

Instituto Tecnológico de Costa Rica



Escuela de Ingeniería en Computación Introducción al Desarrollo de Páginas Web

Apunte de Vue.js

Profesora:

Ericka Solano Fernández

Estudiantes:

Arlem Gabriel Brenes Aguilar

Verano, 2020

Experiencia de Instalación

Para efectos de la instalación, el instructivo fue lo suficientemente claro para realizar cada uno de los pasos correspondientes, además de que la tecnología no ameritaba realizar exhaustivas configuraciones por tanto la experiencia fue excelente.

Aspectos Relevantes de la Tecnología

Typescript

Creado en el 2012 cuando microsoft estaba en lanzamiento de proyectos escalables, nace en el proyecto Strada que era para corregir los problemas que tenía javascript como por ejemplo que no contaba con objetos. Es un lenguaje de código abierto, desarrollado por microsoft, superconjunto de javascript, puede ser compilado

Ventajas: Muchas herramientas y gran comunidad, el código se hace más entendible, finalización de código intellisense, curva de aprendizaje liviana, detecta errores en tiempo de compilación

Desventajas: Complejidad en la configuración del proyecto y tarda tiempo en acostumbrarse al nuevo lenguaje.

Empresas que lo utilizan: BMW, Santander, Overleaf

Vue.js

Creado en el 2014 inicialmente pensado como biblioteca personal. Es open source, es accesible, es versátil y escalable porque usa plugins, cuenta con una gran comunidad, se logra encapsular el código reutilizable y cuenta con template.

Ventajas: Tiene gran similitud con Angular, usa componentes, tiene documentación detallada, es adaptable, excelente integración por su flexibilidad, es gran escalado sirve para trabajos pequeños y grandes

Desventajas: Falta de recursos, riesgo de excesiva flexibilidad para proyectos grandes

Plugins: Vue router, vue-axios, test, vuex

Descripción del Ejercicio

Para efectos del taller el ejercicio consistió en realizar una aplicación sobre tareas por hacer donde se cuenta con dos secciones, la primera sección es acerca del taller donde se muestra la información y la segunda sección consiste de un listado de tareas que pueden crearse, eliminarse, marcarse como pendiente, limpiar las completas.

Evidencias Visuales

Código:

- App.Vue:

```
<template>
  <div>
    <nav class="navbar navbar-expand-lg navbar-dark bg-info">
      <a class="navbar-brand" href="#">
        
      </a>
      <button
        class="navbar-toggler"
        type="button"
        data-toggle="collapse"
        data-target="#navbarNav"
        aria-controls="navbarNav"
        aria-expanded="false"
        aria-label="Toggle navigation"
      >
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item" :class="{active: pagina == 'acercaDe'}"
@click="pagina='acercaDe'">
            <router-link class="nav-link" to="/">Acerca de este
proyecto</router-link>
          </li>
          <li class="nav-item" :class="{active: pagina == 'porHacer'}"
@click="pagina='porHacer'">
            <router-link class="nav-link" to="/porHacer">Por
hacer</router-link>
          </li>
        </ul>
      </div>
    </nav>
    <div class="container">
      <router-view />
    </div>
  </div>
</template>
```

```

</template>

<style>
</style>

<script lang="ts">
export default {
  data() {
    return {
      pagina: "acercaDe"
    };
  }
};
</script>

```

- **AcercaDe.vue:**

```

<template>
  <div class="contenedor">
    <h1>Taller de Vue.js con TypeScript</h1>
    <p>
      Este taller fue realizado para proponer un ejercicio para ser
      desarrollado en clase, de manera que se muestren las funcionalidades que
      propone el framework Vue.js utilizando TypeScript como lenguaje de
      programación. La aplicación que se desarrolla en este taller consiste en
      un
      sistema para administrar tareas o actividades, donde se incluyen todas
      las
      operaciones de leer, crear, modificar y eliminar para los ítemes de
      tareas.
    </p>
  </div>
</template>
<style scoped>
@font-face {
  font-family: sketch;
  src: url(../assets/Sketch_3D.otf);
}
.contenedor {
  display: flex;
  flex-direction: column;

```

```

    justify-content: center;
    align-items: center;
    height: 60vh;
    padding: 0 100px;
  }
  h1 {
    font-family: sketch;
    margin-bottom: 50px;
    font-size: 70px;
    text-align: center;
  }
</style>

```

- **PorHacer.vue:**

```

<template>
  <div class="app">
    <h1 class="titulo">Tareas por hacer</h1>
    <div class="contenedor-principal">
      <input
        v-model="newTodo"
        type="text"
        class="todo-input"
        placeholder="Nueva tarea..."
        @keyup.enter="addTodo"
      />
      <transition-group
        enter-active-class="animated fadeInUp"
        leave-active-class="animated fadeOutDown"
      >
        <div v-for="(todo, index) in todosFiltered" :key="todo.id"
class="todo-item">
          <div class="todo-item-left">
            <input type="checkbox" v-model="todo.completed" />
            <div
              v-if="!todo.editing"
              @dblclick="editTodo(todo)"
              class="todo-item-label"
              :class="{completed : todo.completed}"
            >{{todo.title}}</div>

```

```

        <input
          v-else
          class="todo-item-edit"
          type="text"
          v-model="todo.title"
          @blur="doneEdit(todo)"
          @keyup.enter="doneEdit(todo)"
          v-focus
          @keyup.esc="cancelEdit(todo)"
        />
      </div>
      <div class="remove-item" @click="removeTodo(index)">
        <i class="fas fa-trash-alt"></i>
      </div>
    </div>
  </transition-group>
  <div class="extra-container">
    <div>
      <label>
        <input v-model="chk" type="checkbox" :checked="!anyRemaining"
@change="checkAllTodos" />
        <span class="label-text">Marcar todos como terminados</span>
      </label>
    </div>
    <div>{{ remaining }} pendientes</div>
  </div>
  <div class="extra-container">
    <div>
      <button :class="{active: filter == 'all'}" @click="filter =
='all'">Todos</button>
      <button :class="{active: filter == 'active'}" @click="filter =
'active'">Pendientes</button>
      <button :class="{active: filter == 'completed'}" @click="filter =
'completed'">Terminados</button>
    </div>
    <div>
      <transition name="fade">
        <button v-if="showClearCompletedButton"
@click="clearCompleted">Clear completed</button>
      </transition>
    </div>
  </div>

```

```

        </div>
      </div>
    </div>
  </div>
</template>

<script lang="ts">
import Vue from "vue";
export default Vue.extend({
  name: "todo-list",
  data() {
    return {
      newTodo: "",
      idForTodo: 3,
      beforeEditCache: "",
      filter: "all",
      chk: false,
      todos: [
        {
          id: 1,
          title: "Terminar el laboratorio 2",
          completed: false,
          editing: false
        },
        {
          id: 2,
          title: "Ponerle con el proyecto final! :0",
          completed: false,
          editing: false
        }
      ] as any
    };
  },
  directives: {
    focus: {
      inserted: function(el) {
        el.focus();
      }
    }
  },

```

```

methods: {
  addTodo() {
    if (this.newTodo.trim().length == 0) {
      return;
    }
    this.todos.push({
      id: this.idForTodo,
      title: this.newTodo,
      completed: false
    });
    this.newTodo = "";
    this.idForTodo++;
  },
  removeTodo(index: number) {
    this.todos.splice(index, 1);
  },
  editTodo(todo: any) {
    this.beforeEditCache = todo.title;
    todo.editing = true;
  },
  doneEdit(todo: any) {
    todo.editing = false;
    if (todo.title.trim().length == 0) {
      todo.title = this.beforeEditCache;
    }
  },
  cancelEdit(todo: any) {
    todo.title = this.beforeEditCache;
    todo.editing = false;
  },
  checkAllTodos() {
    var to: any = this.todos;
    var chk_value = this.chk;
    to.forEach((todo: any) => (todo.completed = chk_value));
  },
  clearCompleted() {
    this.todos = this.todos.filter((todo: any) => !todo.completed);
  }
},
computed: {

```



```

    todosFiltered() {
        var to: any = this.todos;
        if (this.filter == "all") {
            return to;
        } else if (this.filter == "active") {
            return to.filter((todo: any) => !todo.completed);
        } else if (this.filter == "completed") {
            return to.filter((todo: any) => todo.completed);
        }
        return to;
    },
    remaining() {
        var to: any = this.todos;
        return to.filter((todo: any) => !todo.completed).length;
    },
    anyRemaining() {
        var rem: any = this.remaining;
        if (rem == 0) {
            this.chk = true;
        } else {
            this.chk = false;
        }
        return rem != 0;
    },
    showClearCompletedButton() {
        var to: any = this.todos;
        return to.filter((todo: any) => todo.completed).length > 0;
    }
}
});
</script>

<style scoped>
@import
url("https://cdnjs.cloudflare.com/ajax/libs/animate.css/3.7.2/animate.min.css"
);
@font-face {
    font-family: sketch;
    src: url(../assets/Sketch_3D.otf);
}

```

```
.app {
  margin-top: 60px;
}

.titulo {
  text-align: center;
  margin-bottom: 50px;
  font-family: sketch;
  font-size: 70px;
}

.contenedor-principal {
  width: 80%;
  margin: 0 auto;
}

.todo-input {
  width: 100%;
  padding: 10px 18px;
  font-size: 18px;
  margin-bottom: 16px;
  border: 1px solid #aaa;
  border-radius: 8px;
}

.todo-input:focus {
  outline: 0;
  border: 1px solid #17a2b8;
}

.todo-item {
  margin-bottom: 12px;
  display: flex;
  align-items: center;
  justify-content: space-between;
  animation-duration: 0.3s;
  padding-right: 20px;
  padding-left: 10px;
  border-radius: 8px;
  background-color: #e9e9e9;
}

.todo-item:hover {
  -webkit-box-shadow: 6px 6px 18px -8px rgba(0, 0, 0, 0.75);
  -moz-box-shadow: 6px 6px 18px -8px rgba(0, 0, 0, 0.75);
  box-shadow: 6px 6px 18px -8px rgba(0, 0, 0, 0.75);
}
```

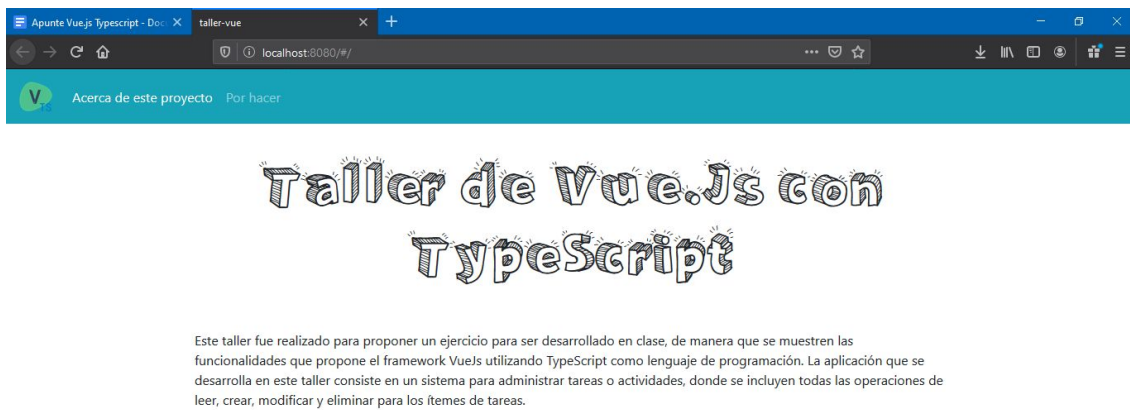
```
}  
.remove-item {  
  cursor: pointer;  
  margin-left: 14px;  
  color: #dc3545;  
}  
.remove-item:hover {  
  color: #b52130;  
}  
.todo-item-left {  
  display: flex;  
  align-items: center;  
  width: 100%;  
}  
.todo-item-label {  
  padding: 10px;  
  margin-left: 12px;  
}  
.todo-item-edit {  
  font-size: 24px;  
  color: #2c3e50;  
  margin-left: 12px;  
  width: 100%;  
  padding: 10px;  
  border: 1px solid #e9e9e9;  
  background-color: #e9e9e9;  
  font-family: "Avenir", Helvetica, Arial, sans-serif;  
}  
.todo-item-edit:focus {  
  outline: none;  
}  
.label-text {  
  margin-left: 20px;  
}  
.label-text:hover {  
  color: #168530;  
  font-weight: bold;  
}  
label {  
  padding-left: 10px;
```

```
}  
.completed {  
  text-decoration: line-through;  
  color: grey;  
}  
.extra-container {  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  font-size: 16px;  
  border-top: 1px solid lightgrey;  
  padding-top: 14px;  
  margin-bottom: 14px;  
}  
button {  
  font-size: 14px;  
  background-color: white;  
  border: 1px solid #aaa;  
  padding: 0.5em;  
  margin-right: 5px;  
  border-radius: 8px;  
}  
button:hover {  
  background: #17a2b8;  
  color: #f2f2f2;  
  border-color: #17a2b8;  
}  
button:focus {  
  outline: none;  
}  
.active {  
  border-color: #17a2b8;  
  border-width: 2px;  
}  
.fade-enter-active,  
.fade-leave-active {  
  transition: opacity 0.2s;  
}  
.fade-enter,  
.fade-leave-to {
```

```
opacity: 0;
}
</style>
```

Resultado Obtenido:

- **Pantalla Acerca del Taller**



- **Pantalla de Administración de Tareas**

