



Servicio Nacional de Aprendizaje SENA

Taller sobre construcción del prototipo del software de acuerdo al análisis de las
características funcionales y de calidad

GA5-220501095-AA1-EV01

Instructor:

Tecnico

Presentado por:

Luis David Conde Sanchez

Ficha: 3145644

Fecha: [17/12/2025]

Introducción

El presente documento analiza los fundamentos de la calidad de software, definidos como el grado de cumplimiento de los requisitos para satisfacer al usuario. Se exploran conceptos esenciales como la usabilidad enfocada en la eficacia y satisfacción del usuario y los estándares internacionales, destacando la norma ISO/IEC 25010.

Además, se revisan las tecnologías y herramientas necesarias para asegurar la calidad en el ciclo de vida del desarrollo. Como aplicación práctica de estos conceptos, el informe documenta la implementación técnica de un Registro de Usuario, integrando una interfaz funcional con una base de datos estructurada en PostgreSQL.

Sección 1 – Taller

Registro de Usuario

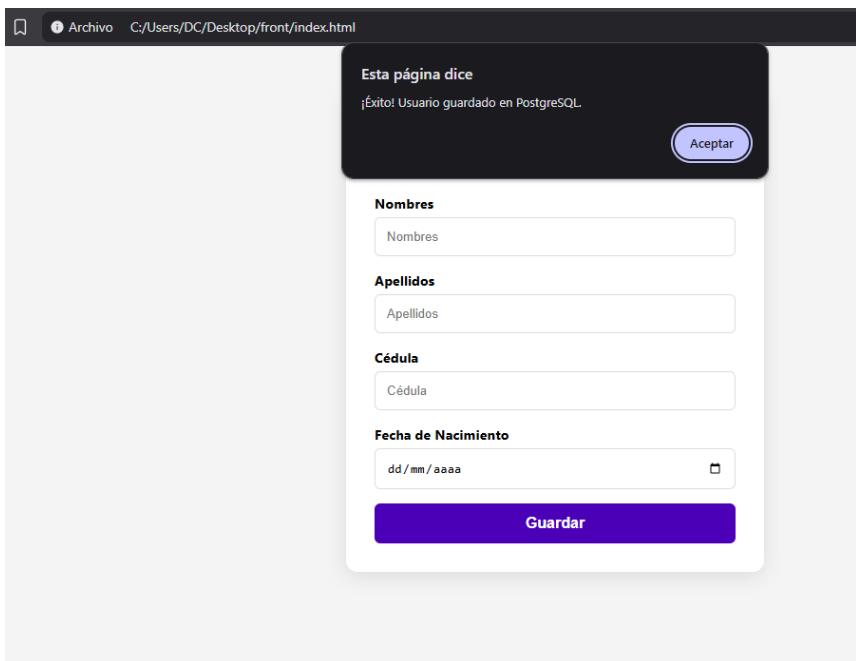
Nombres

Apellidos

Cédula

Fecha de Nacimiento
 dd / mm / aaaa

Guardar



The screenshot shows the pgAdmin 4 interface. In the Object Explorer, under the "Registro" database, a new table named "usuario" is being created. The SQL code in the query editor is:

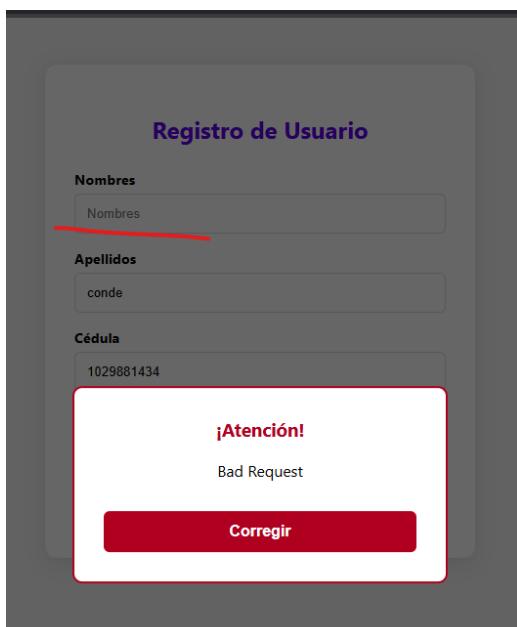
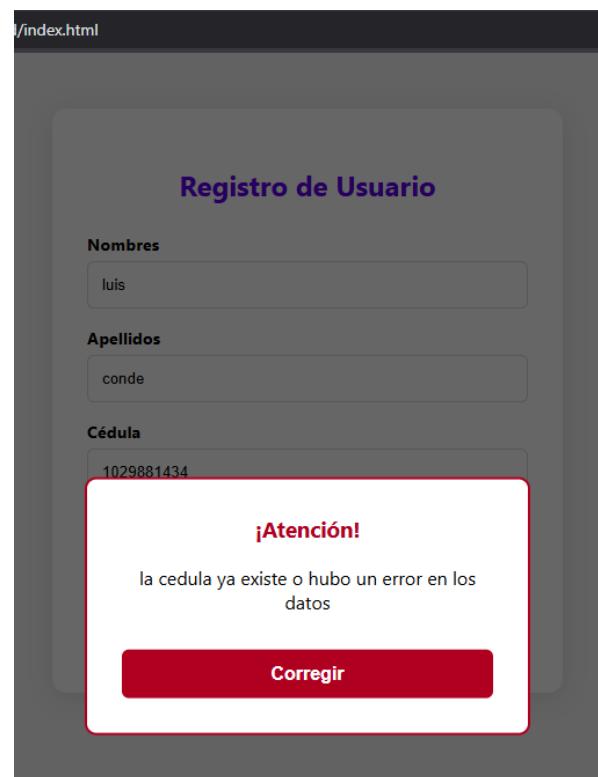
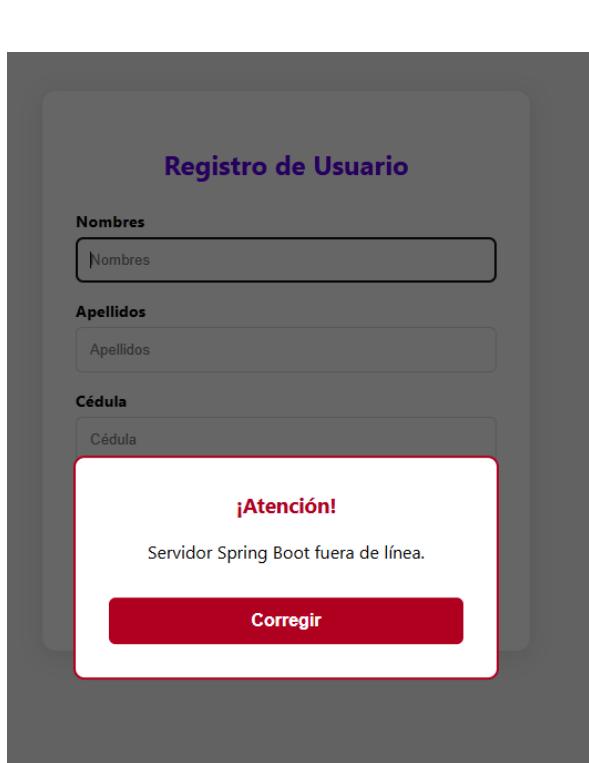
```
create table usuario (
    id serial primary key,
    nombres varchar(100),
    apellidos varchar(100),
    cedula varchar(20) unique,
    fecha_nacimiento date
);
select * from usuario;
```

Below the query editor, the Data Output tab shows the result of the query:

	id	nombres	apellidos	cedula	fecha_nacimiento
1	1	luis	conde	1029881434	2007-07-08

	public
	usuario
	id bigserial
	nombres character varying(255)
	apellidos character varying(255)
	cedula character varying(255)
	fecha_nacimiento date

Mensaje para errores:



Sección 2 – Taller

Fundamentos de la Calidad de Software

Esta investigación presenta una revisión breve sobre los fundamentos de la calidad de software, abordando conceptos clave como su definición, la usabilidad, los estándares de calidad y las tecnologías utilizadas para su implementación. La información fue recopilada a partir de fuentes bibliográficas y recursos confiables disponibles en internet.

1. Definición de calidad de software

La calidad de software se define como el grado en el que un producto de software cumple con los requisitos especificados y satisface las necesidades del usuario final. Según la norma ISO/IEC 25010, un software de calidad debe ser funcional, confiable, usable, eficiente, mantenible y portable. En términos generales, la calidad de software no solo se enfoca en que el sistema funcione, sino en que lo haga de manera correcta, segura y eficiente.

2. Concepto de usabilidad en software

La usabilidad en software hace referencia a la facilidad con la que los usuarios pueden aprender a usar una aplicación, interactuar con ella y lograr sus objetivos de forma efectiva, eficiente y satisfactoria. Un software usable reduce errores, mejora la experiencia del usuario y aumenta la productividad. De acuerdo con la norma ISO 9241, la usabilidad se mide a través de tres factores principales: eficacia, eficiencia y satisfacción del usuario.

3. Estándares de calidad de software

Los estándares de calidad de software son normas que establecen buenas prácticas y criterios para el desarrollo, evaluación y mantenimiento de sistemas informáticos.

Algunos de los estándares más importantes son:

- ISO/IEC 25010: Define el modelo de calidad del producto de software.
- ISO/IEC 12207: Establece los procesos del ciclo de vida del software.
- ISO 9001: Se enfoca en la gestión de la calidad en las organizaciones.
- IEEE 730: Proporciona estándares para planes de aseguramiento de calidad de software.

4. Tecnologías para implementar calidad de software

Existen diversas tecnologías y herramientas que permiten implementar y asegurar la calidad del software durante su desarrollo. Entre las más utilizadas se encuentran:

- Herramientas de pruebas de software como Selenium, JUnit y TestNG.
- Sistemas de control de versiones como Git y GitHub.
- Integración continua y entrega continua (CI/CD) con herramientas como Jenkins y GitLab CI.
- Herramientas de análisis de código como SonarQube.
- Metodologías ágiles como Scrum y DevOps, que promueven la mejora continua.

Fuentes consultadas

1. ISO/IEC 25010: Systems and software Quality Requirements and Evaluation (SQuaRE).
2. ISO 9241 – Ergonomics of human-system interaction.
3. Pressman, R. S. (2010). Ingeniería del Software: Un enfoque práctico. McGraw-Hill.