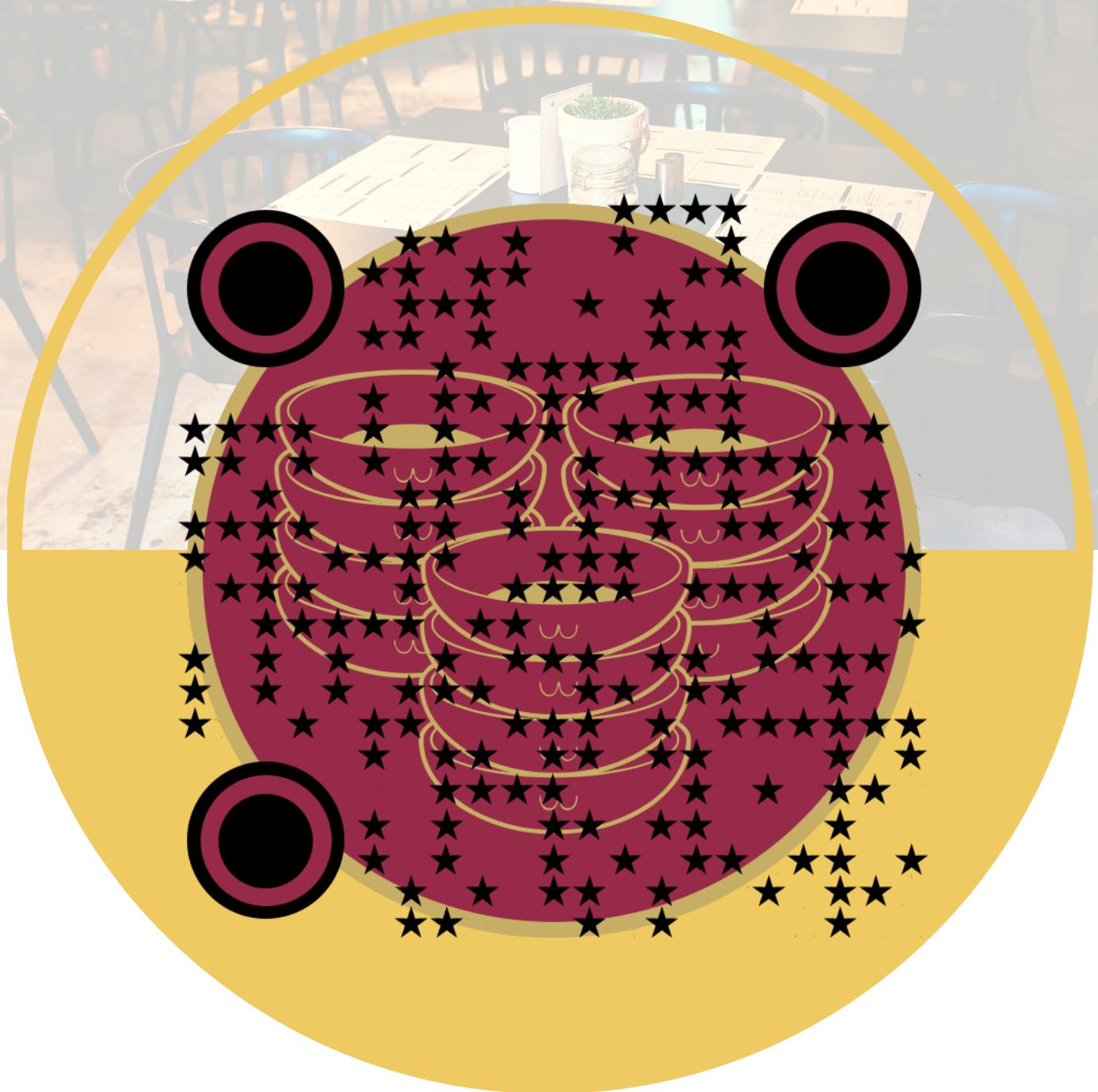


# QR RESTAURANT



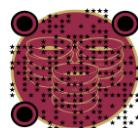
Francisco Almenara Parias

Manuel Cortés Serrano

Cristian Cosano Cejas

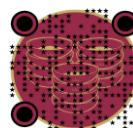
Jose Manuel Gil Rodríguez

Antonio Luis Rodríguez Jiménez



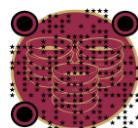
# Índice

<b>PRÁCTICA 1</b>	3
<b>Introducción</b>	3
Definición del problema	4
<b>Alcance del sistema</b>	5
Para qué entorno estamos haciendo la aplicación web.	5
¿Quién es el público que va a utilizar la aplicación web?	5
Resumidamente y formal, ¿en qué va a consistir?	5
¿Qué propone la aplicación web?	6
<b>Objetivos del problema</b>	6
Objetivos para el público	6
Objetivos técnicos	7
<b>Antecedentes</b>	7
<b>Recursos</b>	8
Recursos humanos	8
Recursos hardware	8
Recursos software	8
<b>Reparto de tareas</b>	10
<b>Metodología Ágil. SCRUM.</b>	10
<b>PRÁCTICA 2</b>	12
<b>Captura de requisitos</b>	12
Entrevista con el cliente.	13
<b>Definición de requisitos</b>	13
Requisitos funcionales	13
Restaurante	13
Usuario	14
Requisitos no funcionales	15
Requisitos de información	15
Glosario de términos	16



---

<b>Validación de requisitos</b>	18
Walk-throughs	18
Auditoría	18
Sketches o bocetos	18
Matriz de trazabilidad	19
<b>Requisitos modelados como Casos de Uso</b>	20
Identificación de los casos de uso	20
Diagrama de casos de uso	21
<b>Requisitos modelados como actividades</b>	22
registrarUsuario	22
añadirRestaurante	22
editarRestaurante	22
eliminarRestaurante	23
iniciarSesión	23
registrarAcompañante	23
verMisRestaurantes	24
valorarRestaurante	25
filtrarRestaurante	25
filtrarPorComidaRestaurante	26
filtrarPorValoración	26
filtrar Restaurante Cercano	27
filtrarPorNombreRestaurante	27
mostrarInformaciónRestaurante	28
mostrarHistorial	28
registrarSalida	29
entrarColaEspera	29
mostrarRestaurantesAlternativos	29
registrarEntrada	30
restringirEntrada	30



## PRÁCTICA 1

### Introducción

La hostelería es uno de los sectores más importantes de nuestro país. Sin embargo, el actual contexto COVID-19 obliga a establecer protocolos para que la reapertura de las instalaciones no aumente el riesgo de contagio comunitario, así como establecer las medidas de protección necesarias para las personas trabajadoras en este sector.

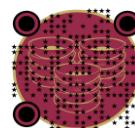
La nueva normalidad tras la pandemia, conlleva tener que modificar ciertos hábitos de consumo que antes de la crisis sanitaria no estaban tan implementados en la sociedad. Al tiempo que el incremento del teletrabajo va normalizando el uso de nuevas herramientas digitales, en ámbitos como la hostelería van creciendo el uso de otros mecanismos, como los códigos QR.

En todas las actividades se deben respetar las distancias de seguridad interpersonal. Para ello y cuando sea necesario, se deberá realizar el correspondiente control de aforos por parte del personal del establecimiento.

Esta herramienta supone una solución para múltiples negocios ya que permite digitalizar su oferta. Los códigos QR funcionan como una especie de escáner a través de la cámara de los dispositivos que permiten fotografiar el código, que el móvil leerá gracias a una aplicación. De este modo se aporta un soporte complementario a las labores del restaurante, gestionando el control de la capacidad máxima en el exterior e interior del local y el porcentaje de uso y aforo permitido según la normativa vigente.

El usuario que escaneé el código, en caso de que no haya aforo disponible, será informado de la disponibilidad del aforo mediante la interfaz de su teléfono móvil.

Es importante señalar, que nuestro sistema se va comunicar con un sistema externo, encargado del control de un mecanismo de torniquete retráctil automático, para el acceso al establecimiento.

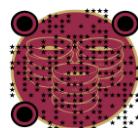


## Definición del problema

Para conocer la necesidad que ha dado lugar a este proyecto, se ha expuesto en la introducción la situación previa de la empresa, que es lo que hizo surgir tal necesidad. En este punto, procedemos a desarrollar la actividad de la empresa y los aspectos de interés de cara a la realización del proyecto. Se plantea el problema detallando todos los aspectos que están especificados, para finalmente seleccionar la tecnología de programación que se usará en nuestra aplicación Web.

QrRestaurant contará con las siguientes funcionalidades:

- Cada **establecimiento** (restaurante, bar, cafetería, etc.) que se adhiera tendrá su apartado, que contará con información del mismo, como su ubicación, detalles de interés, reseñas, etc.
- Cada **establecimiento** aparecerá en el mapa incorporado en QrRestaurant.
- Cada **usuario** que se registre en la aplicación contará con un código QR único.
- Cada **usuario** que desee entrar al establecimiento, deberá hacer uso de su código QR, que le dará acceso si hay disponibilidad de aforo.
- Cada **usuario**, deberá indicar el número de acompañantes con los que accede al establecimiento, que no puedan identificarse por sí mismos, antes de entrar (es decir, en el caso de niños y personas que no dispongan de móvil).
- Cada **usuario**, al abandonar el establecimiento deberá volver a hacer uso del código QR para adaptar el aforo que gestiona el sistema.
- Cada **usuario** debe poder calificar el restaurante y escribir reseñas, si así lo desea.
- La **aplicación** debe contar con un historial para los establecimientos, que almacenará el número de clientes que estuvieron en el restaurante, duración media, etc.
- La **aplicación** mostrará qué establecimientos cuentan con aforo disponible y cuales están llenos.
- La **aplicación** mostrará los establecimientos más cercanos a la ubicación del usuario y el tipo de comida que ofrecen.



## Alcance del sistema

- **Para qué entorno estamos haciendo la aplicación web**

La aplicación está enfocada para dar soporte a un conjunto de restaurantes, para controlar los aforos por la situación derivada de excepcionalidad sanitaria.

Es decir, su funcionalidad principal es el control del público en el sector servicios, por lo que está diseñada para el entorno profesional de la hostelería.

- **¿Quién es el público que va a utilizar la aplicación web?**

La aplicación web está orientada para ser usada por los usuarios, es decir, las personas que asistan a uno de los bares a los que la aplicación da soporte.

Un aspecto a resaltar es que el usuario deberá indicar si está o no acompañado por niños o personas que no puedan identificarse, ya que como es evidente, estos no tienen acceso a dispositivos con los que registrar su asistencia.

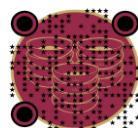
- **Resumidamente y formal, ¿en qué va a consistir?**

Las personas asistentes a uno de los bares que posee el control de acceso proporcionado por la aplicación web, deberá escanear el código. Este código contiene un control de acceso, es decir, si el bar está lleno, indicará al usuario que no puede acceder. Si por el contrario hay hueco en el establecimiento, registrará los datos de los asistentes y permitirá que estos entren.

La aplicación tendrá la opción de marcar la entrada de una o más personas (en caso de niños que no tienen dispositivos móviles).

De igual forma, para salir, la persona debe volver a escanear el mismo código QR, indicando también si está con niños.

Adicionalmente, la aplicación tendrá un apartado para incluir un historial del restaurante, es decir, durante los días anteriores, cuántas



---

personas hubo en el restaurante, la duración media de la persona en el mismo, etc. Y otro apartado para incluir un listado de restaurantes que no estén llenos o que todavía tengan aforo. También incluye la distancia a la que está el restaurante o la información del tipo de comida que sirve.

- **¿Qué propone la aplicación web?**

- **Controlar el aforo de un establecimiento mediante la utilización de códigos QR:** Un código QR es un sistema para almacenar información en una matriz de puntos parecida a un código de barras.

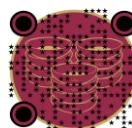
Las ventajas de utilizar códigos QR es que es una herramienta digital muy creativa que permite y acelera la utilización de servicios web en móviles. Se deben leer con un teléfono móvil o con un dispositivo que disponga de la aplicación correspondiente (un lector de códigos QR). No son legibles para el ojo humano. La lectura se lleva a cabo en cuestión de segundos, aún si faltara alguna pieza en el código.

- **Ofrece una gran oportunidad a un restaurante para establecer un vínculo con sus clientes incluso antes de que lleguen a sentarse a la mesa:** Es decir, que un/a usuario vea la actividad de un restaurante en alguna red social y le despierte el interés por pedir una mesa o acercarse es una ventaja indiscutible.

## **Objetivos del problema**

### **Objetivos para el público.**

- Controlar el **aforo** de los restaurantes con la mínima intervención humana para prevenir contagios.
- Facilitar la **búsqueda** por parte de los usuarios de restaurantes cuyo aforo aún no está completo para ahorrar tiempo.
  - Los usuarios deben poder buscar restaurantes que estén disponibles en el momento de la consulta.



- 
- Se debe permitir consultar el historial de aforo del restaurante buscado, para que los clientes consideren si es seguro o no, según su criterio (si tiene menos aforo, será más seguro)
  - Los clientes deben poder valorar su estancia en el restaurante, para que otros usuarios puedan conocer a través de sus opiniones la actuación del restaurante respecto a las medidas sanitarias.
  - Se deben poder ordenar los restaurantes por valoración, facilitando la búsqueda de los mejores restaurantes para los clientes.
  - Se deben poder encontrar los restaurantes más cercanos según la ubicación del cliente.

## Objetivos técnicos

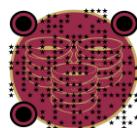
- Utilizar la herramienta **Heroku** como servidor web, ya que facilita el despliegue del código.

## Antecedentes

La hostelería es uno de los sectores más afectados (o tal vez el que más) por la crisis económica derivada por la pandemia del Covid. Este sector ha sido duramente castigado y apenas ha recibido ayudas, por lo que desde QrRestaurant queremos ayudar a proporcionar métodos para garantizar la seguridad en el consumo de estos establecimientos.

En breve nos enfrentaremos al periodo vacacional, el más importante para este sector y su principal fuente de ingresos, por eso necesitamos aportar soluciones. Desde QrRestaurant podemos ayudar a gestionar el control de los aforos, que actualmente supone el mejor método de seguridad y además es una medida exigida por las autoridades.

Para desarrollar, en QrRestaurant nos hemos fijado en la aplicación mundialmente conocida Google Maps. Con el paso del tiempo, se ha ido convirtiendo en una aplicación demasiado extensa, que proporciona una cantidad de información desmesurada, por tanto, nuestra intención es la de utilizar la base de Google Maps y adaptarla a nuestro ámbito de aplicación, la hostelería.



---

Permitiremos que los restaurantes se registren en nuestra aplicación de una forma muy parecida a tripadvisor y utilizaremos los mapas de google maps para que los usuarios puedan conocer la ruta para llegar al destino seleccionado. Todo esto sin perder la esencia de la aplicación, que es buscar restaurantes que cumplen la normativa covid y que podrán autorizar la entrada con los QR de nuestra aplicación.

Por último, todo el mundo conoce la potencia de los códigos QR. En nuestro caso utilizaremos esta tecnología para proporcionar un QR único para cada cliente, que contenga la información necesaria y le permita entrar a los establecimientos siguiendo un control.

## **Recursos**

### **Recursos humanos**

Para el desarrollo de este proyecto software contamos con un equipo especializado en el ámbito, formado por grandes mentes contemporáneas influyentes en el panorama nacional, que actualmente se encuentran progresando en sus estudios en ingeniería software en la universidad de Córdoba. Estamos hablando de profesionales en potencia como Francisco Almenara, Manuel Cortés, Cristian Cosano, Jose Manuel Gil y Antonio Luis Rodriguez. El departamento que conforma el equipo trabajará con esfuerzo y entusiasmo para garantizar al cliente la máxima calidad del trabajo.

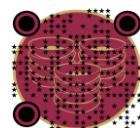
### **Recursos hardware**

Los equipos tecnológicos con los que se trabaja y su capacidad.

Con respecto al hardware del proyecto, hemos personalizado y configurado, hablando de cloud computing, un PaaS llamado Heroku que nos permite un despliegue continuo del sistema. Por lo que se desprende, que será Heroku el recurso hardware que usado en este proyecto.

### **Recursos software**

Para alojar la aplicación hemos utilizado una instancia gratuita en Heroku. Una plataforma como servicio que nos da la posibilidad de alojar nuestra aplicación en sus servidores y mantenerla mediante despliegue continuo a través de un repositorio de código. Esta plataforma nos facilita el desarrollo



---

ya que no tenemos que encargarnos de ninguna configuración del servidor ni del sistema operativo.

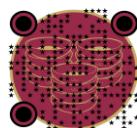
Para los recursos software que vamos a utilizar en este proyecto hemos tenido en cuenta las capacidades y las tecnologías que conocemos cada uno de los integrantes de este grupo.

Después de una larga exposición de las tecnologías que dominamos cada uno hemos decidido que la mejor forma de implementar esta aplicación es con JavaScript, tanto en el lado del servidor como en el lado del cliente. Esto nos facilitará el desarrollo, ya que vamos a usar un mismo lenguaje para toda la aplicación.

Por otro lado, contaremos con una base de datos en MySQL ya que todos hemos trabajado con este sistema de gestión en otras asignaturas.

Como repositorio de código hemos utilizado el sitio web GitHub, que nos permitirá tener nuestro código centralizado y poder realizar cambios cada componente del grupo. Esta herramienta facilitará el trabajo ya que siempre tendremos la última versión del código actualizado.

Para interactuar con el repositorio haremos uso del software de control de versiones la herramienta Git mediante terminal Git Bash que proporciona. Desde ahí subiremos los cambios, bajaremos la nueva versión de código, podremos deshacer cambios y volver a estados anteriores, etc.



## Reparto de tareas

En este grupo de desarrollo contamos con 5 integrantes. Antes de repartir las tareas hemos puesto sobre la mesa las capacidades de cada participante y lo que podrá desarrollar con mejor calidad, pensando en todo momento obtener el mejor resultado. Así, una vez planteado esto, el reparto de tareas ha quedado de la siguiente forma:

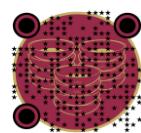
- **Francisco Almenara Parias.** Se encarga del frontend y del desarrollo de la aplicación, concretamente de la creación y el estilo de la web.
- **Manuel Cortés Serrano.** Se encargará tanto del frontend como del backend.
- **Cristian Cosano Cejas.** Se encargará del backend de la aplicación y del manejo del servidor web de la instancia.
- **Jose Manuel Gil Rodríguez.** Se encargará del frontend de la aplicación y de la creación de la base de datos. También ha realizado el logo de QrRestaurant.
- **Antonio Luis Rodríguez Jiménez.** Se encargará del backend de la aplicación y será el Scrum Master del equipo.

## Metodología Ágil. SCRUM.

Como equipo hemos tomado la decisión de que una metodología ágil será lo más idóneo y colaborativo para trabajar el desarrollo del proyecto, debido a que permite que este sea iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Además, debido a la situación actual, es un punto a favor.

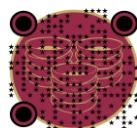
Concretamente, hemos elegido la metodología Scrum. Canvas sería otra opción muy válida pero debido a que Scrum es para nosotros un marco de trabajo más familiar, aportará mayores beneficios.

Para ello utilizaremos Taiga, que es una aplicación asociada con github, que permite organizar proyectos colaborativos con la metodología Scrum. Del mismo modo, se ha tomado la decisión de otorgar el papel de Scrum Master a Antonio Luis Rodriguez, ya que los integrantes del grupo hemos considerado por experiencias anteriores, será el más idóneo para emplear esta responsabilidad por sus grandes actuaciones como líder en el pasado. El resto



---

del equipo serán distribuidos en diversos campos como back, front, UX. Sin embargo, a pesar de que cada participante tenga determinada su tarea, será obligación de todo el equipo compartir tareas comunes y socorrer a aquel compañero que lo precise.



## PRÁCTICA 2

### Captura de requisitos

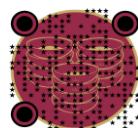
Control de accesos por QR a restaurantes. Debido a la pandemia ocasionada por el COVID, los códigos QR han ganado mucho interés, ya que son casi infinitos y tienen una facilidad de uso enorme.

En más de una ocasión, hemos pensado que hay bares o restaurantes que están llenos y que exceden el aforo máximo establecido por las autoridades competentes. Para ello, este proyecto implementará un sistema en el que los restaurantes, por medio del empresario o socio principal, se darán de alta en la aplicación web, generando un código QR, que estará siempre asociado al restaurante. Además, también existirá un apartado de información sobre el restaurante (o cafetería, bar, etc.), ubicación (con su respectivo mapa) y cualquier detalle que nos parezca de interés.

Los usuarios, personas que asistan a un bar, deben escanear el código, que tendrá un control de acceso, es decir, si está lleno, no se podrá acceder al bar, mientras que si hay hueco, permitirá entrar a la persona. Debe estar la opción de marcar la entrada de una o más personas (en caso de niños que no tienen dispositivos móviles). De igual forma, para salir, la persona debe volver a escanear el mismo código QR, indicando también si está con niños.

Además, la misma aplicación debe tener un apartado para incluir un historial del restaurante, es decir, durante los días anteriores, cuántas personas hubo en el restaurante, la duración media de la persona en el mismo, etc. Y otro apartado para incluir un listado de restaurantes que no estén llenos o que todavía tengan aforo. También podemos incluir información de la distancia a la que está el restaurante o del tipo de comida que sirve.

Por último, una persona que haya estado en un restaurante, debe poder poner opiniones si así lo desea, por lo que también habrá que incluir una sección que muestre los restaurantes por calidad (acorde a las opiniones de las personas).



## Entrevista con el cliente

Para entender el problema y comprender los objetivos a cumplir hacemos uso de esta técnica, siguiendo una estructura marcada, tienen una estructura. Tras la identificación previa de los entrevistados. Y una vez preparada la entrevista, procedemos a su realización.

De esta, se han documentado los resultados:

El cliente convino con nuestro punto de vista, y certificó como positivo el hecho de que sería más productivo que cada usuario registrado en la aplicación tuviese su propio código. Además, se amplió el conocimiento del sistema pues nos especificó una idea antes no concebida, el sistema debería controlar el acceso a través de QR pero también la salida, asimismo debería existir un control de los menores de edad que entren acompañados por adultos.

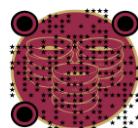
## Definición de requisitos

Especificación, a nivel más formal de los requisitos a satisfacer en el sistema web.

### Requisitos funcionales

#### **Restaurante**

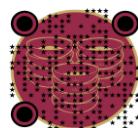
- **RF1** - Cada restaurante debe incluir un historial que muestra estadísticas de los días anteriores.
- **RF2** - El sistema debería prohibir la entrada a un usuario que intenta acceder a un local que supera el aforo.
- **RF3** - El sistema debería indicar al usuario una lista con los restaurantes disponible más cercano en caso de que el restaurante actual tuviera el aforo completo.
- **RF4** - El sistema debería mostrar la distancia a la que se encuentra el restaurante
- **RF5** - El restaurante debería incorporar la carta de comida que se sirve.



- 
- **RF6** - El sistema debería permitir listar los restaurantes por calidad acorde a las opiniones de los usuarios.
  - **RF7** - El sistema debería mostrar una ficha con la información del restaurante.
  - **RF8** - El sistema debería restringir la salida del restaurante hasta que se pase el QR por el lector.
  - **RF9** - El sistema debería de contar con un filtro que muestre los restaurantes con aforo disponible.
  - **RF10** - El sistema debería de contar con filtro que muestre los restaurantes por tipo de comida.
  - **RF11** - El sistema debería mostrar en el mapa la ubicación del restaurante.
  - **RF12** - Cada restaurante debería incluir una lista de espera con usuarios en cola.
  - **RF13** - El administrador del restaurante debe poder añadir restaurantes.
  - **RF14** - El administrador del restaurante debe poder editar restaurantes.
  - **RF15** - El administrador del restaurante debe poder eliminar el restaurante.

## Usuario

- **RF16** - El sistema debería generar un código QR por cada usuario registrado.
- **RF17** - El sistema debería permitir que cada usuario pueda indicar cuando entra a un restaurante si va acompañado de menores.
- **RF18** - El sistema debería permitir poner en cola a un usuario que intenta acceder a un local que supera el aforo.
- **RF19** - El sistema debería permitir a los usuarios añadir reseñas y valorar los restaurantes que visite.
- **RF20** - El usuario no registrado debe poder registrarse en el sistema.
- **RF21** - El usuario debe poder iniciar sesión en el sistema.



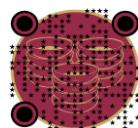
- 
- **RF22** - El sistema debe permitir al usuario filtrar por valoración.
  - **RF23** - El sistema debe permitir buscar restaurantes por ubicación cercana.
  - **RF24** - El sistema debe permitir al administrador ver sus restaurantes.

## Requisitos no funcionales

- **RNF1** - El front del sistema debería estar desarrollado haciendo uso de HTML + CSS + JS
- **RNF2** - El back del sistema debería estar desarrollado en el lenguaje Javascript
- **RNF3** - El sistema debería funcionar siempre.
- **RNF4** - El sistema debería ser responsive
- **RNF5** - El sistema debería permitir generar códigos QR.
- **RNF6** - Para cargar los mapas usaremos la API de GMaps.

## Requisitos de información

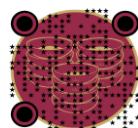
- **RI1** - El sistema debería almacenar los siguientes datos del usuario registrado: nombre, contraseña, email, DNI.
- **RI2** - El sistema debe almacenar del restaurante la siguiente información: nombre, aforo, dirección, localidad, descripción, tipo de comida, carta.
- **RI3** - Información que debe aparecer en el historial de los restaurantes (estadísticas): número de personas, duración media de estancia, porcentaje del aforo ocupado por horas.
- **RI4** - El historial accesible del restaurante será de los últimos 30 días.
- **RI5** - La valoración de los restaurantes tendrá un rango de 1-5 estrellas.
- **RI6** - El sistema debería mostrar al usuario el nombre, descripción, y dirección del restaurante filtrado.



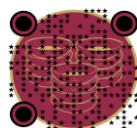
- RI7 - El sistema debería requerir al usuario una categoría específica a la hora de filtrar por tipo de comida.
- RI8 - El sistema debería requerir al usuario una valoración concreta a la hora de filtrar por valoraciones.
- RI9 - El sistema debería requerir al usuario su ubicación a la hora de filtrar por restaurante más cercano.
- RI10 - Distancias expresadas en kilómetros.
- RI11 - Cuando vas a pasar el código QR se debe indicar la cantidad de personas que van a pasar.
- RI12 - La valoración que aparece de cada restaurante es la media de las valoraciones.
- RI13 - Para iniciar sesión el usuario debe introducir su email y contraseña.

## Glosario de términos

- **Front:** Término utilizado para denominar a la parte visual de la aplicación, es decir, parte de la aplicación que interactúa con los usuarios.
- **Back:** Término utilizado para definir la parte trasera de la aplicación, es decir, el acceso a datos, el procesamiento de la información y el envío al usuario que el cliente final no ve.
- **QR:** Código de identificación único que se puede escanear desde muchas aplicaciones acondicionadas para ello.
- **Sistema:** Entendemos por sistema a la aplicación en su conjunto.
- **Interfaz:** Palabra que denomina a un componente visual con el cual se puede interactuar para obtener ciertas funcionalidades.
- **Usuario:** Término que se refiere al cliente final que usará la aplicación.
- **Historial:** Relación detallada y ordenada que se hace sobre los usuarios registrados en cada restaurante.



- **Reseñas:** Opiniones sinceras de usuarios de nuestro negocio. Evalúa la calidad de cualquier tipo de oferta culinaria, tomando en cuenta varios aspectos.
- **JS (JavaScript):** Lenguaje de programación que utilizamos para añadir interactividad a la aplicación web.
- **Html:** Lenguaje de marcas que usamos para estructurar y dar significado a nuestro contenido web.
- **CSS:** Lenguaje que usamos para estilizar los elementos del lenguaje de marcado HTML.
- **Responsive:** Técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles.



## Validación de requisitos

### Walk-throughs

Para la confirmación de requisitos realizada en la autoría, hemos decidido seguir la estrategia de validación basada en **Walk-throughs**.

En ella, estuvimos leyendo junto al cliente cada uno de los requisitos, buscando que se cumpliesen las necesidades del cliente, además de buscando conflictos y requisitos duplicados.

### Auditoría

Después de realizar la validación junto con el profesor a través del método *Walk-throughs*, juntos como equipo hemos realizado una última validación a través de una auditoría en la que hemos hecho una lectura y análisis final de la lista de requisitos para ver que no se nos olvidará añadir ningún requisito.

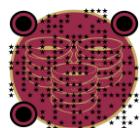
Tras la auditoría pudimos confirmar que había ciertas incongruencias reflejadas en nuestra especificación. Debido a esto, las corregimos rápida y eficazmente introduciendo algunos requisitos actualizados y descartando otros. Por ejemplo: El sistema debe permitir filtrar por valoración, el sistema debería permitir buscar restaurantes por ubicación, el sistema debería permitir al administrador ver sus restaurantes.

Con esto ya podemos dar por concluida la lista de requisitos. A partir de esta lista empezamos a modelar los casos de uso y las diferentes actividades de nuestro sistema.

### Sketches o bocetos

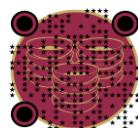
Otra técnica que hemos utilizado para confirmar que los requisitos se cumplen de forma correcta es el uso de bocetos de HTML y CSS (librería Bootstrap 5).

Los bocetos se pueden ver en el siguiente enlace o QR:  
<https://cristiancosano.github.io/QRRestaurant/index.html>



## Matriz de trazabilidad

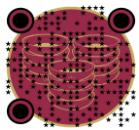
	R F 1	R F 2	R F 3	R F 4	R F 5	R F 6	R F 7	R F 8	R F 9	R F I 0	R F I 1	R F I 2	R F I 3	R F I 4	R F I 5	R F I 6	R F I 7	R F I 8	R F I 9	R F 2 0	R F 2 1	R F 2 2	R F 2 3	R F 2 4	
CU1	X																								
CU2		X																							
CU3		X																							
CU4			X																						
CU5				X	X	X								X											
CU6				X			X					X													
CU7										X															
CU8																						X			
CU9																							X		
CU10											X														
CU11																								X	
CU12																		X							
CU13												X							X						
CU14																				X					
CU15																		X				X			
CU16																							X		
CU17														X											
CU18					X										X										
CU19																X									
CU20								X										X							



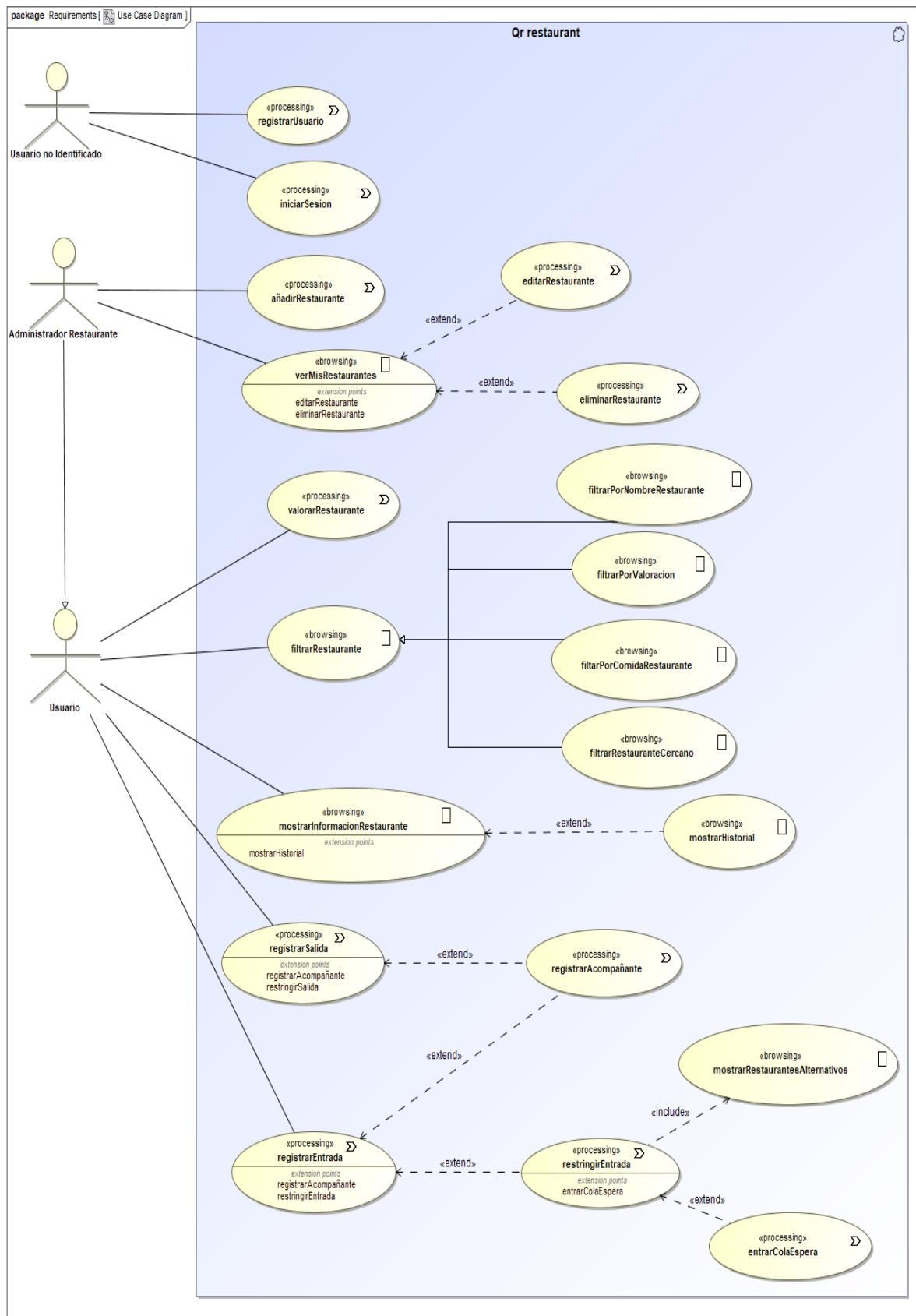
## Requisitos modelados como Casos de Uso

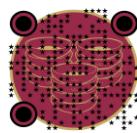
### Identificación de los casos de uso

- CU 1 - mostrarHistorial
- CU 2 - registrarEntrada
- CU 3 - restringirEntrada
- CU 4 - mostrarRestaurantesAlternativos
- CU 5 - mostrarInformacionRestaurante
- CU 6 - filtrarRestaurante
- CU 7 - filtrarPorComidaRestaurante
- CU 8 - filtrarPorValoración
- CU 9 - filtrarPorRestauranteCercano
- CU 10 - mostrarRestauranteMapa
- CU 11 - VerMisRestaurantes
- CU 12 - registrarAcompañantes
- CU 13 - entrarColaDeEspera
- CU 14 - valorarRestaurante
- CU 15 - registrarUsuario
- CU 16 - iniciarSesion
- CU 17 - añadirRestaurante
- CU 18 - editarRestaurante
- CU 19 - eliminarRestaurante
- CU 20 - registrarSalida



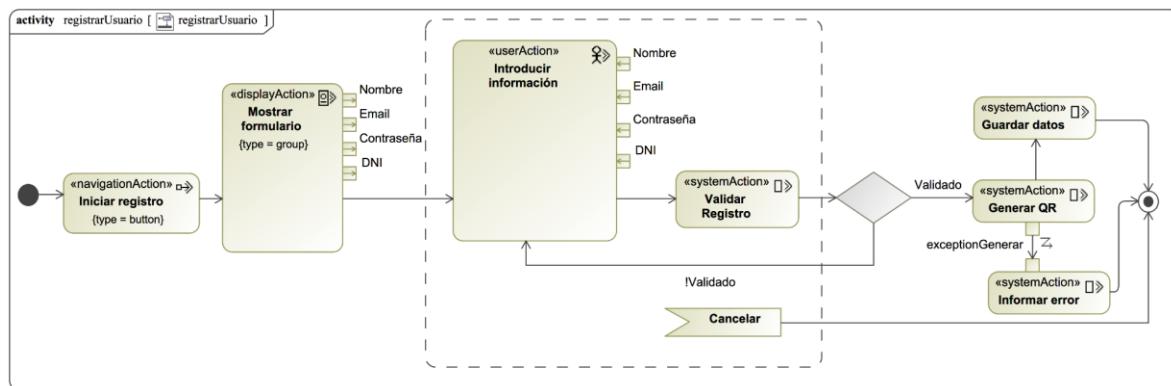
## Diagrama de casos de uso



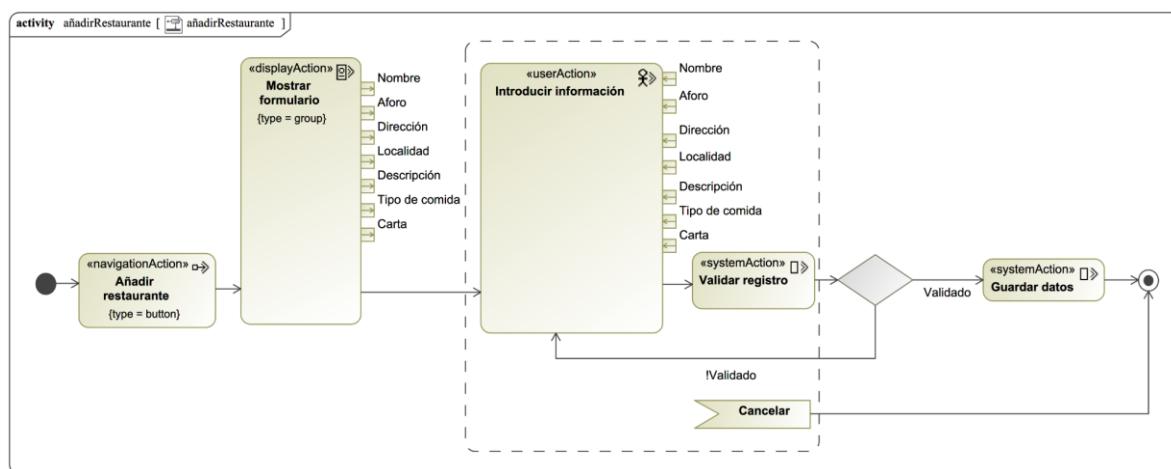


## Requisitos modelados como actividades

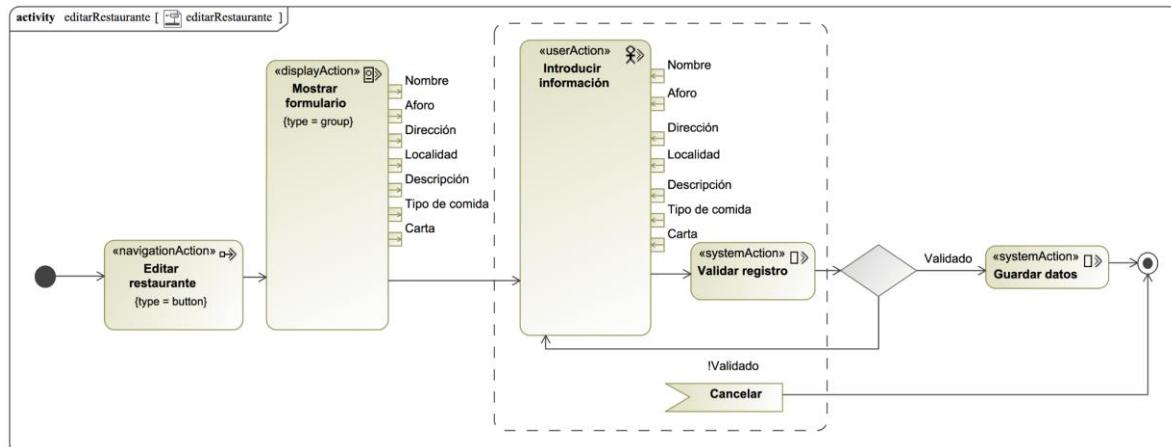
### registrarUsuario

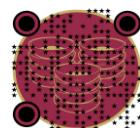


### añadirRestaurante

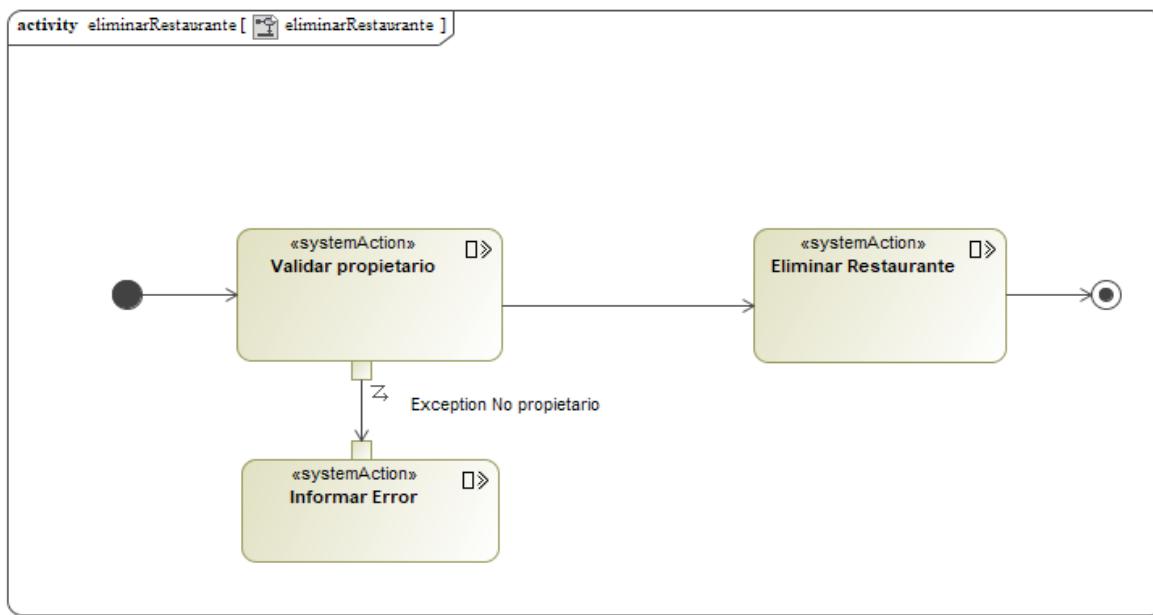


### editarRestaurante

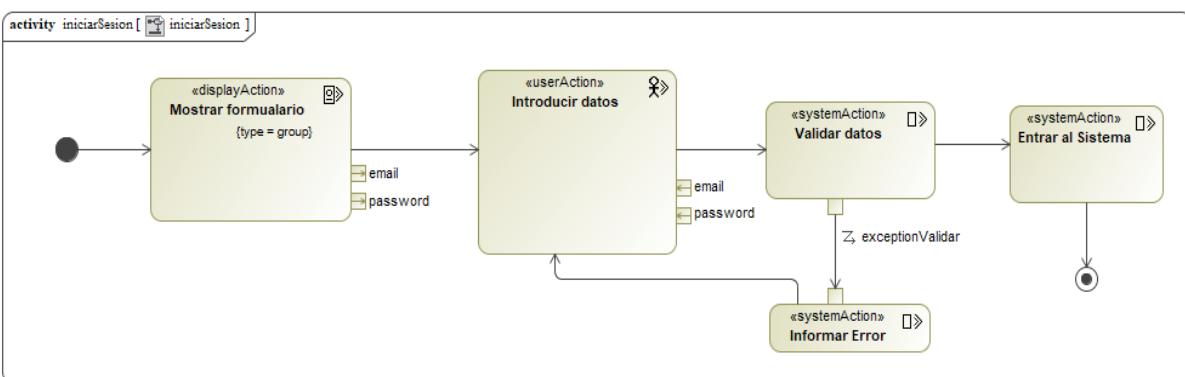


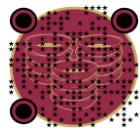


## eliminarRestaurante

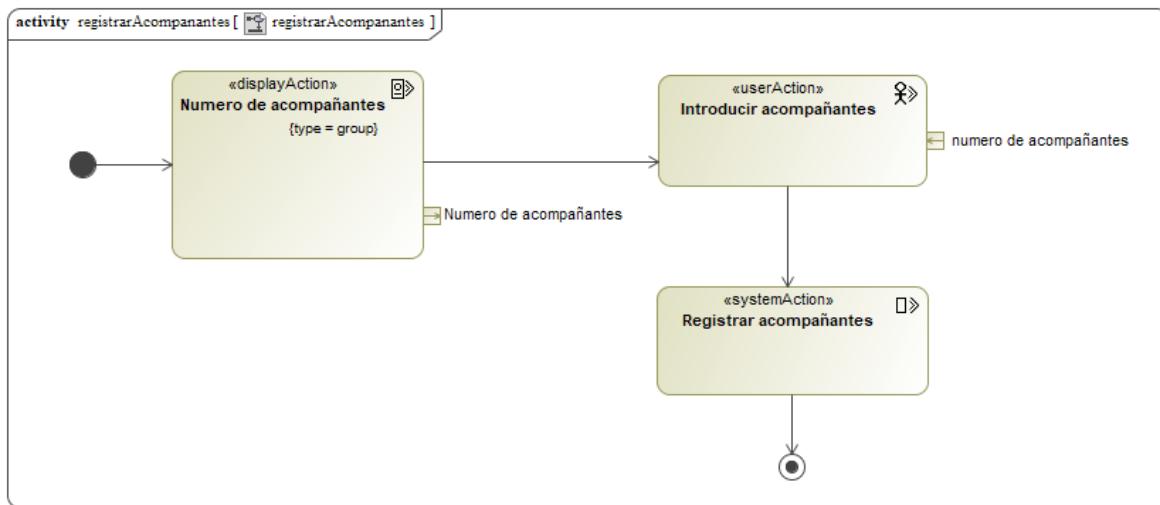


## iniciarSesión

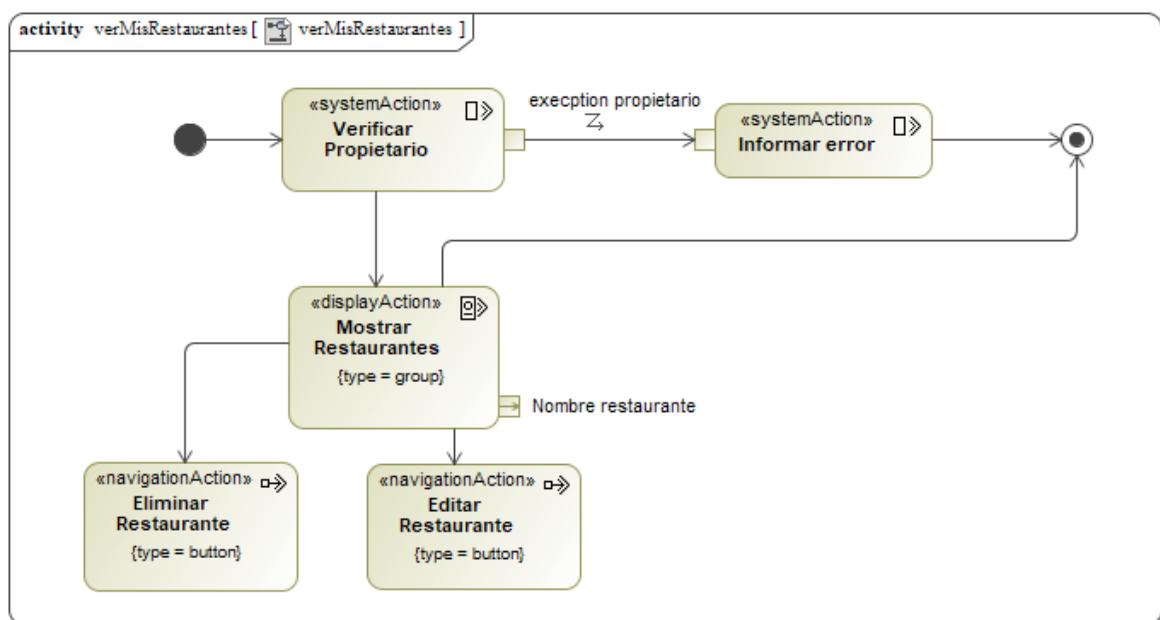


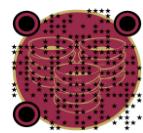


## registrarAcompañante

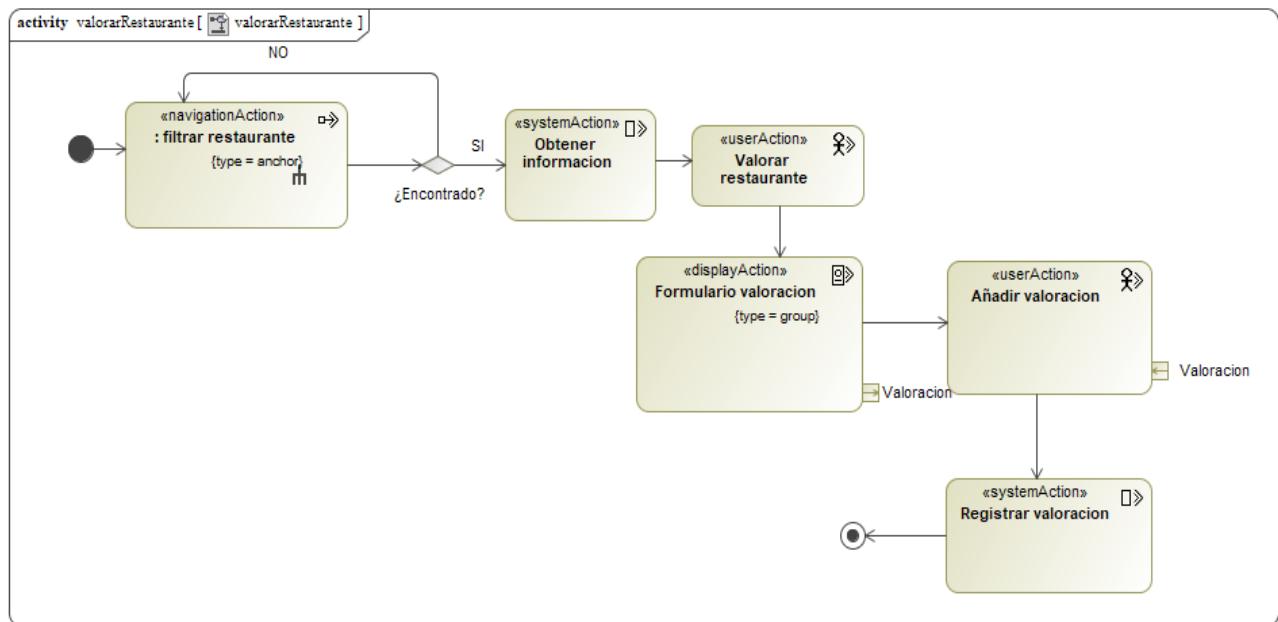


## verMisRestaurantes

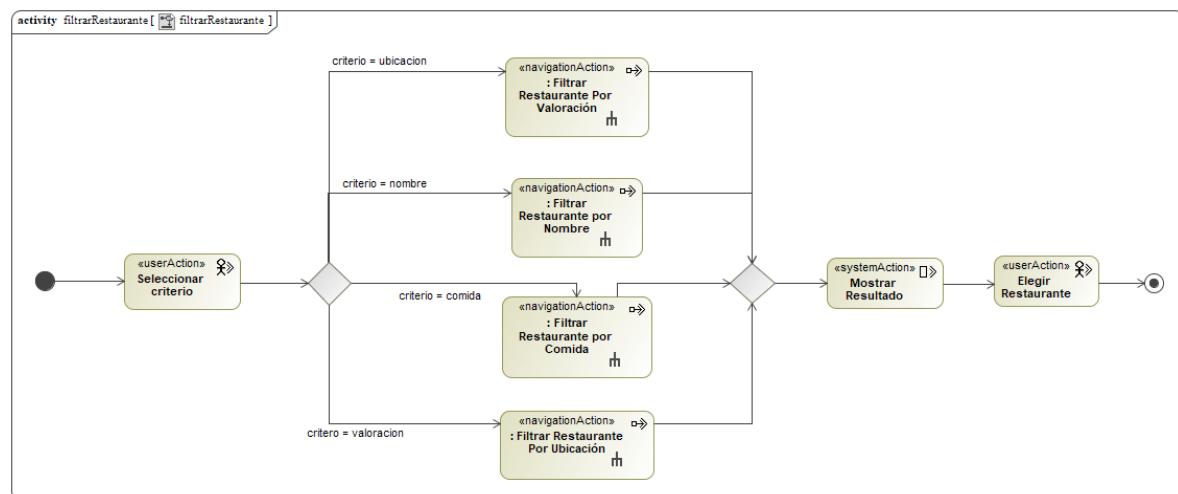


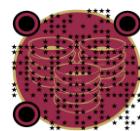


## valorarRestaurante

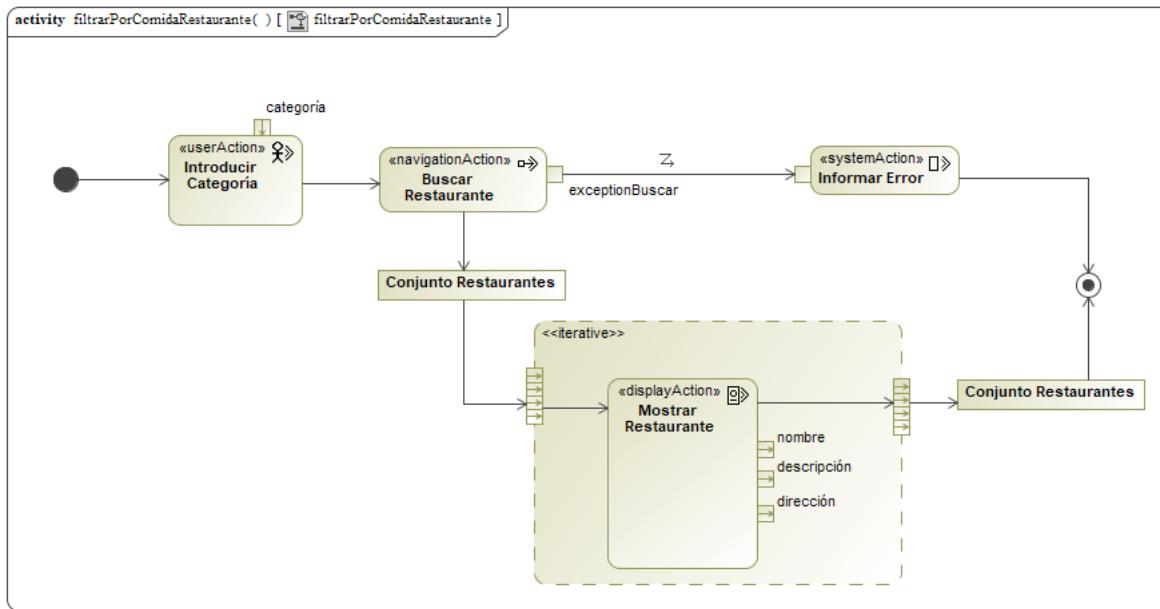


## filtrarRestaurante

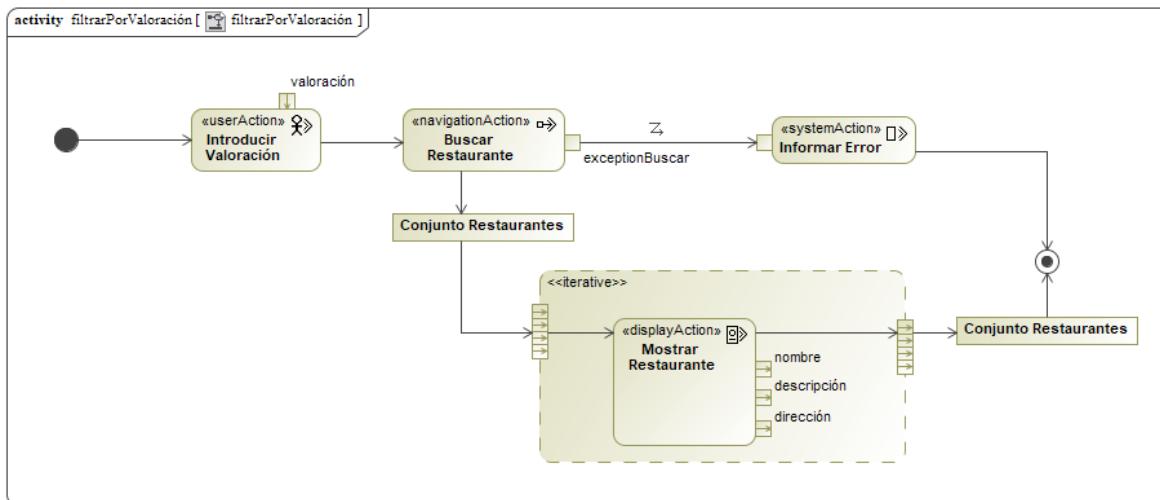


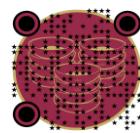


## filtrarPorComidaRestaurante

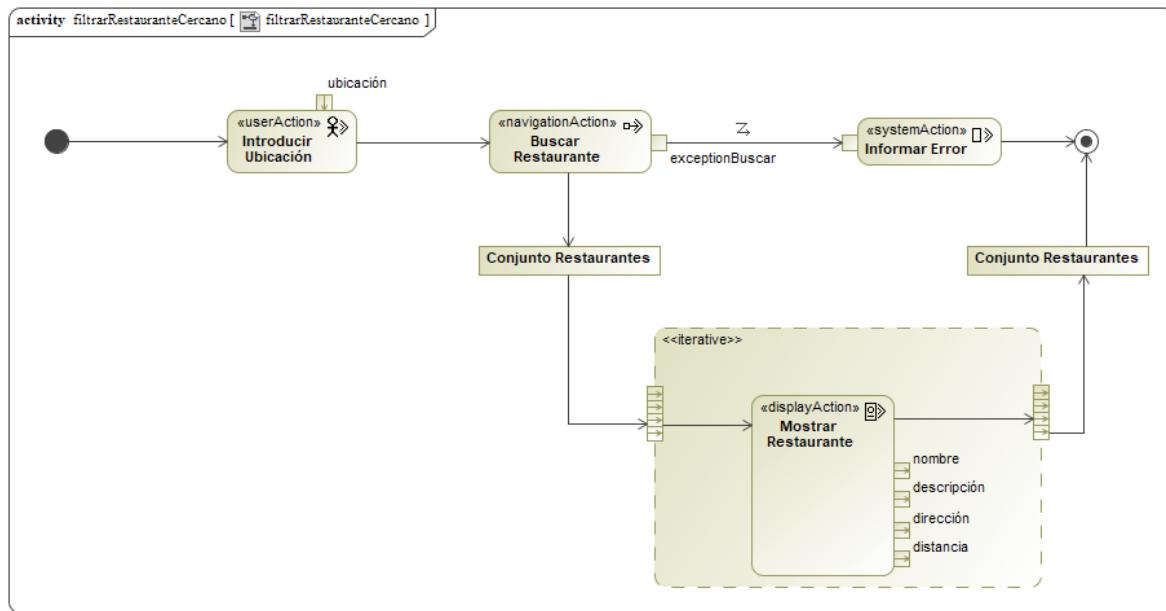


## filtrarPorValoración

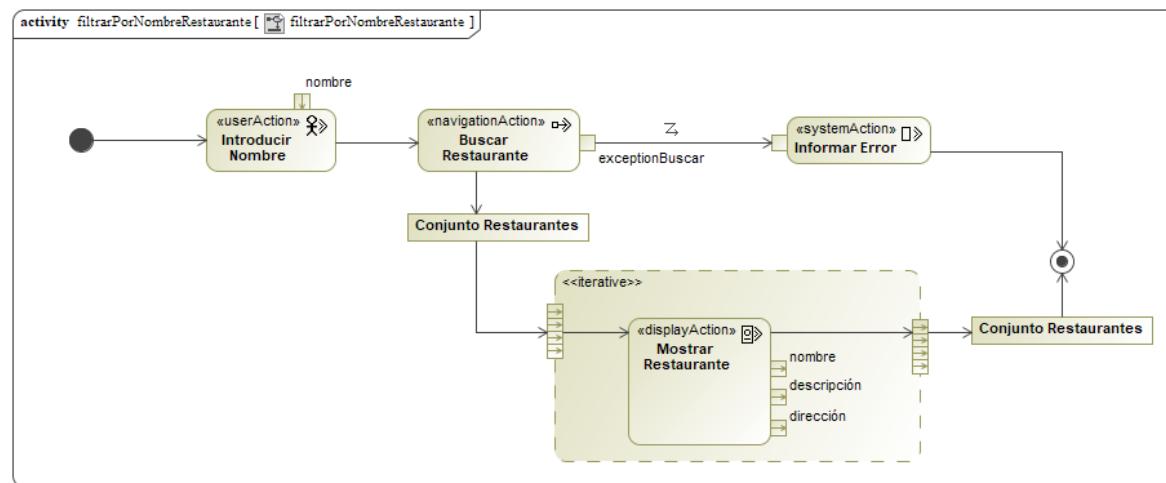


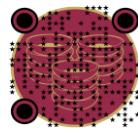


## filtrarRestauranteCercano

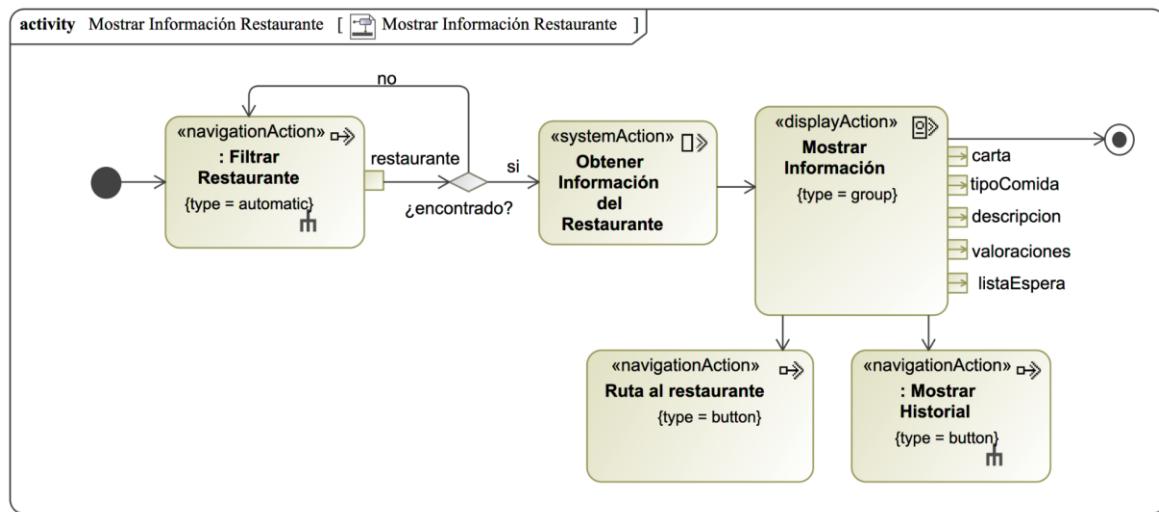


## filtrarPorNombreRestaurante

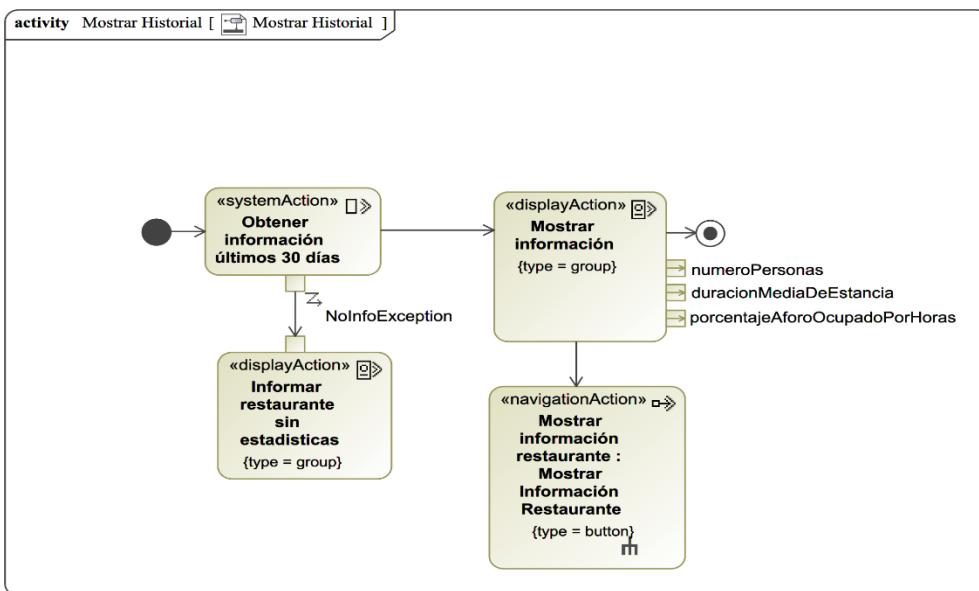


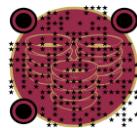


## mostrarInformaciónRestaurante

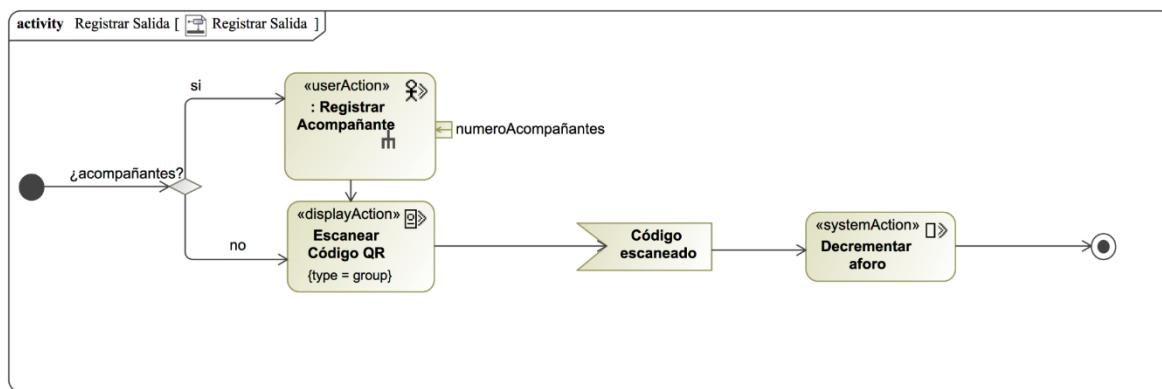


## mostrarHistorial

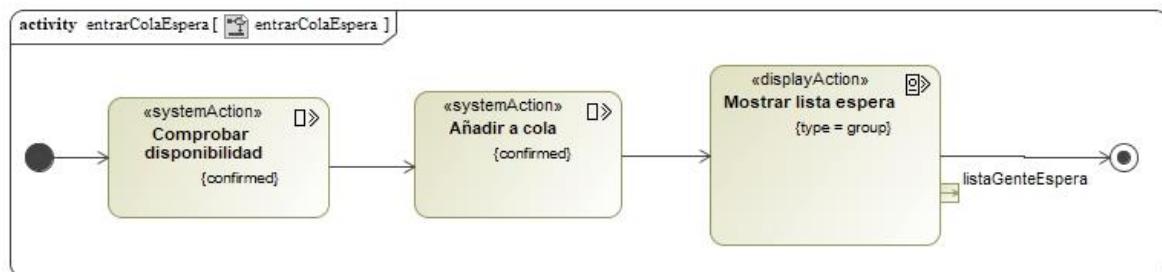




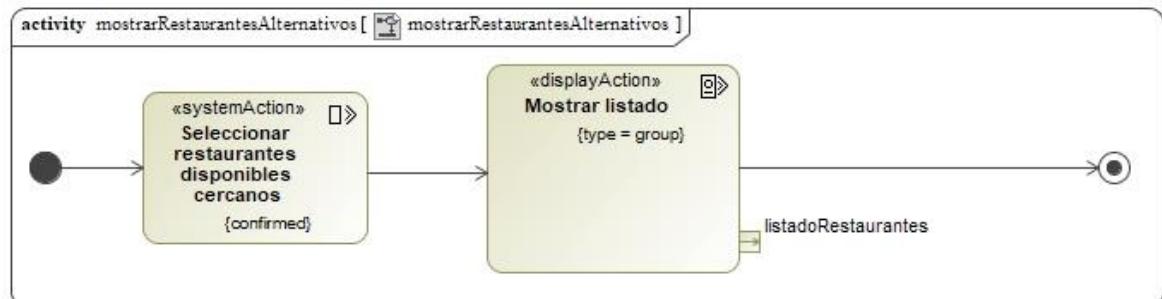
## registrarSalida

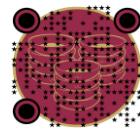


## entrarColaEspera

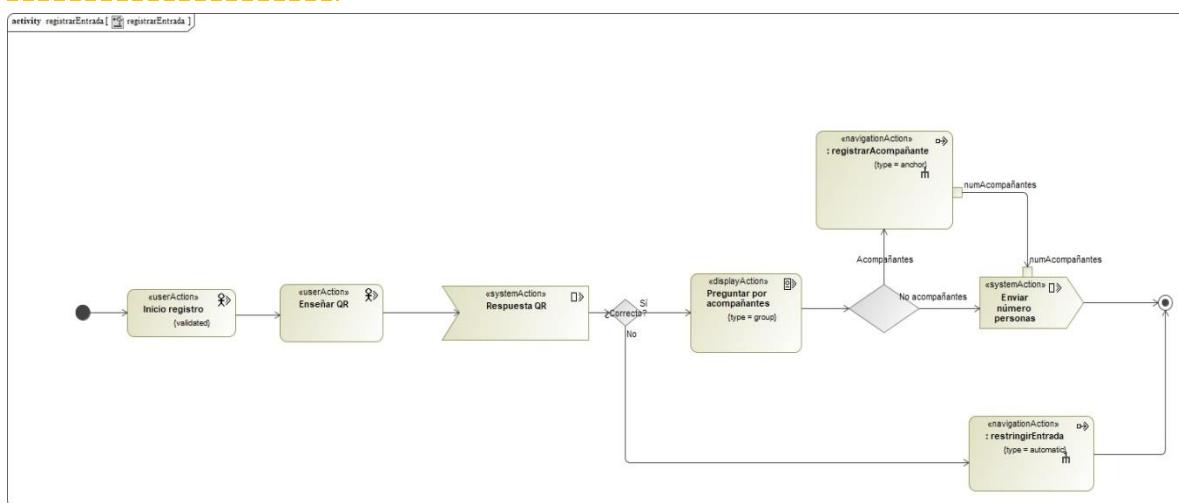


## mostrarRestaurantesAlternativos





## registrarEntrada



## restringirEntrada

