

Universidad Nacional del Nordeste



Facultad de Ciencias Exactas y Naturales y Agrimensura

Cátedra: Ingeniería del Software II

Año: 2017

Trabajo de Campo

Gestión de Heladería (GH-System)

Downar, Alexandre L.U. N°:49939

Gallardo, Juan Pablo L.U. N°:49250

González, Víctor Damián L.U. N°:49289

Monzón, Andrea Romina L.U. N°:49368

Profesora: Lic. Ferraro, María de los Ángeles

Índice de contenidos………………………………………………………………………...1

Capitulo1: Introducción3

1.1 Breve estado del arte3

1.2 Objetivos3

1.3 Fundamentación3

Capitulo 2: Metodología 4

2.1Ciclo de vida4

2.2 Especificación de Requerimientos6

2.2.1 Introducción6

2.2.2 Descripción general6

2.2.3 Requisitos Específicos7

2.3 Planificación y calendarización10

2.4 Plan de Riesgo13

2.5 Arquitectura 15

2.6 Diagramas de diseño y análisis 16

Capitulo 3: Herramientas Utilizadas 26

3.1 Herramientas y/o lenguajes de programación 26

Capitulo 4: Resultados27

4.1 Bibliografías consultadas27

**Índice de Figuras**

[Figura 1 Ciclo de vida 4](#_Toc454279713)

[Figura 2 Projecrt libre (planificacion de tarea) 11](#_Toc454279715)

[Figura 3 Project libre (diagrama de Gantt) 1](#_Toc454279716)2

[Figura 4 Arquitectura Cliente-Servidor 1](#_Toc454279717)4

[Figura 5 Caso de uso Sistema GH-System 1](#_Toc454279718)4

[Figura 6 Diagrama de secuencias autentificar clientes 15](#_Toc454279719)

[Figura 7 Modelos de datos GH System 15](#_Toc454279720)

[Figura 8 Diagrama de secuencias autentificar usuario 16](#_Toc454279721)

[Figura 9 Diagrama de Secuencias Registrar usuarios 19](#_Toc454279722)

[Figura 10 Diagrama de Secuencias editar usuario 22](#_Toc454279723)

[Figura 11 Diagrama de Secuencias eliminar usuario 2](#_Toc454279723)4

[Figura 12 Diagrama de Secuencias commprar producto 26](#_Toc454279723)

[Figura 13 Diagrama de Secuencias registrar producto 2](#_Toc454279723)9

**Índice de tablas**

[Tabla 1 Caracteristicas de Usuario (1) 6](#_Toc454197375)

[Tabla 2 Caracteristicas de Usuario (2) 6](#_Toc454197376)

[Tabla 3 Caracteristicas de Usuario (3) 6](#_Toc454197377)

[Tabla 4 Requisitos funcionales 8](#_Toc454197378)

[Tabla 5 Requerimientos no funcionales 9](#_Toc454197379)

[Tabla 6 Planificación de proyecto 1](#_Toc454197380)1

[Tabla 7 analisis de riesgo 1](#_Toc454197381)3

[Tabla 8 Conversación : Autentificar Clientes 17](#_Toc454197389)

[Tabla 9 Plan de prueba: auntentificar usuario 18](#_Toc454197389)

[Tabla 10 Conversación : Registrar Usuario 20](#_Toc454197389)

[Tabla 11 Plan de prueba : Registrar Usuario 2](#_Toc454197390)1

[Tabla 12 Conversación : Editar usuario 2](#_Toc454197391)3

[Tabla 13 Plan de prueba : editar usuario 2](#_Toc454197392)4

[Tabla 14 Conversacion: Eliminar usuario 2](#_Toc454197392)5

[Tabla 15 Plan de prueba : Eliminar usuario 2](#_Toc454197392)6

[Tabla 16 Conversacion: comprar producto 2](#_Toc454197392)7

[Tabla 17 Plan de prueba : comprar producto 2](#_Toc454197392)9

[Tabla 18 Conversacion: Registrar producto 30](#_Toc454197392)

[Tabla 19 Plan de prueba : Registrar producto](#_Toc454197392) 31

**Capítulo1:Introducción**

**1.1 Breve estado del arte**

Hoy en día se vive en un mundo actualizado, cambiante y competitivo, en el cual es fundamental que una empresa se esté innovando constantemente, lo que ha originado el desarrollo de aplicaciones para las empresas, para cumplir con las necesidades específicas del negocio.

Sin embargo, la empresa “Delicias” aún sigue usando métodos de gestión de forma manual, ya sea tanto como para la compra, venta, pedidos y demás gestiones.

## 1.2 Objetivos

Se establece como objetivo la realización de un Sistema Web de Gestión de Productos que permitirá controlar con más precisión y claridad la compra-venta e inventario de dichos productos, como así también la gestión de empleados a cargo.

## 1.3 Fundamentación

Tras analizar la forma en que la empresa “Delicias” gestiona la compra-venta e inventario de dichos productos, se ha detectado un número de falencias entre las cuales podemos mencionar: la forma en que se registran las compras-ventas, la eficiencia en la realización de los cálculos, el seguimiento del stock.

Por estas razones existe una creciente demanda de sistemas que clarifiquen, organicen y controlen las compras-ventas de la empresa.

En consecuencia a lo analizado, se plantea una solución en Sistemas de Software para cubrir los aspectos ya mencionados, agregándole la gestión de empleados.

**Capítulo 2: Metodología**

**2.1 Ciclo de vida del proyecto**

El ciclo de vida es el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o remplazado (muere).

El modelo elegido para representar el ciclo de vida de nuestro proyecto es el Modelo V porque es el modelo que mejor se adapta a nuestro proyecto ya que sirve para proyectos pequeños, aumenta la comunicación interna entre integrantes del grupo, nos permite corregir errores durante el desarrollo, teniendo así una mejor aceptación del producto final por parte del cliente.

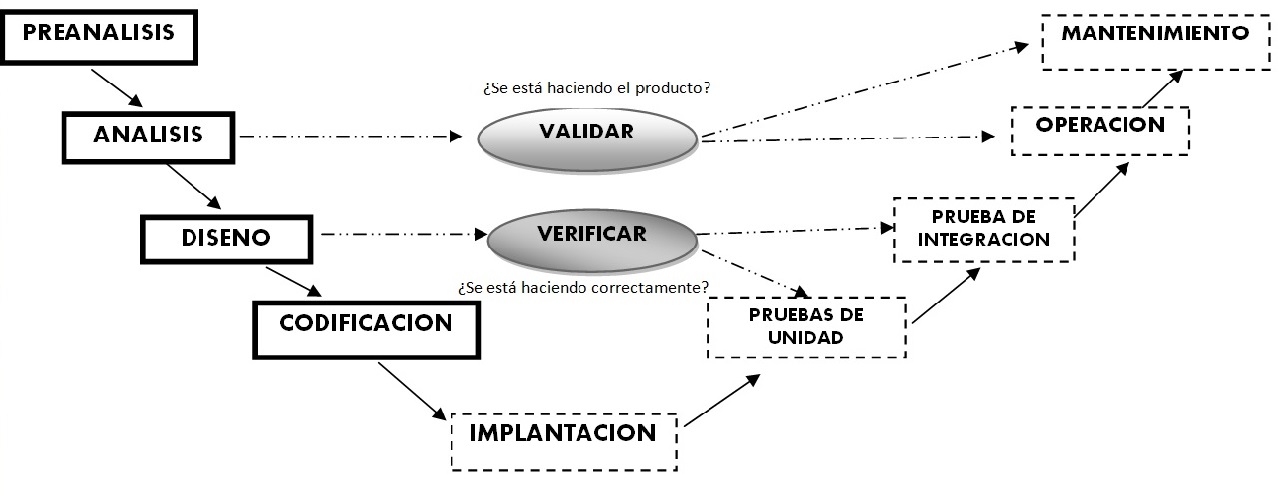


Figura 1. Ciclo de vida

**Pre análisis:**

* Identificar las necesidades del cliente.
* Realizar un análisis técnico y económico del sistema.
* Establecer restricciones de costo y tiempo.
* Evaluar la viabilidad del sistema.

**Análisis:**

Define los flujos de información, las estructuras primarias de datos, las características funcionales del sistema, los requerimientos de rendimiento y las restricciones impuestas por el cliente. Asimismo, se incorporarán los criterios globales de validación que se utilizarán para probar que los requisitos señalados han sido implementados.

**Diseño:**

Proceso de aplicar distintas técnicas y principios con el propósito de definir un dispositivo, proceso o sistema con los suficientes detalles para permitir su realización física.

* Diseño de datos.
* Diseño Arquitectónico.
* Diseño interfaz.
* Diseño procedimental.

**Codificación:**

Codificación del software en el lenguaje determinado.

**Implementación:**

Instalar el sistema para que empiece a trabajar y capacitación de los usuarios del sistema para que puedan utilizarlo.

**Pruebas de Unidad:**

Comprobar el correcto funcionamiento de una unidad de código.

**Pruebas de Integración:**

 Comprobar que todos los elementos unitarios que componen el software, funcionan juntos correctamente probándolos en grupo.

**Operación:**

Prueba del software en las actividades cotidianas de la empresa.

**Mantenimiento:**

El software producido en la fase de desarrollo deber ser mantenido, ya que sufrirá cambios después de que se entregue al cliente. Los cambios pueden ser debido a:

* Errores encontrados.
* Cambios en el entorno externo que el software debe adaptarse.
* Ampliaciones de funcionalidades requeridas por el cliente.

**2.2 Especificación de requerimientos de Software ERS**

## 2.2.1 Introducción

GH-System será un sistema diseñado para trabajar en empresas que se encuentren en el ámbito de los helados, específicamente para la heladería “El Polo”, lo que permitirá su utilización de forma rápida y eficaz por parte del personal de la organización, así como también de los clientes.

El sistema es independiente de otros softwares siendo capaz de interactuar correctamente con sistemas operativos Windows y LINUX.

**2.2.2 Descripción General**

**Características de los usuarios**

|  |  |
| --- | --- |
| **Tipo de usuario** | Administrador |
| **Formación** | Licenciado en Sistemas de Información |
| **Actividades** | Control y manejo del sistema en general |

Tabla 1 Características de usuario (1)

|  |  |
| --- | --- |
| **Tipo de usuario** | Empleado |
| **Formación** | **Conocimientos básicos de computación** |
| **Actividades** | Venta de producto y consulta de stock |

Tabla 2 Características de usuario (2)

|  |  |
| --- | --- |
| **Tipo de usuario** | Cliente |
| **Formación** | **-** |
| **Actividades** | Compra de productos |

Tabla 3 Características de usuario (3)

## Restricciones

* Interfaz amigable con el usuario.
* Lenguajes y tecnologías en uso: PHP y MySQL.
* Los servidores deben ser capaces de atender consultas concurrentemente.
* El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

## Suposiciones y dependencias

* Se asume que los requisitos aquí descritos son estables.
* Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma

# **2.2.3 Requisitos específicos**

## Requisitos comunes de las interfaces

### Interfaces de usuario

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto y, será visualizada desde un navegador de internet.

### Interfaces de hardware

Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

* Adaptadores de red.
* Procesadorocta-core 3.2GHz o superior.
* Memoria mínima de 8GB.
* Mouse.
* Teclado.

### Interfaces de software

* Sistema Operativo: Windows 8.1 o superior. Linux.
* Explorador: Mozilla, Chrome, Firefox, Opera.

### Interfaces de comunicación

Los servidores, usuarios y aplicaciones se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible. Por ejemplo, para transferir archivos o documentos deberán utilizarse protocolos existentes.

## Requisitos funcionales

|  |  |  |
| --- | --- | --- |
| Identificación | Descripción | Prioridad |
| RF#01 | El sistema debe permitir el registro de un nuevo usuario. | Alto |
| RF#02 | El sistema deberá almacenar la información de los clientes. | Alto |
| RF#03 | El sistema deberá detectar si el pedido es válido para ser realizado. | Alto |
| RF#04 | El sistema deberá proveer ayuda u opciones para rellenar los campos de registración | Medio |
| RF#05 | El sistema deberá permitir el ingreso de los pedidos/compras de forma online. | Medio |
| RF#06 | El sistema deberá permitir visualizar los catálogos de productos. | Alto |
| RF#07 | El sistema deberá permitir dar de baja un producto. | Alto |
| RF#08 | El sistema debe registrar los pagos de mensualidad del personal. | Alto |

Tabla 4 Requisitos funcionales

## Requisitos no funcionales

|  |  |  |  |
| --- | --- | --- | --- |
| Identificación | Descripción | Clasificación | Prioridad |
| RNF#01 | El sistema deberá ser implementado en el lenguaje PHP y MYSQL | Organizacional | Medio |
| RNF#02 | El sistema deberá realizarse según el estándar de norma IEEE-830. | Organizacional | Medio |
| RNF#03 | El diseño del sistema se deberá producir con una herramienta CASE en particular, EnterpriseArchitect. | Organizacional | Medio |
| RNF#04 | La documentación del sistema completo se entregará en el término de un mes. | Organizacional | Alto |
| RNF#05 | Se presentará un primer prototipo en 6 meses. | Producto | Medio |
| RNF#06 | El sistema completo deberá estar listo en no máximo 12 meses. | Producto | Medio |
| RNF#07 | El tiempo de reinicio después de un fallo deberá ser de no máximo 30 segundos. | Producto | Alto |
| RNF#08 | El tiempo de respuestas deberá ser de no máximo 1 segundo. | Producto | Alto |

Tabla 5 Requerimientos no Funcionales

### **Requisitos de rendimiento**

* Garantizar que el diseño de las consultas u otro proceso no afecte el desempeño de la base de datos, ni considerablemente el tráfico de la red.

### **Seguridad**

* Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
* Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
* Facilidades y controles para permitir el acceso a la información al personal autorizado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.

### **Fiabilidad**

* El sistema debe tener una interfaz de uso intuitivo y sencillo (amigable).
* La interfaz de usuario debe ajustarse a las características de la web de la heladería, dentro de la cual estará incorporado el sistema de gestión GH-System).

### **Disponibilidad**

* La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes, contar con una contingencia, generación de alarmas.

### **Mantenibilidad**

* El sistema debe disponer de una documentación fácilmente actualizable que permita realizar operaciones de mantenimiento con el menor esfuerzo posible

### **Portabilidad**

* El sistema será implantado bajo la plataforma de Windows y Linux.

**2.3 Planificación y calendarización**

En este apartado del documento de proyecto de software gestionaremos las estimaciones razonables de recursos, costos y planificación temporal para el desarrollo del software que estará a disposición de las empresas aseguradoras.

**Diagrama de Gantt**

Utilizaremos el siguiente diagrama para exponer el tiempo de dedicación previsto para las diferentes tareas que serán realizadas a lo largo del proyecto.

|  |  |  |
| --- | --- | --- |
| Actividades | Predecesoras |  |
| -Pre-análisis |  |  |
| A) Selección de método de educción. | - | 1 |
| B) Selección de preguntas de entrevista. | A | 1 |
| C) Programar entrevista. | A , B | 7 |
| D) Decodificar entrevista. | C | 2 |
| -Análisis |  |  |
| E) Confección de la ERS. | D | 20 |
| F) Análisis de riesgo. | D | 7 |
| G) Entrega de documentación. | E , F | 1 |
| H) Diseño (etapa prototipo). | E | 50 |
| I) Desarrollo (etapa prototipo). | E | 140 |
| J) Entrega del prototipo. | H , I | 1 |

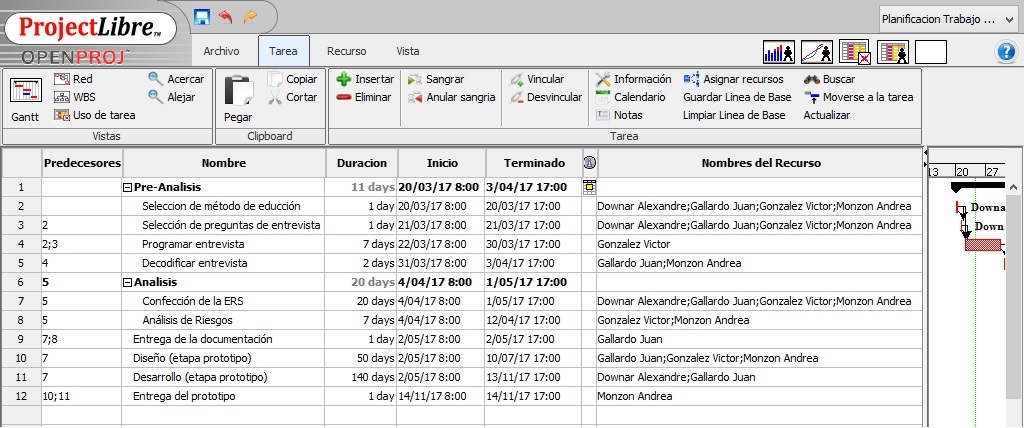
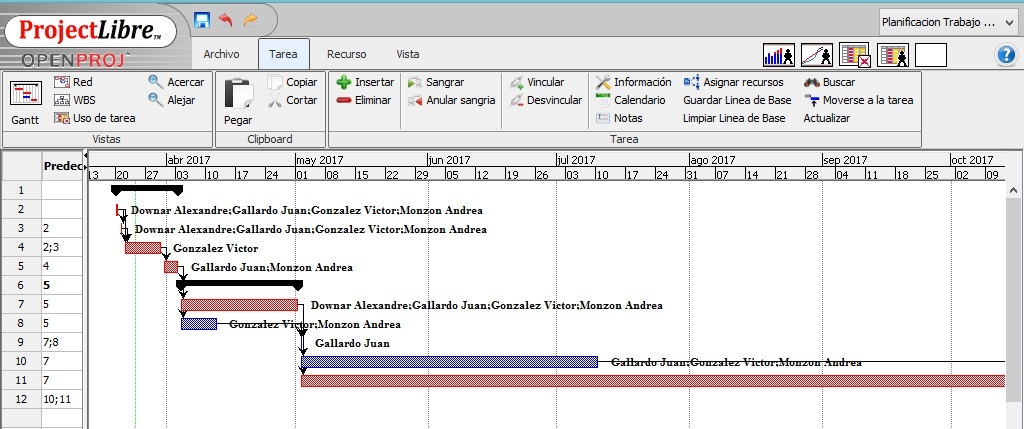
Tabla 6 Planificación del proyecto

Figura 3Project libre (diagrama de Gantt)

**2.4 Plan de Riesgos**

Una estrategia inteligente para llevar a cabo un proyecto de manera eficiente, es la de anticiparse a los problemas o inconvenientes que puedan llegar a surgir mucho antes de iniciar la etapa operativa. El equipo de software establece un plan para gestionar el riesgo. El objetivo principal es evitarlo, pero, dado que no todos los riesgos son evitables, el equipo trabaja para desarrollar un plan de contingencia que le permitirá responder en forma controlada y efectiva.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgos** | **Tipo** | **Acción para evitar** | **Nivel Impacto** | **Nivel Prob.** |
| Cambios de requerimientos del Cliente | Producto | Desarrollar un análisis eficaz de requerimiento. | Tolerable | Moderado |
| Enfermedad del personal | Organización | Reorganización del personal. | Tolerable | Moderado |
| Renuncia de un integrante | Organización | Ofrecer buenas condiciones de contratación y flexibilidad a la hora de realizar la tarea. | Serio | Bajo |
| Dificultad de un integrante en un módulo determinado | Producto | Rotación del personal, todos están capacitados para hacer todo. | Insignificante | Alto |
| Resistencia al cambio de los usuarios finales | Negocio | Cursos de capacitación meses antes de la implementación final. | Tolerable | Alto |
| Se ha subestimado la fecha limite | Producto | Aumento de jornada, incorporación de personal capacitado. | Serio | Moderado |
| Pérdida de fondos | Organización | Posesión de una cuenta de respaldo | Catastrófico | Muy bajo |
| Pérdida de información por robo y/o daño de equipos | Producto | Realizar un respaldo de seguridad al término de cada jornada. | Catastrófico | Muy bajo |
| Falta de control de calidad | Producto | Pruebas y controles exhaustivos y periódicos, Testing. | Serio | Moderado |
| Retraso en alguna fase del proyecto | Organización | Realizar una buena planificación, asignar un tiempo adecuado a cada fase. | Serio | Moderado |

Tabla 7 Análisis de riesgo

**2.5 Arquitectura**

El tipo de arquitectura que nos pareció acorde al sistema que implementamos fue el de cliente-servidor, debido a que es una arquitectura de procesamientos cooperativos en donde los clientes de la heladería, en este caso, pueden acceder a los datos a través de unsitio web de la empresa de forma transparente para saber acerca de los productos que puede adquirir, así como también el personal para realizar los pedidos y actualización de stock.

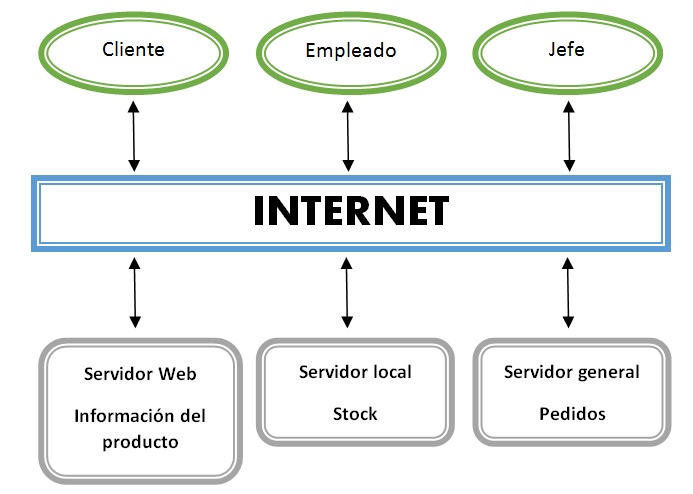


Figura 4 Arquitectura Cliente-Servidor

**2.6 Diagramas de Diseño y Análisis**

**Diagrama de caso de uso**

Figura 5. Caso de Uso Sistema “GH-System”

**Diagrama de clase**

****

Figura 6. Diagrama de clases “GH-System”

**Modelo de Datos**

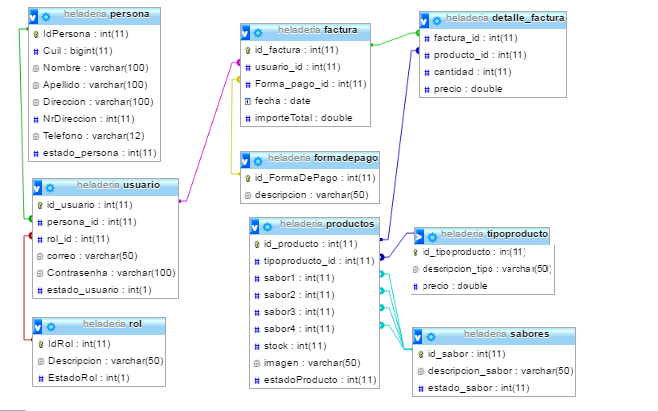
****

Figura 7. Modelo de datos “GH-System”

**Diagramas de secuencia**

****

Figura 8. Autentificar Usuario, curso normal

**Conversaciones y Contrato de Operaciones Críticas**

**Conversación - Caso de uso: autentificar usuario.**

**A: Cliente S: Sistema**

|  |  |  |
| --- | --- | --- |
| **Acción** | **Curso Normal** | **Curso Alternativo** |
| 1. 1. A: Inserta correo y contraseña. | - | - |
| 1. 2. S: Verifica consistencia datos.   S: expCorreo(correo).  expPass(pass). | 3.1 Los datos son correctos.  3.2. S: Iniciar sesión (correo,pass).  3.3. login(correo,pass) | 3.1.1. Los datos son inválidos.  3.1.2. Mensaje: “Nombre de Usuario o Contraseña Incorrectos”.  3.1.3. Volver al paso 1. |
| -4. S: Redirecciona a una página dependiendo del usuario. |  |  |
| 1. 5. Fin Caso de Uso. | - | - |

Tabla 8. Conversación: Autentificar Usuario

**Contrato de Operaciones**

* Nombre: IniciarSesion(correo,pass).
* *Responsabilidades*: El sistema deberá verifica que el correo y contraseña ingresado son correctos.
* *Referencias*: Caso de uso autentificar usuario.
* *Excepciones*: El sistema informa que los datos ingresados son incorrectos cancelando la operación.
* *Pre-condición*: sesión anterior terminada.
* *Post-condición*: El usuario inicia sesión en el sistema y se lo redirecciona a una pagina dependiendo del usuario.

**Plan de prueba- Caso de uso: autentificar usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CP** | **Objetivo** | **Datos de Entrada** | **Resultados Esperados** | **Resultados Obtenidos** |
| 1 | Autentificar Usuario | * Correo electrónico: damian3776@gmail.com (correcto) * Password: 12345678   (correcto) | -Que los datos sean válidos.  -Redireccione a una página dependiendo del usuario. | Ok |
| 2 | Autentificar Usuario | * Correo electrónico: damian3777@gmail.com (incorrecto) * Password: 12345678   (correcto) | -Que los datos sean inválidos.  -Mensaje: “Nombre de Usuario o Contraseña Incorrectos”. | Ok |
| 3 | Autentificar Usuario | * Correo electrónico: damian3776@gmail.com (correcto) * Password: 123(incorrecto) | -Mensaje: “Debe ingresar un mínimo de 8 caracteres”. | Ok |
| 4 | Autentificar Usuario | * Correo electrónico: damian3776(incorrecto) * Password: 12345678 (correcto) | -Mensaje: “Debe ingresar un correo electrónico válido”. | Ok |

Tabla 9. Plan de prueba: Autentificar Usuario

**Diagrama de secuencia**

****

Figura 9. Registrar usuario, curso normal

**Conversación - Caso de uso: Registrar usuario.**

**Actores: Administrador S: Sistema**

|  |  |  |
| --- | --- | --- |
| **Acción** | **Curso Normal** | **Curso Alternativo** |
| 1. S. muestra las opciones de registro de usuario. |  |  |
| 2.A: ingresa correo, apellido, nombre, cuil, contraseña. | - | - |
| 1. 3. S: Verifica formato datos. 2. expCorreo(correo), 3. expLetras(apellido), 4. expLetras(nombre), 5. expCuil(cuil), 6. expPass(pass) | 3.1.1 Los datos son válidos.  3.1.2 Insertar usuario.  Insertar(cuil,nombre,apellido,  Rol,correo,contraseña).  Acceso(Administrador,  alta\_usuario). | 3.2.1 Los datos son inválidos.  3.2.2 S. Mensaje: “existen inconsistencias en los datos ingresados”.  3.2.3 Volver al paso 2.  3.3.1. Los datos ingresados existen en la base de datos.  3.3.2 S. muestra mensaje:” los datos ingresados corresponden a un usuario registrado”.  3.3.3 Volver al paso 2. |
| 1. 4. S. Mensaje “Registro hecho con éxito”. |  |  |
| 1. 5. Fin de Caso de Uso. | - | - |

Tabla 10. Conversación: Registrar Usuario

**Contrato de Operaciones**

* Nombre: Insertar(cuil,nombre,apellido,rol,correo,contraseña).
* *Responsabilidades*: El sistema deberá verificar que los datos ingresados para la registración sean correctos y que el usuario no esté registrado.
* *Referencias*: Caso de uso registrar usuario.
* *Excepciones*: no aplica.
* *Pre-condición*: El sistema permitirá el ingreso de los datos.
* *Post-condición*: El sistema registra usuario.

**Plan de prueba - Caso de uso: Registrar usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CP** | **Objetivo** | **Datos de Entrada** | **Resultados Esperados** | **Resultados Obtenidos** |
| 1 | Registrar Usuario | * Correo electrónico: damian3776@gmail.com (correcto) * Apellido: Gonzalez (correcto) * Nombre: Damian (correcto) * Cuil: 20364694378 (correcto) * Password: 12345678   (correcto) | -Que los datos sean válidos.  -Insertar usuario.  -Mensaje: “Registro hecho con éxito”. | Ok |
| 2 | Registrar Usuario | * Correo electrónico: damian3777 (incorrecto) * Apellido: Gonzalez (correcto) * Nombre: Damian (correcto) * Cuil: 20364694378 (correcto) * Password: 12345678   (correcto) | -Que los datos sean inválidos. -Mensaje: “existen inconsistencias en los datos ingresados”. | Ok |
| 3 | Registrar Usuario | * Correo electrónico: damian3776@gmail.com(correcto, pero ya existe) * Apellido: Gonzalez (correcto) * Nombre: Damian (correcto) * Cuil: 20364694378 (correcto) * Password: 12345678   (correcto) | -Que los datos sean inválidos. -Mensaje: “los datos ingresados corresponden a un usuario registrado”. | Ok |
| 4 | Registrar Usuario | * Correo electrónico: damian3776@gmail.com (correcto) * Apellido: Gonzalez (correcto) * Nombre: Damian (correcto) * Cuil: 36469437 (incorrecto) * Password: 12345678   (correcto) | -Que los datos sean inválidos. -Mensaje: “debe ingresar un numero de Cuil válido”. | Ok |

Tabla 11. Plan de prueba: Registrar Usuario

**Diagrama de secuencia**

Figura 10. Editar usuario, curso normal

**Conversación - Caso de uso: Editar usuario**

**Actores: Administrador S: Sistema**

|  |  |  |
| --- | --- | --- |
| **Acción** | **Curso Normal** | **Curso Alternativo** |
| 1. S: muestra las opciones de usuario. |  |  |
| 2. A: selecciona usuario (id). | - | - |
| 3. S: consultausuario.  getUsuario(id) | 3.1.1. S: Devuelve datos del usuario. | 3.2.1 el usuario ingresado no existe.  3.2.2 S. Mensaje: “el usuario ingresado no existe”.  3.2.3. Volver al paso 2. |
| 4.A: Modifica datos. | 4.1.1. S: Valida estructura de datos.  Validar datos().  4.1.2. S: Editar usuario.  Editar(id,persona,cuil,nom-  bre,apellido,rol,correo).  Acceso(administrador, alta\_usuario). |  |
| 5. S. Mensaje “Cambio realizado con éxito”. |  |  |
| 6. Fin Caso de Uso. | - | - |

Tabla tabla 12. Conversación: Editar usuario

**Contrato de Operaciones**

* Nombre: Editar(id,persona,cuil,nombre,apellido,rol,correo).
* *Responsabilidades*: El sistema deberá editar los datos cargados a cada usuario.
* *Referencias*: Caso de uso Editar usuario.
* *Excepciones*: el usuario ingresado no exista.
* *Pre-condición*: consultar si existe el usuario a modificar.
* *Post-condición*: El sistema editara los datos del usuario.

**Plan de prueba - Caso de uso: Editar usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CP** | **Objetivo** | **Datos de Entrada** | **Resultados Esperados** | **Resultados Obtenidos** |
| 1 | Editar Usuario | * Id\_usuario: 22 (correcto) * Nombre: Damian (decía “Damin”) | -Que el id sea válido.  -Usuario existente.  -Muestra datos del usuario  -Edite usuario.  -Mensaje: “Cambio realizado con éxito”. | Ok |
| 2 | Editar Usuario | * Id\_usuario: 222 (incorrecto) | -Que el id sea inválido.  -Usuario inexistente.  -Mensaje: “El usuario ingresado no existe”. | Ok |
| 3 | Editar Usuario | * Id\_usuario: 22a(incorrecto) | -Que el id sea inválido.  -Mensaje: “Debe ingresar solo caracteres numericos”. | Ok |

Tabla 13.Plan de prueba: Editar Usuario

**Diagrama de secuencia**

****

Figura 11. Eliminar usuario, curso normal

**Conversación - Caso de uso: Eliminar usuario**

**Actores: Administrador S: Sistema**

|  |  |  |
| --- | --- | --- |
| **Acción** | **Curso Normal** | **Curso Alternativo** |
| 1.S: muestra las opciones de usuario. |  |  |
| 2. A: selecciona Usuario. | - | - |
| 3. S: Verifica datos. | 3.1.1 Los datos son válidos.  3.1.2 S: eliminarLogico(id,  Estado). | 3.2.1 Los datos son inválidos.  3.2.2 S. Mensaje: “existen inconsistencias en los datos ingresados”.  3.2.3 Volver al paso 2. |
| 4: S. Mensaje: “Cambio realizado con éxito”. |  |  |
| 1. 5. Fin Caso de Uso. | - | - |

Tabla 14. Conversación: Eliminar Usuario

**Contrato de Operaciones**

* Nombre: eliminarLogico(id,estado).
* *Responsabilidades*: El sistema deberá poder eliminar un usuario en él.
* *Referencias*: Caso de uso Eliminar usuario.
* *Excepciones*: El sistema informa que los datos no son inválidos.
* *Pre-condición*: que se seleccione el usuario a eliminar.
* *Post-condición*: La eliminación del usuario sea de forma éxitos.

**Plan de prueba - Caso de uso: Eliminar usuario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CP** | **Objetivo** | **Datos de Entrada** | **Resultados Esperados** | **Resultados Obtenidos** |
| 1 | Eliminar Usuario | * Usuario:“Id\_usuario: 22”(correcto) * Click en “Eliminar” | -Que el id sea válido.  -Usuario existente.  -Elimine usuario.  -Mensaje: “Cambio realizado con éxito”. | Ok |
| 2 | Eliminar Usuario | * Usuario: “Id\_usuario: 222” (incorrecto) | -Que el id sea inválido.  -Usuario inexistente.  -Mensaje: “Existe inconsistencias en los datos ingresados”. | Ok |
| 3 | Eliminar Usuario | * Usuario: “Id\_usuario: 22” (correcto, pero ya fue eliminado) * Click en “Eliminar” | -Que el id sea válido.  -Usuario existente.  -Mensaje: “El usuario seleccionado ha sido eliminado con anterioridad”. | Ok |

Tabla 15. Plan de prueba: Eliminar Usuario

**Diagrama de secuencia**

****

Figura 12. Comprar producto, curso normal

**Conversación - Caso de uso: Comprar producto**

**Actores: Cliente S: Sistema**

|  |  |  |
| --- | --- | --- |
| **Acción** | **Curso Normal** | **Curso Alternativo** |
| 1.S: muestra el Catalogo de productos |  |  |
| 2. A: se logea como cliente de la pagina. | 2.1 S: verificación de datos y lo direcciona a la pestaña de catalogo.  login(correo,pass); | 2.1.1 verificación de datos. “cliente NO registrado”.  2.1.2 ir al caso de uso “Registrar usuario” |
| 3: A:selecciona el producto a comprar y la cantidad.  verificarProducto(id\_producto,cantidad) | 3.1. S: carga al carrito los productos seleccionados.  Agregar(id\_producto,cantidad) | 3.1.1 S: los productos seleccionados No están disponibles o stock = 0.  3.2.3 Volver al paso 3. |
| 4: A: realizar compra. | 4.1.S:Muestra los datos de la factura y el total de la compra. |  |
| 5: A: Confirmar compra. | 5.1.S:guardar\_detallefactura(id\_producto,cantidad,precio);  5.2.S:actualizar\_stock( id\_producto,cantidad);  5.3.S:“Su compra se a realizado exitosamente” |  |
| 1. 6. Fin Caso de Uso. | - | - |

Tabla 16. Conversación: Comprar Producto

**Contrato de Operaciones**

* Nombre: Agregar Producto(id\_producto,cantidad)
* *Responsabilidades*: El sistema deberá agregar el producto al carrito de compra del cliente.
* *Referencias*: Caso de uso Comprar Producto.
* *Excepciones*: El sistema informa que el cliente o datos ingresados no se encuentra registrados.
* *Pre-condición*: que se seleccione el/los producto/s a comprar.
* *Post-condición*: factura registrada.

**Plan de prueba- Caso de uso: Comprar producto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CP** | **Objetivo** | **Datos de Entrada** | **Resultados Esperados** | **Resultados Obtenidos** |
| 1 | Login Usuario | * Correo electrónico: damian3776@gmail.com (correcto) * Password: 12345678   (correcto) | -Que los datos sean válidos.  -Redireccione a la página de catalogo. | Ok |
| 2 | Login Usuario | * Correo electrónico: damian3777@gmail.com (incorrecto) * Password: 12345678   (correcto) | -Que los datos sean inválidos.  -Mensaje: “cliente NO registrado”. | Ok |
| 3 | agregar al carrito | * Id\_producto: 10(correcto) * cantidad: 1(correcto) | -que el sistema verifique que el producto tenga stock  -agregue a la lista de productos en carrito | Ok |
| 4 | agregar al carrito | * Id\_producto: 10(correcto) * cantidad: 130(correcto,pero supera el stock disponible) | -que el sistema verifique que el producto no tenga stock  -y muestre un mensaje” el valor debe ser inferior o igual al stock” | Ok |
|  | Realizar compra | -click | -que el sistema muestre toda la información de la factura y el total de la compra | Ok |
| 5 | Confirmar compra | * click | -el sistema guarde la informacion de la factura  -actualice el stock  -muestre un mensaje “Su compra se a realizado exitosamente” | Ok |

Tabla 17. Plan de prueba Comprar producto

**Diagrama de secuencia**

****

Figura 13. Registrar producto, curso normal

**Conversación - Caso de uso: Registrar producto**

**Actores: Administrador S: Sistema**

|  |  |  |
| --- | --- | --- |
| **Acción** | **Curso Normal** | **Curso Alternativo** |
| 1.A:Selecciona tipo de producto,sabor, ingresa stock y carga una imagen |  |  |
| 2.S:verificar producto(id\_tipoproducto,sabor1,sabor2,sabor3,sabor4,imagen) | 2.1.S:cargar producto(id\_tipoproducto,sabor1,sabor2,sabor3,sabor4,imagen)  2.2 S: muestra la lista de producto. | 2.1.1.S:muestra mensaje “campo incompleto u erróneo”  2.1.2S:volver paso 1. |
| 3: S: Fin de caso de uso. |  |  |

Tabla 18. Conversación: Registrar Producto

**Contrato de Operaciones**

* Nombre:

Verificar Producto (id\_tipoproducto, sabor1, sabor2, sabor3, sabor4, imagen)

* *Responsabilidades*: El sistema deberá verificar que todos los campos del producto estén completo.
* *Referencias*: Caso de uso Registrar Producto.
* *Excepciones*: El sistema informa que los datos ingresados no son erróneos o incompletos.
* *Pre-condición*: que el tipo de producto y sabor estén registrados.
* *Post-condición*: producto registrado.

**Plan de prueba- Caso de uso: Registrar producto**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CP** | **Objetivo** | **Datos de Entrada** | **Resultados Esperados** | **Resultados Obtenidos** |
| 1 | Verificar producto | -id\_tipoproducto:1 kilos  - sabor1:frutilla  - sabor2:vainilla  - sabor3:crema del cielo  -sabor4:granizado  -stock:100  - imagen:helado.jpg | - Que los datos ingresados sean correctos.  -carga de producto.  - muestra la lista de producto. | Ok |
| 2 | Verificar producto | -id\_tipoproducto:1 kilos  - sabor1:frutilla  - sabor2:vainilla  - sabor3:crema del cielo  -sabor4:granizado  -stock:cien   * - imagen:helado.jpg | -- Que los datos ingresados sean incorrectos.  - muestra mensaje “campo incompleto u erróneo ” | Ok |

Tabla 19. Plan de prueba: Registrar producto

**Capítulo 3: Herramientas utilizadas**

**3.1 Herramientas y/o lenguajes de programación**

En esta sección presentaremos las herramientas utilizadas para la elaboración del proyecto.

**Herramienta de Oficina: Microsoft Word 2016**

Con el Software “*Microsoft Word 2016*” también conocido como “Word 2016” se realizó la presente documentación incluyendo la documentación anexa del estándar IEEE830 de la Especificación de Requisitos de Software”.

Microsoft Word es un software destinado al procesamiento de textos. Fue creado por la empresa Microsoft, y actualmente viene integrado en la suite informática Microsoft Office.

**Herramienta Case: ProjectLibre Versión 1.6.2**

Con el Software “*ProjectLibre*” se realizó la representación de forma gráfica del tiempo de dedicación previsto para las diferentes actividades por medio de un diagrama de Gantt (figura 2).

ProjectLibre es un [software de administración de proyectos](https://es.wikipedia.org/wiki/Software_de_administraci%C3%B3n_de_proyectos) de [código abierto](https://es.wikipedia.org/wiki/C%C3%B3digo_abierto), similar a [Microsoft Project](https://es.wikipedia.org/wiki/Microsoft_Project). ProjectLibre corre sobre la [plataforma Java](https://es.wikipedia.org/wiki/Plataforma_Java), lo que permite ejecutarlo en varios [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo). Es distribuido para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

**Herramienta Case: EnterpriseArchitect Versión 12.1.1227**

Con el Software “*Enterprise Architect 12.1.1227*” se realizaron los diagramas de casos de uso, diagramas de secuencias y diagramas de clases de la presente documentación.

Enterprise Architect es una plataforma de alto desempeño para el modelado, visualización y diseño, basada en el estándar UML 2.4.1. Ofrece trazabilidad completa desde mapas mentales, pasando por los requerimientos y hasta el diseño y la distribución del software, con el nivel de eficiencia, robustez, herramientas de colaboración y seguridad requeridos para sacar adelante proyectos altamente demandantes y cualquier tamaño.

**Capítulo 4: Resultados**

**Referencias**

* Pressman, Roger S. 2010. “INGENIERÍA DEL SOFTWARE. Un enfoque práctico. 7ta edición”. McGraw-Hill. Madrid. ISBN: 978-607-15-0314-5.
* Sommerville, Ian. 2011. “Ingeniería del software. Novena edición”. Pearson Educación. Madrid. ISBN: 978-607-32-0603-7.
* Yourdon, Edward. 1993. “Análisis Estructurado Moderno”. Prentice-Hall Hispanoamericana. Naucalpan de Juárez, Edo. de México. ISBN: 968-880-303-0.