



Ciclo 2019-1
Ingeniería de Software I

Segundo Avance

Alumnos:

- Jhony Angulo
- Sebastián Hurtado
- Leonidas García
- Indhira Ramírez
- Francisco Mejía
- César Salcedo

Profesores:

- Jesús Bellido
- Yamilet Serrano

Fecha de entrega: 06/05/2019
Barranco, 2019

I. Target User Profile

Este proyecto contempla dos tipos de usuarios: comensales y menús. El usuario o comensal es el ciudadano que trabaja/estudia y almuerza fuera de casa; normalmente, en un menú. El menú o restaurante en el que se enfoca esta propuesta es aquel que se encuentra cerca a un centro laboral o de estudio.

¿Por qué ellos?

Este sector es de rápido crecimiento (4% mensual desde Diciembre 2018 según INEI) y tiene un gran volumen de usuarios, ya que cerca del 80% de los peruanos come fuera de su hogar (Arellano Marketing). De aquellos, más del 60% pagan con efectivo, lo cual los ubica dentro del grupo de potenciales clientes de Yape.

Según el INEI, los costos del menú varían por distrito: desde S/.7 en Comas o Santa Anita hasta más de S/.9 en San Isidro o Santiago de Surco. Dado que los restaurantes tienen *tickets* que están dentro del rango admitido por Yape (promedio de S/.20), es factible soportar sus transacciones con la aplicación.

A esto le sumamos la oportunidad de incentivar a más universitarios a obtener Yape, ya que solo el 15% de ellos está afiliado a algún banco (BCP), mientras que el 70% de peruanos ya está bancarizado.

Con esta propuesta se busca alcanzar al 80% de peruanos que busca comer fuera de casa, por motivos laborales o de estudio, para ofrecer el beneficio *core* de Yape más los propuestos por este proyecto: *YapePoints*.

II. Propuesta de Valor

Para los **menús** o **restaurantes**, se brinda un sistema de pagos por celular (*Yape*) y administración de información y cupones (*YapeMaster*). De esta manera, el dueño de un restaurante puede agilizar el pago de servicios, tener mayor control de sus ganancias y promocionar su negocio.

Para los **comensales** o **usuarios**, se ofrece la posibilidad de pagar con celular (*Yape*) y ofrecer beneficios para sus menús favoritos (*YapePoints*).

III. User Stories

Las *User Stories* se han dividido en *Must-Have*, *Nice-to-Have* y *Out of Scope* user stories. Esta información ha sido registrada en *Issue Tracker* de *Github* (*Issues*).

<https://github.com/cs2901/yape-bcp-project-yapepoints/issues>

IV. Use Cases

System	YapePoints
Actor	Usuario
Use Case	UC-01 Usuario paga un menú
Preconditions	El usuario está logueado.
MMS	<ol style="list-style-type: none"> 1. El usuario escanea el código QR por el pago que le proporciona el cajero. 2. El usuario acepta el pago por el monto escaneado.

System	YapePoints
Actor	Restaurante
Use Case	UC-02 Restaurante recibe pago
Preconditions	El usuario está logueado.
MMS	<ol style="list-style-type: none"> 1. El cajero genera el código QR con precio del servicio. 2. El cajero muestra el código QR al cliente. 3. El cliente realiza el pago al escanear el código

System	YapePoints
Actor	Usuario
Use Case	UC-03 Usuario quiere encontrar un lugar cercano para comer
MMS	<ol style="list-style-type: none"> 1. El usuario ingresa a la app de Yape y selecciona la pestaña Restaurantes 2. El usuario selecciona una ubicación manualmente o automáticamente por medio del uso del GPS. 3. La aplicación muestra los restaurantes que están dentro de un radio de 2km.

System	YapePoints
Actor	Propietario de restaurante
Use Case	UC-04 Agregar restaurante
Preconditions	El usuario está logueado a YapeMaster
MMS	<ol style="list-style-type: none"> 1. El usuario selecciona la opción Agregar nueva sede. 2. El usuario llena el formulario con los datos. 3. El usuario confirma la operación al ingresar su contraseña.

System	YapePoints
Actor	Usuario
Use Case	UC-05 Usuario filtra los restaurante por disponibles
Preconditions	El usuario está logueado a Yape
MMS	<ol style="list-style-type: none"> 1. El usuario ingresa a la pestaña Mis cupones. 2. YapePoints muestra las opciones en tendencia. 3. El usuario ingresa palabras clave en el buscador. 4. YapePoints muestra opciones filtradas según las palabras clave.

System	YapeMaster
Actor	Propietario de restaurante
Use Case	UC-06 Login a YapeMaster
Preconditions	UC-04
MMS	<ol style="list-style-type: none"> 1. El usuario ingresa a YapeMaster. 2. El usuario ingresa a la pestaña Mis Restaurantes. 3. El usuario selecciona el restaurante.

System	YapeMaster
Actor	Usuario con acceso a la cuenta del restaurante
Use Case	UC-07 Agregar una nueva promoción
Preconditions	UC-06
MMS	<ol style="list-style-type: none"> 1. El usuario con acceso selecciona la opción Agregar una Nueva Promoción. 2. El usuario carga una imagen (personalizada o predeterminada). 3. El usuario le pone nombre a la promoción. 4. El usuario describe la promoción. 5. El usuario llena un formulario con parámetros de la promoción (stock, fecha, requisitos).

System	YapeMaster
Actor	Usuario con acceso a la cuenta del restaurante
Use Case	UC-08 Administrar promociones
Preconditions	UC-07
MMS	<ol style="list-style-type: none"> 1. El usuario selecciona promociones actuales. 2. El usuario selecciona modificar promoción. 3. El usuario edita los parámetros que quiere modificar.

System	YapeMaster
Actor	Propietario de restaurante
Use Case	UC-09 Agregar trabajadores
Preconditions	UC-06
MMS	<ol style="list-style-type: none"> 1. El propietario selecciona Agregar nuevo trabajador. 2. El propietario selecciona un número entre los contactos del celular. 3. El sistema comprueba la existencia del usuario. 4. El propietario confirma concederle permisos de usuario de YapeMaster al trabajador. 5. El propietario puede editar parámetros de información del trabajador (nombre, cargo).

System	YapeMaster
Actor	Propietario del restaurante
Use Case	UC-10 Eliminar restaurante
Preconditions	UC-06
MMS	<ol style="list-style-type: none"> 4. El propietario selecciona pestaña Mis restaurantes. 5. El propietario selecciona el restaurante que desea eliminar. 6. El propietario selecciona la opción Eliminar restaurante. 7. El sistema le pide al usuario confirmar la acción. 8. El sistema elimina el restaurante.

System	YapeMaster
Actor	Usuario
Use Case	UC-11 Selección de promociones por locación
Preconditions	El usuario está logueado.
MMS	<ol style="list-style-type: none"> 1. El usuario entra a la pestaña Cupones. 2. El usuario selecciona Filtros. 3. El usuario selecciona el filtro por distrito.

Apuntes sobre los Use Cases

- UC-04: Formulario contiene:
 - Nombre del restaurante
 - RUC
 - Razón social
 - Dirección (un link a google maps - junto a un popup que indique como hacerlo o mande a un link).
 - Distrito, Provincia y Departamento
 - Horario de atención (formato por definir)
- UC-04: Validación del registro:
 - Analizar si la verificación debe ser manual.
- UC-08: Para cada promoción debería poder salir 1 menú desplegable con estas opciones
 - Mostrar estadísticas de la promoción [Por definir que mostrar aqui]
 - Modificar promoción [Si es una promoción actual]
 - Detener promoción [Si es una promoción actual]
- UC-09: Considerar eliminar trabajadores.
- UC-10: No debe estar muy visible.
- *YapePoints*: ¿Cómo tener una promoción sin internet? *Screenshot*.
- Out of Scope: El sistema le muestra al usuario promociones que podrían ser de su interés.
- *YapeMaster en general*:
 - Definir un límite de promociones.
 - Cada promoción tiene asignado un código identificador y un QR.

V. Non Functional Requirements

- La aplicación debe mantener su peso. Por este motivo, se debe diseñar una aplicación adicional (YapeMaster) para que los restaurantes gestionen sus promociones.

VI. Glossary

Ticket	Precio de un almuerzo.
Menú	Restaurante ubicado cerca a centros de estudio o de trabajo que ofrece un ticket promedio menor a S/.20.
Comensal	Persona que hace uso de los servicios que ofrece un menú.
Yape	Sistema creado por el BCP que permite pagar servicios con el celular.

YapePoints	Extensión de Yape que permite a un comensal administrar cupones de menús y restaurantes preferidos.
YapeMaster	Plataforma de Yape que permite a un restaurante administrar información de los cupones que ofrece y de pagos hechos con Yape en su negocio.

VII. Product Survey

Formulario	https://docs.google.com/forms/d/1MHhMRx0321j3WHT6mWdnwfQuS5srx-fuSwmOeVT0GcU/edit
Nivel de confianza	90%
Margen de error	10%
Población	2500 comensales
Muestra	77 comensales
Resultados	<ul style="list-style-type: none"> • La mitad de los encuestados tiene entre 18 y 21 años. • Aproximadamente $\frac{2}{3}$ de la población come en menús con mucha frecuencia. • El 87% de la muestra tiene la intención de utilizar su celular para pagar en estos locales • El 70% de los encuestados cuenta con una cuenta bancaria. • El 98% está interesado en obtener descuentos en estos locales por usar su celular.
Observaciones	<ul style="list-style-type: none"> • La muestra está conformada únicamente por estudiantes de UTEC. • Se encuestó a pocos restaurantes. • El 30% de las personas de la encuesta no contaba con una cuenta bancaria. • El 35,5% de las personas encuestadas no era mayor de edad.

VIII. Interviews

- Las necesidades de los menús varían mucho de acuerdo a las características del dueño.

- Los dueños más jóvenes conocen a fondo Yape y les interesa aprovecharlo para captar clientes mediante promociones y de paso obtener mejores reportes.
- Los dueños más viejos son ignorantes técnicos y no desean tener ningún contacto con el servicio.
- Los dueños intermedios desconocen el sistema, pero quisieran poder acceder a todas las promesas que hace la propuesta.
- Los dueños que ya admiten tarjeta no tienen problemas al cobrar, mientras que los que solo aceptan efectivo siempre sufren a la hora de dar vuelto.
- Mientras menos organizado está el local, es más probable que necesite la propuesta, pero a la vez es más probable que la rechace.

Preguntas:

1. *¿Cómo se llama?*
2. *¿Cómo le va con el local? ¿Conoce a muchos universitarios de UTEC?*
3. *¿Cómo paga la gente en su local?*
4. *¿Cuáles son los principales problemas al pagar?*
5. *¿Cuáles son los principales problemas al hacer las cuentas al final del día?*
6. *(problemas con el vuelto, monedas/billetes falsos).*
7. *¿Le gustaría contar con un sistema de pagos sin costo para que la gente pueda pagar desde su celular?*
8. *¿En qué más debería ayudarlo/a este sistema?*
9. *¿Tiene Yape instalado?*
10. *¿Le gustaría compartir promociones a los estudiantes y consumidores mediante este sistema?*

VIII. Features

funciones: lowerCamelCase

variables: snake_case

clases: UpperCamelCase

código language: spanglish

- **Consideraciones**
 - Al hacer login, el ID del usuario se almacena en una sesión.
 - Minimalista: Las opciones de configuración no se incluirán en el menú hamburguesa.
- **Aplicación:** Yape Master
 - **Clase:** Util
 - **Función:** +encryptPassword (String) -> String
 - Recibe un string y retorna el string encriptado.
 - **Función:** +matchPassword (String, String) -> Bool
 - Recibe dos strings y retorna un bool que indique si la encriptación del primer string es el segundo string.

- **Función:** +getSession () -> Mongo.ObjectId
 - Retorna el ID de la sesión actual y del restaurante actual.
- **Clase:** Login
 - **Función:** +render ()
 - Muestra una vista para realizar el login.
 - **Función:** +match (String)
 - Si el string es la contraseña de la actual sesión, redirecciona a la vista de inicio; en caso contrario, muestra un mensaje de error.
- **Clase:** Register
 - **Función:** +render ()
 - Muestra una vista para registrarse.
 - **Función:** +doRegister (JSON)
 - Recibe y procesa los datos obtenidos en el formulario de registro. Si el registro es exitoso, redirecciona al Login; de lo contrario, muestra un mensaje de error.
- **Clase:** Inicio
 - **Función:** +render ()
 - Muestra una vista con la opción de agregar un nuevo restaurante y muestra los actuales restaurantes que administra la sesión actual.
 - **Función:** +addBusiness (JSON)
 - Agrega un nuevo restaurante. Retorna un mensaje de estado.
 - **Función:** -getBusiness ()
 - Retorna un array de restaurantes que administra la sesión actual.
- **Clase:** Information
 - **Función:** +render ()
 - Muestra una vista que contiene información del restaurante de la sesión actual.
 - **Función:** +updateInformation (JSON)
 - Actualiza la información de un restaurante.
- **Clase:** Workers
 - **Función:** +render ()
 - Muestra una vista con la opción de agregar un nuevo trabajador y lista los trabajadores actuales.
 - **Función:** +addWorker (JSON)
 - Agrega un nuevo trabajador. Retorna un mensaje de estado.
 - **Función:** +updateWorker (JSON)
 - Actualiza la información de un trabajador. Retorna un mensaje de estado.
 - **Función:** +deleteWorker (JSON)
 - Elimina un trabajador. Retorna un mensaje de estado.
- **Clase:** Statistics
 - **Función:** +render ()

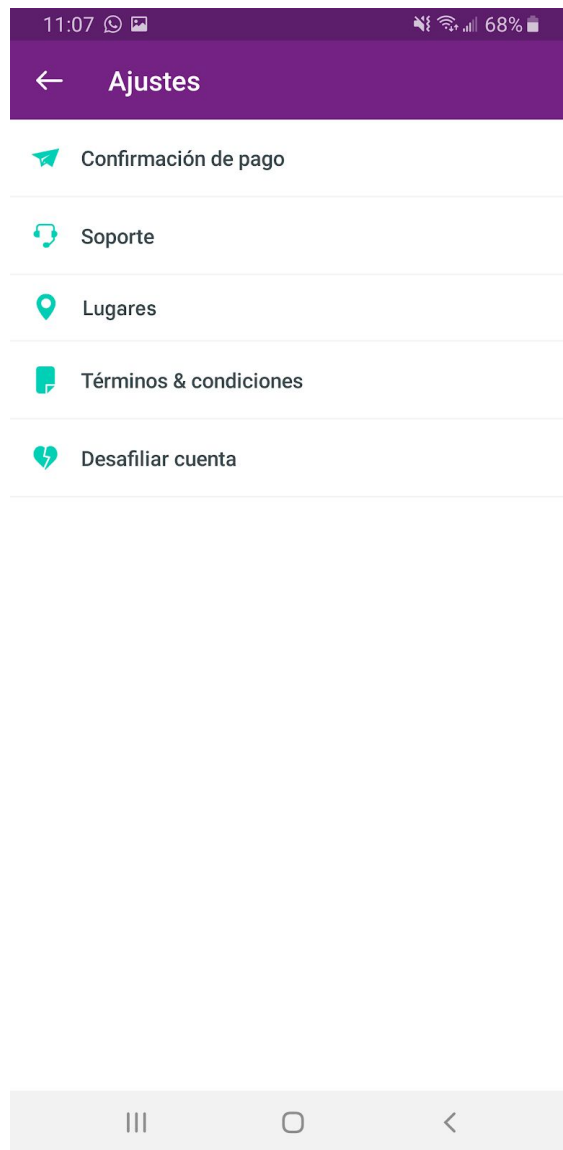
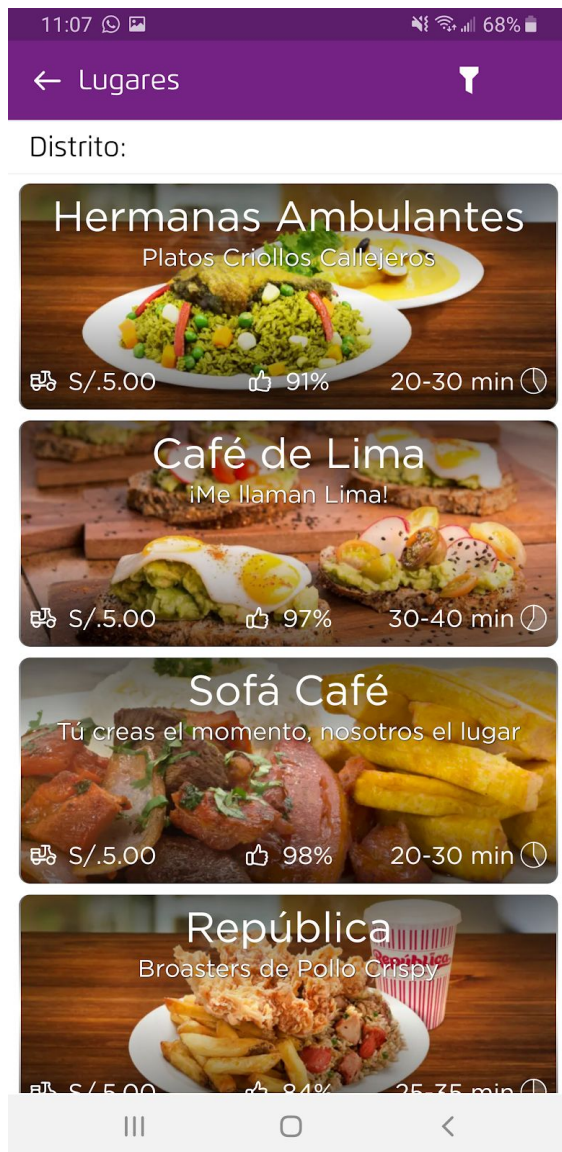
- Muestra una vista con la cantidad de ventas del día, semana y mes.
 - **Función:** -getStatistics ()
 - Obtiene las estadísticas de la sesión actual.
- **Clase:** Menu
 - **Función:** +render ()
 - Muestra una vista con el inventario actual de menús, una opción para agregar un nuevo menú y una opción para seleccionar un día de la semana y poder visualizar y/o armar el menú del día.
 - **Función:** +addMenu (JSON)
 - Agrega un nuevo menú. Retorna un mensaje de estado.
 - **Función:** +deleteMenu (JSON)
 - Elimina un menú. Retorna un mensaje de estado.
 - **Función:** +dragMenu (JSON)
 - Asocia un menú del inventario a un día de la semana.
- **Clase:** Coupon
 - **Función:** +render ()
 - Muestra una vista con los cupones asociados al restaurante actual.
 - **Función:** +addCoupon (JSON)
 - Agrega un nuevo cupón. Retorna un mensaje de estado.
 - **Función:** +deleteCoupon (JSON)
 - Elimina un cupón. Retorna un mensaje de estado.
 - **Función:** +updateCoupon (JSON)
 - Actualiza la información de un cupón. Retorna un mensaje de estado.
- Yape Points
 - **Clase:** Util, Login, Register, ChangePassword [Reutilizar de Yape Master]
 - **Clase:** Inicio
 - **Función:** +render ()
 - Muestra un historial de las transacciones actuales.
 - **Función:** -getTransactions ()
 - Retorna un array de transacciones de la sesión actual.
 - **Función:** +showCoupons (String)
 - Redirecciona a la vista de Coupon.
 - **Clase:** Coupon
 - **Función:** +render (String)
 - Muestra los cupones disponibles en todo Lima.
 - **Función:** +getCoupons (String)
 - Recibe un string que identifica a un distrito de Lima. Muestra los cupones disponibles en ese distrito
 - **Clase:** Notification
 - **Función:** +addNotification ()
 - Agrega una nueva notificación a la sesión actual

IX. Gantt

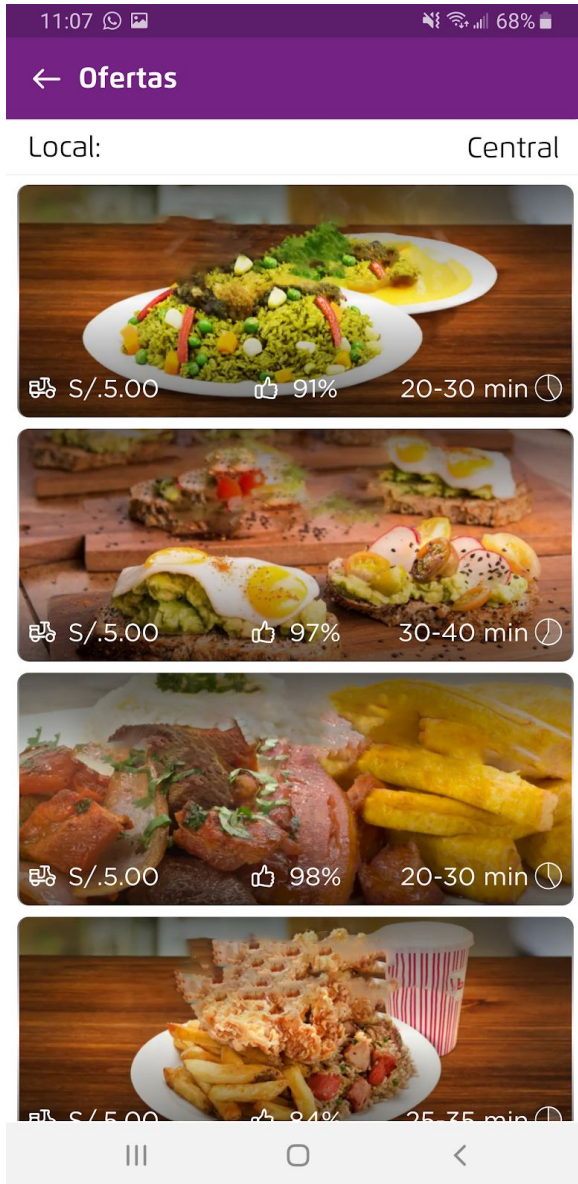
Iteración	Semana 7	Semana 8	Semana 9: Prototipo 2	Semana 10	Semana 11	Semana 12: Prototipo 3	Semana 13: BUFFER	Semana 14: BUFFER	Semana 15: ENTREGA
SPRINT 1									
YapeMaster: Login	Leonidas, Sebastián, Francisco								
YapeMaster: Register	Jhony, César, Indhira								
YapeMaster: ChangePassword		Leonidas, Sebastián, Francisco							
YapeMaster: Util		Jhony, César, Indhira							
SPRINT 2									
YapeMaster: Inicio			Sebastián						
YapeMaster: Workers			Indhira	Indhira					
YapeMaster: Statistics			César, Francisco	César, Francisco	César, Francisco				
YapeMaster: Menu			Johny	Johny					
YapeMaster: Coupons			Leonidas	Leonidas, Sebastián	Leonidas, Sebastián				
SPRINT 1									
YapePoints: Login	Leonidas, Sebastián, Francisco								
YapePoints:	Jhony,								

Register	César, Indhira								
YapePoints: ChangePass word		Leonidas, Sebastián, Francisco							
YapePoints: Util		Jhony, César, Indhira							
SPRINT 3									
YapePoints: Inicio						Sebastián, Francisco			
YapePoints: Coupons						Leonidas, Jhony			
YapePoints: Notifications						César, Indhira, Francisco			

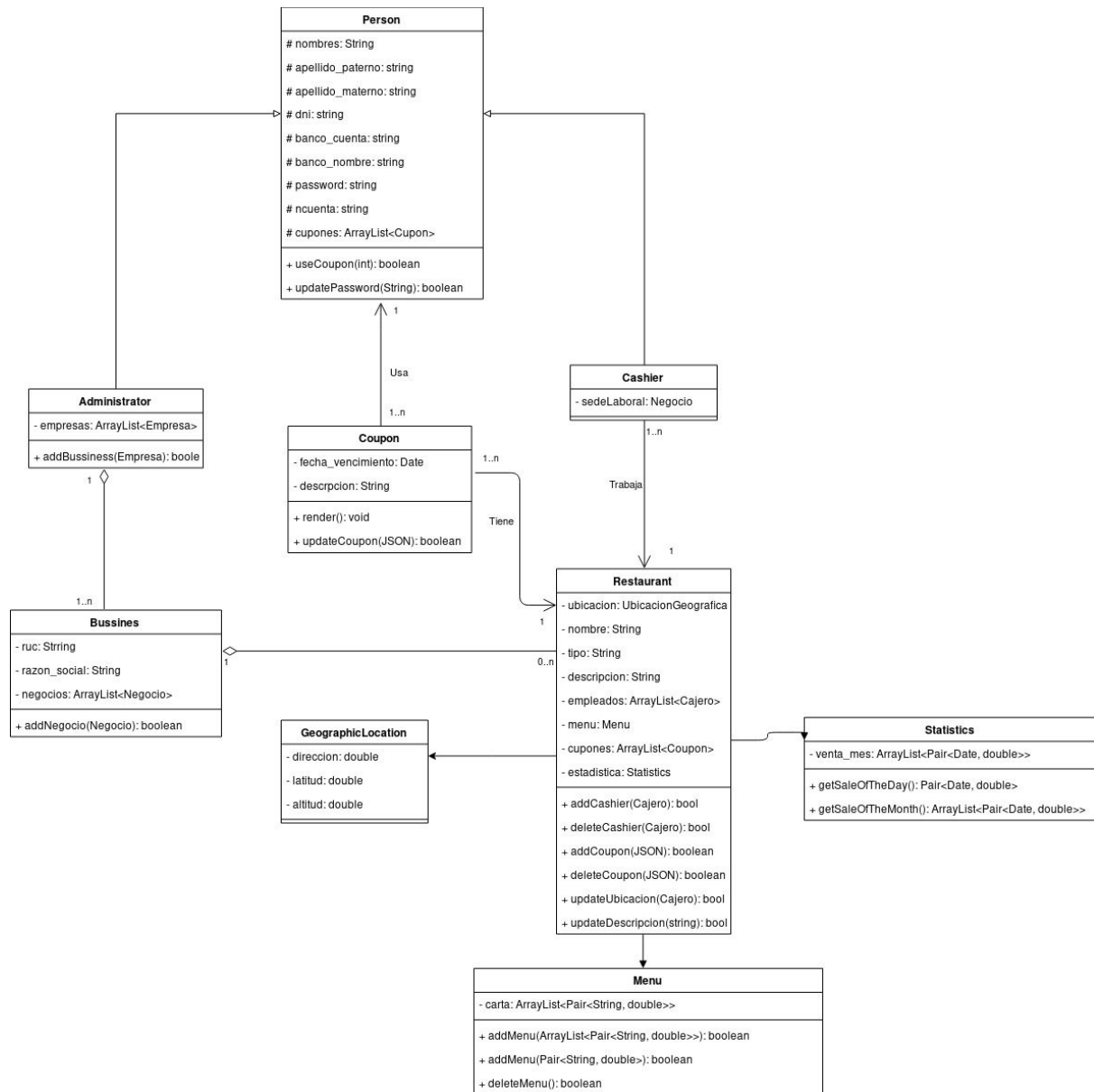
X. Mockups

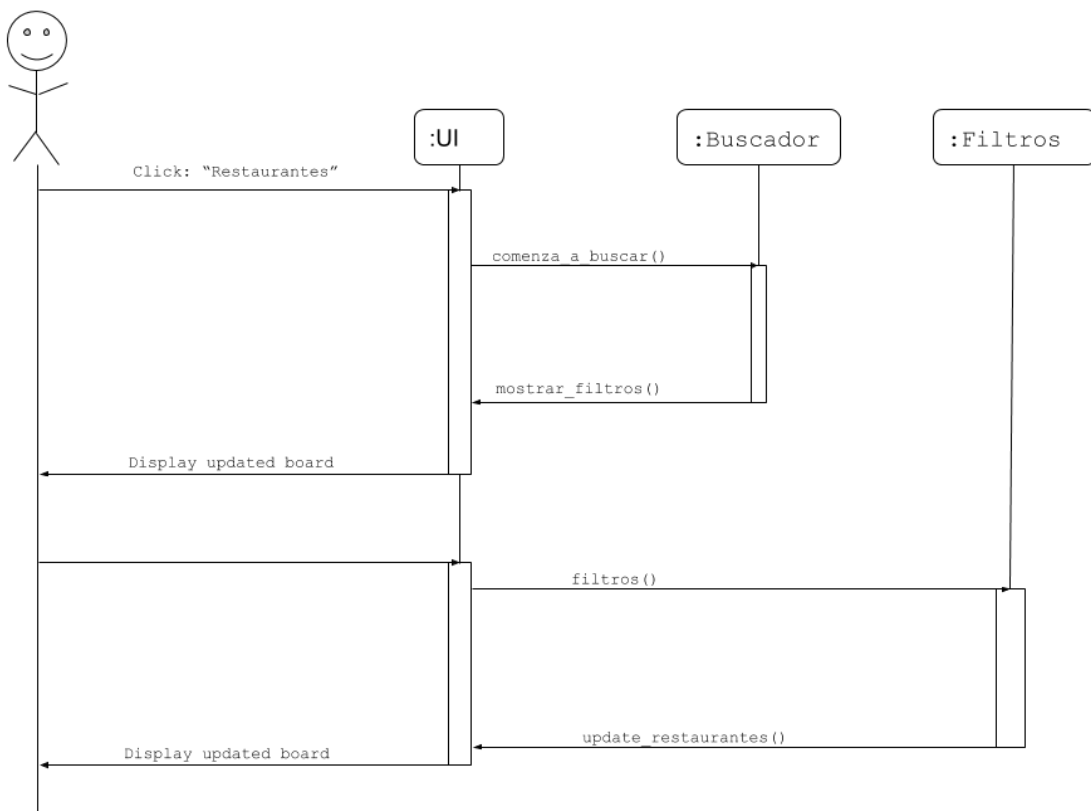
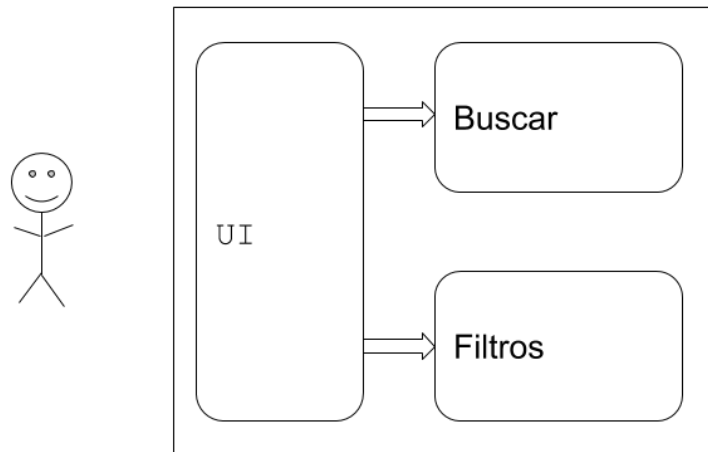






XI. UML





XII. Files

El proyecto se dividirá en el siguiente sistema de archivos:

- YapeMaster: Es el frontend de YapeMaster
 - pages: Vistas para las rutas
 - layouts: Estructuras para las vistas
 - assets: Imágenes, audios, videos
 - middlewares
 - plugins
- Yape-backend:
 - models: Modelos para la base de datos
 - controllers: Middleware para las rutas
 - routes: Rutas del api
 - services: Estará el sistema de notificaciones, email, estadística, etc.
- Yape-points: Frontend de Yape. Tendrá la misma estructura que el frontend de YapeMaster.

XIII. Features and Versions

- **Aplicación: Yape Master**
 - **Leonidas, Sebastián, Francisco: Login Feature**
 - v1.1: Login sencillo, comprobación sin base de datos.
 - v1.2: Mensaje de error y redirección a Inicio.
 - **Jhony, César, Indhira: Register Feature**
 - v1.1: Formulario sencillo.
 - v1.2: Interacción con Base de Datos y redirección a Login.
 - **Jhony, César, Indhira: Util Feature**
 - v1.1: Encriptación de la contraseña.
 - v1.2: Mensaje de aprobación.
 - **Sebastián: Inicio**
 - v1.1 Funcionalidad para agregar un nuevo restaurante.
 - v1.2 Visualización vertical de la lista de restaurantes creados.
 - v1.3 Visualización de información por restaurante.
 - v1.4 Actualizar información de un restaurante.
 - **Indhira: Workers**
 - v1.1 Puedes agregar, eliminar y visualizar a los trabajadores.
 - v1.2 Puedes editar la información de los trabajadores.
 - **César, Francisco: Statistics**
 - v1.1 Visualizar ventas del día, semana y mes.
 - v1.2 Visualizar gráficas de desempeño.
 - **Jhony: Menu**
 - v1.1 Layout con los menús actuales.
 - v1.2 Permite editar, eliminar y agregar menús.
 - v1.3 Permite asociar menús con días de la semana.

- **Leonidas, Sebastián: Coupon**
 - **v1.1** Vista de la lista de cupones.
 - **v1.2** Agregar, editar y eliminar cupones.
- **Extensión: Yape Points**
 - **Leonidas, Sebastián, Francisco: Login Feature**
 - **v1.1:** Login sencillo, comprobación sin base de datos.
 - **v1.2:** Mensaje de error y redirección a Inicio.
 - **Jhony, César, Indhira: Register Feature**
 - **v1.1:** Formulario sencillo.
 - **v1.2:** Interacción con Base de Datos y redirección a Login.
 - **Jhony, César, Indhira: Útil Feature**
 - **v1.1:** Encriptación de la contraseña.
 - **v1.2:** Mensaje de aprobación.
 - **Sebastián, Francisco: Inicio**
 - **v1.1:** Mostrar las transacciones actuales del cliente
 - **Leonidas, Jhony: Coupon**
 - **v1.1:** Mostrar los cupones en Lima
 - **v1.2:** Filtrar los cupones por distrito
 - **César, Indhira: Notifications**
 - **v1.1:** Agregar las notificaciones de los pagos a los *workers*