Registro Diario de Avances

**Fecha:** 01/09/2025

**Autor:**

**Danluis Enrique Romani Mora**

**Eduardo Chaparro Huaman**

**Fabrizcio Argandoña Montalvo**

**Versión:** 1.0

# 1. Introducción

En este documento se muestra el avance de tareas respectivas al Sprint 3 con lo que también se detallan las historias de usuarios pertenecientes y el esfuerzo realizado para completar cada una de ellas, con evidencias del código utilizado, burndown chart y el análisis respectivo al rendimiento del equipo.

# 2. Lineamientos de Diseño

Desarrollar una tercera versión funcional de la aplicación que permita a los usuarios:

* Inicio de sesión y acceso de los alumnos
* Mostrar mensajes en caso de error al momento de iniciar sesión.
* La posibilidad de recuperar la contraseña.
* La visualización del test de inteligencias múl tiples.

# 3. Historias de Usuario y Tareas

| Backlog ID | Historia de Usuario | Tiempo Estimado | Fecha de Inicio | Fecha de Finalización |
| --- | --- | --- | --- | --- |
| HU-1.4 | Acceso de alumnos al sistema | 10 | 1/10/2025 | 3/10/2025 |
| HU-1.5 | Mensaje de error por credenciales incorrectas | 4 | 1/10/2025 | /10/2025 |
| HU-1.6 | Recuperación de contraseña para alumnos | 8 | 1/10/2025 | /10/2025 |
| HU-1.12 | Confirmación visual de cierre de sesión | 3 | /10/2025 | /10/2025 |
| HU-2.1 | Visualización del test en la práctica | 14 | /10/2025 | /10/2025 |

# 4. Evidencias de Código

## 4.1 Historia de Usuario 1.4: Acceso de alumnos al sistema

Archivo Frontend : Login.component.html

<body>

<div class="overlay"></div>

<header>

<h1>INICIO DE SESIÓN</h1>

<h2>Ingrese sus datos</h2>

<div class="logoCole"></div>

</header>

<div class="login-container" [ngClass]="{'fade-mode': animar}">

<div style="display: flex; justify-content: space-between; gap: 10px;">

<button class="crear-button"

[ngClass]="{'active': tipoLogin === 'docente'}"

(click)="setLogin('docente')">Docentes</button>

<button class="crear-button"

[ngClass]="{'active': tipoLogin === 'alumno'}"

(click)="setLogin('alumno')">Alumnos</button>

<button class="crear-button"

[ngClass]="{'active': tipoLogin === 'admin'}"

(click)="setLogin('admin')">Administrador</button>

</div>

<input type="text" class="input-field" placeholder="Usuario" [(ngModel)]="usuario">

<input type="password" class="input-field" placeholder="Contraseña" [(ngModel)]="contrasena">

<button class="login-button" (click)="login()">Ingresar</button>

<p class="recuperar-texto" (click)="recuperar()">¿Olvidaste tu contraseña?</p>

</div>

</body>

Archivo Frontend : Login.component.ts

import { Component } from '@angular/core';

import { Router } from '@angular/router';

import { CommonModule } from '@angular/common';

import { HttpClientModule } from '@angular/common/http';

import { FormsModule } from '@angular/forms';

import { DocentesService } from '../services/docentes.service';

import Swal from 'sweetalert2';

@Component({

selector: 'app-login',

standalone: true,

imports: [CommonModule, HttpClientModule, FormsModule],

templateUrl: './login.component.html',

styleUrl: './login.component.css'

})

export class LoginComponent {

usuario: string = '';

contrasena: string = '';

tipoLogin: string = 'docente'; *// por defecto*

animar: boolean = false;

exito: boolean = false;

constructor(*private* router: Router, *private* docentesService: DocentesService) {}

setLogin(tipo: string) {

this.animar = true; *// activa animación*

setTimeout(() => {

this.tipoLogin = tipo;

this.animar = false; *// termina animación*

}, 300);

}

login() {

if (this.tipoLogin === 'docente') {

*// lógica para docentes*

if (this.usuario =='docente1' && this.contrasena == '1234') {

this.exito = true;

this.router.navigate(['inicio-docentes']);

}

} else if (this.tipoLogin === 'alumno') {

*// lógica para alumnos*

if (this.usuario =='alumno1' && this.contrasena == '1234') {

this.exito = true;

this.router.navigate(['inicio-alumnos']);

}

} else if (this.tipoLogin === 'admin') {

*// lógica para administrador*

if (this.usuario =='admin1' && this.contrasena == '1234') {

this.exito = true;

this.router.navigate(['inicio']);

}

}

if(this.exito==true)

{

Swal.fire({

title: 'Ingreso exitoso',

text: `Ingreso como ${this.tipoLogin}`,

icon: 'success',

confirmButtonText: 'Aceptar',

});

}

else

{

Swal.fire({

title: 'Error',

html: 'Usuario o contraseña incorrectos.<br>Estas intentando ingresar como <b>' + this.tipoLogin + '</b>' + '<br>Si no es correcto cambie de modo de ingreso.',

icon: 'error',

confirmButtonText: 'Aceptar',

});

}

}

recuperar() {

this.router.navigate(['/recuperar-contrasena']);

}

}

Backend donde se recibe las credenciales del alumno, verifica el rate limit y debería autenticar al usuario

@PostMapping("/login")

public ResponseEntity<Map<String, Object>> login(@Valid @RequestBody LoginRequest loginRequest, HttpServletRequest request) {

String clientIP = getClientIP(request);

Map<String, Object> response = new HashMap<>();

*// Rate limiting*

if (!rateLimitService.isLoginAllowed(clientIP) || !rateLimitService.isLoginAllowedByUser(loginRequest.getUsuario())) {

response.put("exito", false);

response.put("mensaje", "Demasiados intentos de login. Intente más tarde.");

auditoriaService.registrarAccion("LOGIN\_BLOQUEADO", "USUARIO", null, "Rate limit excedido", clientIP, loginRequest.getUsuario());

return ResponseEntity.status(429).body(response);

}

try {

response.put("exito", false);

response.put("mensaje", "Funcionalidad de autenticación no implementada.");

return ResponseEntity.status(501).body(response);

} catch (Exception e) {

auditoriaService.registrarLogin(loginRequest.getUsuario(), clientIP, false);

response.put("exito", false);

response.put("mensaje", "Error interno del servidor: " + e.getMessage());

return ResponseEntity.status(500).body(response);

}

}

## 4.2 Historia de Usuario 1.5: Mensaje de error por credenciales incorrectas

Archivo Frontend : Login.component.ts

if(this.exito==true)

{

Swal.fire({

title: 'Ingreso exitoso',

text: `Ingreso como ${this.tipoLogin}`,

icon: 'success',

confirmButtonText: 'Aceptar',

});

}

else

{

Swal.fire({

title: 'Error',

html: 'Usuario o contraseña incorrectos.<br>Estas intentando ingresar como <b>' + this.tipoLogin + '</b>' + '<br>Si no es correcto cambie de modo de ingreso.',

icon: 'error',

confirmButtonText: 'Aceptar',

});

}

## 4.3 Historia de Usuario 1.6: Recuperación de contraseña para alumnos

Archivo Frontend : recuperar-contrasena.component.html

<div class="overlay"></div>

<div class="panel-container">

<div class="panel-header">

<h2>Recuperar Contraseña</h2>

<button class="actividad-button cancelar" (click)="regresar()">Cancelar</button>

</div>

<form (ngSubmit)="confirmar()" #formContrasena="ngForm" class="formulario">

*<!-- Nueva contraseña -->*

<div class="form-group">

<label for="nueva">Nueva Contraseña</label>

<input type="password" id="nueva" name="nueva" [(ngModel)]="nuevaContrasena"

required minlength="6" pattern="^(?=.\*[0-9]).{6,}$" #nuevaCtrl="ngModel" />

<div class="error" \*ngIf="nuevaCtrl.invalid && (nuevaCtrl.dirty || nuevaCtrl.touched)">

<span \*ngIf="nuevaCtrl.errors?.['required']">La contraseña es obligatoria.</span>

<span \*ngIf="nuevaCtrl.errors?.['minlength']">Debe tener al menos 6 caracteres.</span>

<span \*ngIf="nuevaCtrl.errors?.['pattern']">Debe incluir al menos un número.</span>

</div>

</div>

*<!-- Confirmar contraseña -->*

<div class="form-group">

<label for="confirmar">Confirmar Contraseña</label>

<input type="password" id="confirmar" name="confirmar" [(ngModel)]="confirmarContrasena"

required #confirmarCtrl="ngModel" />

<div class="error" \*ngIf="confirmarCtrl.invalid && (confirmarCtrl.dirty || confirmarCtrl.touched)">

<span \*ngIf="confirmarCtrl.errors?.['required']">Debe confirmar la contraseña.</span>

</div>

<div class="error" \*ngIf="confirmarContrasena && confirmarContrasena !== nuevaContrasena">

Las contraseñas no coinciden.

</div>

</div>

*<!-- Botón -->*

<button class="actividad-button" type="submit"

[disabled]="formContrasena.invalid || nuevaContrasena !== confirmarContrasena">

Confirmar

</button>

</form>

</div>

Archivo Frontend : recuperar-contrasena.component.ts

import { Component } from '@angular/core';

import { Router } from '@angular/router';

import { FormsModule } from '@angular/forms';

import { CommonModule } from '@angular/common';

import Swal from 'sweetalert2';

@Component({

selector: 'app-recuperar-contrasena',

templateUrl: './recuperar-contrasena.component.html',

styleUrls: ['./recuperar-contrasena.component.css'],

standalone: true,

imports: [FormsModule,CommonModule]

})

export class RecuperarContrasenaComponent {

nuevaContrasena: string = '';

confirmarContrasena: string = '';

constructor(*private* router: Router) { }

regresar() {

this.router.navigate(['/login']);

}

confirmar() {

*// Validar campos vacíos*

if (!this.nuevaContrasena || !this.confirmarContrasena) {

Swal.fire({

icon: 'error',

title: 'Campos vacíos',

text: 'Debe completar ambos campos.'

});

*return*;

}

*// Validar longitud mínima*

if (this.nuevaContrasena.length < 6) {

Swal.fire({

icon: 'error',

title: 'Contraseña inválida',

text: 'La contraseña debe tener al menos 6 caracteres.'

});

*return*;

}

*// Validar que tenga al menos un número*

*const* regexNumero = /\d/;

if (!regexNumero.test(this.nuevaContrasena)) {

Swal.fire({

icon: 'error',

title: 'Contraseña insegura',

text: 'La contraseña debe contener al menos un número.'

});

*return*;

}

*// Validar que coincidan*

if (this.nuevaContrasena !== this.confirmarContrasena) {

Swal.fire({

icon: 'error',

title: 'Error',

text: 'Las contraseñas no coinciden.'

});

*return*;

}

*// Si todo es válido*

Swal.fire({

icon: 'success',

title: 'Éxito',

text: 'Contraseña cambiada con éxito.'

}).then(() => {

this.router.navigate(['/login']);

});

}

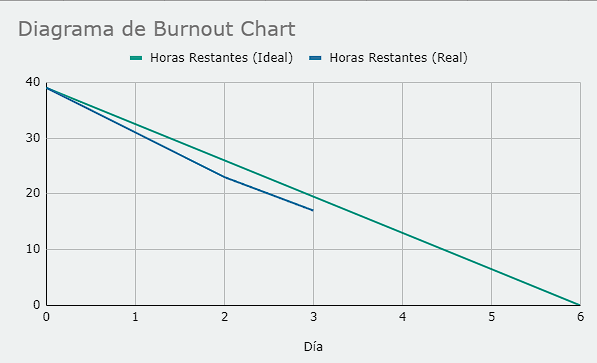
}

## 4.4 Historia de Usuario 1.12: Confirmación visual de cierre de sesión

## 4.5 Historia de Usuario 2.1: Visualización del test en la práctica

# 5. Burndown Chart

| Backlog ID | Historia de Usuario | Tiempo Estimado | 01/10/2025 | 02/10/2025 | 03/10/2025 | 04/10/2025 | 06/10/2025 | 07/10/2025 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dia1** | **Dia2** | **Dia3** | **Dia4** | **Dia5** | **Dia6** |
| HU-1.4 | Acceso de alumnos al sistema | 10 | 5 | 4 | 1 |  |  |  |
| HU-1.5 | Mensaje de error por credenciales incorrectas | 4 | 2 | 0 | 1 |  |  |  |
| HU-1.6 | Recuperación de contraseña para alumnos | 8 | 1 | 4 | 2 |  |  |  |
| HU-1.12 | Confirmación visual de cierre de sesión | 3 |  |  | 1 |  |  |  |
| HU-2.1 | Visualización del test en la práctica | 14 |  |  | 1 |  |  |  |
| Tiempo de trabajo | | 39 | 31 | 23 | 17 |  |  |  |



# 6. Análisis del Sprint

# 7. Conclusiones