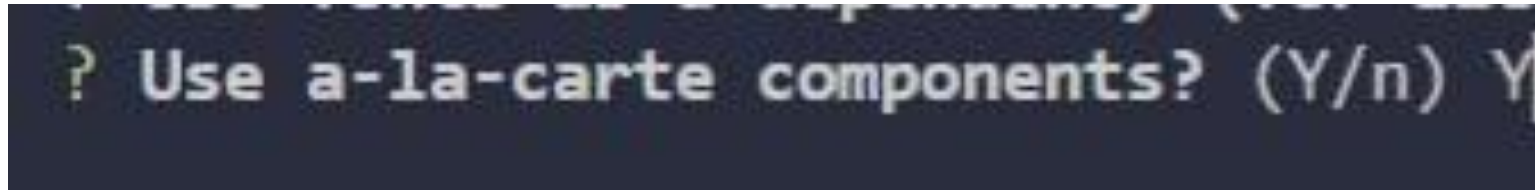
# Frontend

- Escribe N, luego da ENTER

```
? Use fonts as a dependency (for Electron or offline)? (y/N) N
```

# Frontend

- Escribe Y, luego da ENTER

A screenshot of a terminal window with a dark background. The text is light-colored. It shows a prompt character followed by the text "Use a-la-carte components? (Y/n)". The letter "Y" has been typed at the end of the line, and a white cursor is visible at the end of the "Y".

```
? Use a-la-carte components? (Y/n) Y
```

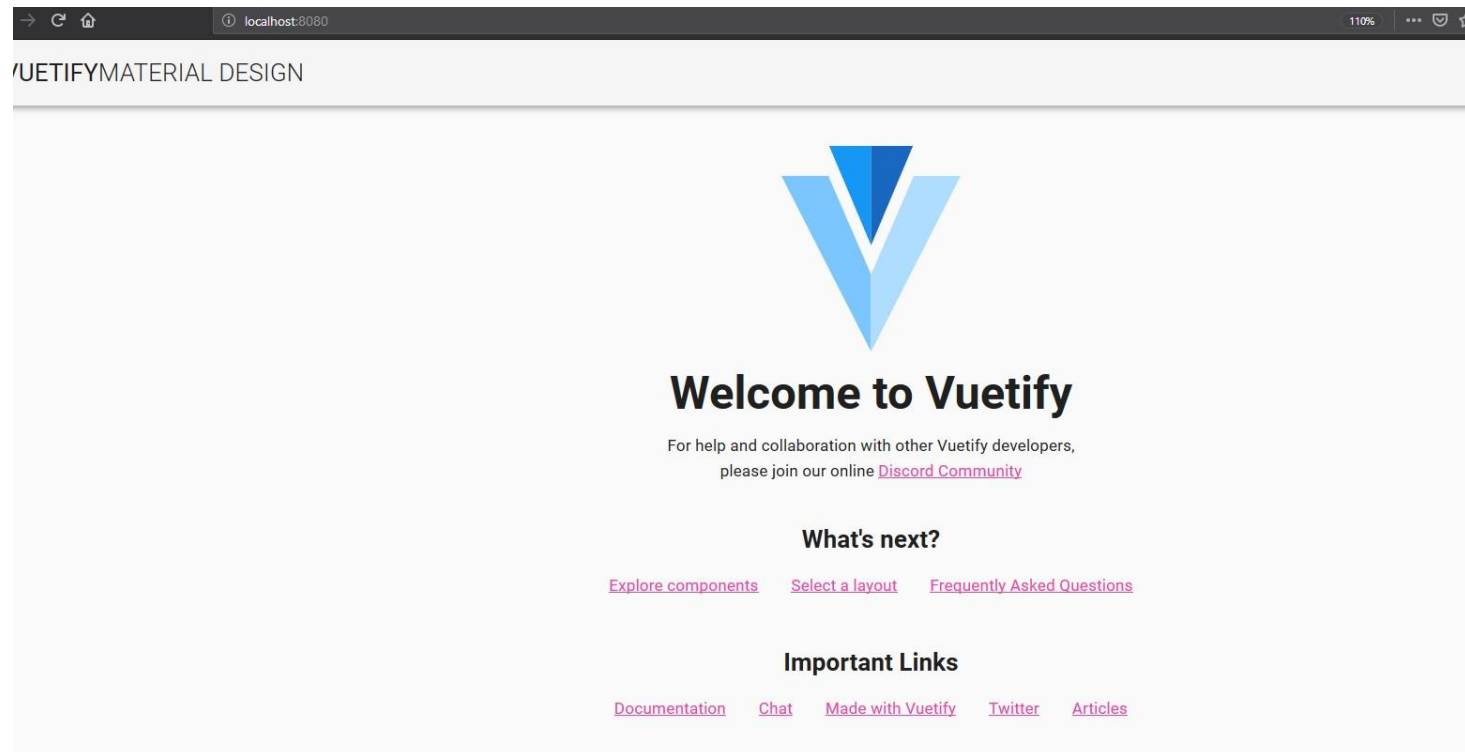
# Frontend

- Seleccionar English, luego da ENTER

```
? Select locale (Use arrow keys)
  Catalan
  German
  Spanish
> English
  Estonian
  Farsi
  Greek
(Move up and down to reveal more choices)
```

- Ingresar a <http://localhost:8080>

# Frontend



- Debe crear la ruta al componente **Paciente** en el archivo **router.js**

# Frontend

```
router.js  x
1  import Vue from 'vue'
2  import Router from 'vue-router'
3  import Home from './views/Home.vue'
4  import Paciente from './components/Paciente.vue'
5
6  Vue.use(Router)
7
8  export default new Router({
9    mode: 'history',
10    base: process.env.BASE_URL,
11    routes: [
12      {
13        path: '/',
14        name: 'home',
15        component: Home
16      },
17      {
18        path: '/pacientes',
19        name: 'pacientes',
20        component: Paciente
21      }
22    ]
23  })
24
```

# Frontend

- Debe instalar axios recuerda que esto lo debes hacer dentro del proyecto hospital.

**<https://github.com/axios/axios>**

```
\Frontend\hospital> npm install axios --save
```

# Frontend

- En el archivo main.js debe agregar la ruta al backend  
axios.defaults.baseURL='http://localhost:500/'

```
import Vue from 'vue'
import './plugins/vuetify'
import App from './App.vue'
import router from './router'
import store from './store'

import axios from 'axios'

Vue.config.productionTip = false

axios.defaults.baseURL = 'http://localhost:5000/'

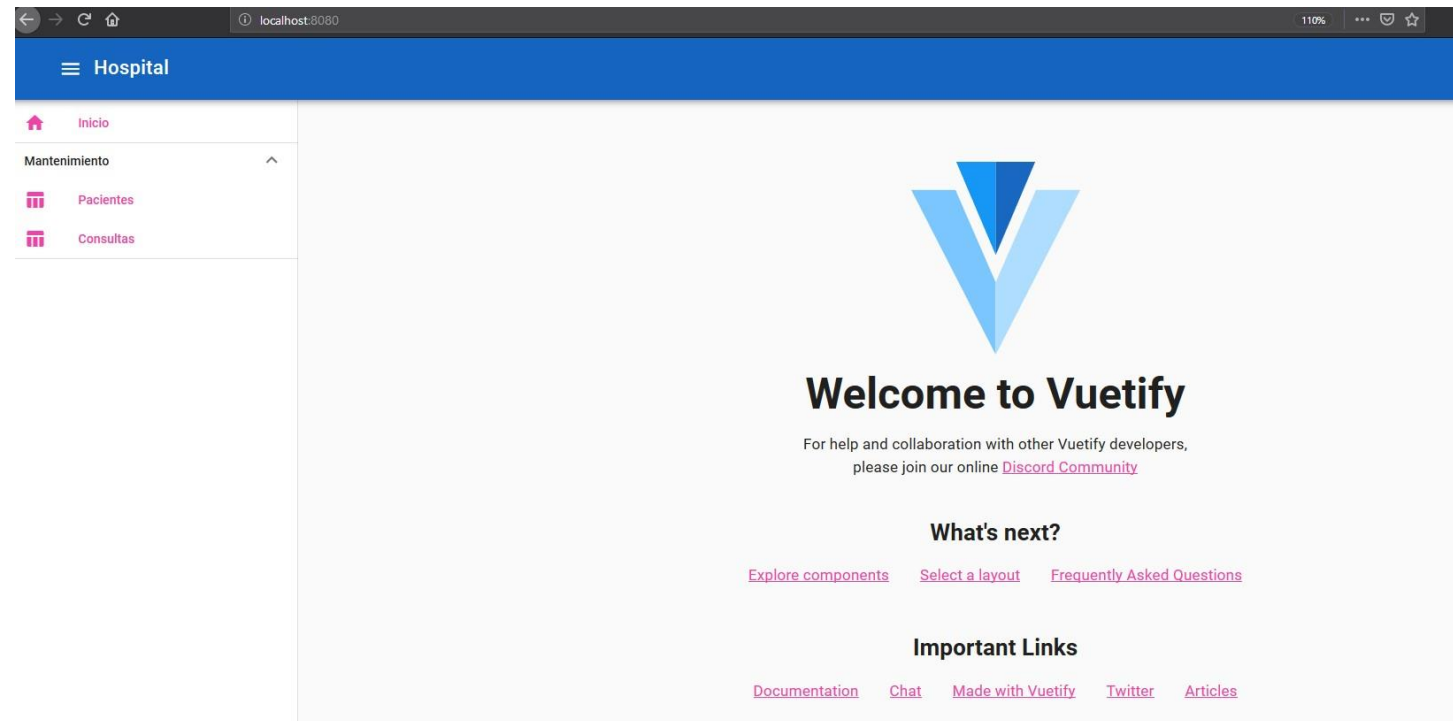
new Vue({
  router,
  store,
  render: h => h(App)
}).$mount('#app')
```

# Frontend

- Copia el contenido del archivo **App.txt** que se ubica en la carpeta **Archivo** y pega dicho contenido en el archivo **App.vue** del proyecto hospital.

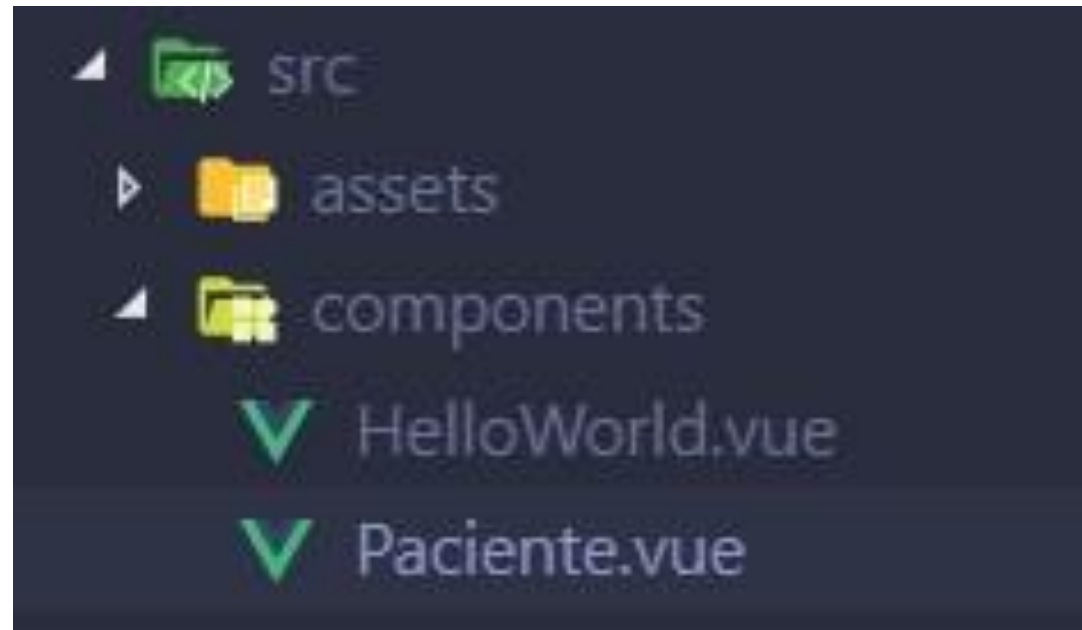


# Frontend



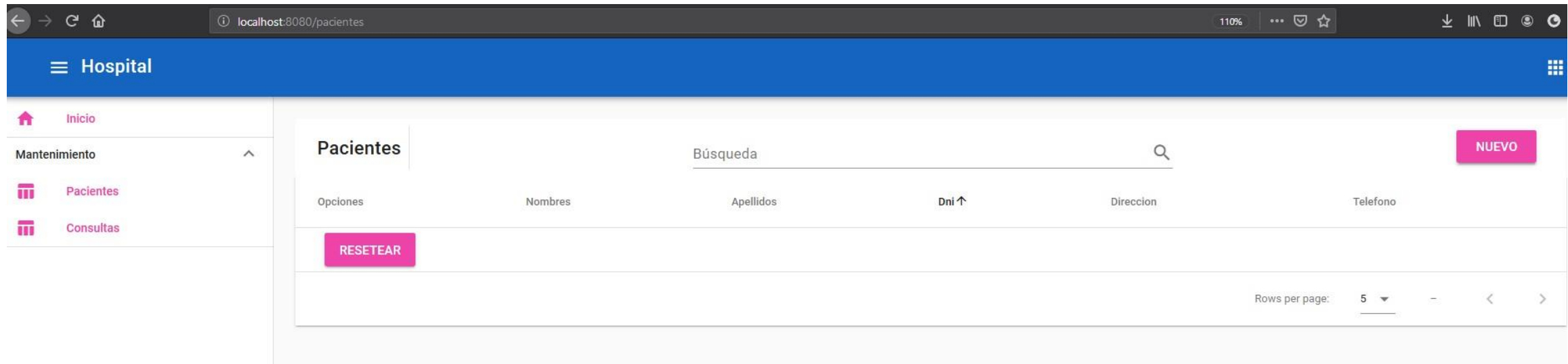
- Debe crear el archivo Paciente.vue en la carpeta src/components

# Frontend



- Copia el contenido del archivo **Paciente.txt** que se ubica en la carpeta **Archivo** y pega dicho contenido en el archivo **Paciente.vue** del proyecto hospital.

# Frontend



# Frontend

- Implementar Model
- Metodo Listar y Create
- Metodo editItem
- Metodo guardar

## COMPONENTS

```
<template>
  <v-layout align-start>
    <v-flex>
      <v-toolbar flat color="white">
        <v-toolbar-title>Pacientes</v-toolbar-title>
        <v-divider class="mx-2" inset vertical></v-divider>
        <v-spacer></v-spacer>
        <v-text-field
          class="text-xs-center"
          v-model="search"
          append-icon="search"
          label="Búsqueda"
          single-line
          hide-details
        ></v-text-field>
        <v-spacer></v-spacer>
        <v-dialog v-model="dialog" max-width="500px">
          <v-btn slot="activator" color="primary" dark class="mb-2">Nuevo</v-btn>
          <v-card>
            <v-card-title>
              <span class="headline">{{ formTitle }}</span>
            </v-card-title>

            <v-card-text>
```

```

    <v-container grid-list-md>
      <v-layout wrap>
        <v-flex xs12 sm12 md12>
          <v-text-field v-model="nombres" label="Nombres"></v-text-field>
        </v-flex>
        <v-flex xs12 sm12 md12>
          <v-text-field v-model="apellidos" label="Apellidos"></v-text-field>
        </v-flex>
        <v-flex xs12 sm12 md12>
          <v-text-field v-model="dni" label="Dni"></v-text-field>
        </v-flex>
        <v-flex xs12 sm12 md12>
          <v-text-field v-model="direccion" label="Direccion"></v-text-field>
        </v-flex>
        <v-flex xs12 sm12 md12>
          <v-text-field v-model="telefono" label="Telefono"></v-text-field>
        </v-flex>
      </v-layout>
    </v-container>
  </v-card-text>

  <v-card-actions>
    <v-spacer></v-spacer>
    <v-btn color="blue darken-1" flat @click.native="close">Cancelar</v-btn>
    <v-btn color="blue darken-1" flat @click.native="guardar">Guardar</v-btn>
  </v-card-actions>
</v-card>
</v-dialog>
</v-toolbar>
<v-data-table :headers="headers" :items="pacientes" :search="search" class="elevation-1">
  <template slot="items" slot-scope="props">

```

```

        <td class="justify-center layout px-0">
            <v-icon small class="mr-2" @click="editItem(props.item)">edit</v-icon>

        </td>
        <td>{{ props.item.nombres }}</td>
        <td>{{ props.item.apellidos }}</td>
        <td>{{ props.item.dni }}</td>
        <td>{{ props.item.direccion }}</td>
        <td>{{ props.item.telefono }}</td>
    </template>
    <template slot="no-data">
        <v-btn color="primary" @click="listar">Resetear</v-btn>
    </template>
</v-data-table>
</v-flex>
</v-layout>
</template>
<script>
import axios from "axios";
export default {
  data() {
    return {
      pacientes: [],
      dialog: false,
      headers: [
        { text: "Opciones", value: "opciones", sortable: false },
        { text: "Nombres", value: "nombres", sortable: false },
        { text: "Apellidos", value: "apellidos", sortable: false },
        { text: "Dni", value: "dni" },
        { text: "Direccion", value: "direccion" },
        { text: "Telefono", value: "telefono" }
      ]
    }
  }
}

```

```
    ],
    search: "",
    editedIndex: -1,

    //Model
    id: "",
    nombres: "",
    apellidos: "",
    dni: "",
    direccion: "",
    telefono: ""
  };
},
computed: {
  formTitle() {
    return this.editedIndex === -1 ? "Nuevo Paciente" : "Actualizar Paciente";
  }
},

watch: {
  dialog(val) {
    val || this.close();
  }
},

created() {
  this.listar();
},
methods: {
  listar() {
    let me = this;
```



```
    axios
      .get("api/paciente")
      .then(function(response) {
        //console.log(response);
        me.pacientes = response.data;
      })
      .catch(function(error) {
        console.log(error);
      });
  },
  editItem(item) {
    this.id = item.id;
    this.nombres = item.nombres;
    this.apellidos = item.apellidos;
    this.dni = item.dni;
    this.direccion = item.direccion;
    this.telefono = item.telefono;

    this.editedIndex = 1;
    this.dialog = true;
  },

  close() {
    this.dialog = false;
  },
  limpiar() {
    this.id = "";
    this.nombres = "";
    this.apellidos = "";
  }
}
```

```

    this.dni = "";
    this.direccion = "";
    this.telefono = "";
  },
  guardar() {
    if (this.editedIndex > -1) {
      //CÃ³digo para editar

      let me = this;
      axios
        .put("api/paciente", {
          id: me.id,
          nombres: me.nombres,
          apellidos: me.apellidos,
          dni: me.dni,
          direccion: me.direccion,
          telefono: me.telefono
        })
        .then(function(response) {
          me.close();
          me.listar();
          me.limpiar();
        })
        .catch(function(error) {
          console.log(error);
        });
    } else {
      //CÃ³digo para guardar
      let me = this;
      axios
        .post("api/paciente", {

```

```

        nombres: me.nombres,
        apellidos: me.apellidos,
        dni: me.dni,
        direccion: me.direccion,
        telefono: me.telefono
    })
    .then(function(response) {
        me.close();
        me.listar();
        me.limpiar();
    })
    .catch(function(error) {
        console.log(error);
    });
    }
}
};
</script>

```

APP

```

<template>
  <v-app id="app">
    <v-navigation-drawer
      fixed
      :clipped="$vuetify.breakpoint.mdAndUp"
      app
      v-model="drawer"
    >

```

```

<v-list dense>
  <template>
    <v-list-tile :to="{name:'home'}">
      <v-list-tile-action>
        <v-icon>home</v-icon>
      </v-list-tile-action>
      <v-list-tile-title>
        Inicio
      </v-list-tile-title>
    </v-list-tile>
  </template>

  <template>
    <v-list-group>
      <v-list-tile slot="activator">
        <v-list-tile-content>
          <v-list-tile-title>
            Mantenimiento
          </v-list-tile-title>
        </v-list-tile-content>
      </v-list-tile>
      <v-list-tile :to="{ name: 'pacientes'}">
        <v-list-tile-action>
          <v-icon>table_chart</v-icon>
        </v-list-tile-action>
        <v-list-tile-content>
          <v-list-tile-title>
            Pacientes
          </v-list-tile-title>
        </v-list-tile-content>
      </v-list-tile>
    </v-list-group>
  </template>

```

```

    <v-list-tile :to="{ name: ''}">
      <v-list-tile-action>
        <v-icon>table_chart</v-icon>
      </v-list-tile-action>
      <v-list-tile-content>
        <v-list-tile-title>
          Consultas
        </v-list-tile-title>
      </v-list-tile-content>
    </v-list-tile>

  </v-list-group>
</template>

```

```

</v-list>
</v-navigation-drawer>
<v-toolbar
  color="blue darken-3"
  dark
  app
  :clipped-left="$vuetify.breakpoint.mdAndUp"
  fixed
>
  <v-toolbar-title style="width: 300px" class="ml-0 pl-3">
    <v-toolbar-side-icon @click.stop="drawer = !drawer"></v-toolbar-side-icon>
    <span class="hidden-sm-and-down">Hospital</span>
  </v-toolbar-title>
  <v-spacer></v-spacer>
  <v-btn icon>

```

```

    <v-icon>apps</v-icon>
  </v-btn>
</v-toolbar>
<v-content>
  <v-container fluid fill-height>
    <v-slide-y-transition mode="out-in">
      <router-view/>
    </v-slide-y-transition>
  </v-container>
</v-content>
<v-footer dark height="auto">
  <v-layout justify-center>
    <v-flex text-xs-center>
      <v-card flat tile color="primary" class="white--text">
        <v-card-text class="white--text pt-0">
          HAMP &copy;2019
        </v-card-text>
      </v-card>
    </v-flex>
  </v-layout>
</v-footer>

</v-app>
</template>

```

```

<script>

```

```

export default {
  name: 'App',
  data () {

```

```
    return {  
      drawer: true,  
    }  
  }  
}  
</script>
```

MAIN

```
import Vue from 'vue'  
import './plugins/vuetify'  
import App from './App.vue'  
import router from './router'  
import store from './store'
```

```
import axios from 'axios'
```

```
Vue.config.productionTip = false  
axios.defaults.baseURL='http://localhost:5000/'  
new Vue({  
  router,
```

```
    store,  
    render: h => h(App)  
  }).$mount('#app')
```

ROUTER

```
import Vue from 'vue'  
import Router from 'vue-router'  
import Home from './views/Home.vue'  
import Paciente from './components/Paciente.vue'
```

```
Vue.use(Router)
```

```
export default new Router({  
  mode: 'history',  
  base: process.env.BASE_URL,  
  routes: [  
    {  
      path: '/',
```



```
    name: 'home',  
    component: Home  
  },  
  {  
    path: '/pacientes',  
    name: 'pacientes',  
    component: Paciente  
  }  
]  
}))
```