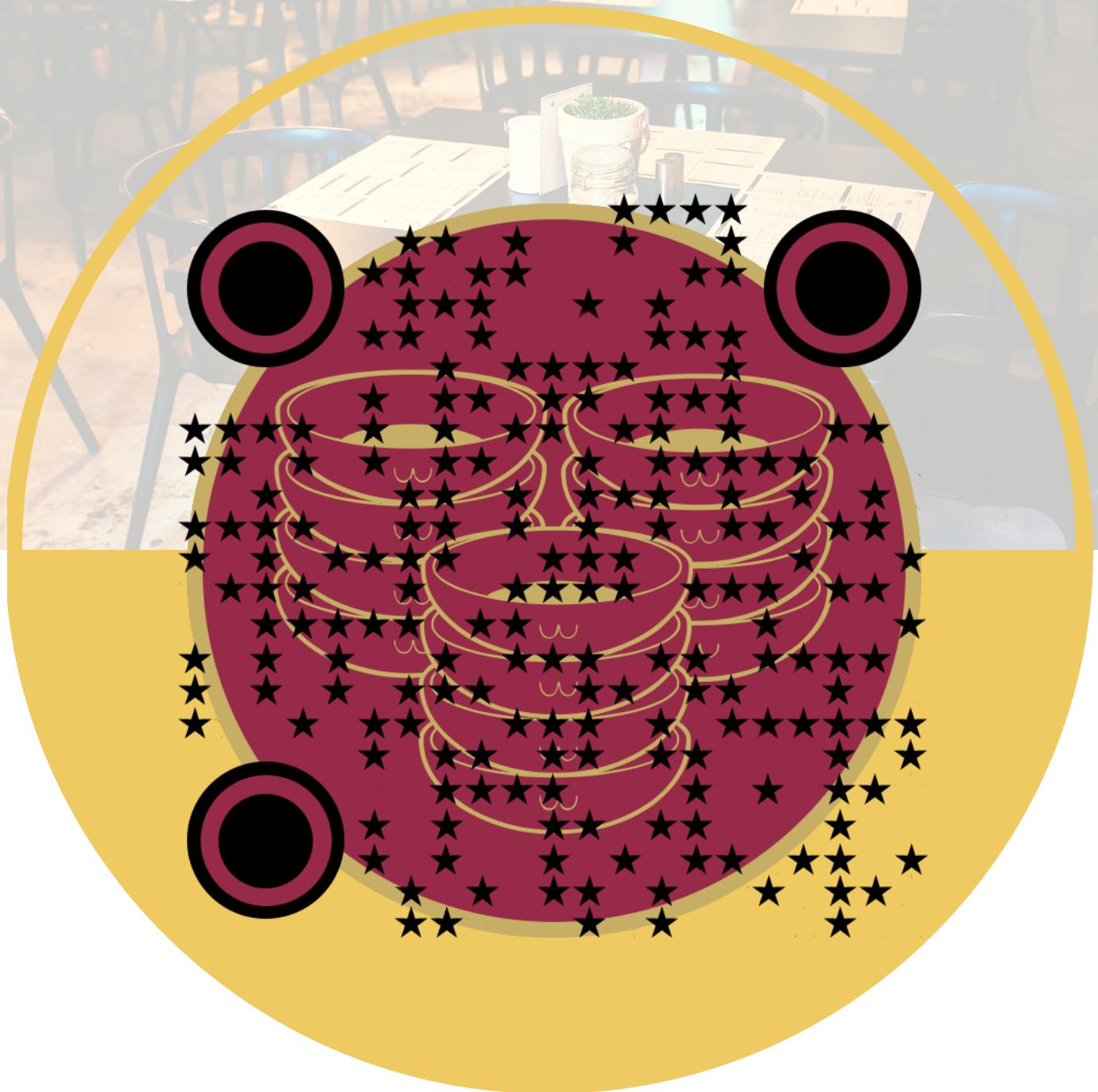


QR RESTAURANT



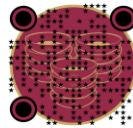
Francisco Almenara Parias

Manuel Cortés Serrano

Cristian Cosano Cejas

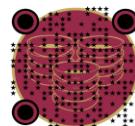
Jose Manuel Gil Rodríguez

Antonio Luis Rodríguez Jiménez

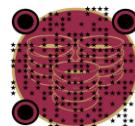


Índice

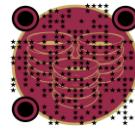
<u>PRÁCTICA 1</u>	5
Introducción	5
Definición del problema	6
Alcance del sistema	7
Para qué entorno estamos haciendo la aplicación web	7
¿Quién es el público que va a utilizar la aplicación web?	7
Resumidamente y formal, ¿en qué va a consistir?	7
¿Qué propone la aplicación web?	8
Objetivos del problema	8
Objetivos para el público	8
Objetivos técnicos	9
Antecedentes	9
Recursos	10
Recursos humanos	10
Recursos hardware	10
Recursos software	10
Reparto de tareas	11
Metodología Ágil. SCRUM.	12
 <u>PRÁCTICA 2</u>	13
Captura de requisitos	13
Entrevista con el cliente	14
Definición de requisitos	14
Requisitos funcionales	14
Restaurante	14
Usuario	15
Requisitos no funcionales	16
Requisitos de información	16
Glosario de términos	17
Validación de requisitos	18
Walk-throughs	18



Auditoría	18
Sketches o bocetos	19
Matriz de trazabilidad	20
Requisitos modelados como Casos de Uso	21
Identificación de los casos de uso	21
Diagrama de casos de uso	22
Requisitos modelados como actividades	23
registrarUsuario	23
añadirRestaurante	23
editarRestaurante	23
eliminarRestaurante	24
iniciarSesión	24
registrarAcompañante	25
verMisRestaurantes	25
valorarRestaurante	26
filtrarRestaurante	26
filtrarPorComidaRestaurante	27
filtrarPorValoración	27
filtrarRestauranteCercano	28
filtrarPorNombreRestaurante	28
mostrarInformaciónRestaurante	29
mostrarHistorial	29
registrarSalida	30
entrarColaEspera	30
mostrarRestaurantesAlternativos	30
registrarEntrada	31
<u>PRÁCTICA 3</u>	32
Modelado de contenido	32
Modelo de contenido	32
Modelo de usuario	33
Modelo de navegación	34
Modelado de proceso	35
Modelo de estructura del proceso	35



Modelo de flujo del proceso	36
registrar	36
iniciarSesion	36
cerrarSesión	37
editarRestaurante	37
añadirRestaurante	38
eliminarRestaurante	38
valorarRestaurante	39
escanearQR	39
Modelado de presentación	40
 <u>PRÁCTICA 4</u>	41
 User eXperience	41
Día 1	41
Día 2	44
Cambios realizados tras las pruebas de UX	48
Compatibilidad de la carta	48
Mensaje en la pantalla de inicio	48
Buscar por valoración	49
Buscar por tipo de comida	50
Pruebas Usabilidad	51
Pruebas de compatibilidad	51
Pruebas de rendimiento	52
Pruebas de seguridad	54
Pruebas de Base de datos	55
Pruebas de funcionalidad	57
Cumplimiento de requisitos	58



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.

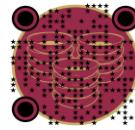
La nueva normalidad tras la pandemia, conlleva tener que modificar ciertos hábitos de consumo que antes de la crisis sanitaria no estaban tan implementados en la sociedad. Al tiempo que el incremento del teletrabajo va normalizando el uso de nuevas herramientas digitales, en ámbitos como la hostelería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

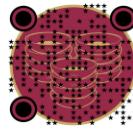


Definición del problema

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.

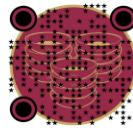
• Resumidamente y formal, ¿en qué va a consistir?

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.

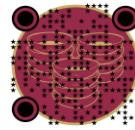
Las ventajas de utilizar códigos QR es que es una herramienta digital muy creativa que permite y acelera la utilización de servicios web en móviles. Se deben leer con un teléfono móvil o con un dispositivo que disponga de la aplicación correspondiente (un lector de códigos QR). No son legibles para el 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.

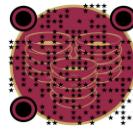
Antecedentes

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

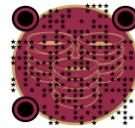
Los equipos tecnológicos con los que se trabaja y su capacidad.

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 será 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.

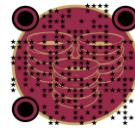
Como repositorio de código hemos utilizado el sitio web GitHub, que nos permitirá tener nuestro código centralizado y poder realizar cambios cada componente del grupo. Esta herramienta facilitará el trabajo ya que siempre tendremos la última versión del código actualizado.

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.

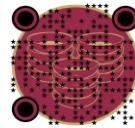


Metodología Ágil. SCRUM.

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 estas responsabilidades 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

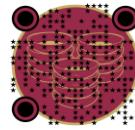
Control de accesos por QR a restaurantes. Debido a la pandemia ocasionada por el COVID, los códigos QR han ganado mucho interés, ya que son casi infinitos y tienen una facilidad de uso enorme.

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.

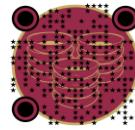
Definición de requisitos

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

Requisitos funcionales

Restaurante

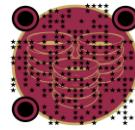
- **RF1** - Cada restaurante debe incluir un historial que muestra estadísticas de los días anteriores.
- **RF2** - El sistema debería prohibir la entrada a un usuario que intenta acceder a un local que supera el aforo.
- **RF3** - El sistema debería indicar al usuario una lista con los restaurantes 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.



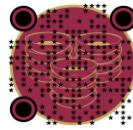
- **RF21** - El usuario debe poder iniciar sesión en el sistema.
- **RF22** - El sistema debe permitir al usuario filtrar por valoración.
- **RF23** - El sistema debe permitir buscar restaurantes por ubicación cercana.
- **RF24** - El sistema debe permitir al administrador ver sus restaurantes.

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

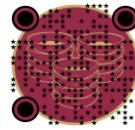
- **RI1** - El sistema debería almacenar los siguientes datos del usuario registrado: nombre, contraseña, email, DNI, código QR.
- **RI2** - El sistema debe almacenar del restaurante la siguiente información: nombre, aforo, dirección, localidad, descripción, tipo de comida, carta y fotos.
- **RI3** - Información que debe aparecer en el historial de los restaurantes (estadísticas): número de personas, duración media de la 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

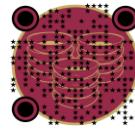
Walk-throughs

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.

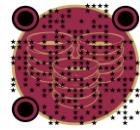
Sketches o bocetos

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:

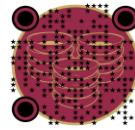
<https://cristiancosano.github.io/QRRestaurant/index.html>





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 10	R F 11	R F 12	R F 13	R F 14	R F 15	R F 16	R F 17	R F 18	R F 19	R F 20	R F 21	R F 22	R F 23	R F 24
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

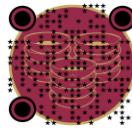
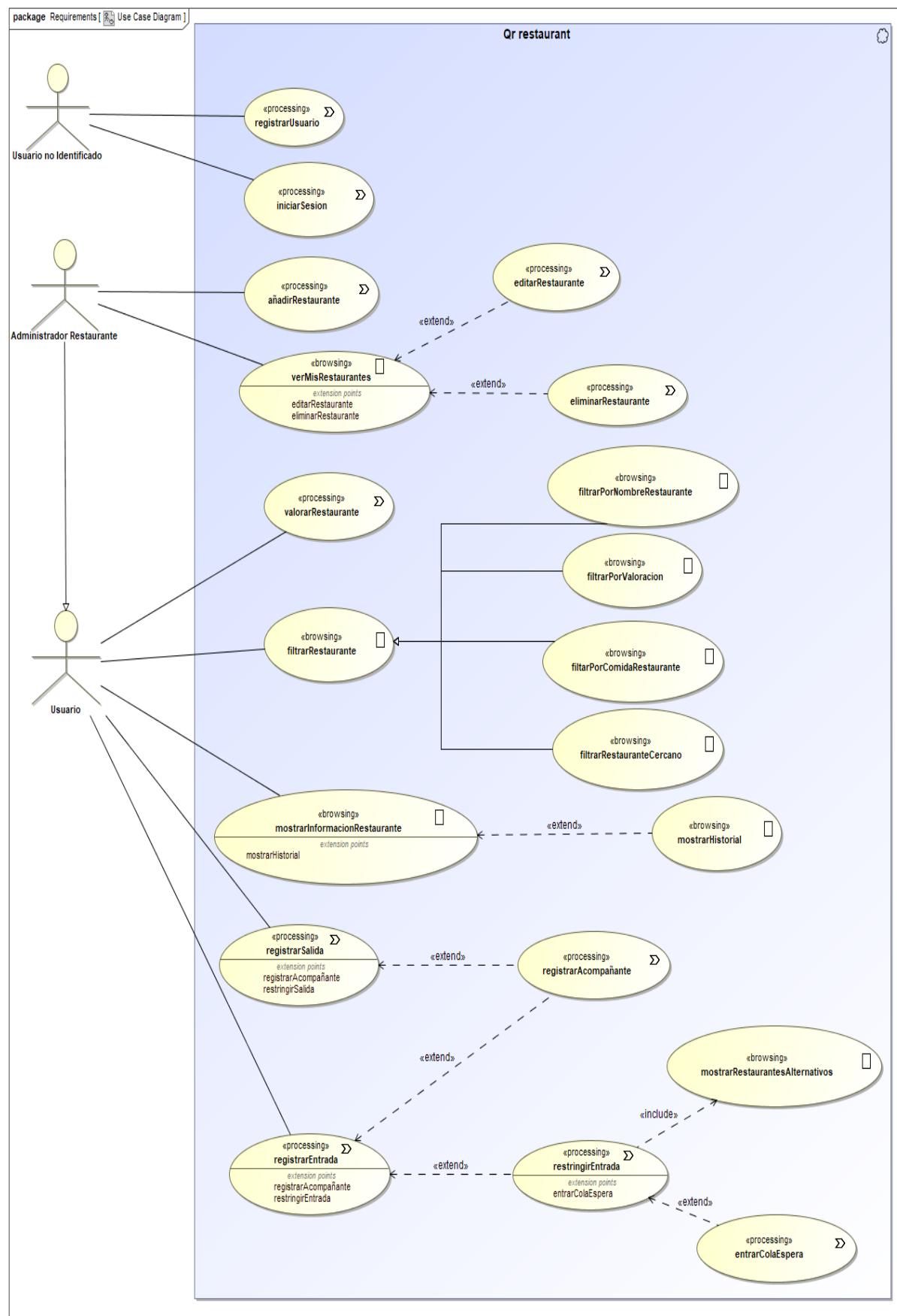
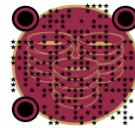


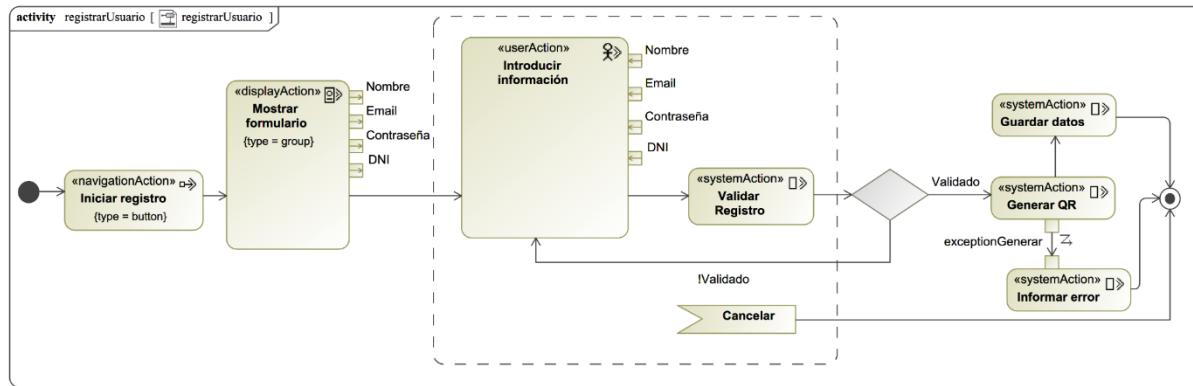
Diagrama de casos de uso



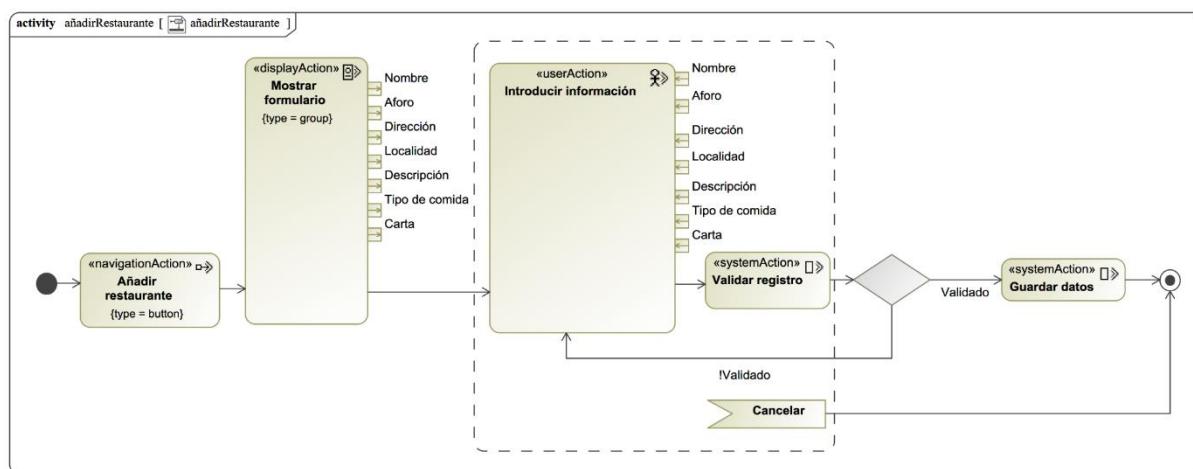


Requisitos modelados como actividades

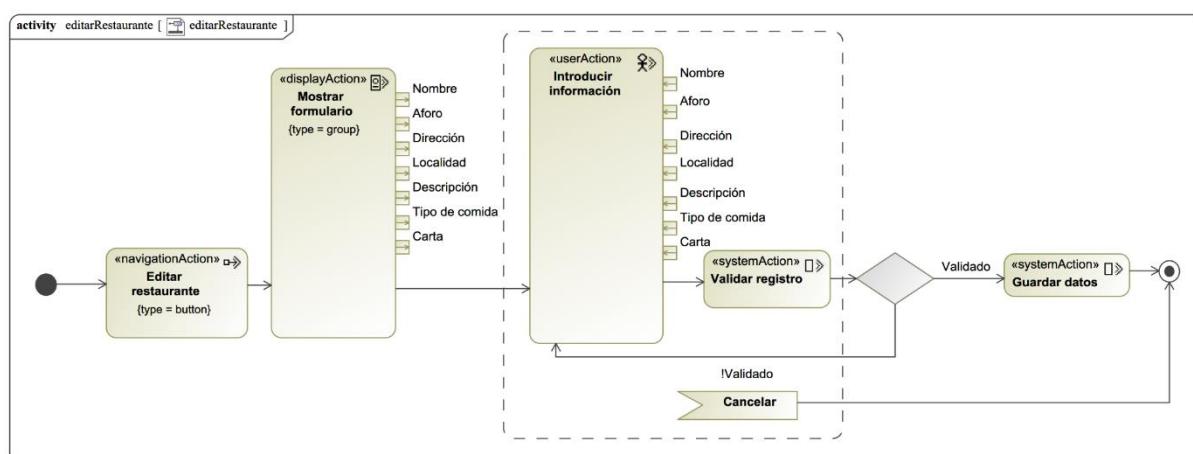
registrarUsuario

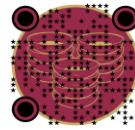


añadirRestaurante

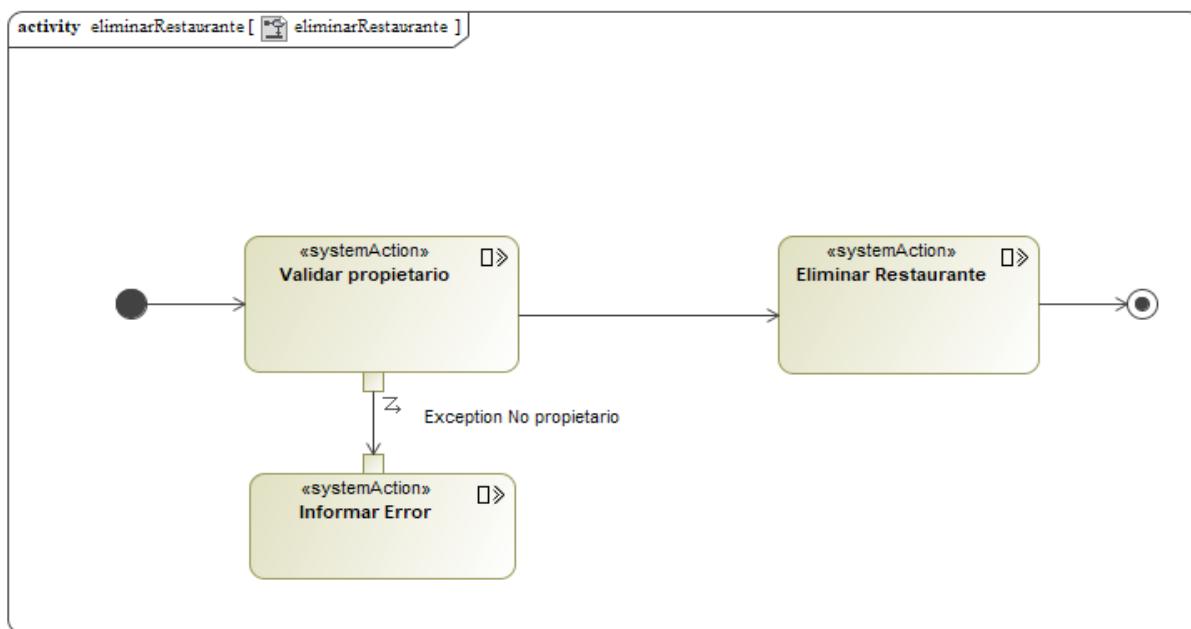


editarRestaurante

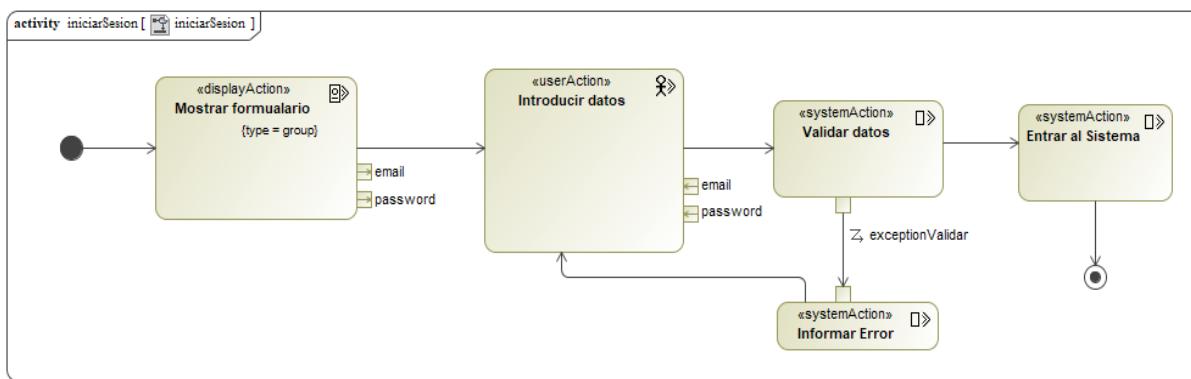


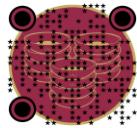


eliminarRestaurante

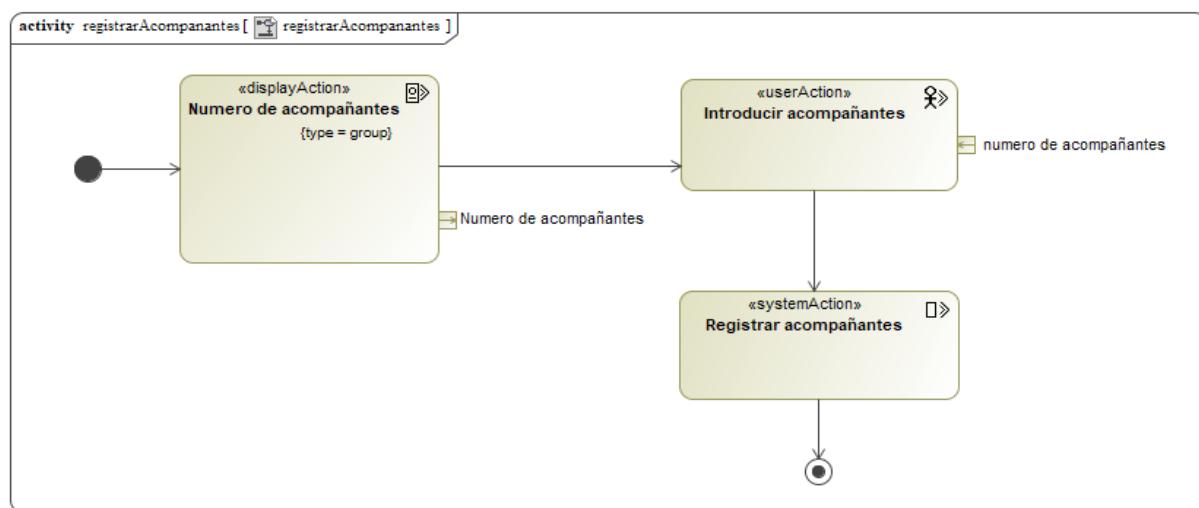


iniciarSesión

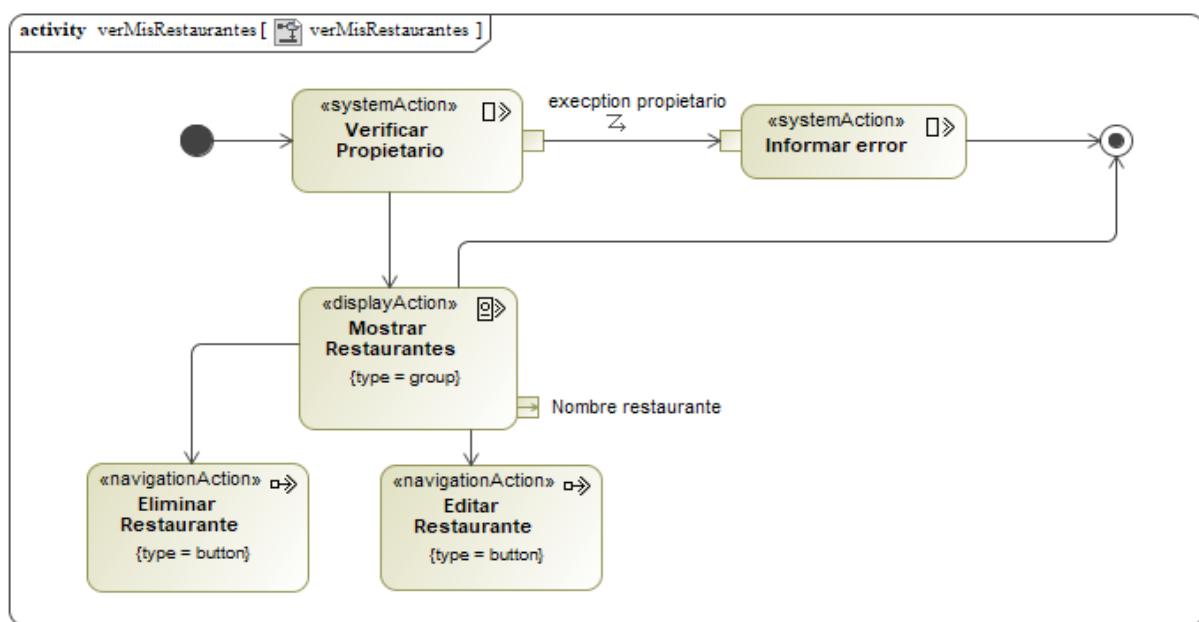


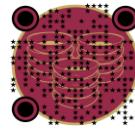


registrarAcompañante

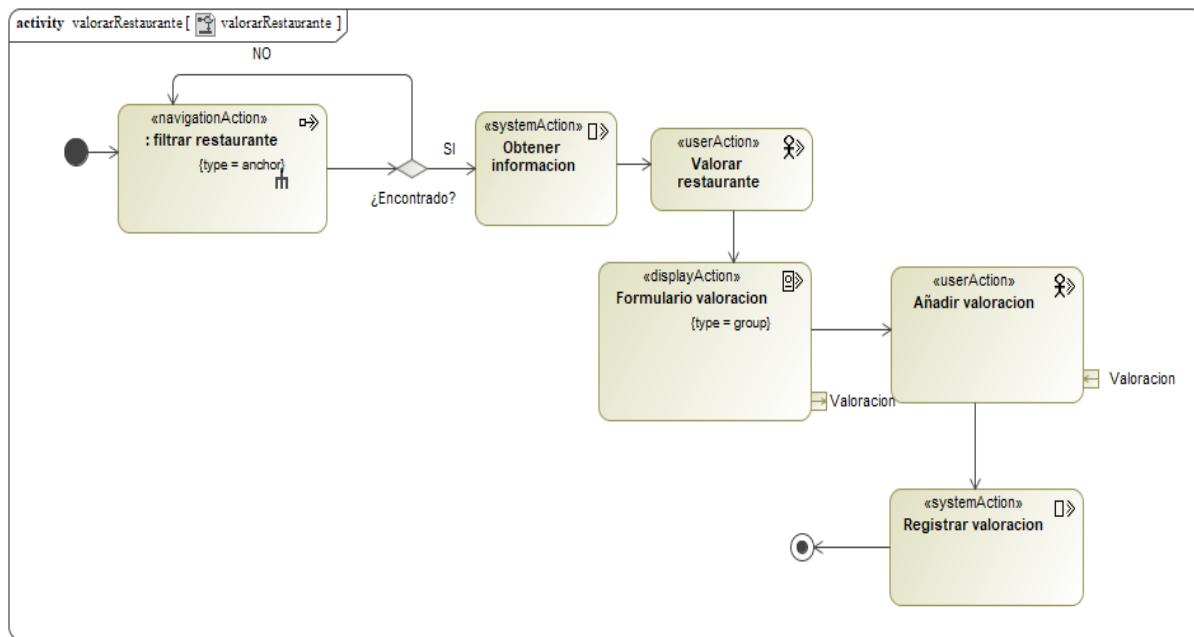


verMisRestaurantes

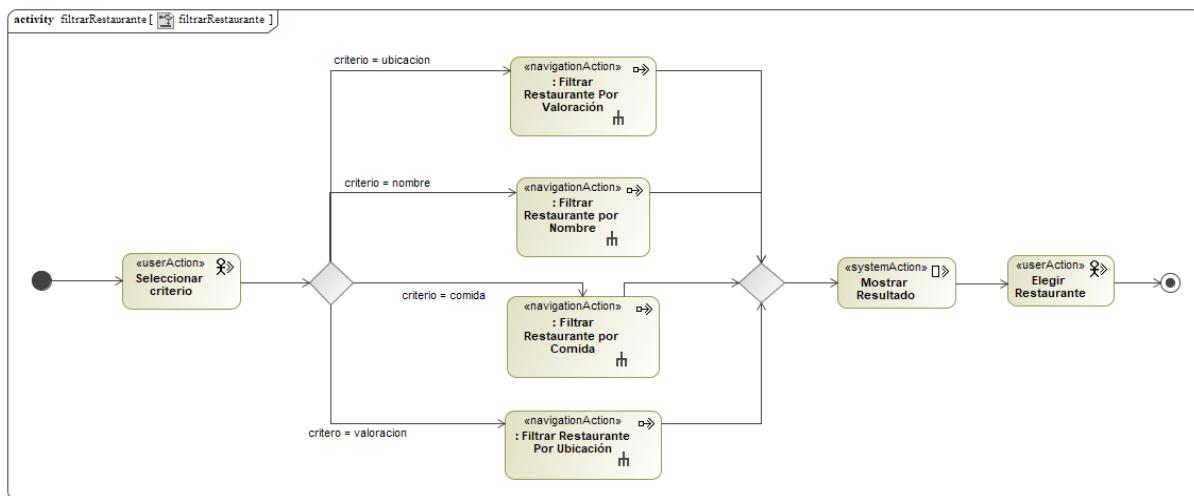


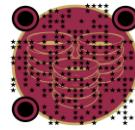


valorarRestaurante

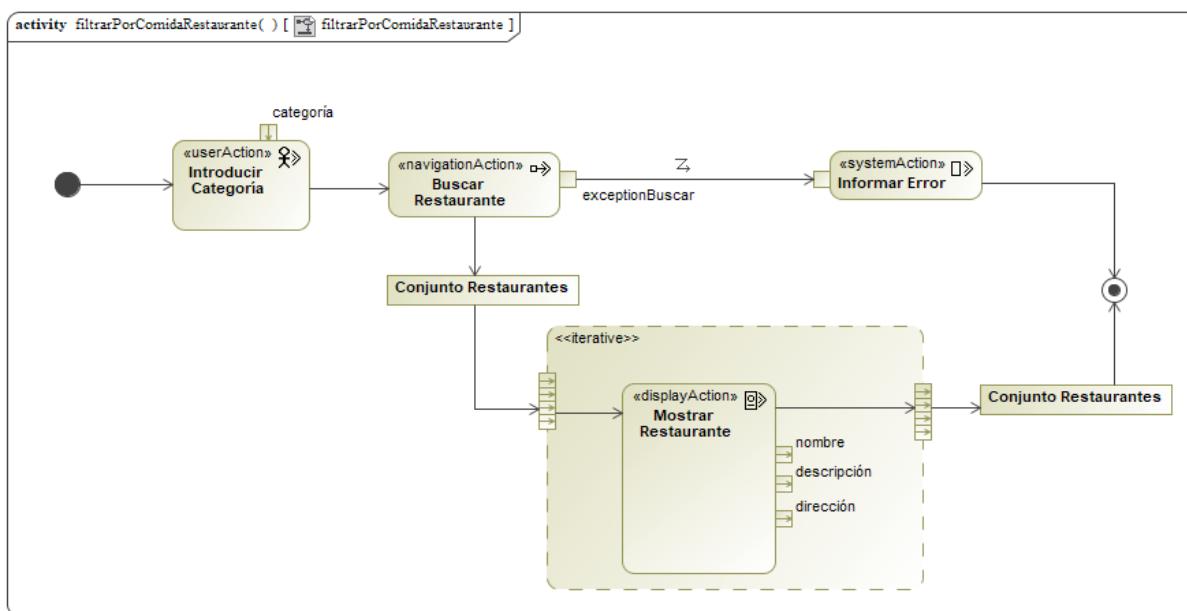


filtrarRestaurante

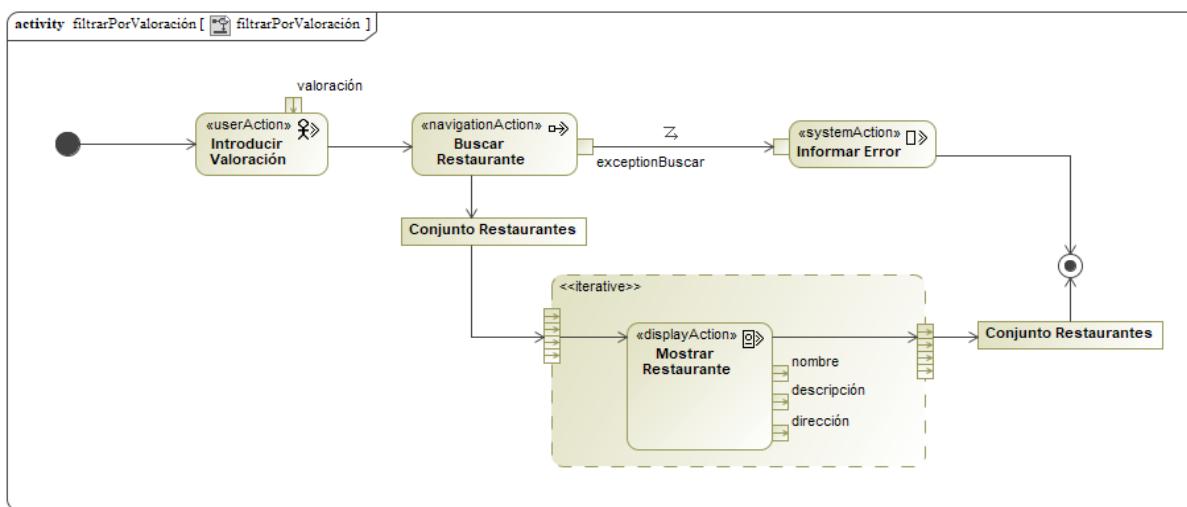


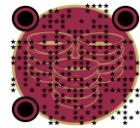


filtrarPorComidaRestaurante

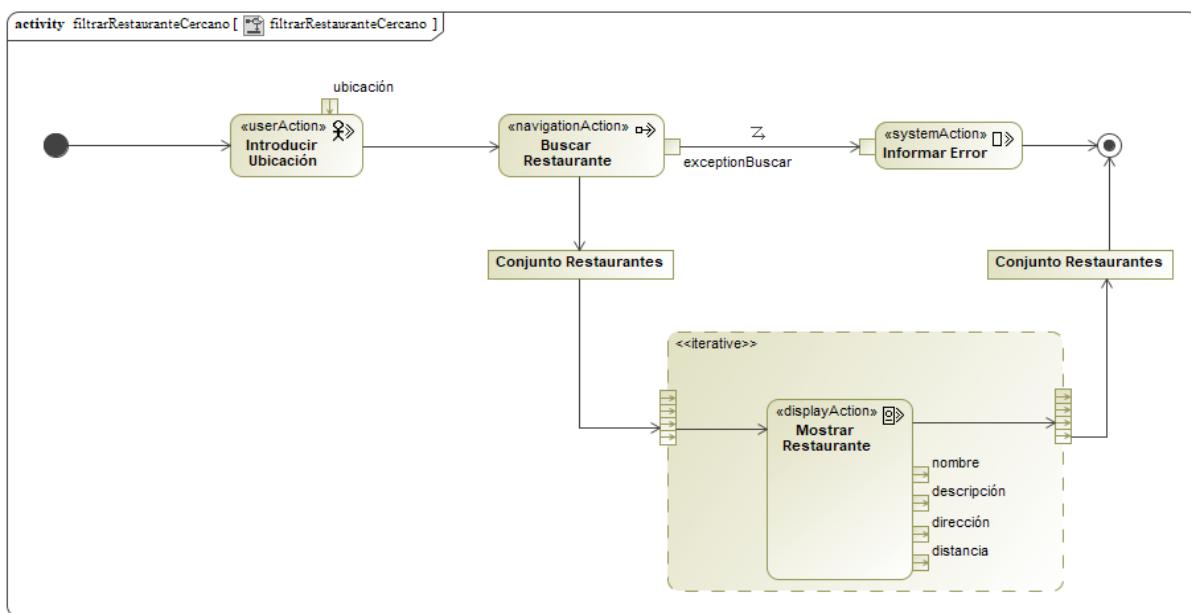


filtrarPorValoración

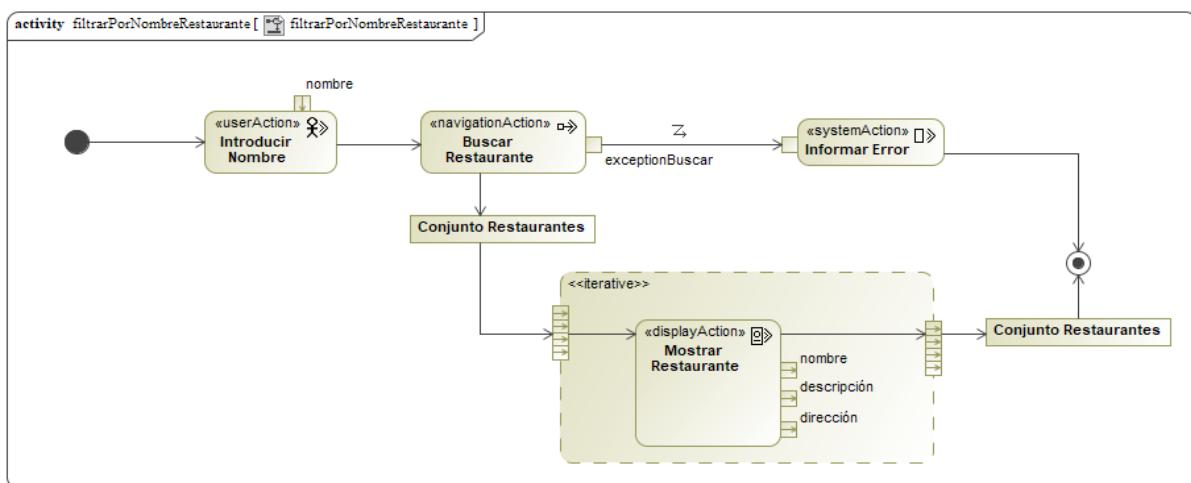


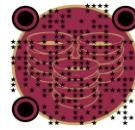


filtrarRestauranteCercano

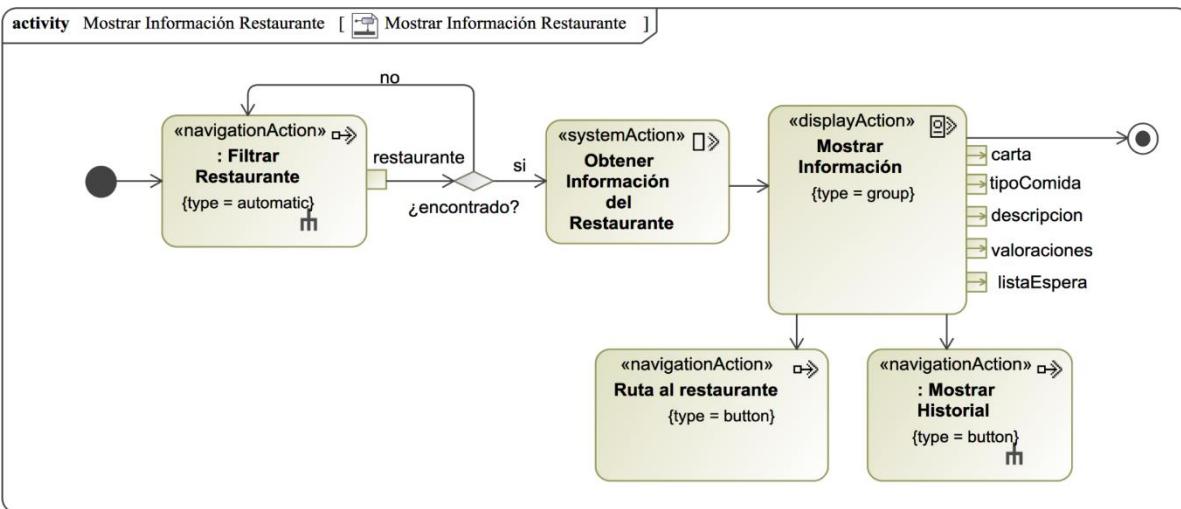


filtrarPorNombreRestaurante

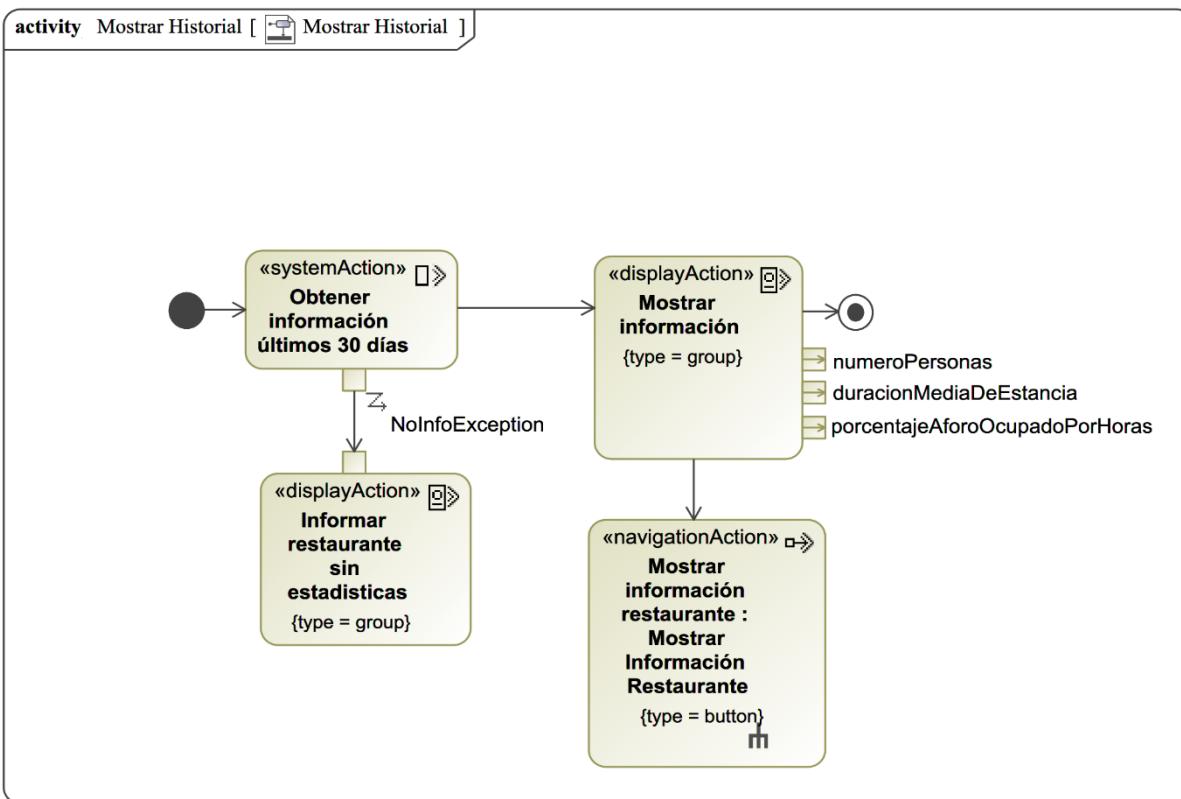


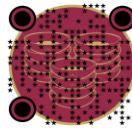


mostrarInformaciónRestaurante

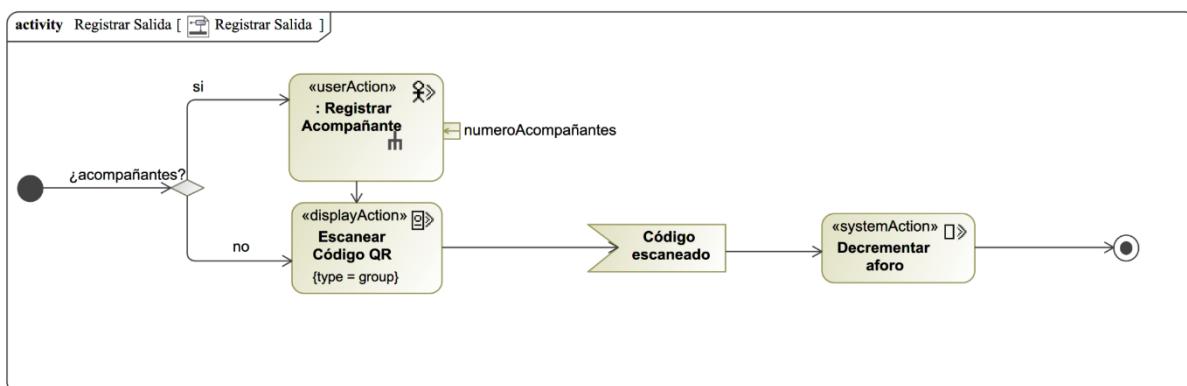


mostrarHistorial

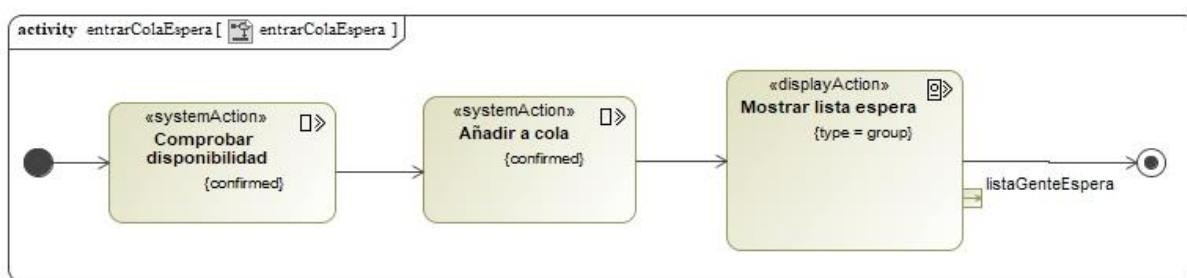




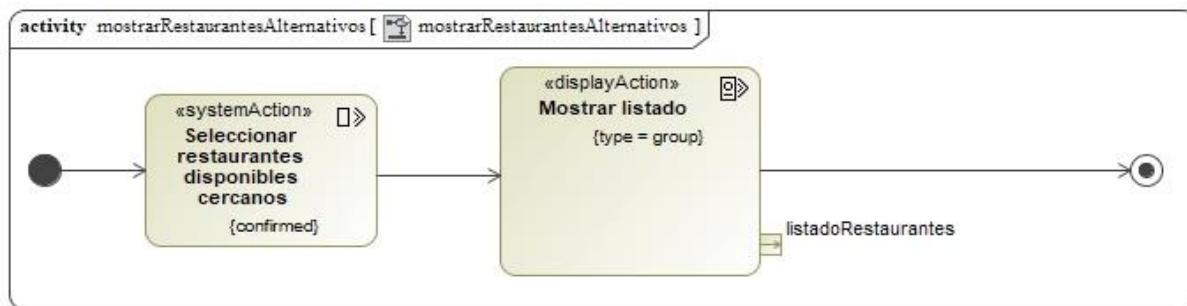
registrarSalida

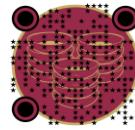


entrarColaEspera

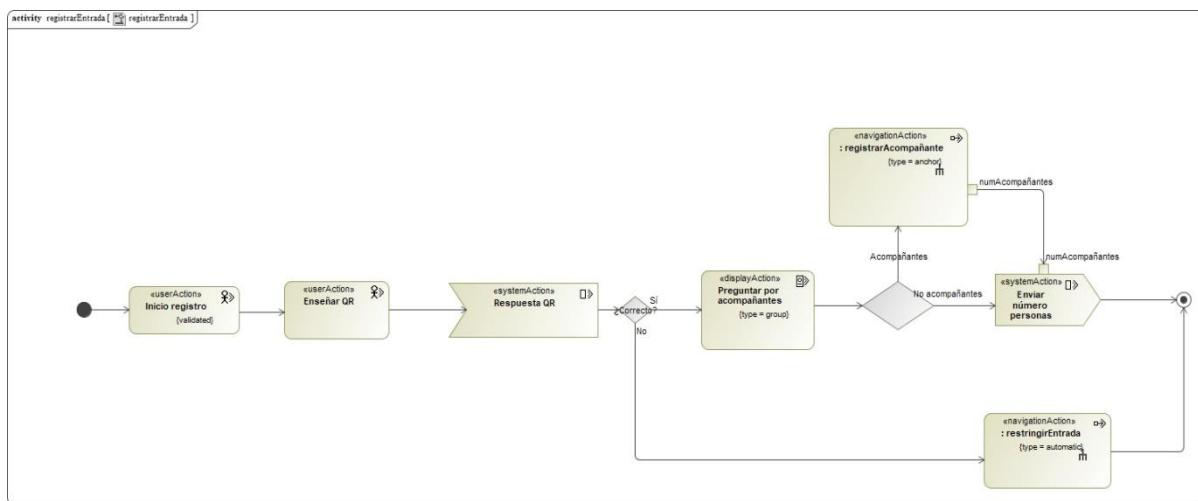


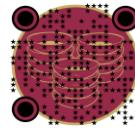
mostrarRestaurantesAlternativos





registrarEntrada





PRÁCTICA 3

Modelado de contenido

Mediante el modelado de contenido proporcionamos una especificación visual de la información relevante en nuestro dominio para el sistema web.

El contenido se modela según dos modelos (ambos representados por diagramas de clase).

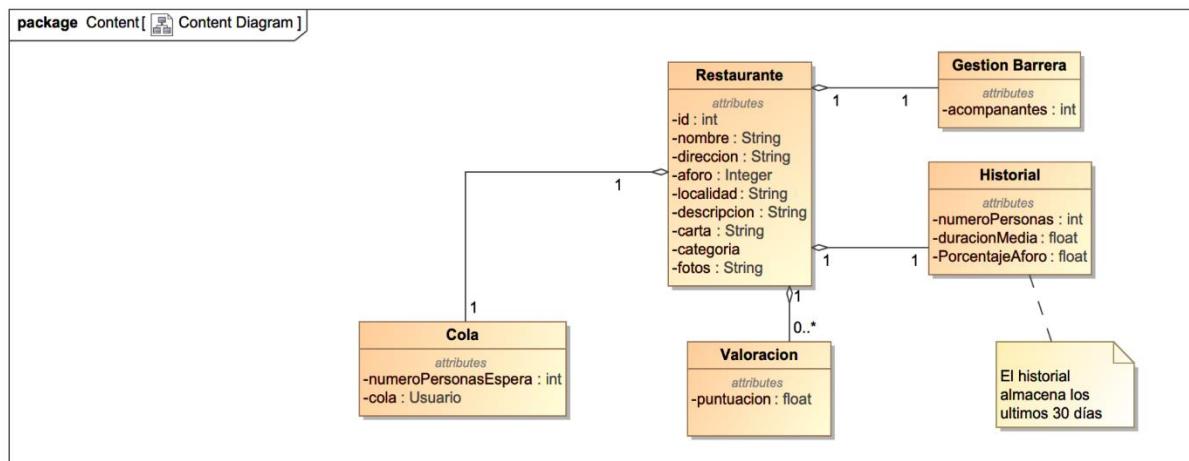
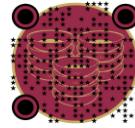
Modelo de contenido

Especificamos cómo se encuentran relacionados los contenidos del sistema, es decir, definimos la estructura de los datos que se encuentran alojados en el sitio web.

El objetivo es representar la estructura del contenido, mediante un diagrama de clases puro de UML, sin entrar en navegación, así como las relaciones existentes entre las diferentes clases.

En nuestro caso, hemos identificado cinco clases,

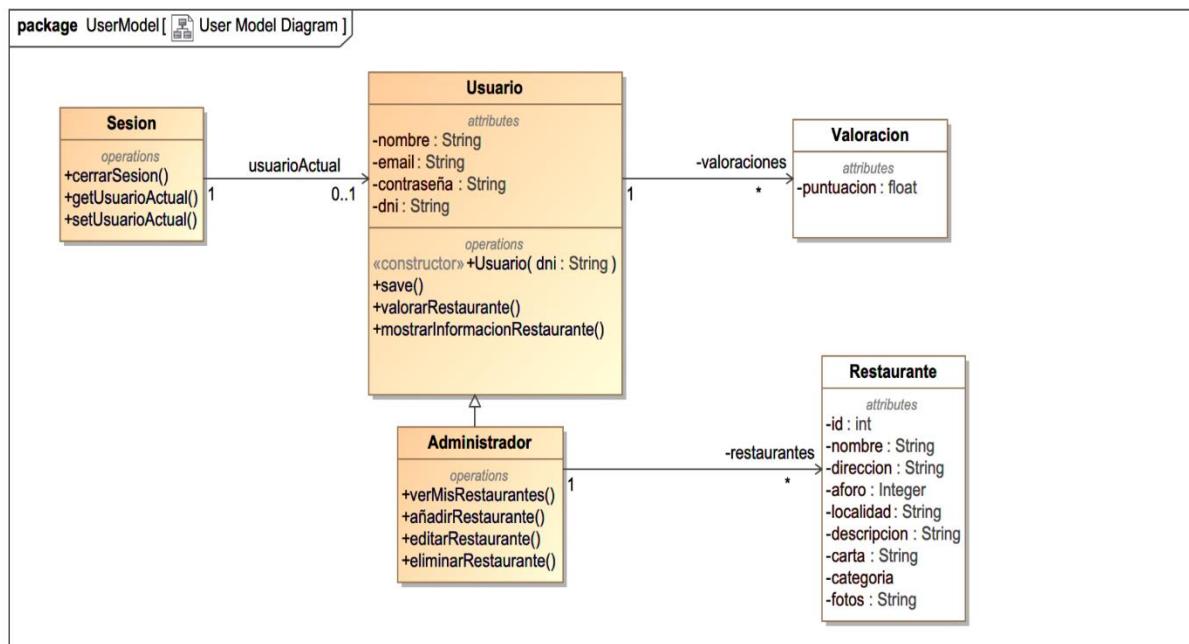
- **Restaurante:** almacenará los datos relativos a un restaurante registrado en el sistema.
- **Historial:** mantiene la información relativa al número de personas que estuvieron en el restaurante, controlando el tiempo medio que estuvieron y el porcentaje de aforo.
- **Cola:** registra el número de personas que esperan su turno en la cola.
- **Gestión de barrera:** guardará el número de acompañantes del cliente registrado en el establecimiento.
- **Valoración:** almacenará la puntuación que el cliente asigne al restaurante al emitir su opinión.

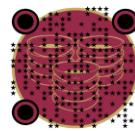


Modelo de usuario

En nuestro sistema, el modelo de usuario tiene dos objetivos diferenciados:

- Contiene las clases que definen qué información es almacenada en el contexto de una sesión. En este caso práctico una sesión está formada por el **usuario actual** y el **administrador**.
- Estas clases contenidas proveen de operaciones que pueden ser usadas en el proceso de negociación de procesos. El comportamiento de estas operaciones no es modelado, pero tiene que ser implementado por separado.



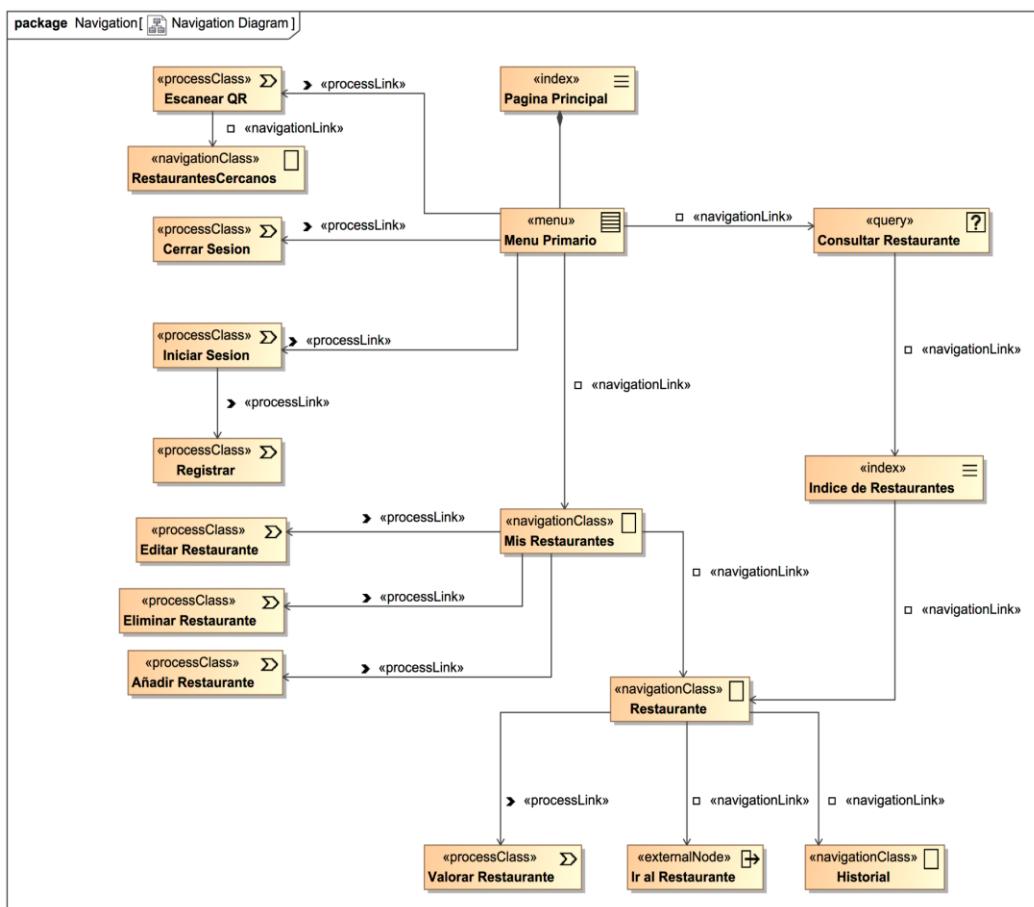


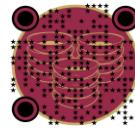
Modelo de navegación

Mediante este modelado, hemos indicado como el sistema de páginas web del sitio está relacionado internamente. Es decir, como se enlazan los elementos de navegación.

Para ello hemos utilizado unidades de navegación llamadas “nodos” conectadas por enlaces de navegación.

- Empezamos con la página de inicio, en nuestro modelo denominada **página principal**. La página de inicio enlaza con **Menu Primario**, que sería nuestra página de índice.
 - Desde la clase **Menu Primario** enlazamos con la clase **Consultar Restaurante**, que es un proceso predefinido, y con las clases **Cerrar Sesion**, **Escanear QR**, **Iniciar Sesion** y **Mis Restaurantes**, que son procesos no definidos.
 - Para finalizar vemos que el resto de contenido mantiene la estructura de conexión entre páginas planteada.





Modelado de proceso

Mediante el modelado de proceso especificamos las acciones que realiza cada clase de proceso en nuestro sistema.

Se divide en:

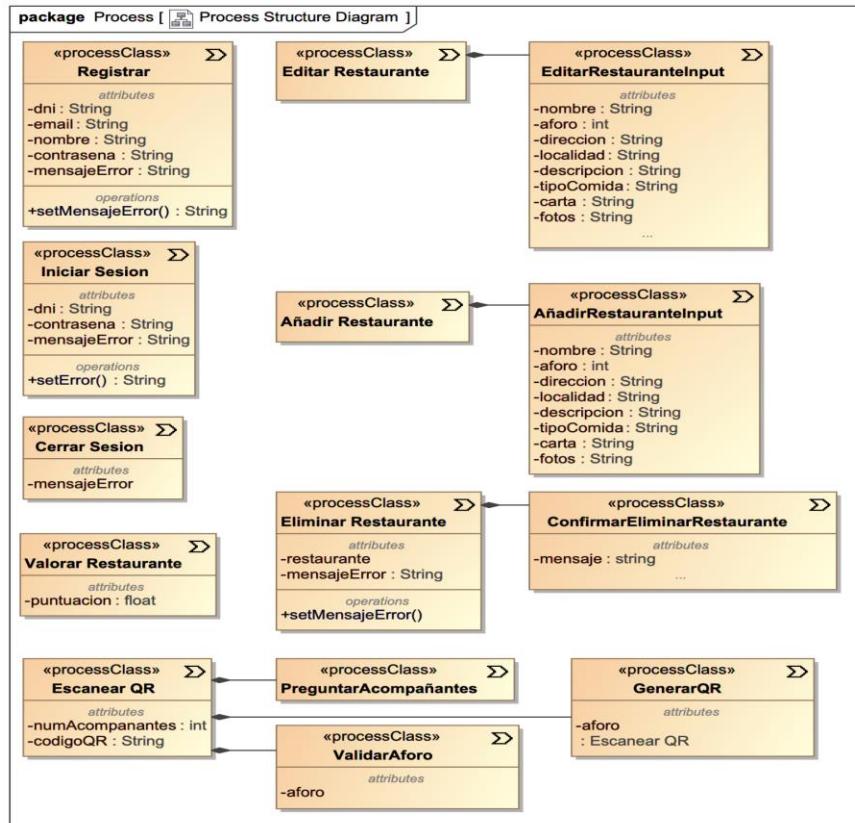
Modelo de estructura del proceso

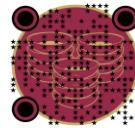
Definimos las relaciones entre las diferentes clases proceso.

Para este fin, creamos un diagrama de clases, usando la transformación de navegación a estructura de proceso que permite MagicUWE, y agregaremos las clases que resultan necesarias y no aparecen predefinidas, como

GenerarQR, Validar Aforo, Preguntar Acompañantes...

Del mismo modo, agregamos otras clases para expresar que las clases asociadas incluyen una confirmación con una pregunta. Esto significa, por ejemplo en el caso de **Eliminar Restaurante**, que si el usuario desea eliminar un restaurante, se mostrará un mensaje y después de hacer clic en aceptar, se eliminará el contacto.





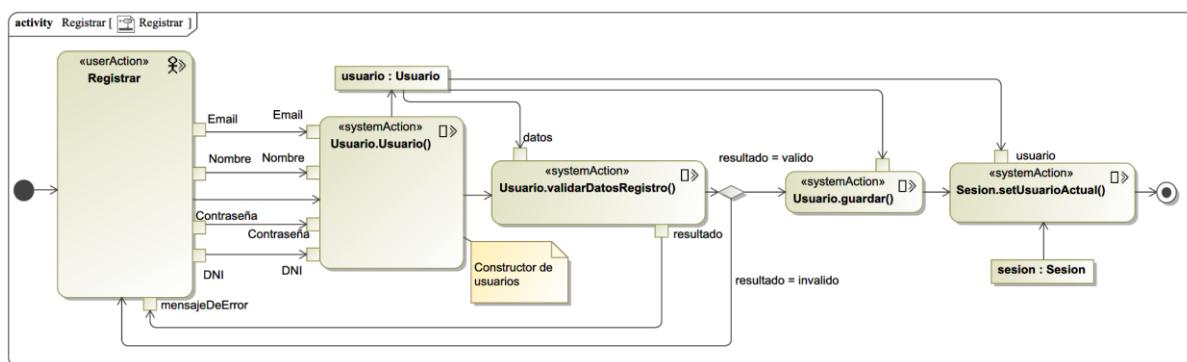
Modelo de flujo del proceso

Especificamos las actividades conectadas con cada proceso.

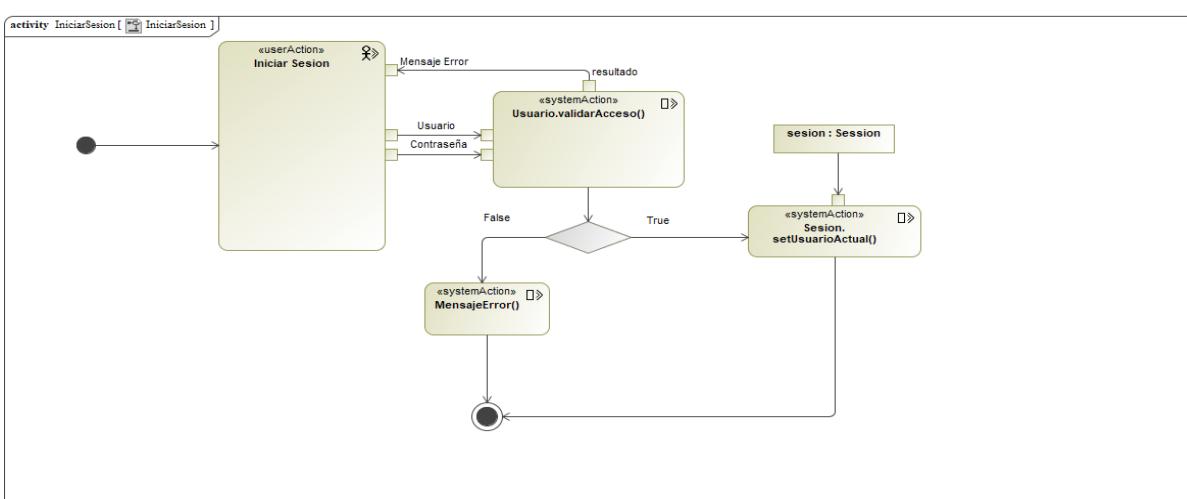
Describimos los comportamientos de una clase proceso, lo que ocurre en detalle dentro de cada una. Para ello nos valemos de un diagrama de actividades.

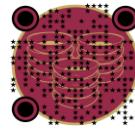
Los modelos se representan con nodos de objetos y las transformaciones como actividades estereotipadas, una cadena de transformaciones define el flujo de control.

registrar

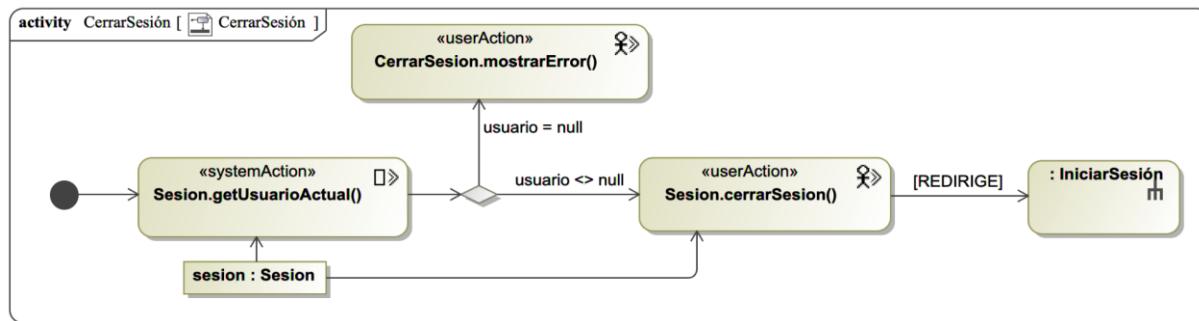


iniciarSesion

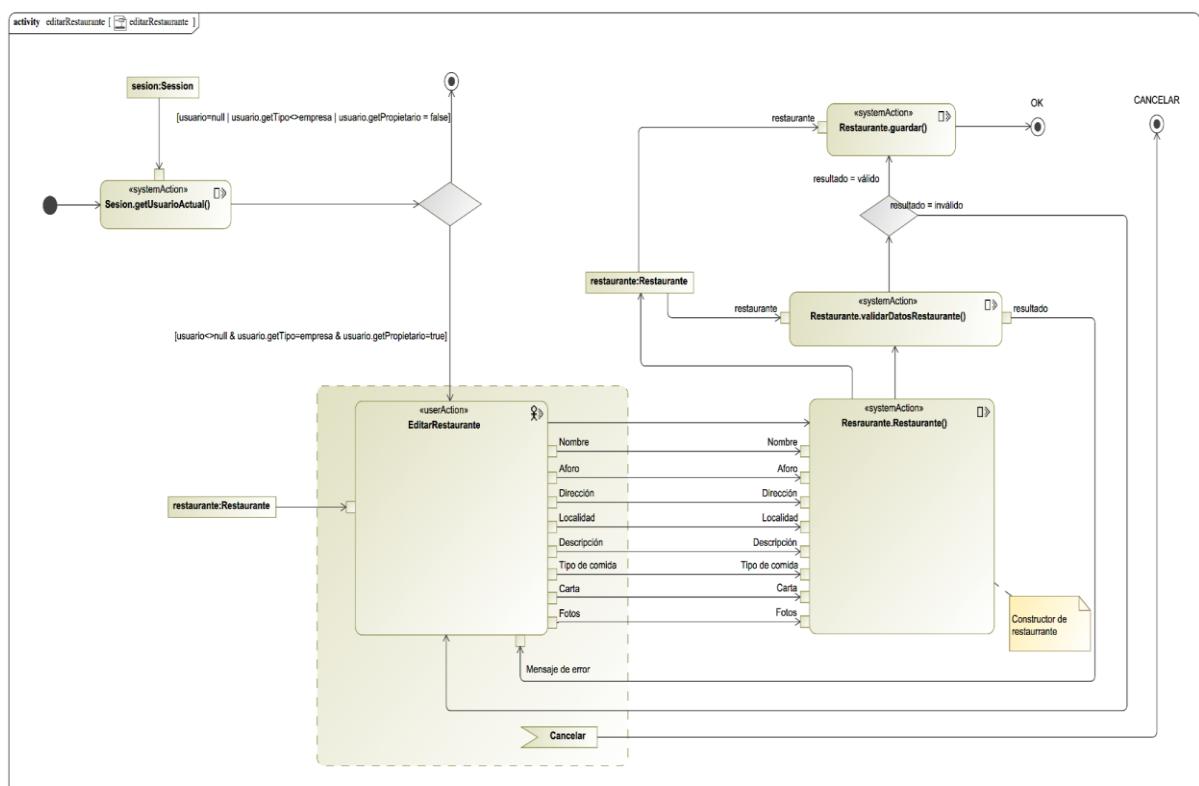


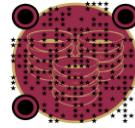


cerrarSesión

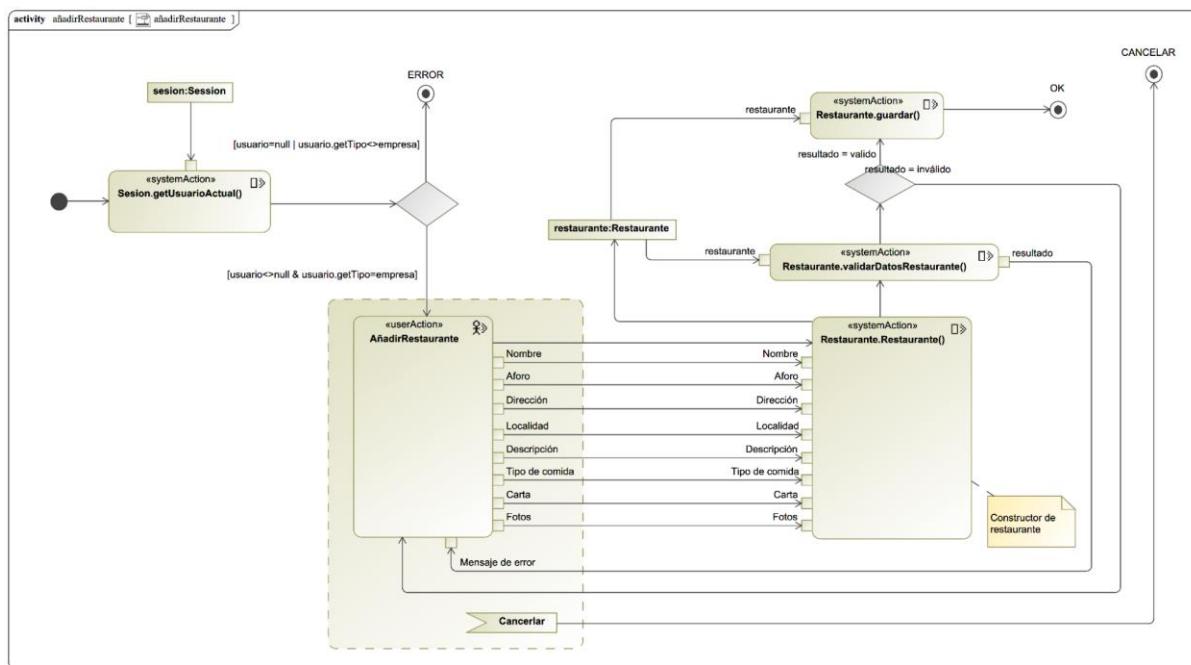


editarRestaurante

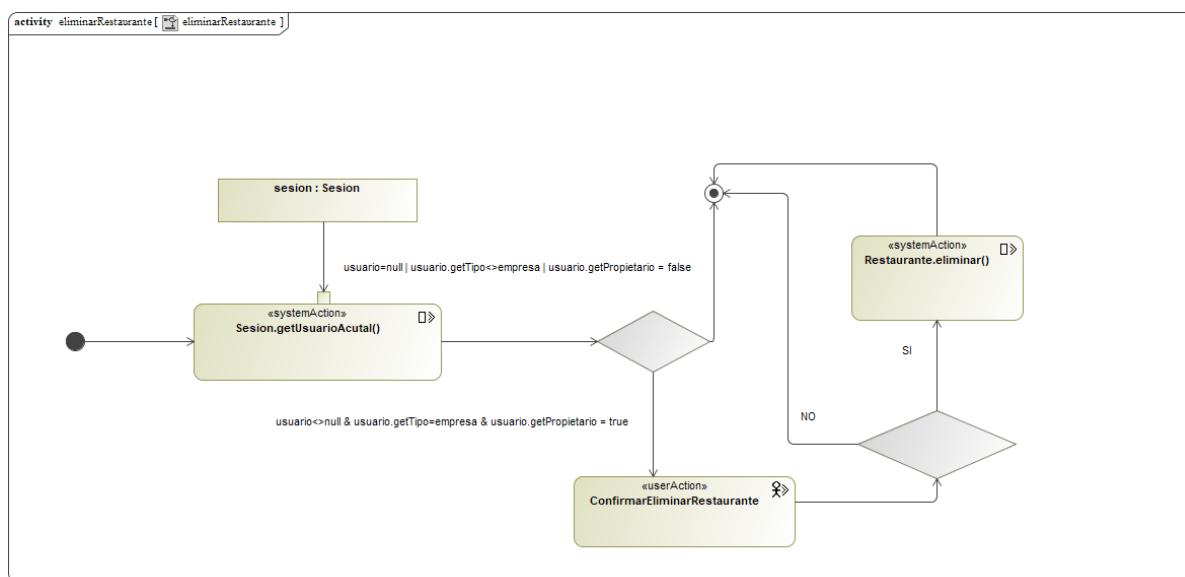


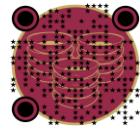


añadirRestaurante

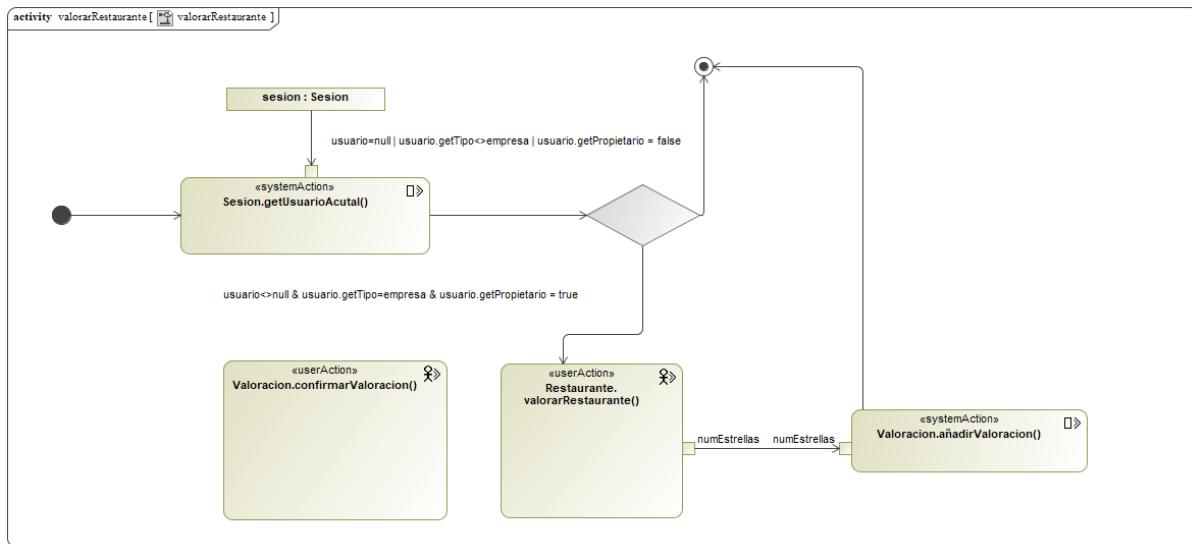


eliminarRestaurante

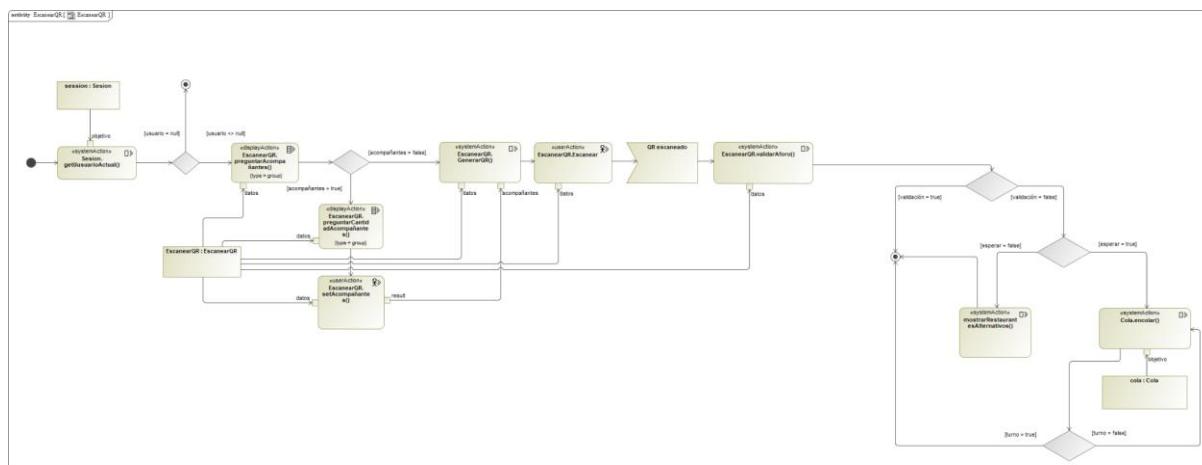


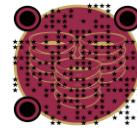


valorarRestaurante



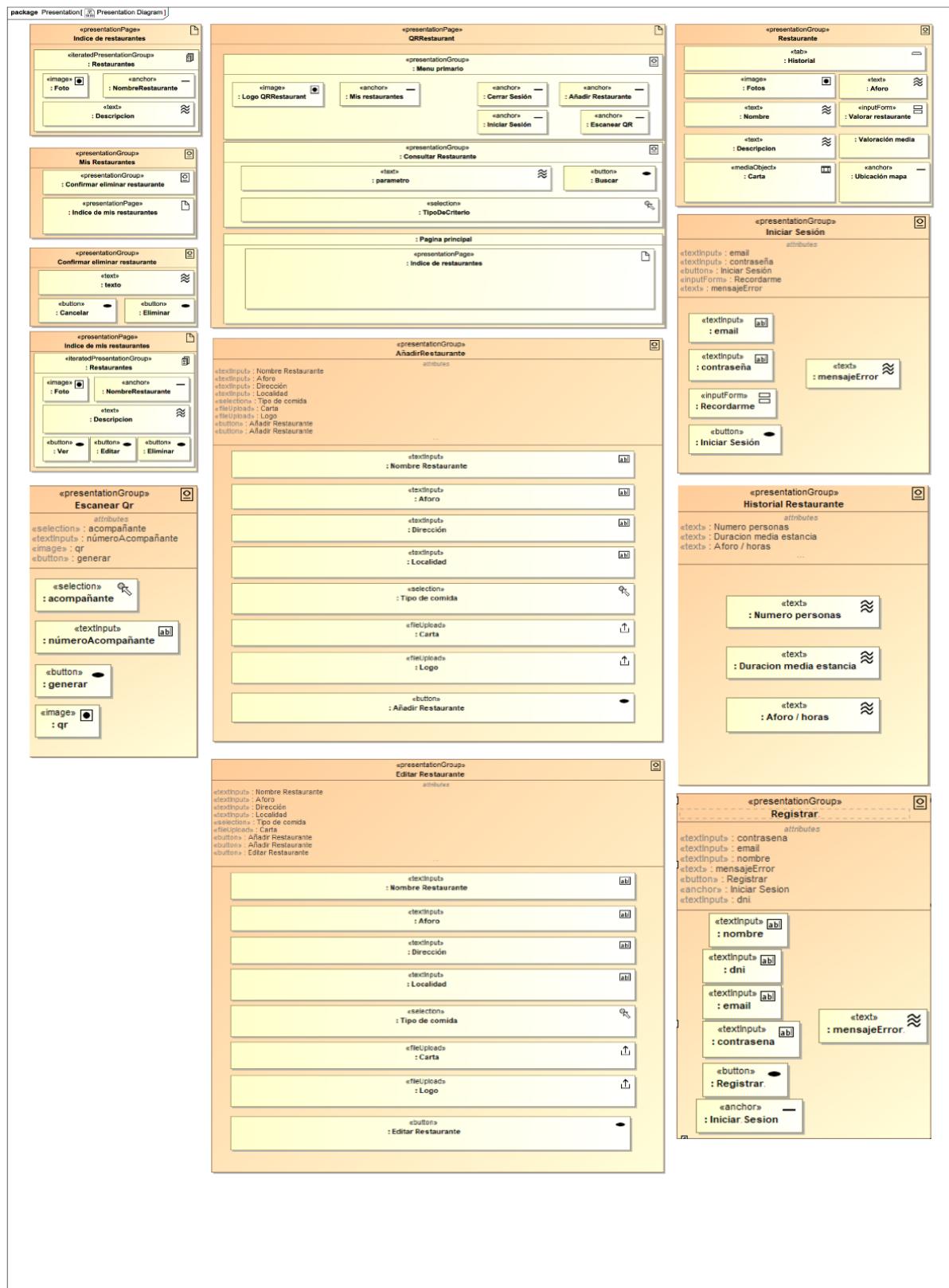
escanearQR

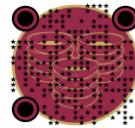




Modelado de presentación

Representamos las clases de navegación de procesos que pertenecen a cada página web.





PRÁCTICA 4

User eXperience

Para realizar estas pruebas, hemos seleccionado usuarios potenciales de la web, y hemos hecho pruebas de esta en diferentes dispositivos y buscadores para detectar posibles errores, comprobar compatibilidades y detectar sugerencias y detalles que echa en falta el usuario

Todos los usuarios deberán de enfrentarse a las siguientes pruebas sin ayuda del desarrollador.

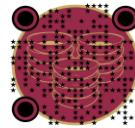
Al final de las pruebas se realizarán unas preguntas a los usuarios y deberán de valorar la web de 0 a 10. Las preguntas que deben responder son:

- **Útil:** ¿Responde a nuestras necesidades?
- **Cómoda:** ¿Se encuentra todo perfectamente?
- **Usable:** ¿Es fácil de utilizar y entender?
- **Deseable:** ¿Recomienda su uso?

- **Día 1:**

- **Usuario 1 (Móvil: Xiaomi - Navegador: navegador de Xiaomi - Android)**

- 1) **Registrarse en el sistema:** el usuario no encuentra el botón para registrarse (sugerencia, poner un botón al lado de iniciar sesión donde aparezca registrarse).
- 2) **Iniciar sesión:** todo correcto.
- 3) **Buscar por tipo de comida:** todo correcto, no utiliza el filtro de comida en el buscador, utiliza tipo de comida en el header.
- 4) **Buscar por nombre:** todo correcto.
- 5) **Buscar por localización:** todo correcto. (Sugerencia, mostrar un mensaje de error si no encuentra restaurante).



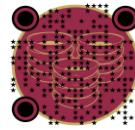
-
- 6) Buscar por valoración:** el usuario no sabe que hay que escribir (sugerencia, al filtrar por valoración, crear un desplegable con los cinco niveles, 1-5 estrellas).
 - 7) Escanear QR:** todo correcto.
 - 8) Ver restaurante:** todo correcto.
 - 9) Valorar restaurante:** todo correcto.
 - 10) Carta:** no es compatible, no se puede ver.
 - 11) Cerrar sesión:** todo correcto.

Experiencia del usuario (0-10)

- **Útil:** Sí (8)
- **Cómoda:** Sí (7)
- **Usable:** No (5)
- **Deseable:** Sí

Sugerencias finales:

- 1) En inicio le gustaría que apareciese una descripción que explique a qué se dedica QrRestaurant.**
 - 2) Le gustaría que hubiese instrucciones de uso (no sabe que tiene que estar registrado para acceder a un restaurante, no sabe que tiene que iniciar sesión para poder escanear el QR).**
-
- **Usuario 2 (PC - Navegador: Opera GX - Windows)**
 - 1) Registrarse en el sistema:** todo correcto.
 - 2) Iniciar sesión:** todo correcto.
 - 3) Buscar por tipo de comida:** todo correcto, no utiliza el filtro de comida en el buscador, utiliza tipo de comida en el header.
 - 4) Buscar por nombre:** todo correcto.



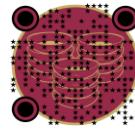
- 5) **Buscar por localización:** todo correcto. (Sugerencia, mostrar un mensaje de error si no encuentra restaurante).
- 6) **Buscar por valoración:** el usuario no sabe que hay que escribir.
- 7) **Escanear QR:** todo correcto.
- 8) **Ver restaurante:** todo correcto.
- 9) **Valorar restaurante:** todo correcto.
- 10) **Carta:** todo correcto.
- 11) **Cerrar sesión:** todo correcto.

Experiencia del usuario (0-10)

- **Útil:** Sí (9)
- **Cómoda:** Sí (10)
- **Usable:** Sí (10)
- **Deseable:** Sí

Sugerencias finales:

- **Usuario 3 (iPad - Navegador: safari - IOS)**
 - 1) **Registrarse en el sistema:** todo correcto.
 - 2) **Iniciar sesión:** todo correcto.
 - 3) **Buscar por tipo de comida:** todo correcto, no utiliza el filtro de comida en el buscador, utiliza tipo de comida en el header.
 - 4) **Buscar por nombre:** todo correcto.
 - 5) **Buscar por localización:** todo correcto. (Sugerencia, mostrar un mensaje de error si no encuentra restaurante).



- 6) **Buscar por valoración:** el usuario no sabe que hay que escribir.
- 7) **Escanear QR:** todo correcto.
- 8) **Ver restaurante:** todo correcto.
- 9) **Valorar restaurante:** todo correcto.
- 10) **Carta:** no es compatible, sólo aparece la primera página de la carta.
- 11) **Cerrar sesión:** todo correcto.

Experiencia del usuario (0-10)

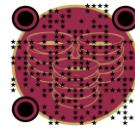
- **Útil:** Sí (7)
- **Cómoda:** Sí (9)
- **Usable:** Sí (10)
- **Deseable:** Sí

Sugerencias finales:

- 1) En inicio le gustaría que apareciese una descripción que explique a qué se dedica QrRestaurant.
-
- **Día 2:**

Cambios realizados tras el primer día de pruebas:

- Escanear QR en la página de inicio (no se puede acceder si no has iniciado sesión).
- Mensaje explicativo en inicio.
- Página de FAQs.
- Pantallas de error.
- Nuevo filtro de valoración, de menor a mayor y de mayor a menor.



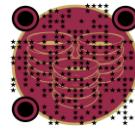
- Nueva forma de mostrar el aforo total, los sitios disponibles y la cola.
- **Usuario 1 (Móvil: Xiaomi - Navegador: navegador de Xiaomi - Android)**
 - 1) Registrarse en el sistema:** todo correcto.
 - 2) Iniciar sesión:** todo correcto.
 - 3) Buscar por tipo de comida:** todo correcto, pero sigue sin gustarle que aparezca uno predefinido.
 - 4) Buscar por nombre:** todo correcto, sólo busca si pone el nombre completo.
 - 5) Buscar por localización:** todo correcto.
 - 6) Buscar por valoración:** todo correcto, pero deberíamos mostrar desde fuera la valoración.
 - 7) Escanear QR:** todo correcto.
 - 8) Ver restaurante:** todo correcto.
 - 9) Valorar restaurante:** todo correcto, debería aparecer cuántas personas han valorado el restaurante.
 - 10) Carta:** todo correcto.
 - 11) Cerrar sesión:** todo correcto.

Experiencia del usuario (0-10)

- **Útil:** Sí (8)
- **Cómoda:** Sí (8)
- **Usable:** Sí (7)
- **Deseable:** Sí

Sugerencias finales:

- 1)** Añadir nuestros correos en el apartado quiénes somos.



- 2) Deberíamos de implementar la verificación de la contraseña (repetir al registrar) y añadir he olvidado mi contraseña.

- **Usuario 2 (PC - Navegador: Opera GX - Windows)**

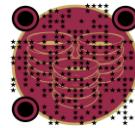
- 1) **Registrarse en el sistema:** todo correcto.
- 2) **Iniciar sesión:** todo correcto.
- 3) **Buscar por tipo de comida:** todo correcto.
- 4) **Buscar por nombre:** todo correcto.
- 5) **Buscar por localización:** todo correcto.
- 6) **Buscar por valoración:** todo correcto.
- 7) **Escanear QR:** todo correcto.
- 8) **Ver restaurante:** todo correcto.
- 9) **Valorar restaurante:** todo correcto.
- 10) **Carta:** todo correcto.
- 11) **Cerrar sesión:** todo correcto.

Experiencia del usuario (0-10)

- **Útil:** Sí (10)
- **Cómoda:** Sí (10)
- **Usable:** Sí (10)
- **Deseable:** Sí

Sugerencias finales:

El usuario ha quedado completamente satisfecho.



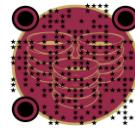
- **Usuario 3 (iPad - Navegador: safari - IOS)**

- 1) Registrarse en el sistema:** todo correcto.
- 2) Iniciar sesión:** todo correcto.
- 3) Buscar por tipo de comida:** todo correcto, no utiliza el filtro de comida en el buscador, utiliza tipo de comida en el header.
- 4) Buscar por nombre:** todo correcto.
- 5) Buscar por localización:** todo correcto.
- 6) Buscar por valoración:** todo correcto.
- 7) Escanear QR:** todo correcto.
- 8) Ver restaurante:** todo correcto.
- 9) Valorar restaurante:** todo correcto.
- 10) Carta:** todo correcto.
- 11) Cerrar sesión:** todo correcto.

Experiencia del usuario (0-10)

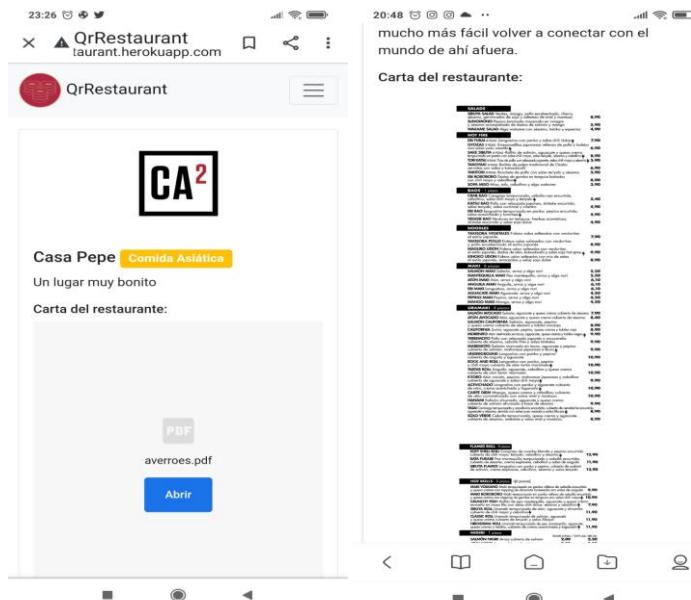
- **Útil:** Sí (8)
- **Cómoda:** Sí (10)
- **Usable:** Sí (10)
- **Deseable:** Sí

Sugerencias finales: - .



Cambios realizados tras las pruebas de UX

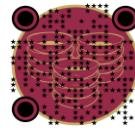
- Compatibilidad de la carta:



- Mensaje en la pantalla de inicio:

The screenshot shows the website's search functionality. At the top, there is a search bar with placeholder text "Introduce el nombre del restaurante", a "Buscar" button, and a dropdown menu labeled "Filtrar por nombre". Below the search bar, two restaurant cards are displayed:

- El Rinconcillo**: Includes the restaurant's logo, name, last update date (16/05/2021 10:32), and a brief description: "Disfruta de deliciosas tapas con historia, herencia de la cocina tradicional andaluza-mozárabe, que cautivan paladares de todo el mundo".
- Sibuya Urban Sushi Bar**: Includes the restaurant's logo, name, last update date (16/05/2021 10:32), and a brief description: "Sibuya Sushi Bar es ese 'peaceful place' donde puedes ir para desconectar. Sushi delicioso, atención al detalle y a la calidad y un ambiente chill único que hará que tu visita a nuestro restaurante te recargue las pilas y sea mucho más fácil volver a conectar con el mundo de afuera".



QrRestaurant Inicio Tipos de comida ▾ Preguntas frecuentes Quiénes somos Cerrar Sesión Escanear QR

Desde QrRestaurant queremos ayudar a los restaurantes a controlar el aforo. Visita nuestras secciones de [preguntas frecuentes](#) y [quiénes somos](#) para más información.

Introduce el nombre del restaurante Buscar Filtrar por nombre ▾

EL RINCONCILLO

El Rinconcillo
Última actualización: 16/05/2021 10:32

Disfruta de deliciosas tapas con historia, herencia de la cocina tradicional andaluza-mozárabe, que cautivan paladares de todo el mundo

SIBUYA
URBAN SUSHI BAR

Sibuya Urban Sushi Bar
Última actualización: 16/05/2021 10:32

Sibuya Sushi Bar es ese "peaceful place" donde puedes ir para desconectar. Sushi delicioso, atención al detalle y a la calidad y un ambiente chill único que hará que tu visita a nuestro restaurante te recargue las pilas y sea mucho más fácil volver a conectar con el mundo de ahí afuera.

● Buscar por valoración:

20:59 Miércoles 12 de mayo No seguro — qrrestaurant.herokuapp.com 86 %

QrRestaurant Inicio Tipos de comida ▾ Quienes somos Iniciar Sesión

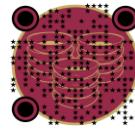
Búsqueda por Valoración del Restaurante Buscar Filtrar por valoración ▾

Casa Pepe
Última actualización: Tue May 11 2021 22:43:41 GMT+0000 (Coordinated Universal Time)

Un lugar muy bonito

Casa Juan
Última actualización: Sun May 09 2021 00:36:07 GMT+0000 (Coordinated Universal Time)

Un lugar muy feo

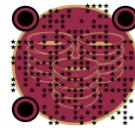


The screenshot shows a mobile browser displaying the QrRestaurant website. At the top, there is a navigation bar with links for 'Inicio', 'Tipos de comida', 'Preguntas frecuentes', 'Quiénes somos', 'Iniciar Sesión', and 'Escanear QR'. Below the navigation, there is a dropdown menu for filtering results by rating, with options 'Mayor a menor valoración' (selected) and 'Menor a mayor valoración'. A message from the app encourages users to control their restaurant's capacity by visiting sections like 'preguntas frecuentes' and 'quiénes somos'. Below this, there are two cards for restaurants:

- El Rinconcillo**: A traditional Andalusian tapas restaurant. It has a logo featuring a red circle with a white 'R', a black banner with 'EL RINCONCILLO' in white, and a detailed description: 'Disfruta de deliciosas tapas con historia, herencia de la cocina tradicional andaluza-mozárabe, que cautivan paladares de todo el mundo'. It was last updated on 16/05/2021 at 10:32.
- Sibuya Urban Sushi Bar**: An Asian restaurant. It has a logo with a stylized 'X' and the word 'SIBUYA' in large letters, with 'URBAN SUSHI BAR' below it. A description states: 'Sibuya Sushi Bar es ese "peaceful place" donde puedes ir para desconectar. Sushi delicioso, atención al detalle y a la calidad y un ambiente chill único que hará que tu visita a nuestro restaurante te recargue las pilas y sea mucho más fácil volver a conectar con el mundo de ahí afuera.' It was last updated on 16/05/2021 at 10:32.

● Buscar por tipo de comida:

This screenshot shows a comparison between two versions of the QrRestaurant mobile application interface. Both screens show a search bar with 'Asiática' selected and a 'Buscar' button. A dropdown menu is open on both screens, listing four filtering options: 'Filtrar por nombre' (radio button), 'Filtrar por tipo de comida' (radio button selected), 'Filtrar por localización' (radio button), and 'Filtrar por valoración' (radio button). The right-hand screen is a darker version of the same interface, showing the same dropdown menu and the same restaurant card for 'El Rinconcillo'.



Pruebas Usabilidad

Una vez realizada la primera versión de la web, el equipo se reunió y cada uno hizo una prueba buscando errores de usabilidad. En esta prueba hemos hecho un recorrido exhaustivo por toda la web buscando errores en las fuentes, faltas ortográficas, etc.

- No hay página de error si llegamos a una URL no contemplada.
- No hay mensajes de error en algunos formularios.
- No ha sido creada la página de FAQs.
- Errores en los roles de quienes somos.
- No hay verificaciones en los formularios.

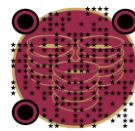
Una vez localizados estos errores, fueron corregidos, quedando la usabilidad de la web mucho mejor.

Pruebas de compatibilidad

Lo más importante para nuestra web es que sea compatible con sistemas móviles, ya que es una web que tendrás que usar estrictamente en un restaurante. Además de haberla probado en PC y en distintos navegadores, hemos comprobado la compatibilidad en diferentes dispositivos móviles.

Para desarrollar la web hemos utilizado Bootstrap para el frontend, que hace que sea responsive y compatible con todos los navegadores y dispositivos. También hemos utilizado JavaScript para el backend y xhr para el acceso a api. Gracias a esto nos aseguramos una mejor compatibilidad.

A continuación, se muestra una tabla de dispositivos y navegadores probados para comprobar la compatibilidad:



Dispositivo	Navegador	Compatibilidad
MacBook Pro / iMac	Google Chrome	Totalmente compatible
MacBook Pro / iMac	Safari	Totalmente compatible
iPhone	Safari	Totalmente compatible
iPad	Safari	Totalmente compatible
iPad	Google	Totalmente compatible
Xiaomi	Navegador	Totalmente compatible
Xiaomi	Google	Totalmente compatible
Windows	Opera GX	Totalmente compatible
Windows	Google Chrome	Totalmente compatible
Windows	Microsoft edge	Totalmente compatible

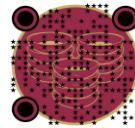
Pruebas de rendimiento

Para probar la velocidad de la web hemos utilizado el servicio de Google de PageSpeed Insight

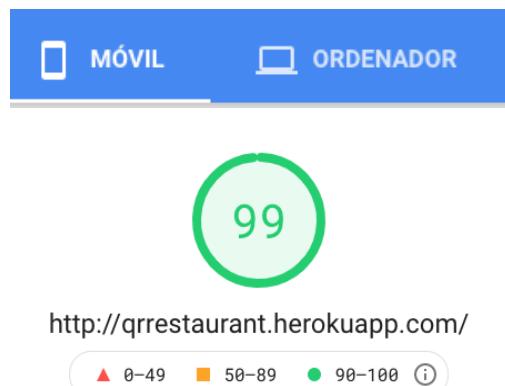
(<https://developers.google.com/speed/pagespeed/insights/>) que prueba la velocidad en general de la web en dispositivos móviles y PC. Se adjuntan resultados de ambos tests. Esta página da información muy detallada de la web. Para más información consultar la web de arriba y pegar la URL de la web de QrRestaurant (<http://qrrestaurant.herokuapp.com/>).

Velocidad en PC



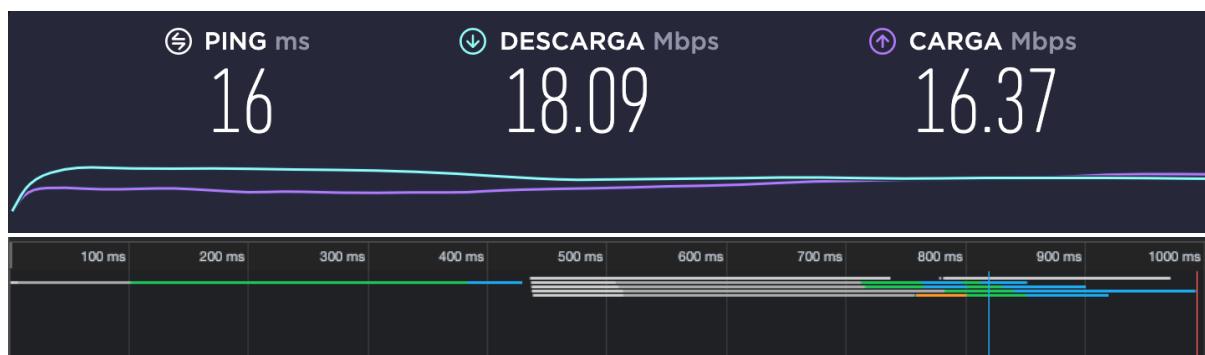


Velocidad en móvil

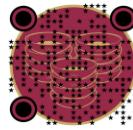


Para determinar el rendimiento de la web se han realizado pruebas a diferentes velocidades.

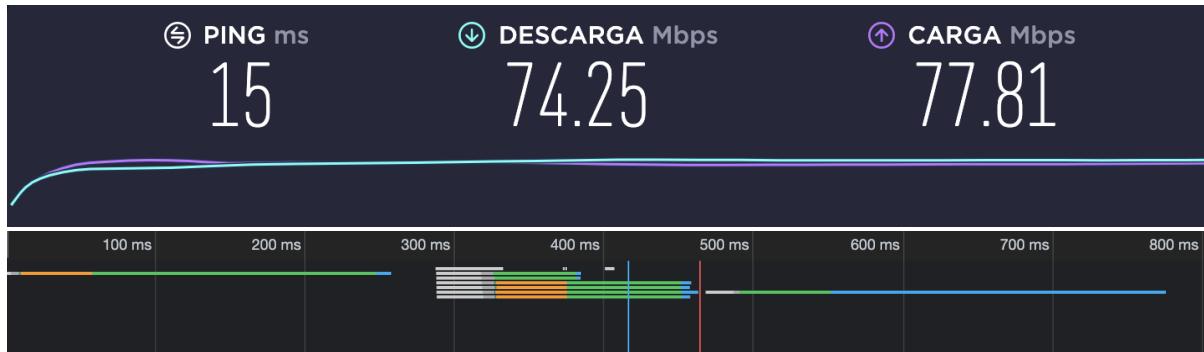
1.



Es la única que ha sufrido “problemas” al acceder a la vista de un restaurante, ya que tiene que cargar varias fotos y el pdf de la carta, pero no es reseñable porque tarda unos 2 segundos en hacer la carga total de la vista, en el peor de los casos.

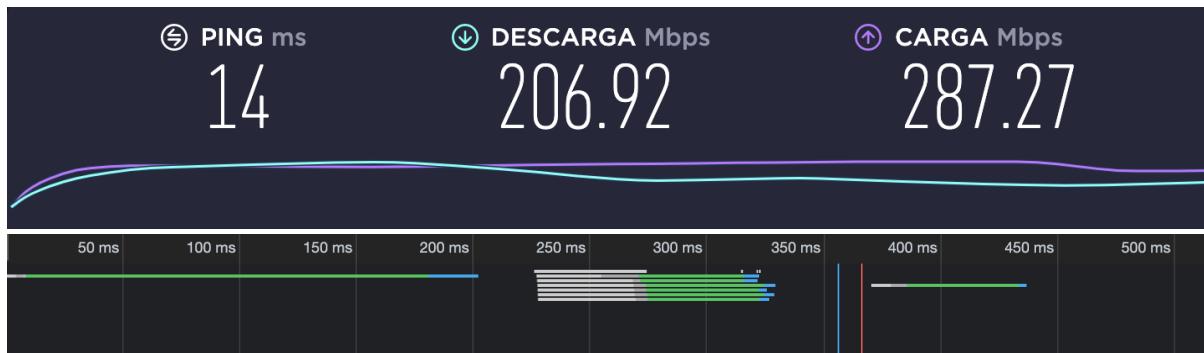


2.



Sin problemas (tarda de media un segundo en cargar).

3.



Sin problemas.

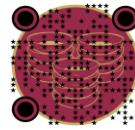
Pruebas de seguridad

Las pruebas de seguridad que hemos realizado son con referencia a las URL, es decir, hemos controlado que no se pueda acceder a donde no se debe si no has iniciado sesión o no eres administrador.

Otra prueba realizada es a los administradores, no puedes acceder a la parte de edición de los restaurantes si no eres administrador del mismo, y por supuesto, no podrás borrar los restaurantes de otro administrador.

Para mayor seguridad, al escribir la contraseña aparece en oculta y hemos implantado que tenga que tener al menos 8 caracteres.

Hemos comprobado que no puedes acceder a las funciones de administrador si no eres un usuario administrador o no has iniciado sesión.



Cuando cierras sesión no puedes acceder a las zonas restringidas.

La última prueba de seguridad ha sido encriptar las contraseñas (tercer campo), en la siguiente imagen podemos ver como se muestra encriptada en la base de datos.

A screenshot of a database table showing user data. The password column contains the value '\$2b\$10\$rP2HxOE0wFx6lf7TSCv1mutlylmLnVVc3ma5nf8rxBq...'. Below the table, there is a timestamp: 1 2021-05-18 15:42:57 2021-05-18 15:42:57.

Pruebas de Base de datos

Hemos creado unos tests de CRUD para probar el rendimiento de la base de datos, para ejecutarlos tendremos que escribir en la terminal **npm run tests**.

Para ejecutar un solo test, en el directorio tests, archivo index.js, comentaremos los que no nos interese probar.

Consultas simples

Tests de usuarios

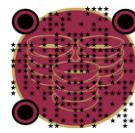
- **Test 1:** crear usuario
- **Test 2:** consultar usuario

```
Resultados TESTS [USUARIO]
2 TOTAL - TESTS [USUARIO]
2 OK - TESTS [USUARIO]
0 ERROR - TESTS [USUARIO]
```

Tests de restaurantes

- **Test 1:** crear restaurante
- **Test 2:** consultar restaurante por nombre
- **Test 3:** consultar restaurante por localización
- **Test 4:** actualizar restaurante
- **Test 5:** eliminar restaurante

```
Resultados TESTS [RESTAURANTE]
5 TOTAL - TESTS [RESTAURANTE]
5 OK - TESTS [RESTAURANTE]
0 ERROR - TESTS [RESTAURANTE]
```



Consultas masivas

Test de usuarios

- **Test 1:** crear 1000 usuarios simultáneos

Se ha realizado 5 veces el test, obteniendo un tiempo medio de 54,4288 segundos, por lo que podemos deducir que en un minuto se pueden registrar 1102 usuarios.

Tests de restaurantes

- **Test 1:** crear 50 restaurantes simultáneos

Se ha realizado 5 veces el test, obteniendo un tiempo medio de 2,4538 segundos, por lo que podemos deducir que en un minuto se pueden añadir 1222 restaurantes.

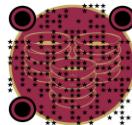
[**Nota:** para más detalles sobre los tests de la base de datos se debe consultar el código]

Integridad referencial

Hemos realizado diferentes pruebas para comprobar la integridad de la base de datos. Por ejemplo, no podremos crear historiales de restaurantes que no existan, no podemos crear un restaurante y que su dueño sea un DNI de un usuario que no esté registrado, etc.

En la siguiente imagen hay un ejemplo del error, dónde se detalla, que este se debe al hecho de que una restricción de clave foránea falla y por tanto no se puede crear.

```
DatabaseError [SequelizeDatabaseError]: Cannot add or update a child row: a foreign key constraint fails
at Query.formatError (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/sequelize/lib/dialects/mysql/query.js:265:16)
at Query.run (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/sequelize/lib/dialects/mysql/query.js:77:18)
at processTicksAndRejections (node:internal/process/task_queues:94:5)
at async /Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/sequelize/lib/sequelize.js:619:16
at async MySQLQueryInterface.insert (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/sequelize/lib/dialects/abstract/query-interface.js:749:21)
at async Restaurant.save (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/sequelize/lib/model.js:3954:35)
at async Function.create (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/sequelize/lib/model.js:2207:12) {
parent: Error: Cannot add or update a child row: a foreign key constraint fails
  at Packet.asError (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/mysql2/lib/packets/packet.js:712:17)
  at Execute.execute (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/mysql2/lib/commands/command.js:28:26)
  at Connection.handlePacket (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/mysql2/lib/connection.js:425:32)
  at PacketParser.onPacket (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/mysql2/lib/connection.js:75:12)
  at PacketParser.executeStart (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/mysql2/lib/packet_parser.js:75:16)
  at Socket.<anonymous> (/Users/gil/Desktop/UCO/3/IW/QRRestaurant/node_modules/mysql2/lib/connection.js:82:25)
  at Socket.emit (node:events:378:20)
  at addChunk (node:internal/streams/readable:313:12)
  at readableAddChunk (node:internal/streams/readable:288:9)
  at Socket.Readable.push (node:internal/streams/readable:227:10) {
```

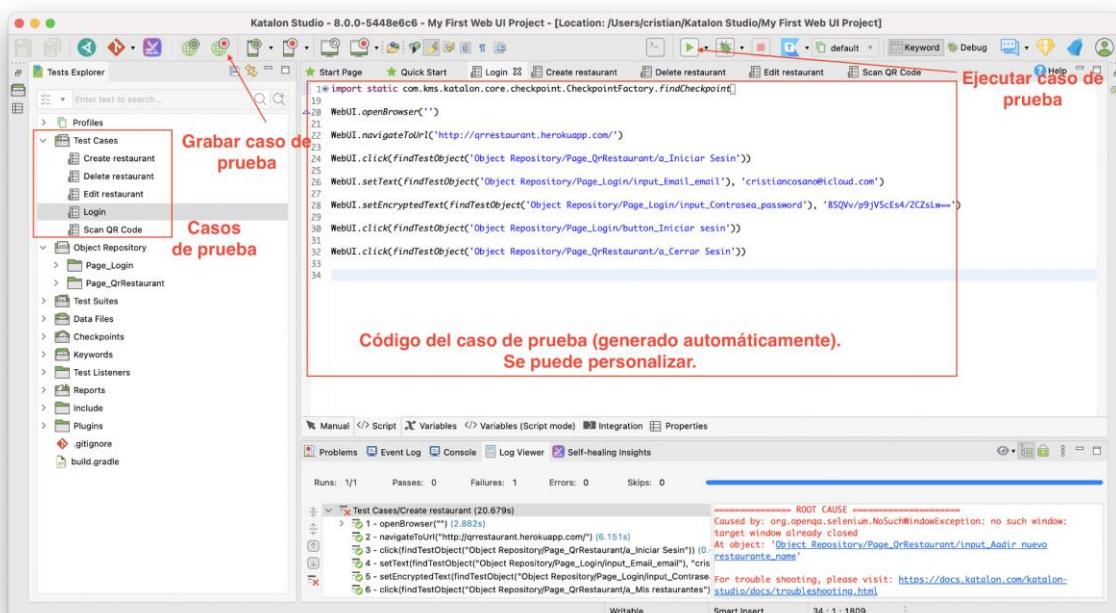


Esta imagen es sólo un ejemplo, se han realizado todas las pruebas posibles para asegurarnos de que cumplimos correctamente la integridad referencial de la base de datos.

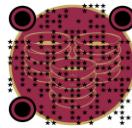
Pruebas de funcionalidad

Para la realización de pruebas de funcionalidad sobre la interfaz hemos utilizado Katalon Studio, un software con versión gratuita que nos permite grabar un conjunto de acciones que realicemos como usuarios, para después reproducirlas de forma automática, evitando tener que repetir los mismos pasos manualmente cada vez que cambiamos algo en el sistema.

En la imagen por ejemplo se ve el código del inicio de sesión que se utiliza para la acción de iniciar sesión en nuestra aplicación. Se abre el navegador, se dirige a la URL <http://qrrestaurant.herokuapp.com/>, hace click en iniciar sesión, introduce email, introduce contraseña, pulsa el botón iniciar sesión y posteriormente pulsa el botón cerrar sesión.



En esta segunda imagen vemos el navegador Chrome siendo controlado por Katalon Studio. A través de una extensión que nos aparece en esta instancia de Chrome, podremos controlar las acciones que se están realizando y detener la grabación cuando esté finalizada.



The screenshot shows a web browser window for 'QrRestaurant'. The top navigation bar includes links for 'Inicio', 'Mis restaurantes', 'Tipos de comida', 'Preguntas frecuentes', 'Quiénes somos', 'Cerrar Sesión', and 'Añadir Restaurante'. Below the navigation, there are two restaurant profiles:

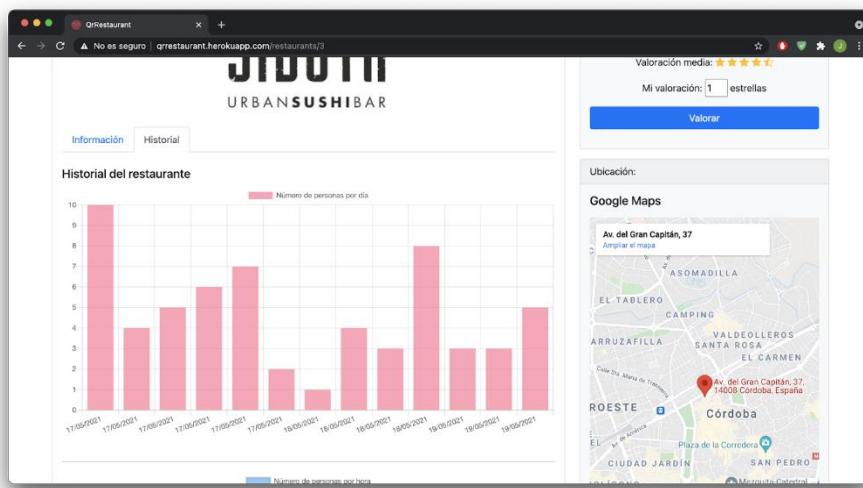
- Casa Pepe de la Judería**: Last updated 18/05/2021 19:24. Includes a logo featuring a diamond shape with 'TP' and the text 'RESTAURANTE CASA PEPE DE LA JUDERIA · Desde 1930 ·'. Buttons for 'Ver restaurante', 'Editar restaurante', and 'Eliminar restaurante' are present.
- El Rinconcillo**: Last updated 18/05/2021 19:24. Includes a logo with the text 'EL RINCONCILLO' and the URL 'qrrestaurant.herokuapp.com/restaurants/1'.

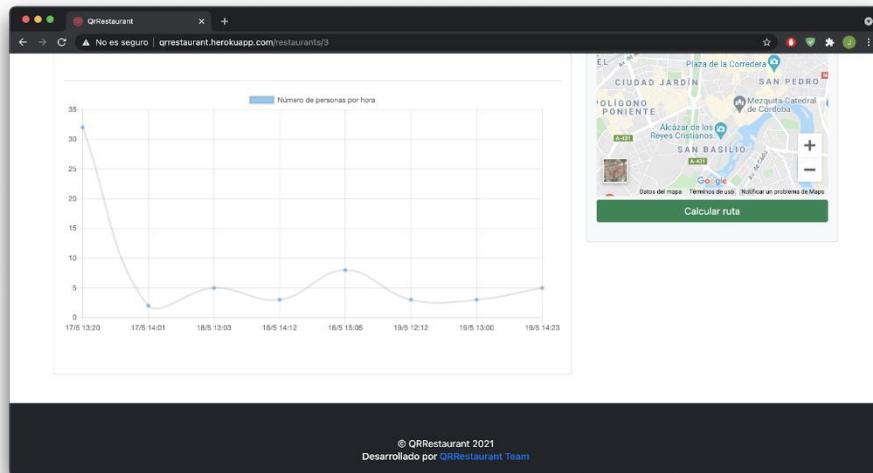
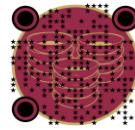
Cumplimiento de requisitos

Restaurante

- **RF1** - Cada restaurante debe incluir un historial que muestra estadísticas de los días anteriores.

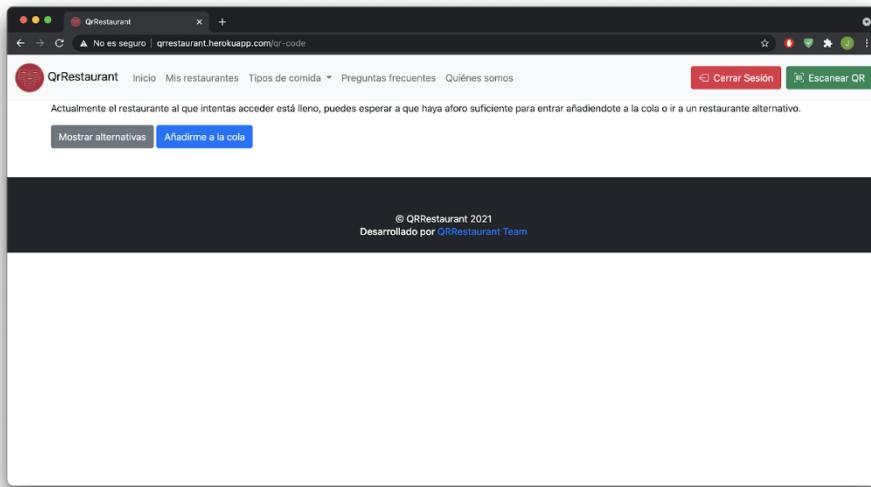
[Completado]





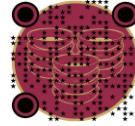
- **RF2** - El sistema debería prohibir la entrada a un usuario que intenta acceder a un local que supera el aforo.

[Completado]



- **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.

[Completado]



The screenshot shows a web browser window for 'QrRestaurant' with the URL 'qrrestaurant.herokuapp.com/restaurants/3/alternatives'. The page title is 'Alternativas al restaurante 'Casa Pepe de la Judería''. It displays information about the restaurant and a list of alternatives:

- Sibuya Urban Sushi Bar**
- Actualmente en el restaurante hay X personas. Puedes esperar o dirigirte a alguno de los siguientes restaurantes alternativos:
- Prefiero esperar. Añadirme a la cola.

The Sibuya Urban Sushi Bar entry includes its logo, a brief description, and a QR code.

- **RF4** - El sistema debería mostrar la distancia a la que se encuentra el restaurante.

[Completado]

The screenshot shows a web browser window for 'QrRestaurant' with the URL 'qrrestaurant.herokuapp.com/restaurants/3'. The page title is 'Carta del restaurante:'. It displays a menu and a Google Map:

CARTA DEL RESTAURANTE:

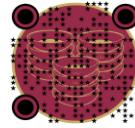
CATEGORÍA	PLATO	DESCRIPCIÓN	PRECIO	
SALADS	SIBUYA SALAD	Verdes, manzana, pepino, cebolla, aceite, chile, mayonesa, queso feta, nata, yogur y aderezo de mostaza y miel.	8,90	
	SUMOMONO	Pepino boligliado marinado en vinagre con cebolla, tomate, aceite de oliva y aderezo.	5,90	
	MARINATE SALAD	Algo diferente con lechuga, tomate, queso feta y especias.	4,90	
	HOT FIRE	EHI TUNA	Ensalada de atún langostino con pepero y salsa chili dulce.	7,90
		GYOZA	Empanadillas japonesas rellenas de pollo y buey, con salsa y wasabi.	6,90
		SABA MIRTA	Atún, salmón, aguacate y queso feta.	8,90
TOFU KATSU		Tofu de pollo con aliñado japonés, salsa de miso y arroz.	5,90	
MARINATE KATSU		Tofu de pollo con aliñado japonés, salsa de miso y arroz.	6,90	
RAMEN		Nudos de sopa con caldo de pollo y verduras.	5,90	
BAGOS	EHI BONCHORO	Dobles de gimbao en tempura bollos con atún y wasabi.	5,90	
	SOPA ANDO	Verde y roja con caldo de pollo y verduras.	3,90	
	BAOS	1 pieza		
	NOODLES	CRAB BAO	Congrio marinado, cebolla roja encocida, mantequilla de trufa y wasabi.	5,40
		KATSU BAO	Pollo con rebozado japonés, shitake encocido, mantequilla de trufa y wasabi.	4,90
		EHI BAO	Langostino tempurizado en pollo, pepino encocido, cebolla roja y kimchi.	4,90
VEGETABLE BAO		Verde y roja con verduras, hierbas aromáticas, mantequilla encocida y salsa soja dulce.	4,90	
ROLLS		MAXICRIS VEGITABLES	Fideos sobre selados con verduras al estilo japonés.	7,90
		DAIMONI	Póker sobre selados con verduras y pollo sazonado al estilo japonés.	8,90
	MAGURO UDON	Iberico sobre selados con verduras al estilo japonés, sopa de udon y salsa soja hot spicy.	9,90	
	KINKOKU UDON	Fideos sobre selados con mix de setas y verduras al estilo japonés y salsa soja dulce.	8,90	
	MAKI	8 piezas		
	SAUMÓN MAKI	Salmón, arroz y algas nori.	5,50	
MARINADERA MAKI	Puerro, manzana, arroz y algas nori.	5,50		
YAKI MAKI				

Google Maps:

A map showing the location of the restaurant at Av. del Gran Capitán, 37, 14008 Córdoba, España. The map also shows surrounding landmarks like the Alcázar de los Reyes Cristianos, Plaza de la Corredera, and Mezquita Catedral de Córdoba.

- **RF5** - El restaurante debería incorporar la carta de comida que se sirve.

[Completado]



The screenshot shows a web browser displaying the QrRestaurant app. On the left, there is a detailed menu for a restaurant, categorized into SALADS, HOT PIZZA, KOREAN, NOODLES, and KAKI. The menu items include various dishes like SIBUWA SALAD, EBI FUSAI, YAKISOBA, etc., with their descriptions and prices. On the right, a map of Córdoba, Spain, is shown with a red marker indicating the restaurant's location at Av del Gran Capitán, 37, 14008 Córdoba, España. Below the map is a green button labeled "Calcular ruta".

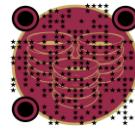
- **RF6** - El sistema debería permitir listar los restaurantes por calidad acorde a las opiniones de los usuarios.

[Completado]

The screenshot shows a list of restaurants on the QrRestaurant platform. At the top, there is a navigation bar with links for Inicio, Mis restaurantes, Tipos de comida, Preguntas frecuentes, and Quiénes somos. There are also buttons for Cerrar Sesión and Escanear QR. Below the navigation, a message encourages users to help restaurants control their客流量. The main area displays two restaurant profiles: "Casa Pepe de la Judería" and "Lhardy". Each profile includes the restaurant's logo, name, establishment year, last update, a brief welcome message, and a small note about its history or特色.

- **RF7** - El sistema debería mostrar una ficha con la información del restaurante.

[Completado]



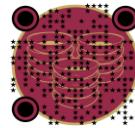
The screenshot shows a web browser window for 'QrRestaurant' with the URL 'qrrestaurant.herokuapp.com/restaurants/3'. The page displays information about the restaurant 'SIBUYA URBAN SUSHIBAR'. Key details include:

- Aforo:** The restaurant has a capacity of 42 people, with 42 available slots.
- Valoraciones:** Average rating is 4 stars. A user can leave a rating of 5 stars.
- Ubicación:** Google Maps showing the location at 'Av. del Gran Capitán, 37' in 'ASOMADILLA'.
- Información:** Describes Sibuya as a "peaceful place" for disconnecting, with delicious sushi and attentive service.
- Histórico:** Shows a history section with a single entry: 'SALADS'.

- **RF8** - El sistema debería restringir la salida del restaurante hasta que se pase el QR por el lector.

[Corrección]: A la hora de implementar el sistema, nos dimos cuenta de que no tenía sentido, ya que es un requisito para el sistema de barreras, que es externo al sistema en cuestión. Nuestra aplicación en lo único que satisfacería en parte este requisito, es facilitando la conexión con la barrera como se muestra en la captura.





- **RF9** - El sistema debería de contar con un filtro que muestre los restaurantes con aforo disponible.

[Completado]

The screenshot shows a web-based application interface for QrRestaurant. At the top, there is a navigation bar with links for 'Inicio', 'Tipos de comida', 'Preguntas frecuentes', and 'Quiénes somos'. On the right side of the header, there are buttons for 'Iniciar Sesión' and 'Escanear QR'. Below the header, there is a search bar with the placeholder 'Introduce el nombre de...'. The main content area displays three restaurant cards:

- Casa Pepe de la Judería**: Logo featuring a diamond shape with 'TP' inside, establishment date 'Desde 1930', and a welcome message: 'Bienvenidos a Casa Pepe de la Judería, el restaurante ideal para sumergirse en la riqueza culinaria cordobesa'.
- Lhardy**: Logo featuring a stylized 'L', establishment date 'Desde 1930', and a welcome message: 'Con el ornato de esta bella fachada definida por el gusto del Segundo Imperio que vuelve ahora a cautivarnos, Lhardy ha sabido conservar celosamente el ambiente cortesano y aristocrático del Madrid del siglo XX, y los comienzos del XXI al mismo tiempo que las mejores fórmulas de la cocina europea'.
- Sibuya Urban Sushi Bar**: Logo featuring a stylized 'X', establishment date 'Desde 1930', and a welcome message: 'Churrasco, Sushi, Doner, Hamburguesas, Frikadeller, Pizzas, Ensaladas, Postres, Bebidas y más'.

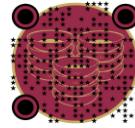
- **RF10** - El sistema debería de contar con filtro que muestre los restaurantes por tipo de comida.

[Completado]

This screenshot shows the same web-based application interface for QrRestaurant, but with a different layout. The filter menu 'Tipos de comida' is now visible at the top left, showing categories like 'Mediterránea', 'Española', 'Asiática', 'Bar de tapas', and 'Vanguardista'. The rest of the interface and the three restaurant cards are identical to the previous screenshot.

- **RF11** - El sistema debería mostrar en el mapa la ubicación del restaurante.

[Completado]



The screenshot shows a web browser displaying the QrRestaurant app. On the left, there is a menu for a restaurant with sections for SALADS, HOT FRIED, and BAOS. On the right, there is a Google Maps interface showing the location of the restaurant at Av. del Gran Capitán, 37, 14001 Córdoba, España.

Categoría	Plato	Precio
SALADS	SHIBUYA SALAD Verdes, mango, pollo espetado, cherr...	8,90
SALADS	SUNDOMACHO Pechuga fumetada marinada en vino...	9,90
SALADS	WAKAME SALAD Algas wakame con almejas, tébilo y espárrago	4,90
HOT FRIED	SHIBUYA CHICKEN Sándwich con pollo y salsa chile dulce	7,90
HOT FRIED	GYOCZA KATSU Empolvado japonés relleno de pollo y bacon	6,90
HOT FRIED	SAKE SUSHI A base de arroz, aguacate y queso crema	6,90
HOT FRIED	SHIBUYA CHICKEN Sándwich con pollo con salsa chile roja, albahaca y queso	8,90
HOT FRIED	TAKEAWAY KATSU Sándwich con pollo frito y salsa chile dulce	6,90
HOT FRIED	TAKITORI KATSU Brocheta de pollo con salsa teriyaki y wasabi	5,90
HOT FRIED	SHIBUYA KATSU Sándwich de gamba en tempura batida	8,90
HOT FRIED	SHIBUYA KATSU Sándwich de pollo frito y salsa chile dulce	6,90
HOT FRIED	SHIBUYA KATSU Sándwich de pollo frito y salsa wasabi	5,90
BAOS	CHAM BAOS Sándwich de pollo frito, cebolla roja encurtida, salsa chile roja y wasabi	5,40
BAOS	KATSU BAOS Pollo con rebozado japonés, cebolla encurtida, salsa chile roja y wasabi	4,90
BAOS	SHIBU BAOS Langostinos tempurados en panca, pepero encurtido, salsa chile roja y wasabi	4,90
BAOS	VEGUE BAOS Verduras en tempura, hierbas aromáticas, salsa chile roja y wasabi	4,90
NOODLES	TAKITORI VEGETALES Fideos sobre sofritos con verduras	7,90
NOODLES	TAKITORI POLLO Fideos sobre sofritos con verduras y pollo frito rebozado de estilo japonés	8,90
NOODLES	SHIBUYA UDON Fideos de trigo, caldo de pollo y salsa rosa hot spicy	9,90
NOODLES	SHIBUYA UDON Fideos de trigo, caldo de pollo, kintoboshi y salsa rosa hot spicy	9,90
MARU ISHIGAKI	MARU ISHIGAKI Sándwich de pollo frito, cebolla roja encurtida, salsa chile roja y wasabi	5,90
MARU ISHIGAKI	MANTOUKURA MARU Pez monjequillo, arroz y salsa rosa	5,90
MARU ISHIGAKI	MANTOUKURA MARU Pez monjequillo, arroz y salsa rosa	5,90

- **RF12** - Cada restaurante debería incluir una lista de espera con usuarios en cola.

[Completado]

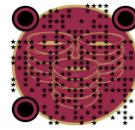
The screenshot shows a web browser displaying the QrRestaurant app. It displays a message indicating that the restaurant is full and suggests looking for alternatives or joining the waitlist. At the bottom, it provides copyright and development information.

Actualmente el restaurante al que intentas acceder está lleno, puedes esperar a que haya aforo suficiente para entrar añadiéndote a la cola o ir a un restaurante alternativo.

© QRRestaurant 2021
Desarrollado por QRRestaurant Team

- **RF13** - El administrador del restaurante debe poder añadir restaurantes.

[Completado]



The screenshot shows a web interface for managing restaurants. At the top, there's a navigation bar with links for 'Inicio', 'Mis restaurantes', 'Tipos de comida', 'Preguntas frecuentes', and 'Quiénes somos'. On the right side of the header are 'Cerrar Sesión' and 'Añadir Restaurante' buttons. Below the header, there are two cards representing different restaurants:

- Lhardy**: A card featuring a logo of a restaurant with a chandelier and a menu. Below it is a short description: "Con el ornato de esta bella fachada definida por el gusto del Segundo Imperio que vuelve ahora a cautivarnos, Lhardy ha sabido conservar celosamente el ambiente cortesano y aristocrático del Madrid del siglo XX, y los comienzos del XXI al mismo tiempo que las mejores fórmulas de la cocina europea". Below the description are three buttons: 'Ver restaurante', 'Editar restaurante', and 'Eliminar restaurante'.
- Sibuya Urban Sushi Bar**: A card featuring a logo with the word 'SIBUYA' in large letters and 'URBAN SUSHIBAR' below it. Below it is a short description: "Sibuya Sushi Bar es ese "peaceful place" donde puedes ir para desconectar. Sushi delicioso, atención al detalle y a la calidad y un ambiente chill único que hará que tu visita a nuestro restaurante te recargue las pilas y sea mucho más fácil volver a conectar con el mundo de ahí fuera.". Below the description are three buttons: 'Ver restaurante', 'Editar restaurante', and 'Eliminar restaurante'.

This screenshot shows a 'Create New Restaurant' form. The title 'Añadir nuevo restaurante' is at the top. The form consists of several input fields and dropdown menus:

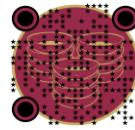
- Nombre del restaurante: A text input field.
- Aforo: A text input field.
- Dirección: A text input field.
- Localidad: A text input field.
- Descripción: A text input field.
- Tipo de comida: A dropdown menu currently set to 'Asiática'.
- Carta: A file upload field labeled 'Seleccionar archivo' with 'Ningún archivo seleccionado'. Below it, it says 'Formatos admitidos: pdf'.
- Fotos: A file upload field labeled 'Elegir archivos' with 'Ningún archivo seleccionado'. Below it, it says 'Formatos admitidos: jpg, png'.

At the bottom is a large blue 'Añadir restaurante' button.

- **RF14** - El administrador del restaurante debe poder editar restaurantes.

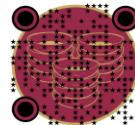
[Completado]

This screenshot shows the same web interface as the first one, but with a focus on the 'Editar restaurante' (Edit restaurant) button. The 'Editar restaurante' button is highlighted in green, indicating it is the active or selected action. The other buttons ('Ver restaurante' and 'Eliminar restaurante') are in their standard grey color.



- **RF15** - El administrador del restaurante debe poder eliminar el restaurante.

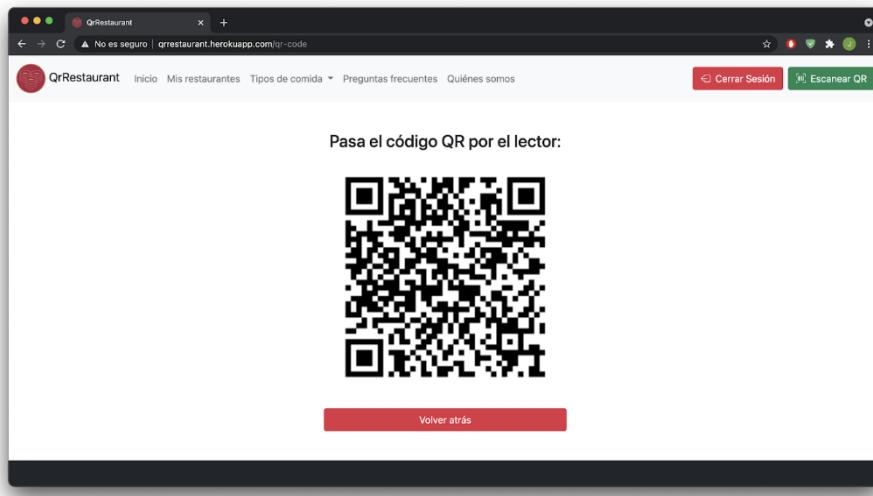
[Completado]



Usuario

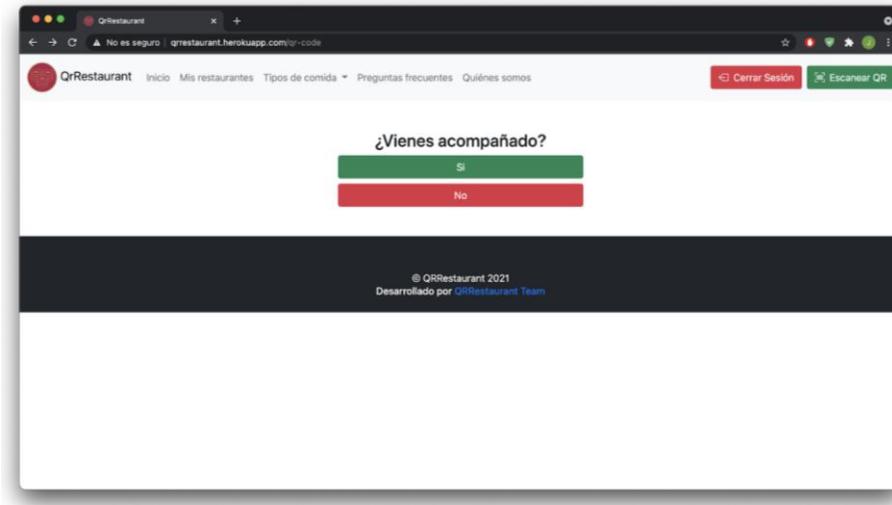
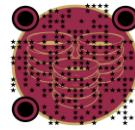
- **RF16** - El sistema debería generar un código QR por cada usuario registrado.

[Completado]



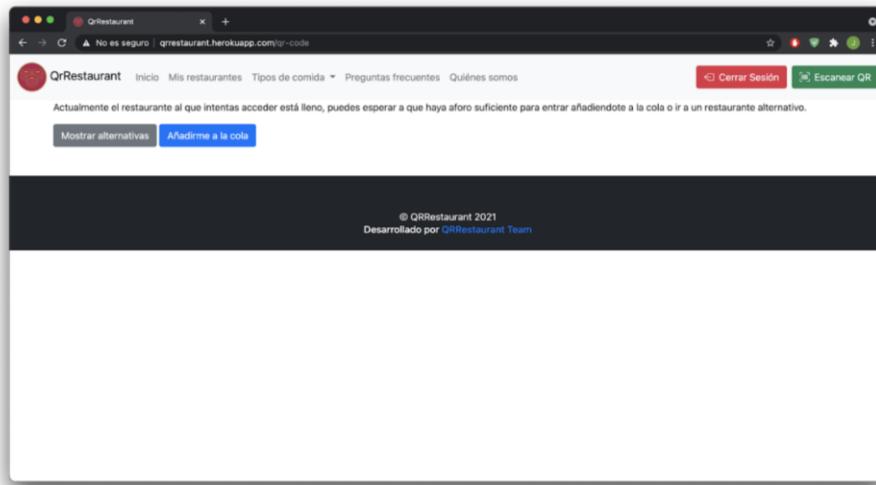
- **RF17** - El sistema debería permitir que cada usuario pueda indicar cuando entra a un restaurante si va acompañado de menores.

[Completado]



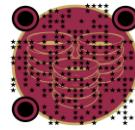
- **RF18** - El sistema debería permitir poner en cola a un usuario que intenta acceder a un local que supera el aforo.

[Completado]



- **RF19** - El sistema debería permitir a los usuarios añadir reseñas y valorar los restaurantes que visite.

[Completado]

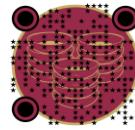


- **RF20** - El usuario no registrado debe poder registrarse en el sistema.

[Completado]

- **RF21** - El usuario debe poder iniciar sesión en el sistema.

[Completado]



Iniciar Sesión

Email: _____

Contraseña: _____

Iniciar sesión

Aún no tienes cuenta? Regístrate aquí

Escanear QR

- **RF22** - El sistema debe permitir al usuario filtrar por valoración.

[Completado]

Desde QrRestaurant queremos ayudar a los restaurantes a controlar el aforo. Visita nuestras secciones de [Preguntas frecuentes](#) y [Quiénes somos](#) para más información.

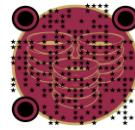
Mayor a menor valoración Buscar Filtrar por valoración

Casa Pepe de la Judería
Última actualización: 19/05/2021 19:05
Bienvenidos a Casa Pepe de la Judería, el restaurante ideal para sumergirse en la riqueza culinaria cordobesa

Lhardy
Última actualización: 19/05/2021 12:01
Con el ornato de esta bella fachada definida por el gusto del Segundo Imperio que vuelve ahora a cautivarnos, Lhardy ha sabido

- **RF23** - El sistema debe permitir buscar restaurantes por ubicación cercana.

[Completado]



The screenshot shows the main search interface of the QrRestaurant app. At the top, there is a navigation bar with links for 'Inicio', 'Tipos de comida', 'Preguntas frecuentes', and 'Quiénes somos'. There are also buttons for 'Iniciar Sesión' and 'Escanear QR'. Below the navigation, a message encourages users to help restaurants control their capacity. A search bar allows users to 'Introduce una dirección o ciudad' with 'Buscar' and 'Filtrar por localización' buttons. Two restaurant cards are displayed: 'Casa Pepe de la Judería' (last updated 19/05/2021 19:05) and 'Lhardy' (last updated 19/05/2021 12:01).

The screenshot shows search results for the location 'Córdoba'. The results are titled 'Resultados por la dirección / ciudad: Córdoba'. It displays two cards: 'Casa Pepe de la Judería' (last updated 19/05/2021 19:05) and 'Sibuya Urban Sushi Bar' (last updated 19/05/2021 12:01). Both cards include the restaurant's logo and a brief description.

- **RF24** - El sistema debe permitir al administrador ver sus restaurantes.

[Completado]

The screenshot shows the 'Mis restaurantes' (My Restaurants) section. It lists the two restaurants previously shown in the search results: 'Lhardy' and 'Sibuya Urban Sushi Bar'. Each card includes the restaurant's logo, name, last update timestamp, and a brief description. Below each card are three buttons: 'Ver restaurante', 'Editar restaurante', and 'Eliminar restaurante'.