**Técnicas de Elicitación de Requisitos**

Carlos Alexander Manotoa Alava

Enlace a GitHub

TÉCNICAS DE ELICITACIÓN DE REQUISITOS

INTRODUCCIÓN

La elicitación de requisitos es una fase crítica cuyo objetivo cosiste en identificar, recopilar, y documentar las necesidades y expectativas de los stakeholders [1].Para lograr esto se emplean diversas técnicas que facilitan la comunicación y el entendimiento entre los desarrolladores y las partes interesadas [2].

1. Elicitación automatizada basada en datos

La elicitación basada en datos se presenta como una solución frente a los métodos tradicionales, se enfoca en aprovechan fuentes de datos dinámicas y diversas (provenientes de interacciones humanas, procesos organizacionales y dispositivos) para identificar y actualizar requisitos de manera continua. La información incluye retroalimentación explícita e implícita de usuarios, registros de eventos, minería de procesos y datos generados por máquinas, la ventaja de esta técnica está es su capacidad de escalar, automatizar y complementarlos requisitos en tiempo real [3].

La técnica consiste en definir límites para ciertas métricas y problemas relacionados con la calidad del sistema. Si se superan esos límites se activan alertas, estos patrones se convierten automáticamente en requisitos, documentados y priorizados [4].

Otra forma de usar la elicitación automática basada en datos es análisis de reseñas, donde la gran cantidad de información hace inviable un seguimiento manual. Para optimizar esto se proponen técnicas de aprendizaje continuo para entrenar modelos de clasificación, a partir de esta estrategia de muestreo de incertidumbre, se logra que los sistemas aprendan a identificar automáticamente errores, valoraciones y experiencias de usuario con mayor precisión y menor esfuerzo humano [5].

1. Design Thinking

El Design Thinking se utiliza para aborda problemas complejos y desarrollar soluciones centrada en los usuarios. A diferencia de los enfoques tradicionales que se centran en la eficiencia técnica desde una perspectiva lógica [6]. Design Thinking se basa en tres dimensiones: *Deseabilidad* (lo que las personas involucradas necesitan), *factibilidad* (lo que la tecnología puede ofrecer) y *viabilidad* (lo que es económicamente sostenible) [7].

1. Role-Playing (Juego de Roles)

Es una técnica de elicitación de requisitos que consiste en la dramatización de situaciones relacionadas con el sistema, donde los participantes asumen roles concretos para explorar diferentes perspectivas [8]. Mientras técnicas tradicionales como entrevistas y encuestas son más estructuradas, Role Playing destaca en contextos donde la interacción humana es crítica [9].

Esta técnica facilita la comunicación y la comprensión, lo cual es importante para capturar requisitos funcionales u no funcionales [10]. Consta de las siguientes fases de aplicación:

 **Preparación y asignación de roles**: se designan roles como cliente, usuario, analista y diseñador, junto con información sobre el proyecto ficticio [11].

 **Dramatización (ejercicio de role-play)**: los participantes interpretan los roles y llevan a cabo entrevistas individuales y dramatización del proceso de elicitación de requisitos [11].

 **Debate posterior**: al finalizar el ejercicio dramatizado, se realiza un análisis comparativo de técnicas y resultados, tal como se describe en la fase de debate [11].

1. Técnica Delphi

La técnica Delphi es un método sistemático y estructurado para la elicitación de requisitos, que busca alcanzar la conformidad de un grupo de expertos, fue desarrollada con propósitos militares, pero se ha extendido a campos como la ingeniería de requisitos, donde es útil para definir requisitos complejos [12].

En la ingeniería de requisitos, Delphi se emplea para recopilar y refinar requisitos mediante rodas de cuestionarios anónimos, evitando el sesgo como el domino de individuos o la conformidad grupal, permitiendo que las opiniones se basen en el mérito de las ideas y no en influencias jerárquicas [13]. Delphi sigue una estructura rigurosa para garantizar resultado confiables y consensuados [14]:

1. **Selección del panel de expertos:** Se elige un grupo multidisciplinario con conocimientos en área de interés.
2. **Rodas iterativas:** Se plantea una primera ronda de preguntas abiertas que ayudan a explorar ideas. En las posteriores se realizan preguntas cerradas basadas en las respuestas previas donde la retroalimentación es controlada. Así continúan las iteraciones hasta que se logre un consenso entre todas las partes.
3. **Criterios de cierre:** Con anticipación se define un criterio de cierre en el que todas las partes este de acuerdo. Por ejemplo, puede ser un nivel de porcentaje, la disparidad de las preguntas o un número de rondas [15].

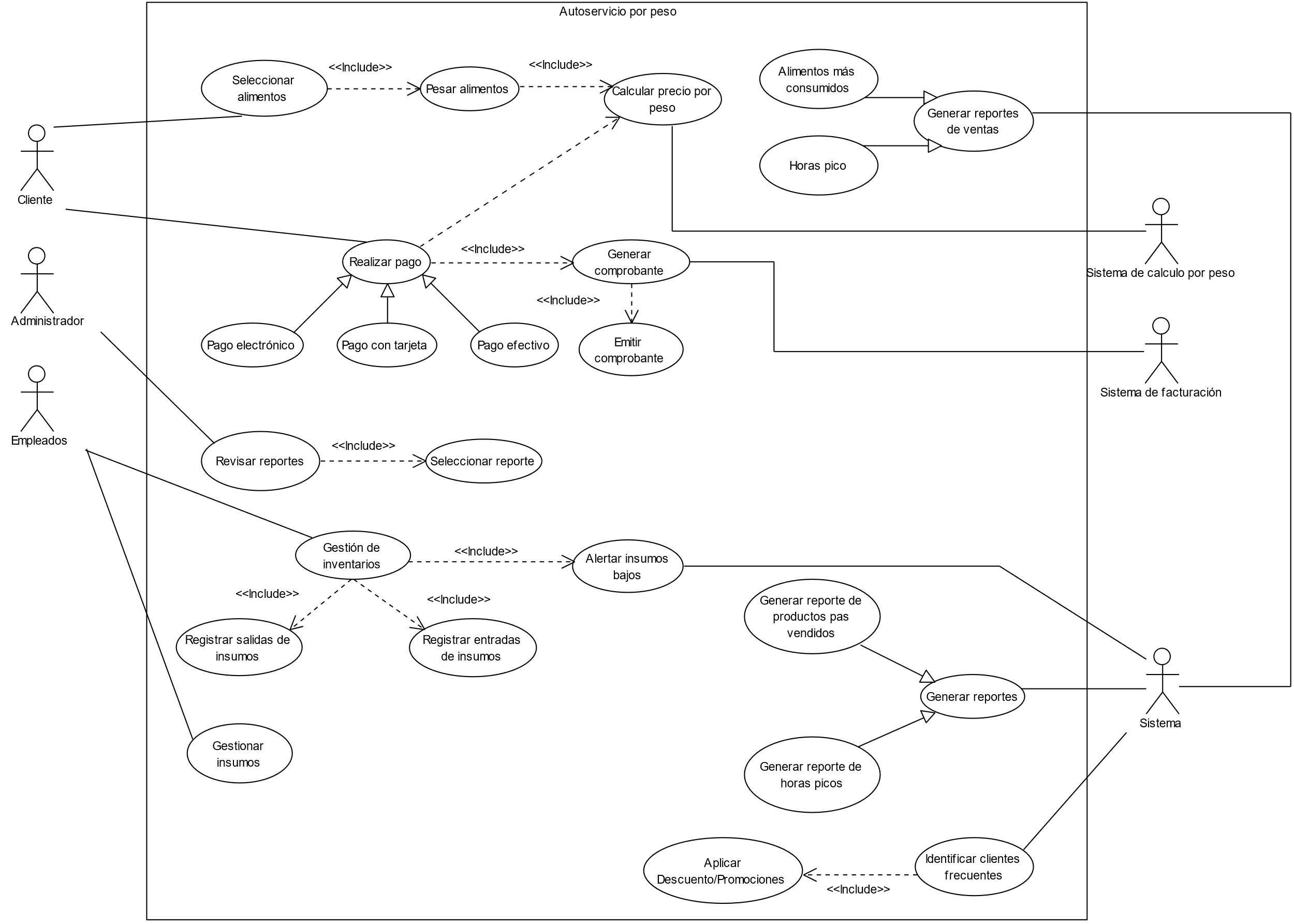
La fortaleza de Delphi se encuentra en su anonimato y ciclo iterativo, que fomenta la participación equitativa y la refinación progresiva de los requisitos. Sin embargo, su éxito depende la calidad del panel y en la definición de consensos

ESCENARIOS DE FLUJO DE INFORMACIÓN DEL SISTEMA DE GESTIÓN DE TITULACIÓN (SGTT)

|  |
| --- |
| **Título y objetivo** |
| **Título:** Solicitar una reunión  **Objetivo:** Automatizar la coordinación entre docentes y alumnos mediante una funcionalidad integrada en el SGTT, garantizando la trazabilidad del proceso. |
| **Actores y roles** |
| **Actores principales:** Alumnos.  **Actores secundarios:** Docentes, coordinadores, personal administrativo. |
| **Precondiciones** |
| * Solo se puede solicitar reuniones entre usuarios que formen parte de un mismo proceso. * Los horarios de las reuniones deben ser en horarios laborales. * Los lugares de encuentro deben ser en las instalaciones de la Universidad Técnica de Quevedo. * No se puede registrar más de una reunión en el mismo horario. * Los horarios no pueden solaparse, y deben tener un mínimo de tiempo de diferencia entre cada una. |
| **Flujo principal (Caso de éxito)** |
| 1. El alumno accede al módulo de “Solicitar nueva reunión” en el SGTT. 2. El sistema muestra una nómina de usuarios asociados a los que puede solicitar una reunión. 3. El alumno selecciona un docente a quién le desea solicitar la reunión. 4. El sistema muestra un calendario con los horarios disponibles del docente. 5. El alumno selecciona un horario disponible. 6. El sistema muestra un formulario con los detalles de la reunión:  * Razón de la reunión * Modalidad de la reunión (Presencial/Virtual) * Lugar de encuentro (En caso de ser presencial)  1. El sistema notifica al docente sobre la solicitud de una reunión. 2. El docente confirma la reunión en el horario, lugar y modalidad establecida. 3. El sistema registra la reunión en el calendario y actualiza la disponibilidad de los usuarios involucrados. 4. El sistema notifica a los usuarios involucrados de que se ha registrado con éxito la reunión. |
| **Flujo alternativo (Docente posterga la reunión)** |
| 1. El alumno accede al módulo de “Solicitar nueva reunión” en el SGTT. 2. El sistema muestra una nómina de usuarios asociados a los que puede solicitar una reunión. 3. El alumno selecciona un docente a quién le desea solicitar la reunión. 4. El sistema muestra un calendario con los horarios disponibles del docente. 5. El alumno selecciona un horario disponible. 6. El sistema muestra un formulario con los detalles de la reunión:  * Razón de la reunión * Modalidad de la reunión (Presencial/Virtual) * Lugar de encuentro (En caso de ser presencial)  1. El sistema notifica al docente sobre la solicitud de una reunión. 2. El docente proponer posponer la reunión en un nuevo horario. 3. El sistema notifica al alumno sobre la solicitud de posponer la reunión. 4. El alumno acepta el nuevo horario propuesto. 5. El sistema registra la reunión pospuesta en el calendario y actualiza la disponibilidad de los usuarios involucrados. 6. El sistema notifica a los usuarios involucrados, que el registro de la reunión pospuesta fue exitoso. |
| **Flujo excepcional** |
| 1. El alumno accede al módulo de “Solicitar nueva reunión” en el SGTT. 2. El sistema muestra una nómina de usuarios asociados a los que puede solicitar una reunión. 3. El alumno selecciona un docente a quién le desea solicitar la reunión. 4. El sistema detecta que el docente tiene permiso médico y no está disponible. 5. El sistema informa de la situación a coordinación. 6. Coordinación asigna un docente de remplazo temporal para la reunión. 7. Se notifica al estudiante sobre el remplazo. 8. El alumno acepta la reunión con el docente de reemplazo. 9. El alumno ingresa los detalles de la reunión  * Razón de la reunión * Modalidad de la reunión (Presencial/Virtual) * Lugar de encuentro (En caso de ser presencial)  1. El reemplazo acepta la reunión en los términos establecidos. 2. El sistema registra la reunión en el calendario y actualiza la disponibilidad de los usuarios involucrados. 3. El sistema notifica a los usuarios involucrados de que se ha registrado con éxito la reunión. |
| **Postcondiciones** |
| * Luego de la confirmación de una reunión, la disponibilidad de los usuarios debe actualizarse automáticamente. * La reunión confirmada debe registrarse junto con los detalles. * Se debe notificar el correcto registro de la reunión. |

|  |
| --- |
| **Título y objetivo** |
| **Título:** Sistema de mensajería.  **Objetivo:** Centralizar la comunicación de las partes interesadas, mediante un sistema de mensajería interno, eliminando el uso de plataformas informales y mejorando la trazabilidad. |
| **Actores y roles** |
| **Actores principales:** Alumnos.  **Actores secundarios:** Docentes, coordinadores, personal administrativo. |
| **Precondiciones** |
| * Solo se puede enviar mensajes entre usuarios que formen parte de un mismo proceso. * Los usuarios deben aceptar las de tratamiento de datos personales. |
| **Flujo principal (Caso de éxito)** |
| 1. El estudiante accede al módulo “Mensajería” en el SGTT. 2. El sistema muestra una lista de contactos disponibles. 3. El estudiante selecciona un docente. 4. El estudiante escribe el mensaje y puede adjuntar documentos. 5. El sistema valida que el formato del documento sea válido. 6. El sistema registra el mensaje con detalles como la fecha y hora. 7. El sistema notifica en tiempo real al docente. 8. El docente recibe la notificación, abre el mensaje y responde. 9. El sistema actualiza la conversación y notifica al estudiante. 10. Cada mensaje se almacena en las historias de mensajería. |
| **Flujo alternativo (Docente posterga la reunión)** |
| 1. El estudiante accede al módulo “Mensajería” en el SGTT. 2. El sistema muestra una lista de contactos disponibles. 3. El estudiante selecciona un docente. 4. El estudiante escribe el mensaje y puede adjuntar documentos. 5. El sistema valida que el formato del documento sea válido. 6. Si se adjunta un formato no permitido, el sistema no permite enviar el mensaje y muestra un mensaje de error. 7. El usuario corrige el formato del documento y envía el mensaje. |
| **Flujo excepcional** |
| 1. El estudiante accede al módulo “Mensajería” en el SGTT. 2. El sistema muestra una lista de contactos disponibles. 3. El estudiante selecciona un docente. 4. El estudiante escribe el mensaje y puede adjuntar documentos. 5. El sistema valida que el formato del documento sea válido. 6. El sistema registra el mensaje con detalles como la fecha y hora. 7. El sistema notifica en tiempo real al docente. 8. Si el docente no responde en 24 horas el sistema reenvía una notificación. 9. Para mensajes urgentes, coordinación también es notificado. |
| **Postcondiciones** |
| * Los mensajes son registrados en el historial con: * Remitente * Destinatario * Fecha * Hora * Documentos adjuntos * Se notifica a las partes interesadas. |

DIAGRAMA DE CASO DE USO: AUTOSERVICIO DE COMIDA POR PESO



HISTORIAS DE USUARIO: AUTOSERVICIO DE COMIDA POR PESO

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario: Identificar clientes frecuentes para descuentos** | | |
| **CCC** | **Elementos** | **Descripción** |
| Carta | ¿Quién?  ¿Qué?  ¿Para qué? | Como **cajero del autoservicio.**  Quiero **identificar rápidamente a un cliente frecuente** al escanear su ID.  Para **aplicar descuentos y promociones,** mejorando su experiencia. |
| Conversación | Descripción | El sistema debe identificar un cliente frecuente al escanear su ID, notificar a los cajeros y aplicar descuentos y promociones automáticamente. |
| Confirmación | Criterios de aceptación | * Reconocer los usuarios frecuentes al escanear el ID. * Si es un cliente frecuente se aplican las promociones y descuentos automáticamente. |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario: Generar reporte de ventas diarias** | | |
| **CCC** | **Elementos** | **Descripción** |
| Carta | ¿Quién?  ¿Qué?  ¿Para qué? | Como **generen del autoservicio.**  Quiero un **reporte diario de ventas** por categoría de alimentos.  Para **analizar** tendencias y ajustar el menú. |
| Conversación | Descripción | El sistema debe generar un reporte con datos de la venta como:   * Total de ventas. * Alimentos más vendidos. * Horas picos de ventas. |
| Confirmación | Criterios de aceptación | * Reconocer los usuarios frecuentes al escanear el ID. * Si es un cliente frecuente se aplican las promociones y descuentos automáticamente. |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario: Alertar sobre stock bajo de insumos** | | |
| **CCC** | **Elementos** | **Descripción** |
| Carta | ¿Quién?  ¿Qué?  ¿Para qué? | Como **cocinero.**  Quiero **recibir una alerta cuando un alimento este por agotarse.**  Para **reponer insumos** a tiempo y no tener cortes durante el día. |
| Conversación | Descripción | El sistema deberá notificar al personal cuando un insumo este por agotarse. |
| Confirmación | Criterios de aceptación | * La alerta debe notificar de un insumo escaso. * La alerta debe contener el nombre. * La alerta debe contener la cantidad restante. |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario: Emitir comprobante en línea** | | |
| **CCC** | **Elementos** | **Descripción** |
| Carta | ¿Quién?  ¿Qué?  ¿Para qué? | Como **cliente**  Quiero **recibir un comprobante** digital de mi compra.  Para tener un registro. |
| Conversación | Descripción | El sistema debe generar un comprobante digital de la compra. |
| Confirmación | Criterios de aceptación | * La facturación debe regirse a normativas legales. * La factura debe contener toda la información necesaria. * La factura debe ser enviado al cliente |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario: Pesar plato y calcular el precio automáticamente** | | |
| **CCC** | **Elementos** | **Descripción** |
| Carta | ¿Quién?  ¿Qué?  ¿Para qué? | Como **cliente** del autoservicio**.**  Quiero que el **sistema pese y calcule el precio total** automáticamente.  Para saber **cuento debo pagar** de manera rápida y precisa, sin errores manuales. |
| Conversación | Descripción | El sistema calcular el precio de los alimentos automáticamente, pesando el plato. |
| Confirmación | Criterios de aceptación | * El cálculo debe ser preciso. * Mostrar información del peso y el precio por separado. * Debe integrarse con el sistema de facturación. |

BIBLIOGRAFíAS

[1] E. Siakas, H. Rahanu, J. Loveday, E. Georgiadou, K. Siakas, and M. Ross, “Managing Ethical Requirements Elicitation,” in *Systems, Software and Services Process Improvement*, M. Yilmaz, P. Clarke, A. Riel, and R. Messnarz, Eds., Cham: Springer Nature Switzerland, 2023, pp. 258–272.

[2] D. Zowghi and C. Coulin, “Requirements Elicitation: A Survey of Techniques, Approaches, and Tools,” in *Engineering and Managing Software Requirements*, A. Aurum and C. Wohlin, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 19–46. doi: 10.1007/3-540-28244-0\_2.

[3] S. Lim, A. Henriksson, and J. Zdravkovic, “Data-Driven Requirements Elicitation: A Systematic Literature Review,” *SN Comput Sci*, vol. 2, no. 1, p. 16, 2021, doi: 10.1007/s42979-020-00416-4.

[4] M. Oriol *et al.*, “Data-driven and tool-supported elicitation of quality requirements in agile companies,” *Software Quality Journal*, vol. 28, no. 3, pp. 931–963, 2020, doi: 10.1007/s11219-020-09509-y.

[5] V. T. Dhinakaran, R. Pulle, N. Ajmeri, and P. K. Murukannaiah, “App Review Analysis Via Active Learning: Reducing Supervision Effort without Compromising Classification Accuracy,” in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, 2018, pp. 170–181. doi: 10.1109/RE.2018.00026.

[6] H. Ferreira Martins, A. de Oliveira Junior, E. Dias Canedo, R. A. Dias Kosloski, R. Ávila Paldês, and E. Costa Oliveira, “Design Thinking: Challenges for Software Requirements Elicitation,” *Information*, vol. 10, no. 12, 2019, doi: 10.3390/info10120371.

[7] D. M. Jennifer Hehn, Walter Brenner, and Manfred Broy, Eds., *Design Thinking for Software Engineering*. Springer Cham, 2022.

[8] D. Zowghi and S. Paryani, “Teaching requirements engineering through role playing: lessons learnt,” in *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.*, 2003, pp. 233–241. doi: 10.1109/ICRE.2003.1232754.

[9] Javier Ferrer, Elena M. Cortés, and María Aragonés, “Aplicación de la técnica Role-Playing,” pp. 1–19, Jan. 2025, doi: https://doi.org/10.31637/epsir-2025-1329.

[10] Daryelin Blackman and Aleojin Rios, “Role-Playing as a communicative strategy,” 2025, pp. 65–78. doi: https://doi.org/10.48204/rea.v4n1.7291.

[11] S. Raghavan, G. Zelesnik, and G. Ford, “Educational Materials Lecture Notes on Requirements Elicitation Software Engineering Institute,” 1994.

[12] Norman Dalkey and Olaf Helmer, “An Experimental Application of the DELPHI Method to the Use of Experts,” 1936.

[13] J. P. Fefer, S. De-Urioste Stone, J. Daigle, and L. Silka, “Using the Delphi Technique to Identify Key Elements for Effective and Sustainable Visitor Use Planning Frameworks,” *Sage Open*, vol. 6, no. 2, Apr. 2016, doi: 10.1177/2158244016643141.

[14] P. Nasa, R. Jain, and D. Juneja, “Delphi methodology in healthcare research: How to decide its appropriateness,” *World J Methodol*, vol. 11, no. 4, pp. 116–129, Jul. 2021, doi: 10.5662/wjm.v11.i4.116.

[15] Jünger S, Payne SA, Brine J, Radbruch L, and Brearley SG, “Guidance on Conducting and REporting DElphi Studies (CREDES) in palliative care: Recommendations based on a methodological systematic review.,” vol. 31, no. 8, Feb. 2017.