

**UNIVERSIDAD INTERNACIONAL DE LAS
AMERICAS**

ESCUELA DE INGENIERÍA INFORMÁTICA

**PROTORIPO FUNCIONAL PARA EL CONTROL Y
GESTIÓN DEL RECURSO HUMANO EN TIENDA
KAROL**

TRABAJO FINAL DE GRADUACIÓN

**MODALIDAD PROYECTO PARA OPTAR POR EL GRADO DE BACHILLERAO EN
INGENIERÍA INFORMÁTICA**

LUIS DANIEL SALAS ACUÑA

AGOSTO, 2020

DEDICATORIA

Dedico mi proyecto de graduación a mis padres, porque son la motivación de mi vida, porque me brindaron la oportunidad y el apoyo para estudiar la carrera que me apasiona, su fe en mí me impulsó durante toda esta etapa a confiar en mí mismo y esforzarme día con día; sin duda, son ellos quienes me guían por el camino correcto en cada etapa de mi vida.

A mi hermano, quien es fundamental en mi vida, al brindarme siempre el amor y el apoyo necesarios.

A mis amigos más cercanos, quienes siempre estuvieron en los momentos más difíciles, ofreciéndome su respaldo, sus consejos y la motivación para seguir adelante.

AGRADECIMIENTOS

Agradecer primero a Dios, por la bendición de contar con salud, con mi familia y de tener la oportunidad de estudiar y formarme como profesional.

Mis agradecimientos a cada uno de los docentes, quienes, con su trabajo, ayudaron a forjar en mi persona los valores de la responsabilidad y ética, su conocimiento, sus correcciones y su apoyo han sido indispensables en mi formación como profesional.

A mis padres, por el apoyo incondicional y su guía, sin ellos nada de esto sería posible.

Contenido

DEDICATORIA.....	2
AGRADECIMIENTOS.....	3
SOLICITUD DE DEFENZA	5
APROBACIÓN DEL TRIBUNAL EXAMINADOR.....	6
CARTA DEL LECTOR	8
CÓDIGO DE ÉTICA	9
CARTA DE REVISIÓN FILOLÓGICA.....	10
DECLARACIÓN JURADA.....	11
CAPÍTULO I: Introducción	21
Planteamiento del problema	21
Objetivos.....	22
Objetivo General.....	22
Objetivos Específicos	22
Justificación	22
Viabilidad Técnica.....	23
Viabilidad Económica	24
Viabilidad Operativa.....	25
Viabilidad Legal	25
Proyecciones.....	26
Alcances.....	26
Limitaciones	29
CAPITULO II: MARCO REFERENCIAL.....	31
Referencia Técnica	31
Referencia de negocio	38
Jornada ordinaria diurna	40

Jornada ordinaria acumulativa.....	41
CAPITULO III: MARCO METODOLÓGICO	43
Enfoques de la investigación	43
Enfoque Cuantitativo.....	43
Enfoque Cualitativo.....	43
Enfoque Mixto.....	44
Enfoque Seleccionado	44
Tipo de investigación	44
Investigación aplicada	44
Investigación pura.....	45
Investigación por utilizar	45
Fuentes de información	45
Variables o unidades de análisis.....	47
Variable Conceptual	47
Variable Operacional.....	47
Variable Instrumental	48
Población	49
Muestra.....	50
Muestreo no probabilístico	51
Muestreos probabilísticos.....	52
Definición de variables que componen la fórmula.....	53
Cálculo de la muestra	54
Instrumentos de recolección de datos.....	54
Entrevista.....	54
Encuesta.....	55
Encuesta Descriptiva.	55
Encuesta Analítica.	55

	14
Proceso para la recolección y análisis de datos	56
CAPÍTULO IV: ANÁLISIS DE RESULTADOS	57
Entrevista	57
Encuesta.....	59
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	64
Conclusiones.....	64
Recomendaciones	65
CAPÍTULO VI PROPUESTA	67
Análisis	67
Análisis detallado del software desarrollado	67
Análisis detallado del hardware requerido	70
Análisis detallado de las telecomunicaciones.....	74
Descripción detallada de la base de datos.	75
Descripción detallada del personal requerido.....	75
Casos de Uso	76
Fuente: (Elaboración Propia).....	96
Diseño.....	96
Arquitectura del sistema	96
Arquitectura del software	97
Diseño de interfaces.....	98
Diseño de base de datos y su diccionario de datos.	118
Diseño de procesos	132
Diseño de Salidas	137
Diagramas UML.....	140
Programación.....	141
Entradas y salidas	141
Procesos	143

Validación.....	145
• Reportes.....	155
Pruebas	157
Referencias	161
APÉNDICES	166
Apéndice A. Guía para entrevista en Tienda Karol.....	166
Apéndice B. Encuesta dirigida a los empleados de Tienda Karol.....	167

Tabla De Cuadros

Cuadro 1. <i>Requerimientos técnicos del prototipo</i>	24
Cuadro 2. <i>Costos de software para desarrollo</i>	24
Cuadro 3. <i>Costos del desarrollo del prototipo</i>	25
Cuadro 4. <i>Análisis de las variables</i>	48
Cuadro 5. <i>Requerimientos de hardware</i>	71
Cuadro 6. <i>Costos Azure por App Service</i>	72
Cuadro 7. <i>Costos Azure por Azure SQL Database Estándar</i>	72
Cuadro 8. <i>Costos servicios de almacenamiento GearHost</i>	73
Cuadro 9. <i>Análisis de telecomunicaciones</i>	74
Cuadro 10. <i>Descripción de la base de datos</i>	75
Cuadro 11. <i>Descripción del personal requerido</i>	75
Cuadro 12. <i>Caso de Uso 01</i>	78
Cuadro 13. <i>Caso de uso 02</i>	79
Cuadro 14. <i>Caso de uso 03</i>	80
Cuadro 15. <i>Caso de Uso 04</i>	81
Cuadro 16. <i>Caso de uso 05</i>	82
Cuadro 17. <i>Caso de uso 06</i>	83
Cuadro 18. <i>Caso de uso 07</i>	84
Cuadro 19. <i>Caso de Uso 08</i>	85
Cuadro 20. <i>Caso de uso 09</i>	86
Cuadro 21. <i>Caso de uso 10</i>	87
Cuadro 22. <i>Caso de uso 11</i>	87
Cuadro 23 <i>caso de uso 12</i>	89
Cuadro 24. <i>Caso de uso 13</i>	91
Cuadro 25. <i>Caso de uso 14</i>	92
Cuadro 26. <i>Casos de uso 15</i>	93
Cuadro 27. <i>Caso de Uso 16</i>	94
Cuadro 28. <i>Caso de uso 17</i>	95
Cuadro 29. <i>Diccionario de Datos – Persona</i>	120
Cuadro 30. <i>Diccionario de Datos – Empleados</i>	120
Cuadro 31. <i>Diccionario de Datos – Usuarios</i>	121
Cuadro 32. <i>Diccionario de Datos – Tipo_Usuario</i>	121

Cuadro 33. <i>Diccionario de Datos – Provincia</i>	121
Cuadro 34. <i>Diccionario de Datos – Canton</i>	122
Cuadro 35. <i>Diccionario de Datos – Distrito</i>	122
Cuadro 36. <i>Diccionario de Datos – Dirección</i>	122
Cuadro 37. <i>Diccionario de Datos – Currículum</i>	123
Cuadro 38. <i>Diccionario de Datos – Correo</i>	123
Cuadro 39. <i>Diccionario de Datos - Aguinaldos</i>	123
Cuadro 40. <i>Diccionario de Datos – Contratos</i>	124
Cuadro 41. <i>Diccionario de Datos – Tipo_Contrato</i>	124
Cuadro 42. <i>Diccionario de Datos –Horarios</i>	124
Cuadro 43. <i>Diccionario de Datos – Cargos</i>	125
Cuadro 44. <i>Diccionario de Datos – Contratos_Conceptos</i>	125
Cuadro 45. <i>Diccionario de Datos – Estado_Justificación</i>	125
Cuadro 46. <i>Diccionario de Datos – Estado_Justificación</i>	126
Cuadro 47. <i>Diccionario de Datos – Estado_Solicitud</i>	126
Cuadro 48. <i>Diccionario de Datos –Vacaciones</i>	126
Cuadro 49. <i>Diccionario de Datos – Historico_Vacaciones</i>	127
Cuadro 50. <i>Diccionario de Datos – Justificaciones</i>	127
Cuadro 51. <i>Diccionario de Datos – Solicitudes</i>	128
Cuadro 52. <i>Diccionario de Datos – Diccionario de Datos - CambioEstadoJustificaciones</i>	128
Cuadro 53. <i>Diccionario de Datos – Diccionario de Datos – CambioEstadoSolicitudes</i>	128
Cuadro 54. <i>Diccionario de Datos – Marcas</i>	129
Cuadro 55. <i>Diccionario de Datos – Tipo_Marcas</i>	129
Cuadro 56. <i>Diccionario de Datos – Telefono</i>	129
Cuadro 57. <i>Diccionario de Datos – Planilla</i>	130
Cuadro 58. <i>Diccionario de Datos – Liquidaciones</i>	130
Cuadro 59. <i>Diccionario de Datos - ISR</i>	131
Cuadro 60. <i>Diccionario de Datos – ISR_Contratos</i>	131
Cuadro 61. <i>Caso de Prueba – Inicio Sesión</i>	157
Cuadro 62. <i>Caso de Prueba – Agregar Persona</i>	158
Cuadro 63. <i>Caso de prueba – Agregar Horario</i>	159
Cuadro 64. <i>Caso de Prueba – Cálculo de planilla</i>	160
Cuadro 65. <i>Caso de Prueba – Nueva Solicitud</i>	160

Tabla de Imágenes

Imagen 1. <i>Porcentaje de uso de tipos de Bases de datos</i>	36
Imagen 2. <i>Etapas del modelo en cascada</i>	37
Imagen 3. <i>Formula para el cálculo de la muestra</i>	53
Imagen 4. <i>Diagrama de Casos de Uso</i>	77
Imagen 5. <i>Arquitectura del sistema</i>	97
Imagen 6. <i>Arquitectura del software</i>	98
Imagen 7. <i>Vista Login</i>	99
Imagen 8. <i>Vista Pantalla Principal Administradores</i>	100
Imagen 9. <i>Vista Pantalla Principal Empleados</i>	100
Imagen 10. <i>Ejemplo Vista Pantalla Principal Opciones Desplegadas</i>	101
Imagen 11. <i>Gestión Tipos de Solicitud</i>	102
Imagen 12. <i>Gestión Tipos de Justificación</i>	102
Imagen 13. <i>Gestión Tipos Contratos</i>	102
Imagen 14. <i>Gestión Estados Solicitud</i>	103
Imagen 15. <i>Gestión Estados Justificación</i>	103
Imagen 16. <i>Gestión Horarios</i>	103
Imagen 17. <i>Gestión Conceptos</i>	104
Imagen 18. <i>Gestión ISR</i>	104
Imagen 19. <i>Formato nuevo para gestión de tipos de solicitudes, justificaciones o contratos</i> . 105	
Imagen 20. <i>Formato Editar para gestión de tipos de solicitudes, justificaciones o contratos</i> . 105	
Imagen 21. <i>Formato Nuevo para gestión de estados de solicitudes o justificaciones</i>	106
Imagen 22. <i>Formato Editar para gestión de estados de solicitudes o justificaciones</i>	106
Imagen 23. <i>Nuevo Horario</i>	106
Imagen 24. <i>Editar Horario</i>	107
Imagen 25. <i>Editar ISR</i>	107
Imagen 26. <i>Personas</i>	107
Imagen 27. <i>Nueva Persona</i>	108
Imagen 28. <i>Editar Persona</i>	109
Imagen 29. <i>Actualizar Currículum</i>	109
Imagen 30. <i>Administrar Correos</i>	109
Imagen 31. <i>Administrar Teléfonos</i>	109
Imagen 32. <i>Empleados</i>	110
Imagen 33. <i>Nuevo Empleado</i>	111

Imagen 34. Editar Empleado	111
Imagen 35. Editar Empleado / Contrato	111
Imagen 36. Cambiar Foto	112
Imagen 37. Asignación de conceptos.....	112
Imagen 38. Nueva Asignación	112
Imagen 39. Gestor de planilla	113
Imagen 40. Calcular Planilla	113
Imagen 41. Deducciones por ISR.....	114
Imagen 42. Deducciones por conceptos	114
Imagen 43. Cálculo de llegadas tardías	114
Imagen 44. Gestor de Vacaciones	115
Imagen 45. Asignar Vacación	115
Imagen 46. Histórico Vacacional	116
Imagen 47. Formato Vistas Gestor de Justificaciones o Solicitudes	116
Imagen 48. Formato Vistas Estados de justificaciones solicitudes	117
Imagen 49. Formato Vistas Nueva Justificación o Solicitud	117
Imagen 50. Formato Vistas Edición Cambio de Estados de Justificaciones o Solicitudes.....	117
Imagen 51. Diagrama Entidad Relación Base de Datos.....	119
Imagen 52. Diagrama de Flujo mantenimiento de Personas	133
Imagen 53. Diagrama de flujo mantenimiento de empleados	134
Imagen 54. Diagrama de flujo gestión de planilla.....	135
Imagen 55. Diagrama de flujo gestor de justificaciones	136
Imagen 56. Diagrama de flujo gestor de solicitudes	137
Imagen 57. Diseño archivo de salida reporte llegadas tardías por empleado.....	138
Imagen 58. Diseño archivo de salida reporte solicitudes por empleado.....	138
Imagen 59. Diseño archivo de salida reporte justificaciones por empleado	138
Imagen 60. Reporte llegadas tardías por empleado	139
Imagen 61. Reporte solicitudes por empleado	139
Imagen 62. Reporte justificaciones por empleado	140
Imagen 63. Diagrama de clases.....	141
Imagen 64. Código de entrada	142
Imagen 65. Salida de datos	143
Imagen 66. Código de proceso.....	144
Imagen 67. Código de validación	145

Imagen 68. Código Módulo Seguridad – Crear usuario	146
Imagen 69. Código mantenimiento de personas – Nueva persona.....	147
Imagen 70. Código mantenimiento de personas – Editar persona.....	148
Imagen 71. Mantenimiento de Personas – Eliminar Persona	148
Imagen 72. Código cálculo de planilla	149
Imagen 73. Código nueva vacación.....	150
Imagen 74. Parte Uno – Código Justificación	151
Imagen 75. Parte dos – Código Justificación	152
Imagen 76. Parte uno - Código nueva solicitud	153
Imagen 77. Parte dos – Código nueva solicitud	154
Imagen 78. Código Reportes – Reporte Pagos por Empleado	155
Imagen 79. Código Reportes – Reporte solicitudes por empleado.....	156

Tabla de Gráficas

Gráfica 1. Pregunta 1.....	59
Gráfica 2. Pregunta 2.....	60
Gráfica 3. Pregunta 3.....	60
Gráfica 4. Pregunta 4.....	61
Gráfica 5. Pregunta 5.....	62
Gráfica 6. Pregunta 6.....	62
Gráfica 7. Pregunta 7.....	63
Gráfica 8. Pregunta 8.....	63

CAPÍTULO I: Introducción

Planteamiento del problema

Tienda Karol es una empresa que labora desde 1986 dedicada especialmente a la venta de ropa y accesorios para damas, caballeros y niños. La empresa cuenta con una única tienda física ubicada en el mercado central de Puriscal y cuenta con un total de seis empleados y dos administradores.

Desde sus inicios, la empresa ha laborado de forma totalmente manual, es decir, carece de la tecnología necesaria para el desarrollo de sus tareas cotidianas; situación que a lo largo del tiempo y con el crecimiento de la empresa ha generado una serie de inconvenientes.

A continuación, se mencionan algunos problemas que en el presente enfrenta la empresa en estudio:

- 1. Atraso en pagos al personal:** Actualmente la empresa lleva a cabo el proceso de pagos a su planilla de forma totalmente manual, esto provoca la constante pérdida de información por errores humanos, tras papeleo y confusiones al realizar este proceso; situación que afecta al personal, pues al recibir su salario ello provoca insatisfacción.
- 2. Confusión en pagos al personal:** No existe un debido control de la información, lo cual genera confusión en los pagos a los empleados, quienes se quejan constantemente al no recibir el pago correspondiente.
- 3. Falta de precisión en los reportes:** Con base en los empleados, salarios y horarios, la empresa genera reportes mensualmente, los cuales presentan errores debido a la inconsistencia de los procesos antes mencionados. Ello ha causado que los reportes no generen información certera para la toma de decisiones de la entidad.
- 4. Atrasos en atención al cliente:** En la actualidad Tienda Karol carece de un proceso estructurado para las solicitudes del personal como lo son vacaciones, permisos extraordinarios, entre otros. Esto ha causado que, en ciertas ocasiones, los permisos sean otorgados a varios empleados simultáneamente. En consecuencia, queda un personal que no da abasto con la demanda de los

clientes, lo cual provoca insatisfacción, molestias y hasta pérdida en ventas por la mala atención a la clientela.

A partir de las situaciones expuestas, es evidente que los administradores requieren de un sistema que los apoye en la gestión de la información certera de sus recursos humanos y el cual les permita acceder a ella en cualquier momento y en cualquier lugar para la toma de decisiones. Por tanto, según los requerimientos de la tienda y de sus empleados, se decidió que el proyecto se elabora como una aplicación web a la cual los usuarios puedan acceder en el momento requerido.

Cabe destacar que la empresa cuenta con servicio de internet actualmente, aspecto que beneficia la implementación de una solución web.

Objetivos

Objetivo General

Desarrollar un prototipo funcional para el control y gestión del recurso humano en Tienda Karol.

Objetivos Específicos

- Analizar los requerimientos para el prototipo funcional según las necesidades de la empresa Tienda Karol.
- Diseñar, según los requerimientos de Tienda Karol, la estructura del prototipo funcional.
- Programar la solución, cumpliendo con los requerimientos de la empresa.
- Probar las funcionalidades del prototipo desarrollado, validando el cumplimiento de los requerimientos de la empresa y el correcto funcionamiento de este.

Justificación

Tienda Karol es una empresa con 34 años en el mercado de la moda. Sus fundadores iniciaron vendiendo prendas de forma ambulante durante muchos años hasta llegar a lo que es hoy; sus administradores nunca fueron personas familiarizadas con la tecnología, razón por la cual, han trabajado siempre de forma manual, utilizando papel y lápiz para el control de la información de sus procesos administrativos.

Dicha forma de trabajar les funcionó en el pasado; sin embargo, con el crecimiento de la empresa y el pasar de los años, el volumen de información y su importancia, ha venido incrementando de gran forma, lo cual ha provocado un aumento en la dificultad de la administración y gestión de la información de la empresa.

A pesar de lo antes indicado, los administradores de Tienda Karol reconocen que un recurso tecnológico les ayudará a solventar sus problemas administrativos y los errores cometidos por el factor humano. Por tanto, en el año 2020 se les propone, entonces, el desarrollo de un prototipo funcional que llegue a solventar los conflictos presentes y la mejora de sus procesos administrativos.

Con la puesta en marcha de este prototipo funcional se pretende brindarle a la empresa una solución y una mejora en su gestión y control del recurso humano, ya que este contará con la información de manera centralizada, segura y disponible en todo momento, capaz de llevar el control de sus empleados, planillas, contratos, entre otros; de esta manera, evitar los problemas actuales, como por ejemplo: la pérdida de información, atrasos o confusiones en pagos al personal, entre otros.

Viabilidad Técnica

Tienda Karol recientemente adquirió equipo de cómputo básico, puesto que los administradores determinaron la necesidad inmediata de implementar recursos tecnológicos para sus gestiones. Así que, actualmente poseen dos equipos de escritorio, ambos con sus respectivos monitores y demás periféricos. Además, cuentan con servicio de internet el cual facilita el desarrollo de un prototipo funcional en una infraestructura web.

Dadas las características técnicas disponibles, ahora en la compañía se optó por la propuesta de una herramienta desarrollada en un entorno web, que pueda ser utilizado por administradores y empleados, no solo desde el local de la empresa, sino también con posibilidad de acceder desde otros sitios. De tal forma, se les ofrece a los administradores la facilidad de contar con la información en cualquier momento y lugar con servicio de internet, luego de implementado el prototipo.

A continuación, se describen las características técnicas necesarias, tanto para el desarrollo como para la implementación de este prototipo funcional para la gestión del recurso humano:

Cuadro 1. *Requerimientos técnicos del prototipo*

Requerimientos para el desarrollo	Requerimientos para la implementación
Computador con las siguientes características: RAM: 4Gb (mínimo). Disco Duro: 500Gb Procesador: 64 bits	Computador con las siguientes características: RAM: 4Gb (mínimo). Disco Duro: 500Gb Procesador: 64 bits
Motor de Base de Datos: SQL Server Express 2017 (gratuito).	Motor de Base de Datos: SQL Server Express 2017 (gratuito).
Entorno de desarrollo: Visual Studio Community 2019 (gratuito).	Servidores para publicación: Microsoft Azure Gear-Host para .Net Apps Somee Hosting

Fuente: (Elaboración propia).

Viabilidad Económica

Se sabe que para materializar una herramienta se requiere de diferentes productos de software adicionales que brinden las facilidades para el desarrollo de esta, además de un motor de base de datos para almacenar la información con la que trabajará eventualmente el prototipo funcional.

Para el desarrollo de este prototipo funcional se utilizará el entorno de desarrollo Visual Studio 2019, el cual cuenta con una versión totalmente gratuita llamada Visual Studio Community; acompañado de un motor de base de datos SQL Server 2017, el cual cuenta también con una versión gratuita llamada SQL Express 2017, tal y como se detalla en el cuadro 2:

Cuadro 2. *Costos de software para desarrollo*

Herramientas de Software	Costo
Visual Studio Community 2019	Licencia gratuita
SQL Express 2017	Licencia gratuita
Costo Total	Gratuito

Fuente: (Elaboración propia).

La elaboración de este prototipo funcional requiere de cuatro etapas: análisis, diseño, desarrollo y pruebas de funcionalidad. Para cada una de ellas es indispensable de horas hombre para ser completadas.

Según Ministerio de Trabajo y Seguridad Social de Costa Rica (2020), un programador de computación en condición de trabajador en ocupación especializada percibe ₡13.872,70 por hora laborada. A continuación, se describen los costos del desarrollo, desglosado por cada una de sus etapas.

Cuadro 3. *Costos del desarrollo del prototipo*

Etapas	Horas	Costo
Análisis	50hrs	₡693,635.00
Diseño	50hrs	₡693,635.00
Desarrollo	360hrs	₡4.994,172.00
Pruebas de funcionalidad	140hrs	₡1.942,178.00
Totales	600hrs	₡8.323,347.00

Fuente: (Elaboración propia).

Viabilidad Operativa

Los fundadores de Tienda Karol nunca se han familiarizado con el mundo tecnológico, sin embargo, hace dos años la administración de la empresa fue tomada por sus hijos, quienes cuentan con mucho más conocimiento en cuanto al uso de computadoras y dispositivos tecnológicos; al igual que sus empleados, quienes, por lo general, se trata de personas jóvenes con una media de edad entre los 18 a los 23 años, lo cual es beneficioso, porque no les será complejo manejar el prototipo propuesto.

Viabilidad Legal

Este proyecto se realizará bajo la jurisdicción de las leyes descritas a continuación:

- **Ley N°6683 - Ley de derechos de autor y derechos conexos:** La presente ley protege las obras, interpretaciones o ejecuciones y fonogramas de autores, artistas, intérpretes o ejecutantes y productores de fonogramas costarricenses, domiciliados o no en el territorio nacional. Se señalan mediante citas bibliográficas y se incluyen en las referencias de este proyecto toda información que provenga de otros autores y sus obras.
- **Ley N°7975 - Ley de información no divulgada:** La presente ley protege la información no divulgada referente a los secretos comerciales e industriales que guarde, con carácter confidencial, una persona física o jurídica para impedir que información legítimamente bajo su control sea divulgada a terceros, adquirida o utilizada sin su consentimiento.

- **Ley N°8039 - Ley de Procedimientos de Observancia de los Derechos de Propiedad:** Establece los procedimientos aplicables en materia civil y penal para la protección de los derechos intelectuales, incluidos los derechos de autor, las herramientas involucradas en el desarrollo de este proyecto utilizarán sus paquetes de licenciamiento gratuito.
- **Ley 8968 – Ley de protección de la Persona Frente al Tratamiento de sus Datos Personales:** Esta ley es de orden público y tiene como objetivo garantizar a cualquier persona, independientemente de su nacionalidad, residencia o domicilio, el respeto a sus derechos fundamentales, concretamente, su derecho a la autodeterminación informativa en relación con su vida o actividad privada y demás derechos de la personalidad, así como la defensa de su libertad e igualdad con respecto al tratamiento automatizado o manual de los datos correspondientes a su persona o bienes; para el desarrollo de este proyecto la información manipulada será de uso exclusivo y con fines educativos.

Las leyes precitadas forman parte de la viabilidad legal de este proyecto y serán cumplidas a cabalidad para su desarrollo.

Proyecciones

Con este prototipo funcional se pretende brindar a Tienda Karol una solución a sus problemas en la gestión del recurso humano, con el apoyo de una herramienta de software que se encargue de optimizar el proceso de análisis y administración del recurso humano; asimismo, se pretende beneficiar la reducción costos, el uso del papel, tiempo y esfuerzo.

A continuación, se expondrán los alcances para el desarrollo de este prototipo para la gestión del recurso humano:

Alcances

A continuación, se describirá cada uno de los alcances de este proyecto:

Alcance Funcional

1. Módulo de Seguridad:

Este módulo está destinado a brindar la seguridad necesaria para la administración de perfiles de usuario y sus roles. Además, se registrará la hora del ingreso al sistema, con el fin de validar el cumplimiento del horario de los colaboradores.

2. Módulo de Reportes:

En este módulo se visualizarán diferentes reportes que están destinados a la toma de decisiones de la empresa:

- Reporte general de pagos.
- Reporte de porcentajes por intervalo de edades
- Reporte de pagos por empleado.
- Reporte de cantidad de llegadas tardías por empleado.
- Reporte de cantidad de ausencias por empleado.
- Reporte de solicitudes por empleado.
- Reporte de justificaciones por empleado.

3. Módulo de Mantenimiento:

- **Personas:** En este submódulo, se ingresará información personal, hoja de vida y demás datos importantes de personas al sistema a las que se les podría asignar un contrato posteriormente si se desea contratar.
- **Gestor de campos:** En este submódulo, se podrán administrar los catálogos con los que trabajará la herramienta, tipos de solicitudes, tipos de justificación, tipos de contratos, estados de solicitudes, estados de justificaciones, horarios y conceptos.
- **Empleados:** En este submódulo, se podrá administrar la información de los empleados contratados, además se podrá contratar a una persona talento previamente registrada en el sistema, asignándole la información de su contrato.

- **Conceptos:** En este submódulo se podrán crear conceptos o porcentajes predefinidos, los cuales pueden ser asignados posteriormente a un contrato.

4. Módulo de Planilla:

Este módulo contará con distintos submódulos:

- **Gestor de planilla:** En este submódulo se procesarán las planillas de forma mensual y se visualizarán todos sus detalles, código de la planilla, código del empleado, nombre del empleado, periodo de la planilla, sueldo base y sueldo neto.
- **Cálculo de llegadas tardías:** En este submódulo se procesarán las horas tardías. según las marcas generadas por los empleados a través de la consola.
- **Gestor de vacaciones:** En este submódulo se podrá gestionar las vacaciones del personal en planilla. Descontará automáticamente la cantidad disponible de vacaciones por empleado.
- **Liquidaciones:** En este submódulo se podrá realizar el proceso de liquidación, y calculará el monto que debe recibir el empleado por la liquidación de su contrato. Tomando en cuenta todos los conceptos.

5. Módulo de Justificaciones:

En este módulo el empleado podrá ingresar justificaciones de ausencias, llegadas tardías y demás. Posterior a ello el sistema notificará a los administradores y mostrará los detalles de la justificación para la toma de decisión. El sistema también notificará al empleado la aprobación o negación de la justificación y sus detalles.

6. Módulo de Solicitudes:

En este módulo el empleado podrá ingresar solicitudes de vacaciones, adelantos, y demás. Posterior a ello, el sistema notificará a los usuarios administradores y mostrará los detalles de la solicitud para la toma de decisión. En caso de tratarse de un adelanto y el administrador lo apruebe, el sistema automáticamente descontará el monto solicitado del pago más cercano al

empleado. En caso de tratarse de vacaciones y el administrador lo apruebe, el sistema descontará automáticamente la cantidad de días de vacaciones disponibles al empleado. El sistema también notificará al empleado la aprobación o negación de la solicitud y sus detalles.

Alcance Metodológico

Para el desarrollo de este proyecto se utiliza una metodología para desarrollo de software, el concepto de metodología del software tal y como lo indican Rivas y otros (2015) en su escrito, es: “El estudio y determinación de cuál es el método más adecuado para dar incremento al software.”

Basado en lo anterior, el objetivo de utilizar una metodología está en brindar los procesos más adecuados para que la materialización de la herramienta del software sea posible sin altibajos.

Este prototipo será trabajado bajo la metodología del ciclo de vida del software, cascada. Este Modelo divide el proyecto en fases; análisis, diseño, implementación, pruebas y mantenimiento. Las fases de implementación y mantenimiento no se tomarán en cuenta en el alcance de este proyecto.

Alcance Tecnológico

El prototipo funcional se desarrollará en un entorno web utilizando una base de datos SQL Express 2017. El aplicativo web será desarrollado utilizando el lenguaje de programación C# utilizando el framework ASP.NET MVC, en el entorno de desarrollo Visual Studio Community 2019.

Si se desea colocar este prototipo web en producción, se deberá realizar la publicación del aplicativo y su base de datos en algún servidor host.

Como recomendación, Gear Host y Somee son herramientas gratuitas para el hosting de aplicaciones y bases de datos.

Limitaciones

Para este proyecto no se realizarán los puntos mencionados a continuación:

- Este prototipo se desarrollará, mas no será implementado en la empresa Tienda Karol; por lo tanto, el prototipo no será publicado en servidores web ni colocado en producción.

- Se explicarán las funcionalidades del prototipo a los administradores de la empresa Tienda Karol; sin embargo, no se realizarán capacitaciones a los usuarios finales ni se crearán manuales de usuario.

CAPITULO II: MARCO REFERENCIAL

Referencia Técnica

En este capítulo se presentarán principalmente conceptos que apoyan el fundamento y la comprensión de este proyecto.

La Informática a través de los años ha venido cambiando las formas de manejo de la información, brindando soluciones automatizadas y certeras, gracias al uso de los computadores.

Herederó (2004) Define informática como “La ciencia que estudia y se ocupa del tratamiento automático y racional de la información o también como la ciencia de los ordenadores” (p.14).

Actualmente, en Tienda Karol la aplicación de la informática es prácticamente nula y su implementación beneficiaría de gran manera a los procesos internos de la empresa, al brindar gran agilidad a los trabajos, disminución de pérdida de información, información centralizada, disminución de costos elevando la productividad y aportando eficacia a los procesos. Para Espinosa (2005), se define como: “El campo de conocimiento que abarca todos los aspectos del diseño y uso de los computadores. De acuerdo con su definición, el computador actúa con dos tipos de informaciones: instrucciones (que indican a la máquina que es lo que tiene que hacer) y datos (que son los elementos que procesa o genera el programa)” (p.2).

Es decir que, con el uso de la informática se logra obtener resultados o datos a partir de instrucciones dadas a un computador y estos datos pueden ser utilizados con distintos fines; esto sucede con ayuda de los sistemas informáticos, en Tienda Karol se busca la creación de un sistema que simplifique las tareas de la administración de recursos humanos por medio de instrucciones dadas y que sus resultados sirvan para los procesos, análisis y la toma de decisiones dentro de la empresa.

Según Pérez & González (2015) los sistemas informáticos son: “Un conjunto de dispositivos, con al menos una CPU o unidad central de proceso, que estarán física y lógicamente conectados entre sí” (p.16). En Tienda Karol se pretende crear un sistema informático web que solucione su problemática actual y para ello, según el texto citado anteriormente, deben existir componentes físicos como el computador y sus elementos, además de los componentes lógicos como lo es, por ejemplo, el sistema o aplicación en cuestión que se debe desarrollar con base en los requerimientos de la empresa para obtener el resultado deseado.

Básicamente, los componentes físicos es lo que se conoce como hardware y los componentes lógicos como software, tal como lo explican Pérez et al. (2015) en su escrito:

- Hardware: Formado por aquellos elementos físicos del SI, siendo elementos hardware el elemento terminal, los canales y los soportes de la información.
- Software: Aquellos elementos del sistema que no tienen naturaleza física y que se usan para el procesamiento de la información (p.18).

A lo largo de los años y con la evolución tecnológica las herramientas o sistemas informáticos han venido apoyando a la gestión de muchas empresas a nivel mundial, brindándoles un control mucho más efectivo de sus actividades diarias, reduciendo costos, esfuerzo humano y solucionando muchos otros problemas; por medio de software y hardware, programadores son capaces de desarrollar herramientas de este tipo con ayuda de lenguajes de programación, que sirven para crear las instrucciones que requiere el computador para interpretar la información y brindar al usuario el resultado esperado, tal como lo indica Mathieu (2015): “Un lenguaje para transmitir a la máquina las órdenes que se le quieren dar; esto es, el lenguaje de programación y una manera de pensar y concebir órdenes a la computadora” (p.10).

Basado en el texto anterior, el computador es capaz de interpretar órdenes por medio de un lenguaje máquina, sin embargo, las ideas o peticiones son adaptadas a dicho lenguaje por los programadores o desarrolladores, personas que se encargan de plasmar en código, instrucciones de manera planificada para obtener un resultado esperado por parte del computador.

Según Bird (2000), “El papel principal de un programador es la definición de una función que permita resolver un problema dado” (p.1).

Para el desarrollo de este prototipo, se utilizará el lenguaje de programación C# en el framework ASP.NET MVC5; framework se traduce al español como: “marco de trabajo”, que es utilizado para el desarrollo de aplicaciones.

Este lenguaje de programación fue seleccionado debido a que su línea de aprendizaje y uso es más simple con respecto a otros lenguajes, brindando una sintaxis más consistente y lógica, además posee la ventaja de contar con el uso del framework ASP.NET MVC5 el cual, según Diaz, (2010):

Fue creado por Microsoft con la idea de proveer a los desarrolladores de aplicaciones web mayor control sobre el HTML, crear direcciones web intuitivas de manera más potente, lo cual ayuda a posicionar mejor las páginas en los motores de búsqueda y, por último, uno de los aspectos más importantes es que ASP.NET MVC nos permite llevar un desarrollo guiado por pruebas facilitándonos la creación y ejecución de las pruebas unitarias del sistema (párr.1).

Un framework puede resumirse como un entorno de desarrollo, que, establece las estructuras y herramientas de programación que facilitan la labor de los desarrolladores.

Este framework de programación es ejecutado por medio del Software Visual Studio, que proporciona las herramientas necesarias para la programación en el entorno .NET y el lenguaje C#, el cual de acuerdo con Sierra (2015):

...se trata de un lenguaje moderno orientado a objetos que permite desarrollar una amplia gama de aplicaciones para la plataforma Microsoft .NET. Más que otra cosa, el objetivo de C# es permitir a todos los desarrolladores en general, y a los de C y C++ en particular, abordar el desarrollo de aplicaciones complejas con facilidad y rapidez, pero sin sacrificar la potencia y el control que ofrecen C y C++ (p.25).

.NET Framework brinda un entorno de programación orientada a objetos, este tipo de programación se caracteriza por la creación de modelos de la vida real dentro de la lógica de la codificación, representando objetos con atributos que aparecen dentro de la solución de la problemática que se desea resolver con el sistema que se va a crear.

Para el desarrollo de este proyecto se prefirió optar por el lenguaje orientado a objetos gracias a que apoya la reutilización de código, su orden y su legibilidad. Además, simplifica la escritura de código porque es posible conceptualizarlo como algo que le sucede a un objeto en la vida real.

Lo mencionado anteriormente, lo explica el autor Álvarez, (2019) en su sitio web, en donde define a la programación orientada a objetos como un paradigma de la programación, una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones, brindando orden y re-utilidad al código programado.

Por ejemplo, durante la programación de este prototipo el cual está destinado a la gestión del recurso humano, se utilizará este tipo de programación, creando una estructura lógica con los distintos objetos relacionados al recurso humano, personas, empleados, contratos, y demás. Cabe destacar que dichos objetos a nivel de desarrollo son reutilizables en cualquier momento gracias a la programación orientada objetos.

La programación Orientada a objetos (POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación.

Para el desarrollo de este prototipo se requiere también de la implementación de una base datos que se encargue de almacenar toda la información que manejará el sistema tanto de entrada como de salida.

Por ejemplo, en Tienda Karol constantemente se requiere almacenar información de contratos, empleados, planillas, vacaciones y demás conceptos, que se especificarán más adelante; a la vez, esta información debe ser consultada en el momento que se requiera para la toma de decisiones, a diferencia de manejarla en papeles, una base de datos integrada con un aplicativo proporciona mayor seguridad, agilidad y respaldo tanto a la información como a los procesos como tal, sin embargo, surge la pregunta: ¿Qué es una base de datos?

Con base en lo que manifiesta Romero y otros (2019), una base de datos se define como: “(...) una colección de información organizada de tal modo que sea fácilmente accesible, gestionada y actualizada” (p.18).

La función de una base de datos es indispensable para un sistema informático, debido a que, al mantener la información centralizada y accesible brinda al usuario grandes beneficios de automatización al lograr manipular y gestionar sus datos por medio de una aplicación que consulte y actualice la información de la base de datos.

Para la creación de una base de datos se requiere de un motor, el cual según Pérez (2011), se conoce como: “El servicio principal para almacenar, procesar y proteger los datos. Asimismo, proporciona acceso controlado y procesamiento rápido de transacciones para cumplir los requisitos de las aplicaciones” (p.1).

Es decir, un motor de base de datos proporciona seguridad y agilidad al manejo de los datos de las aplicaciones para su correcto funcionamiento y es el servicio encargado de almacenar las bases de datos y sus componentes. Para el desarrollo de este proyecto se utilizará el motor de base de datos llamado Microsoft SQL Server, el cual, según Pérez (2011): “Es un sistema de bases de datos profesional de Microsoft. Contiene una variedad

de características y herramientas que se pueden utilizar para desarrollar y administrar bases de datos y soluciones de todo tipo basadas en ellas” (p.1).

Microsoft SQL Server es un sistema de bases de datos relacionales, lo que quiere decir que las bases de datos SQL, cuentan con relaciones entre sus elementos.

Amazon (2020), define las bases de datos relacionales como:

Una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas. Las tablas se utilizan para guardar información sobre los objetos que se van a representar en la base de datos. Cada columna de una tabla guarda un determinado tipo de datos y un campo almacena el valor real de un atributo. Las filas de la tabla representan una recopilación de valores relacionados de un objeto o una entidad. Cada fila de una tabla podría marcarse con un identificador único denominado clave principal, mientras que filas de varias tablas pueden relacionarse con claves extranjeras. Se puede obtener acceso a estos datos de muchas formas distintas sin reorganizar las propias tablas de la base de datos. (párr.1).

Dentro de las herramientas que ofrece Microsoft SQL Server, se encuentra Microsoft SQL Management Studio, que será utilizada para el desarrollo de este proyecto. Esta herramienta proporciona las funcionalidades para el manejo y diseño de las bases de datos en el motor SQL Server y sus componentes, constituye un entorno integrado para obtener acceso, configurar y administrar todos los componentes de SQL Server que además, cuenta con herramientas gráficas suficientes para ofrecer a desarrolladores y administradores con cualquier grado de experticia, acceso a SQL Server (Pérez, 2011).

Al igual que los sistemas que requieren de un lenguaje para su programación, las bases de datos utilizan un lenguaje para su creación, manejo y control. SQL es un lenguaje utilizado para la creación, uso y administración de bases de datos relacionales.

Quintana, 2014 aduce que:

SQL (Structured Query Language) consiste en un lenguaje de programación diseñado especialmente para el acceso a Sistemas de Gestión de Bases de Datos Relacionales (SGBDR). Como la mayor parte de los sistemas actuales son de este tipo, y como el lenguaje SQL es el más ampliamente usado en éstos, se puede decir sin ningún género de dudas que este lenguaje es empleado mayoritariamente en los sistemas existentes hoy en día (p.8).

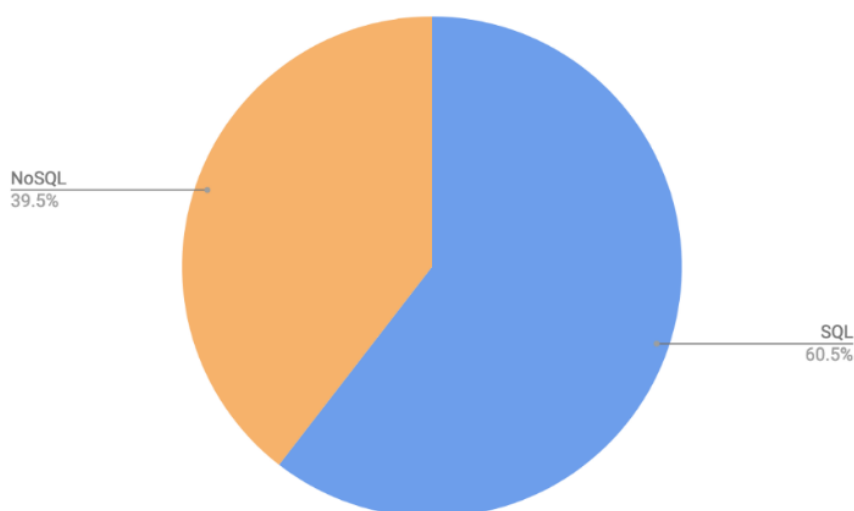
Según lo citado anteriormente, las bases de datos más utilizadas en la actualidad son SQL, bases de datos relacionales, las cuales definen y manipulan datos basados en lenguaje de consulta estructurado (SQL). Estas son las más utilizadas y útiles para manejar datos estructurados que organizan elementos de datos y estandarizan cómo se relacionan entre sí, a diferencia de las bases de datos no relacionadas que permiten almacenar y recuperar datos no estructurados utilizando un esquema dinámico. (ScaleGrid, 2019). En la imagen No. 1, se muestra gráficamente los porcentajes de uso de las bases de datos relacionales (SQL) y las no relacionales (NoSQL).

Imagen 1. *Porcentaje de uso de tipos de Bases de datos*

Fuente: (ScaleGrid, 2019).

SQL Database Use: 60.48%

NoSQL Database Use: 39.52%



Tal como se muestra en la imagen anterior, las bases de datos SQL son más utilizadas con respecto a las bases de datos NoSQL, abarcando un 60,5% de uso. El lenguaje SQL se segmenta en dos tipos de consultas llamadas DDL y DML las cuales son de utilidad para la introducción o consulta de información en una base de datos, para profundizar más este tema el autor Quintana (2014), explica ambos tipos de la siguiente forma:

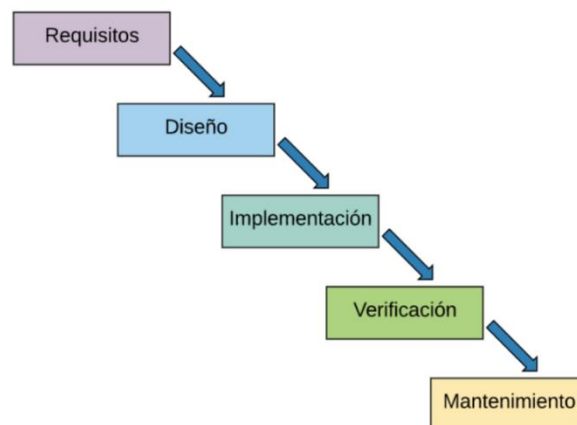
Lenguaje de Definición de Datos (en inglés Data Definition Language o DDL): Incluye aquellas sentencias que sirven para definir los datos o para modificar su definición, como por ejemplo la creación de tablas, índices, etcétera.

Lenguaje de Manipulación de Datos (en inglés Data Manipulation Language o DML): Incluye aquellas sentencias que sirven para manipular o procesar los datos, como por ejemplo la inserción, borrado, modificación o actualización de datos en las tablas (p.9).

Para este proyecto, se utilizará la metodología de desarrollo ciclo de vida en modelo cascada, porque esta nos aporta una planificación más sencilla, admite iteraciones entre sus etapas y provee un producto de calidad, esta metodología cuenta con una serie de fases que apoyan a que el proceso de desarrollo se realice con las mejores prácticas.

Para Villagómez (2017), el modelo de ciclo de vida en cascada se define como: “Una secuencia de fases donde al final de cada una de ellas se reúne la documentación para garantizar que cumple las especificaciones y los requisitos antes de pasar a la fase siguiente.” (párr. 16). En la imagen No. 2 se muestran las fases de este modelo.

Imagen 2. *Etapas del modelo en cascada*



Fuente: (Domínguez, 2017).

En imagen anterior (Imagen 2), se puede observar el nombre y orden secuencial de cada una de las fases o etapas del modelo ciclo de vida en cascada. Para comprender en mayor medida cada una de sus fases, a continuación, se describen brevemente:

- **Requisitos:** Esta etapa se refiere al planteamiento de los requerimientos del software y el problema que debe resolver.
- **Diseño:** En esta etapa se utiliza la información recolectada en la primera etapa para iniciar el diseño del producto.
- **Implementación:** Consiste en el desarrollo de los componentes del producto, basándose en los resultados obtenidos en la etapa de diseño.
- **Verificación:** en esta etapa se verifica que el software cumpla con los requerimientos establecidos por medio de distintas pruebas.
- **Mantenimiento:** Etapa encargada de la corrección de errores que puedan existir en el software.

Dentro del mundo del diseño del software encontramos la arquitectura del software, que es lo que define su estructura, sus componentes y como interactúan cada uno ellos para su funcionamiento y operación. Reynoso (2004) en su investigación lo define como: “La organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.” (p.12). Para este proyecto será implementada la arquitectura cliente-servidor basada en tecnologías web, el autor Marini (2012) define esta arquitectura como:

Un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones clientes realizan peticiones a una o varias aplicaciones servidores, que deben encontrarse en ejecución para atender dichas demandas. Tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes (p.1).

Basado en el texto anterior, la arquitectura cliente-servidor cuenta con dos entidades fundamentales, el cliente, el cual posee la función principal de emitir las peticiones al servidor y este último, como el encargado de procesar la información necesaria para enviar respuestas certeras a las peticiones realizadas por el cliente.

La ventaja de esta arquitectura consiste en la estabilidad brindada a las soluciones, dado que es posible el aumento de la capacidad de las entidades; ello en caso de que se requiera en un momento determinado.

Referencia de negocio

Como se ha indicado anteriormente, este prototipo se desarrollará para solucionar los procesos de gestión del recurso humano en la empresa Tienda Karol, que se dedica a la venta de ropa y accesorios para todo público.

El prototipo por desarrollar pretende contribuir con la administración de recursos humanos, lo que comprende todo lo relacionado con la organización y acrecentamiento del personal que conforma una empresa. Villalpando (2014), define este concepto como:

“Las actividades esenciales de las personas para lograr los objetivos de la organización, consiste en contratar sus servicios, desarrollar sus habilidades y motivarlas para alcanzar altos niveles de desempeño, además de asegurarse de que siguen manteniendo su compromiso con la organización” (p.3).

Un sistema para la administración del recurso humano logra automatizar y centralizar la información de planillas, contratos, empleados, cargas sociales y demás; con el consecuente beneficio y agilización de estos procesos a las empresas. Esto es lo

que se pretende con la creación de un prototipo web para Tienda Karol, apoyar a la solución y automatización de la gestión del recurso humano.

En materia de recursos humanos, se conoce a la planilla o nómina, como el documento o formulario que detalla la información de forma mensual sobre pagos y remuneraciones al personal de una empresa.

El autor Lacalle García (2011), define el término de nómina o planilla como:

“El documento o recibo que se entrega al pagar el salario al trabajador y que, por tanto, sirve al empresario como justificante de haber realizado dicho pago” (p.228).

Por otra parte, se puede entender como empleado o trabajador a la persona que presta sus servicios a una institución a cambio de un salario ya sea mensual o quincenal según se defina.

Según Carballo y Villasmil (2005), se entiende por empleado:

El trabajador en cuya labor predomine el esfuerzo intelectual o no manual. El esfuerzo intelectual, para que un trabajador sea calificado como empleado, puede ser anterior al momento en que presta sus servicios y en este caso consistirá en estudios que haya tenido que realizar para poder prestar eficientemente su labor, sin que pueda considerarse como tal el entrenamiento especial o aprendizaje requerido para el trabajo manual calificado (p.54).

Uno de los grandes problemas actuales en la administración del recurso humano en Tienda Karol es el manejo de planillas de personal, debido a que su procedimiento actual es de forma manual y en ocasiones la información es extraviada, por ende, existen múltiples fallos en los cálculos de pagos y demás, los autores Martínez y Eder (2018) explican que:

El manejo de las planillas de personal en una empresa es un proceso crítico, debido a que es en la planilla en donde se van a realizar los cálculos de los pagos, sean ingresos, descuentos, retenciones y aportaciones por cada trabajador, y almacenar esa información para poder utilizarla a futuro como data histórica y que permitirá tomar decisiones y hacer consultas de forma más rápida, efectiva y confiable (p.8).

Esto se relaciona con la importancia de almacenar la información, sucede con todos los datos asociados a los empleados; de ahí el objetivo de una solución programada que apoye por medio de distintos módulos que ayuden al manejo de la información, automatización de procesos y la generación de reportes mensuales que faciliten la planificación, proyección y toma de decisiones para los administradores de las empresas en sus tareas de gestión del recurso humano.

En materia de recursos humanos los contratos son otro punto muy importante y se vuelve indispensable contar con un registro de cada uno de los contratos del personal, según Tejedo (2013): “Un contrato de trabajo es el acuerdo por el que un trabajador y un empresario manifiestan su voluntad de intercambiar los servicios que presta el primero por la retribución que entrega el segundo, dando lugar así al nacimiento de la relación laboral” (p.16).

Lo anterior se puede resumir en que un contrato laboral, es el acuerdo entre una persona y una empresa, iniciando de esta manera, una relación laboral con las condiciones definidas para ambas partes.

La relación laboral debe darse siempre dentro de lo establecido por las leyes laborales, de no ser así el contrato laboral no tendrá validez en materia judicial para reclamos acerca de su cumplimiento. Para profundizar el concepto de un contrato, el Ministerio de Trabajo y Seguridad Social de Costa Rica (2020), establece en el código de trabajo costarricense, documento el cual rige las normas o leyes sobre las relaciones laborales entre patrones y empleadores, en su artículo 18 que:

Un contrato individual de trabajo sea cual fuere su denominación, es todo aquél en que una persona se obliga a prestar a otra sus servicios o a ejecutarle una obra, bajo la dependencia permanente y dirección inmediata o delegada de ésta, y por una remuneración de cualquier clase o forma. Se presume la existencia de este contrato entre el trabajador que presta sus servicios y la persona que los recibe (p.3).

Con apoyo de este prototipo web también se pretende centralizar este tipo de información, de manera que se pueda consultar información de los contratos siempre y en todo momento de manera rápida, organizada y además que pueda ser editada o manipulada para la toma de decisiones en la compañía.

Por medio de los contratos también se acuerda la jornada laboral. En Costa Rica existen cuatro tipos de jornadas laborales, que se explican a continuación:

Jornada ordinaria diurna

En esta jornada se acuerda un periodo laboral comprendido entre las cinco de la mañana y las siete de la noche. Está compuesta por ocho (8) horas por día y cuarenta y ocho horas (48) por semana.

En trabajos que no sean peligrosos ni insalubres puede ser hasta de diez (10) horas diarias, pero no más de cuarenta y ocho (48) horas semanales, así se conforma lo que se conoce como Jornada Diurna Acumulativa.

Jornada ordinaria nocturna:

Se acuerda un periodo laboral comprendido entre las siete de la noche y las cinco de la mañana del día siguiente. Es de seis (6) horas por día y treinta y seis (36) horas semanales. No es permitida la jornada acumulativa nocturna.

Jornada ordinaria mixta:

Es aquella en que se labora una parte en el período comprendido entre las cinco de la mañana y las siete de la noche y otra parte entre las siete de la noche y las cinco de la mañana; por ejemplo, se ingresa a las dos de la tarde y se sale a las diez de la noche. Es de siete horas por día (7) y cuarenta y dos (42) horas semanales. En trabajos no peligrosos ni insalubres puede ser hasta de ocho (8) horas diarias y cuarenta y ocho (48) horas por semana. Por el hecho que se permite laborar hasta ocho horas por día, también constituye una jornada especial o de excepción. Si se trabaja hasta las diez y treinta de la noche o más, la jornada mixta se convierte en jornada nocturna y será de seis (6) horas por día y treinta y seis (36) horas por semana.

Jornada ordinaria acumulativa

Es aquella mediante la cual las horas que corresponde laborar el sexto día de la semana (que generalmente es sábado), se trabajan en los cinco días anteriores. Para todos los efectos legales, el sexto día (sábado) sigue siendo un día hábil; por eso se toma en cuenta para otorgar vacaciones (Ministerio de Trabajo y Seguridad Social de Costa Rica, 2020, p.36 – p.44)-

En Tienda Karol actualmente para sus empleados rige una jornada laboral ordinaria diurna, la compañía abre sus puertas al público de lunes a sábado desde las 8:00am hasta las 5:00pm y domingos de 8:00am a 1:00pm, sus empleados regulares laboran 8 horas al día de lunes a sábado, 48 horas semanales y un administrador de la tienda labora los domingos.

Las vacaciones y su correcta administración también son de suma importancia, por ejemplo, actualmente en Tienda Karol, constantemente se generan confusiones con respecto a las vacaciones asignadas a los empleados, la información de las vacaciones disponibles por empleado no es precisa, ello genera gran malestar en los colaboradores de la tienda, todo esto producto de una mala administración de la información, en el artículo 153 del Código de Trabajo costarricense se dicta que:

Todo trabajador tiene derecho a vacaciones anuales remuneradas, cuyo mínimo se fija en dos semanas por cada cincuenta semanas de labores continuas, al servicio de un mismo patrono. En caso de terminación del contrato de trabajo antes de cumplir el período de las cincuenta semanas, el trabajador tendrá derecho, como mínimo, a un día de vacaciones por cada mes trabajado, que se le pagará en el momento del retiro de su trabajo (Ministerio de Trabajo y Seguridad Social de Costa Rica, 2020, p.48).

La correcta administración de los recursos humanos y de todas sus variables involucradas es aplicable y de suma importancia para todo tipo de organizaciones sin importar su tamaño. Las actividades de contratación, desarrollo de los trabajadores y el manejo de la información en dicha materia son indispensables para el cumplimiento de los objetivos de una compañía.

A partir de lo anterior, es posible determinar que Tienda Karol es una compañía carente de una óptima administración de su recurso humano, en consecuencia, los procesos manuales han causado serios problemas a sus administradores por extravío de información, escasez de reportes, información poco precisa, entre otras. Por tal motivo, se propone la implementación de una plataforma web, la cual brinde a la compañía seguridad en sus datos y una importante mejora en sus procesos.

CAPITULO III: MARCO METODOLÓGICO

Enfoques de la investigación

Ruiz Medina (2011) explica que:

El enfoque de la investigación es un proceso sistemático, disciplinado y controlado, está directamente relacionada a los métodos de investigación que son dos: método inductivo generalmente asociado con la investigación cualitativa que consiste en ir de los casos particulares a la generalización; mientras que el método deductivo, es asociado habitualmente con la investigación cuantitativa cuya característica es ir de lo general a lo particular (p.152).

Para una investigación, se debe tener claro el rumbo el cual esta tomará, definir dicho rumbo es posible gracias a tres enfoques principales que existen para las investigaciones, a continuación, se explicarán cada uno de estos enfoques.

Enfoque Cuantitativo

Hernández Sampieri y otros (2006) señalan que: “Un enfoque cuantitativo usa la recolección de datos para probar hipótesis, con base en la medición numérica y el análisis estadístico, para establecer patrones de comportamiento y probar teorías” (p.5).

Es decir que, el enfoque cuantitativo de una investigación se refiere a un método en el cual se implementa una recolección de datos concretos o de análisis para probar lo que se está planteando, brindando de esta forma un respaldo que apoye a las conclusiones generales de la investigación.

Enfoque Cualitativo

Sobre este enfoque parafraseando a Sanfeliciano (2018), se indica que los estudios cualitativos no siguen un proceso rígido y secuencial. Estos pueden desarrollar preguntas e hipótesis antes, durante o después de la recolección y análisis de datos. Se basa en un método indagatorio o exploratorio, muchas veces previo al cuantitativo, para afinar las preguntas de investigación o proponer nuevas preguntas en relación con la cuestión de estudio.

Basado en lo anterior, este enfoque utiliza datos cualitativos en lugar de datos concretos como lo son las cifras, por ejemplo, en la investigación cualitativa se hace uso de instrumentos como las entrevistas o encuestas que apoyen a profundizar en el tema con pensamientos, opiniones o argumentos de las personas a partir de preguntas o planteamientos. Por lo general, son datos más difíciles de medir, sin embargo, aportan profundidad a la investigación, claro está que se pueden plantear preguntas de carácter

medible con el fin de aplicar técnicas de un enfoque cuantitativo lo que lleva al siguiente enfoque, el enfoque mixto.

Enfoque Mixto

En este enfoque se utilizan técnicas tanto cuantitativas como cualitativas, tal como aducen Cabezas y otros (2018):

En el enfoque mixto, se utilizan instrumentos como entrevistas, encuestas para saber opiniones de cada cual, sobre el tema en discusión, se reconstruyen hechos y otros, además esas encuestas pueden ser valoradas a través de escalas medibles y se hacen valoraciones numéricas de las mismas, se observan tendencias, frecuencias y se plantean hipótesis que se corroboran a posterior. Tener en cuenta que aquí se integran ambas concepciones combinándose procesos para llegar a resultados más importantes y profundos (p.67).

Enfoque Seleccionado

Para el desarrollo de este proyecto se utilizará un enfoque mixto, que apoye tanto la observación de los procesos de gestión del recurso humano actuales en la empresa Tienda Karol, como a la recolección de opiniones del personal administrativo de la compañía, con el fin de analizar los resultados obtenidos.

La ventaja de manejar un enfoque mixto es que con dicha perspectiva se pueden solucionar, de manera más profunda, los inconvenientes actuales en la compañía, gracias a que se podrán analizar los requerimientos de Tienda Karol a nivel de procesos, por medio de entrevistas y encuestas; de esta forma, obtener resultados más precisos y ajustados a la realidad, los cuales apoyen al cubrimiento de las debilidades actuales en la gestión del recurso humano en la compañía.

Tipo de investigación

Esta investigación se divide en dos tipos: la investigación pura e investigación aplicada. A continuación, se detallará la definición de cada uno de estos tipos de investigaciones.

Investigación aplicada

Para Murillo (2008), la investigación aplicada recibe el nombre de “investigación práctica o empírica”, que se caracteriza porque busca la aplicación o utilización de los conocimientos adquiridos, a la vez que se adquieren otros, después de implementar y sistematizar la práctica basada en investigación. El uso del conocimiento y los resultados

de investigación que da como resultado una forma rigurosa, organizada y sistemática de conocer la realidad.

Con base en el texto anterior, la investigación aplicada se centra en la resolución de un problema basado en la aplicación o utilización de los conocimientos obtenidos y de esta forma satisfacer las necesidades concretamente.

Investigación pura

Según Robles (2017), la idea de la investigación pura consiste en tener un mejor entendimiento de los conceptos fundamentales dentro de un campo científico en particular. No tiene una aplicación inmediata y específica, aunque más adelante sus hallazgos pueden servir para otros usos aplicables. Generalmente, estas investigaciones implican un gran nivel de calibre intelectual, así como el respaldo de una institución respetada. Aquellos que se involucran en estas investigaciones colocan todos sus esfuerzos a la formulación o reformulación de teorías.

Investigación por utilizar

Para este proyecto se utilizará una investigación aplicada, debido a que para el desarrollo de un prototipo funcional se requiere del análisis e implementación de los conocimientos obtenidos de la teoría para dar con el resultado esperado para el prototipo.

Fuentes de información

Se conoce como fuentes de información a aquellos recursos que provean información o datos, tal como lo señala Ruiz (2012): “Las fuentes de información documental son cualquier escrito o testimonio gráfico o visual que proporcionan datos sobre el tema que se está investigando” (p.135).

Estas fuentes de información se pueden dividir en tres tipos primaria, secundarias y terciarias.

A continuación, se detallarán estas etapas:

Fuentes primarias.

Se consideran fuentes primarias de información a aquel material el cual es considerado como de primera mano.

Tal como lo indican los autores Del Castillo y Orozco (2015, citado en López, Guevara y Miladis, 2009): “Las fuentes primarias exponen por primera vez

descubrimientos científicos, observaciones y datos originales. Los datos que proporcionan son de primera mano” (p.112).

De acuerdo con el concepto anterior, las fuentes primarias prácticamente son los artículos científicos publicados, ya sea en formato impreso o electrónico.

El autor Seminario (2011), explica en su blog online que, un artículo científico debe reunir las siguientes condiciones:

- Contener la información necesaria y suficiente para que otros investigadores puedan repetir el trabajo.
- Ha sido sometido a revisión por pares (árbitros o expertos).
- Su visibilidad es alta, es decir, por su tiraje (dado por la revista científica en la cual está publicado), está completamente disponible para los usuarios en general, y para las bases de datos y bibliotecas.

Fuentes Secundarias.

Seminario, (2011) indica que las fuentes secundarias se elaboran con fines de difusión masiva. No necesariamente las elabora el propio investigador. Las fuentes secundarias por excelencia son los libros y los artículos de revisión.

Son documentos que compilan y reseñan la información publicada en las fuentes primarias. Retoman los documentos primarios u originales. Proporcionan una síntesis de la información que existe en los documentos primarios sobre temas de interés; además, se utilizan para remitir a los usuarios a documentos cuyos contenidos puedan ayudar a solucionar sus necesidades de información. Entre las fuentes secundarias se encuentran:

- Compilaciones.
- Obras de referencias.
- Enciclopedias.
- Resúmenes. (Del Castillo y Orozco, 2015).

Fuentes terciarias

Se conocen como fuentes terciarias a el material que proviene de la compilación de varias fuentes primarias y secundarias, Del Castillo y Orozco (2015, citado en López, Guevara y Miladis, 2009) detallan que: “Las fuentes terciarias se refieren a documentos que compendian nombres y títulos de otras revistas y publicaciones periódicas/boletines, así como títulos que contienen referencias” (p.113).

Es decir, las fuentes terciarias se alimentan de fuentes que compilan información publicada, fuentes de segunda mano o fuentes secundarias.

Para este proyecto se utilizarán los tres tipos de fuentes de información, debido a que en primer lugar como es el caso de las fuentes primarias, se elaborarán entrevistas, recolección y observación de datos para el desarrollo del proyecto.

Las fuentes secundarias se utilizarán como complemento en diferentes definiciones y conceptos con el fin de fundamentar, ampliar y materializar de mejor forma las ideas conceptuales planteadas.

Al realizar referencias bibliográficas, se utilizarán fuentes terciarias que aportarán aún más apoyo a las fuentes primarias y secundarias.

Variables o unidades de análisis

Una variable es una propiedad que tiene la posibilidad de variar a través del tiempo tal y como lo describe Cauas (2015): “Es la propiedad o característica de un objeto o fenómeno que presenta variaciones en sucesivas mediciones temporales. De otra forma, se trata de una característica observable o un aspecto discernible en un objeto de estudio que puede adoptar diferentes valores o expresarse en varias categorías” (p.3).

Desde esta definición todo aquello que pueda cambiar tanto cualitativa como cuantitativamente se puede definir como variable.

Variable Conceptual

Cauas (2015) explica que: “En el nivel conceptual se enumeran las propiedades de interés inmediato para la investigación y se postulan las relaciones entre ellas” (p.4).

En términos generales las variables conceptuales son expresadas en conceptos teóricos que apoyan el desarrollo de una investigación.

Variable Operacional

Según Cauas (2015): “En el nivel operacional, el análisis debe poder establecer las asociaciones o correlaciones existentes entre variables tal como se dan en los datos observados y se verifica si esas relaciones se ‘apegan’ al modelo conceptual” (p.4).

Para comprender un poco más este concepto Flores (2017) señala que: “La operacionalización de las variables es un concepto metodológico que consiste en descomponer o desagregar deductivamente las variables que componen el problema de investigación, partiendo desde lo más general a lo más específico” (p.173).

Variable Instrumental

Moreno (2018) explica en su blog online que, la definición instrumental de las variables es aquello en la que se aclara como se estudiará la variable que se acaba de definir, los medios o instrumentos para recoger la información. En mérito de ello deben definirse y elaborarse los instrumentos y medios con que se recolectará la información. Los instrumentos nacen de las variables y de los objetivos. Nunca deberá elaborarse un instrumento sin tener definida la variable o variables.

Cuadro 4. Análisis de las variables

Objetivo Específico	Variable	Variable Conceptual	Variable Operacional	Variable Instrumental
Analizar los requerimientos por implementar en el prototipo funcional según las necesidades de la empresa Tienda Karol.	Casos de uso. Requerimientos del usuario.	Casos de uso: Son una estrategia de captura de requisitos ampliamente aceptada frente a las deficiencias del lenguaje natural. (López, Laguna, y Corral, 2000, p.2). En términos de la informática, se trata de una técnica de recolección de requisitos para un software. Requerimiento: Función que debe cumplir el software para satisfacer las necesidades del usuario	Entrevistas con los administradores de la empresa.	Guías de entrevistas.
Diseñar según los requerimientos de Tienda Karol, la estructura del prototipo funcional.	Diseño del prototipo.	Diseño: El objetivo de la etapa de diseño es expresar los resultados abstractos del análisis, en especificaciones computacionales tangibles. (Díaz, Harari y Amadeo, 2013, p.46). Prototipo: El término prototipo se emplea para nombrar el	Diseño de las pantallas de cada módulo, diseño de la base de datos del sistema y la arquitectura de software.	Mockups.

		primer dispositivo que se desarrolla de algo y sirve como modelo para la fabricación de los siguientes o como muestra (Porto, J. & Merino, M. 2015).		
Programar la solución con el fin de cumplir con los requerimientos de la empresa.	Desarrollo del prototipo.	Realizar la programación con el fin de materializar los módulos requeridos por el usuario, tomando como base los resultados obtenidos con el diseño del prototipo.	Desarrollo del código fuente para las consultas, Pantallas, funciones y todos los requerimientos del sistema.	Visual Studio Community 2019, SQL Server Express 2017.
Probar las funcionalidades del prototipo desarrollado con el fin de validar el cumplimiento de los requerimientos de la empresa y el correcto funcionamiento de este.	Pruebas.	Fundamental para probar la corrección y completitud de etapas previas, como las etapas de requerimientos o la de diseño. A diferencia de la componente de aplicación, donde la evaluación se puede dejar al final del proceso de desarrollo, en el caso de la interfaz, la misma se aplica en todas las etapas. Se la utiliza para completar la etapa de requerimientos, para comprobar lo correcto de los bosquejos de diseño, para evaluar las versiones prototípicas, como también para revisar los aspectos de implementación. (Díaz, Harari y Amadeo, 2013, p.40).	Implementación de pruebas del sistema.	Casos de prueba

Fuente: Elaboración propia.

Población

Una población es el conjunto de elementos con características en común que pueden ser analizadas o estudiadas durante una investigación. López (2004) en su artículo

online, define el concepto de población como: “El conjunto de personas u objetos de los que se desea conocer algo en una investigación. El universo o población puede estar constituido por personas, animales, registros médicos, los nacimientos, las muestras de laboratorio, los accidentes viales entre otros” (p.69).

Para el desarrollo de este proyecto la población será todos los colaboradores que laboran en Tienda Karol, seis empleados y dos administradores.

Esto último debido a que el prototipo funcional estará orientado a la solución de los inconvenientes en los procesos de gestión del recurso humano en Tienda Karol, los cuales, perjudican tanto a los administradores como a los empleados, por ende, son la población seleccionada a la cual se le deben analizar y estudiar sus requerimientos y necesidades actuales.

Muestra

La muestra es una parte de la población, la cual tiene la capacidad de representarla y su uso es con el fin de rescatar a una parte del 100% de la población seleccionada con el propósito de minimizar tiempo y esfuerzo al investigar; por ende, la muestra debe ser bien seleccionada y representativa para que su investigación pueda ser generalizada a toda la población.

López (2004) indica que: “La muestra es un subconjunto o parte del universo o población en que se llevará a cabo la investigación. Hay procedimientos para obtener la cantidad de los componentes de la muestra como fórmulas, lógica y otros que se verá más adelante. La muestra es una parte representativa de la población” (p.107).

Según del Castillo y Orozco (2015) existen diferentes tipos y formas de muestreos: el muestreo probabilístico y el no probabilístico, a continuación, se describen estos tipos de muestreos.

- **El muestreo probabilístico** también es conocido como representativo y/o aleatorio, y consiste en que todos los individuos de la población que se estudia tienen la misma probabilidad de ser seleccionados, por lo que se debe tener acceso a todos los miembros involucrados, es decir, saber cuántos individuos hay y como entablar contacto con ellos (Somers 8C Somers, 2001).
- **Las muestras no probabilísticas** no representan con exactitud a la población. Sin embargo, resultan satisfactorias, y hasta deseables, para ciertos propósitos de investigación. Con mayor frecuencia obedecen a la imposibilidad de que los investigadores obtengan muestras representativas (Somers 8C Somers, 2001).

Muestreo no probabilístico

- **Muestreo accidental o por conveniencia:** El muestreo azaroso produce muestras altamente no representativas y no es recomendado (Neuman, 2000). Cuando el investigador selecciona los casos que son convenientes, se puede obtener fácilmente una muestra que no represente a la población, por lo que este tipo de muestras son poco costosas y rápidas de obtener; sin embargo, fácilmente ocurren errores sistemáticos: a pesar de que se recolecten un gran número de casos, no se podrán generalizar los datos a toda la población.
- **Muestreo por cuotas:** El muestreo por cuotas representa una mejora sobre el muestreo accidental, pues el investigador identifica las categorías relevantes de personas, ya sea por género o grupos de edad, y después decide cuántos buscar para cada categoría (Neuman, 2000). El investigador establece deliberadamente proporciones en la muestra que difieren de las que ocurren de modo natural en la población de acuerdo con los objetivos de la investigación. Su uso es para garantizar que determinados miembros de la población estén representados (Somers & Somers, 2001), lo cual significa una mejora al muestreo accidental, ya que se puede asegurar que algunas diferencias están en la muestra (Neuman, 2000). Una vez establecidas las cuotas, los individuos se seleccionan accidentalmente o por conveniencia.
- **Muestreo propositivo o intencional:** Es un tipo aceptable de muestreo, debido a que usa el juicio de un experto en la selección de los casos o con propósito específico en mente (Neuman, 2000). Apunta a los individuos que se consideran más relevantes para el tema de investigación (Somers & Somers, 2001) y es utilizado en investigaciones exploratorias y de campo. Este muestreo es apropiado en tres situaciones:
- **Muestreo “bola de nieve”:** Consiste en un tipo especial de muestreo propositivo (Somers & Somers, 2001) o un método para identificar casos en una red (Neuman, 2000). El investigador pide a los entrevistados que proporcionen contactos adicionales, donde se estudian grupos particulares de difícil acceso (Somers & Somers, 2001),

cuando los investigadores están interesados en una red interconectada de personas u organizaciones.

- **Muestreo de caso anormal o de caso extremo:** Se utiliza cuando el investigador busca casos que difieran del patrón dominante o de las características predominantes de otros casos. De manera similar que el muestreo propositivo, un investigador utiliza una variedad de técnicas para localizar casos con características específicas, pues su meta es localizar una colección de casos inusuales, diferentes y/o peculiares que no sean representativas del todo. Los casos anormales son seleccionados debido a que son inusuales y el investigador espera aprender más acerca de la vida social al considerar que caen fuera del patrón general.
- **Muestreo secuencial:** Es semejante al muestreo propositivo, pero con la variante de que el investigador continúa reuniendo casos hasta que la información o la diversidad de casos llegan a tal grado que el punto de saturación es alcanzado.
- **Muestreo teórico:** El interés del crecimiento teórico guía la selección de los casos de la muestra; es ahí donde el investigador selecciona los casos basado en la contribución teórica que puedan hacer.

Muestreos probabilísticos

- **Muestreo aleatorio simple:** Cada caso en la población meta tiene las mismas probabilidades de selección (Coolican, 2005).
- **Muestreo aleatorio sistemático:** Se selecciona cada n ésimo caso de la población, donde n es cualquier número. Antes de empezar la selección cada caso debe tener las mismas probabilidades de estar en la muestra. Para hacer que en esos métodos se utilice el muestreo con igualdad de probabilidades debe elegirse el punto de partida en forma aleatoria (Coolican, 2005).
- **Muestreo estratificado:** Una muestra estratificada es una variación de la muestra aleatoria: en lugar de seleccionar todos los casos al azar, se establecen criterios para asegurarse de que ciertos grupos estén

representados en la misma proporción que sus números en la población (Somers 8C Somers, 2001).

Para este proyecto se utiliza un muestreo probabilístico, debido a que busca que todos los individuos de la población cuenten con una probabilidad mayor a cero de ser seleccionados en la muestra.

Existen fórmulas matemáticas ya definidas para el cálculo de la muestra, a continuación, en la imagen No. 3 se muestra la fórmula que utilizada en este proyecto:

Imagen 3. *Formula para el cálculo de la muestra*

$$n = \frac{Z^2 \sigma^2 N}{e^2(N-1) + Z^2 \sigma^2}$$

Fuente: (QuestionPro, 2020)

Definición de variables que componen la fórmula

A continuación, se muestran cada una de las variables que constituyen la fórmula para el cálculo de la muestra y una breve descripción:

n = es el tamaño de la muestra poblacional a obtener.

N = es el tamaño de la población total.

σ = representa la desviación estándar de la población, en caso de no conocer este dato es común hacer uso del valor 0.5 como constante.

Z = es el valor obtenido mediante niveles de confianza. El valor de Z es constante y varía dependiendo el grado de confianza que se desee, siendo 99% el valor más alto (equivalente a 2.58) y 95% (1.96) como el valor mínimo aceptado para considerar el resultado esperado como confiable.

E = representa el límite aceptable de error muestral, generalmente comprende un rango que va desde el 1% (0.01) al 9% (0.09), siendo 5% (0.05) el valor estándar usado en investigaciones.

Cálculo de la muestra

A continuación, se muestra el cálculo de la muestra:

$$n = 2.58^2 \times 0.5 \times 0.5 \times 6 / 0.05 (6-1) + 2.58^2 \times 0.5 \times 0.5$$

$$n=9.9846/ 0.05 * 5 + 2.58^2 \times 0.5 \times 0.5$$

$$n=9.9846/0.25 + 2.58^2 * 0.25$$

$$n=9.9846/0.25+ 1.6641$$

$$n= 9.9846/ 0.25 + 1.6641$$

$$n=9.9846/1.9141$$

$$n=5.21$$

Según el resultado obtenido en el cálculo anterior la muestra es de 5.21, el valor de nivel de confianza 99% (2.58), la probabilidad de fracaso de 5% (0.5) y la precisión de un 5% (0.5), sin embargo, debido a que no existen fracciones de persona y la muestra se acerca a la población total, se toma la población total para ejecutar el estudio, es decir 6 personas.

Instrumentos de recolección de datos

El autor Arias (2012) señala en su investigación que: “Un instrumento de recolección de datos es cualquier recurso, dispositivo o formato (en papel o digital), que se utiliza para obtener, registrar o almacenar información” (p.63).

En función de este concepto los instrumentos apoyarán la recolección de datos para el desarrollo correcto y eficaz del prototipo funcional.

Entrevista.

Una entrevista, se trata de un diálogo entre dos personas, en donde el entrevistador realiza preguntas al informante, con el fin de conocer algún aspecto acerca de un tema en específico; así Peláez y otros (2013) citan que una entrevista, es un proceso de comunicación que se realiza normalmente entre dos personas; en este proceso el entrevistador obtiene información del entrevistado de forma directa. Si se generalizara una entrevista sería una conversación entre dos personas por el mero hecho de comunicarse, en cuya acción la una obtendría información de la otra y viceversa. En tal caso los roles de entrevistador / entrevistado irían cambiando a lo largo de la conversación. La entrevista no se considera una conversación normal, sino una

conversación formal, con una intencionalidad, que lleva implícitos unos objetivos englobados en una Investigación.

A continuación, se exponen los dos tipos de entrevista existentes, la entrevista no estructurada y la entrevista estructurada. Por sus nombres se puede comprender por entrevista no estructurada, como aquella que no cuenta con un orden establecido, mientras que la entrevista estructurada es un poco más estática y definida.

- Entrevista no Estructurada: según Peláez et al. (2013): La entrevista se va construyendo a medida que avanza la entrevista con las respuestas que se dan. Requiere gran preparación por parte de investigador, documentándose previamente sobre todo lo que concierne a los temas que se tratan.
- Entrevista Estructurada: Peláez et al. (2013): El investigador planifica previamente las preguntas mediante un guion preestablecido, secuenciado y dirigido, por lo que dejan poca o ninguna posibilidad al entrevistado de réplica o de salirse del guion.

Para este proyecto, se efectúa una entrevista estructurada, debido a que se planificará con anticipación las preguntas a los entrevistados.

Encuesta.

Una encuesta es un cuestionario diseñado previamente con preguntas dirigidas a información que se desea rescatar acerca de un tema, según la RAE (2020), una encuesta es: “el conjunto de preguntas tipificadas dirigidas a una muestra representativa de grupos sociales, para averiguar estados de opinión o conocer otras cuestiones que les afectan.”

Encuesta Descriptiva.

Una encuesta descriptiva busca describir una realidad de un tema en específico, según el sitio web Insignia (2017) las encuestas descriptivas, reflejan o documentan las actitudes o condiciones presentes. Esto significa que intentan describir en qué situación se encuentra una determinada población en el momento en que se realiza la encuesta.

Encuesta Analítica.

Este tipo de encuestas busca desglosar una problemática para llegar a un mejor análisis, Insignia (2017) detalla en su sitio web que, este tipo de encuestas, buscan además de describir, explicar los porqués de una determinada situación. En este tipo de encuestas las hipótesis que las respaldan suelen contrastarse por medio del examen de por lo menos dos variables, de las que se observan interrelaciones y luego se formulan inferencias explicativas.

Proceso para la recolección y análisis de datos

Para recolectar la información necesaria, se aplicaron dos instrumentos: las entrevistas y las encuestas.

La entrevista se realizó de forma presencial en Tienda Karol y su objetivo fue recolectar los requerimientos del sistema y conocer más a fondo el proceso de gestión del recurso humano actual en la empresa, por ejemplo, como están trabajando las solicitudes de vacaciones, el listado de la planilla, los contratos de los empleados y demás temas relacionados al proceso del recurso humano, todo esto con el fin de analizar los beneficios del desarrollo del prototipo funcional, y los requerimientos de este. Con base en lo antes indicado, esta entrevista se realizó a los administradores de la empresa, ya que estos cuentan con amplio conocimiento de los procesos de gestión actuales.

La encuesta fue dirigida al personal de la empresa con la ayuda de la herramienta gratuita Google Forms, la cual permite recopilar las respuestas de forma fácil y eficiente, las preguntas de esta encuesta son cerradas y puntuales, con el fin de conocer las opiniones del personal de la empresa con respecto a la implementación de un sistema y de qué manera piensan que los beneficiaría. El instrumento aplicado fue de tipo analítico, el cual fue realizado en forma de cuestionario y los resultados obtenidos serán analizados, con el fin de velar por el cumplimiento de los requerimientos del sistema y la solución a la problemática actual en la empresa Tienda Karol.

CAPÍTULO IV: ANÁLISIS DE RESULTADOS

Como ya se indicó, para este proyecto se utilizaron las herramientas de recolección de datos entrevista y encuesta, las cuales fueron aplicadas en Tienda Karol.

Entrevista

La entrevista tiene como objetivo conocer el proceso actual para el control del recurso humano en Tienda Karol, además de los problemas actuales en este proceso y los requerimientos para el prototipo web, según las necesidades de la empresa en estudio. La entrevista aplicada cuenta con 10 preguntas abiertas y fue dirigida a la administradora y propietaria de Tienda Karol.

La entrevista dio inicio con la pregunta: Actualmente, ¿cuál es el proceso para la gestión del recurso humano en Tienda Karol? A lo que la administradora responde que: en la actualidad el proceso se realiza de forma totalmente manual, además indica que “Desde sus inicios se ha realizado de esta manera, sin embargo, siempre ha querido optar por la implementación de soluciones tecnológicas para sus procesos. Menciona que antes no ha tenido la oportunidad de realizar el cambio tecnológico debido a la falta de presupuesto”. Aunado a ello, “que no había tenido la oportunidad de plantear un sistema que fuera factible y funcional para las necesidades con las que cuentan.”

En seguida, se le consulta acerca de la problemática actual por medio de la segunda pregunta: ¿Cuáles son los inconvenientes o problemas principales con el proceso actual? A lo que responde que: “A lo largo de los años y con el crecimiento de la empresa, la constante siempre ha sido la confusión entre los pagos al personal y el proceso de cálculo de la planilla mensual, esto debido a que le demanda mucho tiempo, además comentó que mucha información se perdía y los cálculos no eran del todo precisos.”

Ante su interés de implementar la tecnología en su empresa, se le consulta a la administradora: ¿Alguna vez se ha informado o ha estado en contacto con algún tipo de sistema informático? A lo que contesta que: “Realmente años atrás se le dificultaba mucho y además afirma que por esa razón en la Tienda todos los procesos se ejecutan de forma manual, pero con el pasar del tiempo ha logrado familiarizarse al cambio tecnológico, obteniendo contacto con sistemas informáticos y herramientas de software.”

En la cuarta pregunta se plantea: ¿Cree que un sistema informático puede solventar la problemática actual en el proceso de gestión del recurso humano? A la cual responde afirmativamente e indica que: “Cree firmemente en que un sistema informático

puede automatizarle mucho su trabajo y solucionar muchos problemas administrativos internos.”

En la quinta pregunta, con el fin de conocer los recursos tecnológicos de Tienda Karol se planteó la pregunta: ¿Actualmente, cuentan con equipo de cómputo en la compañía? A lo que la administradora respondió que: “Recientemente Tienda Karol adquirió equipo de cómputo básico, ya que se determinó la necesidad inmediata de implementar recursos tecnológicos para sus gestiones y se encuentran en esa transición, actualmente cuentan con dos equipos de escritorio, ambos con sus respectivos monitores y demás periféricos.”

Posteriormente, se procede a consultarle a la administradora lo siguiente: ¿Estaría dispuesta a invertir en la implementación de este tipo de herramientas? A lo que responde afirmativamente, al indicar que: “Su visión es optar por la automatización de sus procesos y la reducción de costos apoyándose de la tecnología.”

Entonces, se procede a la pregunta número siete, en la cual se le consulta específicamente por el proceso de control del recurso humano: ¿Qué beneficios esperaría con la implementación de un prototipo para la gestión del recurso humano? En su respuesta explica que: “Espera un gran cambio con respecto a las horas que invierte en este proceso, que dicho proceso sea mucho más ágil, preciso en la información y que le brinde a la empresa más orden y control sobre su recurso humano.”

Dado que el prototipo propuesto se trata de una solución web, se procede a preguntar lo siguiente: ¿Actualmente, cuentan con servicio de internet en Tienda Karol? La propietaria indica que sí, explica que: “Fue adquirido recientemente, debido a que se vieron obligados a colocar cámaras de seguridad en la Tienda y fue necesario adquirir el servicio para el monitoreo vía wifi, por lo que contando con internet la futura implementación del prototipo no tendría un costo adicional por el servicio.”

Luego, y con el fin de conocer un poco sobre los requerimientos del sistema para el control del recurso humano, se procede con la siguiente pregunta: En síntesis, ¿cuáles son las características y requerimientos que debe contemplar el sistema para la solución de la problemática actual? A lo que la administradora indica que: “Requiere que la herramienta le permita calcular la planilla automáticamente de forma mensual y que además espera que logre involucrar a los colaboradores en la herramienta y que puedan colocar sus solicitudes, justificaciones y horas de llegada al local para tener un mayor control sobre el personal. Explica que el poder registrar información de contratos es muy importante ya que este tipo de información es vital, también indicó que le interesa que se

calculen las vacaciones disponibles por empleado y que pueda generar reportes para la toma de decisiones, ya que, actualmente no cuenta con reportes precisos.”

Para finalizar la entrevista y luego de conocer más a fondo qué espera la propietaria de Tienda Karol de un sistema para el control de su recurso humano se plantea la última pregunta: ¿Estaría dispuesta a cambiar sus procesos manuales, por un sistema web que los ejecute de manera automatizada? A lo que responde que sí, que “definitivamente manejar sus procesos por medio de un sistema le facilitaría mucho su trabajo y reduciría en gran medida las horas que invierte en el proceso gracias a la automatización que puede brindar, explica que, aunque sería un proceso de adaptación a una nueva forma de trabajar, está dispuesta a cambiar sus procesos manuales para el beneficio y solución de tantos problemas actuales.”

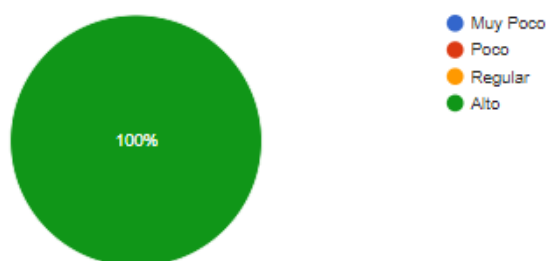
Encuesta

Como se mencionó anteriormente, para la aplicación de la encuesta, se utilizó la herramienta gratuita Google Forms, la encuesta fue dirigida hacia los empleados de Tienda Karol y consta de ocho preguntas de carácter cerrado que buscan conocer la posición de los colaboradores ante la implementación de un sistema para el control del recurso humano.

Gráfica 1. *Pregunta 1*

¿En su rol de colaborador, cuánto cree usted que le beneficiaría la implementación de un sistema informático en Tienda Karol?

6 respuestas



Fuente: (Elaboración propia).

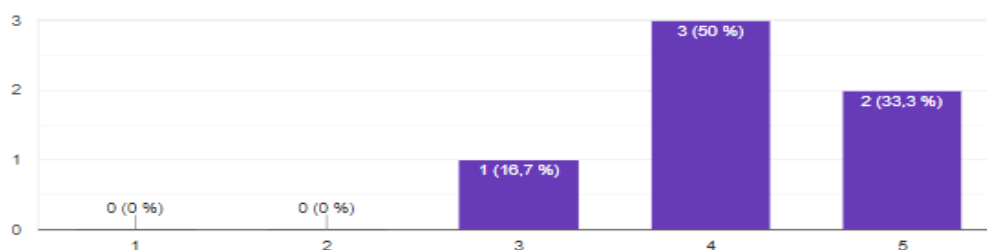
En la primera pregunta de la encuesta, el 100% de los empleados respondieron que consideran que la implementación de un sistema en Tienda Karol les traería un gran

beneficio, lo cual es importante de cara a una futura implementación del sistema para el control del recurso humano dado que, tendría la aceptación y el visto bueno de los empleados.

Gráfica 2. *Pregunta 2*

2- ¿Del 1 al 5, donde 1 significa que no tiene experiencia y 5 significa muy experimentado, que tanta experiencia tiene en el uso de herramientas de software y computadoras?

6 respuestas

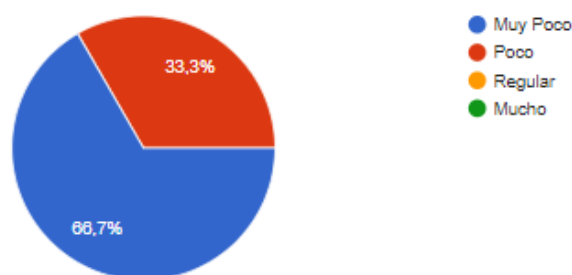


Fuente: (Elaboración propia).

Como lo muestra la gráfica No. 2, los resultados de la segunda pregunta evidencian una buena experiencia en el uso de herramientas de software y computadoras por parte de los colaboradores, lo cual es de suma importancia, debido a que ante una eventual puesta en producción del prototipo no se les debería complicar su uso; sin embargo, de ser necesario con capacitaciones y práctica sus habilidades pueden mejorar.

¿Qué tan eficiente cree usted que sea proceso actual de la gestión del recurso humano?

6 respuestas



Gráfica 3. *Pregunta 3*

Fuente: (Elaboración propia).

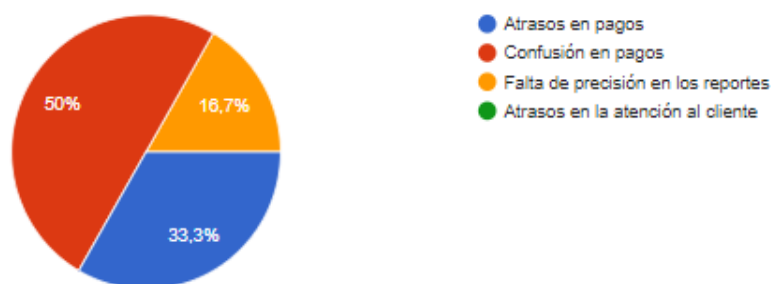
En la tercera pregunta, como lo muestra el gráfico No. 3, los empleados calificaron el proceso actual de la gestión del recurso humano como muy poco y poco eficiente, al ser la opción “muy poco” la de mayor porcentaje, lo que evidencia la disconformidad de los colaboradores con el proceso actual. Uno de los principales propósitos del prototipo

funcional es mejorar en gran medida el proceso de gestión del recurso humano, a tal medida que tanto colaboradores como administradores de Tienda Karol estén conformes con el proceso y les brinde soluciones a los inconvenientes actuales.

Gráfica 4. *Pregunta 4*

¿Según la problemática actual, cual situación de las siguientes opciones es más común?

6 respuestas



Fuente: (Elaboración propia).

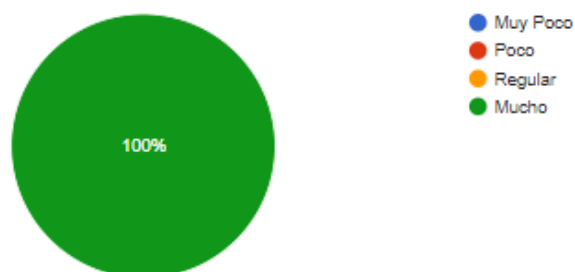
Los resultados obtenidos de la pregunta No. 4 se ilustran en la gráfica No. 4 y muestran variedad en las respuestas; sin embargo, la opción “confusión en pagos” obtuvo el mayor porcentaje de respuesta.

Esta situación es la más común y la que provoca mayor disconformidad por parte de los colaboradores de Tienda Karol, con la propuesta de este prototipo funcional se busca solucionar esta problemática por medio de un módulo de planilla que brinda a los administradores la opción de calcular el monto a pagar correspondiente a cada uno de los empleados, el prototipo funcional busca solucionar esta y todas las demás situaciones que generan actualmente problemas a lo interno de Tienda Karol, con el fin de que sus procesos de gestión con respecto al recurso humano mejoren y consecuentemente genere conformidad con el proceso por parte de los colaboradores y administradores.

Gráfica 5. *Pregunta 5*

¿Qué tanto cree que beneficiaría la implementación de un sistema informático para el proceso de gestión del recurso humano?

6 respuestas



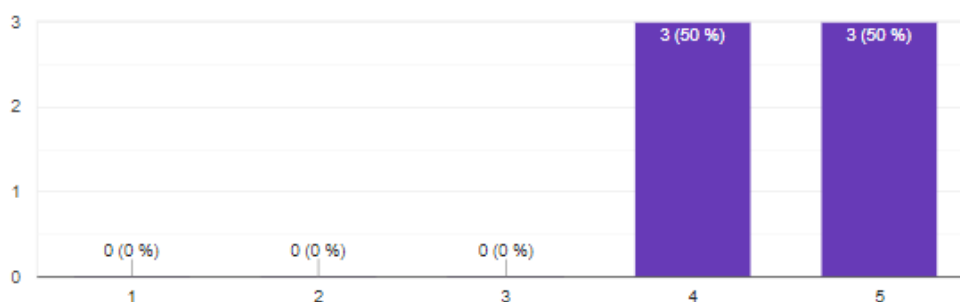
Fuente: (Elaboración propia).

En la pregunta número cinco, y según observa en el gráfico No. 5, la totalidad de los colaboradores de Tienda Karol creen que un sistema para el proceso de gestión del recurso humano beneficiaría mucho en Tienda Karol, lo cual ratifica la posición favorable por parte de los colaboradores a la implementación de este tipo de sistema.

Gráfica 6. *Pregunta 6*

Recientemente Tienda Karol adquirió equipo tecnológico, ¿del 1 al 5 que tanto cree usted que dicho equipo sea apto para un sistema de gestión del recurso humano?, siendo 1 muy poco apto y 5 muy apto.

6 respuestas



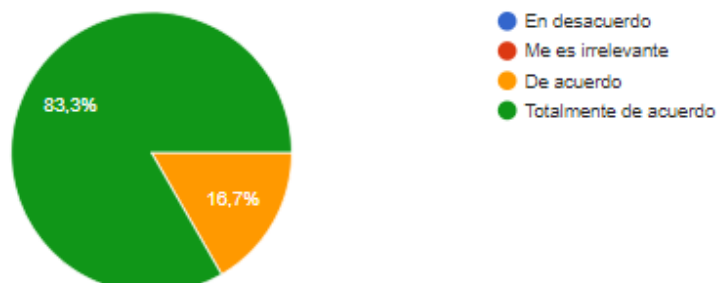
Fuente: (Elaboración propia).

En la sexta pregunta, como se muestra en el gráfico No. 6; los colaboradores opinan positivamente sobre el equipo recién adquirido en Tienda Karol y lo califican como muy apto para la implementación de un sistema, y ello es correcto, sería una gran ventaja dado para que la empresa no tendría que invertir en mejoras al equipo actual o inclusive en la compra de máquinas nuevas ante una futura implementación del prototipo propuesto.

Gráfica 7. *Pregunta 7*

¿Estaría usted de acuerdo en que el proceso de gestión del recurso humano en Tienda Karol se realice de forma más automatizada por medio de un software?

6 respuestas



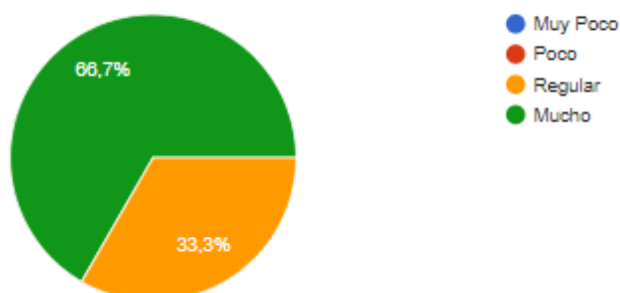
Fuente: (Elaboración propia).

En la séptima pregunta, se obtuvieron resultados que evidencian que los colaboradores están totalmente de acuerdo con que el proceso de gestión del recurso humano en Tienda Karol se realice de forma automatizada por medio de un software, tal y como se muestra en el gráfico No. 7.

Gráfica 8. *Pregunta 8*

¿Qué tan importante cree que sea la implementación de la tecnología en Tienda Karol?

6 respuestas



Fuente: (Elaboración propia).

En la octava y última pregunta se les consulta a los colaboradores si creen que es importante la implementación de la tecnología en Tienda Karol. Tal como se ha ilustrado, en el gráfico No. 8 se evidencia que un 66,7% de los colaboradores opina que es muy importante y el 33,3% opina que la implementación de la tecnología tiene una importancia regular. Lo cual reafirma el interés por la innovación tecnológica por parte de los empleados de Tienda Karol.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

A continuación, se exponen las conclusiones y recomendaciones obtenidas durante el desarrollo del prototipo funcional para el control y gestión del recurso humano en Tienda Karol.

Conclusiones

Con base en los objetivos planteados se logra concluir lo siguiente:

Gracias al apoyo de los administradores y empleados de la Tienda Karol se logró realizar el proceso de análisis de requerimientos del prototipo funcional para el control y gestión del recurso humano en Tienda Karol. Por medio de distintas reuniones presenciales se logró conocer a detalle el proceso y la problemática actual.

Lo mencionado anteriormente de manera conjunta con las entrevistas y cuestionarios realizados, benefició al proceso de análisis, facilitando el planteamiento de los distintos requerimientos del prototipo funcional que logra corregir los inconvenientes en el proceso de gestión del recurso humano en Tienda Karol de tal modo que el análisis se llevó a cabo de manera satisfactoria.

Por otra parte, se concluye que el diseño de la arquitectura del sistema desarrollado se realizó de forma correcta, tomando en cuenta las necesidades y requerimientos planteados, así como el diseño de la base de datos, diseñada a la medida para el almacenamiento estructurado de la información de Tienda Karol.

El diseño de la interfaz gráfica se procuró que fuera amigable con el usuario, además, por petición de los administradores utiliza los colores corporativos y cada una de sus opciones se encuentran organizadas a su gusto, dando como resultado un diseño de interfaces satisfactorio.

Con la codificación realizada, se concluye que la misma cumple con los requerimientos de la empresa y brinda las funcionalidades necesarias. La lógica implementada permite la automatización de los procesos actuales por medio de los distintos módulos creados y brinda como resultado, la solución de la problemática actual en los procesos de gestión del recurso humano.

Sumado a ello, con el fin de corroborar el correcto funcionamiento del sistema, el prototipo fue puesto a prueba, en dichas pruebas se logra identificar que los módulos cumplen con los requerimientos y necesidades de Tienda Karol, de tal forma el proceso de pruebas de funcionamiento fue satisfactorio.

Recomendaciones

En este apartado se detallan recomendaciones con base en los resultados y las conclusiones a las cuales se llegó luego del desarrollo de este proyecto. Estas sugerencias fueron recolectadas con el fin de proponer puntos de mejora al prototipo desarrollado y a Tienda Karol, para que puedan ser aplicados en un futuro si se desea.

En primer lugar, se recomienda a los administradores de Tienda Karol, la futura implementación del prototipo desarrollado, a cargo de los administradores de la empresa, dentro de un periodo de tiempo de 2 a 3 meses máximo, con el fin de solucionar la problemática actual en el proceso de control y gestión del recurso humano, el proceso de implementación tendría una duración aproximada de 2 meses.

Ante una futura implementación de este prototipo y pasada 1 semana de la finalización de dicho proceso, se recomienda un plan de capacitación de 1 semana de duración, los administradores de la empresa serían los encargados de velar porque este proceso se lleve a cabo con apoyo del desarrollador de la herramienta, la capacitación debe ser impartida para cada uno de los usuarios que harán uso del sistema, dado que, deben conocer a fondo su funcionamiento y cada una de sus opciones para su uso correcto.

El objetivo de una herramienta de este tipo es beneficiar y mejorar los procesos de la empresa, no crear confusiones o que su uso se complique por su desconocimiento, situación que se logra evitar con una capacitación de uso impartida por el desarrollador de la herramienta e incluso con la creación de un manual de uso creado sin costo alguno por el desarrollador.

Además, aunque el proceso actual trae como consecuencia muchos inconvenientes, los colaboradores de Tienda Karol se encuentran acostumbrados a él, durante muchos años han manejado este proceso de la misma manera, por lo que se recomienda un periodo de adaptación o prueba ante el cambio en el proceso de 1 mes, a partir del momento en que el sistema se implemente, con responsabilidad de ejecución por parte de los administradores de Tienda Karol, utilizando la herramienta y adaptándose a ella. Esto con el fin de que se cree una nueva costumbre y noten los beneficios del uso de la herramienta y sus procesos automatizados.

También es recomendable que en caso de implementación y pasadas 2 semanas de la finalización del proceso de pruebas, se cuente con un especialista a cargo del mantenimiento del aplicativo y sus componentes, como base de datos, servidores, red, entre otros. Esto para evitar cualquier incidente en la herramienta.

El especialista a cargo se recomienda que cuente con experiencia en programación ASP.NET y administración de bases de datos, actualmente Tienda Karol no cuenta con dicho personal, por lo tanto, se deberá hacer un proceso de contratación de dicho especialista por parte de los administradores de la compañía.

Posterior a la contratación de un especialista a cargo, se sugiere también el desarrollo de un web api para el sistema por parte de dicho especialista, pensando en que en un futuro se implementen más herramientas de software en la empresa y se requiera de una integración con el sistema de control y gestión del recurso humano.

CAPÍTULO VI PROPUESTA

Análisis

Análisis detallado del software desarrollado

A continuación, se detallarán los distintos módulos desarrollados y sus funcionalidades para el prototipo funcional del recurso humano en Tienda Karol.

Seguridad

Este módulo cuenta con dos opciones, las cuales se describen a continuación:

- Configuración de usuarios: Al ingresar a esta opción se podrá visualizar la información de todos los usuarios registrados y será posible la creación, edición o eliminación de los usuarios en el sistema. En la creación de cada usuario se debe asignar su rol, el sistema cuenta con dos tipos de roles, Administrador o empleado, el rol de administrador cuenta con acceso a todas las funcionalidades del sistema mientras que el rol de empleado únicamente cuenta con acceso a algunas funcionalidades. Únicamente los usuarios con rol de administrador pueden visualizar esta opción.
- Marcas: En esta opción todos los usuarios pueden registrar sus marcas de entrada al local y el sistema se encarga de obtener la fecha y hora de la marca, tanto administradores como empleados pueden acceder a esta opción.

Mantenimiento

En este módulo los usuarios con rol de administrador pueden realizar el mantenimiento de las personas, empleados y los catálogos de la herramienta. Este módulo se divide entre un submódulo llamado gestión de campos y las opciones personas y empleados, a continuación, se describen las opciones en detalle.

- Gestión de campos: en este submódulo se encuentran las siguientes opciones para el mantenimiento de cada uno de los catálogos que utilizará la herramienta:
 - Gestión Tipos Solicitud: en esta opción se pueden visualizar, crear, editar o eliminar los tipos de solicitudes con las que trabajará la herramienta.
 - Gestión Tipos Justificación: en esta opción se pueden visualizar, crear, editar o eliminar los tipos de justificaciones con las que trabajará la herramienta.

- Gestión Tipos Contratos: en esta opción se pueden visualizar, crear, editar o eliminar los tipos de contratos con las que trabajará la herramienta.
 - Gestión Estados Solicitud: en esta opción se pueden visualizar, crear, editar o eliminar los tipos estados de solicitud con las que trabajará la herramienta.
 - Gestión Estados Justificación: en esta opción se pueden visualizar, crear, editar o eliminar los tipos estados de justificación con las que trabajará la herramienta.
 - Gestión de Horarios: en esta opción se pueden visualizar, crear, editar o eliminar los horarios con las que trabajará la herramienta para ser asignados a los contratos que se deseen.
 - Gestión de Conceptos: en esta opción se pueden visualizar, crear, editar o eliminar los conceptos que utilizará la herramienta como porcentajes de rebaja o carga a los contratos que se deseen.
 - Gestión del ISR: en esta opción se puede realizar la edición de la tabla para calcular impuesto sobre la renta que utilizará la herramienta. Los montos para el cálculo del impuesto sobre la renta cambian anualmente según el Ministerio de Hacienda de Costa Rica, por lo que es importante contar con la edición de estos campos.
- Personas: en esta opción, el usuario puede visualizar las personas registradas en la herramienta y agregar, editar o eliminar registros.
 - Empleados: en esta opción el usuario puede visualizar los empleados en la herramienta y agregar, editar o eliminar registros.

Planilla

En este módulo los usuarios con el rol de administradores pueden llevar sus procesos de gestión de planilla, vacaciones, llegadas tardías y liquidaciones. A continuación, se describen las opciones a detalle.

- Gestor de planilla: en esta opción los usuarios administradores pueden calcular la planilla de forma mensual, visualizar el historial de las planillas generadas

con toda su información y el detalle de deducciones por impuesto sobre la renta y conceptos asignados.

- **Calcular llegadas tardías:** en esta opción el sistema calcula automáticamente las llegadas tardías, según las marcas registradas.
- **Gestor de vacaciones:** por medio de esta opción los usuarios administradores pueden visualizar la lista de empleados con sus días de vacaciones disponibles, el sistema automáticamente contabiliza los días disponibles de cada empleado, sumando un día de vacación por mes laborado y restando según las asignaciones que se realicen. Los usuarios por medio de esta opción pueden asignar vacaciones a un empleado seleccionado según los días de vacación que este tenga disponibles, además, podrá visualizar el histórico vacacional del empleado que desee.
- **Liquidaciones:** por medio de esta opción los usuarios con rol de administrador podrán realizar el proceso de liquidación de algún empleado y el sistema calculará el monto que le corresponde al funcionario por su liquidación.

Justificaciones

En este módulo los usuarios con el rol de empleado o administrador pueden registrar sus justificaciones con su fecha, descripción y tipo de justificación siendo este último un catálogo configurado previamente en el módulo de mantenimiento, según los tipos de justificación que requiere Tienda Karol, cada vez que se registre una nueva justificación se notificará a los administradores vía correo, a continuación, se describen las opciones de este módulo.

- **Gestor de justificaciones:** en esta opción se pueden registrar las nuevas justificaciones, además se enlista el histórico de justificaciones. Si se trata de un usuario administrador, visualizará el histórico general de todas las justificaciones; mientras que si se trata de un usuario con rol de empleado únicamente visualizará el histórico de la que él ha registrado.
- **Estados de justificaciones:** esta opción solo estará disponible para los usuarios con el rol de administrador y con ella podrán aceptar, denegar o asignarles a las justificaciones cualquiera de los estados configurados en el módulo de mantenimiento.

Solicitudes

En este módulo los usuarios con el rol de empleado o administrador pueden registrar sus solicitudes con su fecha, descripción y tipo de solicitud, al ser este último un catálogo configurado previamente en el módulo de mantenimiento, según los tipos de solicitud que requiere Tienda Karol, cada vez que se registre una nueva solicitud se notificará a los administradores vía correo, a continuación, se describen las opciones de este módulo.

- Gestor de solicitudes: en esta opción se pueden registrar las nuevas solicitudes, además se enlista el histórico de justificaciones. Si se trata de un usuario administrador visualizará el histórico general de todas las justificaciones; mientras que si se trata de un usuario con rol de empleado únicamente visualizará el histórico de solicitudes que él ha registrado.
- Estados de solicitudes: esta opción solo estará disponible para los usuarios con el rol de administrador y con ella podrán aceptar, denegar o asignarles a las solicitudes cualquiera de los estados configurados en el módulo de mantenimiento.

Reportes

En este módulo el usuario puede realizar consulta de cualquiera de los reportes que requiere la empresa. Entre estos reportes se encuentran:

- Reporte general de pagos.
- Reporte de porcentajes por intervalo de edades.
- Reporte de pagos por empleado.
- Reporte de cantidad de llegadas tardías por empleado.
- Reporte de cantidad de ausencias por empleado.
- Reporte de solicitudes por empleado.
- Reporte de justificaciones por empleado.

Análisis detallado del hardware requerido

A continuación, en el cuadro No. 5 se detalla el hardware requerido, tanto para la programación del prototipo como para una futura puesta en producción del sistema para el control del recurso humano en Tienda Karol y sus costos.

Cuadro 5. *Requerimientos de hardware*

Hardware requerido para el desarrollo	Hardware requerido para los equipos en producción	Costos
Computador con las siguientes características: RAM: 4Gb (mínimo). Disco Duro: 500Gb. Procesador: 64 bits - 1,6 GHz o superior.	Computador con las siguientes características: RAM: 4Gb (mínimo). Disco Duro: 500Gb. Procesador: 64 bits - 1,6 GHz o superior.	No existen costos adicionales dado que las máquinas con las que cuenta Tienda Karol cumplen con los requisitos.

Fuente: (Elaboración propia).

Cabe aclarar que, si el prototipo se llegara a colocar en producción, se debe realizar la publicación, tanto de la solución web como de su base de datos. Para ello existen servicios de alojamiento en la nube que no requieren de un hardware o equipo físico adicional; no obstante, este tipo de servicio puede tener un costo mensual o anual; dependiendo del proveedor su costo puede variar. Existen servicios de alojamiento en la nube gratuitos, aunque, muchos de ellos pueden contar con limitantes de almacenamiento. A continuación, se detallan algunos ejemplos de servicios de alojamiento en la nube, los cuales se podrían utilizar ante una eventual puesta en producción del sistema:

- **Microsoft Azure:** Azure es una de las plataformas de computación en la nube más famosas del mercado, fue creada por la compañía Microsoft y ofrece muchos servicios entre los cuales se encuentran la publicación de aplicativos webs y de bases de datos, Microsoft ofrece estos servicios en forma de distintos paquetes modulares. A continuación, en el cuadro No. 6 y No. 7 se visualizan los paquetes y sus precios correspondientes con detalle.

Cuadro 6. Costos Azure por App Service

	GRATIS	COMPARTIDA	BASIC	ESTANDAR	PREMIUM	AISLADO
	Pruébelo gratis	Entorno de desarrollo y pruebas	Entorno dedicado a desarrollo y pruebas	Ejecutar cargas de trabajo de producción	Rendimiento y escala mejorados	Alto rendimiento, mayor seguridad y aislamiento
Aplicaciones web, móviles o de API	10	100	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Espacio en disco	1 GB	1 GB	10 Gb	50 GB	250 GB	1 TB
N° máximo de instancias	*	*	Hasta 3	Hasta 10	Hasta 30	Hasta 100
Dominio personalizado	*	Admitido	Admitido	Admitido	Admitido	Admitido
Escalado automático	*	*	*	Admitido	Admitido	Admitido
Conectividad híbrida de VPN	*	*	*	Admitido	Admitido	Admitido
Aislamiento de red	*	*	*	*	*	Admitido
Precio	gratis	\$9,49/mes	\$54,75/mes	\$73/mes	\$146/mes	\$292/mes

Fuente: (Microsoft,2020).

Cuadro 7. Costos Azure por Azure SQL Database Estándar

DTU	Almacenamiento Incluido	Almacenamiento Máximo	Precio de DTU y almacenamiento
10	250 GB	250 GB	\$14,7187/mes
20	251 GB	251 GB	\$29,4343/mes
50	252 GB	252 GB	\$73,6084/mes
100	253 GB	1 TB	\$147,1772/mes
200	254 GB	2 TB	\$294,3725/mes
400	255 GB	3 TB	\$588,7450/mes
800	256 GB	4 TB	\$1177,4900/mes
1600	257 GB	5 TB	\$2354,9800/mes
3000	258 GB	6 TB	\$4415,5875/mes

Fuente: (Microsoft,2020).

- GearHost: GearHost es una compañía que ofrece servicios de almacenamiento en la nube para soluciones de lenguajes de programación como PHP, Node.js, ambientes .NET Core y también para bases de datos Microsoft SQL 2017 o MySQL 5, lo cual se ajusta al sistema para el control y gestión del recurso humano desarrollado. GearHost ofrece sus servicios en forma de distintos paquetes modulares, que se pueden visualizar en el cuadro No 8.

Cuadro 8. Costos servicios de almacenamiento GearHost

FREE	HOBBY	SMALL	MEDIUM	LARGE
1 CPU	1 CPU	1 CPU	2 CPU	4 CPU
1 Max Work	10 Max Workers	10 Max Workers	10 Max Workers	10 Max Workers
256 MB Application Pool	1 GB Application Pool	2 GB Application Pool	4 GB Application Pool	6 GB Application Pool
5% CPU Usage	15% CPU Usage	100% CPU Usage	100% CPU Usage	100% CPU Usage
1GB Bandwidth	1TB Bandwidth	1TB Bandwidth	1TB Bandwidth	1TB Bandwidth
100 MB Storage	1 GB Storage	1 GB Storage (up to 100GB)	1 GB Storage (up to 100GB)	1 GB Storage (up to 100GB)
Custom Domains	Custom Domains	Custom Domains	Custom Domains	Custom Domains
20min Session Timeout	20min Session Timeout	24hr Session Timeout	24hr Session Timeout	1hr Session Timeout
SSL Support	SSL Support	SSL Support	SSL Support	SSL Support
64BIT Processes	64BIT Processes	64BIT Processes	64BIT Processes	64BIT Processes
Always On Feature	Always On Feature	Always On Feature	Always On Feature	Always On Feature
\$0/ mo	\$5/ mo	\$25/ mo	\$50/ mo	\$100/ mo

Fuente: (GearHost, 2020).

Análisis detallado de las telecomunicaciones

A continuación, en el cuadro No. 9 se detallan los elementos relacionados con las telecomunicaciones que requiere el sistema para su funcionamiento en una futura puesta en producción.

Cuadro 9. Análisis de telecomunicaciones

Tipo de Red	Internet	Equipo	Costos
Para el sistema de control y gestión del recurso humano, es necesaria una red tipo LAN para lograr la conectividad entre las máquinas, tanto al servidor aplicativo web como al servidor de base de datos	Para el uso del prototipo en un ambiente de producción, se requiere de internet para el uso del aplicativo una vez publicado en la red. Se recomienda una conexión estable de no menos de 5mb de velocidad.	Cableado para la conexión. Router.	Tienda Karol ya cuenta con un servicio de internet de 30mb de velocidad, por lo que no representaría algún costo adicional este tema.

Fuente: (Elaboración propia).

Basado en el cuadro anterior, el sistema desarrollado ante una eventual puesta en producción requiere de un servicio de internet para su correcto funcionamiento, además, la publicación en la red del aplicativo, les permitiría a los administradores de Tienda Karol acceder a la herramienta desde cualquier lugar y dispositivo con acceso a internet.

Descripción detallada de la base de datos.

En este apartado y con apoyo del cuadro No. 10, se detalla la información de la base de datos utilizada para el prototipo funcional desarrollado, costos, capacidad y licenciamiento.

Cuadro 10. Descripción de la base de datos

Motor de Base de Datos	Licenciamiento	Costos	Capacidad
SQL Server 2017	Para el desarrollo de este prototipo funcional fue utilizada la versión Express de SQL Server, es una versión gratuita llena de características de SQL Server, ideal para el desarrollo de prototipos.	La versión SQL Server Express no tiene ningún costo.	Admite 1 procesador físico, 1gb de memoria y 10 gb de almacenamiento máximo de base de datos

Fuente: (Elaboración propia).

Para el desarrollo del prototipo funcional para el control y gestión del recurso humano en Tienda Karol, se seleccionó una base de datos SQL en su ambiente Express el cual, tal como se observa en el cuadro No. 10, es gratuito y cuenta con las características suficientes para el desarrollo del prototipo.

Descripción detallada del personal requerido

A continuación, en el cuadro No. 11 se detallan los conocimientos que se requieren por parte de los usuarios para el manejo y gestión de la herramienta desarrollada, la cantidad de personas y el tipo de capacitaciones que se podrían requerir de ser necesario.

Cuadro 11. Descripción del personal requerido

Cantidad de personas	Conocimientos técnicos	Tipo de capacitación
La cantidad de personas es proporcional a la cantidad del personal en la empresa y puede ser utilizado por la cantidad de usuarios que se requiera (6 actualmente).	Se requiere conocimiento en manejo básico de computadoras y páginas web.	Computación básica (únicamente de ser necesario).

Fuente: (Elaboración propia).

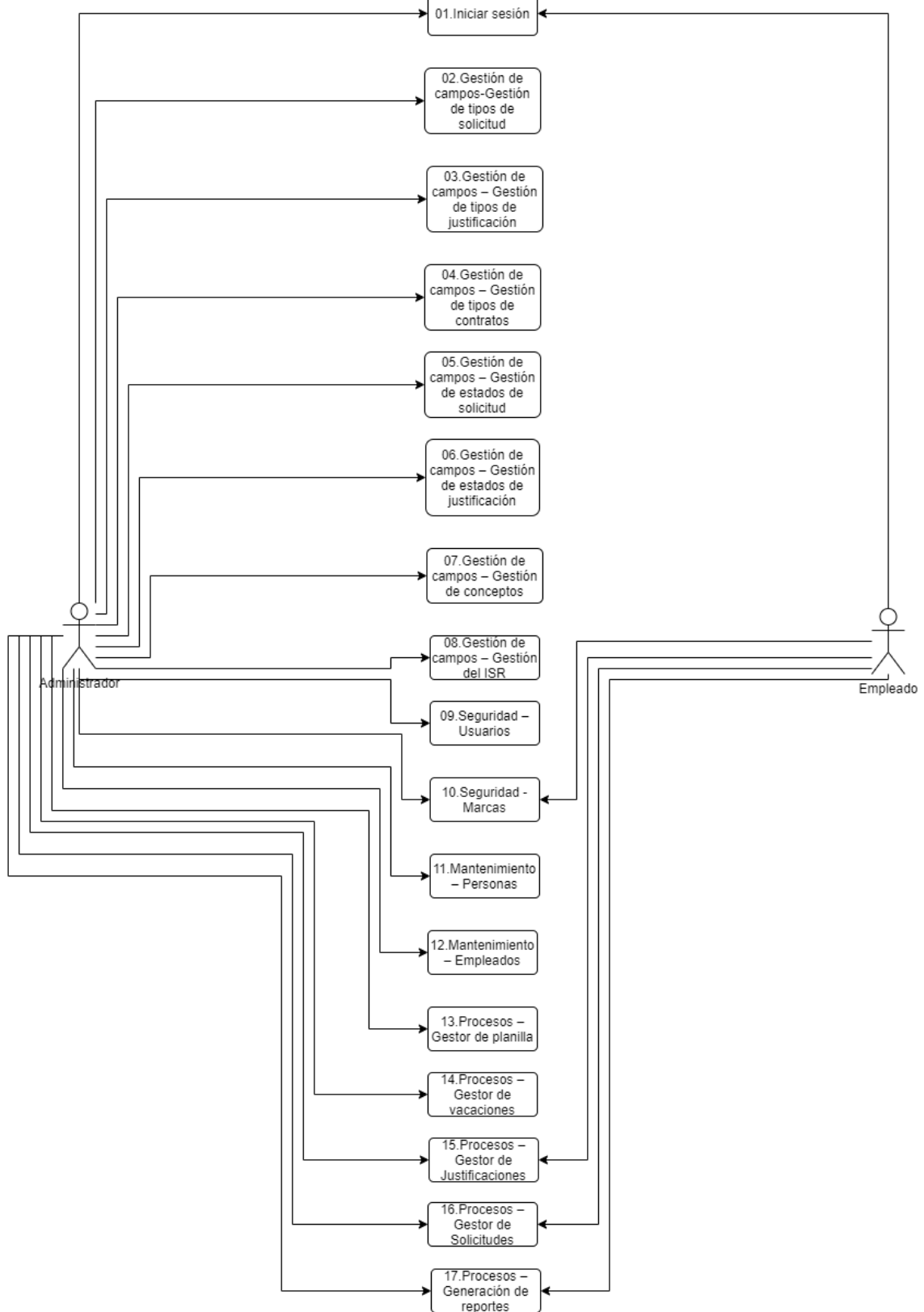
Para el uso del prototipo funcional para el control y la gestión del recurso humano, se requiere de un manejo básico de computadoras y páginas web, la cantidad de personal que se requiere para su uso es proporcional a la cantidad del personal de Tienda Karol; es decir, el sistema puede trabajar con la cantidad de personas que se requieran. En caso de requerirse, se debe aplicar una capacitación de computación básica a los empleados.

Casos de Uso

En este apartado se mostrarán los casos de uso relacionados al prototipo funcional para el control y gestión del recurso humano en Tienda Karol, dónde se indican los actores del sistema y sus interacciones con las distintas funcionalidades.

En la imagen No. 4, se muestra el diagrama de casos de uso. Tal como se puede observar, interactúan 2 actores, administrador y empleado, cada uno con interacciones diferentes con las funcionalidades del sistema según su rol.

Imagen 4. Diagrama de Casos de Uso



Fuente: (Elaboración propia).

Según el diagrama expuesto en la imagen No. 4, a continuación, se desarrollan los casos de uso del prototipo funcional para la gestión y control del recurso humano en Tienda Karol.

Cuadro 12. Caso de Uso 01

Número Caso de Uso: 01	Nombre del Caso de Uso: Iniciar Sesión
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso indica las acciones a seguir por los actores involucrados para el inicio de sesión en la herramienta.
Autor caso de uso:	Daniel Salas Acuña.
Actores relacionados:	Administrador, empleado.
Precondiciones:	El actor debe estar registrado en el sistema con su usuario y contraseña respectiva.
1.1 Flujo Básico del caso de uso	
1.1.1 Actor inicia la herramienta	
1.1.2 Actor digita su nombre de usuario	
1.1.3 Actor ingresa su contraseña	
1.1.4 Actor presiona el botón “ingresar” (FA1)	
2.1 Flujos Alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores	
FA1 – Credenciales incorrectas	2.1.1 Si los datos ingresados por el actor son incorrectos o no existen. La herramienta mostrará un mensaje de error indicando que las credenciales ingresadas son incorrectas.
Postcondiciones	
Usuario ingresa a la herramienta.	

Fuente: (Elaboración propia).

Cuadro 13. Caso de uso 02

Número Caso de Uso: 02	Nombre del Caso de Uso: Gestión de campos – Gestión de tipos de solicitud
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, modificación y eliminación de tipos de solicitudes que utilizará la herramienta.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones:	El actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Gestión de campos > Gestión Tipos Solicitud.	
1.1.3 El sistema muestra la lista de tipos de solicitud actuales y las opciones nuevo, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta.	
1.1.5 Si el autor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta.	
1.1.6 Si el autor selecciona la opción "eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF3 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nuevo Tipo de Solicitud	2.1.1 El actor ingresa el nombre del nuevo tipo de solicitud 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1).
SF2 – Editar Tipo de Solicitud	2.1.1 El autor ingresa un nuevo nombre para el tipo de solicitud. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).
SF3 – Eliminar Tipo de Solicitud	2.1.1 El sistema elimina el registro seleccionado por el autor
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración Propia)

Cuadro 14. *Caso de uso 03*

Número Caso de Uso: 03	Nombre del Caso de Uso: Gestión de campos – Gestión de tipos de justificación
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, modificación y eliminación de tipos de justificación que utilizará la herramienta.
Autor caso de uso:	Daniel Salas Acuña.
Actores relacionados:	Administrador
Precondiciones:	El actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Gestión de campos > Gestión Tipos Justificación.	
1.1.3 El sistema muestra la lista de tipos de justificación actuales y las opciones nuevo, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta.	
1.1.5 Si el autor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta.	
1.1.6 Si el autor selecciona la opción "eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF3 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nuevo Tipo de Justificación	2.1.1 El actor ingresa el nombre del nuevo tipo de justificación. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1).
SF2 – Editar Tipo de Justificación	2.1.1 El autor ingresa un nuevo nombre para el tipo de justificación. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).
SF3 – Eliminar Tipo de Justificación	2.1.1 El sistema elimina el registro seleccionado por el autor.
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración propia).

Cuadro 15. *Caso de Uso 04*

Número Caso de Uso: 04	Nombre del Caso de Uso: Gestión de campos – Gestión de tipos de contratos
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, modificación y eliminación de tipos de contratos que utilizará la herramienta.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones:	El actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Gestión de campos > Gestión Tipos Contratos.	
1.1.3 El sistema muestra la lista de tipos de contratos actuales y las opciones nuevo, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta.	
1.1.5 Si el autor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta.	
1.1.6 Si el autor selecciona la opción "eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF3 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nuevo Tipo de Contrato	2.1.1 El actor ingresa el nombre del nuevo tipo de contrato 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1).
SF2 – Editar Tipo de Contrato	2.1.1 El autor ingresa un nuevo nombre para el tipo de contratos. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).
SF3 – Eliminar Tipo de Contrato	2.1.1 El sistema elimina el registro seleccionado por el autor
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración propia).

Cuadro 16. *Caso de uso 05*

Número Caso de Uso: 05	Nombre del Caso de Uso: Gestión de campos – Gestión de estados de solicitud
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, modificación y eliminación de estados de solicitudes que utilizará la herramienta.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones:	El actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Gestión de campos > Gestión Estados Solicitud.	
1.1.3 El sistema muestra la lista de estados de solicitud actuales y las opciones nuevo, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta.	
1.1.5 Si el autor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta	
1.1.6 Si el autor selecciona la opción "eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF3 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nuevo Estado Solicitud	2.1.1 El actor ingresa el nombre del nuevo estado de solicitud y una breve descripción acerca del estado de solicitud. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1).
SF2 – Editar Estado Solicitud	2.1.1 El autor ingresa un nuevo nombre para el estado de solicitud. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).
SF3 – Eliminar Estado Solicitud	2.1.1 El sistema elimina el registro seleccionado por el autor
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración propia).

Cuadro 17. Caso de uso 06

Número Caso de Uso: 06	Nombre del Caso de Uso: Gestión de campos – Gestión de estados de justificación
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, modificación y eliminación de estados de justificaciones que utilizará la herramienta.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones:	El actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Gestión de campos > Gestión Estados Justificación.	
1.1.3 El sistema muestra la lista de estados de justificación actuales y las opciones nuevo, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta.	
1.1.5 Si el autor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta.	
1.1.6 Si el autor selecciona la opción "eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF3 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nuevo Estado Justificación	2.1.1 El actor ingresa el nombre del nuevo estado de justificación y una breve descripción acerca del estado de justificación. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1).
SF2 – Editar Estado Justificación	2.1.1 El autor ingresa un nuevo nombre para el estado de justificación. 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).
SF3 – Eliminar Estado Justificación	2.1.1 El sistema elimina el registro seleccionado por el autor
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.
Post Condiciones	

Se aplican los cambios en el sistema

Fuente: (Elaboración propia).

Cuadro 18. *Caso de uso 07*

Número Caso de Uso: 07	Nombre del Caso de Uso: Gestión de campos – Gestión de conceptos
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, modificación y eliminación de conceptos que utilizará la herramienta como porcentajes de rebaja o carga a los contratos que se deseen.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones :	El actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Gestión de campos > Gestión de Conceptos	
1.1.3 El sistema muestra la lista de conceptos actuales y las opciones nuevo, editar o eliminar	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta	
1.1.5 Si el autor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta	
1.1.6 Si el autor selecciona la opción "eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF3 se ejecuta	
2.1 Sub Flujos	
SF1 – Nuevo Concepto	2.1.1 El actor ingresa la información solicitada por el sistema: porcentaje del concepto y una breve descripción del concepto 2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1)
SF2 – Editar Concepto	2.1.1 El sistema muestra el formulario con la información actual y el usuario debe editar la información que desee: porcentaje del concepto y la descripción breve del concepto
SF3 – Eliminar Concepto	2.1.1 El sistema elimina el registro seleccionado por el autor.
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.

Post Condiciones
Se aplican los cambios en el sistema

Fuente: (Elaboración propia).

Cuadro 19. *Caso de Uso 08*

Número Caso de Uso: 08	Nombre del Caso de Uso: Gestión de campos – Gestión del ISR
Fecha elaboración:	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de edición de la tabla para calcular impuesto sobre la renta que utilizará la herramienta. Los montos para el cálculo del impuesto sobre la renta cambian anualmente según el ministerio de hacienda de Costa Rica, por tanto, es importante contar con la edición de estos campos.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones:	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Gestión de campos > Gestión del ISR.	
1.1.3 El sistema muestra la tabla con los montos para el cálculo del ISR y un botón “Editar” en cada registro.	
1.1.4 Si el autor selecciona la opción “Editar”, el sistema muestra el formulario con la información actual del registro seleccionado y el usuario debe editar la información: Salario inicial, salario final, porcentaje, periodo.	
1.1.5 El usuario presiona el botón “Guardar” y la información se actualiza	
Flujos Alternos	
FA1 – Validar información	2.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración propia).

Cuadro 20. Caso de uso 09

Número Caso de Uso: 09	Nombre del Caso de Uso: Seguridad – Usuarios
Fecha elaboración:	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, edición y eliminación de los usuarios del sistema.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones:	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Seguridad > Configuración de Usuarios.	
1.1.3 El sistema muestra la información de los usuarios actuales y las opciones nuevo, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta.	
1.1.5 El actor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta.	
1.1.6 El actor selecciona la opción "Eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF3 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nuevo Usuario	<p>2.1.1 El actor ingresa la información solicitada por el sistema: Selecciona un empleado registrado. Selecciona un tipo de usuario. Ingresa un nombre de usuario. Ingresa una contraseña. Selecciona el estado del usuario (Activo, Inactivo).</p> <p>2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).</p>
SF2 – Editar Usuario	2.1.1 El sistema muestra el formulario con la información actual y el usuario debe editar la información que desee: Tipo de Usuario, Nombre de usuario, Contraseña, estado del usuario (Activo, Inactivo)
SF3 – Eliminar Usuario	2.1.1 El sistema elimina el registro seleccionado por el autor
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración propia).

Cuadro 21. *Caso de uso 10*

Número Caso de Uso: 10	Nombre del Caso de Uso: Seguridad - Marcas
Fecha elaboración:	4/5/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación de marcas de entrada en el sistema
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones:	El actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Seguridad > Marcas >	
1.1.3 El sistema muestra el tipo de marca de entrada y la opción marcar entrada, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Marcar Entrada" y el sistema almacena la fecha y hora de la entrada en el sistema. El sistema validará la Información ingresada (FA1).	
Post Condiciones	
Se almacena marca de entrada	

Fuente: (Elaboración propia).

Cuadro 22. *Caso de uso 11*

Número Caso de Uso: 11	Nombre del Caso de Uso: Mantenimiento – Personas
Fecha elaboración	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, edición y eliminación de personas o talentos dentro del sistema.
Autor Caso de Uso:	Daniel Salas Acuña
Actores relaciona-dos:	Administrador
Precondicione s:	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Personas.	

1.1.3 El sistema muestra la información de las personas actuales y las opciones nuevo, ver currículum, actualizar currículum, administrar correos, administrar teléfonos, editar o eliminar.	
1.1.4 Si el autor selecciona la opción "Nuevo" el Sub Flujo SF1 se ejecuta.	
1.1.5 El actor selecciona la opción "Editar" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta.	
1.1.6 El actor selecciona la opción "Ver currículum" en la fila correspondiente al registro al que desea ver el currículum y el Sub Flujo SF3 se ejecuta.	
1.1.7 El actor selecciona la opción "Actualizar currículum" en la fila correspondiente al registro al que desea actualizar y el Sub Flujo SF4 se ejecuta.	
1.1.8 El actor selecciona la opción "Administrar Correos" en la fila correspondiente al registro al que le desea administrar sus correos relacionados y el Sub Flujo SF5 se ejecuta.	
1.1.9 El actor selecciona la opción "Administrar Teléfonos" en la fila correspondiente al registro al que desea administrar sus teléfonos relacionados y el Sub Flujo SF6 se ejecuta.	
1.1.10 El actor selecciona la opción "Eliminar" en la fila correspondiente al registro que se desea eliminar y el Sub Flujo SF7 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nueva Persona	<p>2.1.1 El actor ingresa la información solicitada por el sistema: Cédula de la persona. Nombre de la persona. Apellido 1 de la persona. Apellido 2 de la persona, Fecha de nacimiento de la persona, Correo de la persona, Selecciona un archivo Word, Pdf o Excel correspondiente al currículum vitae de la persona</p> <p>2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1)</p>
SF2 – Editar Usuario	<p>2.1.1 El sistema muestra el formulario con la información actual y el usuario debe editar la información que desee: Cédula, Nombre, Apellido 1, Apellido 2, Fecha de Nacimiento.</p> <p>2.1.2 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1)</p>
SF3 – Ver Currículum	2.1.1 El sistema descarga el archivo correspondiente al registro seleccionado.
SF4 – Actualizar Currículum	<p>2.1.1 El sistema muestra un campo para que el actor seleccione un nuevo archivo de currículum para el registro seleccionado.</p> <p>2.1.2 El actor selecciona el botón "Guardar". El sistema validará la</p>

	Información ingresada (FA1)
SF5 Administrar Correos	– 2.1.1 El sistema muestra un submódulo para que el actor gestione los correos relacionados al registro seleccionado.
SF6 Administrar teléfonos	– 2.1.1 El sistema muestra un submódulo para que el actor gestione los teléfonos relacionados al registro seleccionado.
SF7 Eliminar Usuario	– 2.1.1 El sistema elimina el registro seleccionado por el autor.
Flujos Alternos	
FA – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará.
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración propia).

Cuadro 23 caso de uso 12

Número Caso de Uso: 12	Nombre del Caso de Uso: Mantenimiento – Empleados
Fecha elaboración	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de creación, edición de información relacionada a los empleados
Autor Caso de Uso:	Daniel Salas Acuña
Actores relaciona-dos:	Administrador
Precondicione s:	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Mantenimiento > Empleados.	
1.1.3 El sistema muestra la información de las personas actuales y las opciones cambiar foto, gestionar conceptos y editar.	
1.1.4 Si el autor selecciona la opción "Cambiar Foto" el Sub Flujo SF1 se ejecuta.	
1.1.5 El actor selecciona la opción "Gestionar Conceptos" en la fila correspondiente al registro que se desea editar y el Sub Flujo SF2 se ejecuta.	

1.1.6 El actor selecciona la opción “Editar” en la fila correspondiente al registro al que desea ver el currículum y el Sub Flujo SF3 se ejecuta.	
2.1 Sub Flujos	
SF1 – Cambiar Foto	<p>2.1.1 El sistema muestra la fotografía actual del empleado</p> <p>2.1.2 El sistema ingresa la nueva fotografía.</p> <p>2.1.3 El actor selecciona el botón "Guardar". El sistema validará la Información ingresada (FA1)</p>
SF2 – Gestionar Conceptos	<p>2.1.1 El sistema muestra el formulario con la información actual de conceptos asignados y las opciones “Nueva acción” y “Eliminar”, esta última se muestra en cada registro.</p> <p>2.1.2 Si el usuario selección la opción eliminar, se eliminará el registro seleccionado y se quitará la relación del concepto.</p> <p>2.1.3 Si el actor selecciona la opción “Nueva Asignación” el sistema mostrará el formulario para crear una nueva asignación de un concepto previamente registrado en el sistema.</p> <p>2.1.4 El actor ingresa la información solicitada: -Concepto -Descripción del concepto</p> <p>2.1.5 El actor selecciona la opción “Guardar”. El sistema validará la información ingresada (FA1)</p>
SF3 – Editar	<p>2.1.1 El sistema muestra la información actual del empleado y la opción “Editar Contrato”.</p> <p>2.1.2 El actor selecciona la opción “Editar Contrato” y el sistema muestra la información actual del contrato del empleado.</p> <p>2.1.3 El actor ingresa la información que desea actualizar: - Salario base -Inicio de contrato -Fin del contrato -Tipo de contrato -Horario -Cargo</p> <p>2.1.4 El actor selecciona la opción “Guardar”. El sistema validará la información ingresada (FA1)</p>
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración Propia)

Cuadro 24. Caso de uso 13

Número Caso de Uso: 13	Nombre del Caso de Uso: Procesos – Gestor de planilla
Fecha elaboración:	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de cálculo de la planilla mensual.
Autor caso de uso:	Daniel Salas Acuña
Actores relacionados:	Administrador
Precondiciones :	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Procesos > Gestor de planillas.	
1.1.3 El sistema muestra el historial de planilla y las opciones “Calcular Planilla”, “Ver detalle de deducciones por ISR” en cada registro, “Ver detalle de deducciones por conceptos” en cada registro.	
1.1.4 Si el autor selecciona la opción "Calcular Planilla" el Sub Flujo SF1 se ejecuta.	
1.1.5 El actor selecciona la opción “Ver detalle de deducciones por ISR” en la fila correspondiente al registro al cual desea consultar la deducción aplicada del ISR y el Sub Flujo SF2 se ejecuta.	
1.1.6 El actor selecciona la opción “Ver detalle de deducciones por conceptos” en la fila correspondiente al registro al cual desea consultar la deducción aplicada del ISR y el Sub Flujo SF2 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nuevo cálculo de planilla	2.1.1 El actor ingresa la información solicitada por el sistema: Selecciona el mes a calcular, Selecciona el año, el sistema mostrará el año actual. 2.1.2 El actor selecciona el botón "Calcular". El sistema validará la información ingresada (FA1).
SF2 – Detalle de deducciones por ISR	2.1.1 El sistema mostrará el detalle de deducciones por ISR que se le está aplicando al registro seleccionado.
SF3 – detalle de deducciones por conceptos	2.1.1 El sistema mostrará el detalle de deducciones por conceptos que se le está aplicando al registro seleccionado
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará
Post Condiciones	
Se aplican los cambios en el sistema	

Fuente: (Elaboración propia).

Cuadro 25. Caso de uso 14

Número Caso de Uso: 14	Nombre del Caso de Uso: Procesos – Gestor de Vacaciones
Fecha elaboración:	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de cálculo de vacaciones disponibles por empleado y la asignación de vacaciones.
Autor caso de uso:	Daniel Salas Acuña.
Actores relacionados:	Administrador
Precondiciones :	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona Procesos > Gestor de vacaciones.	
1.1.3 El sistema muestra el historial de vacaciones y las opciones “Calcular vacaciones”, “Asignar vacación” en cada registro, “Ver Histórico”.	
1.1.4 Si el autor selecciona la opción "Calcular Vacaciones" el Sub Flujo SF1 se ejecuta.	
1.1.5 El actor selecciona la opción “Asignar Vacación” en la fila correspondiente al registro al cual desea realizar la nueva asignación y el Sub Flujo SF2 se ejecuta.	
1.1.6 El actor selecciona la opción “Ver histórico” y el Sub Flujo SF2 se ejecuta.	
2.1 Sub Flujos	
SF1 – Calcular vacaciones	2.1.1 El sistema calcula los días disponibles para vacaciones por empleado.
SF2 – Asignar vacación	2.1.1 El sistema muestra un formulario para la asignación de vacaciones, el usuario debe ingresar una fecha de inicio y una fecha final.
SF3 – Ver histórico	2.1.1 El sistema mostrará el histórico vacacional del empleado seleccionado.
Flujos Alternos	
FA1 – Validar información	3.1.1 El sistema valida la información ingresada por el autor y en caso de haber un error, el sistema lo notificará
Post Condiciones	

Fuente: (Elaboración Propia)

Cuadro 26. *Casos de uso 15*

Número Caso de Uso: 15	Nombre del Caso de Uso: Procesos – Gestor de Justificaciones
Fecha elaboración:	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de cálculo de creación de justificaciones.
Autor caso de uso:	Daniel Salas Acuña.
Actores relacionados:	Administrador, Empleado.
Precondiciones:	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona justificaciones > Gestor de justificaciones.	
1.1.3 El sistema muestra el historial de justificaciones y la opción “Nueva”.	
1.1.4 Si el autor selecciona la opción "Nueva" el Sub Flujo SF1 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nueva Justificación	<p>2.1.1 El actor ingresa la información solicitada por el sistema:</p> <ul style="list-style-type: none"> Tipo de justificación Empleado Fecha de Justificación Descripción de la Justificación <p>2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).</p>
Post Condiciones	
Se aplican los cambios en el sistema y se envía la información detallada de la nueva justificación a todos los usuarios administradores y al empleado relacionado con la justificación, vía correo electrónico.	

Fuente: (Elaboración Propia)

Cuadro 27. Caso de Uso 16

Número Caso de Uso: 16	Nombre del Caso de Uso: Procesos – Gestor de Solicitudes
Fecha elaboración:	6/6/2020
Descripción Caso de Uso:	Este caso de uso describe el proceso de cálculo de creación de solicitudes.
Autor caso de uso:	Daniel Salas Acuña.
Actores relacionados:	Administrador, Empleado.
Precondiciones:	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.
1.1 Flujo Básico	
1.1.1 Actor inicia sesión en la herramienta.	
1.1.2 Actor se dirige al menú desplegable y selecciona solicitudes > Gestor de solicitudes.	
1.1.3 El sistema muestra el historial de solicitudes y la opción “Nueva”	
1.1.4 Si el autor selecciona la opción "Nueva" el Sub Flujo SF1 se ejecuta.	
2.1 Sub Flujos	
SF1 – Nueva Solicitud	<p>2.1.1 El actor ingresa la información solicitada por el sistema:</p> <ul style="list-style-type: none"> Tipo de solicitud Empleado Fecha de solicitud Descripción de la solicitud <p>2.1.2 El actor selecciona el botón "Guardar". El sistema validará la información ingresada (FA1).</p>
Post Condiciones	
Se aplican los cambios en el sistema y se envía la información detallada de la nueva solicitud a todos los usuarios administradores y al empleado relacionado con la solicitud, vía correo electrónico.	

Fuente: (Elaboración Propia)

Cuadro 28. Caso de uso 17

Número de Caso de Uso: 17	Caso de Uso: 17	Nombre del Caso de Uso: Procesos – Generación de reportes
Fecha elaboración:	6/6/2020	
Descripción Caso de Uso:	Este caso de uso especifica las acciones a seguir por los actores para poder obtener información de importancia, a partir de la generación de los reportes disponibles en la herramienta.	
Autor caso de uso:	Daniel Salas Acuña	
Actores relacionados:	Administrador	
Precondiciones:	El Actor debe estar registrado en el sistema como administrador y debe haber iniciado sesión.	
1.1 Flujo Básico		
1.1.1 Actor inicia sesión en la herramienta.		
1.1.2 Actor se dirige al menú desplegable y selecciona Reportes.		
1.1.3 El sistema muestra las opciones:		
<ul style="list-style-type: none"> -General de Pagos -Intervalo de edades -Pagos por empleado -Llegadas tardías por empleado -Ausencias por empleado -Justificaciones por empleado -Solicitudes por empleado 		
1.1.4 Si el autor selecciona la opción "General de Pagos" el Sub Flujo SF1 se ejecuta.		
1.1.5 Si el autor selecciona la opción "Intervalo de edades" el Sub Flujo SF2 se ejecuta.		
1.1.6 Si el autor selecciona la opción "Pagos por empleado" el Sub Flujo SF3 se ejecuta.		
1.1.7 Si el autor selecciona la opción "Llegadas tardías por empleado" el Sub Flujo SF4 se ejecuta.		
1.1.8 Si el autor selecciona la opción "Ausencias por empleado" el Sub Flujo SF5 se ejecuta.		
1.1.9 Si el autor selecciona la opción "Justificaciones por empleado" el Sub Flujo SF6 se ejecuta.		
1.1.10 Si el autor selecciona la opción "Solicitudes por empleado" el Sub Flujo SF7 se ejecuta.		
2.1 Sub Flujos		
SF1 – Reporte General de Pagos	2.1.1. Sistema despliega la información general de pagos por planilla.	
SF2 – Reporte intervalo de edades	2.1.1. Sistema despliega reporte tipo “DashBoard” con la información de las edades de los empleados registrados en la herramienta.	
SF3 – Reporte pagos por empleado	2.1.1. Actor ingresa el nombre del empleado. 2.1.2. Sistema despliega la información de pagos según el nombre del empleado ingresado por el usuario (FA1).	

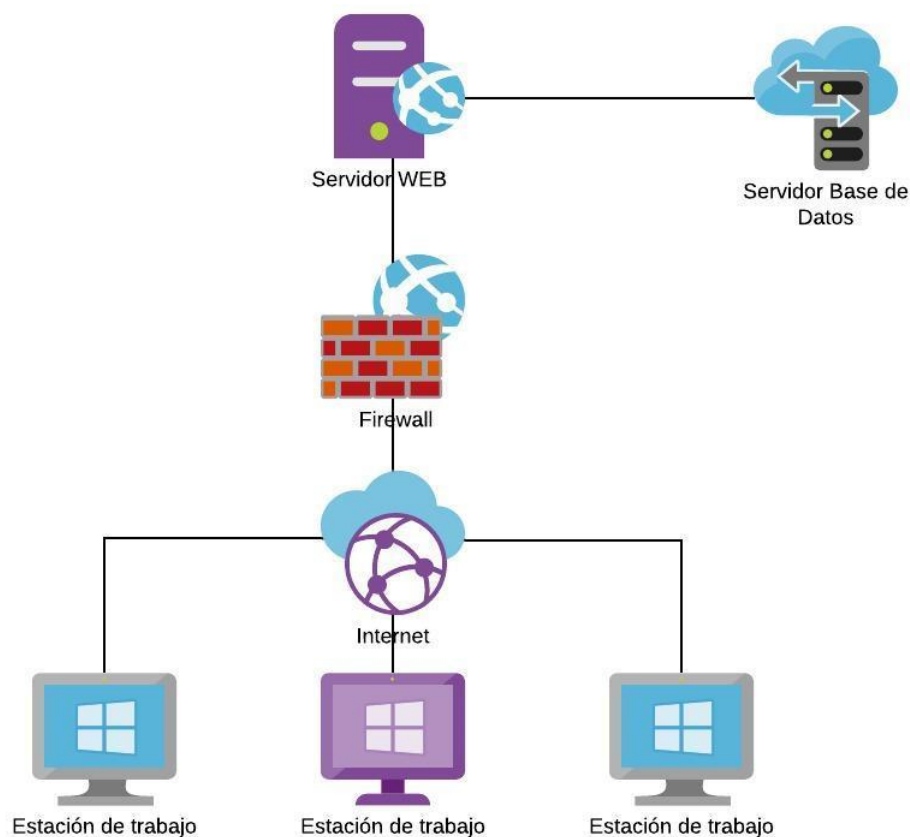
SF4 – Reporte llegadas tardías por empleado	2.1.1. Actor ingresa el nombre del empleado. 2.1.2. Sistema despliega la información de llegadas tardías según el nombre del empleado ingresado por el usuario (FA1).
SF5 – Reporte de ausencias por empleado	2.1.1. Actor ingresa el nombre del empleado. 2.1.2. Sistema despliega la información de ausencias según el nombre del empleado ingresado por el usuario (FA1).
SF6 – reporte de justificaciones por empleado	2.1.1. Actor ingresa el nombre del empleado. 2.1.2. Sistema despliega la información de justificaciones según el nombre del empleado ingresado por el usuario (FA1).
SF7 – reporte de solicitudes por empleado	2.1.1. Actor ingresa el nombre del empleado. 2.1.2. Sistema despliega la información de solicitudes según el nombre del empleado ingresado por el usuario (FA1).
Post Condiciones	
Usuario consulta la información deseada, exitosamente.	

Fuente: (Elaboración Propia)

Diseño

Arquitectura del sistema

A continuación, en la imagen No.5, se muestra el diseño de la arquitectura del sistema ideal para una posible implementación a futuro del prototipo funcional del recurso humano en tienda Karol.

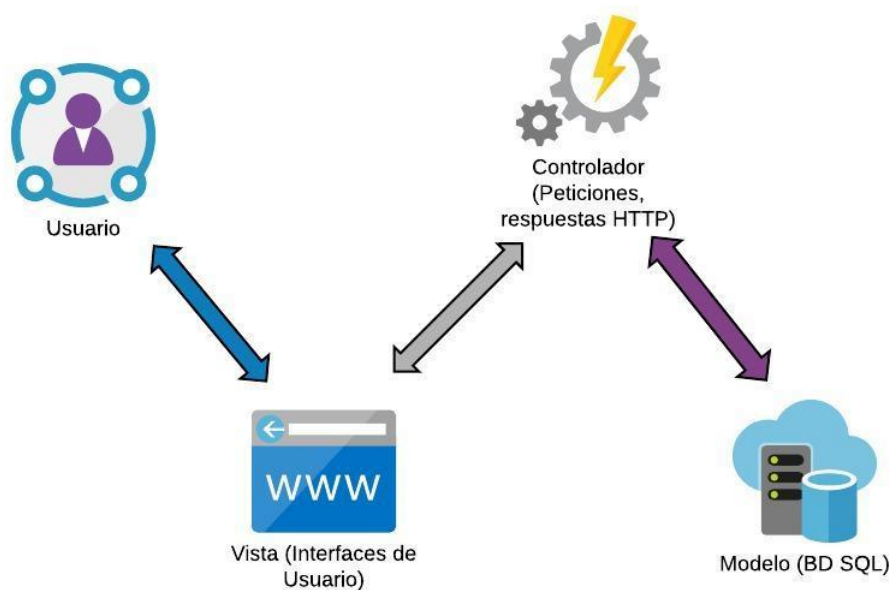
Imagen 5. *Arquitectura del sistema*

Fuente: (Elaboración propia).

La imagen anterior, ejemplifica la interacción entre los componentes del sistema, las estaciones de trabajo se comunican vía red a un servidor web el cual, almacena la consola, dicha consola muestra las vistas necesarias para las peticiones del usuario, cuando la consola detecta una nueva solicitud inmediatamente envía una petición de información al servidor de base de datos, el cual le responde con la información solicitada y la consola web se encarga de mostrarla en pantalla para ser visualizada por los usuarios.

Arquitectura del software

Gracias a la arquitectura de software se puede visualizar de manera gráfica el diseño de un producto de software. El prototipo funcional para la gestión del recurso humano en Tienda Karol, utiliza una arquitectura modelo vista controlador o MVC la cual se puede visualizar en la imagen No 6.

Imagen 6. *Arquitectura del software*

Fuente: (Elaboración propia).

Diseño de interfaces.

A continuación, se muestran las diferentes vistas o interfaces con las que cuenta el sistema para el control y gestión del recurso humano en Tienda Karol. Para el diseño de las vistas se optó por utilizar los colores representativos de la empresa por su logo.

En la imagen No. 7, se puede observar la pantalla de login o de ingreso del sistema, esta pantalla es la primera que verán los usuarios al ingresar a la plataforma y en ella se solicitan las credenciales de acceso por medio de campos de texto. El sistema validará los datos ingresados y brindará el acceso al usuario, según el rol que tenga asignado o le negará el ingreso en caso de haber ingresado información incorrecta.

Imagen 7. Vista Login

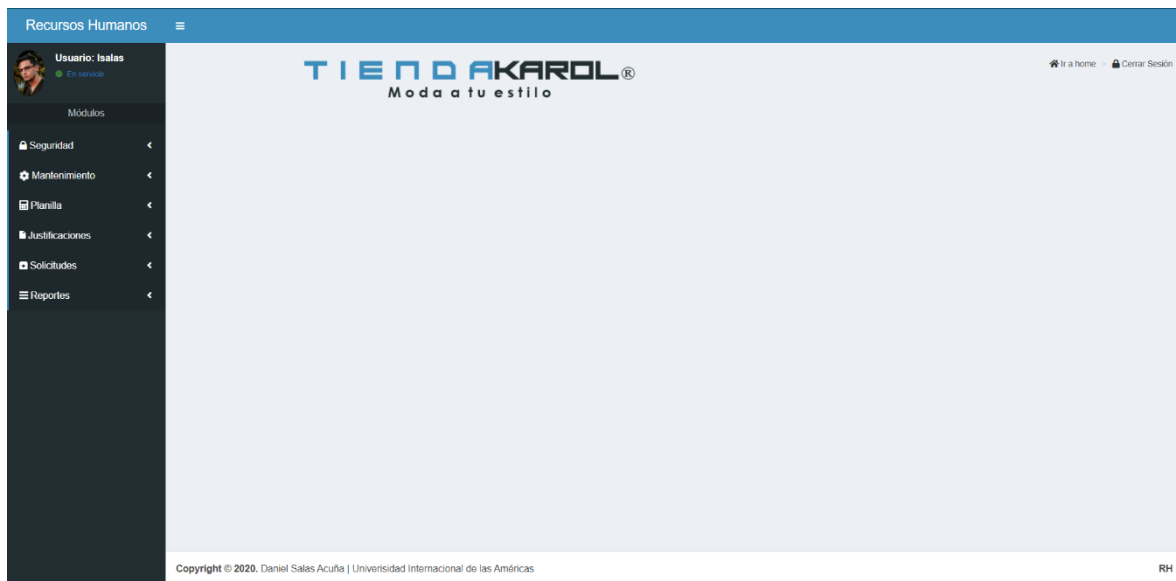


Fuente: (Elaboración propia).

En las imágenes No. 8 y No. 9 se pueden apreciar cada una de las pantallas principales de la herramienta, al ser la primera la que visualizan los usuarios con rol de administrador y la segunda la pantalla que visualizan los usuarios con rol de empleado. La diferencia entre cada una de ellas es únicamente que los usuarios con rol de administrador pueden visualizar y acceder a todas las funcionalidades de la herramienta, mientras que los usuarios con rol de empleado se limitan únicamente a ciertas funcionalidades.

En ambos casos se puede observar en el costado izquierdo un panel desplegable que muestra cada una de las opciones a las cuales el usuario tiene acceso, junto con su fotografía de perfil y su nombre de usuario. Estas opciones, corresponden a cada módulo y al presionarlas con un click izquierdo, se despliega un subpanel con cada uno de los submódulos disponibles, tal como se puede visualizar en la imagen No. 10.

Imagen 8. Vista Pantalla Principal Administradores



Fuente: (Elaboración propia).

Imagen 9. Vista Pantalla Principal Empleados



Fuente: (Elaboración propia).

Imagen 10. Ejemplo Vista Pantalla Principal Opciones Desplegadas



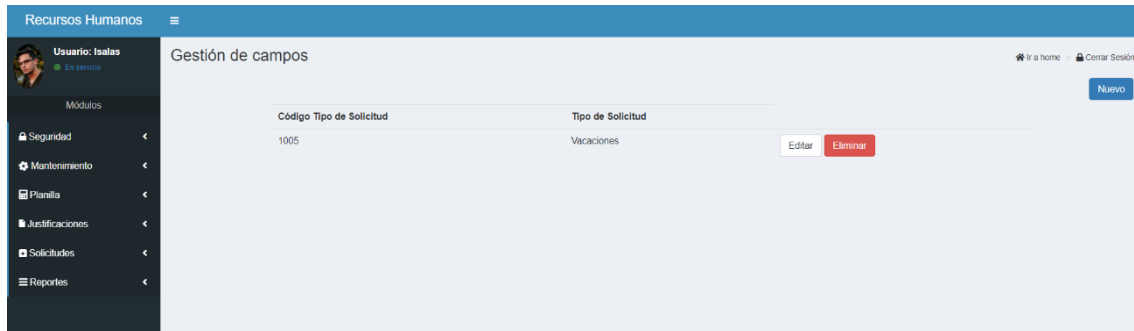
Fuente: (Elaboración propia).

Al ingresar al módulo de mantenimiento, el usuario tendrá la opción de ingresar al submódulo de gestión de campos y seleccionar cualquiera de los mantenimientos disponibles para la gestión de los campos “catálogos”, también podrá ingresar al mantenimiento de personas y empleados. En cada mantenimiento del submódulo de gestión de campos se muestran las opciones nuevo, editar o eliminar; las dos últimas el sistema las muestra por cada registro, de tal forma que el usuario pueda elegir directamente el registro que desea editar o eliminar, con excepción del mantenimiento gestión del ISR el cual, solo cuenta con la opción de editar mostrándose por cada registro.

Lo mencionado anteriormente cambia en el mantenimiento de personas, en el cual se muestran las opciones nuevo, ver currículum, actualizar currículum, editar y eliminar; las cuatro últimas de igual manera se muestran por cada registro.

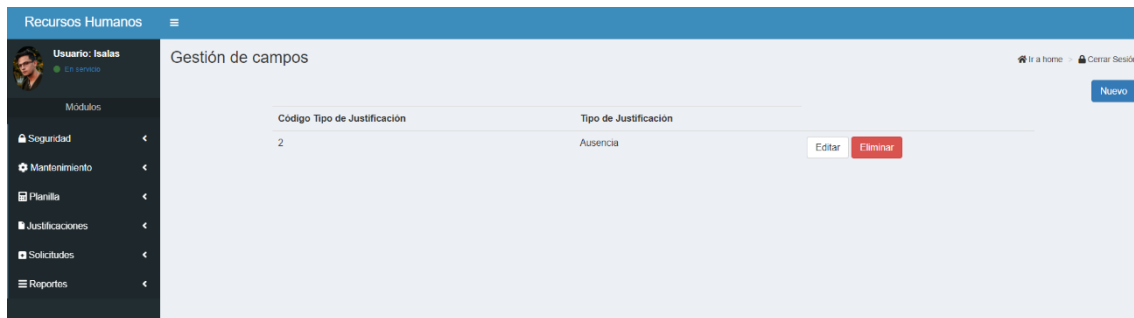
Lo mismo sucede en el mantenimiento de empleados donde las opciones pasan a ser: nuevo, cambiar foto, gestionar conceptos, editar y eliminar; de igual forma las últimas cuatro opciones mostrándose por cada registro.

Imagen 11. Gestión Tipos de Solicitud



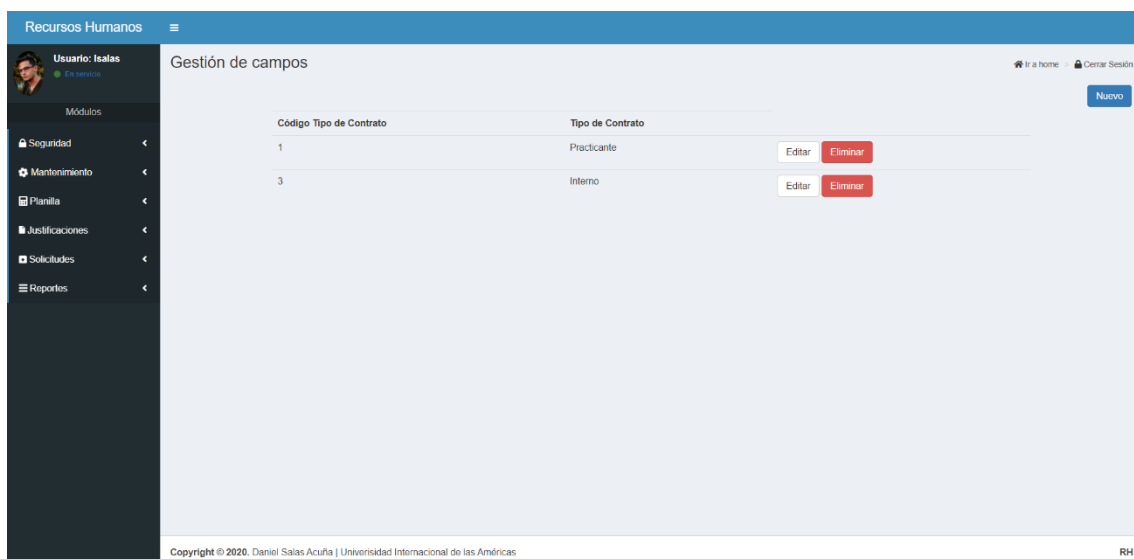
Fuente: (Elaboración propia).

Imagen 12. Gestión Tipos de Justificación



Fuente: (Elaboración propia).

Imagen 13. Gestión Tipos Contratos



Fuente: (Elaboración propia).

Imagen 14. Gestión Estados Solicitud

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestión de campos

Ir a home | Cerrar Sesión

Nuevo

Código Estado de Solicitud	Nombre Estado	Descripción		
1	Aceptada	Este estado indica que la solicitud del empleado fue aceptada	Editar	Eliminar
3	Denegada	Este estado indica que la solicitud del empleado es denegada	Editar	Eliminar

Fuente: (Elaboración propia).

Imagen 15. Gestión Estados Justificación

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestión de campos

Ir a home | Cerrar Sesión

Nuevo

Código Estado de Justificación	Nombre Estado	Descripción		
1	Aceptada	Este estado indica que la justificación del empleado es aceptada	Editar	Eliminar
1002	Denegada	Este estado indica que la justificación del empleado es denegada	Editar	Eliminar

Fuente: (Elaboración propia).

Imagen 16. Gestión Horarios

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestión de Horarios

Ir a home | Cerrar Sesión

Nuevo

Código del Horario	Hora de entrada establecida	Hora de salida establecida	Descripción del horario		
2	08:00:00	16:00:00	Horario diurno	Editar	Eliminar

Fuente: (Elaboración propia).

Imagen 17. Gestión Conceptos

Código del Concepto	Porcentaje	Descripción	Editar	Eliminar
1	14.00	Seguro de salud - Enfermedad y maternidad	Editar	Eliminar
11	14.90	Concepto 1	Editar	Eliminar
14	20.50	Concepto 2	Editar	Eliminar

Fuente: (Elaboración propia).

Imagen 18. Gestión ISR

Id	Salario Inicial	Salario Final	Porcentaje del impuesto	Periodo	Editar
1	840000.00	1123000.00	10.00	2020	Editar
2	1123000.00	2163000.00	15.00	2020	Editar
3	2163000.00	4325000.00	20.00	2020	Editar
4	4325000.00	9999999.00	25.00	2020	Editar

Fuente: (Elaboración propia).

En los mantenimientos gestión de tipos de solicitud, gestión de tipos de justificación y gestión de tipos de contratos al dar clic en la opción nuevo o en la opción editar se visualiza únicamente un campo de texto para ingresar el nombre del tipo de solicitud, justificación o contrato, tal y como se puede observar en la imagen No. 19, dicho formato es igual para los tres mantenimientos. En el caso de la opción editar, el sistema mostrará la información actual en el campo de texto y el usuario deberá cambiarla si desea, tal como se muestra en la imagen No. 20. Dicho formato de edición también es igual para estos tres mantenimientos.

Imagen 19. *Formato nuevo para gestión de tipos de solicitudes, justificaciones o contratos*

Fuente: (Elaboración propia).

Imagen 20. *Formato Editar para gestión de tipos de solicitudes, justificaciones o contratos*

Fuente: (Elaboración propia).

En los mantenimientos gestión de estados de solicitud y gestión estados de justificación, al dar clic en la opción nuevo o en la opción editar se visualizan únicamente dos campos de texto: uno para ingresar el nombre del estado y el segundo para añadir una descripción del estado, tal y como se puede observar en la imagen No. 21, dicho formato es igual para ambos mantenimientos. En el caso de la opción editar el sistema mostrará la información actual en los campos de texto y el usuario deberá cambiarla si desea, tal como se muestra en la imagen No. 22; este formato de edición también es igual para estos dos mantenimientos.

Imagen 21. *Formato Nuevo para gestión de estados de solicitudes o justificaciones*

Fuente: (Elaboración propia).

Imagen 22. *Formato Editar para gestión de estados de solicitudes o justificaciones.*

Fuente: (Elaboración propia).

Para el mantenimiento de gestión de horarios la pantalla de agregar un nuevo tipo de horario se muestra en la imagen No. 23; mientras que la pantalla de edición de un horario previamente registrado se puede observar en la imagen No.24. Al agregar o editar un horario en este mantenimiento el sistema muestra tres campos, en los cuales el usuario debe registrar una hora de entrada, una hora de salida y una descripción. Este último campo el que describe el horario.

Imagen 23. *Nuevo Horario*

Fuente: (Elaboración propia).

Imagen 24. *Editar Horario*

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestión de Campos

Ir a home | Cerrar Sesión

Hora de Entrada: 08:00

Hora de Salida: 16:00

Descripción: Horario diurno

Guardar

Fuente: (Elaboración propia).

En el mantenimiento de gestión del ISR (impuesto sobre la renta), el usuario únicamente puede editar los registros, lo anterior se debe a que el impuesto sobre la renta en nuestro país se maneja únicamente bajo cuatro rubros variables por año. Por esta razón que no existe una opción para agregar un nuevo rubro; sin embargo, si es posible editar cada uno de los rubros según se requiera, dado que estos cambian por periodo.

A continuación, en la imagen No. 25 se muestra la pantalla de edición del mantenimiento de gestión del ISR.

Imagen 25. *Editar ISR*

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestión del Impuesto Sobre la Renta

Ir a home | Cerrar Sesión

Salario Inicial: 840000.00

Salario Final: 1123000.00

Porcentaje: 10.00

Periodo: 2020

Guardar

Fuente: (Elaboración propia).

Imagen 26. *Personas*

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Registro De Personas

Ir a home | Cerrar Sesión

Cédula	Nombre	Apellido1	Apellido2	Fecha Nacimiento	Dirección	
14578945	Freddy	Román	Vega	5-5-1988	San José, Puriscal, Santiago	Costado oeste parque puriscal Ver Curriculum Actualizar Curriculum Administrar Correos Administrar Teléfonos Editar Eliminar

Nuevo

Fuente: (Elaboración propia).

En el caso del mantenimiento de personas, la pantalla de agregar una nueva se muestra en la imagen No. 27; mientras que la pantalla de edición de información de una persona previamente registrada en el sistema se puede observar en la imagen No.28.

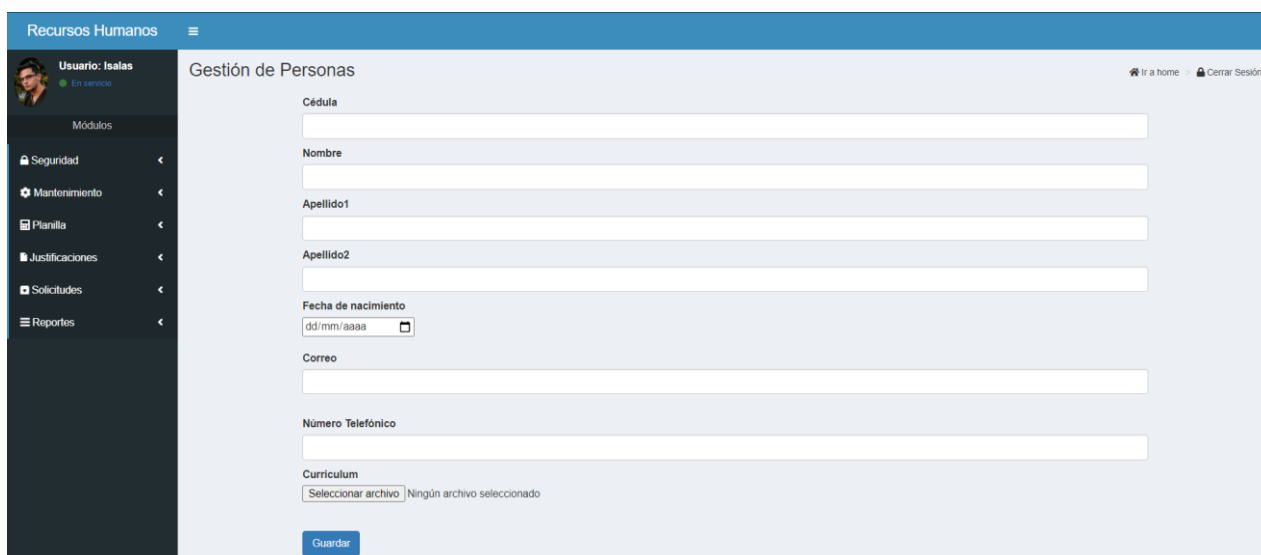
Al dar clic en la opción “Ver Currículum”, el sistema automáticamente abrirá el archivo del currículum con la aplicación predeterminada en la configuración del sistema operativo según su extensión.

Al dar clic en la opción “Actualizar Currículum”, el sistema mostrará una pantalla con un campo para actualizar el currículum de la persona seleccionada, tal y como se muestra en la imagen No.29.

Al dar clic en la opción “Administrar Correos”, el sistema mostrará un submódulo para gestionar los correos de la persona seleccionada, tal y como se muestra en la imagen No.30.

Al dar clic en la opción “Administrar Teléfonos”, el sistema mostrará un submódulo para gestionar los números telefónicos de la persona seleccionada, tal y como se muestra en la imagen No.31.

Imagen 27. Nueva Persona



The screenshot displays the 'Recursos Humanos' (Human Resources) system interface. The main header is 'Recursos Humanos' with a menu icon. The user profile shows 'Usuario: Isaias' and 'En servicio'. The left sidebar lists modules: Seguridad, Mantenimiento, Planilla, Justificaciones, Solicitudes, and Reportes. The main content area is titled 'Gestión de Personas' and contains a form for adding a new person. The form fields are: Cédula, Nombre, Apellido1, Apellido2, Fecha de nacimiento (with a date picker showing 'dd/mm/aaaa'), Correo, and Número Telefónico. Below these fields is a 'Currículum' section with a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'. A 'Guardar' (Save) button is located at the bottom of the form. In the top right corner of the main area, there are links for 'Ir a home' and 'Cerrar Sesión'.

Fuente: (Elaboración propia).

Imagen 28. *Editar Persona*

Recursos Humanos

Usuario: Isalas

Gestión de Personas

Cédula: 117140789

Nombre: Daniel

Apellido1: Salas

Apellido2: Acuna

Correo: lsalascioud7781@gmail.com

Número Telefónico: 86420141

Fecha de nacimiento: 18/07/1998

Guardar

Fuente: (Elaboración propia).

Imagen 29. *Actualizar Currículum*

Recursos Humanos

Usuario: Isalas

Actualización de Currículum

Cédula de la Persona: 174894562

Nombre de Documento Actual: Conceptos para marco referencial.docx

Currículum: Josue Quesada Solis

Currículum: Seleccionar archivo | Ningún archivo seleccionado

Guardar

Fuente: (Elaboración propia).

Imagen 30. *Administrar Correos*

Recursos Humanos

Usuario: Isalas

Gestión de Correos

Nuevo Correo

Cédula Persona	Nombre	Correo	Activo
117140789	Daniel Salas Acuna	luisda7781@hotmail.com	True

Editar Eliminar

Fuente: (Elaboración propia)

Imagen 31. *Administrar Teléfonos*

Recursos Humanos

Usuario: Isalas

Gestión de Teléfonos

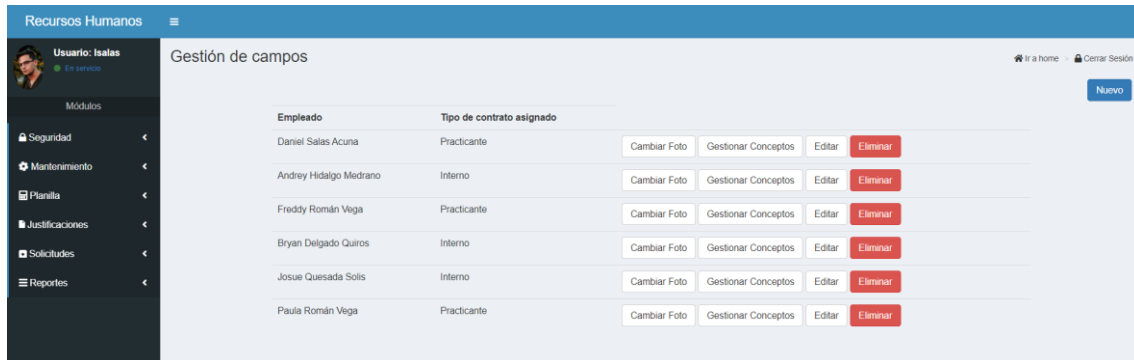
Nuevo Número Telefónico

Cédula Persona	Nombre	Telefono	Activo
117140789	Daniel Salas Acuna	86420141	

Editar Eliminar

Fuente: (Elaboración propia)

Imagen 32. Empleados



Fuente: (Elaboración propia).

En el caso del mantenimiento de empleados la pantalla de agregar un nuevo empleado se muestra en la imagen No. 33, al dar clic en la opción “Editar” la herramienta mostrará la información del empleado junto la opción “Editar Contrato” (ver imagen No.34), al dar clic a dicha opción el sistema desplegará una pantalla con la información actual del contrato asignado al empleado seleccionado y el usuario deberá editarla si así lo desea.

Al dar clic en la opción “Cambiar Foto”, el sistema mostrará una pantalla con un campo para actualizar la fotografía del empleado seleccionado, tal y como se muestra en la imagen No.36.

Cuando el usuario da clic en la opción “Gestionar Conceptos” es redirigido a una pantalla para la asignación de conceptos al empleado seleccionado (ver imagen No.37), en esta pantalla se encuentran disponibles las opciones “Nueva Asignación” y “Eliminar”.

La opción “Nueva Asignación” permite asignar conceptos previamente configurados a los empleados, tal y como lo muestra la imagen No. 38, al dar clic en dicha opción el sistema solicitará que el empleado seleccione un concepto y agregue una breve descripción del concepto a asignar, por otro lado, al dar clic sobre la opción “Eliminar” el sistema borrará la asignación de concepto seleccionada.

Imagen 33. *Nuevo Empleado*

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Nuevo Empleado

Ir a home Cerrar Sesión

Persona a contratar
--Seleccione una persona--

Tipo de Contrato
--Seleccione un contrato--

Fotografía
Seleccionar archivo | Ningún archivo seleccionado

Salario base

Horario
--Seleccione un horario--

Cargo
--Seleccione un cargo--

Inicio de contrato
dd/mm/aaaa

Fin Contrato
dd/mm/aaaa

Guardar

Fuente: (Elaboración propia).

Imagen 34. *Editar Empleado*

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestión de Empleados

Ir a home Cerrar Sesión

Nombre del empleado: Daniel Salas Acuna

Cédula de la persona: 117140789

Contrato actual: Practicante [Editar Contrato](#)

Horario actual: Horario diurno

Cargo actual: Cajero

Salario actual: 450000,00

Fotografía

Fuente: Elaboración propia.

Imagen 35. *Editar Empleado / Contrato*

Recursos Humanos

Usuario: Isalas
En servicio

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestión de Contratos

Ir a home Cerrar Sesión

Salario Base
450000,00

Inicio de Contrato 01/01/2020

Fin de Contrato 31/12/2020

Tipo de Contrato
Practicante

Horario
Horario diurno

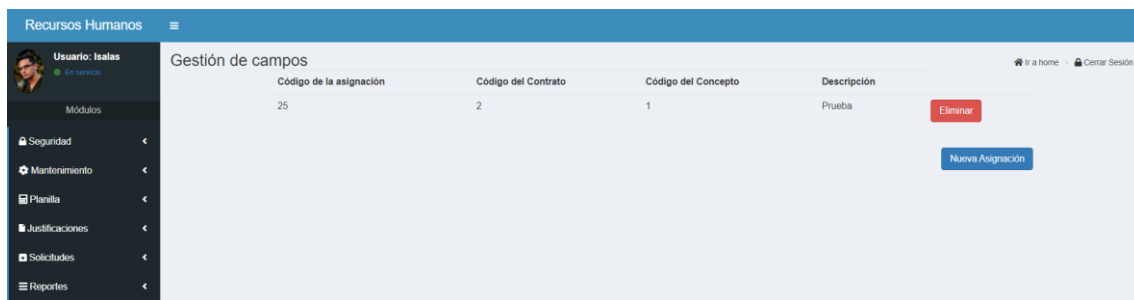
Cargo
Cajero

Guardar

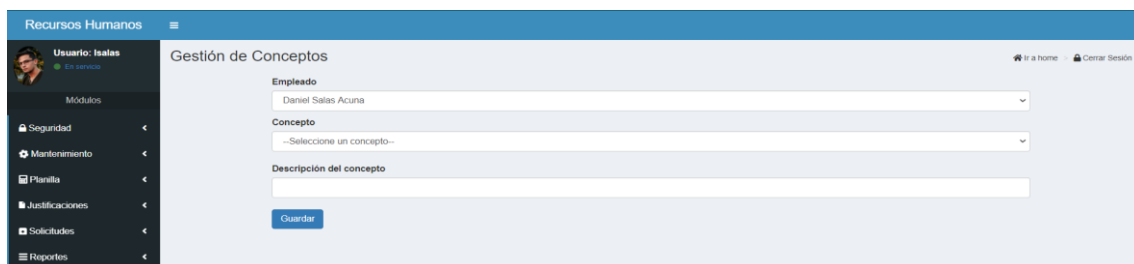
Fuente: (Elaboración propia).

Imagen 36. *Cambiar Foto*

Fuente: (Elaboración propia).

Imagen 37. *Asignación de conceptos*

Fuente: (Elaboración propia).

Imagen 38. *Nueva Asignación*

Fuente: (Elaboración propia).


Al ingresar al módulo de planilla el usuario, podrá visualizar todos los procesos relacionadas con la planilla. Cuando el usuario acceda al proceso de gestor de planilla, podrá visualizar la pantalla que se muestra en la imagen No. 39. En esta pantalla se encuentran las opciones de “Calcular Planilla”, “Ver detalle de deducciones por ISR” y “Ver detalle de deducciones por conceptos”.

Cuando el usuario seleccione la opción “Calcular Planilla” el sistema mostrará la pantalla que se visualiza en la imagen No. 40; asimismo, por medio de ella solicitará el mes y año de la planilla a calcular.

Las opciones “Ver detalle de deducciones por ISR” y “Ver detalle de deducciones por conceptos” se muestran por cada registro de planilla previamente calculados.

Cuando el usuario seleccione la opción “Ver detalle de deducciones por ISR”, el sistema mostrará la información de las deducciones por impuesto sobre la renta que se le están aplicando al registro seleccionado (ver imagen 41), mientras que si selecciona “Ver detalle de deducciones por conceptos” mostrará la información de las deducciones por conceptos que se le están aplicando al registro seleccionado (ver imagen 42).

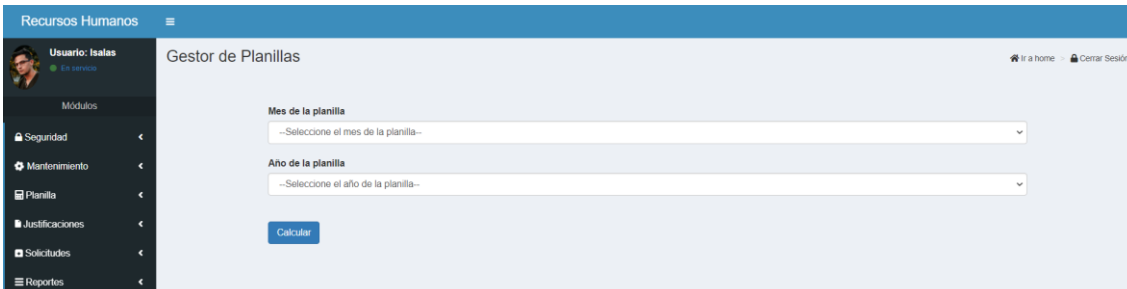
Imagen 39. *Gestor de planilla*



Código de la planilla	Código del Empleado	Nombre completo del empleado	Periodo	Sueldo Base	Sueldo Neto
2020062071	2071	Paula Román Vega	2020-06-28	2173000.00	1996700.00
20200641	41	Daniel Salas Acuna	2020-06-28	2800000.00	2096300.00
20200643	43	Andrey Hidalgo Medrano	2020-06-28	200000.00	158000.00
20200660	60	Freddy Román Vega	2020-06-28	450000.60	319500.43
20200661	61	Bryan Delgado Quiros	2020-06-28	400000.00	344000.00

Fuente: (Elaboración propia).

Imagen 40. *Calcular Planilla*



Recurso Humanos

Usuario: Isalas

Módulos

- Seguridad
- Mantenimiento
- Planilla
- Justificaciones
- Solicitudes
- Reportes

Gestor de Planillas

Mes de la planilla
--Seleccione el mes de la planilla--

Año de la planilla
--Seleccione el año de la planilla--

Calcular

Fuente: (Elaboración propia).

Imagen 41. *Deducciones por ISR*

The screenshot shows a web interface for 'Recursos Humanos'. The user is 'Isalas' and is logged in. The page title is 'Detalle de deducciones por impuesto sobre la renta'. A table displays the following data:

Código del Empleado	Código del Contrato	Porcentaje del ISR
41	2	20.00%

Fuente: (Elaboración propia).

Imagen 42. *Deducciones por conceptos*

The screenshot shows a web interface for 'Recursos Humanos'. The user is 'Isalas' and is logged in. The page title is 'Detalle de deducciones por Conceptos'. A table displays the following data:

Código del empleado	Descripción Concepto	Porcentaje Concepto
41	Prueba	14.00%

Fuente: (Elaboración propia).

Al acceder al proceso de cálculo de llegadas tardías, el sistema mostrará la pantalla que se observa en la imagen No. 43, en la cual se calculan las llegadas tardías por empleado, al tomar en cuenta las marcas de entrada registradas y la hora de entrada establecida.

Imagen 43. *Cálculo de llegadas tardías*

The screenshot shows a web interface for 'Recursos Humanos'. The user is 'Isalas' and is logged in. The page title is 'Gestión de Faltas'. A table displays the following data:

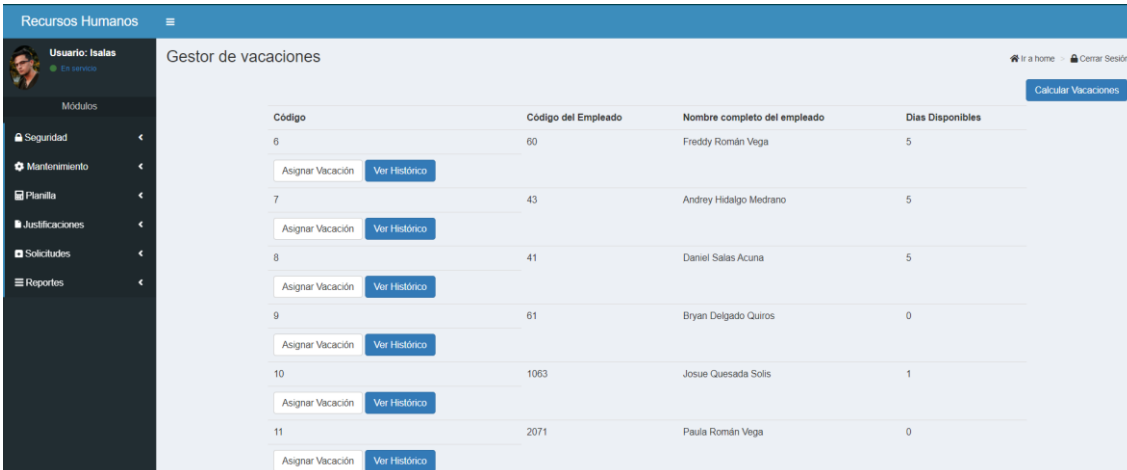
Código del Empleado	Nombre del Empleado	Fecha de la marca	Hora de entrada establecida	Hora de la Marca	Horas Tarde
41	Daniel Salas Acuna	14/5/2020 01:48:42	08:00:00	13:48:42.7033333	5
41	Daniel Salas Acuna	10/6/2020 09:29:52	08:00:00	09:29:52.7066667	1

Fuente: (Elaboración propia).

En el proceso de gestión de vacaciones se podrá observar una vista, tal y como se muestra en la imagen No.44, esta pantalla muestra todos los empleados registrados con su información general y los días disponibles a la fecha en la que se ejecute el botón “Calcular Vacaciones”, este botón se encarga de actualizar los días disponibles.

Como se puede visualizar en la imagen No.44, cada registro cuenta con dos opciones “Asignar Vacación” y “Ver Histórico”; la primera de ellas se encarga de dirigir al usuario a una pantalla en la cual puede asignar vacaciones al empleado seleccionado agregando un rango de fechas (ver imagen No.45); mientras que la opción “Ver Histórico” dirige al usuario a una pantalla en la cual podrá ver el histórico vacacional del empleado seleccionado, tal y como se puede observar en la imagen No.46.

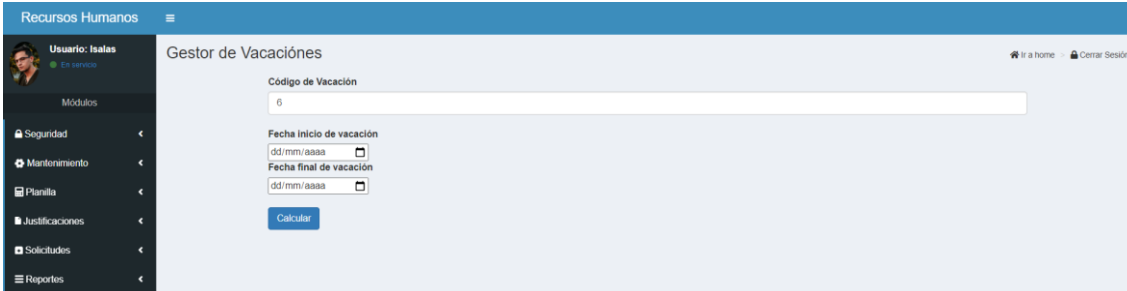
Imagen 44. *Gestor de Vacaciones*



Código	Código del Empleado	Nombre completo del empleado	Días Disponibles
6	60	Freddy Román Vega	5
7	43	Andrey Hidalgo Medrano	5
8	41	Daniel Salas Acuna	5
9	61	Bryan Delgado Quiros	0
10	1063	Josue Quesada Solis	1
11	2071	Paula Román Vega	0

Fuente: (Elaboración propia).

Imagen 45. *Asignar Vacación*



Código de Vacación: 6

Fecha inicio de vacación: dd/mm/aaaa

Fecha final de vacación: dd/mm/aaaa

Calcular

Fuente: (Elaboración propia).

Imagen 46. *Historico Vacacional*

Código de Vacación	Empleado	Fecha de Inicio	Fecha Final
6	Freddy Román Vega	24/6/2020 12:00:00	24/6/2020 12:00:00

Fuente: (Elaboración propia).

Los módulos de justificaciones y solicitudes trabajan con la misma lógica y el mismo formato de vista, en estos módulos los usuarios podrán registrar sus justificaciones o solicitudes. El sistema notificará a todos los usuarios con rol de administrador vía correo cada vez que un usuario registre una nueva justificación o solicitud en la plataforma, también notificará por correo al usuario que realizó el registro, indicándole que su justificación o solicitud se realizó correctamente.

Cada justificación o solicitud cuenta con una serie de estados, el sistema notificará tanto a todos los usuarios con rol de administrador como al usuario con rol de empleado relacionado, cada vez que la justificación o solicitud cambie de estado.

Imagen 47. *Formato Vistas Gestor de Justificaciones o Solicitudes*

Código de Solicitud	Tipo de Solicitud	Nombre de Empleado	Fecha de Solicitud	Fecha de Inicio	Fecha Final	Descripcion	Estado de la Solicitud
38	Vacaciones	Freddy Román Vega	21/6/2020 12:00:00	21/6/2020 12:00:00	30/6/2020 12:00:00	dthg	Cerrada
39	Vacaciones	Paula Román Vega	21/6/2020 12:00:00	1/6/2020 12:00:00	10/6/2020 12:00:00	Prueba SP	Registrada
40	Vacaciones	Andrey Hidalgo Medrano	21/6/2020 12:00:00	28/6/2020 12:00:00	30/6/2020 12:00:00	Prueba SP 2	Registrada

Fuente: (Elaboración propia).

Imagen 48. *Formato Vistas Estados de justificaciones solicitudes*

Código de Solicitud	Tipo de Solicitud	Nombre de Empleado	Fecha de Solicitud	Fecha de Inicio	Fecha Final	Descripción	Estado de la Solicitud
39	Vacaciones	Paula Román Vega	21/6/2020 12:00:00	1/6/2020 12:00:00	10/6/2020 12:00:00	Prueba SP	Registrada
40	Vacaciones	Andrey Hidalgo Medrano	21/6/2020 12:00:00	28/6/2020 12:00:00	30/6/2020 12:00:00	Prueba SP 2	Registrada

Fuente: (Elaboración propia).

Imagen 49. *Formato Vistas Nueva Justificación o Solicitud*

Nueva Solicitud

Tipo De Solicitud: --Seleccione el tipo de solicitud--

Fecha de Inicio: dd/mm/aaaa

Fecha Final: dd/mm/aaaa

Descripción de la solicitud

Guardar

Fuente: (Elaboración propia).

Imagen 50. *Formato Vistas Edición Cambio de Estados de Justificaciones o Solicitudes*

Gestión de solicitudes

Tipo de solicitud: Vacaciones

Nombre de Empleado: Paula Román Vega

Fecha Solicito: 21/6/2020

Fecha inicio: 1/6/2020

Fecha final: 10/6/2020

Estado de solicitud: --Seleccione un estado--

Descripción: Prueba SP

Guardar

Fuente: (Elaboración propia).

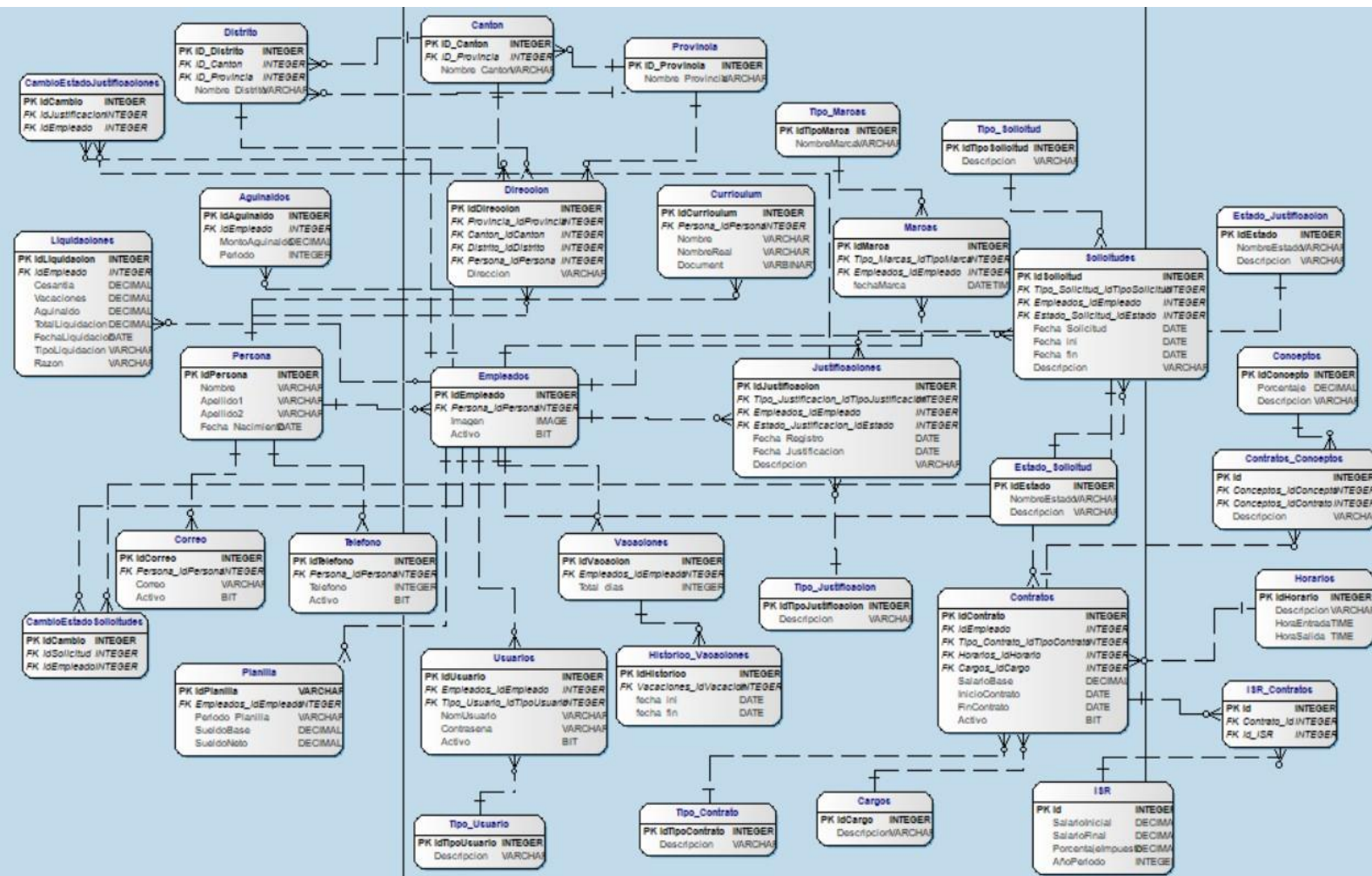
Diseño de base de datos y su diccionario de datos.

En este apartado se expone el diseño de la base de datos para el prototipo funcional que apoya la gestión del recurso humano en Tienda Karol, además del detalle de cada una de las tablas que la conforman.

Diseño de base de datos

A continuación, se expondrá en diseño de la base de datos del prototipo por medio de un diagrama de entidad relación, con sus respectivas relaciones entre las tablas, llaves primarias y atributos. Este diseño fue creado con la ayuda de las herramientas Microsoft SQL Server Management Studio Express Edition y EntityFramework de Visual Studio 2019.

Imagen 51. Diagrama Entidad Relación Base de Datos



Fuente: (Elaboración propia).

*Diccionario de Base de datos*Cuadro 29. *Diccionario de Datos – Persona*

Tabla: Persona			
Descripción: Tabla que almacena la información de cada persona			
Llave Primaria: idPersona			
Campo	Tipo de Dato	Relación	Descripción
idPersona	int	Llave Primaria	Campo que identifica la cédula de la persona.
Nombre	varchar(60)	Atributo	Nombre de la persona.
Apellido1	varchar(60)	Atributo	Primer apellido de la persona.
Apellido2	varchar(60)	Atributo	Segundo apellido de la persona.
Fecha_Nacimiento	date	Atributo	Fecha de nacimiento de la persona.

Fuente: (Elaboración propia).

Cuadro 30. *Diccionario de Datos – Empleados*

Tabla: Empleados			
Descripción: Tabla que almacena la información de cada Empleado			
Llave Primaria: idEmpleado			
Campo	Tipo de Dato	Relación	Descripción
idEmpleado	int	Llave Primaria	Campo que identifica un empleado contratado.
Persona_idPersona	int	Llave Foránea	Campo que identifica a la persona según la tabla "Persona".
Imagen	Image	Atributo	Campo que almacena una fotografía del empleado.
Activo	bit		Campo que indica si el empleado esta activo o inactivo.

Fuente: (Elaboración propia).

Cuadro 31. *Diccionario de Datos – Usuarios*

Tabla: Usuarios			
Descripción: Tabla que almacena la información de los usuarios para el ingreso al sistema			
Llave Primaria: idUsuario			
Campo	Tipo de Dato	Relación	Descripción
idUsuario	int	Llave Primaria	Campo que identifica un usuario.
NomUsuario	varchar(20)	Atributo	Nombre de usuario para el ingreso al sistema.
Contraseña	varchar(20)	Atributo	Contraseña para el ingreso al sistema
Empleados_idEmpleado	int	Llave Foránea	Campo que identifica el empleado, según la tabla Empleados.
Tipo_Usuario_idTipoUsuario	int	Llave Foránea	Campo que identifica el tipo de usuario según la tabla Tipo_Usuario.
Activo	bit	Atributo	Campo que indica si el usuario se encuentra activo.

Fuente: (Elaboración propia).

Cuadro 32. *Diccionario de Datos – Tipo_Usuario*

Tabla: Tipo_Usuario			
Descripción: Tabla que almacena los tipos de usuarios disponibles			
Llave Primaria: idTipoUsuario			
Campo	Tipo de Dato	Relación	Descripción
idTipoUsuario	int	Llave Primaria	Campo que identifica un tipo de usuario.
Descripcion	varchar(100)	Atributo	Descripción del tipo de usuario.

Fuente: (Elaboración propia).

Cuadro 33. *Diccionario de Datos – Provincia*

Tabla: Provincia			
Descripción: Tabla que almacena las provincias de Costa Rica			
Llave Primaria: ID_Provincia			
Campo	Tipo de Dato	Relación	Descripción
ID_Provincia	int	Llave Primaria	Campo que identifica la provincia.
Nombre_Provincia	varchar(10)	Atributo	Nombre de la provincia.

Fuente: (Elaboración propia).

Cuadro 34. *Diccionario de Datos – Canton*

Tabla: Canton			
Descripción: Tabla que almacena los cantones de Costa Rica			
Llave Primaria: ID_Canton			
Campo	Tipo de Dato	Relación	Descripción
ID_Canton	int	Llave Primaria	Campo que identifica el cantón.
Nombre_Canton	varchar(30)	Atributo	Nombre del cantón.
ID_Provincia	int	Llave Foránea	Campo que identifica la provincia, según la tabla Provincia.

Fuente: (Elaboración propia).

Cuadro 35. *Diccionario de Datos – Distrito*

Tabla: Distrito			
Descripción: Tabla que almacena los Distritos de Costa Rica			
Llave Primaria: ID_Distrito			
Campo	Tipo de Dato	Relación	Descripción
ID_Distrito	int	Llave Primaria	Campo que identifica el distrito.
Nombre_Distrito	varchar(30)	Atributo	Nombre del distrito.
ID_Canton	int	Llave Foránea	Campo que identifica el cantón, según la tabla Canton.
ID_Provincia	Int	Llave Foránea	Campo que identifica la provincia, según la tabla Provincia

Fuente: (Elaboración propia).

Cuadro 36. *Diccionario de Datos – Dirección*

Tabla: Direccion			
Descripción: Tabla que almacena las direcciones exactas de las personas			
Llave Primaria: idDireccion			
Campo	Tipo de Dato	Relación	Descripción
idDireccion	int	Llave Primaria	Campo que identifica la dirección.
Provincia_idProvincia	int	Llave Foránea	Campo que identifica la provincia según la tabla Provincia.
Canton_idCanton	int	Llave Foránea	Campo que identifica el cantón según la tabla Canton.
Distrito_idDistrito	int	Llave Foránea	Campo que identifica el distrito según la tabla distrito.
Persona_idPersona	int	Llave Foránea	Campo que identifica a la persona, según la tabla Persona.
Direccion	varchar(200)	Atributo	Dirección Exacta de la persona.

Fuente: (Elaboración propia).

Cuadro 37. *Diccionario de Datos – Currículum*

Tabla: Currículum			
Descripción: Tabla que almacena las hojas de vida de las personas			
Llave Primaria: idCurrículum			
Campo	Tipo de Dato	Relación	Descripción
idCurrículum	int	Llave Primaria	Campo que identifica el currículum.
Nombre	varchar(100)	Atributo	Nombre identificador del currículum.
NombreReal	varchar(100)	Atributo	Nombre real del archivo.
Persona_idPersona	int	Llave Foránea	Campo que identifica la persona, según la tabla Persona.
Document	varbinary(max)	Atributo	Campo que almacena el archivo currículum.

Fuente: (Elaboración propia).

Cuadro 38. *Diccionario de Datos – Correo*

Tabla: Correo			
Descripción: Tabla que almacena los correos de las personas			
Llave Primaria: idCorreo			
Campo	Tipo de Dato	Relación	Descripción
idCorreo	int	Llave Primaria	Campo que identifica el correo.
Persona_idPersona	int	Llave Foránea	Campo que identifica la persona, según la tabla Persona.
Correo	varchar(100)	Atributo	Dirección de correo.
Activo	Bit	Atributo	Campo que indica si el correo se encuentra en estado activo

Fuente: (Elaboración propia).

Cuadro 39. *Diccionario de Datos - Aguinaldos*

Tabla: Aguinaldos			
Descripción: Tabla que almacena los aguinaldos por empleado			
Llave Primaria: idAguinaldo			
Campo	Tipo de Dato	Relación	Descripción
idAguinaldo	int	Llave Primaria	Campo que identifica el Aguinaldo.
idEmpleado	int	Llave Foránea	Campo que identifica al empleado, según la tabla Empleado.
Monto	Decimal (10,2)	Atributo	Monto por aguinaldo
Periodo	int	Atributo	Año del periodo correspondiente al aguinaldo

Fuente: (Elaboración Propia)

Cuadro 40. *Diccionario de Datos – Contratos*

Tabla: Contratos			
Descripción: Tabla que almacena los contratos de los empleados			
Llave Primaria: idContrato			
Campo	Tipo de Dato	Relación	Descripción
idContrato	int	Llave Primaria	Campo que identifica el contrato.
idEmpleado	int	Llave Foránea	Campo que identifica al empleado según la tabla: Empleado
Salario Base	decimal(10,2)	Atributo	Salario base del contrato.
Inicio Contrato	date	Atributo	Fecha en la que inicia el contrato.
Fin Contrato	date	Atributo	Fecha en la que finaliza el contrato.
Tipo_Contrato_idTipoContrato	int	Llave Foránea	Campo que identifica el tipo de contrato según la tabla: Tipo_Contrato.
Horario_idHorario	int	Llave Foránea	Campo que identifica el horario, según la tabla: Horarios.
Cargos_idCargo	int	Llave Foránea	Campo que identifica el cargo, según la tabla: Cargos.
Activo	bit	Atributo	Campo que indica si el contrato está activo o inactivo.

Fuente: (Elaboración propia)

Cuadro 41. *Diccionario de Datos – Tipo_Contrato*

Tabla: Tipo_Contrato			
Descripción: Tabla que almacena los tipos de contratos			
Llave Primaria: idTipoContrato			
Campo	Tipo de Dato	Relación	Descripción
idTipoContrato	int	Llave Primaria	Campo que identifica el tipo de contrato.
Descripcion	varchar(100)	Atributo	Descripción del tipo de contrato.

Fuente: (Elaboración propia).

Cuadro 42. *Diccionario de Datos –Horarios*

Tabla: Horarios			
Descripción: Tabla que almacena los horarios disponibles			
Llave Primaria: idHorario			
Campo	Tipo de Dato	Relación	Descripción
idHorario	int	Llave Primaria	Campo que identifica el horario.
Descripción	varchar(100)	Atributo	Descripción del horario.
HoraEntrada	time(7)	Atributo	Campo que almacena la hora de entrada.
HoraSalida	time(7)	Atributo	Campo que almacena la hora de salida.

Fuente: (Elaboración propia).

Cuadro 43. *Diccionario de Datos – Cargos*

Tabla: Cargos			
Descripción: Tabla que almacena los cargos disponibles			
Llave Primaria: idCargo			
Campo	Tipo de Dato	Relación	Descripción
idCargos	int	Llave Primaria	Campo que identifica el cargo.
Descripción	varchar(100)	Atributo	Descripción del cargo.

Fuente: (Elaboración propia).

Cuadro 44. *Diccionario de Datos – Contratos_Conceptos*

Tabla: Contratos_Conceptos			
Descripción: Tabla que almacena la relación entre los conceptos con los contratos			
Llave Primaria: id			
Campo	Tipo de Dato	Relación	Descripción
id	int	Llave Primaria	Campo que identifica la relación entre el contrato y los conceptos.
Conceptos_idConcepto	int	Llave Foránea	Campo que identifica el concepto.
Conceptos_idContrato	int	Llave Foránea	Campo que identifica el contrato.
Descripcion	varchar(100)	Atributo	Descripción de la relación.

Fuente: (Elaboración propia).

Cuadro 45. *Diccionario de Datos – Estado_Justificación*

Tabla: Estado_Justificación			
Descripción: Tabla que almacena el estado de las justificaciones			
Llave Primaria: idEstado			
Campo	Tipo de Dato	Relación	Descripción
idEstado	int	Llave Primaria	Campo que identifica el estado de la justificación.
NombreEstado	varchar(60)	Atributo	Nombre del estado.
Descripcion	varchar(250)	Atributo	Descripción.

Fuente: (Elaboración propia).

Cuadro 46. *Diccionario de Datos – Estado_Justificación*

Tabla: Estado_Justificacion			
Descripción: Tabla que almacena el estado de las justificaciones			
Llave Primaria: idEstado			
Campo	Tipo de Dato	Relación	Descripción
idEstado	int	Llave Primaria	Campo que identifica el estado de la justificación.
NombreEstado	varchar(60)	Atributo	Nombre del estado.
Descripcion	varchar(250)	Atributo	Descripción.

Fuente: (Elaboración propia).

Cuadro 47. *Diccionario de Datos – Estado_Solicitud*

Tabla: Estado_Solicitud			
Descripción: Tabla que almacena el estado de las solicitudes			
Llave Primaria: idEstado			
Campo	Tipo de Dato	Relación	Descripción
idEstado	int	Llave Primaria	Campo que identifica el estado de la solicitud.
NombreEstado	varchar(60)	Atributo	Nombre del estado.
Descripcion	varchar(250)	Atributo	Descripción.

Fuente: (Elaboración propia).

Cuadro 48. *Diccionario de Datos – Vacaciones*

Tabla: Vacaciones			
Descripción: Tabla que almacena la información de las vacaciones por empleado			
Llave Primaria: idVacacion			
Campo	Tipo de Dato	Relación	Descripción
idVacacion	int	Llave Primaria	Campo que identifica las vacaciones.
Empleados_idEmpleado	int	Llave Foránea	Campo que identifica al empleado, según la tabla Empleado.
Total_dias	int	Atributo	Total de días disponibles.

Fuente: (Elaboración propia).

Cuadro 49. *Diccionario de Datos – Historico_Vacaciones*

Tabla: Historico_Vacaciones			
Descripción: Tabla que almacena la información histórica de las vacaciones			
Llave Primaria: idHorario			
Campo	Tipo de Dato	Relación	Descripción
idHistorico	int	Llave Primaria	Campo que identifica el historial.
Vacaciones_idVacacion	int	Llave Foránea	Campo que identifica la vacación, según la tabla Vacaciones.
fecha_ini	date	Atributo	Fecha de inicio de las vacaciones.
Fecha_fin	date	Atributo	Fecha de finalización de las vacaciones.

Fuente: (Elaboración propia).

Cuadro 50. *Diccionario de Datos – Justificaciones*

Tabla: Justificaciones			
Descripción: Tabla que almacena las justificaciones de los empleados			
Llave Primaria: idJustificación			
Campo	Tipo de Dato	Relación	Descripción
idJustificacion	int	Llave Primaria	Campo que identifica la justificación.
Tipo_Justificacion_idTipoJustificacion	int	Llave Foránea	Campo que identifica el tipo de justificación, según la tabla Tipo_Justificación.
Empleados_idEmpleado	int	Llave Foránea	Campo que identifica el empleado, según la tabla Empleados.
Fecha_Registro	date	Atributo	Fecha de registro de la justificación.
Fecha_Justificacion	date	Atributo	Fecha de la justificación.
Estado_Justificación_idEstado	int	Llave Foránea	Campo que identifica el estado de justificación, según la tabla Estado_Justificación.
Descripción	Varchar(800)	Atributo	Campo que describe la justificación.

Fuente: (Elaboración propia).

Cuadro 51. *Diccionario de Datos – Solicitudes*

Llave Primaria: idSolicitud			
Tabla: Solicitudes			
Descripción: Tabla que almacena solicitudes de los empleados			
idSolicitud	int	Primaria	Campo que identifica la solicitud.
Tipo_Solicitud_idTipoSolicitud	int	Llave Foránea	Campo que identifica el tipo de solicitud, según la tabla Tipo_solicitud.
Empleados_idEmpleado	int	Llave Foránea	Campo que identifica el empleado, según la tabla Empleados.
Fecha_Solicitud	date	Atributo	Fecha de la solicitud.
Fecha_ini	date	Atributo	Fecha de Inicio.
Fecha_fin	date	Atributo	Fecha Final.
Estado_Solicitud_idEstado	int	Llave Foránea	Campo que identifica el estado de solicitud, según la tabla Estado_solicitud.
Descripción	Varchar(800)	Atributo	Campo que describe la solicitud.

Fuente: (Elaboración propia).

Cuadro 52. *Diccionario de Datos – Diccionario de Datos - CambioEstadoJustificaciones*

Llave Primaria: idCambio			
Campo	Tipo de Dato	Relación	Descripción
idcambio	int	Llave Primaria	Campo que identifica el paso de estado de una justificación.
idJustificacion	int	Llave Foránea	Campo que identifica la justificación.
idEmpleado	int	Llave Foránea	Campo que identifica el empleado administrador que realiza el cambio de estado de una justificación, según la tabla Empleados.

Cuadro 53. *Diccionario de Datos – Diccionario de Datos – CambioEstadoSolicitudes*

Llave Primaria: idCambio			
Campo	Tipo de Dato	Relación	Descripción
idcambio	int	Llave Primaria	Campo que identifica el paso de estado de una solicitud.
idSolicitud	int	Llave Foránea	Campo que identifica la solicitud
idEmpleado	int	Llave Foránea	Campo que identifica el empleado administrador que realiza el cambio de

			estado de una solicitud, según la tabla Empleados.
--	--	--	--

Cuadro 54. *Diccionario de Datos – Marcas*

Tabla: Marcas			
Descripción: Tabla que almacena las marcas de ingreso/salida			
Llave Primaria: idMarca			
Campo	Tipo de Dato	Relación	Descripción
idMarca	int	Llave Primaria	Campo que identifica la marca.
Tipo_Marcas_idTipoMarca	int	Llave Foránea	Campo que identifica el tipo de marca, según la tabla Tipo_Marcas.
fechaMarca	datetime	Atributo	Fecha y hora de la marca.
Empleados_idEmpleado	int	Llave Foránea	Campo que identifica el empleado, según la tabla Empleados.

Fuente: (Elaboración propia).

Cuadro 55. *Diccionario de Datos – Tipo_Marcas*

Tabla: Tipo_Marcas			
Descripción: Tabla que almacena los tipos de marcas			
Llave Primaria: idTipoMarca			
Campo	Tipo de Dato	Relación	Descripción
idTipoMarca	int	Llave Primaria	Campo que identifica el tipo de marca.
NombreMarca	varchar(20)	Atributo	Nombre del tipo de marca.

Fuente: (Elaboración propia).

Cuadro 56. *Diccionario de Datos – Telefono*

Tabla: Telefono			
Descripción: Tabla que almacena los teléfonos de las personas			
Llave Primaria: idTelefono			
Campo	Tipo de Dato	Relación	Descripción
idTelefono	int	Llave Primaria	Campo que identifica el teléfono.
Persona_idPersona	int	Llave Foránea	Campo que identifica a la persona, según la tabla Persona.
Telefono	int	Atributo	Campo que almacena el número telefónico.

Fuente: (Elaboración propia)

Cuadro 57. Diccionario de Datos – Planilla

Tabla: Planilla			
Descripción: Tabla que almacena la información de las planillas			
Llave Primaria: idPlanilla			
Campo	Tipo de Dato	Relación	Descripción
idPlanilla	int	Llave Primaria	Campo que identifica la planilla.
Empleados_idEmpleado	int	Llave Foránea	Campo que identifica el empleado, según la tabla Empleados.
Fecha_ini	date	Atributo	Fecha de inicio de la planilla.
Fecha_fin	date	Atributo	Fecha de finalización de la planilla.
SueldoBase	decimal(10,2)	Atributo	Campo que almacena el sueldo base.
SueldoBruto	decimal(10,2)	Atributo	Campo que almacena el sueldo bruto.
SueldoNeto	decimal(10,2)	Atributo	Campo que almacena el sueldo neto.

Fuente: (Elaboración propia).

Cuadro 58. Diccionario de Datos – Liquidaciones

Tabla: Liquidaciones			
Descripción: Tabla que almacena la información de las liquidaciones			
Llave Primaria: idLiquidacion			
Campo	Tipo de Dato	Relación	Descripción
idLiquidacion	int	Llave Primaria	Campo que identifica la liquidación
idPersona	int	Llave Foránea	Campo que identifica a la persona.
Cesantia	decimal(10,2)	Llave Foránea	Campo el monto a pagar por cesantía
Vacaciones	decimal(10,2)	Atributo	Campo el monto a pagar por vacaciones
Aguinaldo	decimal(10,2)	Atributo	Campo el monto a pagar por aguinaldo
TotalLiquidacion	decimal(10,2)	Atributo	Campo el monto a pagar por liquidación
FechaLiquidacion	date	Atributo	Campo la fecha de liquidación
Razon	Varchar(500)	Atributo	Campo que almacena la razón de la liquidación

Fuente: (Elaboración Propia)

Cuadro 59. *Diccionario de Datos - ISR*

Tabla: ISR			
Descripción: Tabla que almacena la información de los montos o rubros con los que trabajará la herramienta para el cálculo del impuesto sobre la renta			
Llave Primaria: idPlanilla			
Campo	Tipo de Dato	Relación	Descripción
id	int	Llave Primaria	Campo que identifica el rubro ISR.
SalarioInicial	decimal(10,2)	atributo	Campo que identifica el monto de salario con el que inicia el cálculo del ISR.
SalarioFinal	decimal(10,2)	Atributo	Campo que identifica el monto de salario con el que finaliza el cálculo del ISR.
PorcentajeImpuesto	decimal(10,2)	Atributo	Porcentaje que se aplica sobre el exceso del campo SalarioInicial hasta SalarioFinal.
AñoPeriodo	int	Atributo	Campo que almacena año del periodo fiscal para el cual corresponde la información ingresada en la tabla ISR.

Fuente: (Elaboración propia).

Cuadro 60. *Diccionario de Datos – ISR_Contratos*

Tabla: ISR_Contratos			
Descripción: Tabla que almacena la información de la relación entre los parámetros ingresados en la tabla ISR con los distintos contratos asignados			
Llave Primaria: id			
Campo	Tipo de Dato	Relación	Descripción
id	int	Llave Primaria	Campo que identifica el id de relación ISR_Contratos.
Contrato_id	int	atributo	Campo que identifica el contrato relacionado.
Id_ISR	int	Atributo	Campo que identifica el id del rubro ISR que aplica para el contrato relacionado.

Fuente: (Elaboración propia).

Diseño de procesos

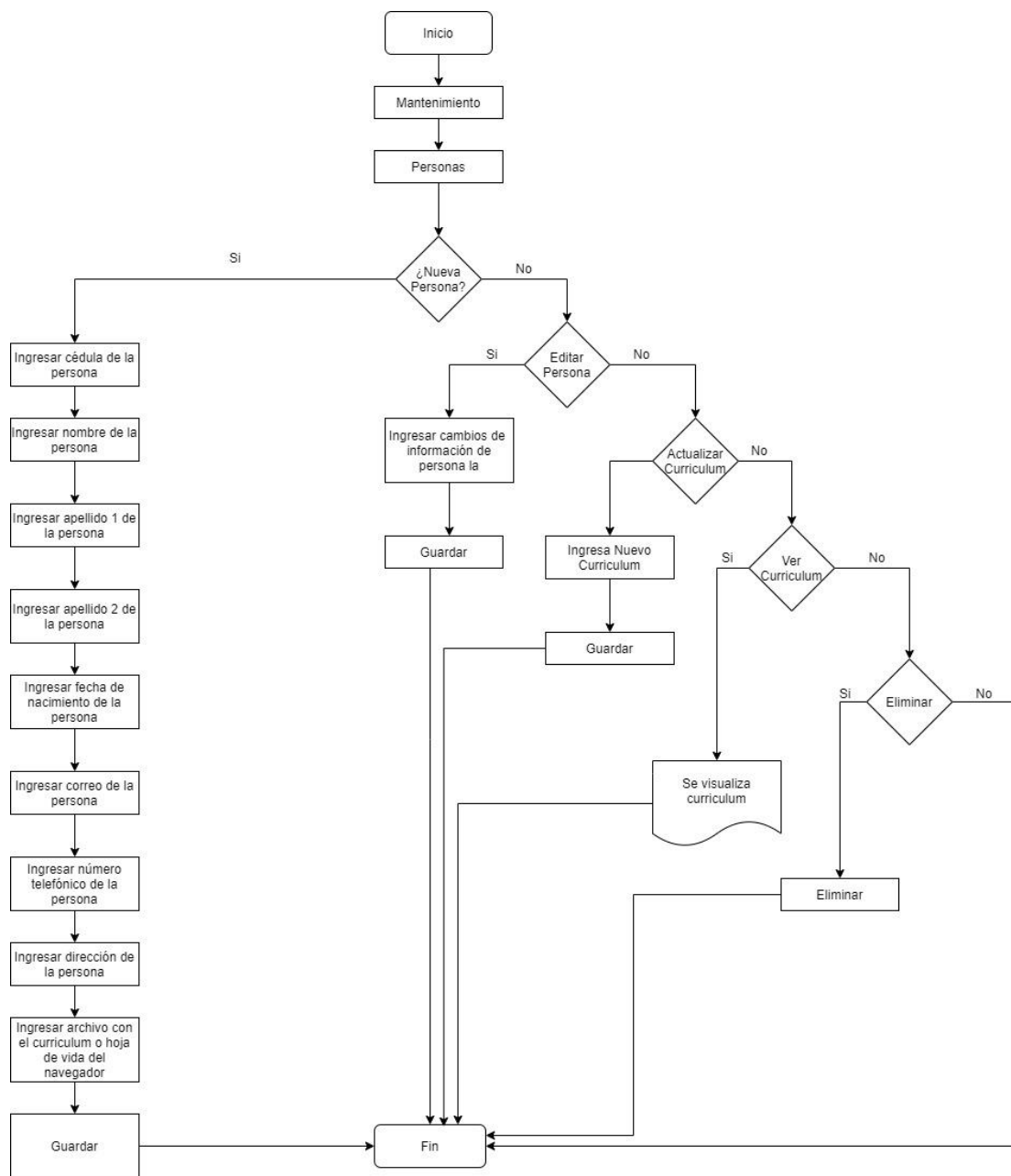
A continuación, se mostrarán cada uno de los diagramas que representan los principales procesos con los que cuenta el prototipo funcional para el control y gestión del recurso humano en Tienda Karol.

Los diagramas ejemplifican el proceso que sigue un usuario al interactuar con un módulo en específico. Los procesos que se detallarán serán los de mantenimiento de personas, mantenimiento de empleados, gestión de planilla, gestor de solicitudes, gestor de justificaciones.

Diagrama de flujo mantenimiento de personas

A continuación, en la imagen No. 50 se muestra el diagrama referente al mantenimiento de personas, cuando un usuario quiere registrar, modificar o eliminar información de una persona ya registrada.

Imagen 52. Diagrama de Flujo mantenimiento de Personas

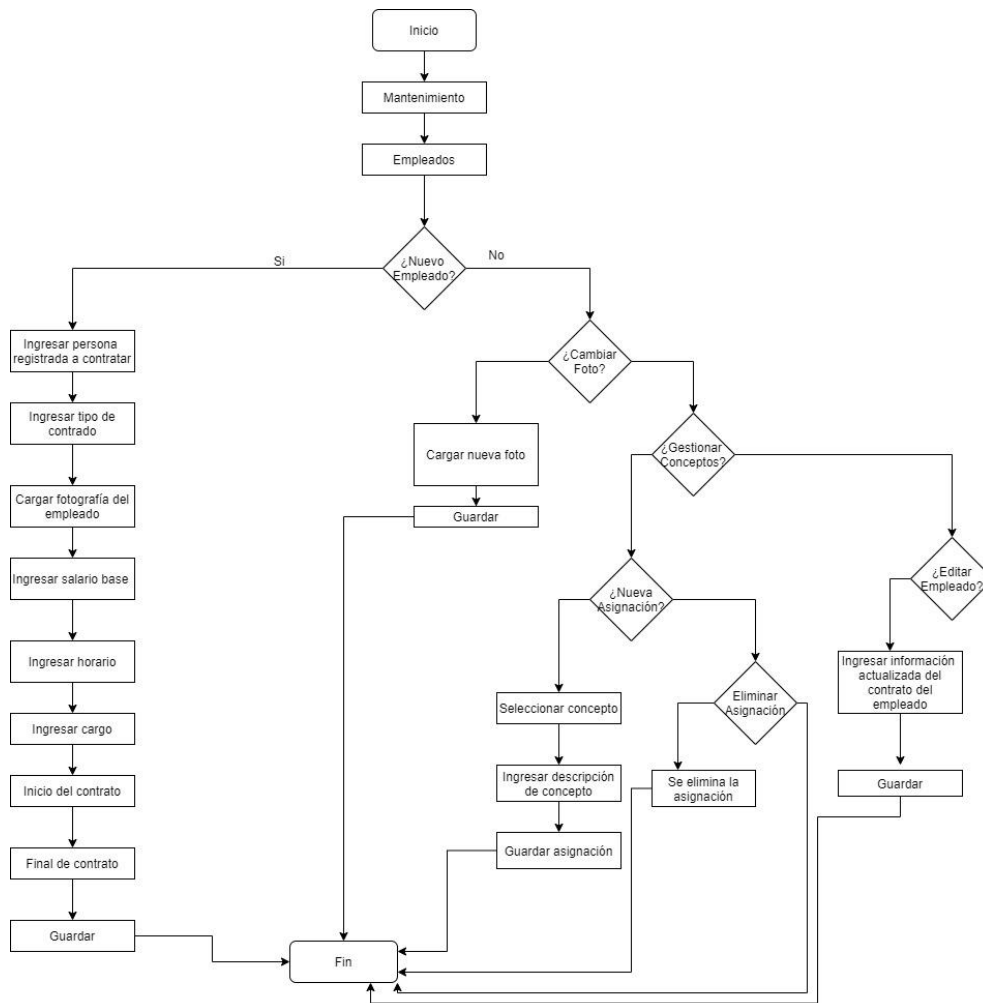


Fuente: (Elaboración propia).

Diagrama de flujo gestión de empleados

En la imagen No. 51 se muestra el diagrama el cual ejemplifica el proceso del mantenimiento de empleados, cuando un usuario desea registrar, modificar o eliminar información de un empleado ya registrado.

Imagen 53. Diagrama de flujo mantenimiento de empleados

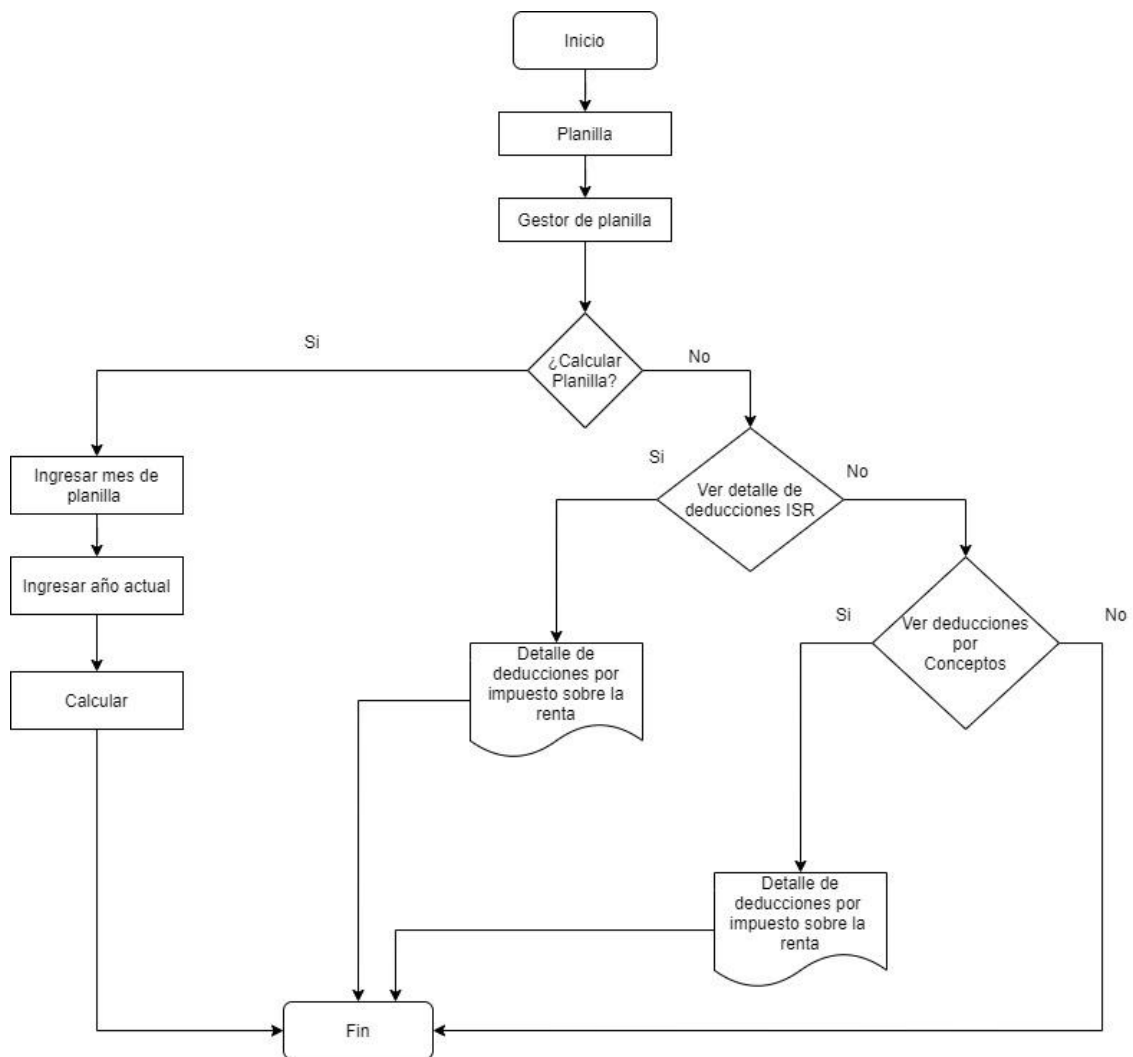


Fuente: (Elaboración propia).

Diagrama de flujo gestión de planilla

A continuación, en la imagen No. 52 se muestra el diagrama referente al proceso de gestión de planilla, cuando un usuario quiere calcular la planilla o ver el historial de planillas.

Imagen 54. Diagrama de flujo gestión de planilla

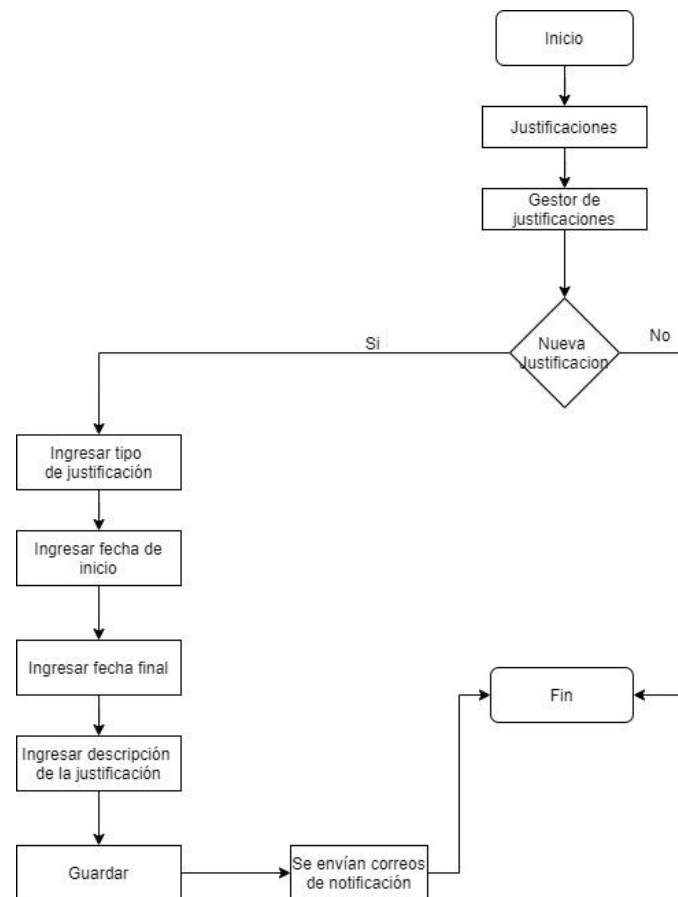


Fuente: (Elaboración propia).

Diagrama de flujo gestor de Justificaciones

En la imagen No. 53, se muestra el diagrama referente al proceso de gestión de justificaciones, cuando un usuario quiere realizar una nueva justificación.

Imagen 55. Diagrama de flujo gestor de justificaciones

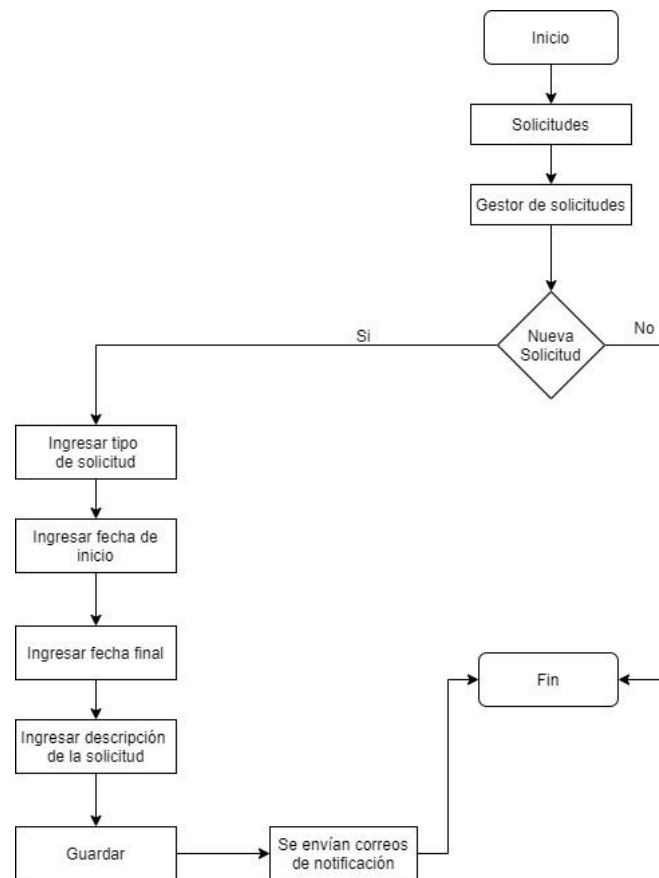


Fuente: (Elaboración propia).

Diagrama de flujo gestor de solicitudes

En la imagen No. 54, se muestra el diagrama referente al proceso de gestión de solicitudes, cuando un usuario quiere realizar una nueva solicitud. Este proceso trabaja de la misma manera que el proceso de gestión de justificaciones.

Imagen 56. Diagrama de flujo gestor de solicitudes



Fuente: (Elaboración propia).

Diseño de Salidas

En este apartado, se mostrarán el diseño de archivos, salidas del prototipo y reportes utilizados para la salida de la información cuando un usuario realiza consultas.

Diseño de archivos usados para la salida de información

Los archivos para la salida de información son generados a partir de la exportación de una consulta en la herramienta, dicha exportación genera un archivo .xls con la información de salida, el usuario deberá escoger la ubicación donde desea descargarlo.

Los archivos .xls pueden ser visualizados con la herramienta Excel del paquete de Microsoft Office.

En la imagen No. 55 se observa el archivo con los datos de las llegadas tardías por empleado.

Imagen 57. *Diseño archivo de salida reporte llegadas tardías por empleado*

	A	B	C	D	E	F
1	Código del Empleado	Nombre del Empleado	Fecha de la marca	Hora de entrada establecida	Hora de la marca	Horas tarde
2	41	Daniel Salas Acuña	24/6/2020	08:00:00	13:48:42	5
3	41	Daniel Salas Acuña	25/6/2020	09:00:00	09:29:52	1
4	41	Daniel Salas Acuña	26/6/2020	10:00:00	14:05:41	6
5						
6						

Fuente: (Elaboración propia).

En la imagen No. 56 se puede observar el diseño del archivo de salida del reporte de solicitudes por empleado.

Imagen 58. *Diseño archivo de salida reporte solicitudes por empleado*

	A	B	C	D	E	F	G
1	Código de Solicitud	Tipo de Solicitud	Nombre de Empleado	Fecha de Registro	Fecha de Solicitud	Descripción	Estado de la Solicitud
2	38	Vacaciones	Freddy Román Vega	21/06/2020 0:00	23/06/2020 0:00	Solicitud de vacaciones	Cerrada
3	41	Tarde Libre	Freddy Román Vega	26/06/2020 0:00	28/06/2020 0:00	Solicitud de vacaciones	Registrada
4							

Fuente: (Elaboración propia).

En la imagen No. 57 se visualiza el diseño del archivo de salida del reporte de justificaciones por empleado.

Imagen 59. *Diseño archivo de salida reporte justificaciones por empleado*

	A	B	C	D	E	F	G
1	Código de Justificación	Tipo de Justificación	Nombre de Empleado	Fecha de Registro	Fecha de Justificación	Descripción	Estado de la Justificación
2	2	Ausencia	Bryan Delgado Quiros	28/06/2020 0:00	27/04/2020 0:00	Se justifica ausencia del día 27 de abril del 2020	Registrada

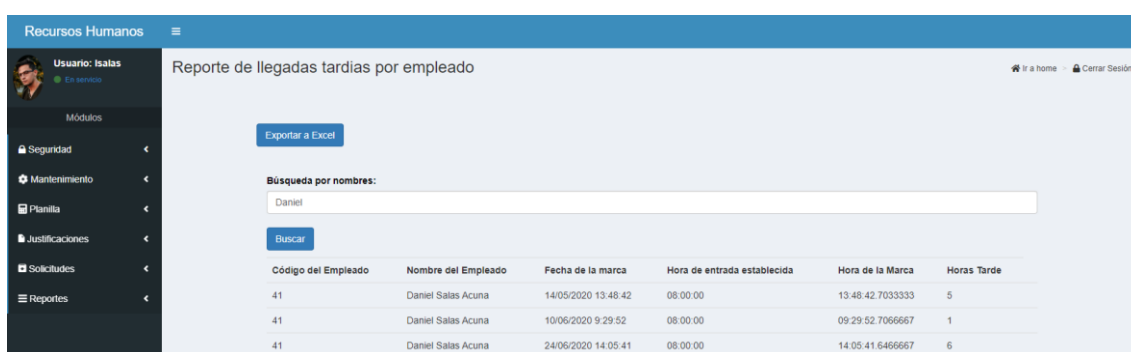
Fuente: (Elaboración propia).

Diseño de reportes usados para la salida de información

En el siguiente apartado se muestra el diseño de los reportes utilizados en la herramienta los cuales están ligados a los archivos utilizados para la salida de información.

A continuación, en la imagen No. 58, se muestra el reporte de llegadas tardías por empleado.

Imagen 60. Reporte llegadas tardías por empleado



Código del Empleado	Nombre del Empleado	Fecha de la marca	Hora de entrada establecida	Hora de la Marca	Horas Tarde
41	Daniel Salas Acuna	14/05/2020 13:48:42	08:00:00	13:48:42.7033333	5
41	Daniel Salas Acuna	10/06/2020 9:29:52	08:00:00	09:29:52.7066667	1
41	Daniel Salas Acuna	24/06/2020 14:05:41	08:00:00	14:05:41.6466667	6

Fuente: (Elaboración Propia)

En la siguiente imagen (No. 59), se muestra el reporte para visualizar las solicitudes según el empleado que las realizó.

Imagen 61. Reporte solicitudes por empleado



Código de Solicitud	Tipo de Solicitud	Nombre de Empleado	Fecha de Registro	Fecha de Solicitud	Descripción	Estado de la Solicitud
38	Vacaciones	Freddy Román Vega	21/06/2020 0:00:00	21/06/2020 0:00:00	Solicitud de vacaciones para el día 21 de junio del 2020	Cerrada
41	Tarde Libre	Freddy Román Vega	24/06/2020 0:00:00	24/06/2020 0:00:00	Solicitud de tarde libre	Registrada

Fuente: (Elaboración Propia)

A continuación, en la imagen No.60, se muestra el reporte para visualizar las justificaciones según el empleado que las realizó.

Imagen 62. *Reporte justificaciones por empleado*

Reporte de Justificaciones por empleado

Exportar a Excel

Búsqueda por nombres:
Bryan

Buscar

Código de Justificación	Tipo de Justificación	Nombre de Empleado	Fecha de Registro	Fecha de Justificación	Descripción	Estado de la Justificación
2	Ausencia	Bryan Delgado Quiros	28/06/2020 0:00:00	27/04/2020 0:00:00	Se justifica ausencia del día 27 de abril del 2020	Registrada

Fuente: (Elaboración Propia)

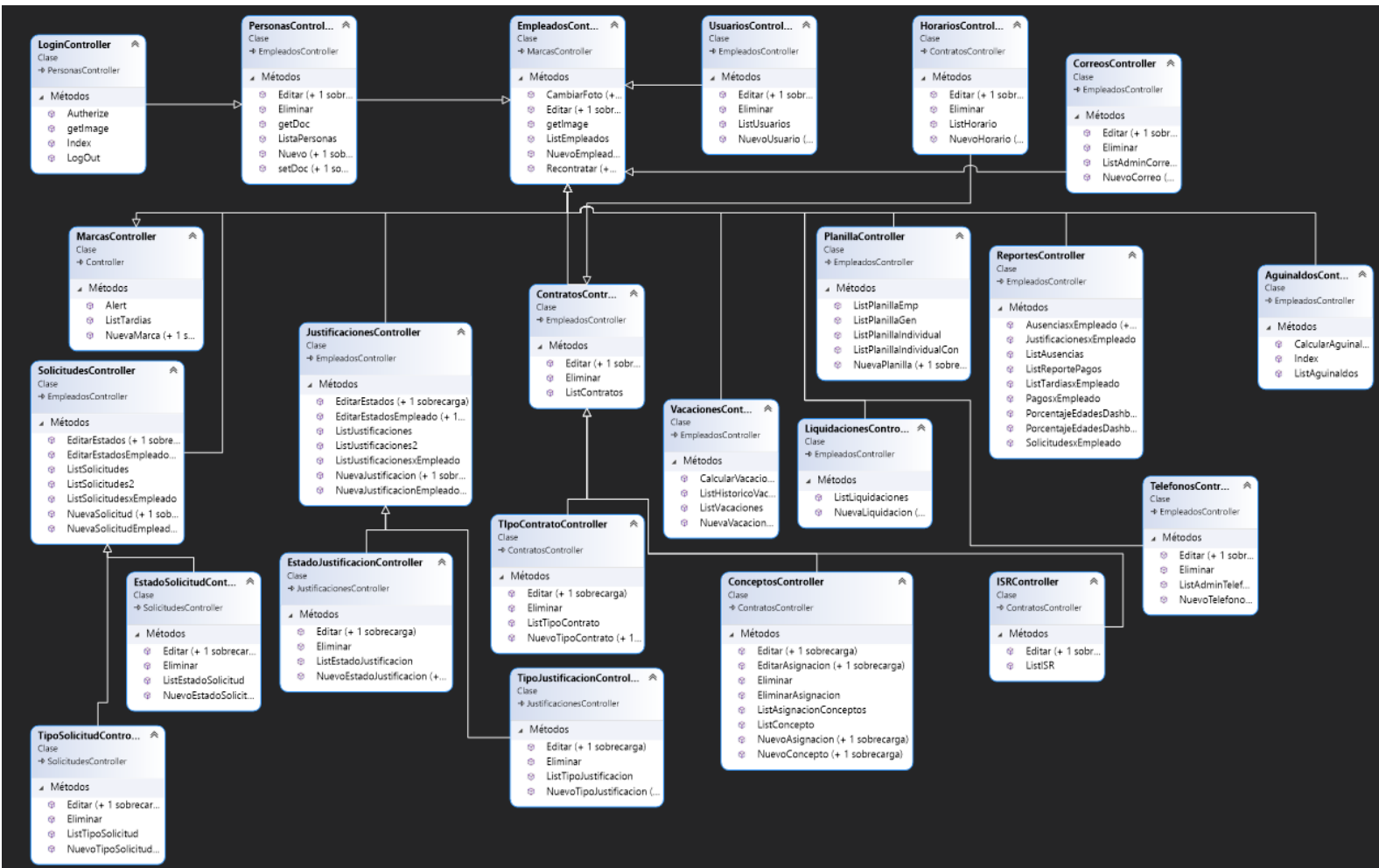
Diagramas UML

A continuación, se puede observar el diagrama de clases utilizadas en el desarrollo de prototipo funcional para el control y gestión del recurso humano en Tienda Karol.

Diagrama de clases

En la imagen No.61, se puede visualizar el diagrama de clases correspondiente al producto de software desarrollado, en este se muestran las clases con sus distintos métodos y relaciones entre ellas.

Imagen 63. Diagrama de clases



Fuente: (Elaboración propia).

Programación

En este apartado se mostrarán pequeños fragmentos del código utilizado para las entradas, salidas, procesos, validaciones y los módulos señalados en el alcance del proyecto.

Entradas y salidas

A continuación, se mostrarán distintos ejemplos del código utilizado para entradas y salidas de datos del prototipo.

Para ejemplificar la entrada de datos se expone por medio de la imagen No. 62 el código utilizado para la modificación de información de una persona.

Imagen 64. Código de entrada

```

139 public ActionResult Editar(int id)
140 {
141     CreatePersonasModel model = new CreatePersonasModel();
142
143     using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
144     {
145         var oPersona = db.Persona.Find(id);
146         var oCorreo = db.Correo.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
147         var oTelefono = db.Telefono.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
148         model.Nombre = oPersona.Nombre;
149         model.Apellido1 = oPersona.Apellido1;
150         model.Apellido2 = oPersona.Apellido2;
151         model.idPersona = oPersona.idPersona;
152         model.Fecha_Nacimiento = oPersona.Fecha_Nacimiento;
153         model.Correo = oCorreo.Correo1;
154         model.Telefono = oTelefono.Telefono1;
155     }
156     return View(model);
157 }
158 [HttpPost]
159 public ActionResult Editar(CreatePersonasModel model)
160 {
161     RRHH_KAROLEntities db = new RRHH_KAROLEntities();
162     try
163     {
164         if (ModelState.IsValid)
165         {
166             using (db)
167             {
168                 var oPersona = db.Persona.Find(model.idPersona);
169                 var oCorreo = db.Correo.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
170                 var oTelefono = db.Telefono.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
171                 oPersona.idPersona = model.idPersona;
172                 oPersona.Nombre = model.Nombre;
173                 oPersona.Apellido1 = model.Apellido1;
174                 oPersona.Apellido2 = model.Apellido2;
175                 oPersona.Fecha_Nacimiento = model.Fecha_Nacimiento;
176                 oCorreo.Correo1 = model.Correo;
177                 oTelefono.Telefono1 = model.Telefono;
178                 db.Entry(oPersona).State = System.Data.Entity.EntityState.Modified;
179                 db.Entry(oCorreo).State = System.Data.Entity.EntityState.Modified;
180                 db.Entry(oTelefono).State = System.Data.Entity.EntityState.Modified;
181                 db.SaveChanges();
182             }
183             return Redirect("~/Personas/ListaPersonas");
184         }
185         return View(model);
186     }
187     catch (Exception)
188     {
189         TempData["Error"] = "Ha ocurrido un error, verifique la información ingresada";
190         ViewBag.Mensaje = TempData["Error"];
191         return RedirectToAction("ListaPersonas");
192     }
193 }
194 }
195 }

```

Fuente: (Elaboración propia).

A continuación, en la imagen No. 63 se muestra el código para mostrar la salida de datos en forma de tabla de personas registradas en el sistema.

Imagen 65. Salida de datos

```
O referencias
22 public ActionResult ListaPersonas()
23 {
24     List<ListPersonasViewModel> list;
25     try
26     {
27         if (TempData["Error"] == null || TempData["Exito"] == null)
28         {
29             TempData["Error"] = string.Empty;
30             TempData["Exito"] = string.Empty;
31         }
32         else
33         {
34             ViewBag.Mensaje = TempData["Error"].ToString();
35             ViewBag.Mensaje2 = TempData["Exito"].ToString();
36         }
37     }
38     catch (Exception)
39     {
40     }
41     throw;
42 }
43
44 using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
45 {
46     list = (from d in db.Persona
47             join Correo in db.Correo on d.idPersona equals Correo.Persona_idPersona
48             join Telefono in db.Telefono on d.idPersona equals Telefono.Persona_idPersona
49             select new ListPersonasViewModel
50             {
51                 idPersona = d.idPersona,
52                 Nombre = d.Nombre,
53                 Apellido1 = d.Apellido1,
54                 Apellido2 = d.Apellido2,
55                 Fecha_Nacimiento = d.Fecha_Nacimiento,
56                 Correo = Correo.Correo1,
57                 Telefono = Telefono.Telefono1
58             }).ToList();
59 }
60
61 return View(list);
62 }
63
64
```

Fuente: (Elaboración propia).

Procesos

En la siguiente imagen (No.64), se muestra el código utilizado por el prototipo para realizar el proceso de cálculo de planilla y mostrar los resultados al usuario con toda la información necesaria.

Imagen 66. Código de proceso

```

0 referencias
131 public ActionResult NuevaPlanilla()
132 {
133     try
134     {
135         ViewBag.Mensaje = TempData["Error"].ToString();
136     }
137     catch
138     {
139     }
140     List<SelectListItem> meses = new List<SelectListItem>();
141     using (Models.RRHH_KAROLEntities db = new Models.RRHH_KAROLEntities())
142     {
143         meses.Add(new SelectListItem() { Text = "Enero", Value = "01" });
144         meses.Add(new SelectListItem() { Text = "Febrero", Value = "02" });
145         meses.Add(new SelectListItem() { Text = "Marzo", Value = "03" });
146         meses.Add(new SelectListItem() { Text = "Abril", Value = "04" });
147         meses.Add(new SelectListItem() { Text = "Mayo", Value = "05" });
148         meses.Add(new SelectListItem() { Text = "Junio", Value = "06" });
149         meses.Add(new SelectListItem() { Text = "Julio", Value = "07" });
150         meses.Add(new SelectListItem() { Text = "Agosto", Value = "08" });
151         meses.Add(new SelectListItem() { Text = "Septiembre", Value = "09" });
152         meses.Add(new SelectListItem() { Text = "Octubre", Value = "10" });
153         meses.Add(new SelectListItem() { Text = "Noviembre", Value = "11" });
154         meses.Add(new SelectListItem() { Text = "Diciembre", Value = "12" });
155     }
156     ViewBag.Opciones = meses;
157 }
158 return View();
159 }
160 [HttpPost]
0 referencias
161 public ActionResult NuevaPlanilla(CreatePlanilla model)
162 {
163     EmpID_S E = new EmpID_S();
164     List<EmpID_S> results;
165     try
166     {
167         if (ModelState.IsValid)
168         {
169             using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
170             {
171                 results = db.Database.SqlQuery<EmpID_S>
172                     (@"select E.idEmpleado idEmpleado
173                        from Empleados E
174                        join Persona P on E.Persona_idPersona = P.idPersona
175                        join Contratos C on C.idContrato = E.Contratos_idContrato"
176                        ).ToList();
177             }
178             foreach (var item in results)
179             {
180                 db.CalcularPlanilla(item.idEmpleado, model.mes, model.año);
181             }
182             return Redirect("ListPlanillaGen/");
183         }
184         return View(model);
185     }
186     catch (Exception ex)
187     {
188         if (ex.ToString().Contains("System.Data.SqlClient.SqlException: Violation of PRIMARY KEY constraint 'PK_Planilla_48F05C9029BC88A'.")
189         {
190             TempData["Error"] = "Ya fue calculada la planilla del mes " + model.mes + " y el año " + model.año;
191         }
192         else
193         {
194             TempData["Error"] = "Ha ocurrido un error, contacte al administrador " + ex.Message;
195         }
196     }
197     ViewBag.Mensaje = TempData["Error"];
198     return RedirectToAction("ListPlanillaGen");
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }

```

Fuente: (Elaboración propia).

Validación

A continuación, en la imagen No. 65, se muestra un fragmento de código en donde el sistema valida la información ingresada por el usuario para el ingreso a la herramienta. Si las credenciales ingresadas por el usuario son correctas puede ingresar a la herramienta, de lo contrario se le mostrará en pantalla un mensaje de error.

Imagen 67. Código de validación

```

33 [HttpPost]
34
35 @referencias
36 public ActionResult Authorize(finalrh.Models.Usuarios usrModel)
37 {
38     using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
39     {
40         var userDetails = db.Usuarios.Where(x => x.NomUsuario == usrModel.NomUsuario && x.Contrasena == usrModel.Contrasena).FirstOrDefault();
41
42         if (userDetails == null)
43         {
44             usrModel.LoginErrorMessage = "Credenciales incorrectas";
45             TempData["Error"] = "Credenciales incorrectas";
46             ViewBag.Mensaje = TempData["Error"];
47             return RedirectToAction("Index");
48             //return View("Index", usrModel);
49         }
50
51         if (userDetails.Activo == false)
52         {
53             usrModel.LoginErrorMessage = "Este usuario se encuentra deshabilitado";
54             TempData["Error"] = "Este usuario se encuentra deshabilitado";
55             ViewBag.Mensaje = TempData["Error"];
56             return RedirectToAction("Index");
57             //return View("Index", usrModel);
58         }
59         else
60         {
61             var Empleado1 = (from Empleados in db.Empleados
62                             join Usuario in db.Usuarios on Empleados.idEmpleado equals userDetails.Empleados_idEmpleado
63                             where Usuario.NomUsuario == userDetails.NomUsuario
64                             select Empleados).FirstOrDefault();
65             Session["userID"] = userDetails.idUsuario;
66             Session["userName"] = userDetails.NomUsuario;
67             Session["TypeUser"] = userDetails.Tipo_Usuario_idTipousuario;
68             Session["EmployeeID"] = userDetails.Empleados_idEmpleado;
69             Session["Foto"] = Empleado1.Imagen;
70             return RedirectToAction("Index", "Home");
71         }
72     }
73 }

```

Fuente: (Elaboración propia).

Módulos señalados en el alcance

- Seguridad:

En la imagen No. 66 se muestra el fragmento de código en el cual se crea un usuario para un empleado previamente registrado en el sistema. Con este usuario el empleado ya podrá ingresar a la herramienta

Imagen 68. Código Módulo Seguridad – Crear usuario

```

127 [HttpPost]
128 0 referencias
129 public ActionResult NuevoUsuario(NuevoUsuario model)
130 {
131     var oUsuario = new Usuarios();
132     int permiso = 0;
133     List<UsuariosView> listUsuarios;
134     try
135     {
136         if (ModelState.IsValid)
137         {
138             using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
139             {
140                 listUsuarios =
141                 (from d in db.Usuarios
142                  select new UsuariosView
143                  {
144                      ID_Usuario = d.idUsuario,
145                      ID_Empleado = d.Empleados_idEmpleado
146                  }).ToList();
147                 foreach (var item in listUsuarios)
148                 {
149                     if (item.ID_Empleado == model.Empleado_idEmpleado)
150                     {
151                         permiso = 0;
152                     }
153                     else
154                     {
155                         permiso = 1;
156                     }
157                 }
158                 if (permiso == 1)
159                 {
160                     oUsuario.NomUsuario = model.NombreUsuario;
161                     oUsuario.Contrasena = model.Contrasena;
162                     oUsuario.Tipo_Usuario_idTipoUsuario = model.Tipo_Usuario;
163                     oUsuario.Empleados_idEmpleado = model.Empleado_idEmpleado;
164                     oUsuario.Activo = model.Activo;
165
166                     db.Usuarios.Add(oUsuario);
167
168                     db.SaveChanges();
169                 }
170                 else
171                 {
172                     TempData["Error"] = "El empleado seleccionado ya cuenta con un usuario registrado";
173                     ViewBag.Mensaje = TempData["Error"];
174                 }
175             }
176             return Redirect("ListUsuarios/");
177         }
178         return View(model);
179     }
180     catch (Exception)
181     {
182         TempData["Error"] = "Ha ocurrido un error, contacte al administrador del sistema";
183         ViewBag.Mensaje = TempData["Error"];
184         return RedirectToAction("ListUsuarios");
185     }
186 }

```

Fuente: (Elaboración propia).

- Mantenimiento

A continuación, en las imágenes No. 67, No. 68 y No.69 se muestra el código implementado en el módulo de mantenimiento, en estas imágenes se toma como ejemplo el mantenimiento de personas.

Imagen 69. Código mantenimiento de personas – Nueva persona

```

[HttpPost]
0 referencias
public ActionResult Nuevo(CreatePersonasModel model)
{
    try
    {
        var oPersona = new Persona();
        var oCurriculum = new Curriculum();
        var oCorreo = new Correo();
        var oTelefono = new Telefono();
        HttpPostedFileBase FileBase = Request.Files[0];
        byte[] file = null;
        Stream myStream = FileBase.InputStream;

        using (MemoryStream ms = new MemoryStream())
        {
            myStream.CopyTo(ms);
            file = ms.ToArray();
        }
        var supportedTypes = new[] { ".txt", ".doc", ".docx", ".pdf", ".xls", ".xlsx" };
        var fileExt = System.IO.Path.GetExtension(FileBase.FileName).Substring(1);
        if (!supportedTypes.Contains(fileExt))
        {
            TempData["Error"] = "El formato del archivo seleccionado no es válido - Solo se pueden subir archivos WORD/PDF/EXCEL/TXT";
            ViewBag.Mensaje = TempData["Error"];
            return RedirectToAction("ListaPersonas");
        }

        if (ModelState.IsValid)
        {
            using (RRHH_KARDEntities db = new RRHH_KARDEntities())
            {
                oPersona.idPersona = model.idPersona;
                oPersona.Nombre = model.Nombre;
                oPersona.Apellido1 = model.Apellido1;
                oPersona.Apellido2 = model.Apellido2;
                oPersona.Fecha_Nacimiento = model.Fecha_Nacimiento;
                oCorreo.Persona_idPersona = model.idPersona;
                oCorreo.Correo = model.Correo;
                oTelefono.Persona_idPersona = model.idPersona;
                oTelefono.Telefono1 = model.Telefono;
                oCurriculum.Nombre = "Curriculum: " + model.Nombre + " " + model.Apellido1 + " " + model.Apellido2;
                oCurriculum.Document = file;
                oCurriculum.NombreReal = FileBase.FileName;
                oCurriculum.Persona_idPersona = model.idPersona;
                db.Persona.Add(oPersona);
                db.Correo.Add(oCorreo);
                db.Telefono.Add(oTelefono);
                db.Curriculum.Add(oCurriculum);
                db.SaveChanges();
            }
            return Redirect("ListaPersonas/");
        }

        return View(model);
    }
    catch (Exception ex)
    {
        TempData["Error"] = "Se ha producido un error: " + ex.Message + " contacte al administrador";
        ViewBag.Mensaje = TempData["Error"];
        return RedirectToAction("ListaPersonas");
    }
}

```

Fuente: (Elaboración propia).

Imagen 70. Código mantenimiento de personas – Editar persona

```

0 referencias
public ActionResult Editar(int id)
{
    CreatePersonasModel model = new CreatePersonasModel();

    using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
    {
        var oPersona = db.Persona.Find(id);
        var oCorreo = db.Correo.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
        var oTelefono = db.Telefono.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
        model.Nombre = oPersona.Nombre;
        model.Apellido1 = oPersona.Apellido1;
        model.Apellido2 = oPersona.Apellido2;
        model.idPersona = oPersona.idPersona;
        model.Fecha_Nacimiento = oPersona.Fecha_Nacimiento;
        model.Correo = oCorreo.Correo1;
        model.Telefono = oTelefono.Telefono1;
    }
    return View(model);
}

[HttpPost]
0 referencias
public ActionResult Editar(CreatePersonasModel model)
{
    RRHH_KAROLEntities db = new RRHH_KAROLEntities();
    try
    {
        if (ModelState.IsValid)
        {
            using (db)
            {
                var oPersona = db.Persona.Find(model.idPersona);
                var oCorreo = db.Correo.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
                var oTelefono = db.Telefono.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
                oPersona.idPersona = model.idPersona;
                oPersona.Nombre = model.Nombre;
                oPersona.Apellido1 = model.Apellido1;
                oPersona.Apellido2 = model.Apellido2;
                oPersona.Fecha_Nacimiento = model.Fecha_Nacimiento;
                oCorreo.Correo1 = model.Correo;
                oTelefono.Telefono1 = model.Telefono;
                db.Entry(oPersona).State = System.Data.Entity.EntityState.Modified;
                db.Entry(oCorreo).State = System.Data.Entity.EntityState.Modified;
                db.Entry(oTelefono).State = System.Data.Entity.EntityState.Modified;
                db.SaveChanges();
            }

            return Redirect("~/Personas/ListaPersonas");
        }
        return View(model);
    }
    catch (Exception)
    {
        TempData["Error"] = "Ha ocurrido un error, verifique la información ingresada";
        ViewBag.Mensaje = TempData["Error"];
        return RedirectToAction("ListaPersonas");
    }
}

```

Fuente: (Elaboración propia).

Imagen 71. Mantenimiento de Personas – Eliminar Persona

```

public ActionResult Eliminar(int id)
{
    try
    {
        using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
        {
            var oCurriculum = db.Curriculum.Where(e => e.Persona_idPersona == id).FirstOrDefault();
            var oPersona = db.Persona.Find(id);
            var oCorreo = db.Correo.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
            var oTelefono = db.Telefono.Where(x => x.Persona_idPersona == oPersona.idPersona).FirstOrDefault();
            if (oCurriculum != null || oCorreo != null)
            {
                db.Curriculum.Remove(oCurriculum);
                db.Persona.Remove(oPersona);
                db.Correo.Remove(oCorreo);
                db.Telefono.Remove(oTelefono);
                db.SaveChanges();
            }
            else
            {
                db.Persona.Remove(oPersona);
                db.SaveChanges();
            }
        }
        return Redirect("~/Personas/ListaPersonas");
    }
    catch (Exception ex)
    {
        TempData["Error"] = "Se ha producido el error: " + ex.Message + "contacte al administrador";
        ViewBag.Mensaje = TempData["Error"];
        return RedirectToAction("ListaPersonas");
    }
}

```

Fuente: (Elaboración propia).

- Planilla:

A continuación, en la imagen No. 70 se muestra el fragmento del código utilizado para el proceso de generación de planilla, mientras que en la imagen No. 71 se muestra el código utilizado en el prototipo para la asignación de vacaciones.

Imagen 72. Código cálculo de planilla

```

public ActionResult NuevaPlanilla()
{
    try
    {
        ViewBag.Mensaje = TempData["Error"].ToString();
    }
    catch
    {
    }
    List<SelectListItem> meses = new List<SelectListItem>();
    using (Models.RRHH_KAROLEntities db = new Models.RRHH_KAROLEntities())
    {
        meses.Add(new SelectListItem() { Text = "Enero", Value = "01" });
        meses.Add(new SelectListItem() { Text = "Febrero", Value = "02" });
        meses.Add(new SelectListItem() { Text = "Marzo", Value = "03" });
        meses.Add(new SelectListItem() { Text = "Abril", Value = "04" });
        meses.Add(new SelectListItem() { Text = "Mayo", Value = "05" });
        meses.Add(new SelectListItem() { Text = "Junio", Value = "06" });
        meses.Add(new SelectListItem() { Text = "Julio", Value = "07" });
        meses.Add(new SelectListItem() { Text = "Agosto", Value = "08" });
        meses.Add(new SelectListItem() { Text = "Septiembre", Value = "09" });
        meses.Add(new SelectListItem() { Text = "Octubre", Value = "10" });
        meses.Add(new SelectListItem() { Text = "Noviembre", Value = "11" });
        meses.Add(new SelectListItem() { Text = "Diciembre", Value = "12" });
    }

    ViewBag.Opciones = meses;
}
return View();
}
[HttpPost]
public ActionResult NuevaPlanilla(CreatePlanilla model)
{
    EmpID_S E = new EmpID_S();
    List<EmpID_S> results;
    try
    {
        if (ModelState.IsValid)
        {
            using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
            {
                results = db.Database.SqlQuery<EmpID_S>
                (@"select E.idEmpleado idEmpleado
                from Empleados E
                join Persona P on E.Persona_idPersona = P.idPersona
                join Contratos C on C.idContrato = E.Contratos_idContrato"
                ).ToList();

                foreach (var item in results)
                {
                    db.CalcularPlanilla(item.idEmpleado, model.mes, model.año);
                }
            }
            return Redirect("ListPlanillaGen/");
        }
        return View(model);
    }
    catch (Exception ex)
    {
        if (ex.ToString().Contains("System.Data.SqlClient.SqlException: Violation of PRIMARY KEY constraint 'PK_Planilla_4BF85C9029BCB8A'.")
        {
            TempData["Error"] = "Ya fue calculada la planilla del mes " + model.mes + " y el año " + model.año;
        }
        else
        {
            TempData["Error"] = "Ha ocurrido un error, contacte al administrador " + ex.Message;
        }
    }

    ViewBag.Mensaje = TempData["Error"];
    return RedirectToAction("ListPlanillaGen");
}

```

Fuente: (Elaboración propia).

Imagen 73. Código nueva vacación

```

210 public ActionResult NuevaVacacion(VacacionesManager model)
211 {
212     try
213     {
214         if (ModelState.IsValid)
215         {
216             using (RHHM_KAROLEntities db = new RHHM_KAROLEntities())
217             {
218                 List<VacacionesManager> list;
219                 List<VacacionesManager> list2;
220                 list = (from d in db.Vacaciones
221                     join e in db.Empleados on d.Empleados_idEmpleado equals e.idEmpleado
222                     join p in db.Persona on e.Persona_idPersona equals p.idPersona
223                     select new VacacionesManager
224                     {
225                         idVacacion = d.idVacacion,
226                         Total dias = d.Total_dias
227                     }).ToList();
228
229                 list2 = (from d in db.Vacaciones
230                     join e in db.Empleados on d.Empleados_idEmpleado equals e.idEmpleado
231                     join p in db.Persona on e.Persona_idPersona equals p.idPersona
232                     join h in db.Historico_Vacaciones on d.idVacacion equals h.Vacaciones_idVacacion
233                     select new VacacionesManager
234                     {
235                         idVacacion = d.idVacacion,
236                         Total dias = d.Total_dias,
237                         Fecha_Ini = h.fecha_ini,
238                         Fecha_Fin = h.fecha_fin
239                     }).ToList();
240
241                 if (model.Fecha_Ini < DateTime.Now.Date)
242                 {
243                     TempData["Error"] = "La fecha de inicio que está ingresando ya pasó";
244                     ViewBag_Mensaje = TempData["Error"];
245                     return RedirectToAction("ListVacaciones");
246                 }
247                 if (model.Fecha_Ini == model.Fecha_Fin)
248                 {
249                     TempData["Error"] = "La fecha de inicio no puede ser igual a la fecha final";
250                     ViewBag_Mensaje = TempData["Error"];
251                     return RedirectToAction("ListVacaciones");
252                 }
253                 if (model.Fecha_Ini > model.Fecha_Fin)
254                 {
255                     TempData["Error"] = "La fecha de inicio no puede ser mayor a la de finalización";
256                     ViewBag_Mensaje = TempData["Error"];
257                     return RedirectToAction("ListVacaciones");
258                 }
259                 foreach (var item in list)
260                 {
261                     if (item.idVacacion == model.idVacacion && item.Total_dias == 0)
262                     {
263                         TempData["Error"] = "El empleado seleccionado no cuenta con vacaciones disponibles";
264                         ViewBag_Mensaje = TempData["Error"];
265                         return RedirectToAction("ListVacaciones");
266                     }
267                 }
268                 foreach (var item in list2)
269                 {
270                     if (item.idVacacion == model.idVacacion && item.Fecha_Ini == model.Fecha_Ini || item.Fecha_Fin == model.Fecha_Fin)
271                     {
272                         TempData["Error"] = "El empleado seleccionado ya cuenta con vacaciones asignadas en la fecha seleccionada";
273                         ViewBag_Mensaje = TempData["Error"];
274                         return RedirectToAction("ListVacaciones");
275                     }
276                 }
277
278                 int daysDiff = ((TimeSpan)(model.Fecha_Fin - model.Fecha_Ini)).Days;
279                 foreach (var item in list)
280                 {
281                     if (item.idVacacion == model.idVacacion && item.Total_dias < daysDiff)
282                     {
283                         TempData["Error"] = "El empleado seleccionado no cuenta con vacaciones disponibles ha seleccionado " + daysDiff + " Dias y solo cuenta con " + item.Total_dias + " Dias.";
284                         ViewBag_Mensaje = TempData["Error"];
285                         return RedirectToAction("ListVacaciones");
286                     }
287                 }
288
289                 db.Database.ExecuteSqlCommand("EXEC GenerarVacacion @p0, @p1, @p2",
290                     model.idVacacion, // Valor 1er parametro: @p0
291                     model.Fecha_Ini, // Valor 1er parametro: @p1
292                     model.Fecha_Fin); // Valor 2do parametro: @p2
293                 return RedirectToAction("ListVacaciones");
294             }
295             return View(model);
296         }
297         catch (Exception ex)
298         {
299             TempData["Error"] = "Ha ocurrido un error, contacte al administrador " + ex.Message;
300         }
301
302         ViewBag_Mensaje = TempData["Error"];
303         return RedirectToAction("ListVacaciones");
304     }
305 }
306
307
308
309
310
311
312
313
314
315
316
317

```

Fuente: (Elaboración propia).

- Justificaciones

En la imagen No. 72 y No. 73 se puede observar la codificación realizada para el registro de una nueva justificación.

Imagen 74. *Parte Uno – Código Justificación*

```
[HttpPost]
0 referencias
public ActionResult NuevaJustificacion(JustificacionesManager model)
{
    try
    {
        if (ModelState.IsValid)
        {
            using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
            {
                List<ListCorreos> listCorreos = (from P in db.Persona
                join E in db.Empleados on P.idPersona equals E.Persona_idPersona
                join U in db.Usuarios on E.idEmpleado equals U.Empleados_idEmpleado
                join C in db.Correo on P.idPersona equals C.Persona_idPersona
                where U.TipoUsuario_idTipoUsuario == 1
                select new ListCorreos
                {
                    idCorreo = C.idCorreo,
                    Persona_idPersona = C.Persona_idPersona,
                    Correo = C.Correo1,
                    NombrePersona = P.Nombre + " " + P.Apellido1 + " " + P.Apellido2
                }).ToList();

                var oJustificacion = new Justificaciones();
                var oEmpleado = db.Empleados.Find(model.IdEmpleado);
                var oPersona = db.Persona.Find(oEmpleado.Persona_idPersona);
                var oTipoJustificacion = db.Tipo_Justificacion.Find(model.IdTipoJustificacion);
                var oEstadoJustificacion = db.Estado_Justificacion.Find(model.IdEstadoJustificacion);
                if (model.FechaIni > DateTime.Now.Date)
                {
                    TempData["Error"] = "La fecha de justificacion no puede ser mayor a la de el día de hoy: " + DateTime.Now.Date;
                    ViewBag.Mensaje = TempData["Error"];
                    return RedirectToAction("ListJustificaciones2");
                }
                else if (model.FechaIni <= DateTime.Now.Date)
                {
                    oJustificacion.idJustificacion = model.IdJustificacion;
                    oJustificacion.Tipo_Justificacion_idTipoJustificacion = model.IdTipoJustificacion;
                    oJustificacion.Empleados_idEmpleado = model.IdEmpleado;
                    oJustificacion.Fecha_Registro = DateTime.Now.Date;
                    oJustificacion.Fecha_Justificacion = model.FechaIni;
                    oJustificacion.Estado_Justificacion_idEstado = 1003;
                    oJustificacion.Descripcion = model.Descripcion;
                    db.Justificaciones.Add(oJustificacion);
                    db.SaveChanges();
                    TempData["Exito"] = "Se ha registrado la Justificacion";
                    ViewBag.Mensaje2 = TempData["Exito"];
                }
            }
        }
    }
}
```

Fuente: (Elaboración propia).

Imagen 75. Parte dos – Código Justificación

```

    }
    foreach (var item in listCorreos)
    {
        string colaborador = oPersona.Nombre + " " + oPersona.Apellido1 + " " + oPersona.Apellido2;
        string asunto = "La Justificacion #: " + oJustificacion.idJustificacion + " ha sido registrada por: " + colaborador;
        string stCuerpoHTML = "<!DOCTYPE html>";
        stCuerpoHTML += "<html lang='es'>";
        stCuerpoHTML += "<head>";
        stCuerpoHTML += "<meta charset='utf - 8'>";
        stCuerpoHTML += "<title>La Justificacion # " + oJustificacion.idJustificacion + " se encuentra en el estado registrada " + "</title>";
        stCuerpoHTML += "</head>";
        stCuerpoHTML += "<body style='background - color: black '>";
        stCuerpoHTML += "<table style='max - width: 600px; padding: 10px; margin: 0 auto; border - collapse: collapse; '> ";
        stCuerpoHTML += "<tr>";
        stCuerpoHTML += "<td style='padding: 0 '>";
        stCuerpoHTML += "</td>";
        stCuerpoHTML += "</tr>";
        stCuerpoHTML += "<tr>";
        stCuerpoHTML += "<td style='background - color: #ecf1f1';";
        stCuerpoHTML += "<div style='color: #34495e; margin: 4% 18% 2%; text-align: justify;font-family: sans-serif'>";
        stCuerpoHTML += "<h2 style='color: #e67e22; margin: 0 0 7px'>Estimado(A) administrador(A): " + item.NombrePersona + "</h2>";
        stCuerpoHTML += "<p style='margin: 2px; font - size: 15px'>";
        stCuerpoHTML += "El colaborador(A): " + colaborador + " ha registrado una nueva Justificacion ";
        stCuerpoHTML += "la cual se encuentra en el estado: Registrado ";
        stCuerpoHTML += "<br/>";
        stCuerpoHTML += "<br/>";
        stCuerpoHTML += "A continuación el detalle de la Justificacion: ";
        stCuerpoHTML += "<br/>";
        stCuerpoHTML += "<br/>";
        stCuerpoHTML += "<br/>";
        stCuerpoHTML += oJustificacion.Descripcion;
        stCuerpoHTML += "</p>";
        stCuerpoHTML += "<p style='color: #b3b3b3; font-size: 12px; text-align: center;margin: 30px 0 0'>Copyright © Sistema para el control y gestión del recurso humano Tienda Karol</p>";
        stCuerpoHTML += "</div>";
        stCuerpoHTML += "</td>";
        stCuerpoHTML += "</tr>";
        stCuerpoHTML += "</table>";
        stCuerpoHTML += "</body>";
        stCuerpoHTML += "</html>";
        string para = item.Correo;
        MailMessage correo = new MailMessage();
        correo.From = new MailAddress("tiendakarolpuriscal@gmail.com");
        correo.To.Add(para);
        correo.Subject = asunto;
        correo.Body = stCuerpoHTML;
        correo.IsBodyHtml = true;
        correo.Priority = MailPriority.Normal;

        SmtplibClient smtp = new SmtplibClient();
        smtp.Host = "smtp.gmail.com";
        smtp.Port = 25;
        smtp.EnableSsl = true;
        smtp.UseDefaultCredentials = true;
        string sCuentaCorreo = "tiendakarolpuriscal@gmail.com";
        string sPasswordCorreo = "a";
        smtp.Credentials = new System.Net.NetworkCredential(sCuentaCorreo, sPasswordCorreo);
        smtp.Send(correo);
    }
}
catch (Exception)
{
    throw;
}
return Redirect("ListJustificaciones/");
}
return View(model);
}
catch (Exception ex)
{
    TempData["Error"] = "Se ha producido un error: " + ex.Message + " contacte al administrador";
    ViewBag.Mensaje = TempData["Error"];
    return RedirectToAction("ListJustificaciones");
}
}
}

```

Fuente: (Elaboración propia).

- Solicitudes

En las imágenes No. 74 y No. 75, se puede observar la codificación realizada para el registro de una nueva solicitud.

Imagen 76. Parte uno - Código nueva solicitud

```

public ActionResult NuevaSolicitud(SolicitudesManager model)
{
    try
    {
        if (ModelState.IsValid)
        {
            using (RRHM_KAROLEntities db = new RRHM_KAROLEntities())
            {
                List<ListCorreos> listCorreos = (from P in db.Persona
                                                join E in db.Empleados on P.IdPersona equals E.Persona_idPersona
                                                join U in db.Usuarios on E.IdEmpleado equals U.Empleados_idEmpleado
                                                join C in db.Correo on P.IdPersona equals C.Persona_idPersona
                                                where U.Tipo_Usuario_idTipoUsuario == 1
                                                select new ListCorreos
                                                {
                                                    IdCorreo = C.IdCorreo,
                                                    Persona_idPersona = C.Persona_idPersona,
                                                    Correo = C.Correo1,
                                                    NombrePersona = P.Nombre + " " + P.Apellido1 + " " + P.Apellido2
                                                }).ToList();

                var oSolicitud = new Solicitudes();

                var oEmpleado = db.Empleados.Find(model.IdEmpleado);
                var oPersona = db.Persona.Find(oEmpleado.Persona_idPersona);
                var oTipoSolicitud = db.Tipo_Solicitud.Find(model.IdTipoSolicitud);
                var oEstadoSolicitud = db.Estado_Solicitud.Find(model.IdEstadoSolicitud);

                if (model.FechaIni > model.FechaFin)
                {
                    TempData["Error"] = "La fecha de inicio no puede ser mayor a la fecha final";
                    ViewBag.Mensaje = TempData["Error"];
                    return RedirectToAction("ListSolicitudes2");
                }
                else if (model.FechaIni < DateTime.Now.Date || model.FechaFin < DateTime.Now.Date)
                {
                    TempData["Error"] = "La fechas ingresadas no son válidas";
                    ViewBag.Mensaje = TempData["Error"];
                    return RedirectToAction("ListSolicitudes2");
                }
                else if (model.FechaIni < model.FechaFin)
                {
                    oSolicitud.IdSolicitud = model.IdSolicitud;
                    oSolicitud.Tipo_Solicitud_idTipoSolicitud = model.IdTipoSolicitud;
                    oSolicitud.Empleados_idEmpleado = model.IdEmpleado;
                    oSolicitud.Fecha_Solicitud = DateTime.Now.Date;
                    oSolicitud.Fecha_ini = model.FechaIni;
                    oSolicitud.Fecha_fin = model.FechaFin;
                    oSolicitud.Estado_Solicitud_idEstado = 1002;
                    oSolicitud.Descripcion = model.Descripcion;
                    db.Solicitudes.Add(oSolicitud);
                    db.SaveChanges();
                    TempData["Exito"] = "Se ha registrado la solicitud";
                    ViewBag.Mensaje2 = TempData["Exito"];
                }
            }
        }
    }
}

```

Fuente: (Elaboración propia).

Imagen 77. Parte dos – Código nueva solicitud

```

foreach (var item in listCorreos)
{
    string colaborador = oPersona.Nombre + " " + oPersona.Apellido1 + " " + oPersona.Apellido2;
    string asunto = "La solicitud #: " + oSolicitud.idSolicitud + " ha sido registrada por: " + colaborador;
    string stCuerpoHTML = "<DOCTYPE html>";
    stCuerpoHTML += "<html lang='es'>";
    stCuerpoHTML += "<head>";
    stCuerpoHTML += "<meta charset='utf-8'>";
    stCuerpoHTML += "<title>La solicitud # " + oSolicitud.idSolicitud + " se encuentra en el estado registrada " + "</title>";
    stCuerpoHTML += "</head>";
    stCuerpoHTML += "<body style='background-color: black'>";
    stCuerpoHTML += "<table style='max-width: 600px; padding: 10px; margin: 0 auto; border-collapse: collapse;'> ";
    stCuerpoHTML += "<tr>";
    stCuerpoHTML += "<td style='padding: 0'>";
    //stCuerpoHTML += "<img style='padding: 0; display: block; src='~/img/Logo-Tienda.png' width='100%' height='100%'>";
    stCuerpoHTML += "</td>";
    stCuerpoHTML += "</tr>";
    stCuerpoHTML += "<tr>";
    stCuerpoHTML += "<td style='background-color: #ecf0f1'>";
    stCuerpoHTML += "<div style='color: #34495e; margin: 4% 10% 2%; text-align: justify; font-family: sans-serif'>";
    stCuerpoHTML += "<h2 style='color: #677e22; margin: 0 0 7px'>Estimado(A) administrador(A): " + itoe.NombrePersona + "</h2>";
    stCuerpoHTML += "<p style='margin: 2px; font-size: 15px'>";
    stCuerpoHTML += "El colaborador(A): " + colaborador + " ha registrado una nueva solicitud ";
    stCuerpoHTML += "la cual se encuentra en el estado: Registrado ";
    stCuerpoHTML += "<br/>";
    stCuerpoHTML += "<br/>";
    stCuerpoHTML += "A continuación el detalle de la solicitud: ";
    stCuerpoHTML += "<br/>";
    stCuerpoHTML += "<br/>";
    stCuerpoHTML += "<br/>";
    stCuerpoHTML += oSolicitud.Descripción;
    stCuerpoHTML += "</p>";
    stCuerpoHTML += "<p style='color: #838383; font-size: 12px; text-align: center; margin: 30px 0 0'>Copyright © Sistema para el control y gestión del recurso humano Tienda Karolc/ps";
    stCuerpoHTML += "</div>";
    stCuerpoHTML += "</td>";
    stCuerpoHTML += "</tr>";
    stCuerpoHTML += "</table>";
    stCuerpoHTML += "</body>";
    stCuerpoHTML += "</html>";

    string para = item.Correo;
    MailMessage correo = new MailMessage();
    correo.From = new MailAddress("tiendakarolpuriscal@gmail.com");
    correo.To.Add(para);
    correo.Subject = asunto;
    correo.Body = stCuerpoHTML;
    correo.IsBodyHtml = true;
    correo.Priority = MailPriority.Normal;

    SmtpClient smtp = new SmtpClient();
    smtp.Host = "smtp.gmail.com";
    smtp.Port = 25;
    smtp.EnableSsl = true;
    smtp.UseDefaultCredentials = true;
    string sCuentaCorreo = "tiendakarolpuriscal@gmail.com";
    string sPasswordCorreo = "XXXXXXXXXX";
    smtp.Credentials = new System.Net.NetworkCredential(sCuentaCorreo, sPasswordCorreo);
    smtp.Send(correo);
}

}
catch (Exception ex)
{
    TempData["Error"] = "Se ha producido un error: " + ex.Message + " contacte al administrador";
    ViewBag.Mensaje = TempData["Error"];
}
}

return RedirectToAction("listSolicitudes/");
}

return View(model);
}
catch (Exception ex)
{
    TempData["Error"] = "Se ha producido un error: " + ex.Message + " contacte al administrador";
    ViewBag.Mensaje = TempData["Error"];
    return RedirectToAction("listSolicitudes/");
}
}
}

```

Fuente: (Elaboración propia).

- **Reportes**

En las imágenes No.76 y No.77 se puede observar la codificación realizada para la generación de reportes.

Imagen 78. Código Reportes – Reporte Pagos por Empleado

```

0 referencias
public ActionResult PagosxEmpleado(String Nombre_Empleado)
{
    List<ReportePagos> list;
    List<ReportePagos> result = new List<ReportePagos>();
    try
    {
        if (TempData["Error"] == null || TempData["Exito"] == null)
        {
            TempData["Error"] = string.Empty;
            TempData["Exito"] = string.Empty;
        }
        else
        {
            ViewBag.Mensaje = TempData["Error"].ToString();
            ViewBag.Mensaje2 = TempData["Exito"].ToString();
        }
    }
    catch (Exception)
    {
        throw;
    }

    using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
    {
        list = (from d in db.Planilla
                join e in db.Empleados on d.Empleados_idEmpleado equals e.idEmpleado
                join per in db.Persona on e.Persona_idPersona equals per.idPersona
                select new ReportePagos
                {
                    idEmpleado = e.idEmpleado,
                    NombreEmpleado = per.Nombre + " " + per.Apellido1 + " " + per.Apellido2,
                    Periodo = d.Periodo_Planilla,
                    SueldoBase = d.SueldoBase,
                    SueldoNeto = d.SueldoNeto
                }).ToList();
        if (!String.IsNullOrEmpty(Nombre_Empleado))
        {
            result = list.Where(d => d.NombreEmpleado.Contains(Nombre_Empleado)).ToList();
        }
    }
    return View(result);
}

```

Fuente: (Elaboración Propia)

Imagen 79. Código Reportes – Reporte solicitudes por empleado

```

@referencias
public ActionResult SolicitudesxEmpleado(string Nombre_Empleado)
{
    List<SolicitudesManager> list;
    List<SolicitudesManager> result = new List<SolicitudesManager>();
    try
    {
        if (TempData["Error"] == null || TempData["Exito"] == null)
        {
            TempData["Error"] = string.Empty;
            TempData["Exito"] = string.Empty;
        }
        else
        {
            ViewBag.Mensaje = TempData["Error"].ToString();
            ViewBag.Mensaje2 = TempData["Exito"].ToString();
        }
    }
    catch (Exception)
    {
        throw;
    }

    using (RRHH_KAROLEntities db = new RRHH_KAROLEntities())
    {
        list = (from d in db.Solicitudes
                join S in db.Estado_Solicitud on d.Estado_Solicitud_idEstado equals S.idEstado
                join T in db.Tipo_Solicitud on d.Tipo_Solicitud_idTipoSolicitud equals T.idTipoSolicitud
                join E in db.Empleados on d.Empleados_idEmpleado equals E.idEmpleado
                join P in db.Persona on E.Persona_idPersona equals P.idPersona
                select new SolicitudesManager
                {
                    IdSolicitud = d.idSolicitud,
                    IdTipoSolicitud = d.Tipo_Solicitud_idTipoSolicitud,
                    NombreTipoSolicitud = T.Descripcion,
                    IdEmpleado = E.idEmpleado,
                    NombreEmpleado = P.Nombre + " " + P.Apellido1 + " " + P.Apellido2,
                    FechaSolicitud = d.Fecha_Solicitud,
                    FechaIni = d.Fecha_ini,
                    FechaFin = d.Fecha_fin,
                    IdEstadoSolicitud = d.Estado_Solicitud_idEstado,
                    NombreEstadoSolicitud = S.NombreEstado,
                    Descripcion = d.Descripcion
                }
                ).ToList();

        if (!string.IsNullOrEmpty(Nombre_Empleado))
        {
            result = list.Where(d => d.NombreEmpleado.Contains(Nombre_Empleado)).ToList();
        }
    }
    return View(result);
}

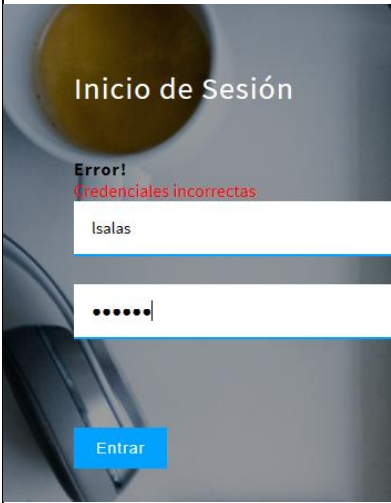
```

Fuente: (Elaboración Propia)

Pruebas



En esta sección se documentan las pruebas realizadas al prototipo, el resultado esperado y el obtenido al realizar las pruebas correspondientes.

Cuadro 61. *Caso de Prueba – Inicio Sesión*

Caso de prueba – Inicio de sesión	
Número caso de prueba: 01	Módulo: Seguridad
<p>Descripción: Esta prueba está dirigida al módulo de seguridad y valida el ingreso al sistema con credenciales de un usuario previamente registrado.</p>	
<p>Resultado esperado: Ingreso denegado cuando las credenciales ingresadas por el usuario son incorrectas.</p>	
<p>Prueba realizada: Se insertan credenciales incorrectas.</p>	
<p>Resultado obtenido: El sistema valida las credenciales ingresadas y no permite el ingreso al detectar que son incorrectas.</p>	<p>El sistema muestra mensaje de error:</p> 



Fuente: (Elaboración propia).

Cuadro 62. Caso de Prueba – Agregar Persona

Caso de prueba – Agregar Persona	
Número caso de prueba: 02	Módulo: Mantenim iento
Descripción: Esta prueba está dirigida al módulo de mantenimiento y la opción de agregar nueva persona, valida el registro inválido de una persona.	
Resultado esperado: El sistema debe validar que la cédula ingresada por el usuario no se encuentre registrada anteriormente y no se duplique.	
Prueba realizada: Se ingresa un número de cédula previamente registrado en el sistema.	
Resultado obtenido: El sistema valida la información ingresada y muestra un mensaje de error si es incorrecta o por el contrario un mensaje de éxito si es correcta.	
Mensaje de Error:	
 Error! La cédula ingresada pertenece a una persona previamente registrada en el sistema	
Mensaje de Éxito:	
 Éxito! La persona se ha registrado con éxito	

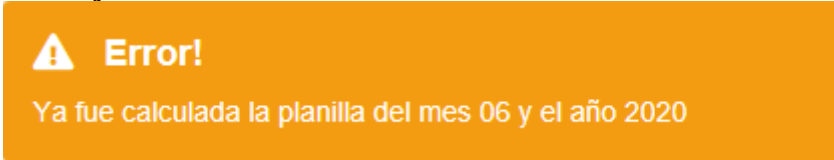
Fuente: (Elaboración propia).

Cuadro 63. Caso de prueba – Agregar Horario

Caso de prueba – Agregar Horario	
Número caso de prueba: 03	Módulo: Mantenimiento- Gestión de campos
Descripción: Esta prueba está dirigida al módulo de mantenimiento, el submódulo gestión de campos la opción de agregar un nuevo horario, valida el registro inválido de un horario.	
Resultado esperado: El sistema debe validar que la hora final del horario no sea menor a la hora de inicio ingresada por el usuario.	
Prueba realizada: Se ingresa una hora de final mayor a la hora de inicio.	
Resultado obtenido: El sistema valida la información ingresada y muestra un mensaje de error si es incorrecta o por el contrario un mensaje de éxito si es correcta.	
Mensaje de Error:	
 Error! La hora de entrada no puede ser mayor a la de salida, favor, validar	
Mensaje de Éxito:	
 Éxito! El horario: Horario caso de prueba 03 se ha registrado con éxito	

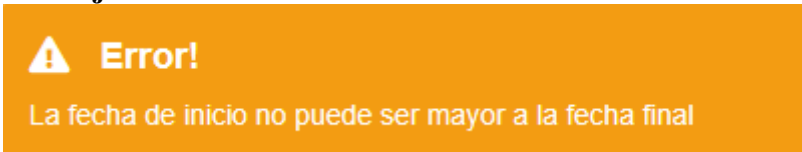
Fuente: (Elaboración propia).

Cuadro 64. *Caso de Prueba – Cálculo de planilla*

Caso de prueba – Cálculo de planilla	
Número caso de prueba: 04	Módulo: Planilla
Descripción: Esta prueba está dirigida al módulo de planilla y la opción de cálculo de planilla, valida el cálculo válido de la planilla	
Resultado esperado: El sistema debe validar que el mes ingresado por el usuario para el cálculo de la planilla no se encuentre calculado anteriormente.	
Prueba realizada: Se ingresa un número mes de planilla previamente calculado en el sistema.	
Resultado obtenido: El sistema valida la información ingresada y muestra un mensaje de error si es incorrecta.	
Mensaje de Error:	
	

Fuente: (Elaboración propia).

Cuadro 65. *Caso de Prueba – Nueva Solicitud*

Caso de prueba – Nueva Solicitud	
Número caso de prueba: 05	Módulo: Solicitudes
Descripción: Esta prueba está dirigida al módulo de solicitud y la opción de nueva solicitud.	
Resultado esperado: El sistema debe validar que la fecha final de la solicitud no sea mayor a la fecha de inicio.	
Prueba realizada: Se ingresa una fecha final menor a la fecha de inicio de la solicitud.	
Resultado obtenido: El sistema valida la información ingresada y muestra un mensaje de error si es incorrecta.	
Mensaje de Error:	
	

Fuente: (Elaboración propia)

Referencias

Arias, F. G. (2012). *El Proyecto de Investigación Sexta Edición*. Venezuela: EPISTEME.

Alvarez, M. A. (11 de Diciembre de 2019). *Qué es la programación orientada a objetos*. Obtenido de <https://desarrolloweb.com/articulos/499.php>

Amazon AWS. (2020). *¿Qué es una base de datos relacional?* Obtenido de <https://aws.amazon.com/es/relational-database/#:~:text=Blog%20de%20bases%20de%20datos,tablas%20con%20columnas%20y%20filas>.

Bird, R (2000). Introducción a la programación funcional con haskell. En R. Bird, *Introducción a la programación funcional con haskell* (pág. 1). Madrid: PEARSON EDUCACIÓN.

Carballo Mena, C., & Villasmil Prieto, H. (2005). *Ley Orgánica del trabajo y su reglamento*. Caracas: Publicaciones UCAB.

Ceballos Sierra, F. J. (2015). Enciclopedia de Microsoft Visual C#: interfaces gráficas y aplicaciones para Internet con Windows Forms y ASP.NET (4a. ed.). Madrid, Spain: RAMA Editorial. Recuperado de <https://elibro.net/es/ereader/bibliouia/62510>

Cabezas E., Andrade D., Torres J., (2018), *Introducción a la metodología de la investigación científica*, Sangolquí, Ecuador, Comisión Editorial de la Universidad de las Fuerzas Armadas ESPE.

Cauas, D. (2015). Definición de las variables, enfoque y tipo de investigación. *Bogotá: biblioteca electrónica de la universidad Nacional de Colombia, 2*.

Del Castillo, C. C. y Olivares Orozco, S. (2015). Metodología de la investigación. México D.F, Mexico: Grupo Editorial Patria. Recuperado de <https://elibro.net/es/ereader/bibliouia/39410?page=133>.

- Del Castillo, C. C. y Olivares Orozco, S. (2015). Metodología de la investigación. México D.F, Mexico: Grupo Editorial Patria. Recuperado de <https://elibro.net/es/ereader/bibliouia/39410?page=129>.
- Díaz, J. Harari, I. y Paola Amadeo, A. (2013). Guía de recomendaciones para diseño de software centrado en el usuario. Argentina: D - Editorial de la Universidad Nacional de La Plata. Recuperado de <https://elibro.net/es/ereader/bibliouia/66410?page=48>.
- Domínguez, P. (2017). *En qué consiste el modelo en cascada*. Obtenido de <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>
- Diaz, A. (03 de Febrero de 2010). *Cuando usar ASP.NET MVC*. Obtenido de <https://highscalability.wordpress.com/2010/02/03/cuando-usar-asp-net-mvc/>
- Flores Villalpando, R. (2014). Administración de recursos humanos. México, D.F, Mexico: Editorial Digital UNID. Recuperado de <https://elibro.net/es/ereader/bibliouia/41180?page=3>.
- Flores, M. I. N. (2007). Las variables: Estructura y función en la hipótesis. *Investigación educativa*, 11(20), 163-182.
- Gorman, K., Hirt, A., Noderer, D., Rowland-Jones, J., Sirpal, A., Ryan, D., and Woody, B (2019). *Introducing Microsoft SQL Server 2019. Reliability, scalability, and security both on premises and in the cloud*. Packt Publishing.
- Insignia (2017). El insignia: *Tipos de encuestas y sus características*. Recuperado de: <https://blog.elinsignia.com/2017/06/13/tipos-de-encuestas-y-sus-caracteristicas/>
- Juganaru Mathieu, M. (2015). Introducción a la programación. México D.F, Mexico: Grupo Editorial Patria. Recuperado de <https://elibro.net/es/ereader/bibliouia/39449>

Lacalle García, G. (2011). Operaciones administrativas de recursos humanos. Madrid: Editex.

López, O., Laguna, M. A., & Corral, J. M. M. (2000). Generación Automática de Casos de Uso para Desarrollo de Software Basado en Reutilización. In *JISBD* (pp. 89-102).

López, P. L. (2004). Población muestra y muestreo. *Punto cero*, 9(08), 69-74.

Moreno Pérez, J. C. y Santos González, M. (2015). Sistemas informáticos y redes locales. Madrid, Spain: RA-MA Editorial. Recuperado de <https://elibro.net/es/ereader/bibliouia/62492>

Marini, E. (2012). *El modelo cliente/servidor*. Obtenido de: <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>

Martínez, A., & Eder, G. (2018). Desarrollo de un sistema de planillas para la mejora de la gestión de planillas del personal en la embotelladora Enrique Cassinelli e hijos sac.

Murillo, W. (2008). La investigación científica. Recuperado de: <https://www.monografias.com/trabajos15/invest-cientifica/investcientifica.shtml>

Martínez Ruiz, H. (2012). Metodología de la investigación. México, D.F, Mexico: Cengage Learning. Recuperado de <https://elibro.net/es/ereader/bibliouia/39957?page=151>.

Moreno, E. (2018). *Definición Instrumental de las variables*. Obtenido de <http://tesis-investigacion-cientifica.blogspot.com/2018/03/definicion-instrumental-de-las-variables.html>

Microsoft (2020). Ilustración de Costos Azure por App Service [Imagen]. Recuperado de: <https://azure.microsoft.com/es-es/pricing/details/app-service/windows/>

Microsoft (2020). Ilustración de Costos Azure SQL Database [Imagen]. Recuperado de: <https://azure.microsoft.com/es-es/pricing/details/sql-database/single/>

Prieto Espinosa, A. (2005). Conceptos de informática. Aravaca, Madrid, Spain: McGraw-Hill España. Recuperado de <https://elibro.net/es/ereader/bibliouia/50305>

- Pulido Romero, E. Escobar Domínguez, Ó. y Núñez Pérez, J. Á. (2019). Base de datos. Grupo Editorial Patria. Recuperado de <https://elibro.net/es/ereader/bibliouia/121283>
- Pérez, M. (2011). *Microsoft SQL Server 2008 R2. Motor de base de datos y administración*. RC Libros.
- Porto, J. P., & Merino, M. (2015). *Definición de Prototipo*. Obtenido de <https://definicion.de/prototipo/>
- Peláez, A., Rodríguez, J., Ramírez, S., Pérez, L., Vázquez, A., & González, L. (2013). La entrevista. *Universidad autónoma de México*. [En línea].[Online].[cited 2012 Septiembre 30. Disponible en: http://www.uam.es/personal_pdi/stmaria/jmurillo/InvestigacionEE/Presentaciones/Curso_10/E.
- Question Pro. (2020). *¿Cómo determinar el tamaño de la muestra de una investigación de mercados?* Recuperado de: <https://www.questionpro.com/blog/es/como-determinar-el-tamano-de-una-muestra/>
- Quintana, G. (2014). *Aprende SQL*. Castelló de la Plana, Spain: Universitat Jaume I. Servei de Comunicació i Publicacions. Recuperado de: <https://elibro.net/es/ereader/bibliouia/53252?page=8>
- Real Academia Española: *Diccionario de la lengua española*, 23.^a ed., [versión 23.3 en línea]. <<https://dle.rae.es>> [11/06/2020].
- Rivas, C. I., Corona, V. P., Gutiérrez, J. F., & Hernández, L. (2015). Metodologías actuales de desarrollo de software. *Revista de Tecnología e Innovación*, 2(5), 980-986.
- Robles, F. (2017). *Investigación Pura: Características, Tipos, Ejemplos*. Recuperado de: <https://www.lifeder.com/investigacion-pura/>
- R. H., Collado, C. F., Lucio, P. B., Valencia, S. M., & Torres, C. P. M. (2006). *Metodología de la investigación* (Vol. 6). México, DF: Mcgraw-hill.

Ruiz Medina, M. I. (marzo de 2011). Tesis Doctoral: Políticas públicas en salud y su impacto en el seguro popular en Culiacán, Sinaloa, México. Recuperado de: http://www.eumed.net/tesis-doctorales/2012/mirm/cualitativo_cuantitativo_mixto.html

Reynoso, C. B. (2004). Introducción a la Arquitectura de Software. *Universidad de Buenos Aires*, 33.

Sanfeliciano, A. (2018). *Diseños de investigación: enfoque cualitativo y cuantitativo*. Recuperado de: <https://lamenteesmaravillosa.com/disenos-de-investigacion-enfoque-cualitativo-y-cuantitativo/>

ScaleGrid. (04 de Marzo de 2019). *2019 Database Trends – SQL vs. NoSQL, Top Databases, Single vs. Multiple Database Use*. Obtenido de <https://scalegrid.io/blog/2019-database-trends-sql-vs-nosql-top-databases-single-vs-multiple-database-use/>

Seminario, J. F. (22 de Marzo de 2011). *Fuentes de información para la investigación*. Obtenido de Fuentes de información para la investigación: <http://investigacionyacademia.blogspot.com/2011/03/fuentes-de-informacion-para-la.html>

Tejedo Sanz, J. (2013). *Gestión de Recursos Humanos*. Madrid, Spain: Macmillan Iberia, S.A. Recuperado de <https://elibro.net/es/ereader/bibliouia/42960?page=17>.

Villagómez, C. (2017). *Modelo en Cascada*. Obtenido de <https://es.ccm.net/contents/223-ciclo-de-vida-del-software>

APÉNDICES

Apéndice A. Guía para entrevista en Tienda Karol

La presente entrevista tiene como objetivo conocer el proceso actual para el control del recurso humano en Tienda Karol, además de los problemas actuales en este proceso y los requerimientos para el prototipo web según las necesidades de Tienda Karol. Esta entrevista cuenta con 10 preguntas abiertas. Las cuales se muestran a continuación.

1. Actualmente, ¿cuál es el proceso para la gestión del recurso humano en Tienda Karol?
2. ¿Cuáles son los inconvenientes o problemas principales con el proceso actual?
3. ¿Alguna vez se ha informado o ha estado en contacto con algún tipo de sistema informático?
4. ¿Cree que un sistema informático puede solventar la problemática actual en el proceso de gestión del recurso humano?
¿Actualmente, ¿cuentan con equipo de cómputo en la compañía?
5. ¿Estaría dispuesto a invertir en la implementación de este tipo de herramientas?
6. ¿Qué beneficios esperaría con la implementación del prototipo para la gestión del recurso humano?
7. Actualmente, ¿cuentan con servicio de internet en Tienda Karol?
8. En síntesis, ¿cuáles son las características y requerimientos que debe contemplar el sistema para la solución de la problemática actual?
9. ¿Estaría dispuesto a cambiar sus procesos manuales actuales, por un sistema web el cual ejecute dichos procesos de manera automatizada?

Apéndice B. Encuesta dirigida a los empleados de Tienda Karol

Guía para encuesta colaboradores

Universidad Internacional de las Américas - Encuesta para funcionarios de la empresa Tienda Karol

Esta encuesta se aplica con el fin de conocer sus opiniones con respecto a la implementación de un sistema y de qué manera piensan que les beneficiaría.

***Obligatorio**

¿En su rol de colaborador, cuánto cree usted que le beneficiaría la implementación de un sistema informático en Tienda Karol? *

- Muy Poco
- Poco
- Regular
- Alto

2- ¿Del 1 al 5, donde 1 significa que no tiene experiencia y 5 significa muy experimentado, que tanta experiencia tiene en el uso de herramientas de software y computadoras? *

- | | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

¿Qué tan eficiente cree usted que sea proceso actual de la gestión del recurso humano? *

- Muy Poco
- Poco
- Regular
- Mucho

¿Según la problemática actual, cual situación de las siguientes opciones es más común? *

- Atrasos en pagos
- Confusión en pagos
- Falta de precisión en los reportes
- Atrasos en la atención al cliente

¿Qué tanto cree que beneficiaría la implementación de un sistema informático para el proceso de gestión del recurso humano? *

- Muy Poco
- Poco
- Regular
- Mucho

Recientemente Tienda Karol adquirió equipo tecnológico, ¿del 1 al 5 que tanto cree usted que dicho equipo sea apto para un sistema de gestión del recurso humano?, siendo 1 muy poco apto y 5 muy apto. *

- 1 2 3 4 5
-

¿Qué tan de acuerdo está usted en que el proceso de gestión del recurso humano en Tienda Karol se realice de forma más automatizada por medio de un software? *

- Muy Poco
- Poco
- Regular
- Mucho

¿Qué tan importante cree que sea la implementación de la tecnología en Tienda Karol? *

- Muy Poco
- Poco
- Regular
- Mucho

