



**Centro de Biotecnología
Agropecuaria**
Regional Cundinamarca



**TALLER SOBRE CONSTRUCCIÓN DEL PROTOTIPO DEL SOFTWARE DE ACUERDO CON EL
ANÁLISIS DE LAS CARACTERÍSTICAS FUNCIONALES Y DE CALIDAD.**

**YULIET FAIZULI PACHON CARO
JORGE MILLER GUTIERREZ OSPINA
NÉSTOR FABIAN GUTIERREZ SABOGAL**

**SENA
ANÁLISIS Y DESARROLLO DE SOFTWARE – 2721520**

**GUSTAVO ADOLFO CHACON
FEBRERO 2024**



TABLA DE CONTENIDO

INTRODUCCIÓN.....	3
PROTOTIPO DE INGRESO	4
PROTOTIPO INTERFAZ DE USUARIO	4
PROTOTIPO PARA CADA MODULO	5
PROTOTIPO POSIBLES ERRORES	¡Error! Marcador no definido.
DEFINICION DE CALIDAD DE SOFTWARE	6
CONCEPTO DE USABILIDAD EN SOFTWARE	7
ESTÁNDARES DE CALIDAD DE SOFTWARE.....	8
QUE TECNOLOGÍAS EXISTEN PARA IMPLEMENTAR CALIDAD.....	9
Gestión de Proyectos y Colaboración:	9
Control de Versiones:	9
Integración Continua (CI) y Despliegue Continuo (CD):.....	9
Pruebas Automatizadas:	9
Análisis Estático de Código:	10
Monitoreo y Registro:	10
Gestión de Configuración:	10
Pruebas de Seguridad:	10
Colaboración y Comunicación:	10



**Centro de Biotecnología
Agropecuaria**
Regional Cundinamarca



INTRODUCCIÓN

En el siguiente trabajo encontraremos la interfaz gráfica y mapa de navegación de nuestro aplicativo Sistema de Información y Gestión Consultorio Odontológico MJVG

Este proyecto se crea a través de Diseño y desarrollo de la aplicación de escritorio en Java utilizando JavaFX, creación y optimización de la base de datos en MySQL para almacenar la información relevante, implementación de protocolos de comunicación cliente-servidor basados en HTTP y JSON., la cual nos brinda excelentes herramientas y diseños, para la interfaz gráfica y el mapa de navegación.



PROTOTIPO DE INGRESO

A login form prototype with a light blue background and a grey border. At the top center is a circular icon of a man in a suit. Below it are two input fields labeled 'Username' and 'Password'. A blue 'Login' button is positioned below the password field, and a blue 'Register' link is at the bottom.

PROTOTIPO DE REGISTRO

A registration form prototype with a light blue background and a grey border. The title 'REGISTER' is at the top. Below it are five input fields labeled 'Name', 'Last Name', 'Identification', 'Date', and 'Password'. A blue 'Login' button is at the bottom.



PALETA DE COLORES

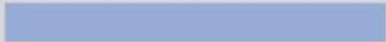
paleta de colores

HSL 0 - 1 - 84 - 100%

HSL 227 - 65 - 80 - 100%



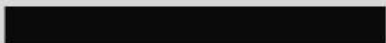
HSL 220 - 46 - 72 - 100%



HSL 241 - 84 - 43 - 100%



HSL 0 - 1 - 4 - 100%



MENSAJES DE ERROR

Mensajes de error

Debes llenar todos los campos

Usuario o contraseña invalidos

Campo obligatorio

Contraseña Debil





DEFINICION DE CALIDAD DE SOFTWARE

La calidad de software se refiere a la medida en que un programa de computadora satisface los requisitos establecidos y las expectativas del usuario. Se evalúa en términos de varios atributos, entre los que se incluyen la eficacia, eficiencia, fiabilidad, mantenibilidad, portabilidad, usabilidad y seguridad del software.

Aquí hay algunas definiciones clave relacionadas con la calidad del software:

1. **Eficacia:** La capacidad del software para realizar las funciones para las que fue diseñado de manera precisa y completa.
2. **Eficiencia:** La capacidad del software para lograr sus objetivos con la menor cantidad posible de recursos, como tiempo y memoria.
3. **Fiabilidad:** La capacidad del software para realizar de manera consistente y sin errores durante un período de tiempo determinado.
4. **Mantenibilidad:** La facilidad con la que el software puede modificarse para corregir errores, mejorar el rendimiento o adaptarse a cambios en el entorno.
5. **Portabilidad:** La capacidad del software para funcionar eficientemente en diferentes entornos, como diferentes sistemas operativos o configuraciones de hardware.
6. **Usabilidad:** La facilidad con la que los usuarios pueden aprender, operar y utilizar el software.
7. **Seguridad:** La capacidad del software para proteger los datos y los recursos del sistema contra accesos no autorizados y ataques maliciosos.

Lograr la calidad del software implica un proceso de desarrollo y pruebas riguroso, así como la adopción de buenas prácticas de ingeniería de software. La gestión de la calidad del software es fundamental para garantizar que el producto final cumpla con los estándares de calidad y satisfaga las necesidades y expectativas de los usuarios.



CONCEPTO DE USABILIDAD EN SOFTWARE

La usabilidad en software se refiere a la medida en que un producto de software puede ser utilizado de manera efectiva, eficiente y satisfactoria por los usuarios, teniendo en cuenta su experiencia general durante la interacción. La usabilidad se centra en la facilidad de uso y en la experiencia del usuario (UX). Un software con alta usabilidad es aquel que permite a los usuarios realizar sus tareas de manera eficaz, sin frustración y con una curva de aprendizaje razonable.

Algunos conceptos clave asociados con la usabilidad en software incluyen:

1. **Facilidad de Aprendizaje:** ** La rapidez con la que los nuevos usuarios pueden aprender a utilizar el software y realizar tareas básicas.
2. **Eficiencia:** La rapidez con la que los usuarios pueden realizar tareas una vez que han aprendido a usar el software.
3. **Memorabilidad:** La facilidad con la que los usuarios pueden recordar cómo utilizar el software después de un período de tiempo sin utilizarlo.
4. **Errores:** La frecuencia y gravedad de los errores cometidos por los usuarios al interactuar con el software, así como la facilidad de recuperación de esos errores.
5. **Satisfacción del Usuario:** La percepción general de los usuarios sobre la comodidad, la utilidad y la satisfacción proporcionada por el software.
6. **Accesibilidad:** La capacidad del software para ser utilizado por personas con diversas habilidades y discapacidades.

La usabilidad se evalúa mediante pruebas de usabilidad, estudios de usuarios y la aplicación de principios de diseño centrados en el usuario. Un diseño de software centrado en la usabilidad contribuye a una experiencia de usuario positiva, lo que a su vez puede aumentar la adopción del producto y la satisfacción del usuario.



ESTÁNDARES DE CALIDAD DE SOFTWARE

Existen varios estándares y marcos de referencia reconocidos a nivel internacional que se utilizan para establecer y evaluar la calidad del software. Algunos de los estándares más destacados incluyen:

1. ISO/IEC 25000 (SQuaRE): La serie de normas ISO/IEC 25000, también conocida como SQuaRE (Software Quality Requirements and Evaluation), proporciona un marco para la evaluación de la calidad del software. Incluye estándares específicos, como ISO/IEC 25010, que se centra en la calidad del producto de software, y ISO/IEC 25040, que aborda la evaluación de la calidad en el proceso de software.
2. ISO 9001:2015: Aunque no está específicamente diseñada para software, la norma ISO 9001 es un estándar de gestión de la calidad aplicable a cualquier tipo de organización. Puede adaptarse para abordar la calidad en el desarrollo y mantenimiento de software.
3. IEEE 730: Este estándar del IEEE (Institute of Electrical and Electronics Engineers) establece las pautas para la calidad del software mediante la definición de procesos y actividades relacionadas con la gestión de la calidad en el desarrollo de software.
4. CMMI (Capability Maturity Model Integration): Desarrollado por el Software Engineering Institute (SEI), el CMMI es un modelo de mejora de procesos que proporciona un conjunto de mejores prácticas para el desarrollo y la mejora de procesos de software y sistemas.
5. ISO/IEC 9126: Este estándar, ahora reemplazado por la serie ISO/IEC 25000, definía un modelo para la evaluación de la calidad del software, abordando características como la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, el mantenimiento y la portabilidad.
6. IEEE 1061: Este estándar proporciona directrices para la realización de pruebas de software y la medición de la calidad del software.

Es importante tener en cuenta que la elección de un estándar puede depender de diversos factores, como el tipo de software, la industria y los requisitos específicos de la organización. Además, algunos estándares pueden complementarse entre sí para abordar diferentes aspectos de



la calidad del software. La adopción y cumplimiento de estos estándares pueden ayudar a garantizar prácticas de desarrollo de software de alta calidad.

QUE TECNOLOGÍAS EXISTEN PARA IMPLEMENTAR CALIDAD

La implementación de calidad en el desarrollo de software implica el uso de diversas tecnologías y herramientas. Estas tecnologías están diseñadas para ayudar en la planificación, desarrollo, prueba y gestión de proyectos, con el objetivo de mejorar la calidad del software. A continuación, se mencionan algunas tecnologías clave:

Gestión de Proyectos y Colaboración:

- ✓ ****JIRA: **** Una herramienta de seguimiento y gestión de proyectos que se utiliza comúnmente para la planificación, seguimiento y gestión de tareas y problemas.
- ✓ ****Trello: **** Una aplicación de gestión de proyectos basada en tableros que facilita la colaboración visual y la organización de tareas.

Control de Versiones:

- ✓ ****Git: **** Un sistema de control de versiones distribuido que permite el seguimiento de cambios en el código fuente y la colaboración entre desarrolladores.

Integración Continua (CI) y Despliegue Continuo (CD):

- ✓ ****Jenkins: **** Una herramienta de integración continua que automatiza el proceso de construcción y prueba del software.
- ✓ ****Travis CI: **** Un servicio de integración continua para proyectos alojados en repositorios de GitHub.

Pruebas Automatizadas:

- ✓ ****Selenium: **** Una herramienta para la automatización de pruebas funcionales de aplicaciones web.
- ✓ ****JUnit y TestNG: **** Frameworks de prueba para aplicaciones Java.
- ✓ ****PyTest y unittest: **** Frameworks de prueba para aplicaciones Python.



Análisis Estático de Código:

- ✓ ****SonarQube:** ****** Una plataforma para evaluar la calidad del código fuente mediante el análisis estático.
- ✓ ****Linters (por ejemplo, ESLint, Pylint):** ****** Herramientas que identifican y corrigen problemas de estilo y errores en el código fuente.

Monitoreo y Registro:

- ✓ ****ELK Stack (Elasticsearch, Logstash, Kibana):** ****** Utilizado para la recopilación, almacenamiento y visualización de registros.
- ✓ ****New Relic y AppDynamics:** ****** Herramientas de monitoreo de aplicaciones que ayudan a identificar problemas de rendimiento.

Gestión de Configuración:

- ✓ ****Ansible y Puppet:** ****** Herramientas para la gestión de configuración y la automatización de la implementación.
- ✓ ****Docker y Kubernetes:** ****** Tecnologías de contenedores que facilitan la implementación y la gestión de aplicaciones.

Pruebas de Seguridad:

- ✓ ****OWASP ZAP (Zed Attack Proxy):** ****** Una herramienta de prueba de seguridad para encontrar vulnerabilidades en aplicaciones web.

Colaboración y Comunicación:

- ✓ ****Slack:** ****** Una plataforma de mensajería para la comunicación en equipos.
- ✓ ****Confluence:** ****** Una herramienta de colaboración para la creación y compartición de documentos.

La selección de tecnologías dependerá de las necesidades específicas del proyecto y del entorno de desarrollo. Integrar estas tecnologías de manera efectiva puede contribuir significativamente a mejorar la calidad del software durante todo el ciclo de vida del desarrollo.