A picture containing text

Description automatically generated

Tecnológico de Costa Rica

Escuela de ingeniería en computación

Curso CI4301-Bases de datos I

Grupo #02-Profesor: Franco Quiros Ramirez

Tarea programada #1

Fernando Daniel Brenes Reyes-Carne 2020097446

Erick Astorga Gamboa-Carne 2019018565

I Semestre

23 de Marzo, 2022

Introducción

Este proyecto se trataba de una simple aplicación web de log in que muestre artículos de una tabla de una base de datos. Para la creación de dicha página web se utilizó PHP como lenguaje y MS SQL como motor de base de datos. Como se puede apreciar en la tabla de análisis de resultados, el proyecto presenta casi una funcionalidad al 100% excepto por mínimos detalles en la sección de inserción de artículos y ambos miembros del grupo trabajaron en un tiempo parecido, esto se puede apreciar en los gráficos extraídos del repositorio de GitHub.

Repositorio

<https://github.com/ErickAG115/Tarea_Porgramada_1_BD>

Bitácora

<https://www.blogger.com/blog/posts/4961574016019014567>

Análisis de resultados

|  |  |
| --- | --- |
| Item | Resultado |
| XML Carga de datos | 100% funcional  Todos los datos son cargados con éxito sin problemas |
| SP creados | 100% funcional  Los stored procedures son funcionales y tiene sus respectivas validaciones |
| Conexión a la BD | 100% funcional  Se logra conectar a la base de datos con procesos php con éxito y sin problemas |
| Validación de usuario y contraseña | 100% funcional  La validación de usuarios y contraseña funcionan y validan correctamente cuando un usuario es erróneo o cuando un espacio está vacío |
| Listado de artículos con filtrado por nombre | 100% funcional  Se filtra correctamente la palabra ingresada en el programa al buscar en la tabla articulo en la columna nombre, se llega a tomar en cuenta la palabra ingresada en cualquier parte del texto |
| Listado de artículos filtrado por cantidad | 100% funcional  Se despliegan todos los artículos y su precio en orden alfabético ascendente por nombre.  Si la caja de texto NO está vacía y se da click en “Filtrar Cantidad”, se despliegan los primeros N artículos y su precio según su orden alfabético ascendente por nombre. Donde N es valor capturado en la caja de texto (filtro) |
| Validación de inserción de artículo | 100% funcional  Se valida correctamente que no exista un artículo con el mismo nombre y se retorna un valor booleano para que la aplicación imprima el mensaje de error o de función respectivo |
| Inserción de artículo | 98% funcional  La inserción en si funciona sin problemas, lo que evita que su funcionalidad esté al 100% es que los botones de inserción hacen que la tabla desaparezca |
| Base de datos | 100% funcional inserta correctamente, se pueden leer los datos y son consistentes con los inserts agregados, y están creadas las tablas que requiere el proyecto |

Métricas del proyecto

|  |  |  |
| --- | --- | --- |
| Nombre de métrica | Valor | Comentario |
| Horas Trabajadas | Daniel: 17 horas  Erick: 18 horas | Distribuimos bien el tiempo de trabajo y lo hicimos eficientemente |
| Entradas del github | Daniel: 7 commits  Erick: 11 commits | Nos distribuimos el trabajo de manera eficiente y pareja |
| Datos de pruebas | 40 pruebas | Varias pruebas para asegurar el funcionamiento del programa |
| Tiempo de pruebas realizadas | 3 horas | Nos tomamos el tiempo escogiendo pruebas para probar la eficiencia del programa |
| Cantidad de tablas de la base de datos | 2 tablas, la de artículo y usuario | Las únicas 2 tablas requeridas para el proyecto |

A screenshot of a computer

Description automatically generated with medium confidence