

Dishcovery.

IDEA INNOVADORA:

Aplicación de delivery que, además del pedido normal, ofrece un modo sorpresa, el usuario define un presupuesto, tipo de comida, restricciones y nivel de aventura culinaria, y el restaurante envía un plato que cumpla con esos parámetros, cada pedido "modo sorpresa" regalará cierta cantidad de puntos a el cliente que luego de 100 de estos podrá escoger un platillo gratis. El objetivo es convertir cada pedido en una experiencia emocionante y que el cliente acumule puntos para obtener un premio.

•Valor diferencial: Factor sorpresa que genera curiosidad y fidelidad.

Ideal para foodies o personas indecisas.

Posibilidad de compartir la experiencia en redes desde la app.

•Gamificación: puntos por descubrir nuevos platos o completar "retos gastronómicos".

Descripción General De La Aplicación:

El objetivo de esta práctica es diseñar una aplicación de delivery de comida que replique el modelo de negocios de plataformas existentes como rappi, uber eats o domicilios.com. la aplicación debe ser funcional para tres tipos de usuarios:

1. Usuarios finales: personas que realicen una pedido de comida.
2. Restaurante: el restaurante debe preparar y despachar los pedidos.
3. Repartidor: personas encargadas de llevar el producto al usuario final.

REQUISITOS FUNCIONALES:

1. El sistema debe permitir a los usuarios (clientes, restaurante, repartidor) registrarse con un correo electrónico y contraseña
2. El sistema debe permitir a los usuarios iniciar y cerrar sesión de forma segura.
3. El sistema debe permitir a los usuarios editar la información de su perfil (nombre, teléfono, dirección, etc.)
4. El sistema debe permitir a los usuarios recuperar su contraseña en caso de olvido.
5. El sistema debe permitir al cliente buscar restaurantes por nombre, tipo de comida, o ubicación.
6. El sistema debe mostrar el menú de cada restaurante con descripción, precio e imagen representativa del producto.
7. El sistema debe permitir al cliente añadir productos a un carrito de compras.
8. El sistema debe permitir al cliente realizar un pedido, seleccionando una dirección de entrega y un método de pago.
9. El sistema debe permitir al cliente rastrear el estado de su pedido en tiempo real (ej. "En preparación", "En camino").
10. El sistema debe permitir al cliente ver su historial de pedidos.
11. El sistema debe permitir al cliente calificar y dejar comentarios sobre el restaurante y el repartidor.
12. El sistema debe notificar al restaurante cuando un nuevo pedido ha sido realizado.
13. El sistema debe permitir al restaurante ver los detalles del pedido (productos, dirección de entrega del cliente).
14. El sistema debe permitir al restaurante aceptar o rechazar un pedido.
15. El sistema debe permitir al restaurante actualizar el estado del pedido (ej. "En preparación", "Listo para recoger").
16. El sistema debe permitir al restaurante gestionar su menú (añadir, editar, eliminar productos y precios).
17. El sistema debe permitir al restaurante ver sus estadísticas de ventas e historial de pedidos.
18. Funcionalidades para el Repartidor

19. El sistema debe notificar al repartidor sobre pedidos disponibles cerca de su ubicación.
20. El sistema debe permitir al repartidor aceptar o rechazar un pedido.
21. El sistema debe mostrar al repartidor la ubicación del restaurante y la dirección de entrega del cliente.
22. El sistema debe permitir al repartidor actualizar el estado del pedido (ej. "Recogido", "Entregado").
23. El sistema debe mostrar al repartidor el historial de sus entregas y sus ganancias.
24. El sistema debe integrarse con una pasarela de pagos para procesar transacciones con tarjeta de crédito/débito.
25. El sistema debe permitir la opción de pago en efectivo al momento de la entrega.

REQUISITOS NO FUNCIONALES

1. El sistema debe poder manejar un gran volumen de pedidos simultáneos sin degradar su rendimiento.
2. La interfaz de usuario debe ser intuitiva y fácil de usar para todos los tipos de actores.
3. El sistema debe proteger la información personal y de pago de los usuarios mediante cifrado.
4. El sistema debe autenticar y autorizar a los usuarios antes de permitir el acceso a funcionalidades específicas.
5. El sistema debe estar diseñado para permitir la adición de nuevas funcionalidades y el crecimiento de la base de usuarios.
6. El código del sistema debe ser fácil de entender y modificar por otros desarrolladores.

HISTORIAS DE USUARIO

Como cliente, quiero poder buscar restaurantes cercanos a mi ubicación para encontrar opciones para mi comida.

Como cliente, quiero ver el menú de un restaurante con fotos y precios para saber qué puedo pedir.

Como cliente, quiero poder añadir productos a mi carrito de compras para preparar mi pedido.

Como cliente, quiero poder ingresar mi dirección y método de pago para completar mi pedido de forma sencilla.

Como cliente, quiero poder ver el estado de mi pedido en tiempo real para saber cuándo llegará mi comida.

Como un restaurante, quiero ser notificado de un nuevo pedido para poder empezar a prepararlo rápidamente.

Como un restaurante, quiero poder aceptar o rechazar un pedido para gestionar mi capacidad de servicio.

Como un restaurante, quiero poder marcar un pedido como "Listo para recoger" para que el repartidor sepa cuándo debe ir a buscarlo.

Como un restaurante, quiero poder añadir o quitar productos de mi menú para mantenerlo actualizado.

Historias de Usuario para el Repartidor

Como repartidor, quiero recibir notificaciones de nuevos pedidos para decidir si los acepto.

Como repartidor, quiero ver la dirección del restaurante y la del cliente en un mapa para optimizar mi ruta de entrega.

Como repartidor, quiero poder marcar un pedido como "Entregado" para que el sistema me asigne el siguiente.

Como administrador, quiero que el sistema procese los pagos de forma segura y los distribuya a los restaurantes y repartidores.

Como administrador, quiero poder ver el número de pedidos activos y la ubicación de los repartidores para garantizar la eficiencia del servicio.

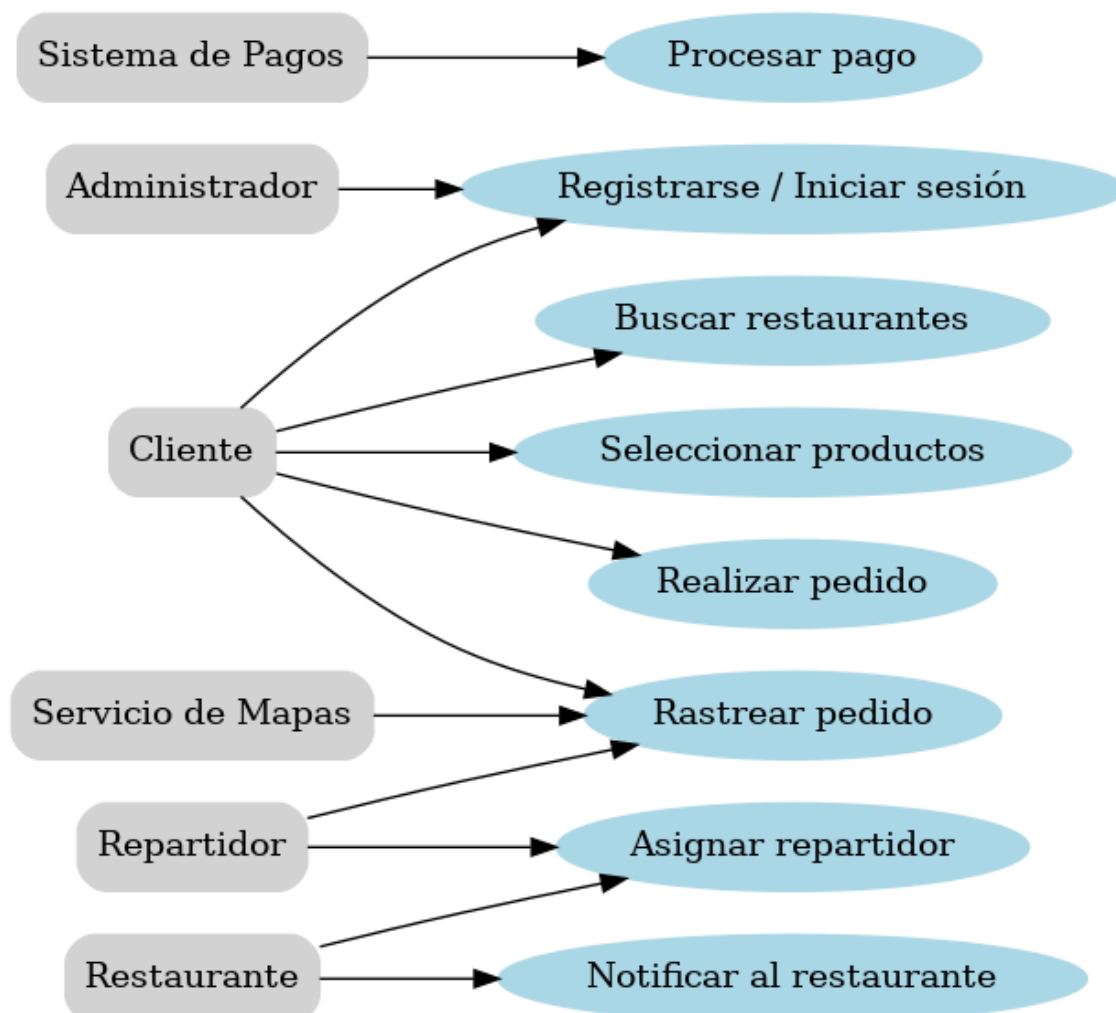
TIPO DE USUARIO

- Usuario final (cliente que ordena un pedido)
- Restaurante (que recibe y prepara pedidos)
- Repartidor (que entrega el pedido)
- Administrador del sistema

SISTEMAS EXTERNOS INVOLUCRADOS

- Pasarela de pagos (PSE, crédito, débito, demás)
- Servicio de geolocalización (Google Maps, Waze)
- Redes sociales para login

DIAGRAMA UML



FLUJO DE USO

REALIZAR PEDIDO

Actores principales:

- Cliente
- Restaurante
- Repartidor

Actores secundarios:

- Sistema de Pagos
- Servicio de Mapas
- Módulo de Puntos y Gamificación

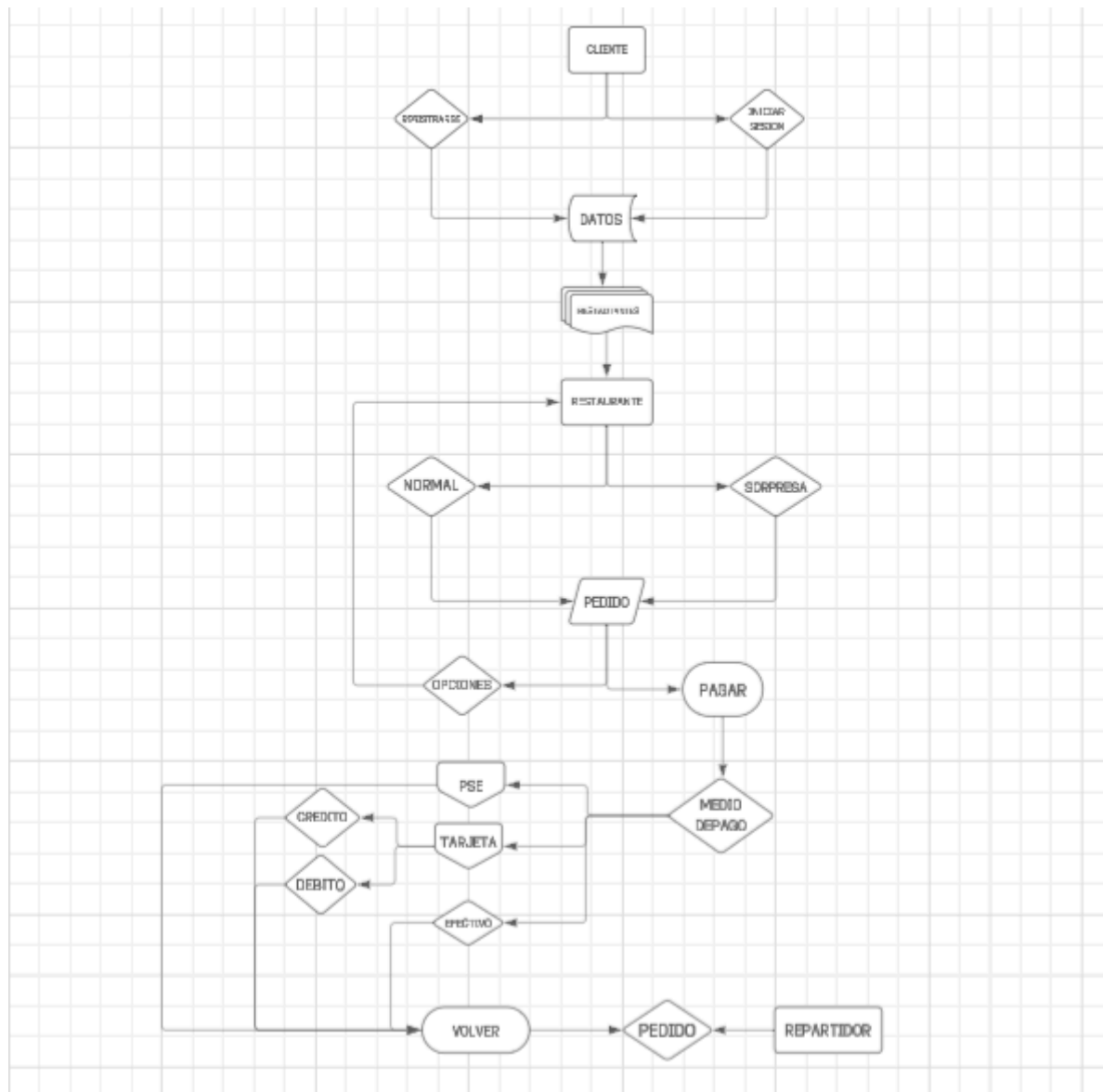
Objetivo: Permitir al cliente pedir comida (normal o "modo sorpresa") y recibirla, generando una experiencia emocionante y acumulando puntos.

FLUJO PRINCIPAL

1. El Cliente abre la aplicación y se autentica.
2. El sistema muestra restaurantes disponibles según ubicación.
3. Cliente selecciona:
 - Pedido normal: elige restaurantes y productos del menú.
 - Modo sorpresa: define presupuesto, tipo de comida, restricciones y nivel de aventura culinaria.
4. El sistema muestra un resumen del pedido (en modo sorpresa no se revela el plato exacto).
5. El cliente confirma el pedido y elige el método de pago.
6. El sistema procesa el pago a través del Sistema de Pagos.
7. El Restaurante recibe la orden:
 - Pedido normal: prepara los productos solicitados.
 - Modo sorpresa: selecciona un plato que cumpla con parámetros definidos.
8. El sistema asigna un repartidor disponible, que usa el Servicio de Mapas para la ruta.
9. El Repartidor recoge el pedido y lo entrega al Cliente.
10. Cliente confirma la entrega en la app.
11. El sistema registra el pedido:
 - Si es modo sorpresa: añade puntos al perfil del cliente (por ejemplo, 10 puntos por pedido sorpresa).
 - Al llegar a 100 puntos, el cliente puede reclamar un plato gratis.
12. El sistema ofrece la opción de compartir la experiencia en redes sociales desde la app.

FLUJOS ALTERNATIVOS / EXCEPCIONES

- El restaurante rechaza el pedido → el sistema ofrece alternativas.
- Falla en el pago → solicitar nuevo método de pago.
- No hay repartidor → poner en espera y notificar al cliente.
- Cliente cancela antes de que el restaurante acepte → devolución de dinero.



WIREFRAME

<https://www.figma.com/proto/N1aW5FRI4O5GKPFfFZTHeb/Untitled?node-id=0-1&t=gsKtD5Z4UBtbI9VY-1>