**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 EconoMás tiene varias sedes distribuidas entre, Pital, Viento Fresco y 2 ubicados en Aguas Zarcas. La oficina de la administración está ubicada en uno de un punto de venta de Aguas Zarcas, desde ahí tiene el manejo de todos los demás.

Entre el personal administrativo está la Supervisora y Gestión de talento humano, esos dos puestos están encargados de la creación y administración de horarios para todo el personal de la cadena.

La creación de horarios lo realizan cada por semana y es uno de los problemas que tienen en la parte administrativa, ya que dichos horarios se realizan en Excel y están propensos a muchos fallos, por ejemplo, más de 48 horas semanales por empleado establecidas por el código de trabajo en el artículo 136, aparecer en dos horarios de diferentes puntos de venta a la misma, dos días libres por empleado en una semana, y muchos fallos que se pueden cometer.

**Justificación del problema.**

El déficit de poder crear horarios más rápido y de una manera más sencilla y cómoda ha perjudicado mucho esta empresa, y es malo para la eficiencia del personal.

Este problema hace más difícil distribuir estos horarios a todo el personal a tiempo y de una manera eficiente, muchas veces pueden tardar hasta el sábado para mandar el horario de la siguiente semana y eso puede perjudicar a muchos empleados y algunas veces también a la parte administrativa.

Por eso se tomó la decisión de realizar una aplicación que pueda satisfacer todas las necesidades de esta empresa a la hora de la creación de horarios, para que así pueden mejorar la calidad de atención al colaborador y así mismo a los clientes.

**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 un lenguaje de programación web muy conocido como los es PHP, y en este caso será con el framework LARAVEL ya que se considera que facilitara de una gran manera la creación de esta aplicación web, también 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.

Se usará el patrón MVC, trabajando con este patrón se facilitará 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.

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

**Requerimientos funcionales**

|  |  |
| --- | --- |
| Requisito Funcional-001 | Agregar Usuario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arias |
| 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-001 | Editar Usuario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arias |
| 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 1: Requisito Funcional-001 – Editar Usuario*

|  |  |
| --- | --- |
| Requisito Funcional-001 | Eliminar Usuario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arias |
| 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 1: Requisito Funcional-001 – Eliminar Usuario*

* + 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