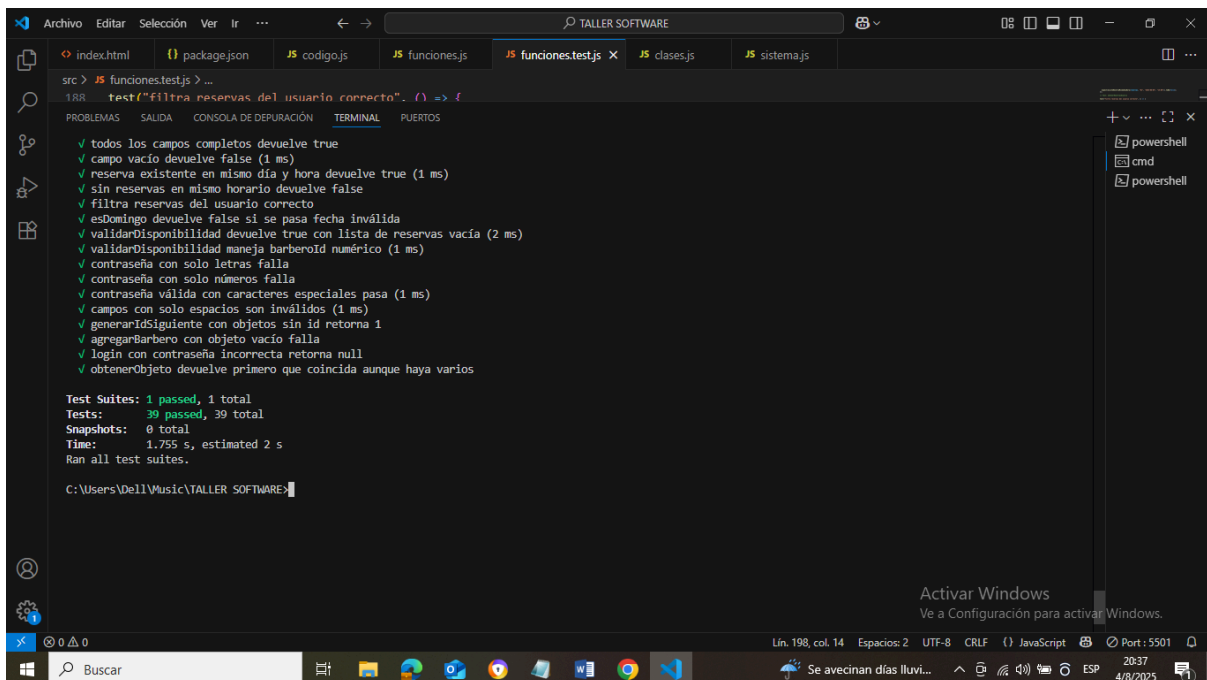


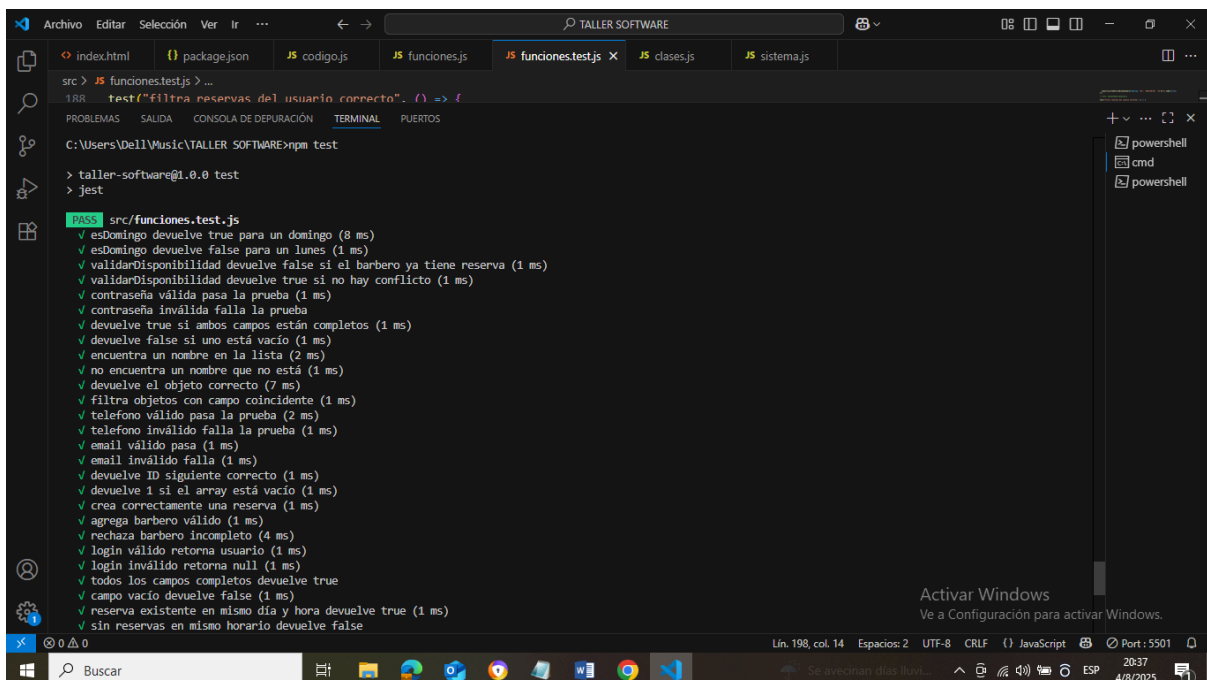
Entrega 3 - Pablo Larnaudie Natalia Rebella

Evidencias Testing Unitario



This screenshot shows the VS Code interface with the 'funciones.test.js' file open. The terminal displays the output of a Jest test run, showing 39 tests passed. The tests cover various scenarios for a reservation system, including validation of dates, user information, and reservation details.

```
src > JS funciones.test.js > ...  
1RR test(/"filtra reservas del usuario correcto". () => {  
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS  
✓ todos los campos completos devuelve true  
✓ campo vacío devuelve false (1 ms)  
✓ reserva existente en mismo día y hora devuelve true (1 ms)  
✓ sin reservas en mismo horario devuelve false  
✓ filtra reservas del usuario correcto  
✓ esDomingo devuelve false si se pasa fecha inválida  
✓ validarDisponibilidad devuelve true con lista de reservas vacía (2 ms)  
✓ validarDisponibilidad maneja barberoid numérico (1 ms)  
✓ contraseña con solo letras falla  
✓ contraseña con solo números falla  
✓ contraseña válida con caracteres especiales pasa (1 ms)  
✓ campos con solo espacios son inválidos (1 ms)  
✓ generarIdSiguiente con objetos sin id retorna 1  
✓ agregarBarbero con objeto vacío falla  
✓ login con contraseña incorrecta retorna null  
✓ obtenerObjeto devuelve primero que coincida aunque haya varios  
Test Suites: 1 passed, 1 total  
Tests: 39 passed, 39 total  
Snapshots: 0 total  
Time: 1.755 s, estimated 2 s  
Ran all test suites.  
C:\Users\Dell\Music\TALLER SOFTWARE>
```



This screenshot shows the VS Code interface after running the command 'npm test'. The terminal displays the output of the test run, showing 39 tests passed. The tests cover various scenarios for a reservation system, including validation of dates, user information, and reservation details.

```
C:\Users\Dell\Music\TALLER SOFTWARE>npm test  
> taller-software@1.0.0 test  
> jest  
PASS src/funciones.test.js  
✓ esDomingo devuelve true para un domingo (8 ms)  
✓ esDomingo devuelve false para un lunes (1 ms)  
✓ validarDisponibilidad devuelve false si el barbero ya tiene reserva (1 ms)  
✓ validarDisponibilidad devuelve true si no hay conflicto (1 ms)  
✓ contraseña válida pasa la prueba (1 ms)  
✓ contraseña inválida falla la prueba  
✓ devuelve true si ambos campos están completos (1 ms)  
✓ devuelve false si uno está vacío (1 ms)  
✓ encuentra un nombre en la lista (2 ms)  
✓ no encuentra un nombre que no está (1 ms)  
✓ devuelve el objeto correcto (7 ms)  
✓ filtra objetos con campo coincidente (1 ms)  
✓ telefono válido pasa la prueba (2 ms)  
✓ telefono inválido falla la prueba (1 ms)  
✓ email válido pasa (1 ms)  
✓ email inválido falla (1 ms)  
✓ devuelve ID siguiente correcto (1 ms)  
✓ devuelve 1 si el array está vacío (1 ms)  
✓ crea correctamente una reserva (1 ms)  
✓ agrega barbero válido (1 ms)  
✓ rechaza barbero incompleto (4 ms)  
✓ login válido retorna usuario (1 ms)  
✓ login inválido retorna null (1 ms)  
✓ todos los campos completos devuelve true  
✓ campo vacío devuelve false (1 ms)  
✓ reserva existente en mismo día y hora devuelve true (1 ms)  
✓ sin reservas en mismo horario devuelve false
```

This screenshot shows the VS Code editor with the file 'funciones.test.js' open. The code contains several Jest tests for a barber shop system. The tests are as follows:

- Test: validarDisponibilidad**
 - Test 1: 'validarDisponibilidad devuelve false si el barbero ya tiene reserva'. It sets a reservation for a barber at 10:00 on 2025-08-04 and expects the function to return false.
 - Test 2: 'validarDisponibilidad devuelve true si no hay conflicto'. It sets a reservation for a barber at 11:00 on 2025-08-04 and expects the function to return true.
- Test: verificarFormatoContraseña**
 - Test 1: 'contraseña válida pasa la prueba'. It checks if 'Abc123' is a valid password format and expects true.
 - Test 2: 'contraseña inválida falla la prueba'. It checks if 'abc123' is a valid password format and expects false.
- Test: validarCamposVaciosRegistro**
 - Test 1: 'devuelve true si ambos campos están completos'. It checks if 'Juan' and 'Clave123' are valid registration fields and expects true.

The status bar at the bottom indicates 'Lín. 198, col. 14', 'Espacios: 2', 'UTF-8', 'CRLF', 'JavaScript', and 'Port: 5501'. A Windows activation watermark is visible in the bottom right corner.

This screenshot shows the VS Code editor with the file 'funciones.test.js' open, displaying the continuation of the test suite. The tests are as follows:

- Test: validarCamposVaciosRegistro**
 - Test 2: 'devuelve false si uno está vacío'. It checks if an empty string and 'Clave123' are valid registration fields and expects false.
- Test: buscarNombre**
 - Test 1: 'encuentra un nombre en la lista'. It sets a list with 'Ana' and expects the function to find 'Ana' and return true.
 - Test 2: 'no encuentra un nombre que no está'. It sets a list with 'Ana' and expects the function to not find 'Pedro' and return false.
- Test: obtenerObjeto**
 - Test 1: 'devuelve el objeto correcto'. It sets a list with two objects (id: 1 and id: 2) and expects the function to return the object with id: 2.
- Test: obtenerObjetoEnArray**

The status bar at the bottom indicates 'Lín. 198, col. 14', 'Espacios: 2', 'UTF-8', 'CRLF', 'JavaScript', and 'Port: 5501'. A Windows activation watermark is visible in the bottom right corner.

Archivo Editar Selección Ver Ir ... TALLER SOFTWARE

index.html package.json JS codigojs JS funciones.js JS funciones.test.js JS clases.js JS sistemajs

```
src > JS funciones.test.js > ...
1  const {
2    esDomingo,
3    validarDisponibilidad,
4    verificarFormatoContraseña,
5    validarCamposVaciosRegistro,
6    buscarNombre,
7    obtenerObjeto,
8    obtenerObjetoEnArray,
9    validarTelefono,
10   validarEmail,
11   generarIdSiguiente,
12   crearReserva,
13   agregarBarbero,
14   login,
15   camposFormularioCompletos,
16   existeReservaMismoDiaHora,
17   obtenerReservasUsuario
18 } = require("../src/funciones");
19
20 // Test: esDomingo
21
22 test("esDomingo devuelve true para un domingo", () => {
23   expect(esDomingo("2025-08-03")).toBe(true); // domingo
24 });
25
26 test("esDomingo devuelve false para un lunes", () => {
27   expect(esDomingo("2025-08-04")).toBe(false); // lunes
28 });
29
30 // Test: validarDisponibilidad
31
32 test("validarDisponibilidad devuelve false si el barbero ya tiene reserva", () => {
```

Activar Windows
Ve a Configuración para activar Windows.

Lín. 198, col. 14 Espacios: 2 UTF-8 CRLF () JavaScript Port: 5501

Se acercan días lluvi...

20:37
4/8/2025

Archivo Editar Selección Ver Ir ... TALLER SOFTWARE

index.html package.json JS codigojs JS funciones.js JS funciones.test.js JS clases.js JS sistemajs

```
src > JS funciones.test.js > ...
85 // Test: obtenerObjetoEnArray
86
87 test("filtra objetos con campo coincidente", () => {
88   const lista = [{ id: 1 }, { id: 2 }, { id: 1 }];
89   expect(obtenerObjetoEnArray(lista, "id", 1)).toHaveLength(2);
90 });
91
92 // Test: validarTelefono
93
94 test("telefono válido pasa la prueba", () => {
95   expect(validarTelefono("091234567")).toBe(true);
96 });
97
98 test("telefono inválido falla la prueba", () => {
99   expect(validarTelefono("081234567")).toBe(false);
100 });
101
102 // Test: validarEmail
103
104 test("email válido pasa", () => {
105   expect(validarEmail("test@example.com")).toBe(true);
106 });
107
108 test("email inválido falla", () => {
109   expect(validarEmail("test@com")).toBe(false);
110 });
111
112 // Test: generarIdSiguiente
113
114 test("devuelve ID siguiente correcto", () => {
115   expect(generarIdSiguiente([{ id: 1 }, { id: 5 }])).toBe(6);
116 });
```

Activar Windows
Ve a Configuración para activar Windows.

Lín. 198, col. 14 Espacios: 2 UTF-8 CRLF () JavaScript Port: 5501

20:38
4/8/2025


```
Archivo Editar Selección Ver Ir ... TALLER SOFTWARE
index.html package.json JS codigojs JS funciones.js JS funciones.testjs JS clases.js JS sistemajs
src > JS funciones.testjs > ...
158 // Test: camposFormularioCompletos
159
160 test("todos los campos completos devuelve true", () => {
161   const campos = { a: "uno", b: "dos" };
162   expect(camposFormularioCompletos(campos)).toBe(true);
163 });
164
165 test("campo vacío devuelve false", () => {
166   const campos = { a: "", b: "dos" };
167   expect(camposFormularioCompletos(campos)).toBe(false);
168 });
169
170 // Test: existeReservaMismoDiaHora
171
172 test("reserva existente en mismo día y hora devuelve true", () => {
173   const reservas = [
174     { barbero: { id: "B1" }, fecha: "2025-08-05", hora: "15:00" }
175   ];
176   expect(existeReservaMismoDiaHora(reservas, "B1", "2025-08-05", "15:00")).toBe(true);
177 });
178
179 test("sin reservas en mismo horario devuelve false", () => {
180   const reservas = [
181     { barbero: { id: "B1" }, fecha: "2025-08-05", hora: "14:00" }
182   ];
183   expect(existeReservaMismoDiaHora(reservas, "B1", "2025-08-05", "15:00")).toBe(false);
184 });
185
186 // Test: obtenerReservasUsuario
187
188 test("filtra reservas del usuario correcto", () => {
189   const reservas = [
190     { idCliente: 1 },
191     { idCliente: 2 },
192     { idCliente: 1 }
193   ];
194   const result = obtenerReservasUsuario(reservas, 1);
195   expect(result).toHaveLength(2);
196 });
197
198 //CASOS BORDE
199
200 // esDomingo con fecha inválida
201 test("esDomingo devuelve false si se pasa fecha inválida", () => {
202   expect(esDomingo("fecha_invalida")).toBe(false); // podría fallar, depende de tu validación
203 });
204
205 // validarDisponibilidad con reservas vacías
206 test("validarDisponibilidad devuelve true con lista de reservas vacía", () => {
207   const reservas = [];
208   expect(validarDisponibilidad("B1", "2025-08-05", "10:00", reservas)).toBe(true);
209 });
210
211 // validarDisponibilidad con id numérico en vez de string
212 test("validarDisponibilidad maneja barberoId numérico", () => {
213   const reservas = [
214     { barbero: { id: 1 }, fecha: "2025-08-05", hora: "10:00" }
215   ];
216   expect(validarDisponibilidad(2, "2025-08-05", "10:00", reservas)).toBe(true);
217 });
```

Activar Windows
Ve a Configuración para activar Windows.

Lín. 198, col. 14 Espacios: 2 UTF-8 CRLF () JavaScript Port: 5501

Se acercan días lluvi...

20:39 4/8/2025

```
Archivo Editar Selección Ver Ir ... TALLER SOFTWARE
index.html package.json JS codigojs JS funciones.js JS funciones.testjs JS clases.js JS sistemajs
src > JS funciones.testjs > ...
186 // Test: obtenerReservasUsuario
187
188 test("filtra reservas del usuario correcto", () => {
189   const reservas = [
190     { idCliente: 1 },
191     { idCliente: 2 },
192     { idCliente: 1 }
193   ];
194   const result = obtenerReservasUsuario(reservas, 1);
195   expect(result).toHaveLength(2);
196 });
197
198 //CASOS BORDE
199
200 // esDomingo con fecha inválida
201 test("esDomingo devuelve false si se pasa fecha inválida", () => {
202   expect(esDomingo("fecha_invalida")).toBe(false); // podría fallar, depende de tu validación
203 });
204
205 // validarDisponibilidad con reservas vacías
206 test("validarDisponibilidad devuelve true con lista de reservas vacía", () => {
207   const reservas = [];
208   expect(validarDisponibilidad("B1", "2025-08-05", "10:00", reservas)).toBe(true);
209 });
210
211 // validarDisponibilidad con id numérico en vez de string
212 test("validarDisponibilidad maneja barberoId numérico", () => {
213   const reservas = [
214     { barbero: { id: 1 }, fecha: "2025-08-05", hora: "10:00" }
215   ];
216   expect(validarDisponibilidad(2, "2025-08-05", "10:00", reservas)).toBe(true);
217 });
```

Activar Windows
Ve a Configuración para activar Windows.

Lín. 198, col. 14 Espacios: 2 UTF-8 CRLF () JavaScript Port: 5501

Se acercan días lluvi...

20:39 4/8/2025

This screenshot shows the VS Code editor with the file 'funciones.test.js' open. The code contains several Jest tests for a system. The tests include:

- `validarDisponibilidad`: Tests that a reservation is valid based on barber ID, date, and time.
- `verificarFormatoContraseña`: Tests password validation rules (letters only, numbers only, and special characters).
- `validarCamposVaciosRegistro`: Tests that registration fields are not empty.
- `generarIdSiguiente`: Tests that the next ID is generated correctly from a list of objects.

```
210
211 // validarDisponibilidad con id numérico en vez de string
212 test('validarDisponibilidad maneja barberoId numérico', () => {
213   const reservas = [
214     { barbero: { id: 1 }, fecha: "2025-08-05", hora: "10:00" }
215   ];
216   expect(validarDisponibilidad(2, "2025-08-05", "10:00", reservas)).toBe(true);
217 });
218
219 // verificarFormatoContraseña con solo letras
220 test('contraseña con solo letras falla', () => {
221   expect(verificarFormatoContraseña("SoloLetras")).toBe(false);
222 });
223
224 // verificarFormatoContraseña con solo números
225 test('contraseña con solo números falla', () => {
226   expect(verificarFormatoContraseña("12345678")).toBe(false);
227 });
228
229 // verificarFormatoContraseña con caracteres especiales
230 test('contraseña válida con caracteres especiales pasa', () => {
231   expect(verificarFormatoContraseña("Abc123$%")).toBe(true);
232 });
233
234 // validarCamposVaciosRegistro con espacios
235 test('campos con solo espacios son inválidos', () => {
236   expect(validarCamposVaciosRegistro(" ", " ")).toBe(false);
237 });
238
239 // generarIdSiguiente con objetos sin id
240 test('generarIdSiguiente con objetos sin id retorna 1', () => {
241   const lista = [{}, {}, {}];
242   expect(generarIdSiguiente(lista)).toBe(1);
243 });
```

The status bar at the bottom indicates 'Lín. 198, col. 14', 'Espacios: 2', 'UTF-8', 'CRLF', 'JavaScript', and 'Port: 5501'.

This screenshot shows the VS Code editor with the file 'funciones.test.js' open, displaying the continuation of the test suite. The tests include:

- `agregarBarbero`: Tests that adding a barber with empty fields fails.
- `login`: Tests that login fails with incorrect password.
- `obtenerObjeto`: Tests that the first object is returned from a list when searching by name.

```
238
239 // generarIdSiguiente con objetos sin id
240 test('generarIdSiguiente con objetos sin id retorna 1', () => {
241   const lista = [{}, {}, {}];
242   expect(generarIdSiguiente(lista)).toBe(1);
243 });
244
245 // agregarBarbero con campos vacíos
246 test('agregarBarbero con objeto vacío falla', () => {
247   const barberos = [];
248   expect(agregarBarbero(barberos, {})).toBe(false);
249 });
250
251 // login con nombre correcto pero contraseña incorrecta
252 test('login con contraseña incorrecta retorna null', () => {
253   const usuarios = [{ id: 1, nombre: "juan", contraseña: "Abc123", email: "", telefono: "" }];
254   expect(login("juan", "wrong", "user", usuarios, [])).toBe(null);
255 });
256
257 // obtenerObjeto cuando hay varios con mismo campo
258 test('obtenerObjeto devuelve primero que coincida aunque haya varios', () => {
259   const lista = [
260     { nombre: "natalia", id: 1 },
261     { nombre: "natalia", id: 2 }
262   ];
263   const res = obtenerObjeto(lista, "nombre", "natalia");
264   expect(res.id).toBe(1); // debe devolver el primero
265 });
```

The status bar at the bottom indicates 'Lín. 198, col. 14', 'Espacios: 2', 'UTF-8', 'CRLF', 'JavaScript', and 'Port: 5501'.

Estado Del Sprint

