

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



INGENIERÍA EN COMPUTACIÓN Facultad de Ciencias Químicas e Ingeniería (Campus Tijuana)

Alumno:
Moreno Lopez Luis Mario (1271304)
Grupo: 571

Profesor: Lara Camacho Evangelina
Materia: Sistemas Embebidos.

Tipos de sistemas embebidos:

- Sistemas embebidos autónomos:
 - Microondas: el microondas es un sistema embebido autónomo pues no depende de alguna computadora o sistema externo para funcionar mas que los comandos que se introducen por el keypad.
 - cámaras de tesla: El sistema de cámaras de tesla es autónomo pues siempre está monitoreando a su alrededor para identificar objetos y actuar en cualquiera de los casos
- Sistemas embebidos de tiempo real
 - Medidor de glucosa: este sistema embebido también es del tipo hard, ya que no debe de perder ningún dato que mide y siempre está monitoreando la glucosa de la persona que lo trae.
 - GPS: El GPS es uno de los mejores ejemplos de este tipo de sistema pues en cada momento está monitoreando las coordenadas actuales de la persona que lo está usando
- Sistema en red
 - Cajeros automáticos: Los cajeros automáticos están interconectados hacia un gran sistema bancario y es por eso por lo que se puede sacar dinero desde cualquier cajero automático y se verá reflejado en todas partes, todo lo que se haga queda registrado.
 - Red de sensores inalámbricas: Como el nombre lo indica este tipo de red donde todos los sensores están conectados hacia un mismo nodo Gateway donde siempre le están mandando los datos que miden.
- Sistemas móviles:
 - Celulares: los celulares son el sistema embebido móvil que más se usa actualmente, los celulares cuentan con gran cantidad de sensores o periféricos todo alimentándose por la batería por lo que lo hacen el perfecto sistema embebido móvil
 - Reloj inteligente: El reloj inteligente se puede colocar en la muñeca del usuario y llevar a cualquier lado y estar funcionando gracias a su batería recargable

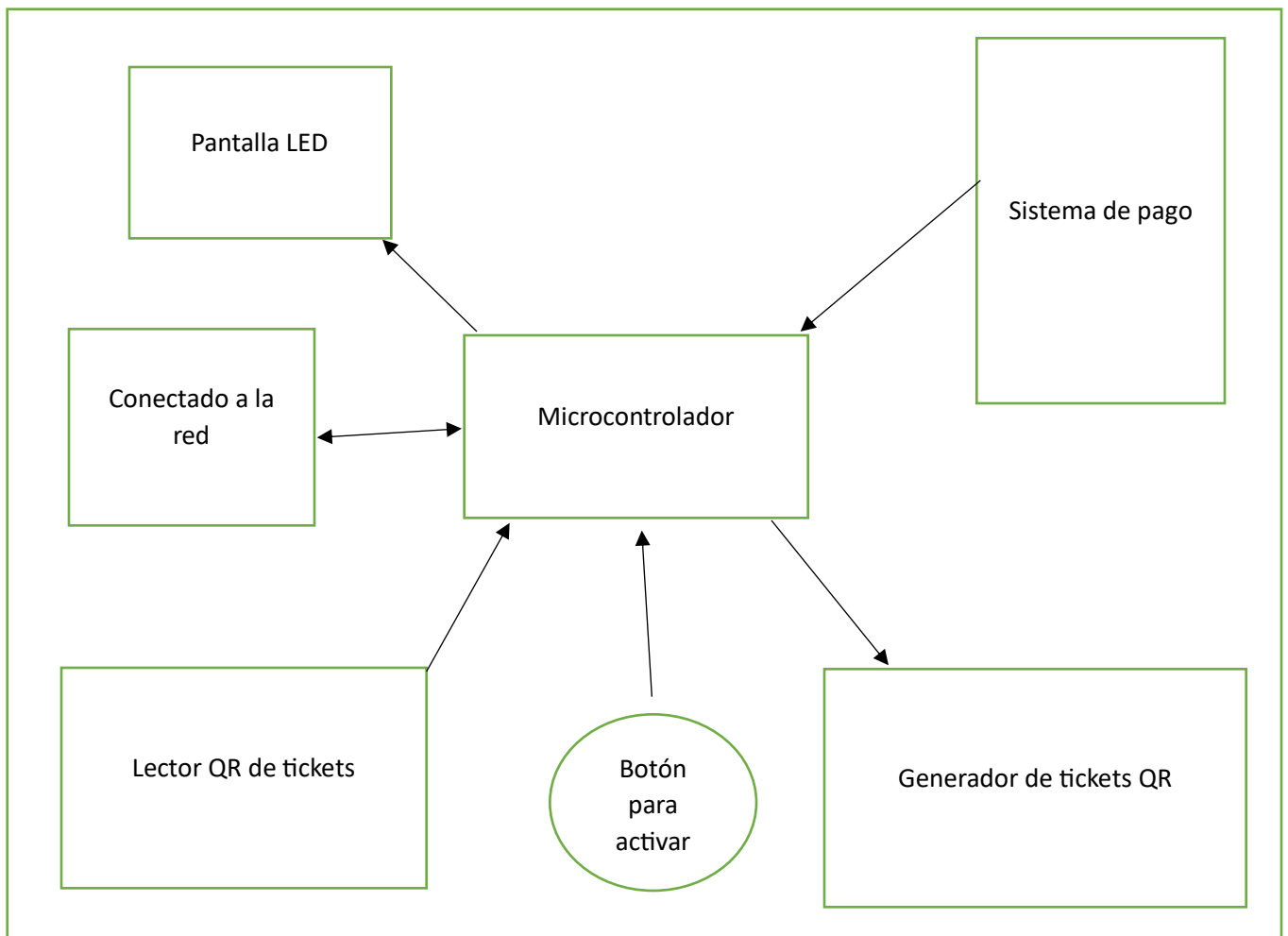
Diseño de una maquina para pago de estacionamiento

1. Abstracción

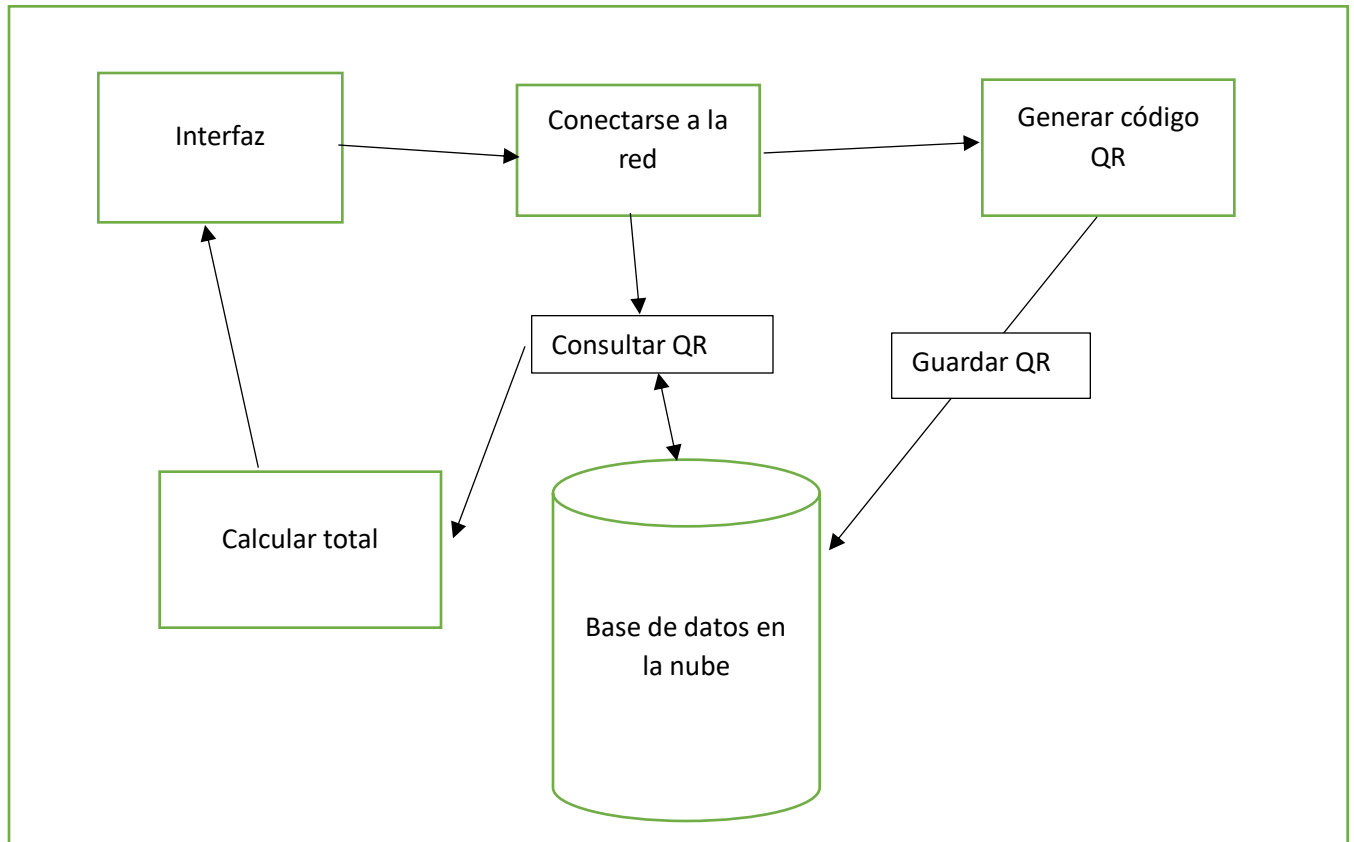
Se necesitan varios componentes o partes principales para poder funcionar, el primero registrar el ticket que emite, sistema de pago y obtener el total de tiempo de un ticket para poder calcular el precio

2. Arquitectura de hardware y software:

Arquitectura física

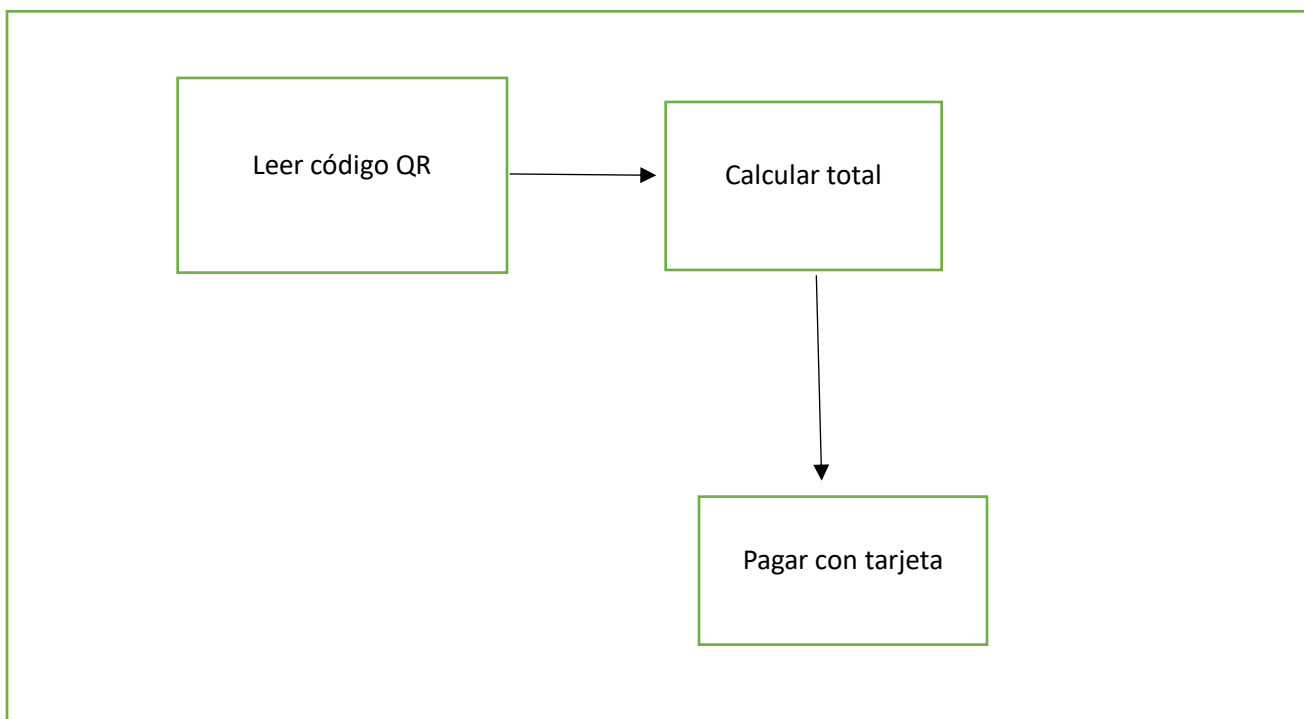


Arquitectura de software



3. Propiedades extra funcionales:

Función de pagar con tarjetas



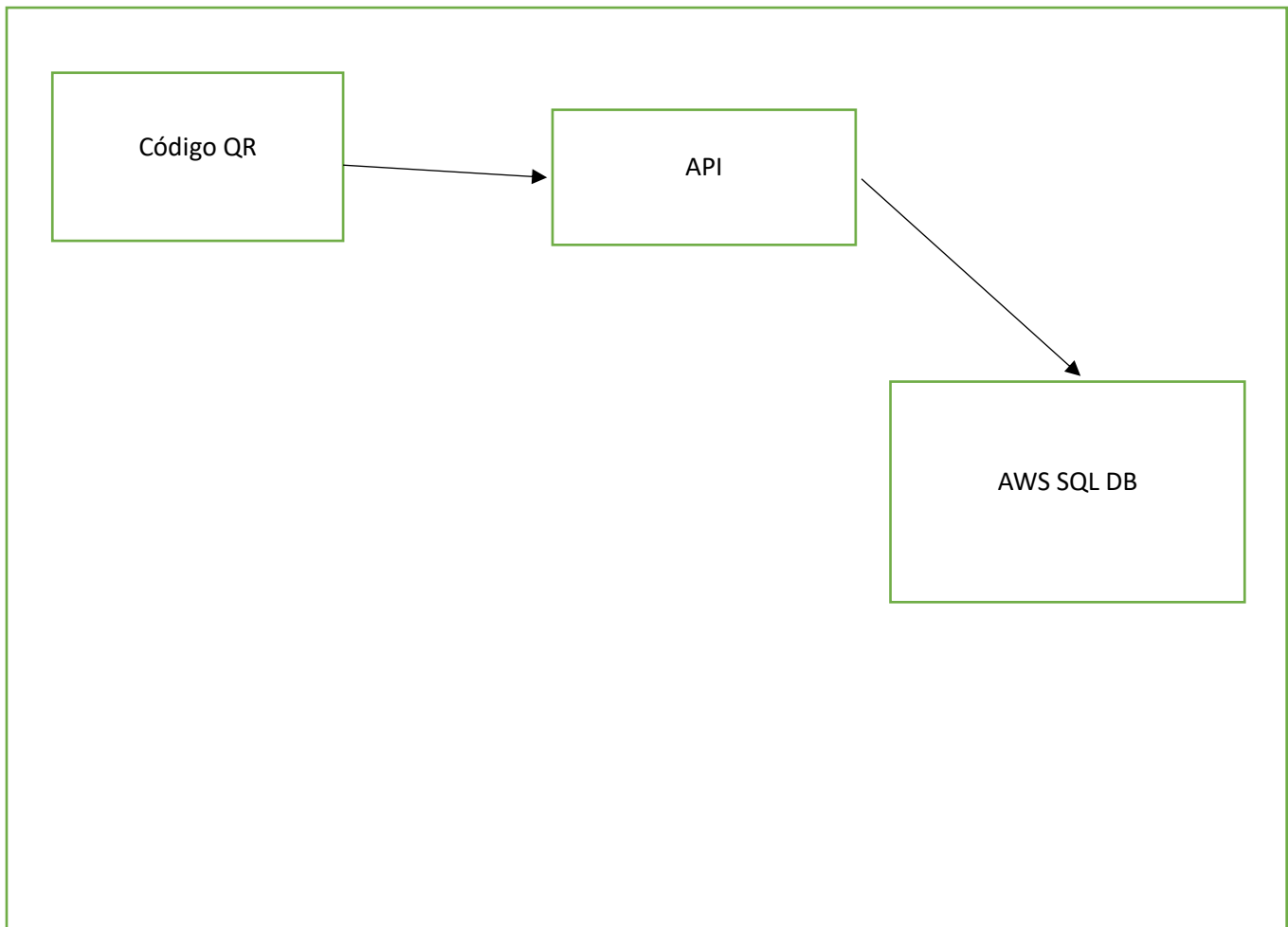
4. Familia de diseños relacionada con el sistema:

Un sistema parecido al sistema que se esta ejemplificando son las máquinas de tickets de parking de la marca RTB, estos cuentan con las propiedades extras y la arquitectura física es muy parecida a que se está construyendo.

5. Diseño modular:

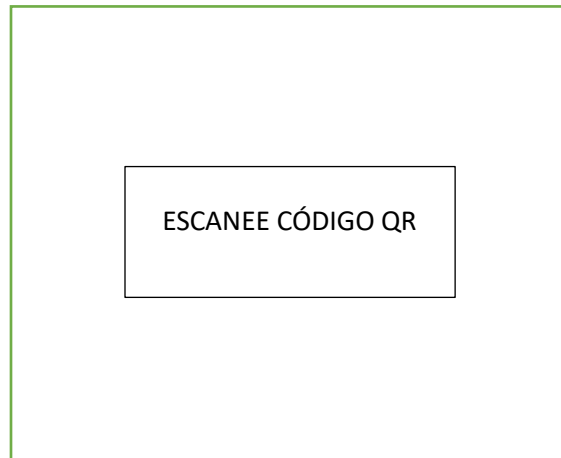
- Módulo de generación de tickets con código QR: Este módulo contiene el botón que avisa que se necesita un nuevo ticket y este lo genera.
- Módulo de pago: Este módulo contiene el sistema de pagos que recibe monedas o tarjetas y realiza la transacción.
- Modulo de obtener tiempo transcurrido: Este modulo va de la mano con los dos anteriores y se separó para hacer esta.

6. Mapeo



7. Diseño de interfaz de usuario

Para la interfaz de usuario se espera que sea lo más intuitiva posible para evitar confusiones con los usuarios. La pantalla principal estará el mensaje de “Escanear código QR” y este se quedará así hasta que no sea escaneado uno.



Una vez escaneado el código QR saldrá el total a pagar junto con los dos sistemas de pago:



Una vez pagada internamente se registra el pago y el usuario ya no hace nada más.

8. Refinamiento

Se puede refinar todos los componentes para optimizar tanto el software como el hardware. En el código la optimización de los códigos QR para que se generen de forma más rápida y de igual manera una lectura más rápida.

Métricas en el diseño:

- Rendimiento: Al estar conectado con la nube el proceso de lectura y de obtener el total a pagar puede bajar el rendimiento del sistema pues se depende de la buena conexión que se tenga.
- Tiempos en los que el sistema debe completar un proceso o tarea: Se estima que desde que se lee el código QR hasta que el usuario pague van desde los 30 segundos hasta 2 minutos dependiendo del usuario, pero cada proceso individual tarda de 500ms hasta 2 segundos
- Tamaño: El tamaño de todo el sistema es relativamente pequeño, pero para facilidad y ergonomía de los usuarios el sistema interno estará en una caja de 30cm de ancho 30cm de largo y 10cm de profundidad.
- Seguridad: El sistema encriptará la información antes de mandarla a la nube para evitar fugas de datos.
- Costo de manufactura: Se estima que el puro sistema cueste unos 50 dólares
- Costo de ingeniería:
- Prototipado: El tiempo que toma en desarrollar se estima que máximo sean 2 meses de trabajo
- Tiempo de salida al mercado: El sistema ya funcionando una vez después del prototipo se estima que sea en 4 meses.

Retos en el diseño de sistemas embebido:

- Adaptabilidad al entorno: El sistema se pondrán en áreas especiales para su uso
- Consumo de energía: Se espera que sea de bajo consumo cuando no esté operando y cuando esté en operación normal puede gastar unos 100W/h
- Área ocupada: El área no excederá de 2 metros cuadrados
- Empaquetado e integración: El sistema se pondrá en una caja cuadrada.
- Actualización de hardware y software: El software se estará actualizando constantemente siguiendo las actualizaciones de
- Seguridad: La seguridad física la caja estará dentro de otra caja y el software ese
- Costo: El costo de todo el equipo tiene costo de 150 dólares

Preguntas que suelen sugerir:

- ¿Cuánto hardware se necesita? Poder de cómputo del CPU, memoria, etc.
Para la CPU se requiere una de bajo a medio. gama, las CPU puede tener una memoria RAM de 512KB, solo para poder conectarse a la red y obtener los datos desde ahí.
- Qué periféricos se necesitan. Se van a implementar por hardware o software.
Los periféricos de hardware se necesitan una pantalla LCD para mostrar los mensajes a los usuarios, un tragamonedas, lector de códigos QR y lector de tarjetas.
- ¿Cómo se pueden cumplir las restricciones de tiempo?
 - Uso de hardware más rápido o software más ingenioso.
 - Sistema de tiempo real o diseño personalizado

Para el hardware se puede implementar un lector de códigos QR más rápidos y se puede optimizar el código para que ejecute la acción de buscar ese código QR en la base de datos de forma eficiente

- ¿Cómo se minimiza el consumo de potencia?
Se minimiza cuando llega a cierto tiempo de inactividad todas las operaciones se ponen en modo ahorro de energía y solo se despiertan cuando un código QR se vuelve a leer
- ¿Cómo se minimiza el costo?
Los costos se reducen haciendo que todo sea en la nube, así para que solo los periféricos necesarios estén dentro de sistema y lo demás sea por la nube
- ¿Cómo se asegura la seguridad y confiabilidad del sistema?
Como cualquier otro producto se pondría a prueba por un equipo de pruebas que lo somete a todo tipo de pruebas
- ¿Cómo se logra el tiempo de salida al mercado?
Llevando un buen control de todas las fases en el desarrollo de este producto para poder cumplir con el tiempo establecido

Referencias:

- Aaryaman, J. (2016, 31 de marzo). Embedded Systems in Baking Ovens. Prezi. <https://prezi.com/tc4yd8nqe3qm/embedded-systems-in-baking-ovens/>
- Blumenscheid, B. (2021, 4 de junio). 10 ejemplos reales de sistemas empotrados. digi. <https://es.digi.com/blog/post/examples-of-embedded-systems>
- Knöll, P. (2022, 20 de agosto). Paper Ticket Dispenser For Parking Machines: 5 Important Questions Before Integration. linkedin. <https://www.linkedin.com/pulse/paper-ticket-dispenser-parking-machines-5-important-questions-knöll/>
- Networked Embedded Systems (NES). (2021, 21 de septiembre). SpringerLink. https://link.springer.com/chapter/10.1007/978-981-16-3293-8_8
- PARKING TICKET MACHINES. (s.f.). RTB. https://www.rtbsafetraffic.com/wp-content/uploads/RTB_parking_ticket_machines.pdf