BELLYNDAS
FOOD

Manual técnico.

Ysabel Suárez Palomo

ÍNDICE

Introducción

La aplicación BELLYNDAS ha sido desarrollada con el propósito de brindar una solución tecnológica eficiente y moderna para la gestión de un restaurante. Su objetivo principal es los desafíos operativos y optimizar los procesos relacionados con el gestión de mesas, pedidos, cobros y empleados en el entorno de un restaurante, mediante la integración de un sistema TPV (Terminal de Punto de Venta) y dispositivos móviles que incluirán PDA (Asistente Digital Personal).

Esta aplicación permite llevar un seguimiento en tiempo real de los estados de las mesas, así como realizar operaciones como toma de pedidos, eliminación de mesas, impresión de cuentas y movimientos entre mesas. La integración con Firebase asegura la sincronización y actualización constante de los datos.

Análisis del problema

a. Problemática

El problema que se pretende abordar es la falta de eficiencia en la gestión de mesas y pedidos en un restaurante. Los procesos manuales o desactualizados pueden generar retrasos y errores, lo que afecta la experiencia del cliente y la eficiencia operativa del establecimiento.

b. Clientes potenciales

Los clientes potenciales de BELLYNDAS son restaurantes y establecimientos similares que buscan mejorar la eficiencia de sus operaciones y brindar un servicio de calidad a sus clientes. Este software es especialmente relevante para restaurantes con un alto volumen de pedidos y una necesidad de agilizar los procesos de toma de comandas, gestión de mesas y control de pagos.

c. Análisis DAFO

Fortalezas.

La aplicación ofrece una solución integral y moderna para la gestión de mesas y pedidos en restaurantes. La integración con Firebase garantiza la sincronización en tiempo real y la actualización constante de los datos.

Debilidades.

La aplicación no gestiona inventarios ni realiza seguimiento de stocks. Esto puede ser una limitación para aquellos establecimientos que requieran un control más detallado de sus recursos. No muestra las órdenes enviadas en barra y cocina, ni imprime cuentas.

Oportunidades.
Existe un mercado potencial de restaurantes y establecimientos similares que buscan mejorar la eficiencia de sus operaciones. La aplicación BELLYNDAS puede posicionarse como una solución innovadora y confiable para satisfacer esta necesidad.

Amenazas.
La competencia en el sector de software para la gestión de restaurantes es alta. Es necesario realizar una estrategia de marketing efectiva y brindar un soporte técnico de calidad para destacarse en el mercado.
Cambios en la tecnología o requisitos legales que puedan afectar el funcionamiento del sistema.

d. Monetización y beneficios

Suscripción mensual/anual.
Se ofrecería una suscripción mensual o anual que incluya acceso a la aplicación y servicios adicionales como actualizaciones continuas o soporte técnico. El precio de la suscripción dependerá del valor percibido de la aplicación y los servicios ofrecidos, así como del tamaño y tipo de restaurante al que se dirige. Considerando que soy una persona recién graduada y deseo establecer un precio atractivo, se podría considerar un rango de precio mensual entre 300€ - 400€ dependiendo de las características y beneficios proporcionados.

Diseño de la solución

a. Tecnologías elegidas

El sistema de TPV y PDA para el restaurante BELLYNDAS utiliza las siguientes tecnologías:

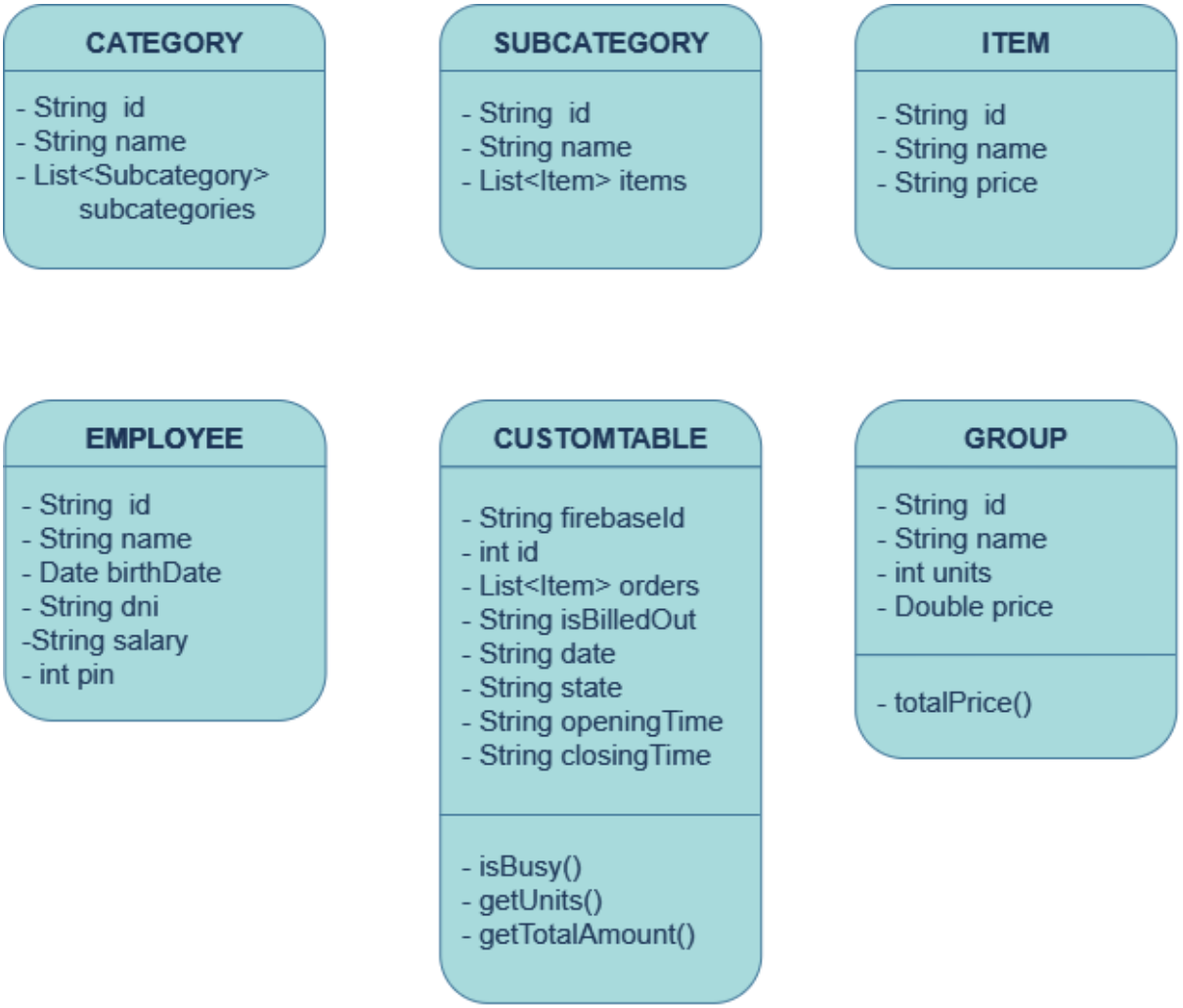
- Lenguaje de programación: Dart.
- Framework: Flutter
- Sistema de Información: Firebase.
- Entorno de desarrollo: Visual Studio Code.

b. Arquitectura

La arquitectura del sistema sigue un enfoque de arquitectura de cliente-servidor. La aplicación móvil actúa como cliente y se comunica con el servidor Firebase para obtener y enviar datos en tiempo real.

c. Diagrama de clases

A continuación se presenta el diagrama de clases simplificado del sistema de TPV y PDA



- **Category**: representa las dos secciones en que se divide un menú (bebidas y comidas).

- **Subcategory**: representa las secciones en las que las categorias esta dividida (entrantes, carne, postre, etc)

- **Item**: representa los elementos consumidos por el cliente, que se introducen en el atributo orders de la customTable

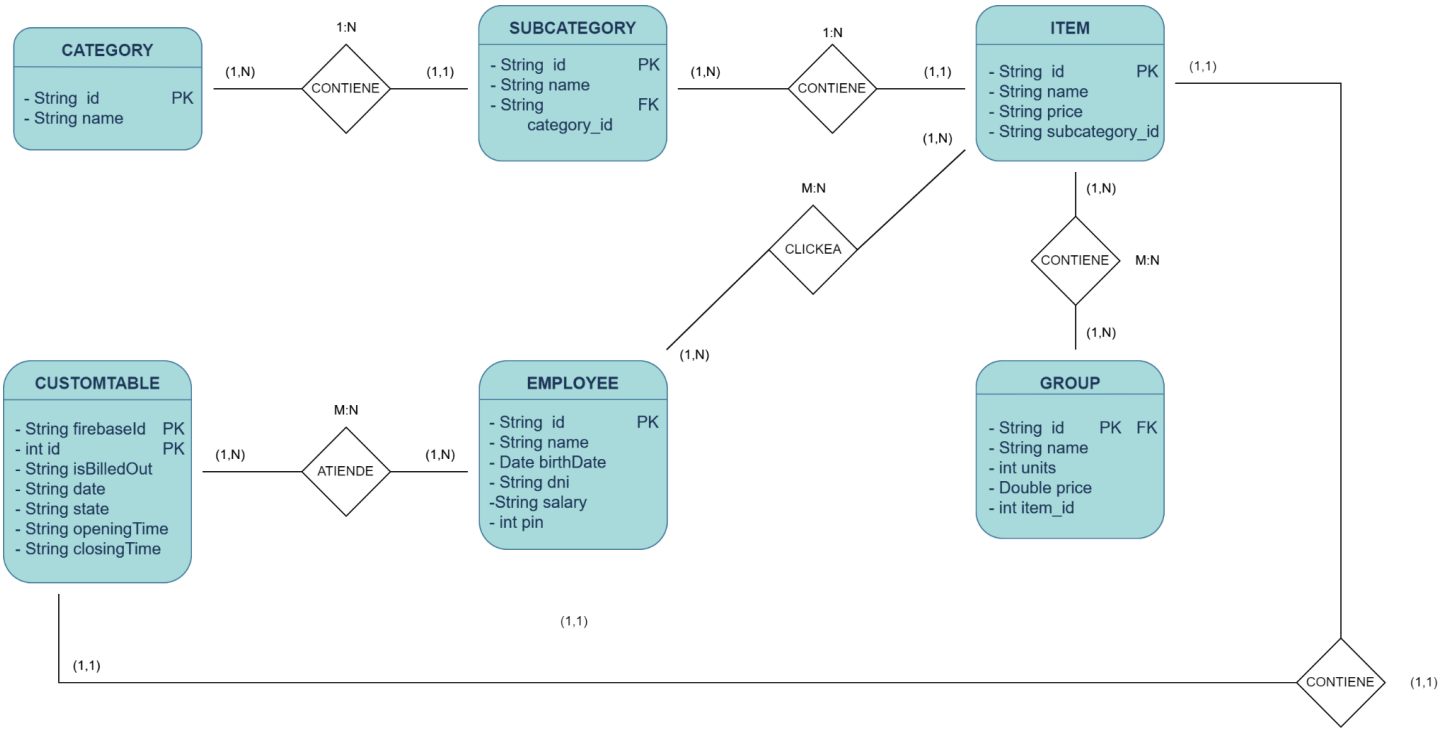
- **CustomTable**: representa las mesas del restaurante con la que mayormente interaccionamos en la aplicación.

- **Employee**: representa los empleados, necesarios para el inicio de sesión.

- **Group**: clase creada para poder manejar el atributo orders de la customTable.

d. Diagrama E/R

El sistema de TPV y PDA para el restaurante BELLYNDAS no utiliza una base de datos relacional, por lo que no es necesario un diagrama E/R.



- **Category**: hay una relación de **uno a muchos** entre **Category y Subcategory**, donde una categoría puede tener varias subcategorías, pero una subcategoría pertenece a una única categoría.

- **Subcategory:** tiene clave foránea **'category_id'** en Subcategory. Hay una relación de **uno a muchos** entre **Category y Subcategory** tal como se explica en Category.

- **Item:** al igual que Subcategory, contiene una referencia subcategory que apunta a la subcategoría a la que pertenece mediante la clave foránea **'subcategory_id'**. Hay una relación de **uno a muchos** entre **Subcategory e Item**, donde una subcategoría puede tener varios artículos, pero un artículo pertenece a una única subcategoría.

- **CustomTable:** hay una relación de **uno a muchos** entre **CustomTable e Item**, donde una mesa personalizada puede tener varios artículos en su lista de pedidos, y un artículo puede estar presente en múltiples mesas personalizadas.

- **Employee:** no esta directamente relacionado con ninguna de ellas, sin embargo es el mediador entre ellas. Estableciendo así dos relaciones:
    - **uno a muchos** entre **Empleado e Item** ya que cada empleado puede seleccionar varios items para almacenarlos en una **CustomTable**, pero al introducir dicho item en la mesa, el item es copiado a otro de su misma clase pero con distinto identificador.
    - **muchos a muchos** entre **Empleado y CustomTable** ya que varios empleados pueden enviar pedidos a varias mesas, y a su vez, cada mesa puede recibir pedidos de varios empleados.

- **Group:** tiene una clave foránea, **'item_id'** estableciendo una relación de **uno a muchos** entre **Group e Item,** puesto que en un grupo puede haber muchos items, pero un item solo pertenece a un group.

e. Consideraciones técnicas

Desplegar la app en local:

1. Descargamos Visual Studio Code.
2. Descargamos Flutter, GUIA.
3. Descargar el repositorio de "BELLYNDAS" y guardarlo en una carpeta.
    a. TPV
    b. PDA
4. Nos dirigimos a Visual Studio Code y abrimos ambas carpetas en ventanas distintas. Para cada aplicación pulsamos CRTL + SHIFT + P > Select Devide > (Seleccionamos los emuladores correspondientes). Para TPV necesitamos una pantalla de mayor tamaño a una tablet para verla según se ha programado. Para PDA en cualquier móvil podremos verla correctamente.
5. Pulsamos F5 y podremos ver como se accionan los dispositivos y la aplicación se activa. Podremos interaccionar con ella cuando en ambas se abra la pagina para iniciar sesión.

Para ejecutar firebase:

1. Crea un proyecto en Firebase:
    a. Ve al sitio web de Firebase (https://firebase.google.com) y accede a tu cuenta de Google.
    b. Crea un nuevo proyecto en Firebase haciendo clic en "Añadir proyecto" y siguiendo las instrucciones.
    c. Una vez creado el proyecto, selecciona "Agregar aplicación" y elige la plataforma en la que estás desarrollando tu aplicación (por ejemplo, Android o iOS).

2. Configura tu proyecto de Visual Studio:
    a. Abre tu proyecto en Visual Studio y asegúrate de tener instalado el SDK de Firebase para la plataforma que estás utilizando (por ejemplo, Firebase SDK para Android o Firebase SDK para iOS).

    b. Agrega las dependencias necesarias al archivo de configuración de tu proyecto. Esto puede implicar agregar el archivo de configuración de Firebase (google-services.json para Android o GoogleService-Info.plist para iOS) al directorio raíz de tu proyecto.

3. Integra Firebase en tu código,
4. Prueba y verifica la integración de Firebase.

Documentación de la solución

Repositorio para TPV: https://github.com/ysasuareez/TPV
Repositorio para PDA: https://github.com/ysasuareez/PDA

Enlaces de interés

Documentación Flutter: https://docs.flutter.dev/
Documentación Dart: https://dart.dev/guides
Documentacion Firebase: https://firebase.google.com/docs?hl=es-419