|  |
| --- |
|  |

Especificación de requisitos de software

**Proyecto: QuickMenu.**

|  |  |  |
| --- | --- | --- |
|  |  | Mayo del 2018 |

Contenido

[Contenido 2](#_Toc515366058)

[1 Introducción 3](#_Toc515366059)

[1.1 Propósito 3](#_Toc515366060)

[1.2 Alcance 3](#_Toc515366061)

[1.3 Personal involucrado 3](#_Toc515366062)

[1.4 Definiciones, acrónimos y abreviaturas 4](#_Toc515366063)

[1.5 Referencias 4](#_Toc515366064)

[1.6 Resumen 4](#_Toc515366065)

[2 Descripción general 4](#_Toc515366066)

[2.1 Objetivo General 4](#_Toc515366067)

[2.2 Objetivos Específicos 4](#_Toc515366068)

[2.3 Perspectiva del producto 4](#_Toc515366069)

[2.4 Funcionalidad del producto 5](#_Toc515366070)

[2.5 Características de los usuarios 29](#_Toc515366071)

[2.6 Restricciones 30](#_Toc515366072)

[2.7 Suposiciones y dependencias 30](#_Toc515366073)

[3 Marco Teórico 30](#_Toc515366074)

[4 Requisitos específicos 34](#_Toc515366075)

[4.1 Requisitos comunes de las interfaces 37](#_Toc515366076)

[4.1.1 Interfaces de usuario 37](#_Toc515366077)

[4.1.2 Interfaces de hardware 38](#_Toc515366078)

[4.1.3 Interfaces de software 38](#_Toc515366079)

[4.1.4 Interfaces de comunicación 38](#_Toc515366080)

# Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el proyecto QuickMenu, que es una plataforma web por dominios para la gestión de restaurantes y para sitios de comida.

## Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema de información web que permitirá gestionar distintos procesos respecto a los diferentes pedidos y empleados en los sitios de comida. Éste será utilizado por clientes de restaurantes, personal de cocina, camarero y directivos.

## Alcance

Esta especificación de requisitos está dirigida al usuario del sistema, donde se pretende especificar las características básicas del sistema de gestión QuickMenu y buscar potenciales puntos de mejora.

## Personal involucrado

|  |  |
| --- | --- |
| **Nombre** | Nicolás Angarita Ortiz |
| **Rol** | Programador Responsive, Analista. |
| **Categoría Profesional** | Ingeniero de Sistemas. |
| **Responsabilidad** | Adaptación de página web a dispositivos moviles |
| **Información de contacto** | [nicolas-angaritao@unilibre.edu.co](mailto:nicolas-angaritao@unilibre.edu.co) |

|  |  |
| --- | --- |
| **Nombre** | Santiago Jara González |
| **Rol** | Analista, Programador Back-End |
| **Categoría Profesional** | Ingeniero de Sistemas |
| **Responsabilidad** | Programación en ASP.NET y gestión BD en MySQL |
| **Información de contacto** | [santiago-jarag@unilibre.edu.co](mailto:santiago-jarag@unilibre.edu.co) |

|  |  |
| --- | --- |
| **Nombre** | David Leonardo Ruiz Carreño |
| **Rol** | Analista, Programador Full Stack |
| **Categoría Profesional** | Ingeniero de Sistemas |
| **Responsabilidad** | Programación Front-end, diseño de aplicación. |
| **Información de contacto** | [davidl-ruizc@unilibre.edu.co](mailto:davidl-ruizc@unilibre.edu.co) |

## Definiciones, acrónimos y abreviaturas

|  |  |
| --- | --- |
| ***Nombre*** | ***Descripción*** |
| **Usuario** | Persona que usará el sistema para gestionar procesos |
| **ERS** | Especificación de Requisitos Software |
| **RF** | Requerimiento Funcional |
| **RNF** | Requerimiento No Funcional |

## Referencias

|  |  |
| --- | --- |
| **Titulo del Documento** | **Referencia** |
| Standard IEEE 830 - 1998 | IEEE |
|  |  |

## Resumen

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

# Descripción general

## Objetivo General

Desarrollar una plataforma web para la creación y gestión de sitios web para establecimientos de comida o similares.

## Objetivos Específicos

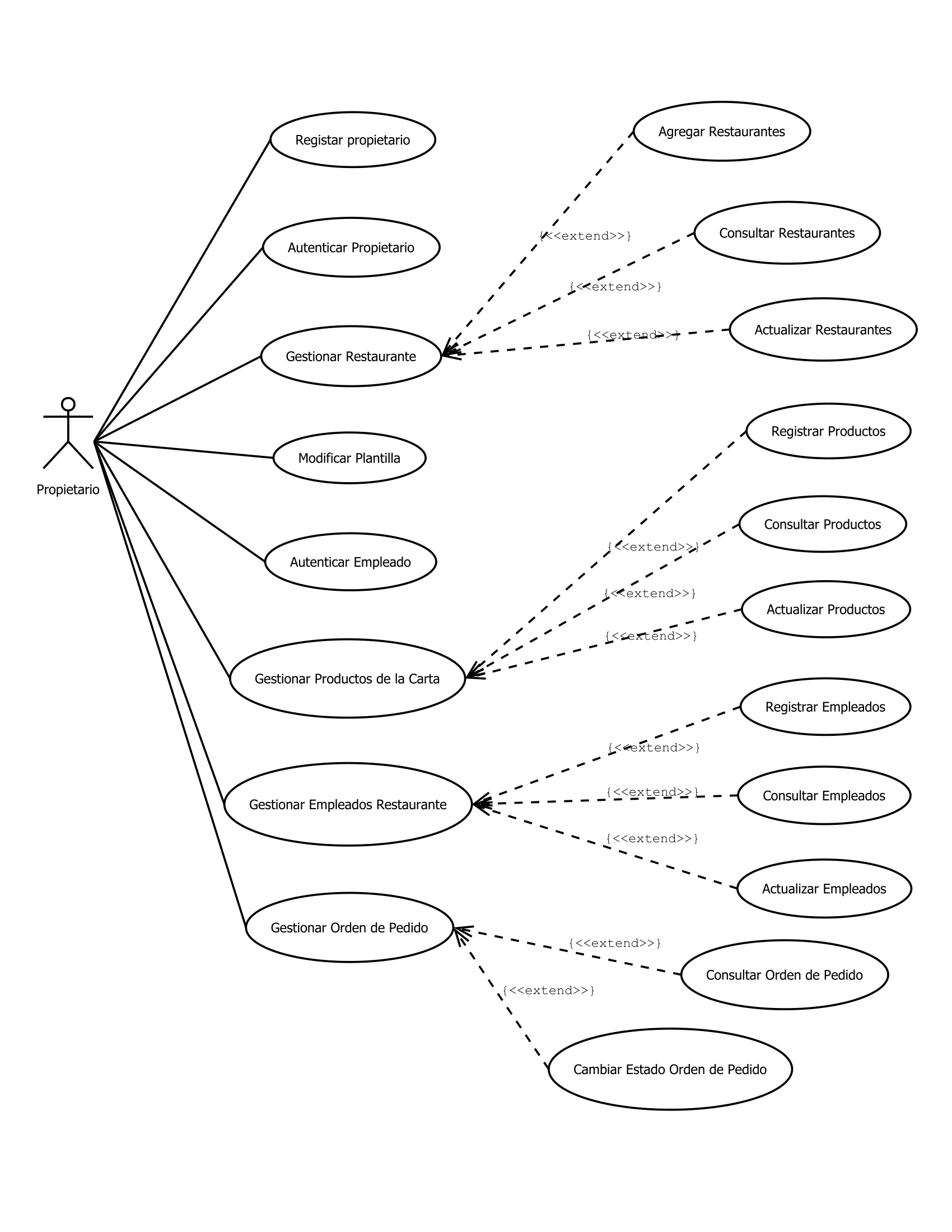
* Diseñar y crear las vistas de usuario en lenguaje de etiquetado HTML
* Diseñar y crear los modelos de base de datos en el sistema gestor de bases de datos MySQL
* Generar la solución en ASP .NET MVC aplicando los diseños y bases de datos creados anteriormente.

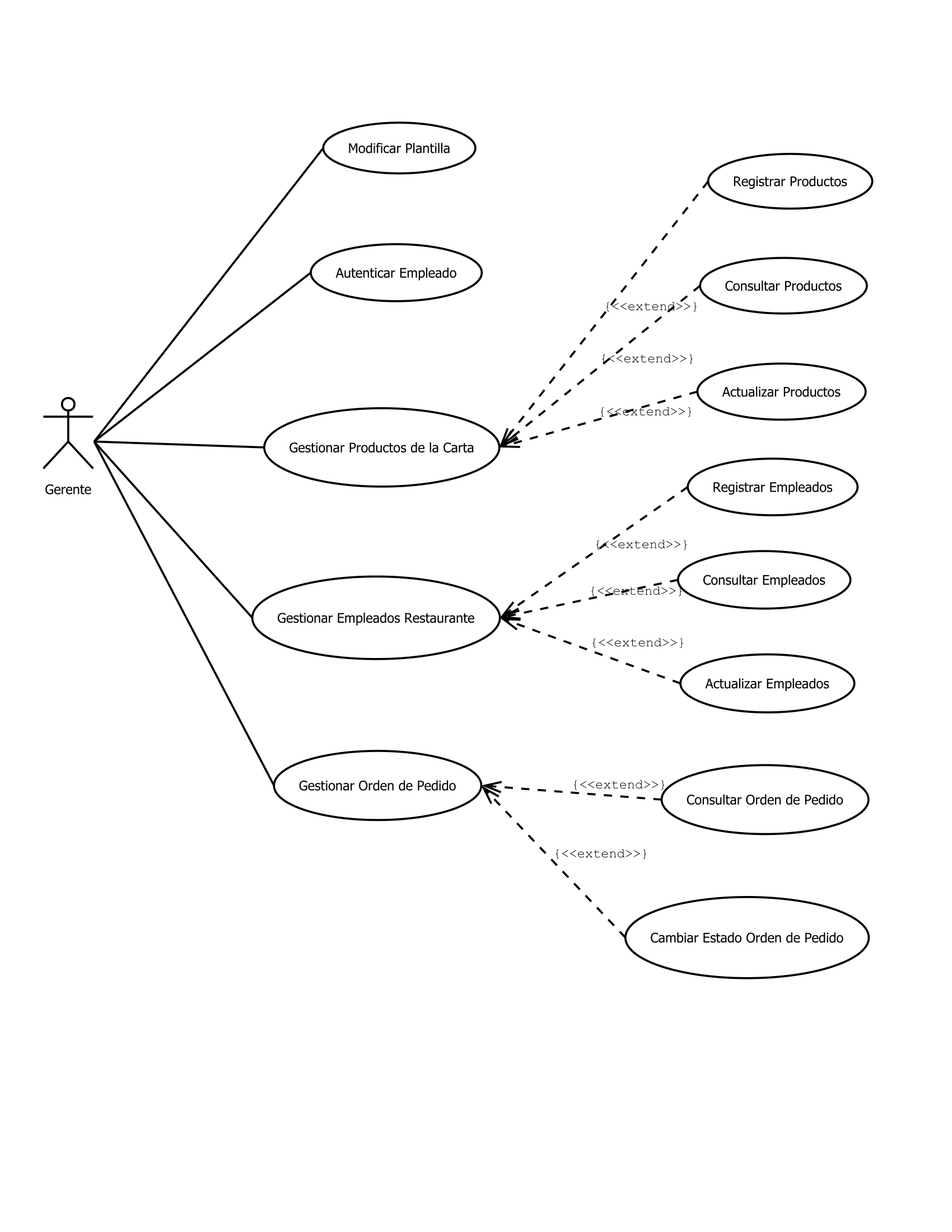
## Perspectiva del producto

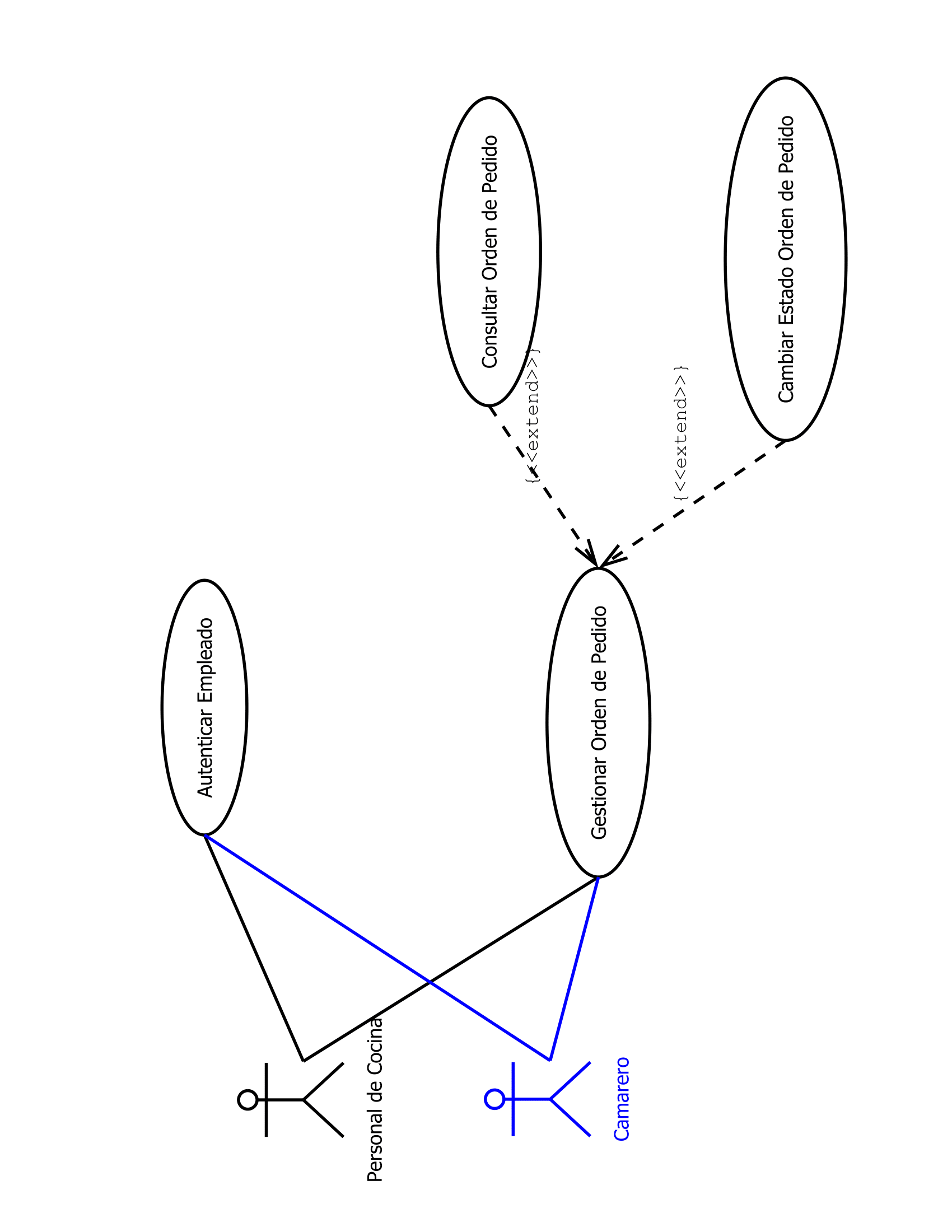
El sistema QuickMenu será un producto diseñado para trabajar en entornos WEB, lo que permitirá su utilización de forma rápida y eficaz, además de contar con soporte para dispositivos móviles logrando una mejor respuesta.

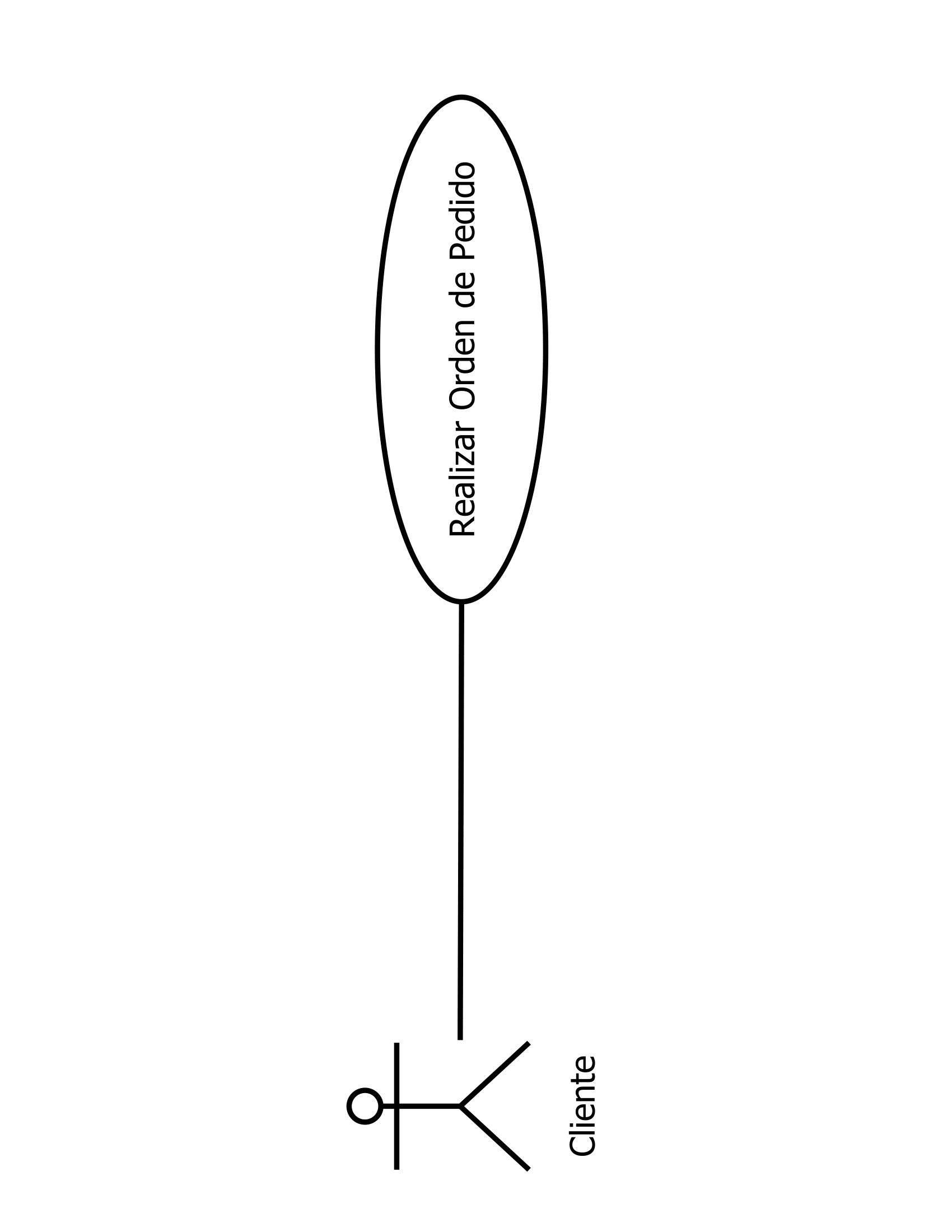
## Funcionalidad del producto

**2.4.1 Diagramas de Casos de Uso**





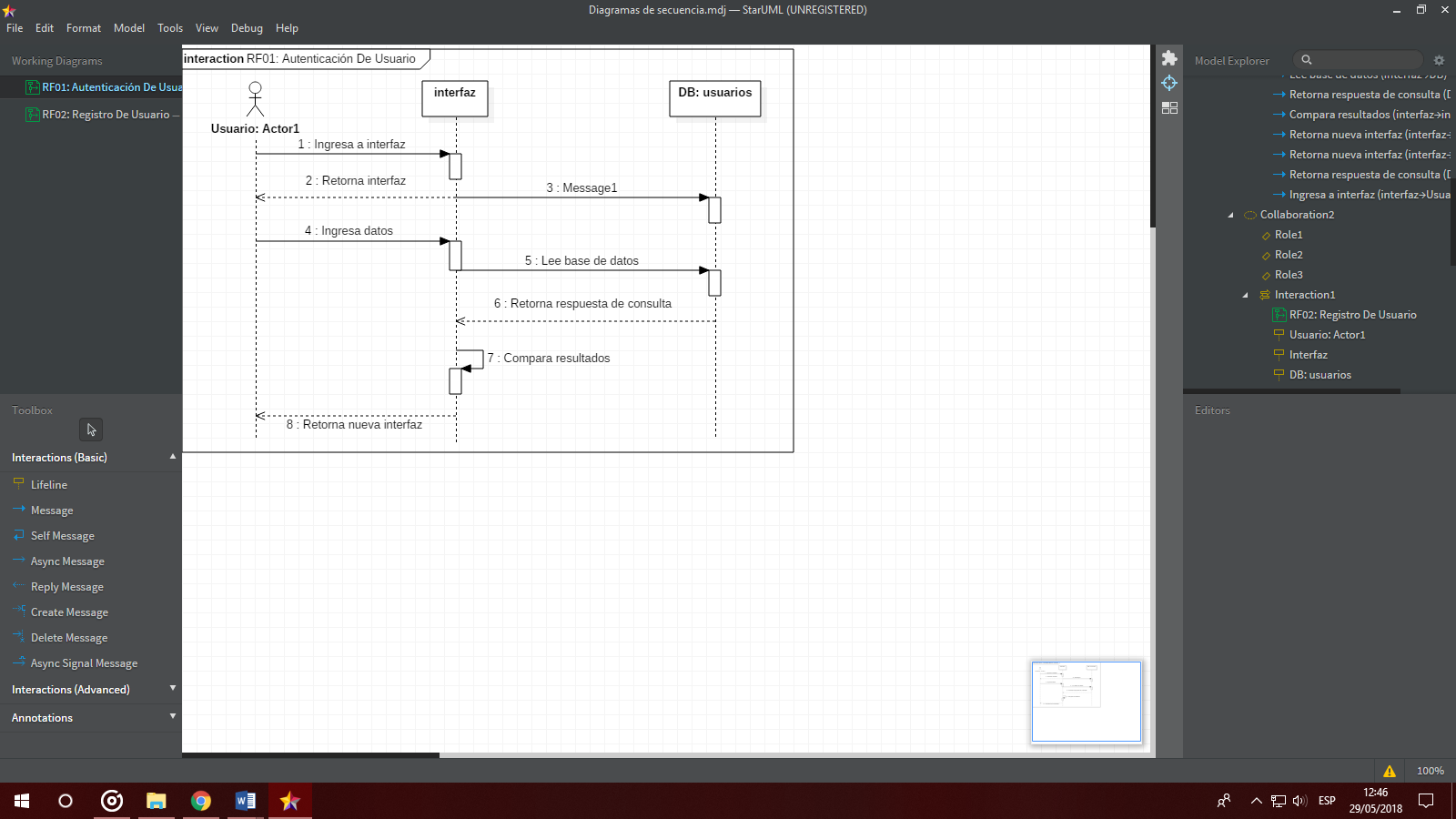




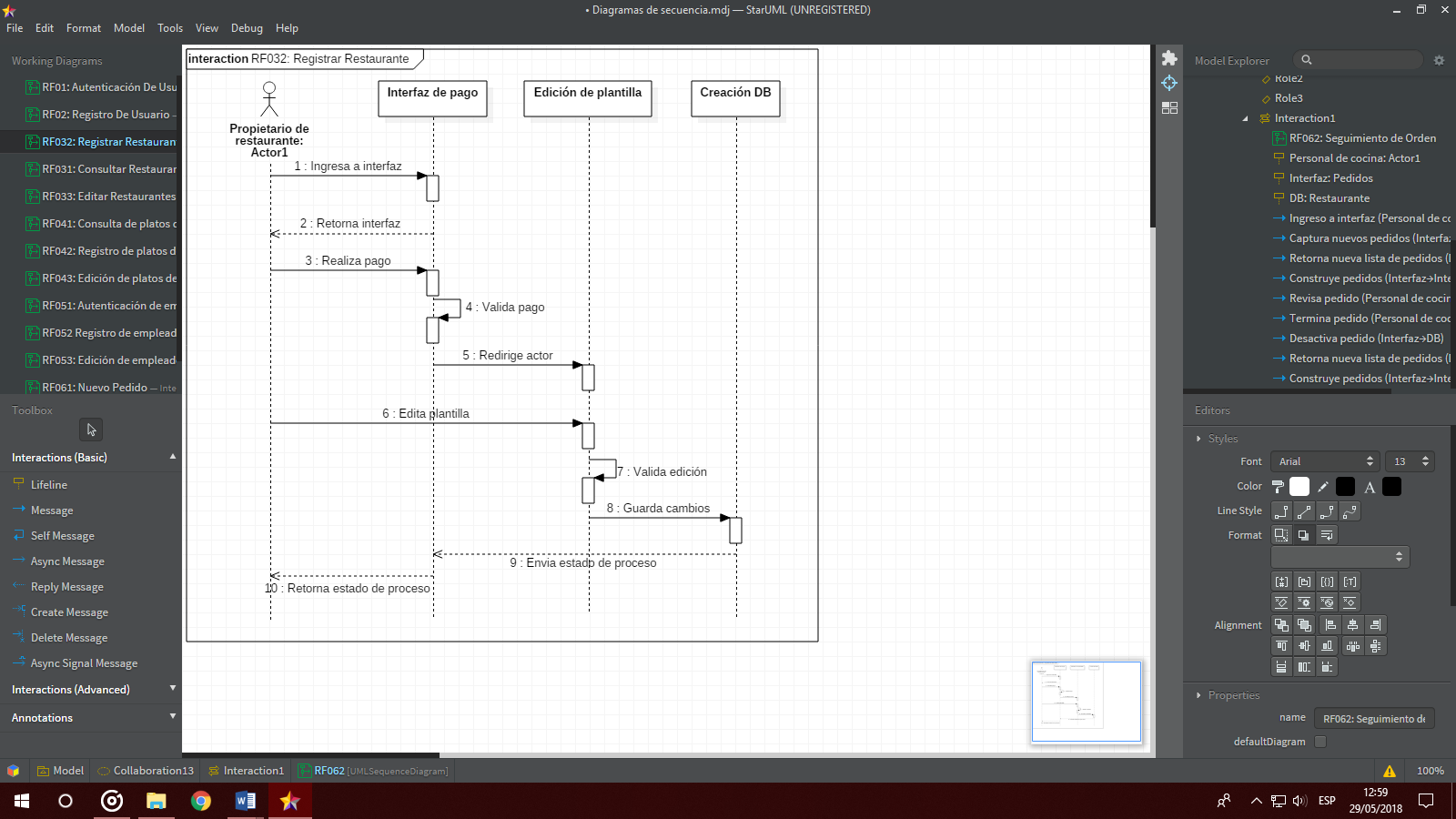
**2.4.2 Descripción Casos de Uso – Diagramas de Secuencia**

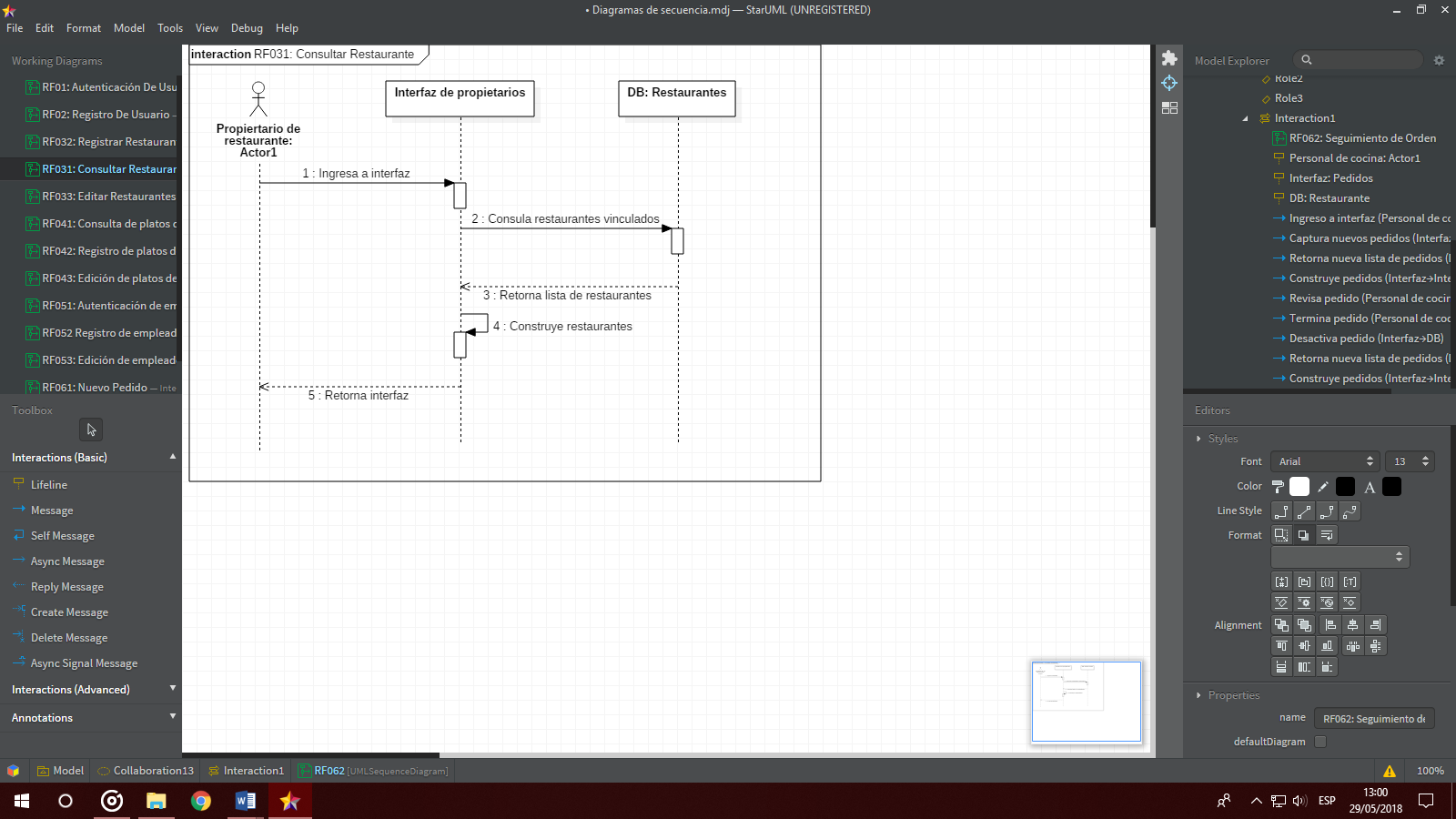
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | REGISTRAR PROPIETARIO SITIO COMIDA | | | | | | | | CU1 | |
| **Actores** | | Propietario. | | | | | | | | | |
| **Referencias** | | N/A | | | | | | | | | |
| **Precondición** | | N/A | | | | | | | | | |
| **Postcondición** | | Inserción de registro en la base de datos | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | |
| Registrar propietarios en la base de datos para el acceso al módulo de gestión de restaurantes. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | |
| En la tabla User los propietarios se podrán registrar ingresando datos básicos (nombre, correo, contraseña). | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | |
| 1 | El propietario ingresa al website de quickmenu y selecciona la opción de registrarse | | | 2 | | El propietario diligencia el formulario | | | | | |
| 3 | Se validan los campos, se realiza la inserción SQL. | | | 4 | | Se presenta el panel de gestión de restaurantes | | | | | |
|  | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | |
| 3a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | |
| **Frecuencia esperada** | | | 10 mensuales | | **Rendimiento** | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | Alta | | | |
|  | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | |
| **Campos obligatorios:** nombre, correo, contraseña | | | | | | | | | | | |
|  | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | AUTENTICACION PROPIETARIOS | | | | | | | | CU2 | |
| **Actores** | | Propietario | | | | | | | | | |
| **Referencias** | | Registrar Propietario | | | | | | | | | |
| **Precondición** | | Debe haber un propietario registrado | | | | | | | | | |
| **Postcondición** | | Muestra módulo de gestión de restaurantes | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | |
| Permitir al propietario ingresar al sistema. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | |
| Las personas que se encuentren activas en la tabla user pueden ingresar al sistema usando correo y una contraseña. Se valida su información y se le habilita el ingreso al sistema. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | |
| 1 | El propietario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del propietario. Comparación de correo y contraseña con la base de datos. Se muestra módulo de gestión de restaurantes | | | | | |
|  | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | |
| **Frecuencia esperada** | | | 2-5 diarias | | **Rendimiento** | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | Alta | | | |
|  | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | |
|  | | | | | | | | | | | |

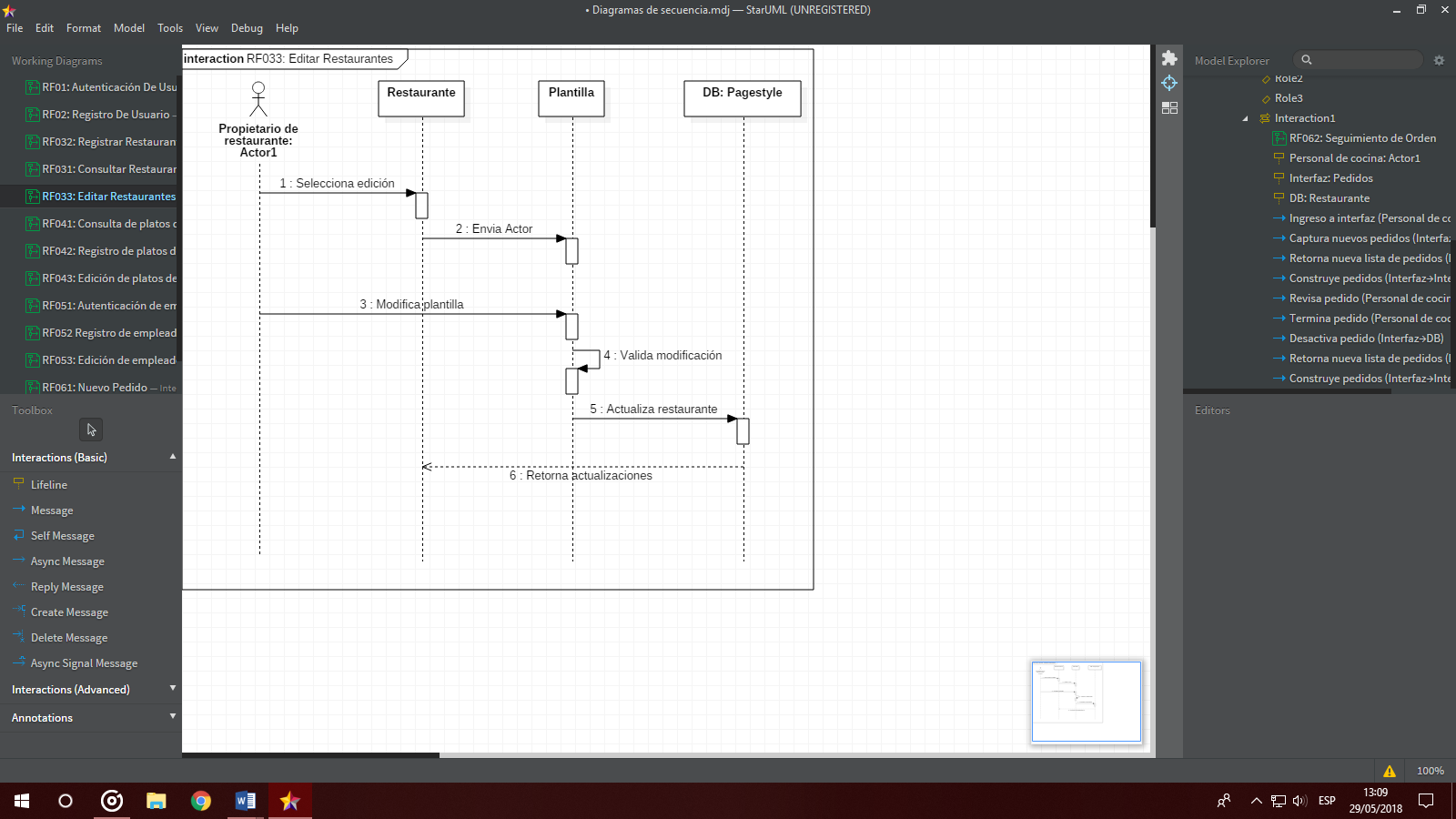


|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | REGISTRAR RESTAURANTE | | | | | | | | CU3 | |
| **Actores** | | Propietario. | | | | | | | | | |
| **Referencias** | | Ingreso al sistema | | | | | | | | | |
| **Precondición** | | Validación de usuario satisfactoria. | | | | | | | | | |
| **Postcondición** | | Inserción de registro en la base de datos | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | |
| Registrar restaurantes en la base de datos principal para realizar su gestión con su propia base de datos generada automáticamente. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | |
| En la tabla restaurant los propietarios podrán registrar diferentes restaurantes ingresando datos básicos del restaurante (nombre). | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | |
| 1 | El propietario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del usuario. Comparación de correo y contraseña con la base de datos. Muestra módulo de gestión | | | | | |
| 3 | El usuario selecciona la opción registrar restaurante | | | 4 | | Se abre el formulario de registro de restaurantes | | | | | |
| 5 | El usuario diligencia el formulario de registro de restaurante. | | | 6 | | Se validan los campos. Y se realiza la inserción SQL. Se muestra modulo actualizado. | | | | | |
|  | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | |
| 6a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | |
| **Frecuencia esperada** | | | 10 mensuales | | **Rendimiento** | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | Alta | | | |
|  | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | |
| **Campos obligatorios:** nombre | | | | | | | | | | | |

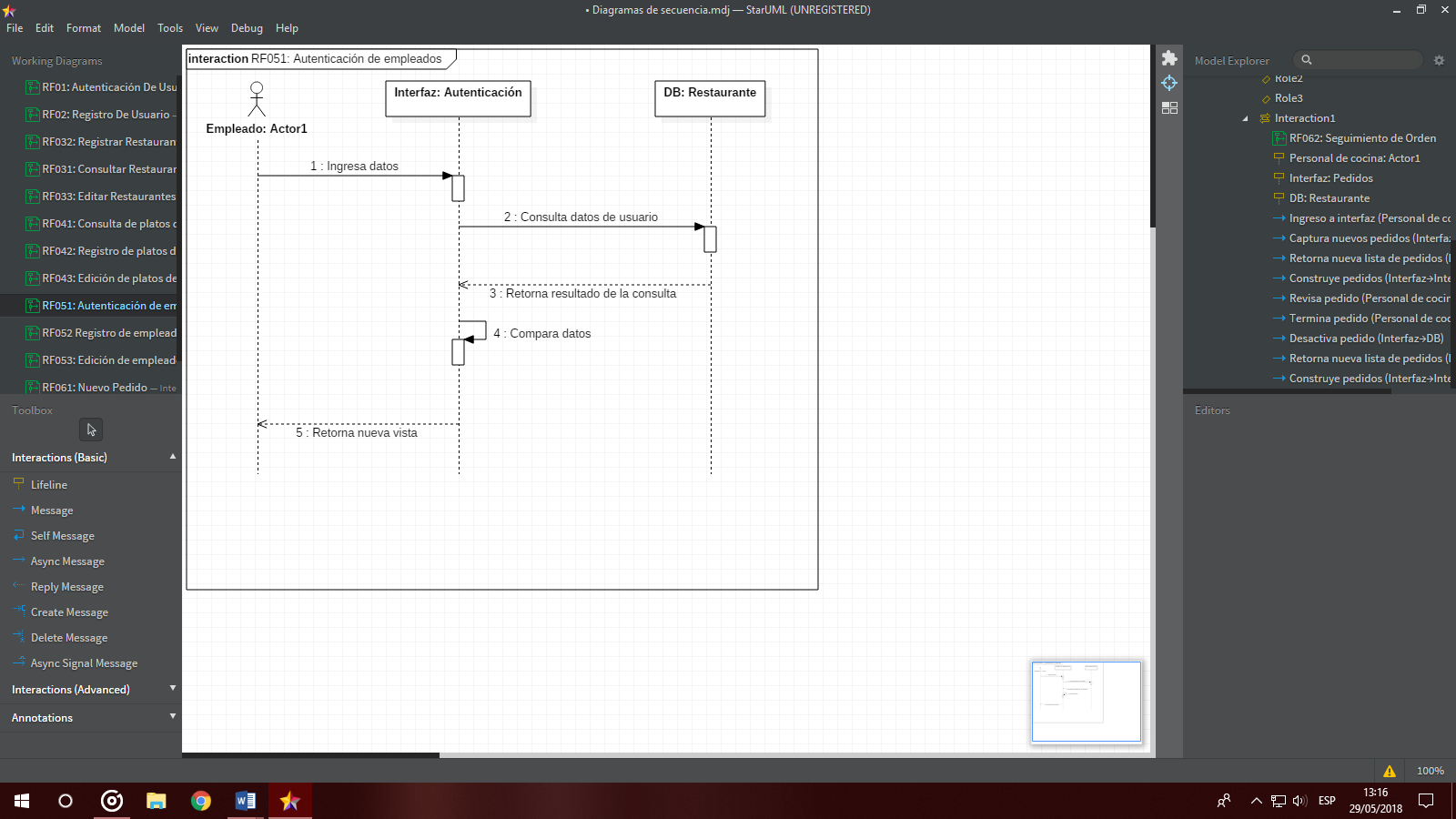




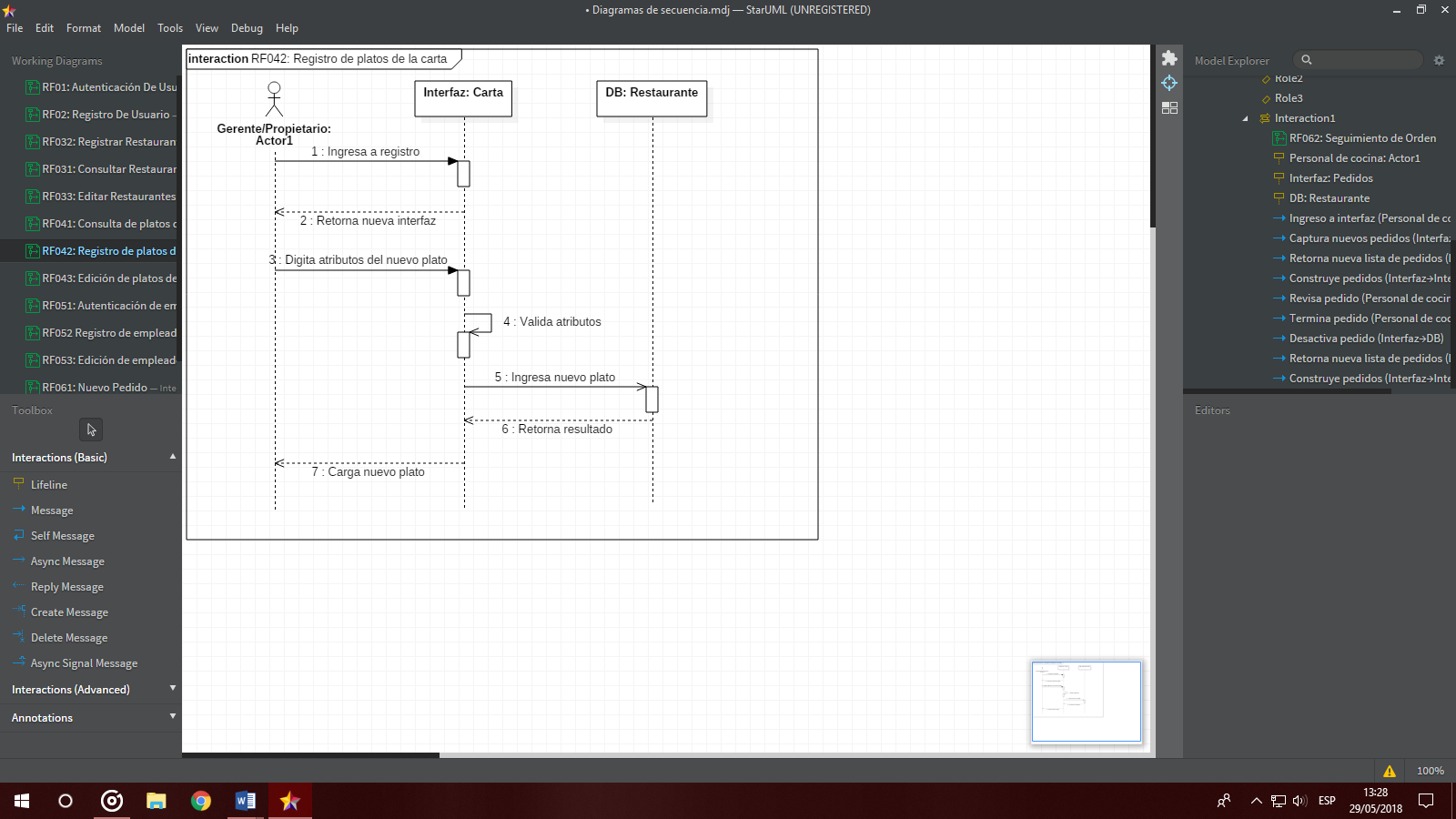
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | MODIFICAR PLANTILLA SITIO | | | | | | | | CU4 | |
| **Actores** | | Propietario-Gerente. | | | | | | | | | |
| **Referencias** | | Ingreso al sistema-Registrar Restaurante | | | | | | | | | |
| **Precondición** | | Validación de usuario satisfactoria-Restaurante registrado. | | | | | | | | | |
| **Postcondición** | | Inserción de registro en la base de datos con detalles del sitio web | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | |
| Modificar y personalizar el sitio web para cada restaurante | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | |
| En la tabla pagestyle se guardaran los cambios en la plantilla realizados por el gerente o propietario. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | |
| 1 | El usuario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del usuario. Comparación de correo y contraseña con la base de datos. Muestra módulo de gestión | | | | | |
| 3 | El usuario selecciona la opción plantilla | | | 4 | | Se abre el formulario de modificación de plantilla | | | | | |
| 5 | El usuario diligencia el formulario de modificación de plantilla. | | | 6 | | Se validan los campos. Y se realiza la inserción SQL. Se actualiza sitio web. | | | | | |
|  | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | |
| 6a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | |
| **Frecuencia esperada** | | | 1 o rara vez | | **Rendimiento** | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | Alta | | | |
|  | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | |
|  | | | | | | | | | | | |

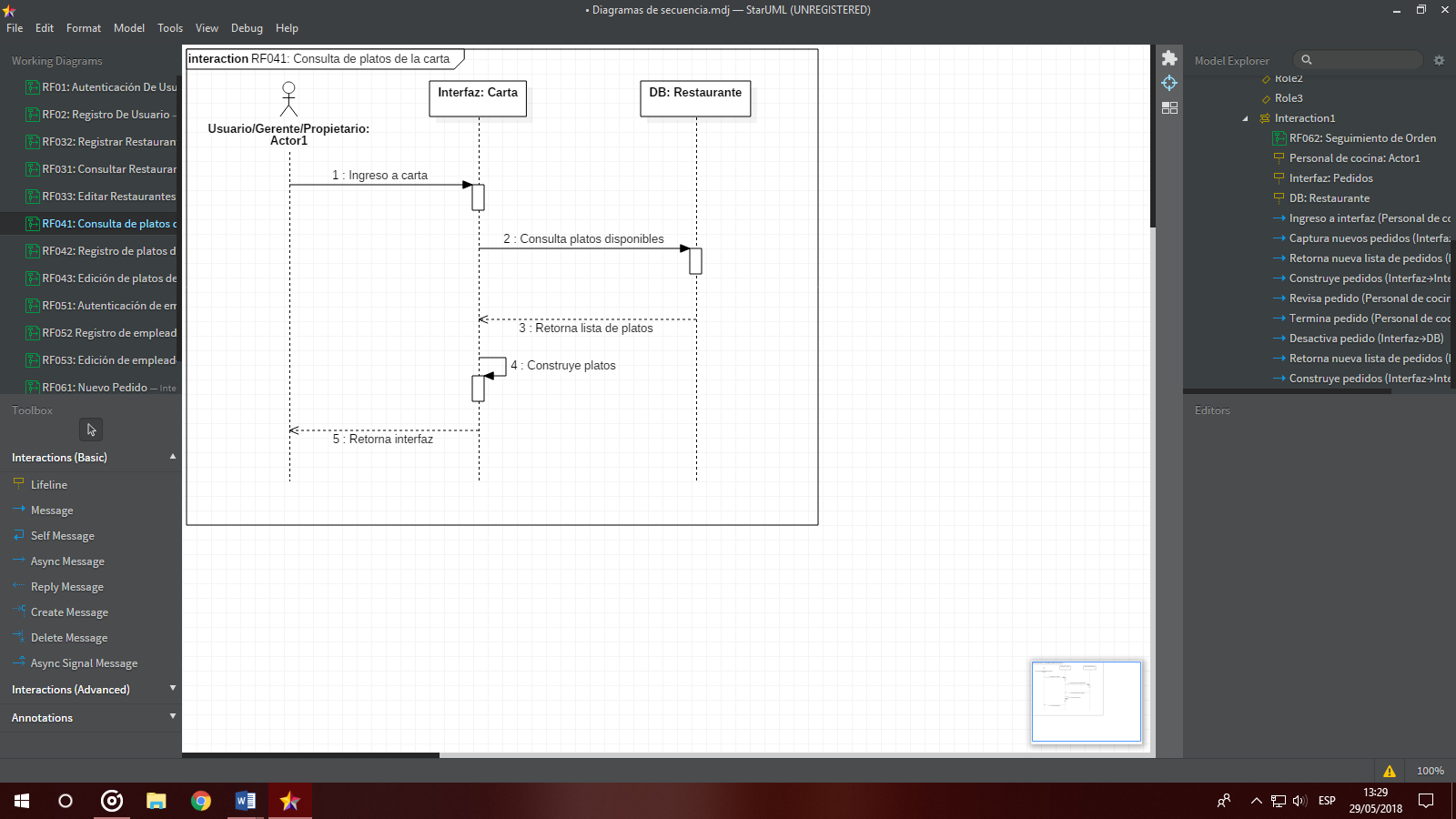


|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | AUTENTICACION EMPLEADOS | | | | | | | | CU5 | |
| **Actores** | | Gerente-Personal Cocina-Camarero | | | | | | | | | |
| **Referencias** | | Registrar Empleado | | | | | | | | | |
| **Precondición** | | Debe haber un empleado registrado | | | | | | | | | |
| **Postcondición** | | Muestra módulo de gestión | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | |
| Permitir a los empleados ingresar al sistema según sus permisos. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | |
| Las personas que se encuentren activas en la tabla employee pueden ingresar al sistema usando correo y una contraseña. Se valida su información y se le habilita el ingreso al sistema con los permisos de usuario correspondientes. | | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | |
| 1 | El propietario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del propietario. Comparación de correo y contraseña con la base de datos. Se muestra módulo de gestión según permisos | | | | | |
|  | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | |
|  | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | |
| **Frecuencia esperada** | | | 100 diarias | | **Rendimiento** | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | Alta | | | |
|  | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | |
|  | | | | | | | | | | | |

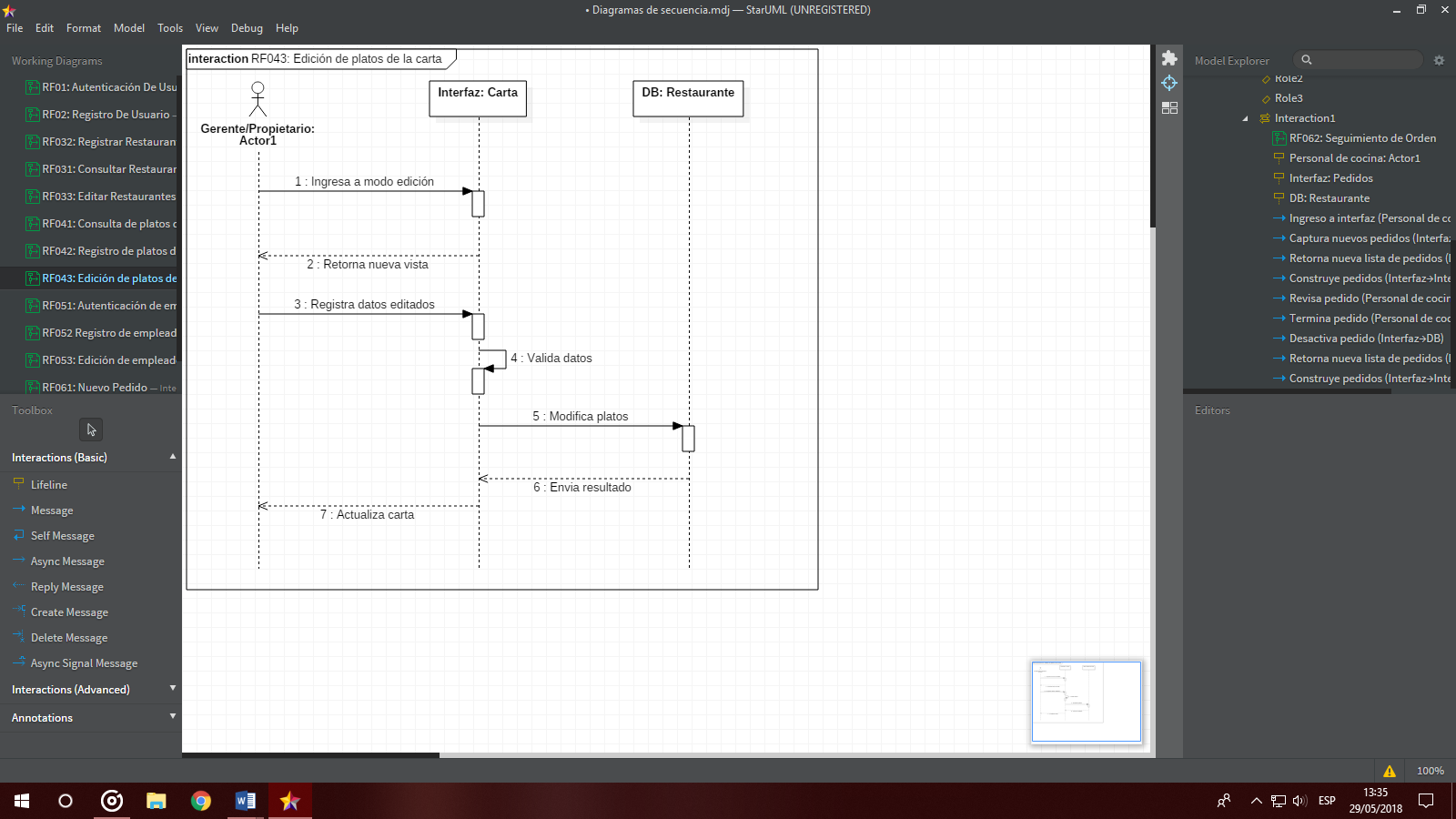


|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | REGISTRAR PRODUCTO DE LA CARTA | | | | | | | | | CU6.1 | |
| **Actores** | | Propietario-Gerente. | | | | | | | | | | |
| **Referencias** | | Ingreso al sistema | | | | | | | | | | |
| **Precondición** | | Validación de usuario satisfactoria. (Validación de permisos) | | | | | | | | | | |
| **Postcondición** | | Inserción de registro en la base de datos | | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | | |
| Registrar productos en la base de datos para ser mostrados en la carta. Esta función la puede realizar el propietario o gerente. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | | |
| En la tabla producto el usuario podrá registrar diferentes productos ingresando datos básicos del producto (id, nombre, precio, imagen, detalles). | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | | |
| 1 | El usuario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del usuario. Comparación de correo, contraseña y cargo con la base de datos. Muestra módulo gestión según permisos | | | | | | |
| 3 | El usuario selecciona la opción registrar producto. | | | 4 | | Se abre el formulario de registro de producto | | | | | | |
| 5 | El usuario diligencia el formulario de registro de producto. | | | 6 | | Se validan los campos. Y se realiza la inserción SQL. Se presenta aviso de confirmación de inserción. | | | | | | |
|  | | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | | |
| 6a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | | |
| **Frecuencia esperada** | | | 10-15 en primera vez- 1 mensual | | **Rendimiento** | | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | | Alta | | | |
|  | | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | | |
| **Campos obligatorios:** id, nombre, precio | | | | | | | | | | | | |

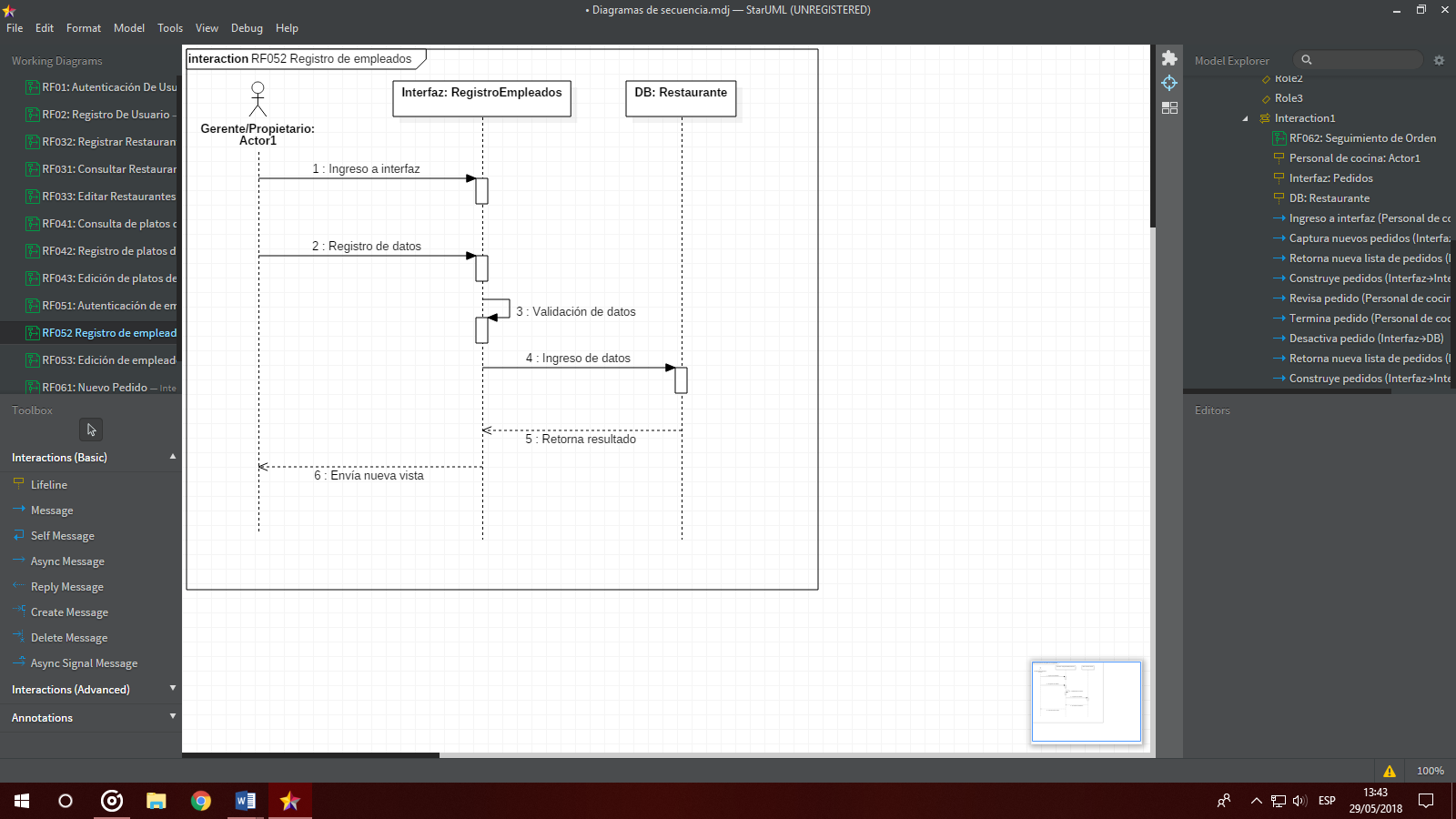




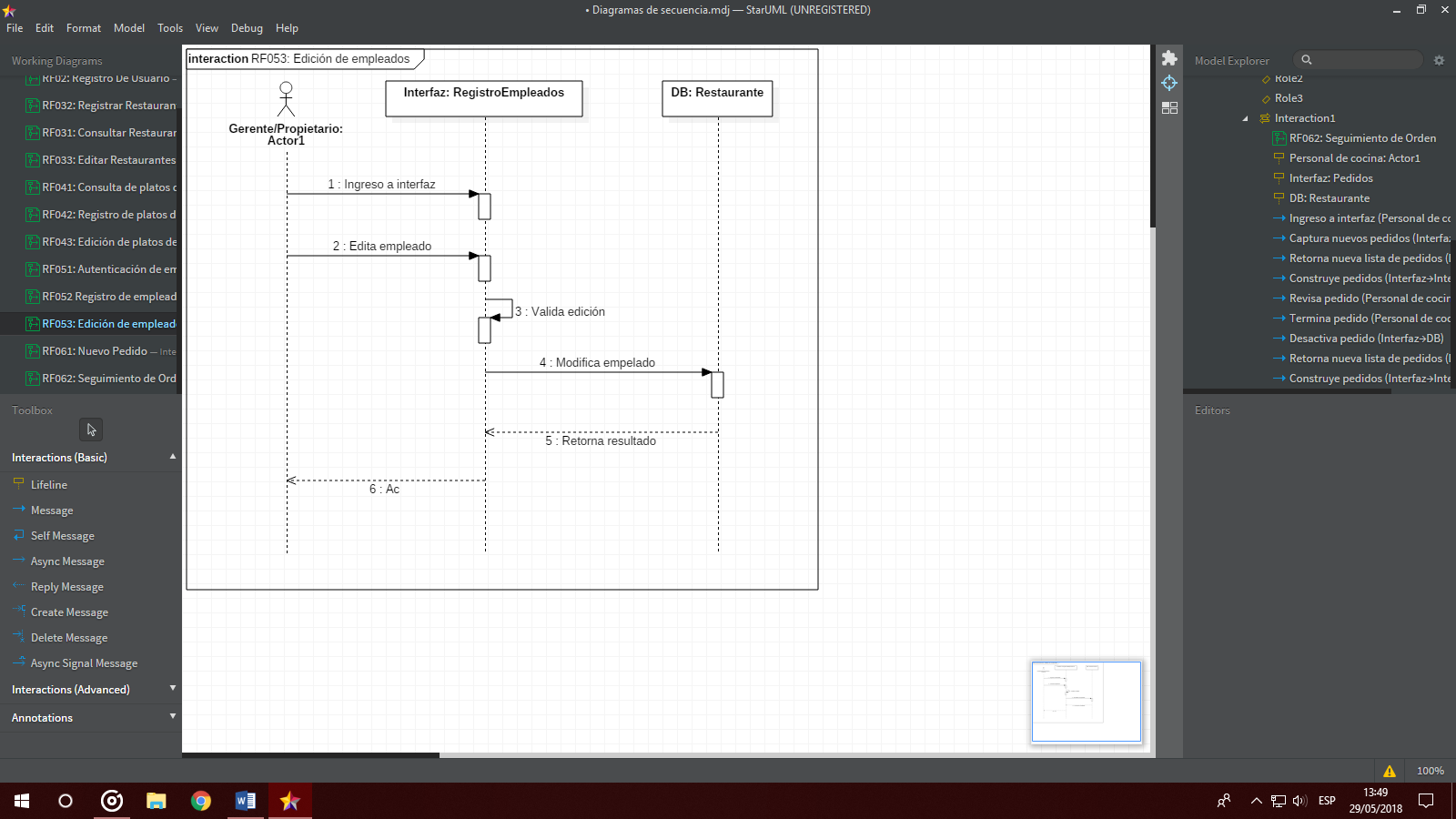
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | EDITAR PRODUCTO DE LA CARTA | | | | | | | | | CU6.2 | |
| **Actores** | | Propietario-Gerente. | | | | | | | | | | |
| **Referencias** | | Ingreso al sistema-Registrar producto | | | | | | | | | | |
| **Precondición** | | Validación de usuario satisfactoria. (Validación de permisos)-Producto Registrado | | | | | | | | | | |
| **Postcondición** | | Actualización de registro en la base de datos | | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | | |
| Actualizar productos en la base de datos para ser mostrados en la carta. Esta función la puede realizar el propietario o gerente. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | | |
| En la tabla producto el usuario podrá actualizar diferentes productos ingresando datos básicos del producto (nombre, precio, imagen, detalles). | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | | |
| 1 | El usuario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del usuario. Comparación de correo, contraseña y cargo con la base de datos. Muestra módulo gestión según permisos | | | | | | |
| 3 | El usuario selecciona la opción editar producto. | | | 4 | | Se abre el formulario de edición de producto | | | | | | |
| 5 | El usuario diligencia el formulario de edición de producto. | | | 6 | | Se validan los campos. Y se realiza la actualización SQL. Se presenta aviso de confirmación. | | | | | | |
|  | | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | | |
| 6a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | | |
| **Frecuencia esperada** | | | 1 mensual | | **Rendimiento** | | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | | Alta | | | |
|  | | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | | |
| **Campos obligatorios:** id, nombre, precio | | | | | | | | | | | | |



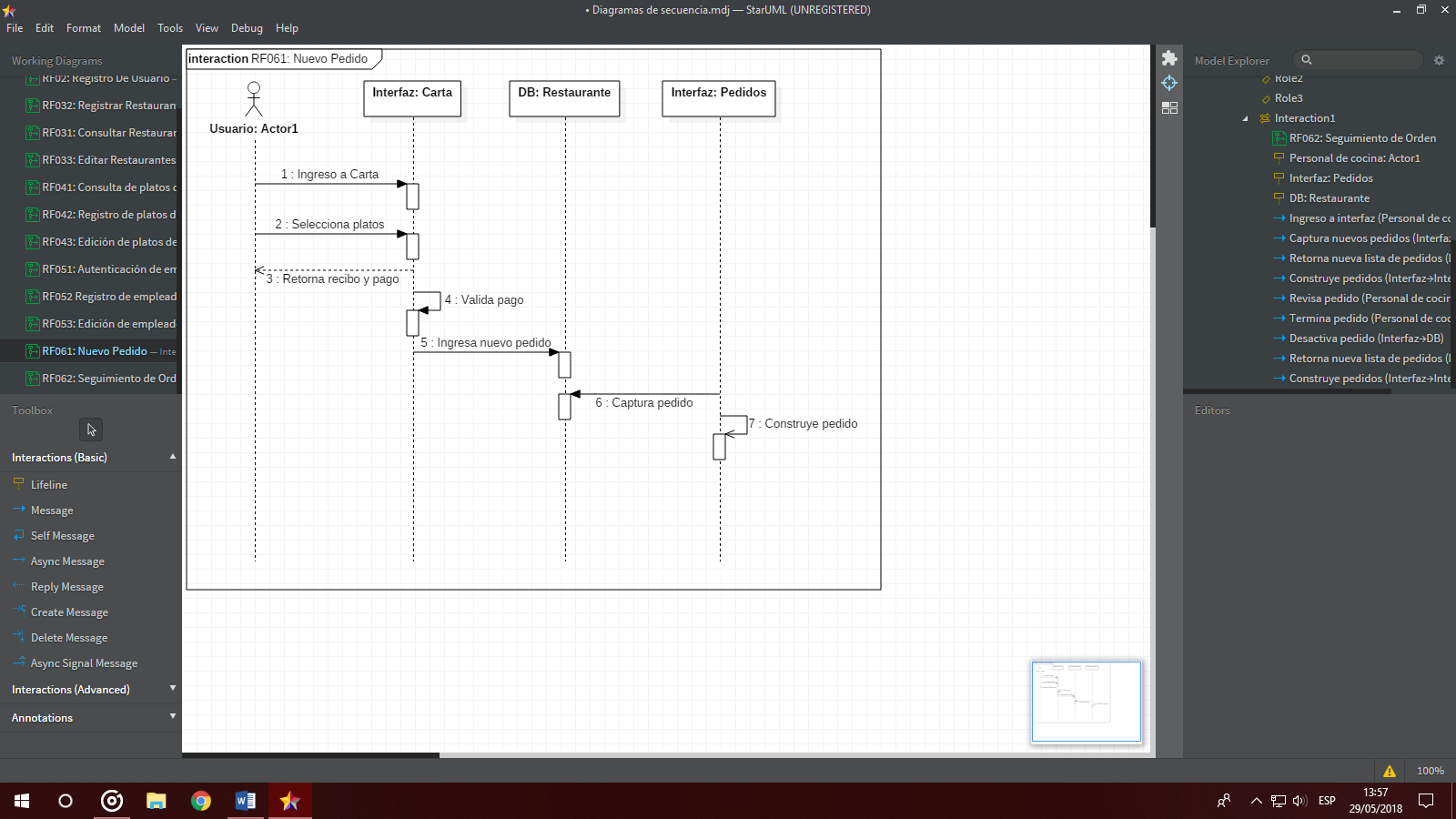
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | REGISTRAR EMPLEADO | | | | | | | | | CU7.1 | |
| **Actores** | | Propietario-Gerente. | | | | | | | | | | |
| **Referencias** | | Ingreso al sistema | | | | | | | | | | |
| **Precondición** | | Validación de usuario satisfactoria. (Validación de permisos) | | | | | | | | | | |
| **Postcondición** | | Inserción de registro en la base de datos | | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | | |
| Registrar empleados en la base de datos para que puedan acceder al sistema. Esta función la puede realizar el propietario o gerente. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | | |
| En la tabla empleado el usuario podrá registrar diferentes empleados ingresando datos básicos del empleado (nombre, email, contraseña, cargo). | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | | |
| 1 | El usuario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del usuario. Comparación de correo, contraseña y cargo con la base de datos. Muestra módulo gestión según permisos | | | | | | |
| 3 | El usuario selecciona la opción registrar empleado. | | | 4 | | Se abre el formulario de registro de empleado. | | | | | | |
| 5 | El usuario diligencia el formulario de registro de empleado. | | | 6 | | Se validan los campos. Y se realiza la inserción SQL. Se presenta aviso de confirmación de inserción. | | | | | | |
|  | | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | | |
| 6a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | | |
| **Frecuencia esperada** | | | 1 mensual | | **Rendimiento** | | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | | Alta | | | |
|  | | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | | |
| **Campos obligatorios:** correo, nombre, contraseña, cargo | | | | | | | | | | | | |



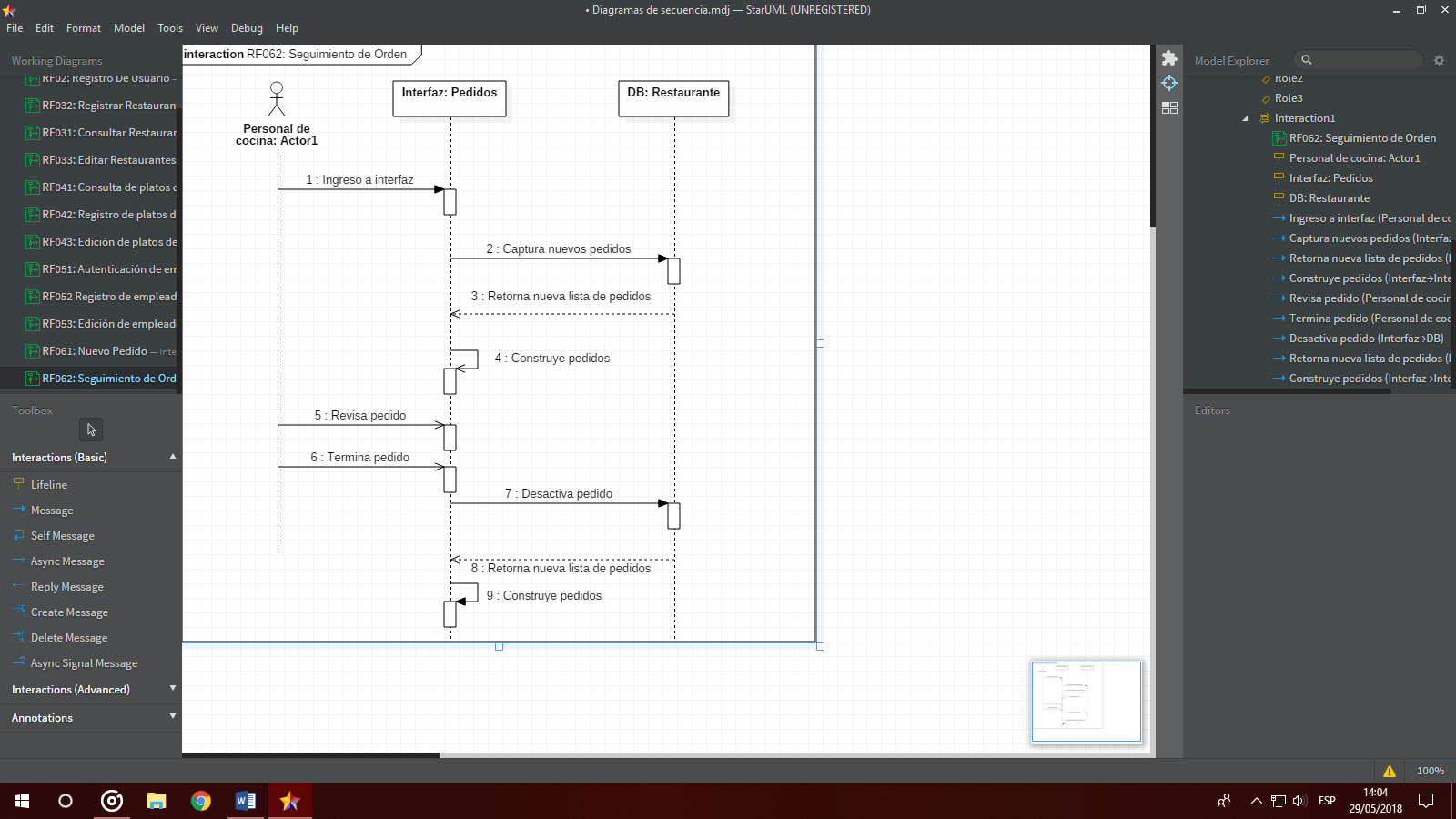
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | EDITAR EMPLEADO | | | | | | | | | CU7.2 | |
| **Actores** | | Propietario-Gerente. | | | | | | | | | | |
| **Referencias** | | Ingreso al sistema | | | | | | | | | | |
| **Precondición** | | Validación de usuario satisfactoria. (Validación de permisos) | | | | | | | | | | |
| **Postcondición** | | Inserción de registro en la base de datos | | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | | |
| Editar empleados en la base de datos para que puedan acceder al sistema. Esta función la puede realizar el propietario o gerente. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | | |
| En la tabla empleado el usuario podrá actualizar diferentes empleados ingresando datos básicos del empleado (nombre, contraseña, cargo). | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | | |
| 1 | El usuario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del usuario. Comparación de correo, contraseña y cargo con la base de datos. Muestra módulo gestión según permisos | | | | | | |
| 3 | El usuario selecciona la opción editar empleado. | | | 4 | | Se abre el formulario de edición de empleado. | | | | | | |
| 5 | El usuario diligencia el formulario de edición de empleado. | | | 6 | | Se validan los campos. Y se realiza la actualización SQL. Se presenta aviso de confirmación. | | | | | | |
|  | | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | | |
| 6a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | | |
| **Frecuencia esperada** | | | 1 mensual | | **Rendimiento** | | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | | Alta | | | |
|  | | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | | |
| **Campos obligatorios:** correo, nombre, contraseña, cargo | | | | | | | | | | | | |



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | REGISTRAR ORDEN DE PEDIDO | | | | | | | | | CU8 | |
| **Actores** | | Cliente. | | | | | | | | | | |
| **Referencias** | | Registrar producto | | | | | | | | | | |
| **Precondición** | | Producto registrado | | | | | | | | | | |
| **Postcondición** | | Inserción de registro en la base de datos | | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | | |
| Registrar ordenes en la base de datos para que puedan ser realizadas por el personal de cocina. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | | |
| En la tabla orden el usuario podrá registrar diferentes órdenes ingresando datos básicos de la orden y los productos realacionados (mesa, método de pago , detalles). | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | | |
| 1 | El usuario ingresa al sitio web del restaurante y selecciona el menú. | | | 2 | | El usuario añade a su carrito de compras los productos deseados. | | | | | | |
| 3 | El usuario selecciona la opción finalizar orden y diligencia la información requerida. | | | 4 | | Se validan los campos. Y se realiza la inserción SQL. Se presenta aviso de confirmación de inserción. | | | | | | |
|  | | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | | |
| 4a | La validación indica que hay información incorrecta. No se permite el registro | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | | |
| **Frecuencia esperada** | | | 100 diarias | | **Rendimiento** | | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | | Alta | | | |
|  | | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CASO DE USO** | | SEGUIMIENTO ORDEN DE PEDIDO | | | | | | | | | CU9 | |
| **Actores** | | Propietario-Gerente-Personal Cocina-Camarero. | | | | | | | | | | |
| **Referencias** | | Ingreso al sistema | | | | | | | | | | |
| **Precondición** | | Validación de usuario satisfactoria. (Validación de permisos) | | | | | | | | | | |
| **Postcondición** | | Actualizacion de registro en la base de datos | | | | | | | | | | |
| **Autor** | | Jara González Santiago  Ruiz Carreño David  Angarita Ortiz Nicolas | | | | | **Fecha** | Mayo 29 de 2018 | | **Versión** | | 1 |
|  | | | | | | | | | | | | |
| **Propósito** | | | | | | | | | | | | |
| Mostrar información de la orden y modificar su estado según su avance. Esta función la puede realizar el propietario o gerente y los demás empleados. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Resumen** | | | | | | | | | | | | |
| En la tabla orden esta toda la información del pedido y se puede actualizar su estado según avance la preparacion. | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Curso Normal (Básico)** | | | | | | | | | | | | |
| 1 | El usuario ingresa al sistema empleando su correo y contraseña. | | | 2 | | Se valida la identidad del usuario. Comparación de correo, contraseña y cargo con la base de datos. Muestra módulo gestión según permisos | | | | | | |
| 3 | El usuario selecciona la opción órdenes. | | | 4 | | Se muestra la información de la orden y su estado. | | | | | | |
| 5 | El usuario cambia el estado de la orden. | | | 6 | | Se realiza la actualización SQL. Se presenta aviso de confirmación. | | | | | | |
|  | | | | | | | | | | | | |
| **Cursos Alternos** | | | | | | | | | | | | |
| 2a | Usuario sin permisos de ingreso, se valida la información ingresada y es rechazada. | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Otros Datos** | | | | | | | | | | | | |
| **Frecuencia esperada** | | | 1 mensual | | **Rendimiento** | | | | Satisfactorio | | | |
| **Importancia** | | | Vital | | **Urgencia** | | | | Alta | | | |
| **Estado** | | | Activo | | **Estabilidad** | | | | Alta | | | |
|  | | | | | | | | | | | | |
| **Comentarios** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |



## Características de los usuarios

|  |  |
| --- | --- |
| **Tipo de usuario** | Propietario de sitio de comida. |
| **Formación** | No aplica. |
| **Actividades** | Control y gestión de los diferentes dominios en su propiedad |

|  |  |
| --- | --- |
| **Tipo de usuario** | Gerente de restaurante |
| **Formación** | Administrativa |
| **Actividades** | Gestión de diseño de la página y del restaurante en general |

|  |  |
| --- | --- |
| **Tipo de usuario** | Personal de cocina |
| **Formación** | Culinaria |
| **Actividades** | Elaboración de los alimentos |

|  |  |
| --- | --- |
| **Tipo de usuario** | Camarero |
| **Formación** | No aplica |
| **Actividades** | Servicio al cliente |

|  |  |
| --- | --- |
| **Tipo de usuario** | Cliente |
| **Formación** | No aplica |
| **Actividades** | Realizar pedidos |

.

## Restricciones

* Interfaz para ser usada con internet.
* Uso de Dominio (X)
* Lenguajes y tecnologías en uso: HTML, CSS, Javascript, ASP.NET, MySQL.
* Los servidores deben ser capaces de atender consultas concurrentemente.
* El sistema se diseñará según un modelo cliente/servidor, Modelo Vista Controlador (MVC) y programación orientada a objetos (POO).
* 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

# Marco Teórico

**Internet**

El internet (o, también, la internet) es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, lo cual garantiza que las redes físicas heterogéneas que la componen formen una red lógica única de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California (Estados Unidos).

Uno de los servicios que más éxito ha tenido en internet ha sido la World Wide Web (WWW o la Web), hasta tal punto que es habitual la confusión entre ambos términos. La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Esta fue un desarrollo posterior (1990) y utiliza internet como medio de transmisión.

Existen, por tanto, muchos otros servicios y protocolos en internet, aparte de la Web: el envío de correo electrónico (SMTP), la transmisión de archivos (FTP y P2P), las conversaciones en línea (IRC), la mensajería instantánea y presencia, la transmisión de contenido y comunicación multimedia —telefonía (VoIP), televisión (IPTV)—, los boletines electrónicos (NNTP), el acceso remoto a otros dispositivos (SSH y Telnet) o los juegos en línea.

El uso de internet creció rápidamente en el hemisferio occidental desde la mitad de la década de 1990, y desde el final de la década en el resto del mundo. En los 20 años desde 1995, el uso de internet se ha multiplicado por 100, cubriendo en 2015 a la tercera parte de la población mundial. La mayoría de las industrias de comunicación, incluyendo telefonía, radio, televisión, correo postal y periódicos tradicionales están siendo transformadas o redefinidas por el Internet, permitiendo el nacimiento de nuevos servicios como correo electrónico (email), telefonía por internet, televisión por Internet, música digital, y video digital. Las industrias de publicación de periódicos, libros y otros medios impresos se están adaptando a la tecnología de los sitios web, o están siendo reconvertidos en blogs, web feeds o agregadores de noticias online (p. ej., Google Noticias). Internet también ha permitido o acelerado nuevas formas de interacción personal a través de mensajería instantánea, foros de Internet, y redes sociales como Facebook. El comercio electrónico ha crecido exponencialmente para tanto grandes cadenas como para pequeños y mediana empresa o nuevos emprendedores, ya que permite servir a mercados más grandes y vender productos y servicios completamente en línea. Relaciones business-to-business y de servicios financieros en línea en internet han afectado las cadenas

de suministro de industrias completas.

**World Wide Web**

En informática, la World Wide Web (WWW) o red informática mundial es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener textos, imágenes, vídeos u otros contenidos multimedia, y navega a través de esas páginas usando hiperenlaces.

La Web se desarrolló entre marzo de 1989 y diciembre de 1990.2 por el inglés Tim Berners-Lee con la ayuda del belga Robert Cailliau mientras trabajaban en el CERN en Ginebra, Suiza, y publicado en 1992. Desde entonces, Berners-Lee ha jugado un papel activo guiando el desarrollo de estándares Web (como los lenguajes de marcado con los que se crean las páginas web), y en los últimos años ha abogado por su visión de una Web semántica. Utilizando los conceptos de sus anteriores sistemas de hipertexto como ENQUIRE, el físico británico Tim Berners-Lee, un científico de la computación y en ese tiempo de los empleados del CERN, ahora director del World Wide Web Consortium (W3C), escribió una propuesta en marzo de 1989 con lo que se convertiría en la World Wide Web. La propuesta de 1989 fue destinada a un sistema de comunicación CERN, pero Berners-Lee finalmente se dio cuenta que el concepto podría aplicarse en todo el mundo. En la CERN, la organización europea de investigación cerca de Ginebra, en la frontera entre Francia y Suiza, Berners-Lee y el científico de la computación belga Robert Cailliau propusieron en 1990 utilizar el hipertexto "para vincular y acceder a información de diversos tipos como una red de nodos en los que el usuario puede navegar a voluntad”, y Berners-Lee terminó el primer sitio web en diciembre de ese año. Berners-Lee publicó el proyecto en el grupo de noticias alt.hypertext el 7 de agosto de 1991.

**Microsoft .NET**

.NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permite un rápido desarrollo de aplicaciones. Basada en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.

.NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Oracle Corporation y a los diversos framework de desarrollo web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones –o como la misma plataforma las denomina, soluciones– permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows.

.NET Framework se incluye en Windows Server 2008, Windows Vista y Windows 7. De igual manera, la versión actual de dicho componente puede ser instalada en Windows XP, y en la familia de sistemas operativos Windows Server 2003. Una versión "reducida" de .NET Framework está disponible para la plataforma Windows Mobile, incluyendo teléfonos inteligentes.

La norma (incluido en ECMA-335, ISO/IEC 23271) que define el conjunto de funciones que debe implementar la biblioteca de clases base (BCL por sus siglas en inglés, tal vez el más importante de los componentes de la plataforma), define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma independientemente del sistema operativo para el cual haya sido implementada. El Proyecto Mono emprendido por Ximian pretende realizar la implementación de la norma para varios sistemas operativos adicionales bajo el marco del código abierto.

Los principales componentes del marco de trabajo son:

- El conjunto de lenguajes de programación.

- La biblioteca de clases base o BCL.

- El entorno común de ejecución para lenguajes, o CLR (Common Language Runtime) por sus siglas en inglés.

Debido a la publicación de la norma para la infraestructura común de lenguajes (CLI por sus siglas en inglés), el desarrollo de lenguajes se facilita, por lo que el marco de trabajo .NET soporta ya más de 20 lenguajes de programación y es posible desarrollar cualquiera de los tipos de aplicaciones soportados en la plataforma con cualquiera de ellos, lo que elimina las diferencias que existían entre lo que era posible hacer con uno u otro lenguaje.

Algunos de los lenguajes desarrollados para el marco de trabajo .NET son: C#, Visual Basic .NET, Delphi (Object Pascal), C++, F#, J#, Perl, Python, Fortran, Prolog (existen al menos dos implementaciones, el P# y el Prolog.NET), Cobol y PowerBuilder.

**MVC**

Modelo-vista-controlador (MVC) 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.

De manera genérica, los componentes de MVC se podrían definir como sigue:

- El Modelo: Es la representación de la información con la cual el sistema opera, por lo tanto, gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

- El Controlador: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

- La Vista: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto, requiere de dicho 'modelo' la información que debe representar como salida.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo de control que se sigue generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)

- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. Este uso del patrón Observador no es posible en las aplicaciones Web puesto que las clases de la vista están desconectadas del modelo y del controlador. En general el controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista. Por ejemplo, en el MVC usado por Apple en su framework Cocoa. Suele citarse como Modelo-Interface-Control, una variación del MVC más puro

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente...

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos que debe utilizar la aplicación; en líneas generales del MVC dicha gestión corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre la Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque originalmente MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón (ver apartado siguiente "Frameworks MVC"); estos frameworks se diferencian básicamente en la interpretación de como las funciones MVC se dividen entre cliente y servidor.

Los primeros frameworks MVC para desarrollo web planteaban un enfoque de cliente ligero en el que casi todas las funciones, tanto de la vista, el modelo y el controlador recaían en el servidor. En este enfoque, el cliente manda la petición de cualquier hiperenlace o formulario al controlador y después recibe de la vista una página completa y actualizada (u otro documento); tanto el modelo como el controlador (y buena parte de la vista) están completamente alojados en el servidor. Como las tecnologías web han madurado, ahora existen frameworks como JavaScriptMVC, Backbone o jQuery que permiten que ciertos componentes MVC se ejecuten parcial o totalmente en el cliente.

**ASP .NET MVC**

El ASP.NET MVC Framework es un framework de aplicaciones web que implementa el patrón modelo-vista-controlador (MVC).

Basado en ASP.NET, permite a los desarrolladores de software construir una aplicación web como una composición de tres funciones: modelo, vista y controlador.

En marzo de 2009 se hizo pública la primera versión de ASP.NET MVC. El patrón de arquitectura MVC (model-view-controller) no es nuevo (data de 1979) ni es algo que haya inventado Microsoft. Existen muchos frameworks de desarrollo web populares que utilizan MVC, como por ejemplo Ruby on Rails, Spring o Apache Struts. MVC es un patrón de arquitectura que ayuda a crear una separación lógica entre el modelo (información y lógica de negocio), la vista (la lógica de presentación) y el controlador (intermediario entre la vista y el modelo).

Uno de los pilares básicos de ASP.NET MVC es el concepto de enrutamiento (routing), lo que permite a las aplicaciones aceptar peticiones a URL que no se corresponden con ficheros físicos en el servidor. Por ejemplo, en ASP.NET Web Forms las URL tienen el siguiente formato “http://website/products.aspx?category=dvd” en el que físicamente existe un fichero products.aspx en la raíz del sitio web. En MVC la misma URL tendría el siguiente aspecto “http://website/products/dvd” sin que el servidor web necesariamente contenga una carpeta products con una subcarpeta dvd. De forma predeterminada, ASP.NET MVC enruta las peticiones al controlador y a la vista adecuada en función de la URL. Es decir, en el ejemplo anterior, nos devolverá la vista dvd del controlador products.

# Requisitos específicos

**Requerimientos Funcionales**

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF01 |
| **Nombre del Requerimiento:** | Registro Usuarios(Propietarios de restaurante). |
| **Características:** | Los propietarios deberán registrarse en el sistema para acceder a cualquier parte del sistema y gestionar restaurantes. |
| **Descripción del requerimiento:** | El sistema permitirá al usuario con rol de propietario registrarse. El usuario debe suministrar datos como: E-mail, Usuario y Password. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF03 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF02 |
| **Nombre del Requerimiento:** | Autenticación de Usuario(Propietario). |
| **Características:** | Los usuarios deberán identificarse para acceder a cualquier parte del sistema a excepción de los clientes. |
| **Descripción del requerimiento:** | El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF03 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF03 |
| **Nombre del Requerimiento:** | Gestión Restaurantes. |
| **Características:** | Los restaurantes pueden ser agregados, consultados y actualizados |
| **Descripción del requerimiento:** | El sistema ofrecerá al usuario con rol de propietario la opción de añadir y gestionar restaurantes al sistema. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF04 |
| **Nombre del Requerimiento:** | Modificar plantilla Website. |
| **Características:** | El sistema ofrecerá personalización de la página web para cada restaurante o dominio. |
| **Descripción del requerimiento:** | El propietario o gerente tendrán acceso a una plantilla de personalización de la página web. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF05 |
| **Nombre del Requerimiento:** | Autenticación de Usuario(empleado). |
| **Características:** | Los usuarios deberán identificarse para acceder a cualquier parte del sistema de cada restaurante a excepción de los clientes. |
| **Descripción del requerimiento:** | El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF03 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF06 |
| **Nombre del Requerimiento:** | Gestión de platos de la carta. |
| **Características:** | En cada restaurante se podrán gestionar los diferentes productos que serán ofrecidos a los clientes. |
| **Descripción del requerimiento:** | El propietario o gerente tiene la posibilidad de añadir los productos con los siguientes datos: Nombre, Precio, Imagen, Características adicionales. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF07 |
| **Nombre del Requerimiento:** | Gestión de empleados del restaurante. |
| **Características:** | Cada restaurante y dominio tendrá diferentes empleados que pueden ser gestionados desde un módulo de gestión destinado para ello. |
| **Descripción del requerimiento:** | El propietario o gerente tiene la posibilidad de añadir los empleados con los siguientes datos: Email, Nombre, Password y Cargo. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF08 |
| **Nombre del Requerimiento:** | Realizar pedido. |
| **Características:** | Permite a los clientes pedir elementos de la carta que están en la página web. |
| **Descripción del requerimiento:** | El cliente por medio de un carrito de compras virtual puede ordenar lo que desee y será enviado al personal de cocina. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RF09 |
| **Nombre del Requerimiento:** | Gestión de orden de pedido. |
| **Características:** | Permite consultar y actualizar el estado de avance de la orden y los productos ordenados por el cliente. |
| **Descripción del requerimiento:** | El personal de cocina y los camareros tendrán acceso a un panel donde se muestran los productos a elaborar y pueden modificar el estado de avance. |
| **Requerimiento NO funcional:** | * RNF01 * RNF02 * RNF04 |
| **Prioridad del requerimiento:**  Alta | |

**Requerimientos No Funcionales.**

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF01 |
| **Nombre del Requerimiento:** | Interfaz del sistema |
| **Características:** | El sistema presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema. |
| **Descripción del requerimiento:** | El sistema debe tener una interfaz de uso intuitiva y sencilla. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF02 |
| **Nombre del Requerimiento:** | Conexión con BD |
| **Características:** | Diferentes procesos del sistema necesitan conexión con las bases de datos disponibles para el sistema |
| **Descripción del requerimiento:** | La conexión a las bases de datos se realiza mediante el conector de MySQL para .NET |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF03 |
| **Nombre del Requerimiento:** | Encriptación y seguridad de la información |
| **Características:** | Los datos suministrados por los clientes son de alta confidencialidad por lo cual necesitan que se les brinde la seguridad de que su información será bien manejada y estará en un lugar seguro. |
| **Descripción del requerimiento:** | Se utilizara MD5 como sistema de encriptamiento. |
| **Prioridad del requerimiento:**  Alta | |

|  |  |
| --- | --- |
| **Identificación del requerimiento:** | RNF04 |
| **Nombre del Requerimiento:** | Nivel de Perfiles de Usuario |
| **Características:** | Garantizara al usuario el acceso de información de acuerdo al nivel que posee. |
| **Descripción del requerimiento:** | Cada usuario tendrá acceso a las características del sistema que incumben a su cargo. |
| **Prioridad del requerimiento:**  Alta | |

## 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 cómputo en perfecto estado para la gestión del restaurante o de un dispositivo móvil para las demás características.

### Interfaces de software

* Navegador compatible con HTML5 Y CSS3.

### Interfaces de comunicación

Los servidores, clientes 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 (FTP u otros convenientes).