**UNIVERSIDAD TÉCNICA NACIONAL**

**SEDE SAN CARLOS**

**CARRERA**

**Ingeniería del Software**

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

**ECOMASTER**

**ELABORADO POR**

Kevin Arias Arce

Roque Chacón Corrales

2018.

**Contenido**

[Índice de figuras 2](#_Toc507054461)

[Índice de Tablas 3](#_Toc507054462)

[Descripción del problema 4](#_Toc507054463)

[Justificación del problema. 5](#_Toc507054464)

[Objetivo general. 6](#_Toc507054465)

[Objetivos específicos. 7](#_Toc507054466)

[Herramientas y técnicas que se utilizan 7](#_Toc507054467)

[GIT: 9](#_Toc507054468)

[Requerimientos funcionales, no funcionales y del sistema. 11](#_Toc507054469)

[Solución del problema. 26](#_Toc507054470)

[UML: diagrama de casos de uso, diagrama de clases, diagrama de base de datos, diagrama de la arquitectura. 26](#_Toc507054471)

[Criterios de aceptación (deben utilizar la plantilla que se adjunta). 26](#_Toc507054472)

[Resultados de las pruebas (describir los tipos de pruebas que utilizaron y los resultados de las mismas). 28](#_Toc507054473)

[Conclusiones. 28](#_Toc507054474)

[Recomendaciones. 28](#_Toc507054475)

[Anexos (documentación otorgada por el cliente y demás aspectos que propios del proyecto que el equipo de trabajo considere necesario). 28](#_Toc507054476)

[\*Plan de contingencia y/o plan de implementación. 28](#_Toc507054477)

[Anexos 28](#_Toc507054478)

[Información del cliente 28](#_Toc507054479)

# **Índice de figuras**

Figura 1: Logo de PHP…………………………………………………………………….8

Figura 2: Logo de Laravel…………………………………………………………………9

Figura 3: Logo Git…………………………………..……………………………………..10

Figura 4: Logo MySQL…………………………………………..………………………..10

Figura 5: Estructura del patrón de diseño MVC…………………………………...………11

Índice de Tablas

[Tabla 1: Requisito Funcional-001 – Agregar Usuario 12](#_Toc490500185)

[Tabla 2: Requisito Funcional-002 – Editar Usuario](#_Toc490500185) 13

[Tabla 3: Requisito Funcional-003 – Eliminar Usuario](#_Toc490500186) 14

[Tabla 4: Requisito Funcional-004 – Ver Usuarios](#_Toc490500187) 15

[Tabla 5: Requisito Funcional-005 – Agregar Supermercado](#_Toc490500188) 16

[Tabla 6: Requisito Funcional-006 – Editar Supermercado](#_Toc490500189) 17

[Tabla 7: Requisito Funcional-007 – Eliminar Supermercados](#_Toc490500190) 18

[Tabla 8: Requisito Funcional-008 – Ver Supermercados](#_Toc490500191) 19

[Tabla 9: Requisito Funcional-009 – Crear de Horario](#_Toc490500192) 20

[Tabla 10: Requisito Funcional-010 – Editar Horario](#_Toc490500193) 21

[Tabla 11: Requisito Funcional-011 – Eliminar Horarios](#_Toc490500194) 22

[Tabla 12: Requisito Funcional-012 – Ver Horarios](#_Toc490500195) 23

[Tabla 13: Requisito Funcional-013 – Login](#_Toc490500196) 24

Tabla 14: Requisito Funcional-014 – Reportes…………………………………………….25

**Descripción del problema.**

La cadena de supermercados EconoMás tiene varias sedes ubicadas en la provincia de Alajuela, distribuidas entre los cantones de Pital, Viento Fresco y dos más ubicadas en Aguas Zarcas, en una de estas dos últimas sedes la compañía cuenta con la oficina de administración, desde ese punto, la empresa hace la toma de decisiones.

Entre el personal administrativo, está la Supervisora y la encargada de Gestión de talento humano, esos dos puestos, están a cargo de la creación y administración de horarios, para todo el personal de la cadena, esto se refiere a que cada empleado sea asignado en un tiempo específico a un rol dentro de una de las sedes de la empresa.

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

# **Justificación del problema.**

El déficit de poder crear horarios más rápido y de una manera más sencilla y cómoda ha perjudicado en tiempo de toma de decisiones a la empresa EcónoMas, tanto que en momentos se ha tenido problemas en choques de horario de un determinado empleado, lo que genera una ineficiencia de los trabajadores.

El problema es la lentitud por la complejidad que lleva asignar un horario a un empleado, problema se agrava con el tiempo de espera que requiere la distribución de estos horarios a todo el personal ya que se hace uno a uno vía mensaje de texto, muchas veces pueden tardar hasta el sábado para ser enviado el horario a los empleados de la siguiente semana, esto perjudica a empleados y también a la parte administrativa.

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

# **Objetivo general.**

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

# **Objetivos específicos.**

1. Determinar cuáles son los factores, que se toman en cuenta para la creación de un horario.
2. Diseñar una interfaz amigable, para brindarle al usuario una usabilidad sencilla.
3. Construir reportes, para la administración de los horarios, para toma de decisiones de la empresa EconoMas.

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

Se usará el patrón MVC, trabajando con este patrón se facilitará el entendimiento del código, ya que separa el mismo en tres capas: MODELOS, VISTAS y CONTROLADORES, esto favorece la calidad de la aplicación y permitiría la escalabilidad a futuro.

PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.



Figura 1: Logo de PHP

**PHP es un lenguaje totalmente libre y abierto**. Tiene una curva de aprendizaje muy baja, su sintaxis es simple y cumple estándares básicos de la programación orientada a objetos. No son necesarios complejos entornos de desarrollo, que incluso necesitan su propio periodo de aprendizaje. Puedes programar en PHP sin más ayuda que el bloc de notas, todos los IDEs disponibles son gratuitos y los entornos de desarrollo son de rápida y fácil configuración.

LARAVEL

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

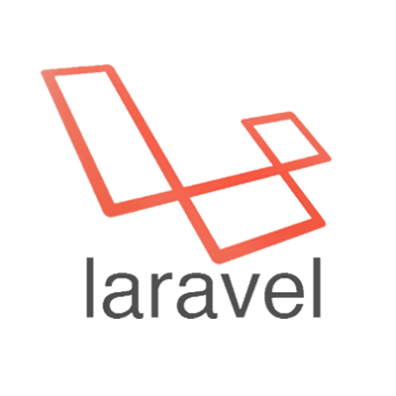


Figura 2: Logo de Laravel

Se utiliza un lenguaje de programación web muy conocido como los es PHP, y en este caso será con el framework LARAVEL ya que se considera que facilitara de una gran manera la creación de esta aplicación web, también permite la reducción de costos y tiempos en el desarrollo y mantenimiento otra razón por cual usar LARAVEL es que la curva de aprendizaje es relativamente baja.

### 

### GIT:

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.



Figura 3: Logo Git

MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL A.B. fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por Oracle Corporation en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.



Figura 4: Logo MySQL

MVC

Modelo-vista-controlador es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. ​ Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.



Figura 5: Estructura del patrón de diseño MVC

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

**Requerimientos funcionales**

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

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

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

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

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

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

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

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

|  |  |
| --- | --- |
| Requisito Funcional-005 | Agregar Supermercado. |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita agregar Supermercados a la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario selecciona la opción de ingreso y se muestra la pantalla correspondiente. | | El administrador del sistema ingresa los datos respectivos al nuevo Supermercado. | | | El sistema verifica que los datos ingresados sean correctos. Si los datos son correctos, el sistema realiza la inserción y muestra un mensaje informativo. Por el contrario, si los datos son incorrectos el sistema le indica al usuario que corrija los datos y realice nuevamente el ingreso de datos. | | | |
| Importancia | Vital |
| Comentarios | Hay diferentes sedes. |

*Tabla 5: Requisito Funcional-005 – Agregar Supermercado*

|  |  |
| --- | --- |
| Requisito Funcional-006 | Editar Supermercado |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita editar Supermercados y actualizarlos en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de editar y se muestra la pantalla correspondiente. | | El administrador del sistema ingresa los datos respectivos al Supermercado. | | | El sistema verifica que los datos ingresados sean correctos. Si los datos son correctos, el sistema realiza la inserción y muestra un mensaje informativo. Por el contrario, si los datos son incorrectos el sistema le indica al usuario que corrija los datos y realice nuevamente el ingreso de datos. | | | |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden editar el punto de venta. |

*Tabla 6: Requisito Funcional-006 – Editar Supermercado*

|  |  |
| --- | --- |
| Requisito Funcional-007 | Eliminar Supermercado |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita eliminar Supermercados y actualizarlos en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de eliminar y se muestra la pantalla correspondiente.  Se le muestra una alerta y si y solo si el administrador la acepta se elimina de la base de datos. | |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden eliminar Supermercados. |

*Tabla 7: Requisito Funcional-007 – Eliminar Supermercados*

|  |  |
| --- | --- |
| Requisito Funcional-008 | Ver lista de Supermercados. |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita ver Supermercados almacenados en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de ver y se muestra la pantalla correspondiente. | |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden ver los Supermercado. |

*Tabla 8: Requisito Funcional-008 – Ver Supermercados*

|  |  |
| --- | --- |
| Requisito Funcional-009 | Crear Horario. |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita la creación de horarios y almacenarlos a la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario selecciona la opción de ingreso a un supermercado y se muestra la pantalla correspondiente. | | El administrador del sistema ingresa los datos respectivos al nuevo horario. Se debe asignar el empleado para ese horario. | | | El sistema verifica que los datos ingresados sean correctos. Si los datos son correctos, el sistema realiza la inserción y muestra un mensaje informativo. Por el contrario, si los datos son incorrectos el sistema le indica al usuario que corrija los datos y realice nuevamente el ingreso de datos. | | | |
| Importancia | Vital |
| Comentarios | El empleado puede tener diferentes horarios en diferentes supermercados, siempre y cuando los horarios no choquen. |

*Tabla 9: Requisito Funcional-009 – Crear de Horario*

|  |  |
| --- | --- |
| Requisito Funcional-010 | Editar Horario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita editar Horarios y actualizarlos en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de editar y se muestra la pantalla correspondiente. | | El administrador del sistema ingresa los datos respectivos al Horario. Se debe asignar el empleado. | | | El sistema verifica que los datos ingresados sean correctos. Si los datos son correctos, el sistema realiza la inserción y muestra un mensaje informativo. Por el contrario, si los datos son incorrectos el sistema le indica al usuario que corrija los datos y realice nuevamente el ingreso de datos. | | | |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden editar los horarios. |

*Tabla 10: Requisito Funcional-010 – Editar Horario*

|  |  |
| --- | --- |
| Requisito Funcional-011 | Eliminar Horario |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita eliminar Horarios y actualizarlos en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de eliminar y se muestra la pantalla correspondiente.  Se le muestra una alerta y si y solo si el administrador la acepta se elimina de la base de datos. | |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden eliminar Horarios. |

*Tabla 11: Requisito Funcional-011 – Eliminar Horarios.*

|  |  |
| --- | --- |
| Requisito Funcional-012 | Ver lista de Horarios |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Descripción | El sistema deberá contar con un módulo que permita ver Horarios almacenados en la Base de datos. |
| Precondición | Login |
| Secuencia normal | |  | | --- | | El usuario(administrador) selecciona la opción de ver y se muestra la pantalla correspondiente, lo podrá ver tanto por supermercados como por empleado.. | |
| Importancia | Vital |
| Comentarios | Solo los administradores pueden ver los horarios. |

*Tabla 12: Requisito Funcional-012 – Ver Horarios*

|  |  |
| --- | --- |
| Requisito Funcional-013 | Login |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Requisitos asociados | Administradores en la base de datos. |
| Descripción | El sistema deberá autentificar las credenciales de los usuarios que apliquen para ingresar a la plataforma. |
| Secuencia normal | |  | | --- | | Se ingresa Usuario y contraseña, el sistema verifica credenciales. | |
| Importancia | Vital |
| Comentarios | Hay diferentes roles. Verificar roles también. |

*Tabla 13: Requisito Funcional-013 – Login*

|  |  |
| --- | --- |
| Requisito Funcional-014 | Generar Reportes |
| Versión | 1.0 |
| Autor | Roque Chacón – Kevin Arce |
| Requisitos asociados | CRUD de horarios. |
| Descripción | El sistema luego de la confirmación de los horarios creados, genera dos reportes, uno el horario de los empleados y seguidamente se podrá visualizar también cantidad de horas trabajadas por empleado por semana, mes. |
| Secuencia normal | |  | | --- | | Al pulsar el botón de reportes el mismo le generara los reportes, sino los hay genera una alerta. | |
| Importancia | Vital |
| Comentarios | Son dos reportes. Horarios-productividad de semana y mes. |

*Tabla 14: Requisito Funcional-014 – Reportes*

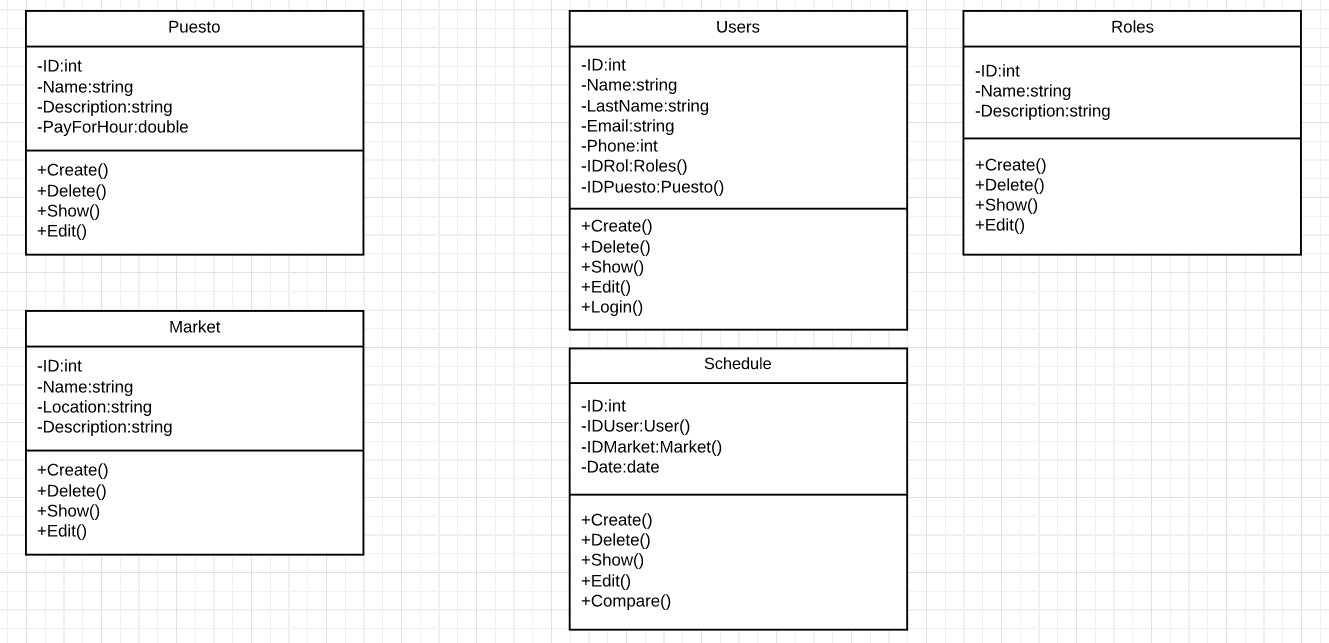
# **Solución del problema.**

# **UML: diagrama de casos de uso, diagrama de clases, diagrama de base de datos, diagrama de la arquitectura.**

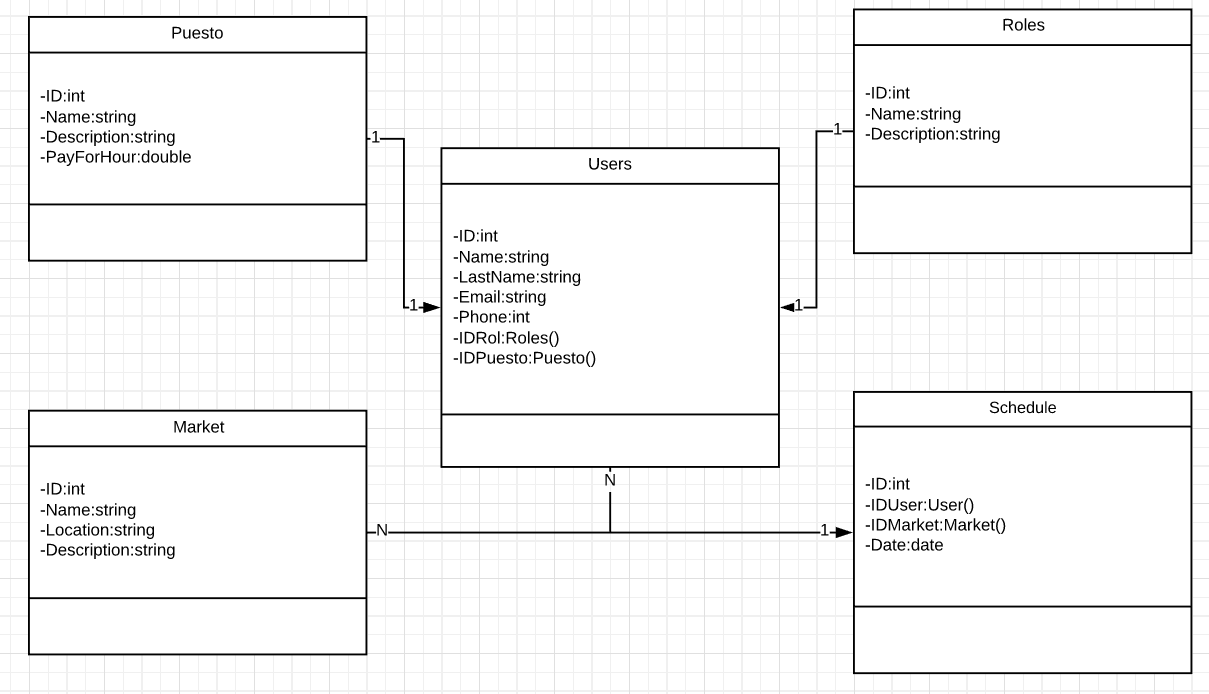
**Diagrama de casos de uso.**

FALTA ESTE.

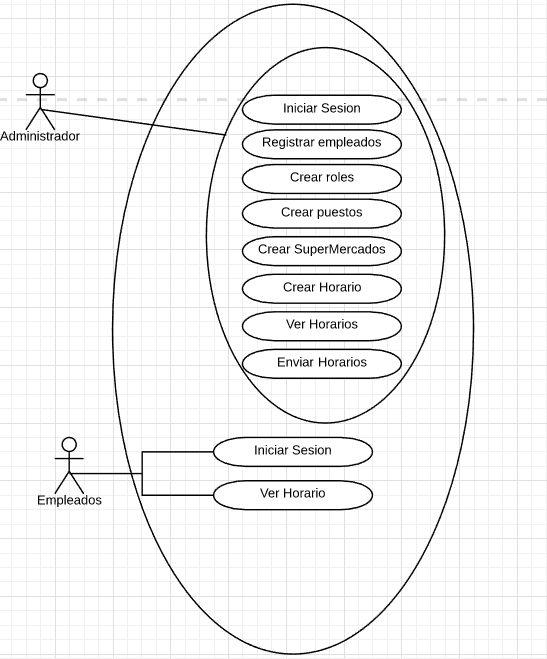
**Diagrama de clases.**



**Diagrama de base de datos.**



**Diagrama de Arquitectura**



# **Criterios de aceptación (deben utilizar la plantilla que se adjunta).**

|  |  |
| --- | --- |
| **Caso de prueba** | |
| Código | CP009 |
| Hardware y software | Computadora, Laptop / SIM2WEB |
| Sistema operativo | Windows 10 |
| Navegador | Google Chrome |
| Resolución | N/A |
| Comentarios | N/A |
| Nombre | Pruebas sobre el menú, Botón de servicios, Botón de productos de servicio. |
| Descripción | Verificar que el Botón redirige correctamente la vista principal de productos de servicios. |
| Encargado | Jonathan Vargas Alvarado |
| Precondiciones | * Tener una computadora de escritorio o laptop con acceso a internet y a la red interna de la Municipalidad de San Carlos. * Tener un usuario en el sistema de la Municipalidad de San Carlos |
| Pasos | 1. Ingresar al acceso directo a Sim2Web. 2. Hacer clic en el Botón de flotillas en el menú. 3. Luego hacer clic en el Botón de Servicios. 4. Hacer clic en el Botón de productos de servicios |
| Resultados esperados | **Pass:** Se debe mostrar el apartado principal de productos de servicios.    **Fail:** El Botón de productos de servicios no muestra ni redirige a nada. |
| Prioridad | Alta |
| Notas | N/A |

# **Resultados de las pruebas (describir los tipos de pruebas que utilizaron y los resultados de las mismas).**

# **Conclusiones.**

# **Recomendaciones.**

# **Anexos (documentación otorgada por el cliente y demás aspectos que propios del proyecto que el equipo de trabajo considere necesario).**

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

# **Anexos**

# **Información del cliente**