

# QR Restaurant

APOYAMOS TU SEGURIDAD



Francisco Almenara Parias  
Manuel Cortés Serrano  
Cristian Cosano Cejas  
Antonio Luis Rodríguez Jiménez  
Jose Manuel Gil Rodríguez

# Índice

- 1- Introducción
- 2- Definición del problema
- 3- Alcance del sistema
- 4- Objetivos del problema
- 5- Antecedentes
- 6- Recursos
- 7- Captura de requisitos
- 8- Definición de requisitos
- 9- Validación de requisitos
- 10- Modelado de casos de uso
- 11- Modelado de actividades



# Introducción

## COVID-19

El contexto actual obliga a establecer protocolos para mantener la distancia de seguridad.

## NUEVA NORMALIDAD

El nuevo panorama desemboca en controlar el aforo en establecimientos cerrados como restaurantes.

## VISIÓN A FUTURO

Los códigos QR dan soporte a los restaurantes para la gestión del aforo y reduciendo el contacto.



# Definición del problema

## LOCALES CON AFORO LIMITADO

Los locales se han visto obligados a reducir su aforo.

## AMBIGÜEDAD EN EL CUMPLIMIENTO DE NORMAS

El usuario no conoce si se está controlando el aforo.

## DESCONOCIMIENTO DE LA NORMATIVA EN LOS ESTABLECIMIENTOS

El usuario debe saber si el restaurante.

## PÉRDIDA DE TIEMPO PARA EL USUARIO

El usuario puede acudir a un local que esté completo.

## PERDIDA DE DINERO

Los locales no están generando beneficios debido al covid



# Alcance del sistema



Estandarizar la seguridad en restaurantes



Enfocado en los clientes de estos locales



Ámbito nacional



Mejorar la calidad del servicio

# Objetivos del problema



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.

# Antecedentes



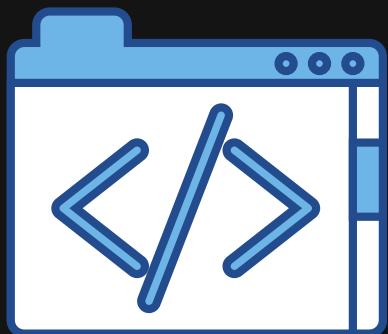
GMaps y TripAdvisor son muy buenas plataformas, pero no nos permiten tener información de aforo ni controlarlo en los restaurantes

# Recursos



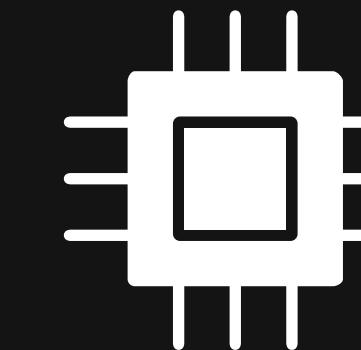
## HUMANOS

Equipo profesional formado por 5 integrantes



## SOFTWARE

Utilizaremos JavaScript tanto para el backend como para el frontend.



## HARDWARE

Se usará un Paas llamado Heroku el cual hace un despliegue continuo del sistema





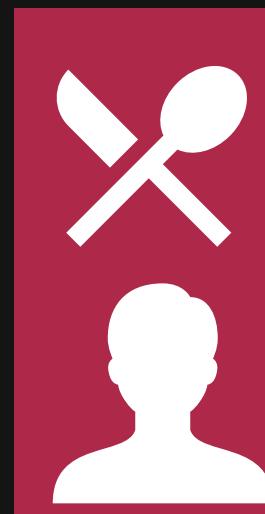
# Captura de requisitos

Para realizar la captura de requisitos nos basamos principalmente en un enunciado del problema que se nos proporcionó y una entrevista con el cliente de cual obtuvimos una lista preliminar de requisitos. También realizamos una brainstorm entre todos los miembros del equipo

# Definición de requisitos

Clasificados como:

## REQUISITOS FUNCIONALES



Restaurante  
Usuario

## REQUISITOS NO FUNCIONALES



Restricciones de  
dominio

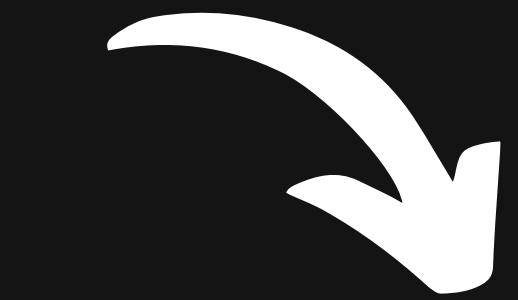
## REQUISITOS DE INFORMACIÓN



Información del  
sistema

# Validación de requisitos

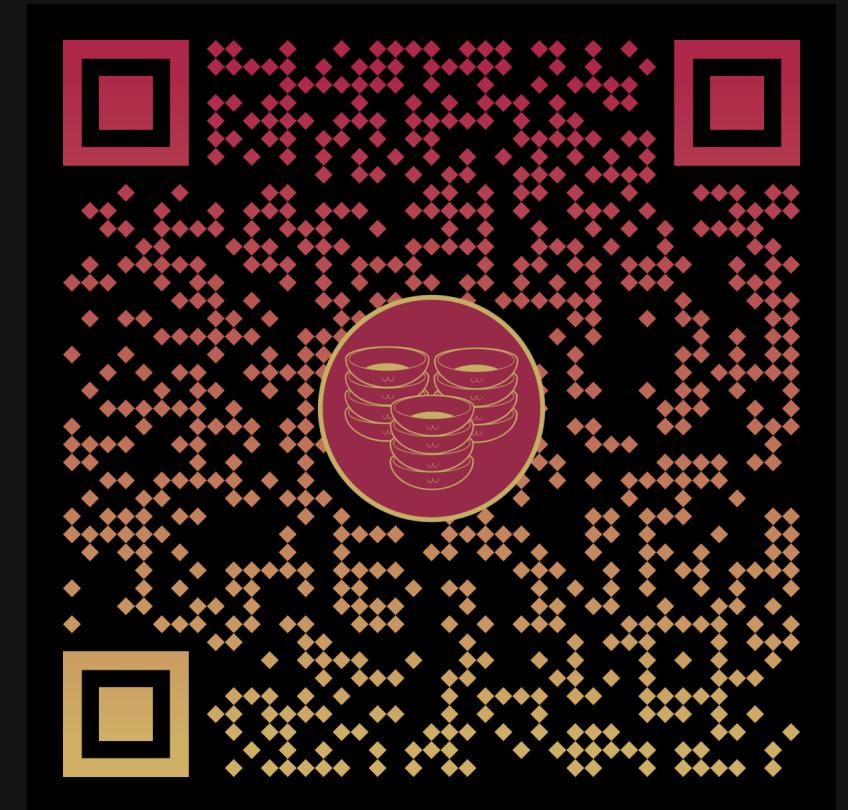
1 WALK-THROUGHS



2 AUDITORÍA DEL EQUIPO DE DESARROLLO

3 SKETCHES O BOCETOS

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



# Modelado de casos de uso



Usuario no registrado



Usuario registrado



Administrador de restaurantes

Descripción general de las funcionalidades del sistema y los actores que intervienen en el sistema.

Generación de la matriz de trazabilidad para comprobar conflictos con los requisitos y refinarlos.

[Ver diagrama](#)

# Modelado de actividades

Información más detallada que en los casos de uso no se muestra. Las acciones de los casos de uso y los datos se presentan como actividades.

[Ver diagrama](#)

# ¿Alguna pregunta?

