



YelaTPV

Reinventando la caja registradora

Pablo Sirvent Jiménez
2ºDAM B

Título del proyecto	2
Introducción	2
Objetivo	2
Justificación	2
Análisis de lo existente	3
Necesidades empresariales	3
Recursos Humanos	3
Trabajadores de la empresa	3
Contratos de trabajo a realizar	3
Obligaciones en materia de Seguridad Social	3
Prevención de riesgos laborales:	4
Riesgos laborales y medidas preventivas	4
Iniciativa emprendedora:	5
Forma jurídica	5
Trámites administrativos	5
Requisitos funcionales	5
Pero antes... algunos de los objetivos y limitaciones	5
Objetivos específicos	5
Limitaciones	6
Requisitos funcionales principales de la Aplicación (YelaTPV)	6
Personas o entidades (Target)	6
Diseño y análisis	7
Diagrama de Arquitectura	7
Diagrama de casos de uso	8
Codificación	11
Lenguajes de programación empleados	11
Herramientas utilizadas, entornos de desarrollo y plataformas	11
Código relevante	11
Manual de usuario y Requisitos/Instalación	14
Requisitos previos e Instalación	14
Instalando la Aplicación	14
Configuración previa (Parte 1)	15
Configuración previa (Parte 2)	18
Entendiendo el entorno	19
Dashboard	20
Añadiendo categorías a nuestra app	21
Añadiendo productos a nuestra app	22
¡Vamos a vender! Parte 1	24
¡Vamos a vender! Parte 2	26
Fidelización y códigos de descuento	27
Veamos nuestras ventas	29
Gestionando nuestros usuarios	30
Bibliografía	31
Libros, artículos, apuntes y direcciones web	31
Apéndices	31

Título del proyecto

YelaTPV – Reinventando la caja registradora

Introducción

Objetivo

El siguiente documento, tiene como objetivo declarar todos los aspectos relacionados con el proyecto de Fin de Ciclo (DAM) de Pablo Sirvent Jiménez.

El proyecto consiste en un punto de venta de productos (o bien servicios) donde el usuario podrá gestionar su empresa sencillamente.

Uno de los objetivos de YelaTPV es simple: disponer de una venta, rápida y sencilla.

Convirtiendo tu Smartphone o Tablet en un completo, fácil e intuitivo punto de venta, promete diversos usos, desde un punto de venta para una cafetería, pasando por un pub o incluso puede ser útil para la venta de entradas de una feria o conferencia.

Las ideas pueden ser miles con este completo sistema.

Justificación

YelaTPV nace de la necesidad de diversos comercios, de tener un registro de sus ventas y de tener una contabilidad mínima, por lo que YelaTPV se convierte en una fácil e intuitiva solución.

Este punto de venta no reemplaza a un sistema de facturación de su empresa, si no que facilita la tarea de tener centralizado todos los recibos, productos y categorías de una misma empresa.

YelaTPV surge de la solución de evitar los complejos puntos de venta, cajas registradoras complicadas y tecnicismos.

Análisis de lo existente

La mayoría de las soluciones que el mercado ofrece pone muy complicado el hecho de tener un punto de venta, son de pago o directamente complican el hecho de insertar productos.

Normalmente las versiones que existen en la play store hacen demasiadas funcionalidades o requieren de complicadas configuraciones. Incluso existen versiones que no están actualizadas a la ley a día de hoy.

Por ello la filosofía y la innovación de YelaTPV es básica, tener de una venta rápida, fácil y sencilla.

Necesidades empresariales

Recursos Humanos

Trabajadores de la empresa

- Plantilla de programadores de diseño y frontend
- Plantilla de programadores de backend
- Plantilla de posicionamiento Web, estrategia y márketing.
- Recursos humanos, Contabilidad.
- Equipo comercial y de asistencia.

Contratos de trabajo a realizar

Contratos a autónomos, por obra o servicio determinado o bien temporales de 3 a 6 meses si es presencialmente.

Obligaciones en materia de Seguridad Social

Constitución de una sociedad limitada, contratos y altas en seguridad social.

Prevención de riesgos laborales:

Riesgos laborales y medidas preventivas

Debido a que el mantenimiento y desarrollo de estas aplicaciones es diverso, no es obligatorio tener una sede o oficina donde poder desarrollar el transcurso del proyecto YelaTPV”, por lo que la mayoría de personal se encontraría a distancia o en remoto.

En el caso de existir una oficina (bien para desarrollos o comerciales), es necesario destacar los siguientes riesgos laborales:

Factores de riesgo	Riesgo	Medidas
Orden o limpieza	Caídas, choques, contusiones	<ul style="list-style-type: none">• Eliminar obstáculos de las zonas de paso (cableado).• Dejar los pasillos despejados, libres de obstáculos, facilitando el paso de las personas y evitando la caída por tropiezos.• Evitar que los cables eléctricos de máquinas registradoras, ordenadores, teléfono, etc. estén situados en zonas de paso.• Señalizar las zonas mojadas o con presencia de irregularidades
Herramientas	Cortes	<ul style="list-style-type: none">• Seleccionar herramientas de buena calidad.• Sustituir las herramientas en mal estado.• Sólo deben utilizarse para el uso que estén fabricadas.• No llevarlas en el bolsillo.
Contactos eléctricos	Exposición	<ul style="list-style-type: none">• Para desconectar el equipo tire de la clavija, nunca del cable.• No utilice equipos ni instalaciones cuando estén mojados.• Como primera medida, en caso de incidentes o avería, desconecte la corriente y avise al resto de los trabajadores.
Incendios	Quemaduras, inhalación	<ul style="list-style-type: none">• Seguir una estricta política de revisiones de los equipos.• Comunicar cualquier anomalía que se detecte.• Realizar el mantenimiento y retimbrado periódico de extintores y demás equipos contra incendios por personal cualificado.• Ubicar los extintores de incendios en lugares accesibles y bien señalizados.

Iniciativa emprendedora:

Forma jurídica

-Sociedad Limitada, con un capital social inicial de 3000€.

Trámites administrativos

- Cuenta bancaria y depositar capital social inicial
- Solicitud de la denominación de social
- Solicitud de NIF provisional en la agencia tributaria
- Firma de la escritura de constitución de la sociedad en la Notaria
- Inscripción en el registro mercantil
- Solicitud del NIF definitivo
- Alta en seguridad social y alta trabajadores
- Plan de prevención de riesgos laborales

Requisitos funcionales

Pero antes... algunos de los objetivos y limitaciones

Desarrollar e implementar un soporte digital, fácil y sencillo, donde el usuario podrá gestionar sus ventas fácilmente, teniendo una relación entre lo que se está vendiendo y lo que se ha vendido.

El soporte principal y único es Android.

Objetivos específicos

Los objetivos específicos asignados a este proyecto son:

- Desarrollar un punto de venta facturable, con gestión de productos y categorías, a la vez de un punto de fidelización donde podrá el empresario o trabajador poder fidelizar con el cliente.
- En referencia al punto número 1 de objetivos específicos, la aplicación se desarrollará con una base de datos local, sobre Android.
- Subir la primera versión y posteriores versiones al mercado de aplicaciones de Google (A día de hoy, play store).
- Poder vender sin depender de una conexión a Internet activa

Limitaciones

Debido a que es un software que estará en continuo desarrollo, por ahora tiene algunas limitaciones como son las siguientes:

- La aplicación se ejecutará solamente en dispositivos con Android y versiones superiores a 7.0 para disponer de más funcionalidades y nuevas.
- Para el escaneo de códigos de barras, es necesario una cámara con un mínimo de 3 megapíxeles.
- Productos, Categorías y Tickets no se pueden borrar. Usuarios y fidelizaciones sí que son posibles de borrar. Cuestiones de lógica, legalidad y base de datos no hacen posible la eliminación de los cuestionados.

Requisitos funcionales principales de la Aplicación (YelaTPV)

- La aplicación debe poder facilitar la tarea y gestionar las ventas fácilmente
- La aplicación debe poder facilitar y tener una contabilidad de los productos de una venta
- La aplicación debe poder insertar, gestionar los productos de una aplicación.
- La aplicación debe poder tener un sistema de gestión de usuarios con su código PIN
- Poder gestionar categorías y productos, a su vez la posibilidad de tener centralizados todos los productos y asociar productos a muchas categorías.
- Poder escanear y tener un sistema de gestión mediante códigos de barras.

Personas o entidades (Target)

Esta aplicación está enfocada a empresarios o propios trabajadores de una empresa donde utilizarán este sistema a un bajo nivel.

No es necesaria experiencia en el sistema ya que es una aplicación enfocada a una venta fácil, sencilla y intuitiva.

Diseño y análisis

Diagrama de Arquitectura

DIAGRAMA DE LA ARQUITECTURA DE LA APP

1617dam1b Pablo Sirvent Jiménez | May 25, 2018

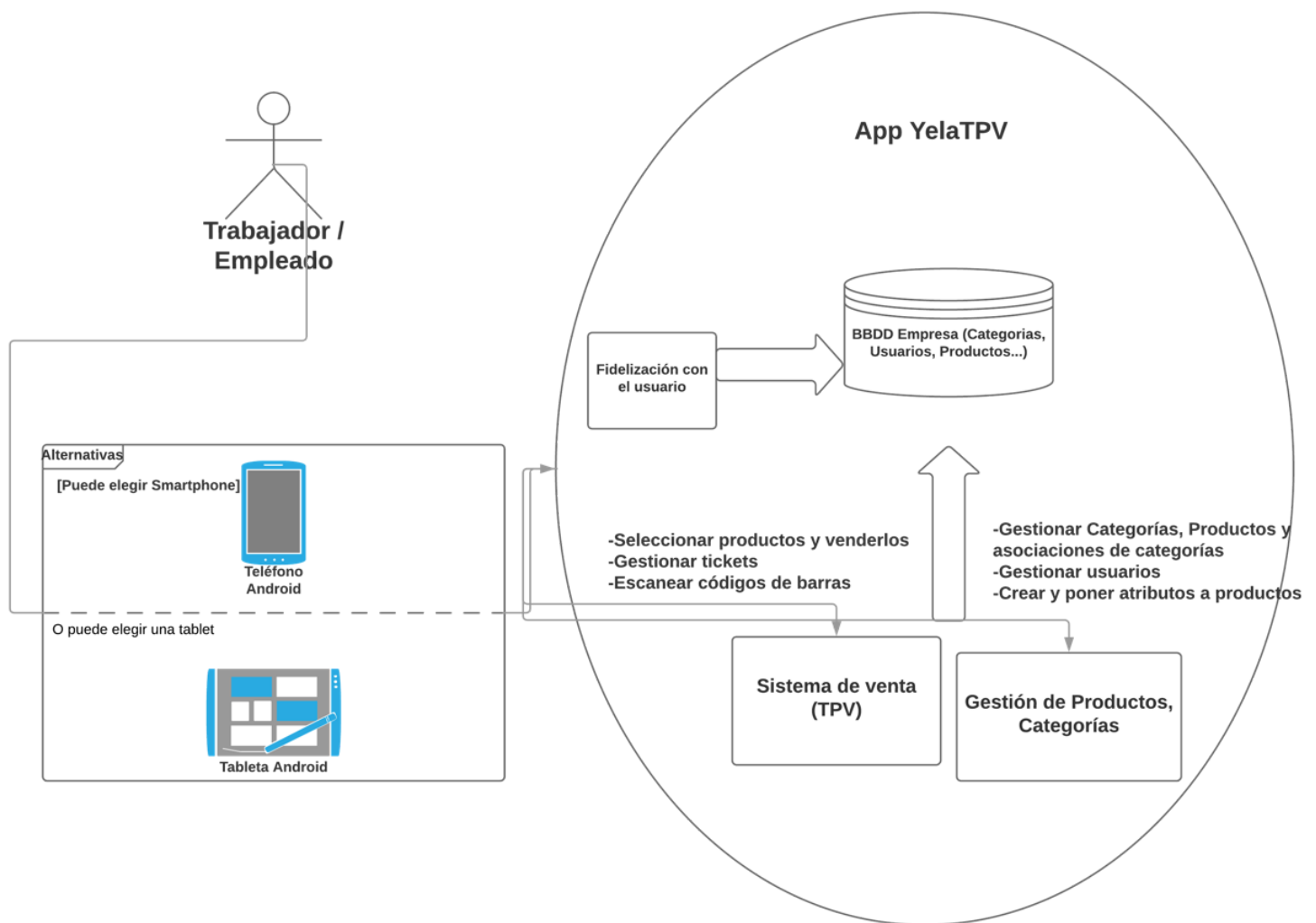


Diagrama de casos de uso

DIAGRAMA DE CASOS DE USO

1617dam1b Pablo Sirvent Jiménez | May 25, 2018

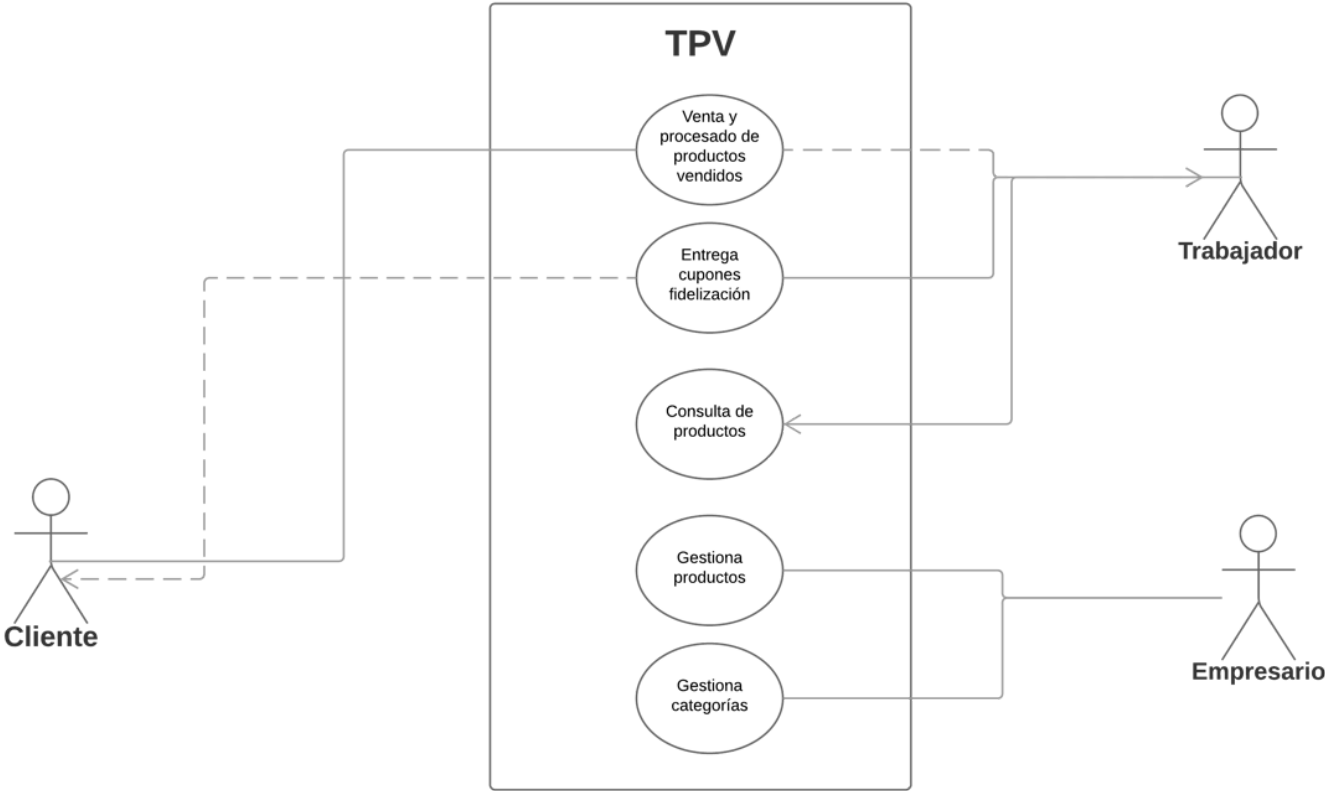


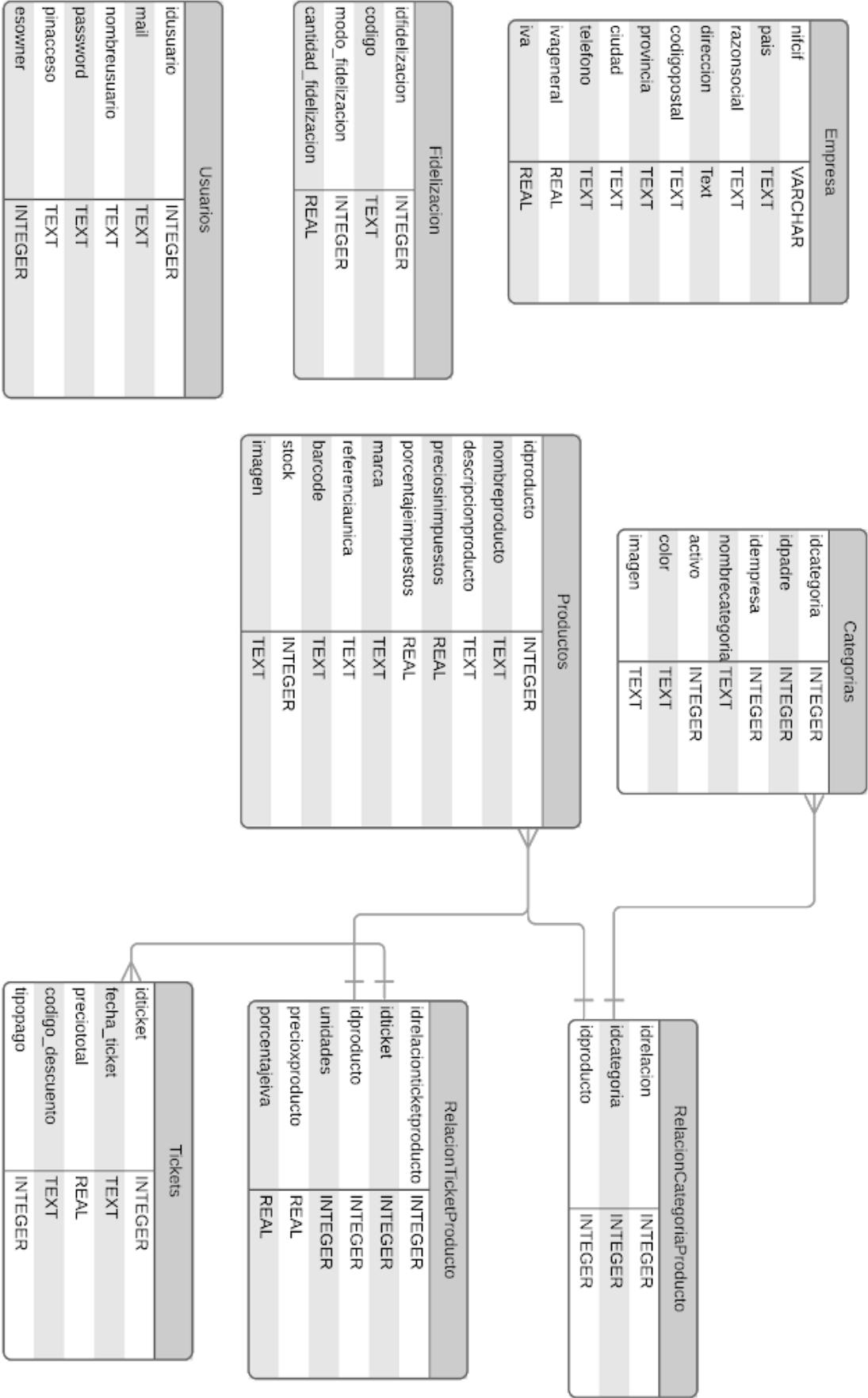
Diagrama de clases

Para evitar un complejo manejo de las clases y las bases de datos, se ha optado por no utilizar relaciones.

<pre>+ RelacionCategoriaproducto Implements Serializable ~ fields ~ getIdrelacion : Integer ~ Idcategoria : Integer ~ Idproducto : Integer ~ constructors ~ methods + getIdrelacion () : Integer + setIdrelacion (Idrelacion : Integer) : void + getIdcategoria () : Integer + setIdcategoria (Idcategoria : Integer) : void + getIdproducto () : Integer + setIdproducto (Idproducto : Integer) : void</pre>	<pre>+ Categoria Implements Serializable ~ fields ~ Idcategoria : Integer ~ Idpadre : Integer ~ Idempresa : Integer ~ nombrecategoria : String ~ activo : Integer ~ color : String ~ imagen : String ~ constructors ~ methods + getIdcategoria () : Integer + setIdcategoria (Idcategoria : Integer) : void + getIdpadre () : Integer + setIdpadre (Idpadre : Integer) : void + getIdempresa () : Integer + setIdempresa (Idempresa : Integer) : void + getNombrecategoria () : String + setNombrecategoria (nombrecategoria : String) : void + getActivo () : Integer + setActivo (activo : Integer) : void + getColor () : String + setColor (color : String) : void + getImagen () : String + setImagen (imagen : String) : void</pre>	<pre>+ Usuario Implements Serializable ~ fields ~ Idusuario : Integer ~ mail : String ~ nombreusuario : String ~ password : String ~ pinacceso : String ~ esowner : Boolean ~ constructors ~ methods + toString () : String + getIdusuario () : Integer + setIdusuario (Idusuario : Integer) : void + getEmail () : String + setEmail (mail : String) : void + getNombreusuario () : String + setNombreusuario (nombreusuario : String) : void + getPassword () : String + setPassword (password : String) : void + getPinacceso () : String + setPinacceso (pinacceso : String) : void + getEsowner () : Boolean + setEsOwner (esowner : Boolean) : void</pre>	<pre>+ Fidelizacion ~ fields ~ Idfidelizacion : String ~ codigo : String ~ modo_fidelizacion : Integer ~ cantidad_fidelizacion : Float ~ constructors ~ methods + getIdfidelizacion () : String + setIdfidelizacion (Idfidelizacion : String) : void + getCodigo () : String + setCodigo (codigo : String) : void + getModo_fidelizacion () : Integer + setModo_fidelizacion (modo_fidelizacion : Integer) : void + getCantidad_fidelizacion () : Float + setCantidad_fidelizacion (cantidad_fidelizacion : Float) : void</pre>
<pre>+ Empresa Implements Serializable ~ fields ~ nifcfr : String ~ pais : String ~ razonsocial : String ~ direccion : String ~ codigopostal : String ~ provincia : String ~ ciudad : String ~ telefono : String ~ Iva_general : Boolean ~ Iva : Float ~ constructors ~ methods + getPais () : String + setPais (pais : String) : void + getRazonsocial () : String + setRazonsocial (razonsocial : String) : void + getNifcfr () : String + setNifcfr (nifcfr : String) : void + getDireccion () : String + setDireccion (direccion : String) : void + getCodigopostal () : String + setCodigopostal (codigopostal : String) : void + getProvincia () : String + setProvincia (provincia : String) : void + getCiudad () : String + setCiudad (ciudad : String) : void + getTelefono () : String + setTelefono (telefono : String) : void + getIva_general () : Boolean + setIva_general (Iva_general : Boolean) : void + getIva () : Float + setIva (Iva : Float) : void</pre>	<pre>+ RelacionTicketProducto ~ fields ~ Idrelacionticketproducto : Integer ~ Idticket : Integer ~ Idproducto : Integer ~ unidades : Integer ~ precioproducto : Float ~ porcentajeIva : Float ~ constructors ~ methods + getIdrelacionticketproducto () : Integer + setIdrelacionticketproducto (Idrelacionticketproducto : Integer) : void + getIdticket () : Integer + setIdticket (Idticket : Integer) : void + getIdproducto () : Integer + setIdproducto (Idproducto : Integer) : void + getUnidades () : Integer + setUnidades (unidades : Integer) : void + getPrecioproducto () : Float + setPrecioproducto (precioproducto : Float) : void + getPorcentajeIva () : Float + setPorcentajeIva (porcentajeIva : Float) : void</pre>	<pre>+ Tickets ~ fields ~ Idticket : String ~ fecha_ticket : String ~ preciototal : Float ~ codigo_descuento : String ~ tipoPago : Integer ~ constructors ~ methods + getIdticket () : String + setIdticket (Idticket : String) : void + getFecha_ticket () : String + setFecha_ticket (fecha_ticket : String) : void + getPreciototal () : Float + setPreciototal (preciototal : Float) : void + getCodigo_descuento () : String + setCodigo_descuento (codigo_descuento : String) : void + getTipoPago () : Integer + setTipoPago (tipoPago : Integer) : void</pre>	<pre>+ Producto Implements Serializable ~ fields ~ Idproducto : Integer ~ nombreproducto : String ~ descripcionproducto : String ~ preciosimpuestos : Float ~ porcentajeImpuestos : Float ~ marca : String ~ referenciaIva : String ~ barcode : String ~ stock : Integer ~ imagen : String ~ constructors ~ methods + getIdproducto () : Integer + setIdproducto (Idproducto : Integer) : void + getNombreproducto () : String + setNombreproducto (nombreproducto : String) : void + getDescripcionproducto () : String + setDescriptionproducto (descripcionproducto : String) : void + getPreciosimpuestos () : Float + setPreciosimpuestos (preciosimpuestos : Float) : void + getPorcentajeImpuestos () : Float + setPorcentajeImpuestos (porcentajeImpuesto : Float) : void + getMarca () : String + setMarca (marca : String) : void + getReferenciaIva () : String + setReferenciaIva (referenciaIva : String) : void + getBarcode () : String + setBarcode (barcode : String) : void + getStock () : Integer + setStock (stock : Integer) : void + getImagen () : String + setImagen (imagen : String) : void</pre>

Diagrama de datos de BD Relacional

Para evitar un complejo manejo de las clases y las bases de datos, se ha optado por no utilizar relaciones.



Codificación

Lenguajes de programación empleados

Java como principal lenguaje de programación.

Herramientas utilizadas, entornos de desarrollo y plataformas

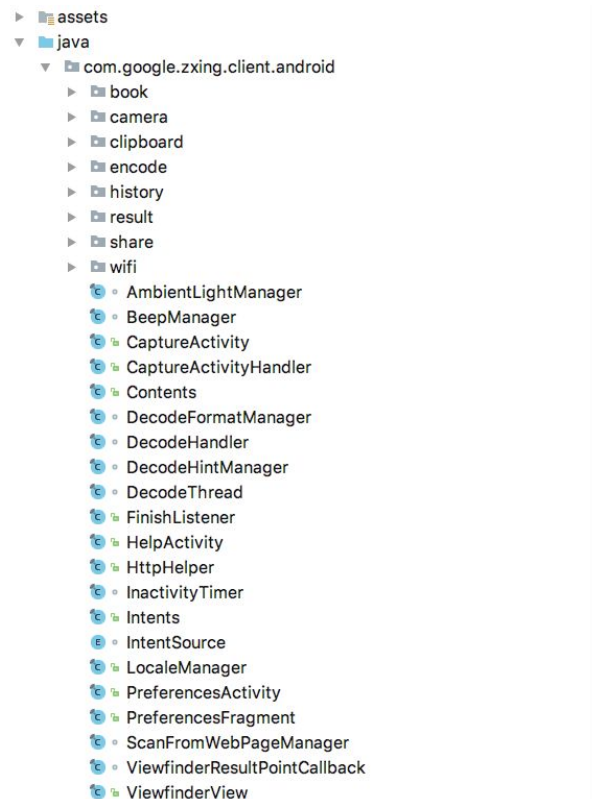
Desarrollo y programación:

- S.O: Macintosh
- IDE: Android Studio 3.0.1
- Emulador de Android Integrado en versión 7.0
- Plugin de generación de clases integrado en Android Studio: SimpleUML Diagram
- Gestor de bases de datos: DB Browser for SQLite
- Creación de Diagramas de uso, de modelo entidad/relación y de arquitectura de aplicación: LucidChart

Código relevante

Se ha utilizado un framework externo integrado (en lugar de tenerla instalada, ya tenerla integrada en la aplicación) para poder escanear barcodes (o códigos de barras).

El framework se llama zxing. Abajo podemos ver una imagen donde están todas las clases necesarias para poder hacer un scan de un código de barras.



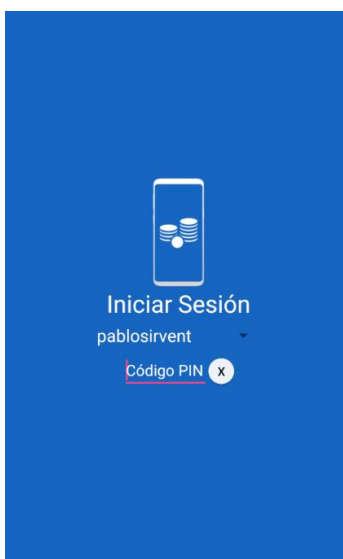
Una de las peculiaridades de esta aplicación es el uso de Fragments para la gestión de cada pantalla. Por lo que ha facilitado mucho la implementación de las secciones del sistema.

Disponemos de un método llamado “setFragment”, que en función del parámetro que le pase el navigationview, cambiará el fragment en función de lo requerido. En el caso del número 0, es el dashboard o página principal.

```
public void setfragment(int position) {
    FragmentManager fragmentManager;
    FragmentTransaction fragmentTransaction;
    switch (position) {
        case 0:
            // Dashboard
            Dashboard dashboard_fragment = new Dashboard();
            if(nuevo){
                dashboard_fragment.setNuevo(true);
            }else{
                dashboard_fragment.setNuevo(false);
            }
            fragmentManager = getSupportFragmentManager();
            fragmentTransaction = fragmentManager.beginTransaction();
            fragmentTransaction.setCustomAnimations(android.R.anim.fade_in, android.R.anim.fade_out);
            fragmentTransaction.replace(R.id.fragment, dashboard_fragment);
            fragmentTransaction.addToBackStack(null);
            fragmentTransaction.commit();
            break;
    }
}
```

- ▼ Fragments
- Ajustes
- CategoriaGestion
- Dashboard
- Fidelizacion
- MiEmpresa
- ProductoGestion
- Tickets_TPV
- TPV
- UsuarioGestion

También existe un sistema de login mediante PIN muy básico, que permite cambiar entre distintos usuarios del sistema



```
campo_pin.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void afterTextChanged(Editable editable) {
        BDUsuarios bdusuarios = new BDUsuarios(getApplicationContext(), name: "Usuarios", factory: null, version: 1);
        Cursor usuario_cursor = bdusuarios.getUsuarioCursorPorID(1);

        String campopin_bd = "nodebesalir";
        if (usuario_cursor.moveToFirst()) {
            while (!usuario_cursor.isAfterLast()) {
                campopin_bd = usuario_cursor.getString(usuario_cursor.getColumnIndex( s: "pinacceso"));
                usuario_cursor.moveToNext();
            }
        }
        usuario_cursor.close();
        String introducido = campo_pin.getText().toString();

        if (introducido.equals(userselected.getPinacceso().toString())) {
            Intent intent = new Intent(getApplicationContext(), Home.class);
            intent.putExtra( name: "IDUSER_LOG", userselected.getIdusuario().toString());
            startActivity(intent);
            finish();
        }
    }
});
```


En el código puesto anteriormente, detectamos mediante un listener del texto, si el pin coincide con el pin del usuario seleccionado. Es útil y muy seguro utilizar un sistema de pin.

Una curiosidad que tiene en especial, es un splashscreen con un timer, que da un juego al diseño de la aplicación muy especial.

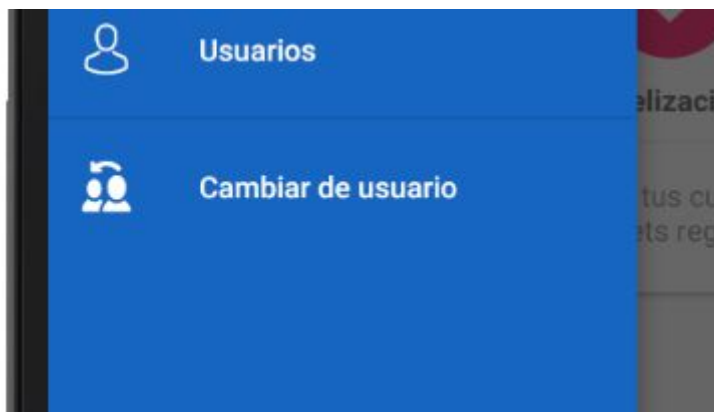
Una vez cargado el splashscreen durante 1 segundo, buscamos si existen empresas activas o registradas, si no existen (el caso de una nueva descarga de la app o borrado de datos), mostraremos un intent del registro de la empresa, si no, mandará a la pantalla de Login mediante PIN.



```
boolean empresa_registrada = false;
BDEmpresa bdempresa = new BDEmpresa( context: this, name: "Empresa", factory: null, version: 1);
Cursor empresas_activas = bdempresa.getEmpresaCursor();

if(empresas_activas.getCount() != 0){
    empresas_activas.close();
    Intent intent = new Intent( packageContext: this, Login.class);
    startActivity(intent);
    finish();
}else{
    empresas_activas.close();
    Intent intent = new Intent( packageContext: this, RegistroEmpresa.class);
    startActivity(intent);
    finish();
}
```

También podremos cambiar de usuario en cualquier momento y cerrar nuestra sesión sin problema alguno.



Manual de usuario y Requisitos/Instalación

Requisitos previos e Instalación

Bienvenido a YelaTPV. Antes de empezar con la instalación, es necesario revisar algunas cosas para la instalación:

- Teléfono/Tablet con Android 7.0 o superior (Superior a API 24)
- Cámara de 3Megapíxeles para el escaneo de códigos de barras
- Pantalla de 4 o más pulgadas
- Conexión a Internet si la app es descargada por la Play Store*

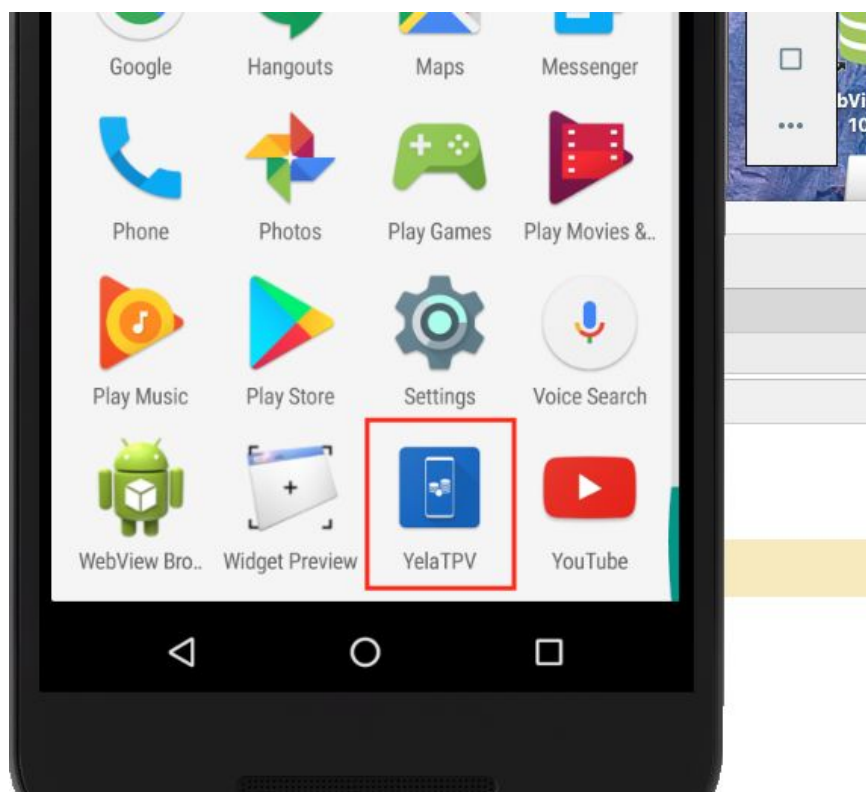
*Para su uso después de la descarga no es necesaria una conexión a Internet, por lo que se puede usar sin problema alguno.

Instalando la Aplicación

En el caso de descargar la aplicación por la Play Store, se puede buscar con el nombre de "YelaTPV".

En el caso de bajarse una APK generada, puedes instalar la aplicación directamente sin nada previo.

La base de datos se crea cuando una empresa es creada, por lo que no es necesaria una configuración previa.



Configuración previa (Parte 1)

Una vez arrancada la aplicación, veremos la pantalla de inicio o splashscreen tal que así:

-----1 segundo ----->




Podemos proseguir con la instalación y configuración, clickamos “¡VAMOS!”

En la siguiente pantalla podrás rellenar los datos de tu empresa que son los siguientes:

- Razón Social → Para la creación de tickets
- CIF/NIF → Para la identificación en tickets (por ley)
- Dirección
- Provincia
- Ciudad
- Código Postal
- Teléfono

-----Rellenamos campos ----->



Datos de empresa

Razón Social

CIF/NIF

Dirección

Provincia

Ciudad

Código Postal

Teléfono

SIGUIENTE



Datos de empresa

Restauracion Sirvent S.L

48768980X

Calle Asunción 3

Alicante

Alicante

03015

664894954

SIGUIENTE

Apretamos “Siguiente”.

Verás una pantalla como la siguiente, donde casi habrás terminado. La configuración es mínima para no entretener demasiado.

Aquí podremos configurar cómo deseamos el IVA. Si un específico para cada producto (Donde lo editaremos en la ficha de producto más adelante) o si tienes un IVA para todos los productos general. ¡Recuerda que esto no se puede cambiar!



¡Ya casi hemos terminado!

¿Cómo deseas el IVA?

☐ IVA específico para cada producto

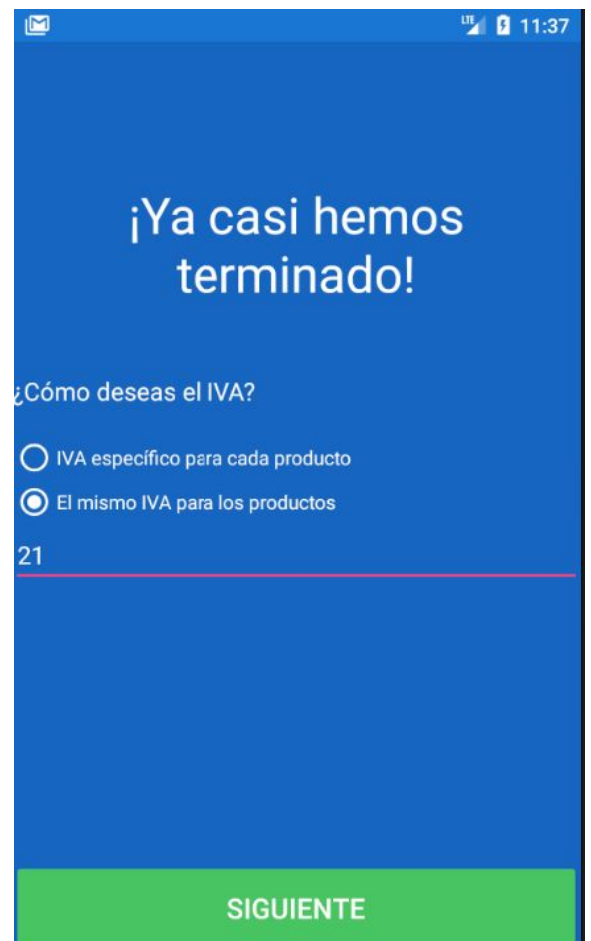
☐ El mismo IVA para los productos

Para la demostración usaremos el IVA específico de cada producto, ya que nuestros productos no usan siempre el 21% (Actual en mayo 2018).

Ejemplo 1



Ejemplo 2



Seguiremos y apretaremos “Siguiente” otra vez.

Como comprobaremos, ya es el último paso. Sí, así de fácil.

No requiere nada más por nuestra parte, ahora solamente tocará configurar las credenciales de acceso del primer usuario del sistema, donde lo veremos en el siguiente punto.

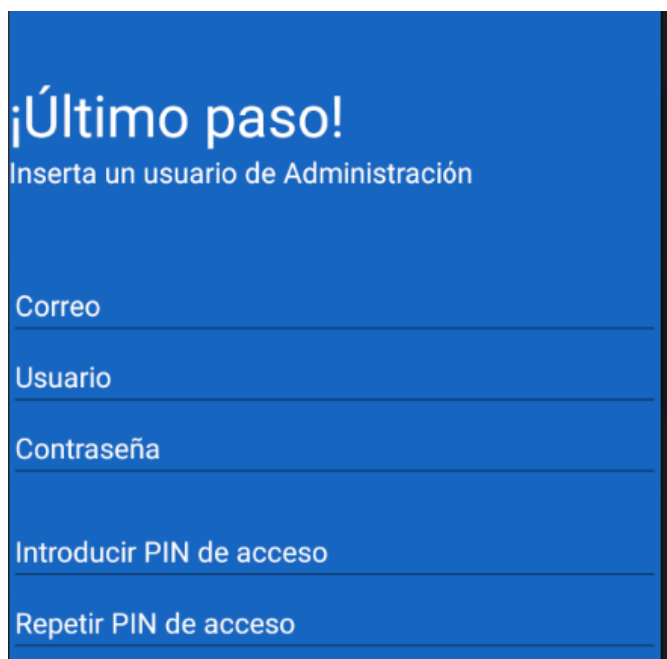
Configuración previa (Parte 2)

Este apartado es muy pequeño y solamente nos detendremos para ver cómo el dueño o administrador del punto de venta, podrá configurar fácilmente sus credenciales.

Tendremos 5 campos:

- Correo
- Usuario → Para identificar a los distintos usuarios
- Contraseña → Solamente se usará en futuras actualizaciones
- Pin 1 → Pin superior a 2 caracteres
- Pin 1 repetido → Para verificar el PIN 1

Debería quedar de algo así -----> A algo así



¡Último paso!

Inserta un usuario de Administración

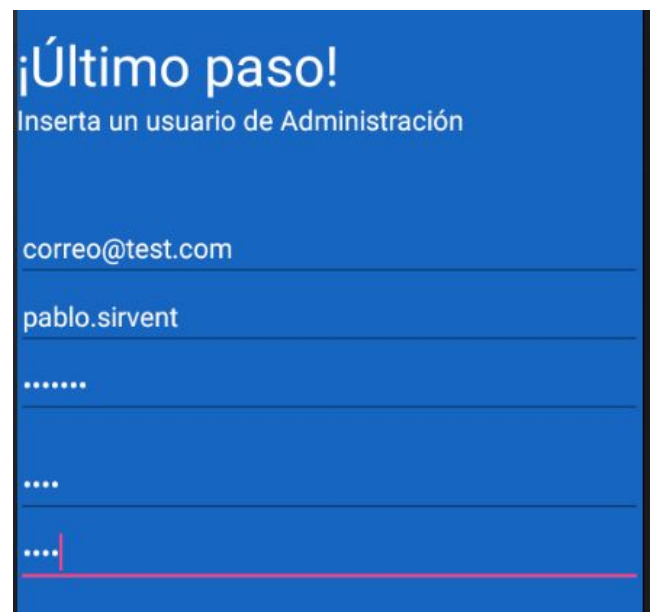
Correo

Usuario

Contraseña

Introducir PIN de acceso

Repetir PIN de acceso



¡Último paso!

Inserta un usuario de Administración

correo@test.com

pablo.sirvent

.....

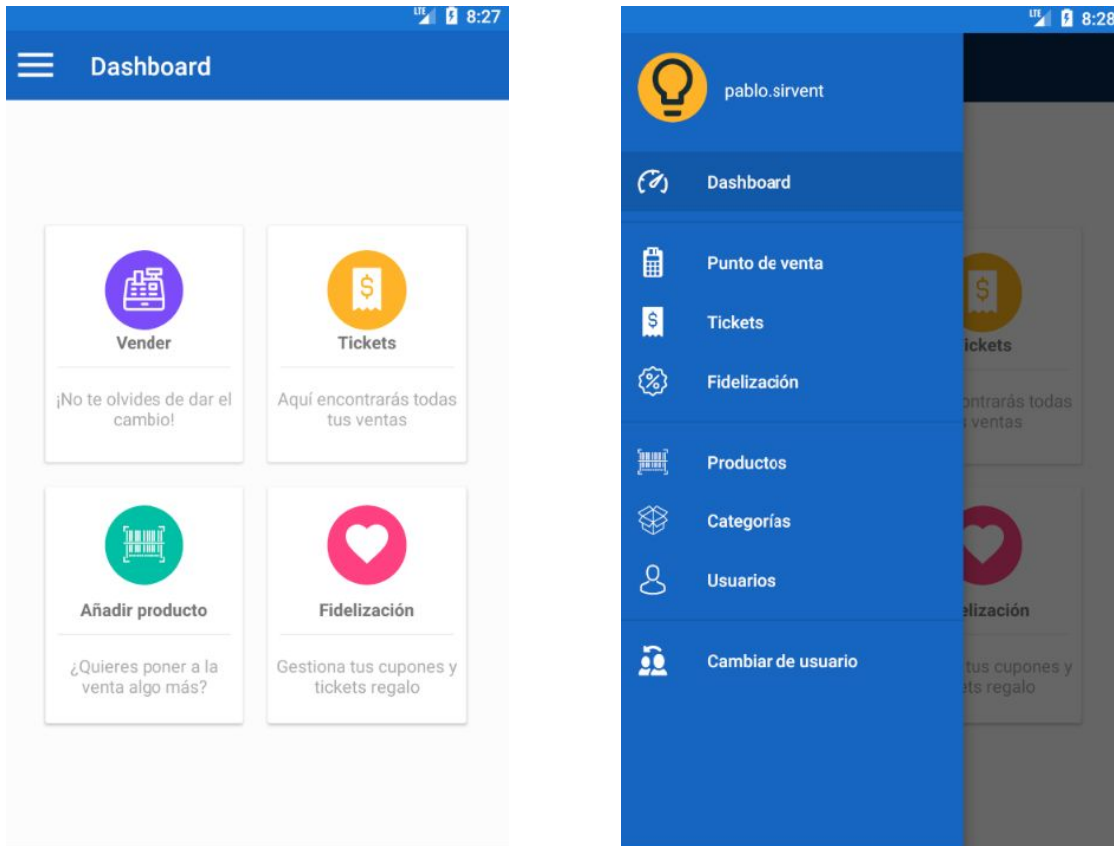
....

....

Finalizamos y... ¡Ya está! En los siguientes apartados podemos continuar con el manual de YelaTPV.

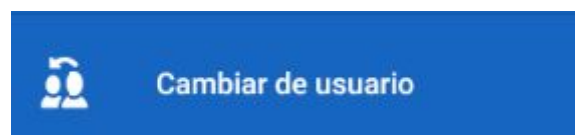
Entendiendo el entorno

Una vez configurado todo, nos saltará a la pantalla principal, donde tendremos un menú de navegación al lateral y una pantalla principal, como la siguiente.



En la primera pantalla (dashboard) tendremos un resumen de las principales acciones que podremos hacer. Pero donde nos tendremos que fijar realmente es en el menú de navegación. En los siguientes apartados explicaremos cada sección y lo que podremos hacer en ellas.

Existe una sección al final, que es “Cambiar de usuario”, donde podremos cambiar de usuario fácilmente si tuviéramos más usuarios, pero esto lo explicaremos más adelante, en la sección “Usuarios”.



Dashboard

Para entender antes el resto del entorno, tendremos que ver la pantalla principal, donde realmente será donde tendremos nuestras acciones principales y donde tendremos la mayoría de actividad de nuestra empresa.



Vender

¡No te olvides de dar el cambio!

Botón Vender:

Aquí nos llevará a la sección “Punto de venta” del menú de navegación. Aquí podremos encargarnos de hacer las ventas oportunas que tengamos. Pero antes...



Tickets

Aquí encontrarás todas tus ventas

Botón Tickets

Aquí podremos visualizar todos tickets o ventas hechas con el punto de venta, podremos ver los artículos vendidos, cantidades totales, forma de pago, etc.



Añadir producto

¿Quieres poner a la venta algo más?

Botón añadir producto

Este botón nos llevará directamente a la sección de productos totales, donde podremos añadir un producto y personalizarlo a nuestro gusto.



Fidelización

Gestiona tus cupones y tickets regalo

Botón Fidelización

Con esta acción podemos gestionar todo lo relacionado con cupones o tickets de regalo (son lo mismo, la única diferencia es que pueden ser cupones de descuento en porcentaje o tickets de regalo de x euros)

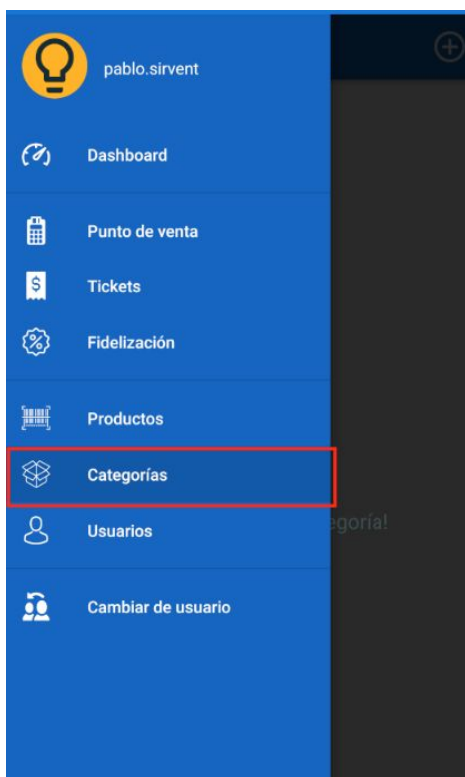
Añadiendo categorías a nuestra app

Para hacer uso de los productos tendremos que añadir ciertas categorías. Si bien es cierto que podemos tener productos sin categorías pero estos productos no se visualizarán si tenemos categorías añadidas al sistema.

En nuestro caso elegiremos que sí que tendremos categorías y productos asociados a estas categorías. Recuerda que tenemos que tener permisos de administración para poder hacer estas gestiones.

Cómo añadir categorías:

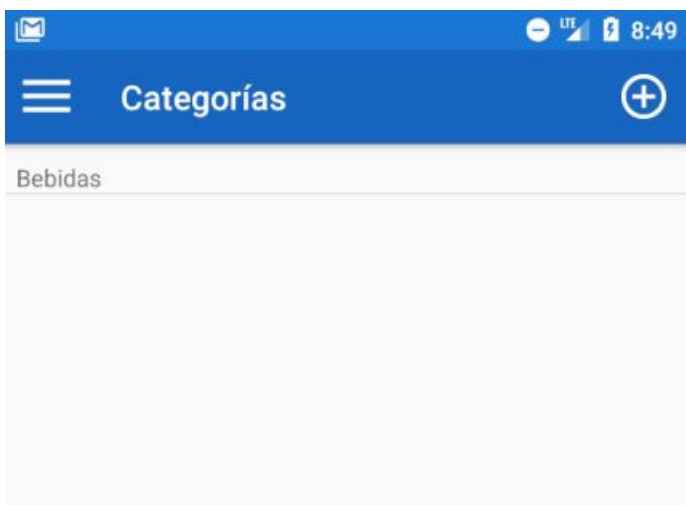
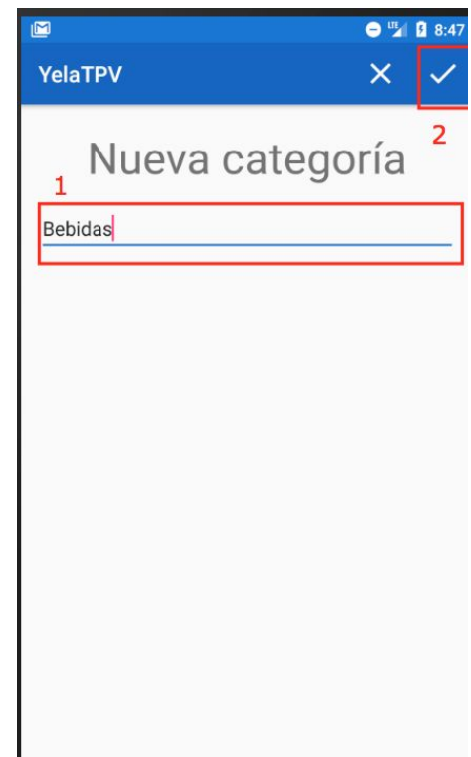
1



2



3

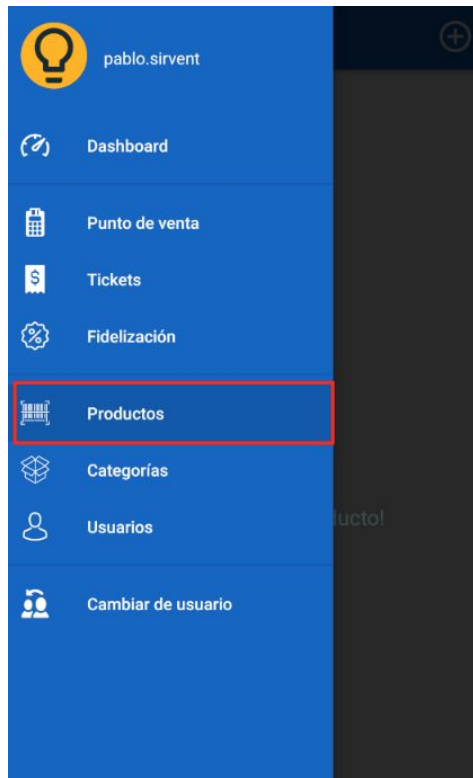


¡Listo! ¿Has visto qué fácil?

Ya tendríamos una categoría añadida a nuestra aplicación.

Ahora el siguiente paso es añadir productos.

Añadiendo productos a nuestra app



Para entender bien lo que se nos pide, es necesario aclarar los siguientes aspectos de un producto:

A screenshot of the 'Nuevo producto' form. The form has a blue header with the text 'YelaTPV' and a close button (X) and a checkmark button (✓). The main title is 'Nuevo producto'. Below the title, there is a camera icon and a button labeled 'SELECCIONAR IMAGEN'. Underneath, there is a dropdown menu labeled 'Selecciona categoría:' with 'Bebidas' selected. Below the dropdown, there is a text input field labeled 'Referencia única *'. To the right of this field is a 'SCAN' button. Below the 'Referencia única' field, there are several more text input fields: 'Código de barras', 'Nombre Producto *', 'Descripción acerca del producto', 'Precio sin impuestos *', 'Nombre de la marca', and 'Stock actual *'.

- Imagen : Podemos seleccionar una imagen o bien hacerle una fotografía a nuestro producto
- Selección de Categoría: Aquí seleccionaremos la o las categorías asociadas a nuestro producto
- Referencia única: La referencia única es para diferenciar entre distintos productos, si bien no controlamos aún si esta referencia puede repetirse pero es más que nada para que lleves un control de tus productos
- Código de barras: Puedes escanear o insertar un código de barras (da igual la codificación)
 - Nombre del producto
 - Descripción del producto
 - Precio sin impuestos
 - Nombre de la marca
- Stock actual: Es meramente informativo y en futuras versiones se harán descuento de stocks, por ahora no está en uso.

Un ejemplo de cómo podría insertarse un producto sería este:

The image displays two screenshots of a mobile application interface. The left screenshot shows the 'Nuevo producto' (New product) form. It has a blue header with 'YelaTPV' and a camera icon. The form includes a 'SELECCIONAR IMAGEN' button, a category dropdown set to 'Bebidas', a reference field 'COCACOLA_REF_00001', a barcode field '8888000076009' with a 'SCAN' button, and several text input fields containing 'Lata Coca Cola 330', '330ML', '0.70', 'Coca Cola', and '1000'. The right screenshot shows the 'Productos' (Products) list. It has a blue header with a menu icon and a plus icon. The list displays the newly added product 'Lata Coca Cola 330' with its reference 'COCACOLA_REF_00001' and a small image of a Coca-Cola can.

Es obligado tener permisos de administración del usuario para poder editar o añadir productos

CONSEJOS a la hora de crear productos:

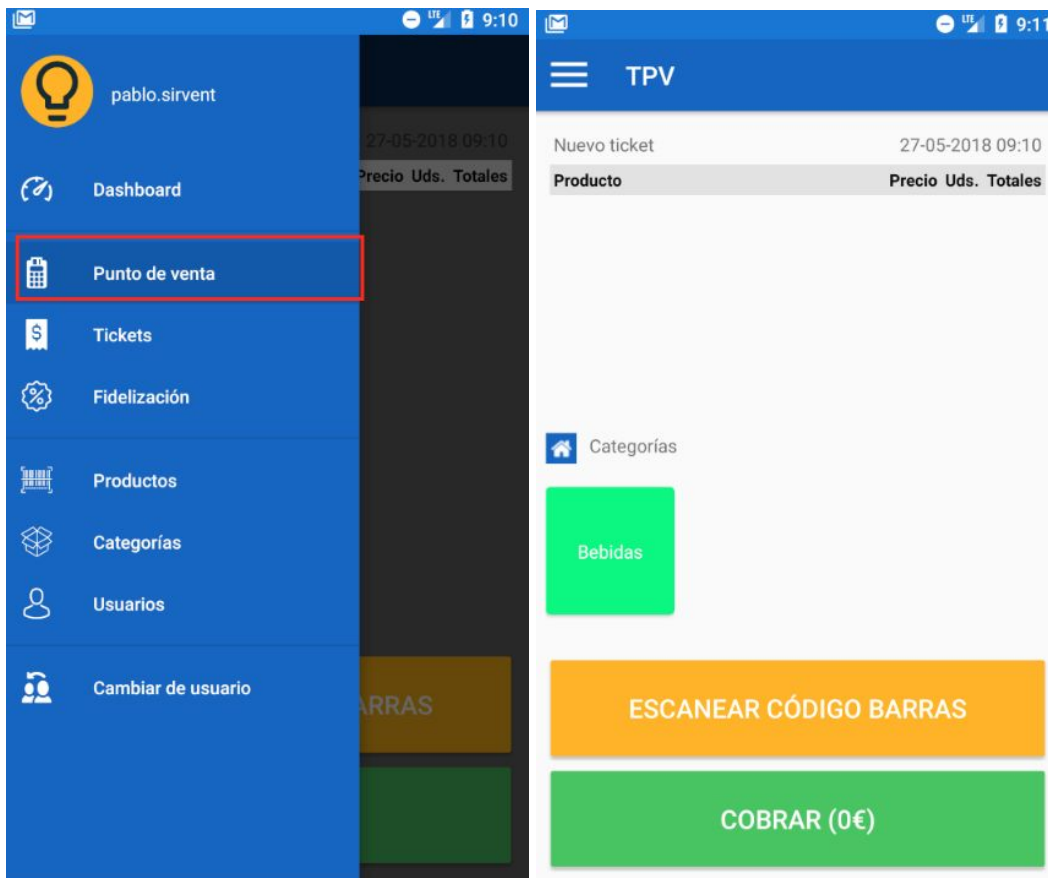
- Para aprovechar al máximo esta herramienta, es conveniente que las imágenes estén sobre fondo blanco a poder ser.
- Puedes seleccionar distintas categorías según las necesites
- Nuestra configuración de IVA era un 21% a todos los productos, como bien hicimos al principio de la configuración de la empresa. Por lo que aquí no se nos pide el IVA en ningún momento, pero puedes insertarlo si quieres. Como verás en la imagen de abajo, nosotros hemos elegido el mismo IVA y no nos aparece el campo para poder editar el IVA

The image shows a dialog box with a blue background. The title is '¿Cómo deseas el IVA?'. There are two radio button options: 'IVA específico para cada producto' (unselected) and 'El mismo IVA para los productos' (selected). The number '21' is visible at the bottom left of the dialog box.

¡Vamos a vender! Parte 1

Ya tendremos casi todo listo. No hace falta configuración previa ni nada, la mayoría de cosas ya las tenemos hechas.

Vamos a entender el entorno del punto de venta.

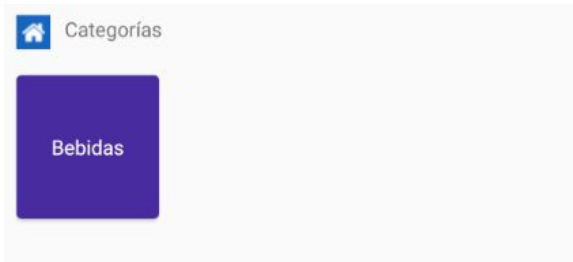


Ya estamos en el punto de venta. Como podremos observar tenemos distintas secciones:



Aquí tenemos el listado de productos que vamos añadiendo. Tenemos:

- Fecha de creación
- Productos que se van añadiendo
- Precio de cada producto, cantidad y sus totales

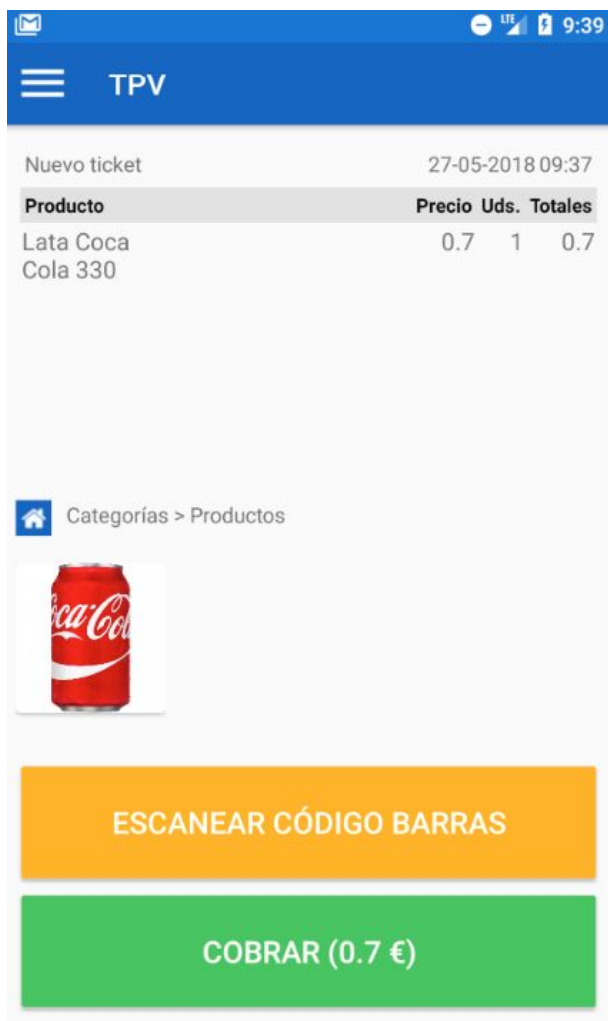


Aquí tenemos el visor de categorías y productos.

Tenemos un botón de "Home" por si accedemos a una categoría y no añadimos nada. Si clickamos "Bebidas" tendremos lo siguiente:



Tendremos los productos que hayamos insertado en esa categoría. Si el producto tiene imagen, su texto desaparece. Podemos clickar en el producto para añadirlo a la venta.



Ya tendremos el producto añadido.

Si queremos aprovechar para añadir otro producto de otra categoría, tendremos que clickar en el botón de la pequeña casa para volver a todas las categorías

Si quisiéramos borrar un producto, tan sólo habría que mantener la línea del producto a borrar.

Aprovechamos esto para explicar los dos botones de abajo. Esta es la última sección de la vista del punto de venta.

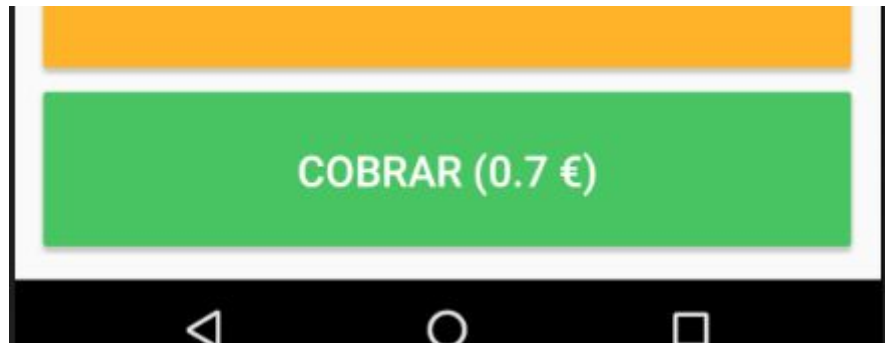
Botón "Escanear Código barras" → Abre el visor o la cámara para escanear un producto. Si escanea algo, y existe en el sistema, lo insertará.

Botón "Cobrar" → Esto lo veremos en la siguiente parte.

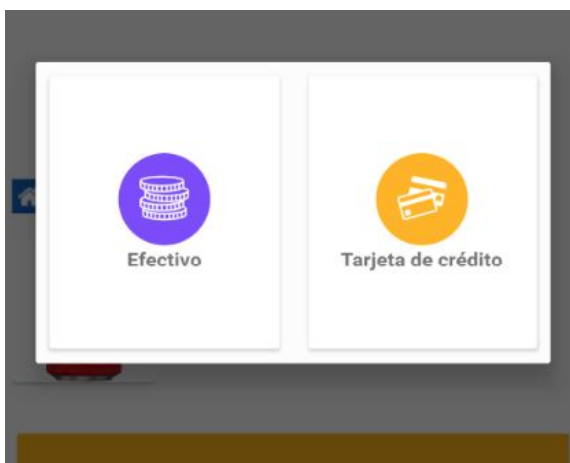
¡Vamos a vender! Parte 2

Ya hemos visto cómo añadir productos a la venta. Ahora sólo nos falta, la finalización de un ticket o una venta.

Clickaremos en el botón de cobrar donde nos saldrá el precio final.



Nos saldrá la siguiente ventana:



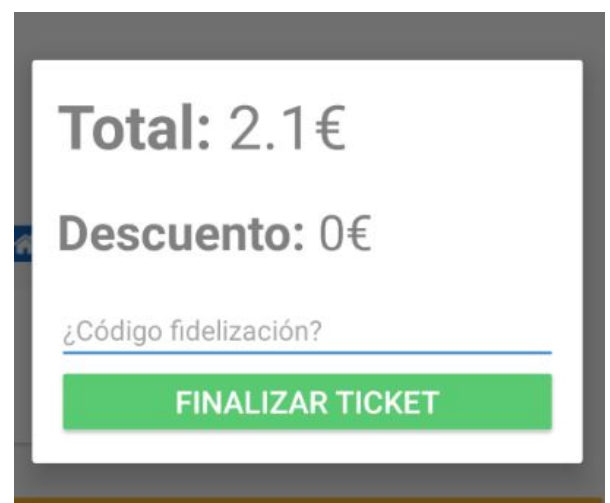
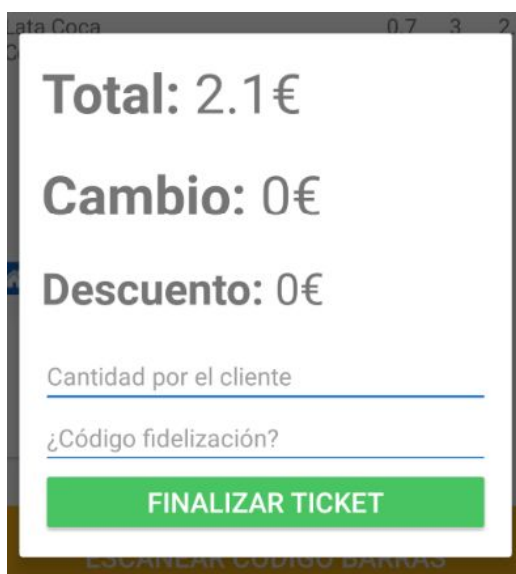
Botón “Efectivo” → Si la persona en cuestión nos ha pagado en efectivo.

Botón “Tarjeta de crédito” → Si nos han pagado con tarjeta de crédito.

Para finalizar ticket, indicaremos las cantidades de cambio y finalizaremos.

Efectivo

Tarjeta crédito

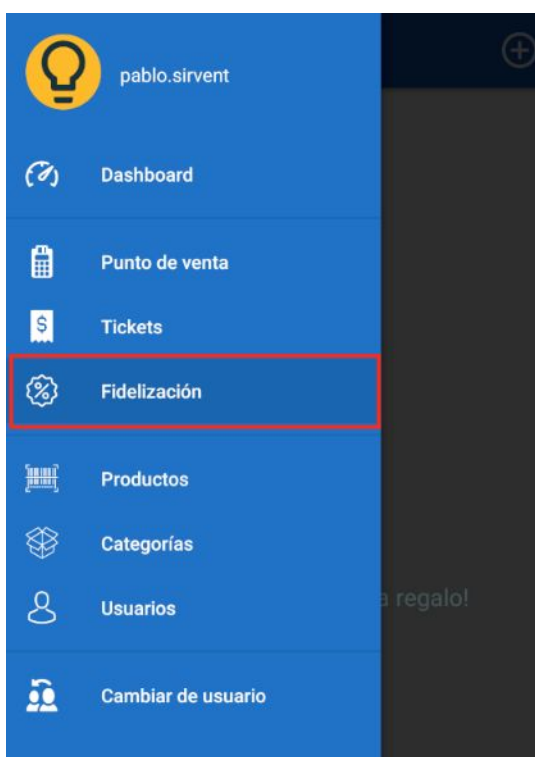


Fidelización y códigos de descuento

Como habrás visto en las anteriores pantallas, a la hora de finalizar ticket, tienes un campo para rellenar un código de fidelización.

Esto nos servirá para indicar descuentos tanto de % como de € a nuestros clientes.

Se gestionan en la siguiente sección:



Podemos añadir un código de fidelización de la siguiente manera:

Código Ticket: Código ÚNICO de nuestro ticket regalo.

Euros o porcentaje: Indicaremos la cantidad de lo que queramos descontar y exactamente si será un porcentaje o una cantidad en euros.

YelaTPV

Nuevo ticket regalo

REGALO5

☒ Euros ☐ Porcentaje

5.0

Lo aplicaremos a la hora de vender en la siguiente pantalla:

DESCUENTO. 0€

Cantidad por el cliente

¿Código fidelización?

FINALIZAR TICKET

DESCUENTO. 0€

Cantidad por el cliente

REGALO5

FINALIZAR TICKET

Total: 5.0€

Cambio: 0€

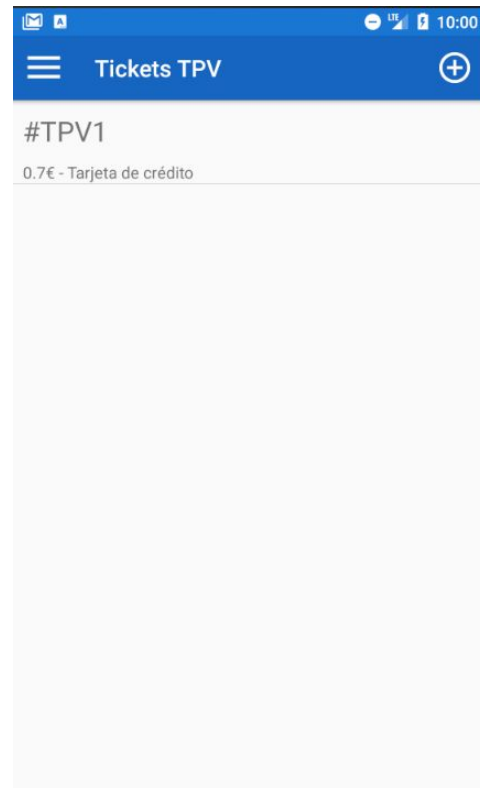
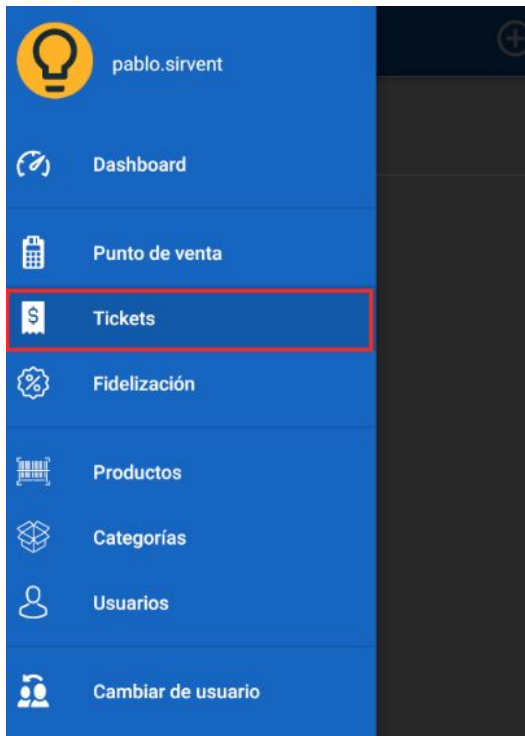
Descuento: 5.0€

Como hemos visto, acabamos de aplicar un descuento de 5€ mediante un código que hemos añadido.

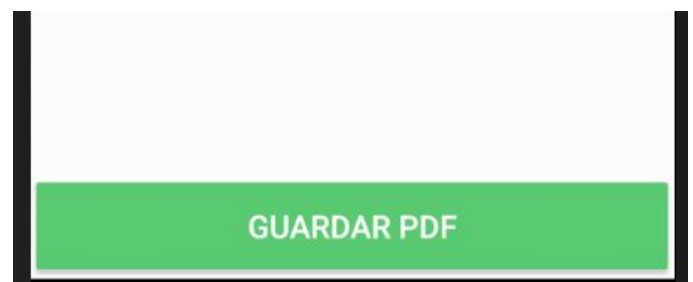
Veamos nuestras ventas

Una de las funciones que probablemente queramos tener, es ver nuestras ventas hechas.

Para ello iremos a la siguiente sección:



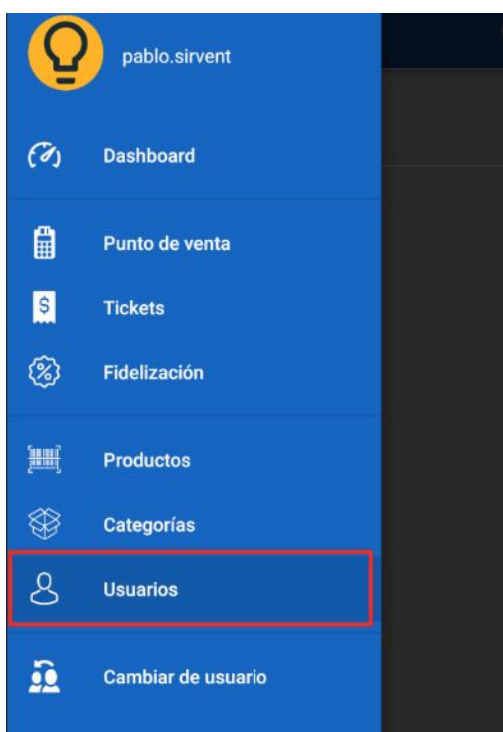
Tendremos un resumen de los tickets que hemos hecho a lo largo de nuestra app. Podremos incluso descargar el PDF para imprimirlo, donde se ubicará en la carpeta Descargas de nuestro móvil con el número del ticket como nombre (TPVXXX.pdf)



Gestionando nuestros usuarios

Uno de los apartados que podemos manejar en nuestra app, son los usuarios y gestionar sus credenciales. Esto sólo es posible si tenemos permisos de administración en el propio usuario.

Iremos a:



Esta sección siempre tendrá un usuario, que es el creador o administrador de la app.

Si clickamos o añadimos uno nos aparecerá lo siguiente:

A screenshot of the 'Nuevo usuario' form. The title bar at the top says 'YelaTPV' with a close button (X) and a checkmark button (✓). The form has the title 'Nuevo usuario' and four input fields: 'Correo electrónico', 'Nombre de usuario', 'Contraseña', and 'PIN acceso'. At the bottom, there is a checkbox labeled 'Permisos de administración'.

- Correo electrónico: Nuestro correo electrónico
 - Nombre de usuario: para identificar a los usuarios
 - Contraseña: no tiene uso, pero en un futuro se usará para iniciar sesión en futuras características (estadísticas web, por ejemplo).
 - PIN de acceso: Para establecer un PIN de acceso
 - Permisos de administración: Si es un trabajador o un administrador, no podrá crear productos ni categorías, ni gestionar usuarios



Ejemplo de cómo sería la pantalla de Login

Bibliografía

Libros, artículos, apuntes y direcciones web

La mayoría del código está hecha desde tutoriales y apuntes del profesorado del IES DOCTOR BALMIS.

-CLICK RECYCLERVIEW: <https://antonioleiva.com/recyclerview-listener/>

-SPINNER MULTISELECTION:

<http://androiddhina.blogspot.com.es/2016/02/android-multi-selection-spinner.html>

-BARCODE SCANNER: <https://github.com/zxing/zxing>

-LUCID CHART: <https://www.lucidchart.com/>

-FRAMEWORK PDF's (iTextG): <https://developers.itextpdf.com/itextg-android>

-DB BROWSER: <http://sqlitebrowser.org/>

-Material Design y RecyclerView propio de Google

-Y otras pequeñas dudas resueltas con <https://stackoverflow.com/>

Apéndices

- TPV: Terminal de punto de venta
- Point of sale: Punto de venta
- Punto de venta: punto donde se reúnen los elementos necesarios para crear una mínima factura de un producto o servicio contra clientes
- Stock: Cantidades de un producto en un almacén o ubicación
- Ticket: Formato pequeño de una factura legal
- Categoría o familia: Asociación ficticia que se le da a un producto para englobarlo con otros productos de la misma familia (Una coca cola, pertenecería a la familia o categoría bebidas, por ejemplo)
- Código de barras: elemento gráfico representado en barras que representa una línea de números