**UNIVERSIDAD TÉCNICA NACIONAL**

**SEDE SAN CARLOS**

**CARRERA**

**Ingeniería del Software**

**Documentación del Proyecto Final del curso Ingeniería en Software**

**ECOMASTER**

**ELABORADO POR**

**Kevin Arias Arce**

**Roque Chacon Corrales**

Primer cuatrimestre, 2018.

Índice general.

* + Índice de figuras.
  + Índice de tablas.

**Descripción del problema.**

La cadena de supermercados ECONOMAS busca satisfacer las necesidades del cliente, en donde el objetivo principal es ofrecer a los consumidores variedad de productos, precios accesibles, excelente trato y trato personalizado. Por estos objetivos esta empresa quiere modernizar y automatizar lo más posible su servicio para darle al cliente un precio justo. Para ello se analiza diferentes variantes que pueden llevar al cliente a pagar un poco más por el producto. El resultado de este análisis da muestra que la creación y toma de decisiones a la hora que desarrollar los horarios de los trabajadores toma un tiempo significativo, ya que se crea en Excel y toda la lógica de choques de horarios lo hace un usuario, lo cual podría presentar fallos. Al tener que crear diferentes horarios por semana, de las distintas sedes, esto hace que el costo de los productos de eleve lo cual se quiere evitar, por esta importante razón se desea implementar la tecnología en la creación de horarios.

**Justificación del problema.**

**Objetivo general.**

Desarrollar una aplicación web para el mejoramiento de la gestión de horarios, para la empresa ECONOMAS.

**Objetivos específicos.**

1. Entender el funcionamiento de la empresa.
2. Determinar cuáles son los factores que se toman en cuenta para la creación de un horario.
3. Construir reportes, los cuales brindaran una eficiente toma de decisiones.

**Herramientas y técnicas que se utilizan (justificarlas). Incluir qué patrones de diseño están implementados.**

Se utiliza PHP como lenguaje de programación, pero para ello se implementa el framework LARAVEL ya que se considera que facilitara de una gran manera la creación de esta aplicación web, para ello se usara el patrón MVC, trabajando con este patrón se facilitara el entendimiento del código, ya que separa el mismo en tres capas: MODELOS, VISTAS y CONTROLADORES, esto favorece la calidad de la aplicación y permitiría la escalabilidad a futuro.

LARAVEL permite la reducción de costos y tiempos en el desarrollo y mantenimiento otra razón por cual usar LARAVEL es que la curva de aprendizaje es relativamente baja.

**Requerimientos funcionales, no funcionales y del sistema.**

**Requerimientos funcionales**

|  |  |
| --- | --- |
| Requisito Funcional-001 | Agregar Usuario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Requisitos asociados |  |
| Descripción | El sistema deberá contar con un módulo que permita agregar Usuarios a la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario selecciona la opción de ingreso y se muestra la pantalla correspondiente. | | El administrador del sistema ingresa los datos respectivos al nuevo usuario. Se debe asignar el rol del empleado. | | | El sistema verifica que los datos ingresados sean correctos. Si los datos son correctos, el sistema realiza la inserción y muestra un mensaje informativo. Por el contrario, si los datos son incorrectos el sistema le indica al usuario que corrija los datos y realice nuevamente el ingreso de datos. | | | |
| Post Condición |  |
| Importancia | Vital |
| Comentarios | Se crean diferentes tipos de roles. |

*Tabla 1: Requisito Funcional-001 – Agregar Usuario*

|  |  |
| --- | --- |
| Requisito Funcional-002 | Editar Usuario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Requisitos asociados |  |
| Descripción | El sistema deberá contar con un módulo que permita editar Usuarios y actualizarlos en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de editar y se muestra la pantalla correspondiente. | | El administrador del sistema ingresa los datos respectivos al usuario. Se debe asignar el rol del empleado. | | | El sistema verifica que los datos ingresados sean correctos. Si los datos son correctos, el sistema realiza la inserción y muestra un mensaje informativo. Por el contrario, si los datos son incorrectos el sistema le indica al usuario que corrija los datos y realice nuevamente el ingreso de datos. | | | |
| Post Condición |  |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden editar el perfil. |

*Tabla 2: Requisito Funcional-002 – Editar Usuario*

|  |  |
| --- | --- |
| Requisito Funcional-003 | Eliminar Usuario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Requisitos asociados |  |
| Descripción | El sistema deberá contar con un módulo que permita eliminar Usuarios y actualizarlos en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de eliminar y se muestra la pantalla correspondiente.  Se le muestra una alerta y si y solo si el administrador la acepta se elimina de la base de datos. | |
| Post Condición |  |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden eliminar usuarios. |

*Tabla 3: Requisito Funcional-003 – Eliminar Usuario*

|  |  |
| --- | --- |
| Requisito Funcional-004 | Ver lista de Usuarios(empleados) |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Requisitos asociados |  |
| Descripción | El sistema deberá contar con un módulo que permita ver Usuarios almacenados en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de ver y se muestra la pantalla correspondiente. | |
| Post Condición |  |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden ver los usuarios (empleados). |

*Tabla 4: Requisito Funcional-004 – Ver Usuarios*

* + Solución del problema.
  + UML: diagrama de casos de uso, diagrama de clases, diagrama de base de datos, diagrama de la arquitectura.
  + Criterios de aceptación (deben utilizar la plantilla que se adjunta).
  + Resultados de las pruebas (describir los tipos de pruebas que utilizaron y los resultados de las mismas).
  + Conclusiones.
  + Recomendaciones.
  + Anexos (documentación otorgada por el cliente y demás aspectos que propios del proyecto que el equipo de trabajo considere necesario).

\*Plan de contingencia y/o plan de implementación.

ANEXOS

INFORMACION DEL CLIENTE