



Ciclo 2019-1
Ingeniería de Software I

Tercer Avance

Alumnos:

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

Profesores:

- Jesús Bellido
- Yamilet Serrano

Fecha de entrega: 20/05/2019
Barranco, 2019

I. Target User Profile

Este proyecto contempla dos tipos de usuarios: yaperos y restaurantes. El yapero trabaja o estudia y con frecuencia almuerza fuera de casa en un restaurante con menú de precio asequible. El restaurante en el que se enfoca esta propuesta es aquel que se encuentra cerca a un centro laboral o de estudio.

Los yaperos son jóvenes de 19 a 34 años de los sectores C, B y A, principalmente (Yape, BCP). Dentro de este público, tienen dos perfiles más específicos:

- Universitarios con potencial: Jóvenes de 19 a 24 años que prefieren pagar con su celular a llevar dinero en efectivo. Tienen ingresos de 1000 soles mensuales, producto de los primeros empleos.
- Jóvenes con potencial: Trabajadores de 29 a 34 años con ingreso mensual alto (más de 5000 soles al mes) que llevan un estilo de vida activo (sus tickets en Yape superan los 50 soles diarios).

¿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 este, 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 diario del restaurante 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: *YapeCupones*.

Según el CPI, en los sectores A, B y C, los jóvenes de ambas categorías suman 1,4 millones de personas. El 80% come fuera de casa, por lo que hay un mercado potencial de 1,1 millones de jóvenes.

II. Propuesta de Valor

Para los **yaperos**, encontrar los mejores descuentos en un solo lugar. Esto, mediante la posibilidad de pagar con celular (*Yape*) y obtener beneficios en restaurantes afiliados(*YapeCupones*).

Para los **restaurantes**, promocionar su negocio ante los *Yaperos*. Primero, un sistema de pagos por celular (*Yape*) y administración de cupones (mediante *YapeCupones*). De esta manera, el dueño de un restaurante puede entrar al ecosistema de pagos *Yape* y promocionar su negocio mediante cupones.

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	YapeCupones
Actor	Dueño de negocio
Use Case	UC-01 - Crear cuenta
MMS	<ol style="list-style-type: none"> 1. El usuario entra a la web de YapeCupones. 2. El usuario selecciona Registrarse 3. El sistema redirecciona a un formulario. 4. El usuario completa el formulario. 5. EL usuario selecciona crear cuenta. 6. El sistema crea la cuenta en YapeCupones y redirecciona a la vista home. <p>Use case ends.</p>
Extentions	<p>5a. El sistema detecta que hay campos sin completar</p> <ol style="list-style-type: none"> 5a1. Solicita que complete todos los campos 5a2. El usuario entra otros datos <p>Los pasos 5a1, 5a2 son repetidos hasta que se ingrese todos los datos. Use case se reanuda desde el paso 6</p> <p>6a. El sistema detecta que ya hay un usuario registrado con la información ingresada</p> <ol style="list-style-type: none"> 6a1. Redirecciona al formulario <p>El paso 6a1 se repite hasta que ingrese una información diferente</p> <p>*a En cualquier momento el usuario puede volver a la página de inicio</p> <ol style="list-style-type: none"> *a1. Se redirecciona a la página principal home. <p>User case ends.</p>

System	YapeCupones
Actor	Usuario con acceso a la cuenta de un restaurante
Use Case	UC-02 - Agregar una nueva promoción
Preconditions	UC-01, Estar logueado
MMS	<ol style="list-style-type: none"> 1. El usuario con acceso selecciona la opción Agregar Cupón. 2. El usuario carga una imagen. 3. El usuario le pone nombre al cupón. 4. El usuario llena un formulario con parámetros de la promoción. 5. El usuario hace click en publicar. 6. El sistema redirige al usuario a su página principal, mostrando así su nuevo cupón creado. <p>User case ends.</p>
Extentions	<p>5a. El sistema detecta que hay campos sin completar</p> <p>5a1. Solicita que complete todos los campos</p> <p>5a2. El usuario entra otros datos</p> <p>Los pasos 5a1, 5a2 son repetidos hasta que se ingrese todos los datos.</p> <p>Use case se reanuda desde el paso 6.</p> <p>*a En cualquier momento el usuario puede volver a la página de inicio</p> <p>*a1. Se redirecciona a la página principal home.</p> <p>User case ends.</p>

System	YapeCupones
Actor	Usuario con acceso a la cuenta de un restaurante
Use Case	UC-03 Editar cupón
Preconditions	UC-02, Haber agregado el cupón que desea editar
MMS	<ol style="list-style-type: none"> 1. El usuario selecciona la vista de cupones. 2. El usuario selecciona la opción Editar, que está a la derecha del cupón que quiere cambiar. 3. El usuario edita los campos que desee. 4. El usuario hace click en guardar los cambios. <p>User case ends.</p>
Extentions	<p>4a. El sistema detecta que hay campos sin completar</p> <p>4a1. Solicita que complete todos los campos</p> <p>4a2. El usuario entra otros datos</p>

	<p>Los pasos 4a1, 4a2 son repetidos hasta que se ingrese todos los datos. Use case se reanuda desde el paso 6</p> <p>*a En cualquier momento el usuario puede volver a la página de inicio</p> <p>*a1. Se redirecciona a la página principal home. User case ends.</p>
--	--

System	YapeCupones
Actor	Usuario con acceso a la cuenta de un restaurante
Use Case	UC-04 Eliminar promoción
Preconditions	UC-02
MMS	<ol style="list-style-type: none"> 1. El propietario selecciona el cupón que desea eliminar. 2. El propietario selecciona la opción Eliminar Cupón 3. El sistema elimina el cupón. <p>User case ends.</p>

System	YapeCupones
Actor	Dueño de restaurante
Use Case	UC-05 Recuperar contraseña
MMS	<ol style="list-style-type: none"> 1. El usuario selecciona Recuperar. 2. El usuario ingresa su email y da click en enviar correo. 3. El sistema le pide la clave de recuperación. 4. El usuario recibe un correo con una clave de recuperación. 5. El usuario ingresa esta clave. 6. El sistema lo redirecciona a una pestaña que le pide la nueva contraseña y su autenticación. 7. El usuario envía esta nueva contraseña. 8. La contraseña es cambiada. <p>User case ends.</p>
Extentions	<p>2a. El sistema detecta que no hay un usuario registrado con ese email Use case ends.</p> <p>7a. El sistema detecta que hay campos sin completar</p> <p>7a1. Solicita que complete todos los campos</p> <p>7a2. El usuario entra otros datos Los pasos 5a1, 5a2 son repetidos hasta que se ingrese todos los datos. Use case se reanuda desde el paso 6.</p> <p>7b. El sistema la contraseña y su verificación no coinciden.</p>

	<p>7b1. Solicita que complete de nuevo los campos</p> <p>7b2. El usuario entra otros datos</p> <p>Los pasos 7b1, 7b2 son repetidos hasta que se ingrese todos los datos.</p> <p>Use case se reanuda desde el paso 8.</p> <p>*a En cualquier momento el usuario puede volver a la página de inicio</p> <p>*a1. Se redirecciona a la página principal home.</p> <p>User case ends.</p>
--	--

System	YapeCupones
Actor	Usuario de Yape que busca cupones
Use Case	UC-06 Acceder a pestaña Cupones
Preconditions	Estar logueado a Yape
MMS	<p>9. Hacer click en el menú de opciones de Yape.</p> <p>10. Hacer click en lugares.</p> <p>User case ends.</p>

System	YapeCupones
Actor	Usuario de Yape
Use Case	UC-07 Selecciona un cupón
Preconditions	UC-06, el usuario está logueado.
MMS	<p>7. El usuario hace click a la pestaña YapeCupones.</p> <p>8. El usuario busca un cupón en la vista principal con los filtros que necesite (de distrito).</p> <p>9. El usuario ve un cupón que le interesa</p> <p>10. El usuario selecciona ver dirección</p> <p>11. El usuario es redireccionado google maps y visualiza la ubicación del establecimiento</p> <p>User case ends.</p>

System	YapeCupones
Actor	Usuario de Yape
Use Case	UC-08 Usuario quiere filtrar cupones.
Preconditions	UC-06, el usuario está logueado.

MMS	<ol style="list-style-type: none"> 1. El usuario ingresa a la app de Yape y selecciona la pestaña YapeCupones. 2. El usuario selecciona un distrito. 3. La aplicación muestra los cupones de los restaurantes que se encuentran en esos distritos. <p>User case ends.</p>
------------	---

V. Non Functional Requirements

- Usabilidad: la plataforma debe ser compatible con todos los dispositivos móviles que soporten previamente el uso de Yape.
- Manejabilidad: la plataforma debe tener una interfaz sencilla, similar a la de Yape, para no confundir a los usuarios.
- Performance: la plataforma debe ser capaz de brindar la información a los usuarios en un tiempo efectivo (debe ser tan o más rápida que Yape).
- Seguridad: la plataforma debe garantizar la integridad de la información de los usuarios.
- Escalabilidad: la plataforma debe soportar una gran cantidad de cupones a la vez, sin comprometer la vista que tienen los usuarios.
- Interoperabilidad: la plataforma debe funcionar como un complemento de Yape, no como un servicio independiente.

VI. Glossary

Ticket	Precio de un almuerzo o menú.
Restaurante	Establecimiento en el que se preparan y sirven comidas. Nos enfocamos en restaurantes cercanos al centro laboral y de estudios con un ticket promedio menor a S/.20 por menú o almuerzo.
Yape	Sistema creado por el BCP que permite pagar servicios con el celular.
YapeCupones	Plataforma web (disponible en móvil) sincronizada con Yape que permite a un Yapero encontrar los mejores descuentos.

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 restaurantes 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

Muestra: 7 entrevistas a restaurantes pequeños cercanos a la Universidad de Ingeniería y Tecnología (Barranco).

Hallazgos:

- Las necesidades de los restaurantes 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?*

IX. 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:** YapeCupones
 - **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 ()

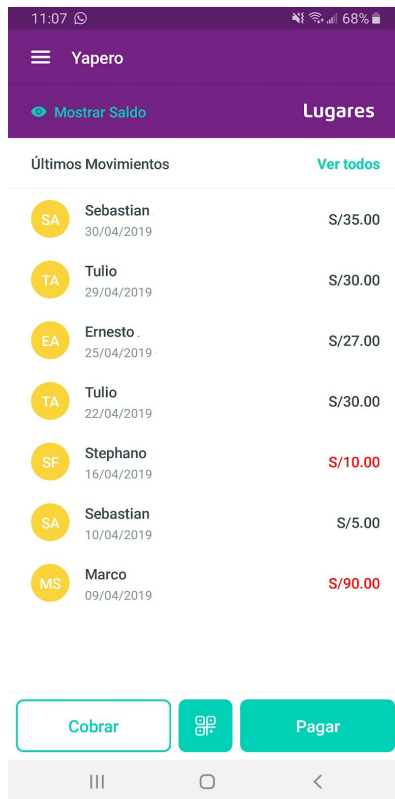
- Retorna un array de restaurantes que administra la sesión actual.
- **Clase:** Information
 - **Función:** +render ()
 - Muestra una vista con la pagina principal que contiene información del restaurante de la sesión actual.
 - **Función:** +updateInformation (JSON)
 - Actualiza la información de un restaurante.
- **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
 - **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.

X. Gantt

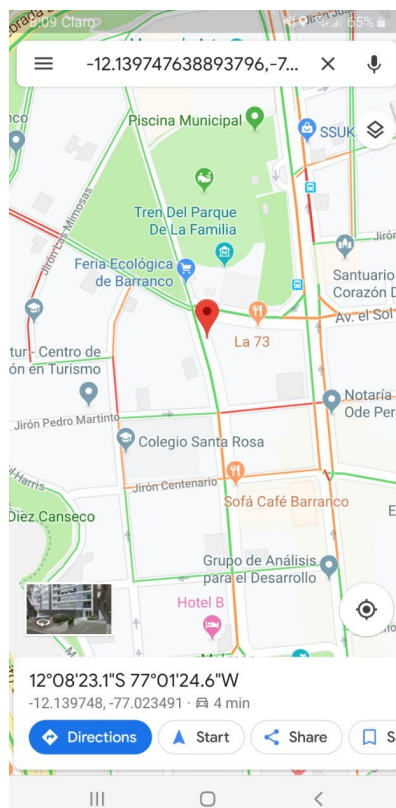
Iteración	Semana 7	Semana 8	Semana 9: Prototipo 2	Semana 10	Semana 11	Semana 12: Prototipo 3	Semana 13: BUFFER	Semana 14: BUFFER	Semana 15: ENTREGA
SPRINT 1									
YapeMaster: Login	Leonidas Sebastián Francisco								
YapeMaster: Register	Jhony, César, Indhira								
YapeMaster: ChangePassword		Leonidas Sebastián Francisco							
YapeMaster: Util		Jhony, César, Indhira							
SPRINT 2									
YapeMaster: Inicio			Sebastián, Leonidas						
YapeMaster: Coupons			Leonidas Indhira	Leonidas Sebastián indhira	Leonidas Sebastián indhira				
YapeMaster: Opciones para Coupons			Cesar, Jhony, Francisco	Cesar, Jhony Francisco	Cesar, Jhony Francisco				
SPRINT 3									
YapePoints: Login	Leonidas Sebastián Francisco								
YapePoints: Register	Jhony, César, Indhira								
YapePoints: ChangePassword		Leonidas Sebastián Francisco							
YapePoints: Util		Jhony, César, Indhira							
SPRINT 4									
YapePoints: Inicio						Sebastián Francisco			
YapePoints: Coupons						Leonidas Johny			
YapePoints: Notifications						César Indhira			

XI. Vistas

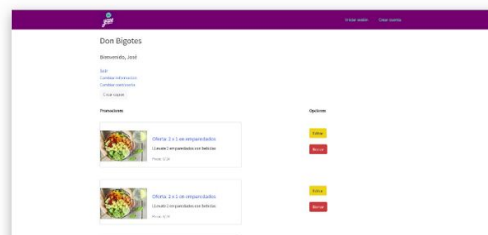
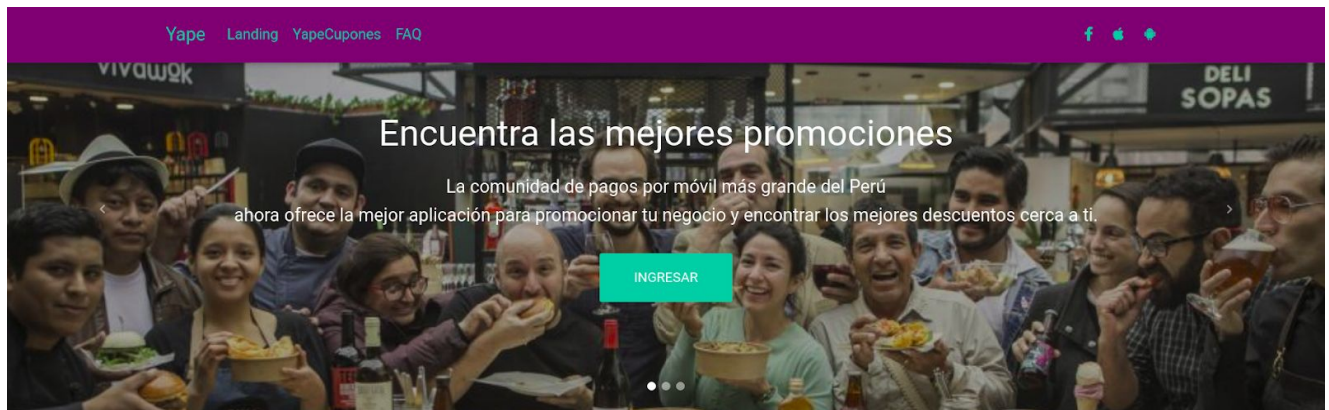
Vista del acceso directo desde Yape:



Vista de usuario Yapero:



Vista del restaurante

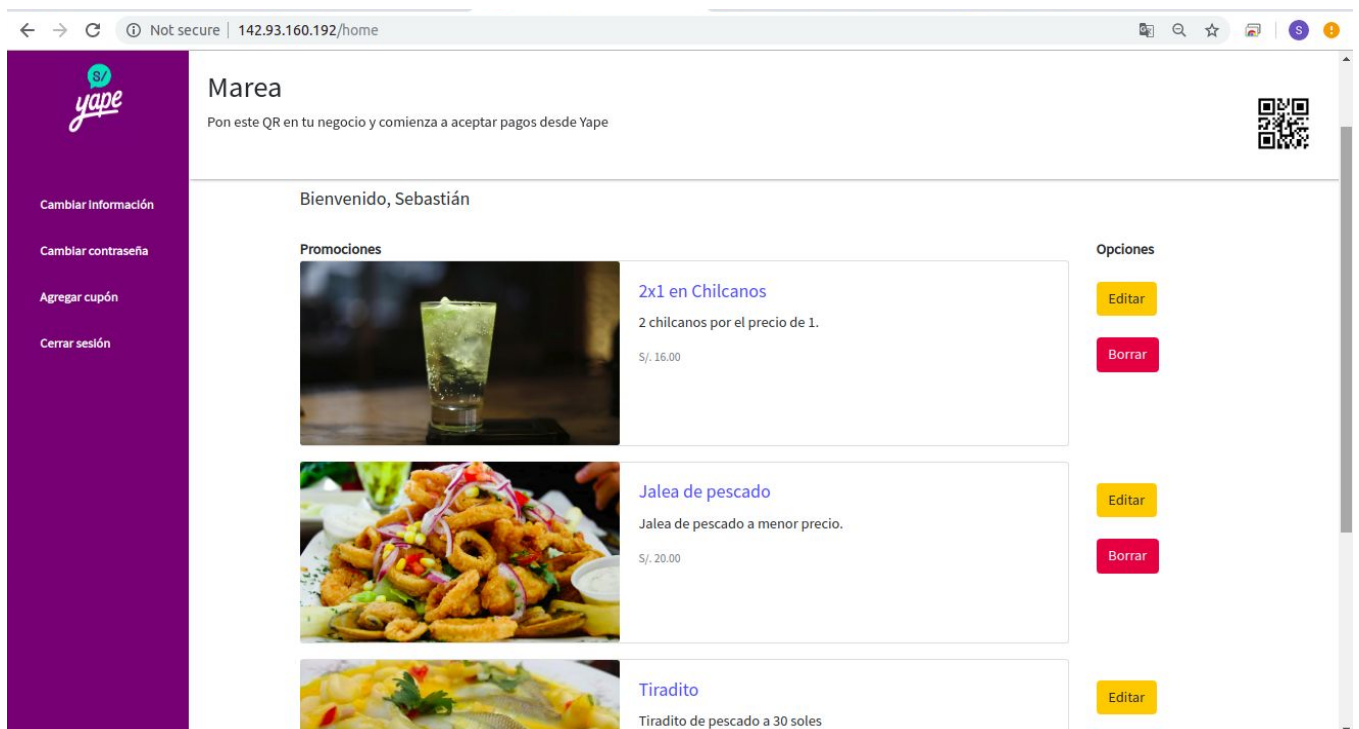


Propuesta de valor

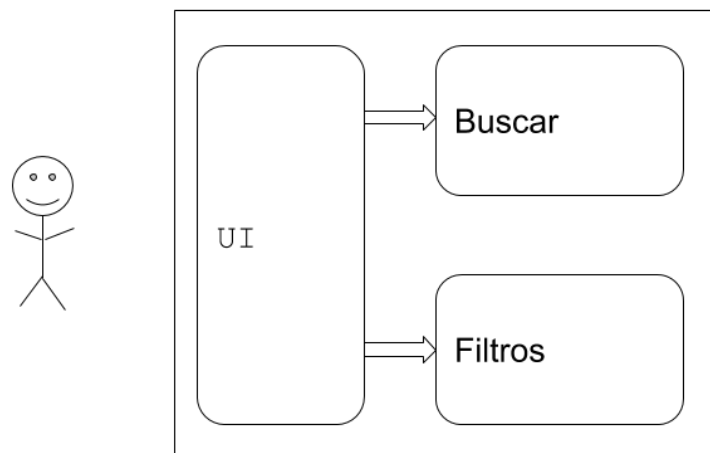
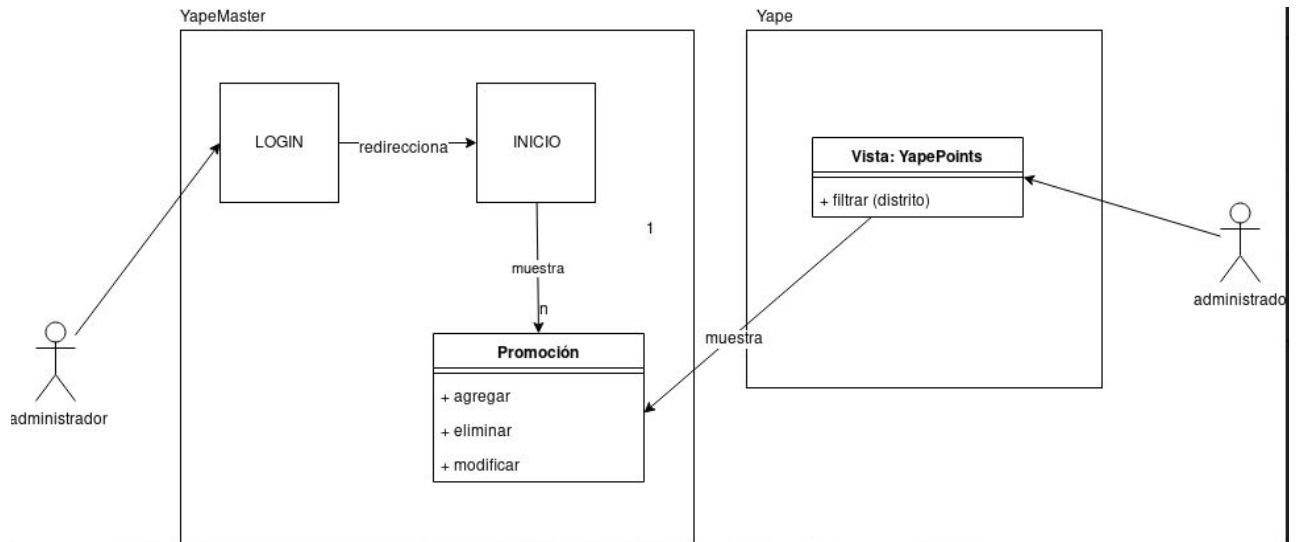
YapeCupones es la mejor manera de promocionar tus negocios ante un **millón de yaperos**.

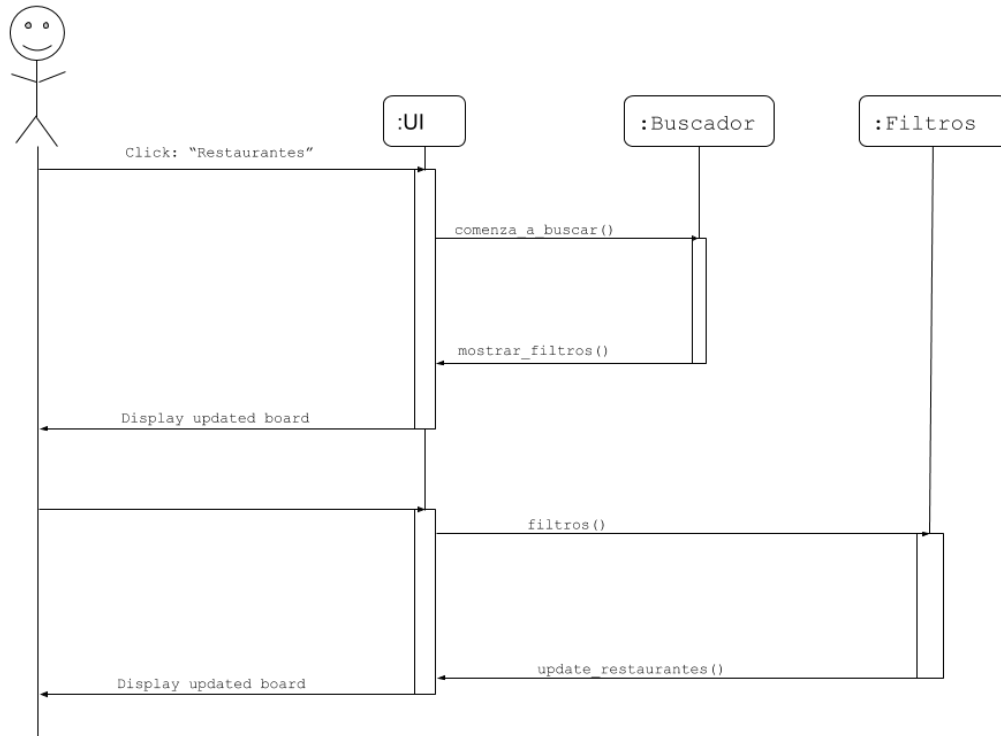
Más detalles sobre nosotros abajo.

Más de **1M+** de usuarios, agrega **100+** ofertas para tu negocio. Acceso y uso totalmente gratuito.



XII. UML





XIII. Files

El proyecto se dividirá en el siguiente sistema de archivos:

- YapeCupones (restaurante): Es la vista de YapeCupones al rol de administrador de restaurante.
 - pages: Vistas para las rutas
 - layouts: Estructuras para las vistas
 - assets: Imagenes, audios, videos
 - middlewares
 - plugins
- Yape-backend:
 - models: Modelos para la base de datos
 - controllolers: Middleware para las rutas
 - routes: Rutas del api
 - services: Estará el sistema de notificaciones, email, estadística, etc.
- Yape-points: Frontend de Yape. Vista para los usuarios de Yape.

XIV. Features and Versions

- **Aplicación: Yape Cupones**
 - **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, Leonidas: 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.
 - **Leonidas, Sebastián, Indhira: Coupons**
 - **v1.1** Vista de la lista de cupones.
 - **v1.2** Agregar.
 - **Cesar, Jhony, Francisco: Opciones para coupons**
 - **v1.1** Editar y eliminar cupones
- **Extensión: Yape**
 - **Leonidas, Sebastián, Francisco: Login Feature**
 - **v1.1:** Login sencillo, comprobación sin base de datos.
 - **v1.2:** Mensaje de error y redirección a Inicio.
 - **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

XV. User Guide

El proyecto propone brindar a los usuarios de Yape los mejores descuentos en un solo lugar y a los restaurantes, las facilidades para promocionar su negocio mediante cupones de descuento en Yape.

YapeCupones existe fuera del aplicativo Yape. Sin embargo, el usuario debe contar con Yape para entrar a YapeCupones y el restaurante debe ser un YapeNegocio para registrarse en YapeCupones con un rol de administrador de restaurante.

Markdown Format on github:

<https://github.com/cs2901/yape-bcp-project-YapePoints/blob/master/UserGuide.md>

1. Introduction:

El mercado de jóvenes que comen fuera de casa y tienen el perfil de Yapero es de 1,1 millones en Lima.

2. Requirements:

a. Functional:

- i. Sincronización entre restaurantes y yaperos.
- ii. Restaurantes: Agregar, editar y eliminar cupones.
- iii. Yaperos: Buscar y filtrar cupones.
- iv. Autenticación de cuentas con Yape-BCP.

b. Non-Functional:

- i. Usabilidad: la plataforma debe ser compatible con todos los dispositivos móviles que soporten previamente el uso de Yape.
- ii. Manejabilidad: la plataforma debe tener una interfaz sencilla, similar a la de Yape, para no confundir a los usuarios.
- iii. Performance: la plataforma debe ser capaz de brindar la información a los usuarios en un tiempo efectivo (debe ser tan o más rápida que Yape).
- iv. Seguridad: la plataforma debe garantizar la integridad de la información de los usuarios.
- v. Escalabilidad: la plataforma debe soportar una gran cantidad de cupones a la vez, sin comprometer la vista que tienen los usuarios.
- vi. Interoperabilidad: la plataforma debe funcionar como un complemento de Yape, no como un servicio independiente.

3. Features:

- a. Yaperos, Filtrar cupones: **filtrar**.
- b. Restaurantes, agregar promoción: **agregar**.

- c. Restaurantes, editar promoción: **modificar**.
- d. Restaurantes, eliminar promociones: **eliminar**.

4. Glossary

- a. Ticket: Precio de un almuerzo o menú.
- b. Restaurante: Establecimiento en el que se preparan y sirven comidas. Nos enfocamos en restaurantes cercanos al centro laboral y de estudios con un ticket promedio menor a S/.20 por menú o almuerzo.
- c. Yape: Sistema creado por el BCP que permite hacer transferencias usando el número de celular.
- d. *YapeUser: Plataforma móvil implementada con Yape que permite a un Yapero visualizar los cupones.
- e. Yape Cupones: Plataforma web para la gestión de cupones para los restaurantes.
- f. Yapero: Usuario de Yape.
- g. Yape Negocio: Un negocio que está afiliado a Yape para usarlo como medio de pago.

5. Anexo A (User Stories and Use Cases).

XVI. Developer Guide

1. Introduction:

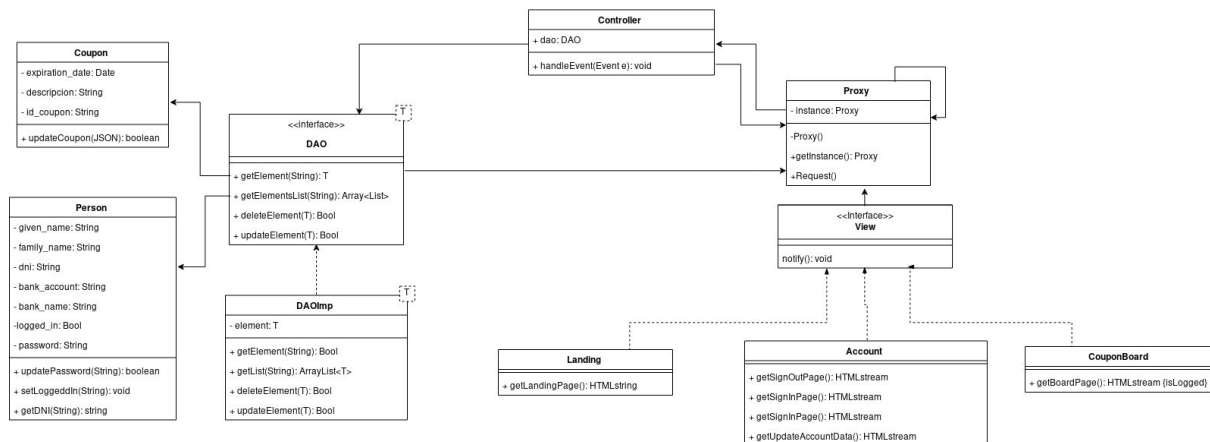
2. Features

- a. Yaperos, Filtrar cupones: **filtrar**.
 - i. Se implementa una búsqueda por distritos y un ordenamiento de lista por cercanía, para simplificar la carga del aplicativo, lo cual lo hace una mejor opción que poseer distintos filtros.
- b. Restaurantes, agregar promoción: **agregar**.
 - i. Se implementa un botón para poder agregar cupones mediante el cual recuperas información como ubicación del restaurante, porque este es un valor que el usuario podría cambiar si posee más locales. Esto es mejor que tener una vista de locales porque nos centramos en cupones y promoción de marca, no en administración de locales.
- c. Restaurantes, editar promoción: **modificar**.
 - i. Se implementa la opción de modificar o editar un cupón porque pueden haber errores al momento de la creación de uno.
- d. Restaurantes, eliminar promoción: **eliminar**.
 - i. Se implementa la opción de eliminar promociones, para frenar las promociones cuando sea necesario. Esto es mejor que tener un contador en el stock o en fecha de vencimiento porque le damos el control a los restaurantes.

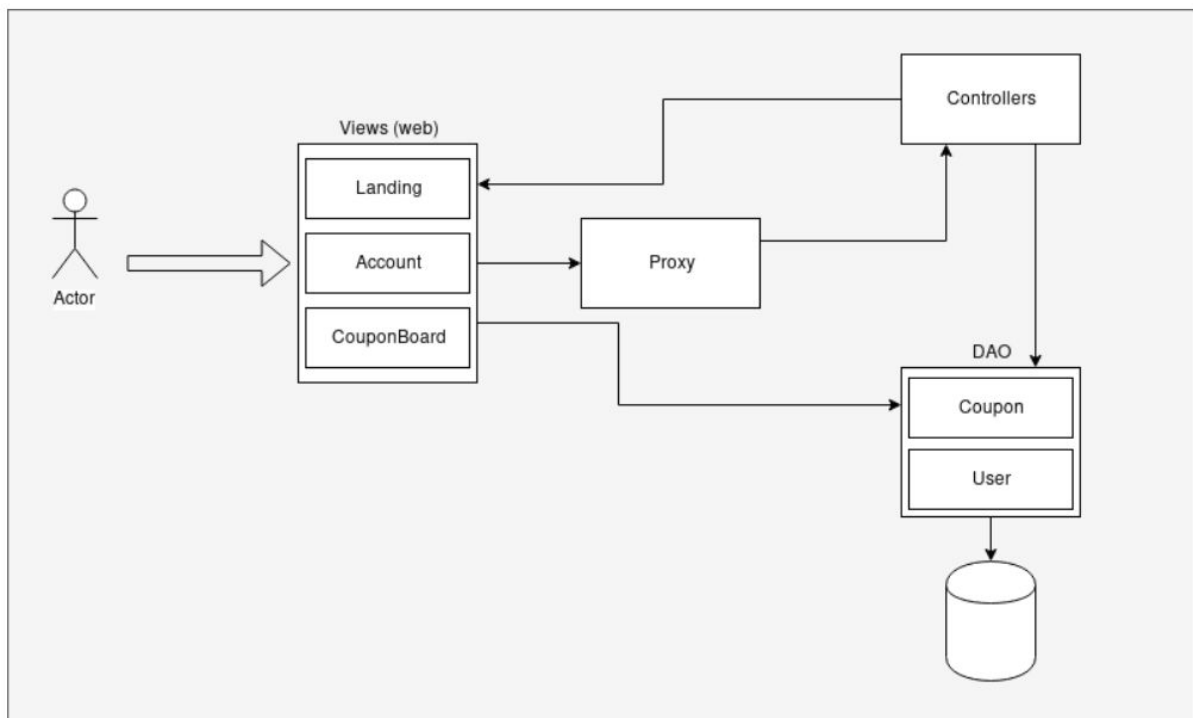
3. Design

a. Architecture

i. Class Diagram

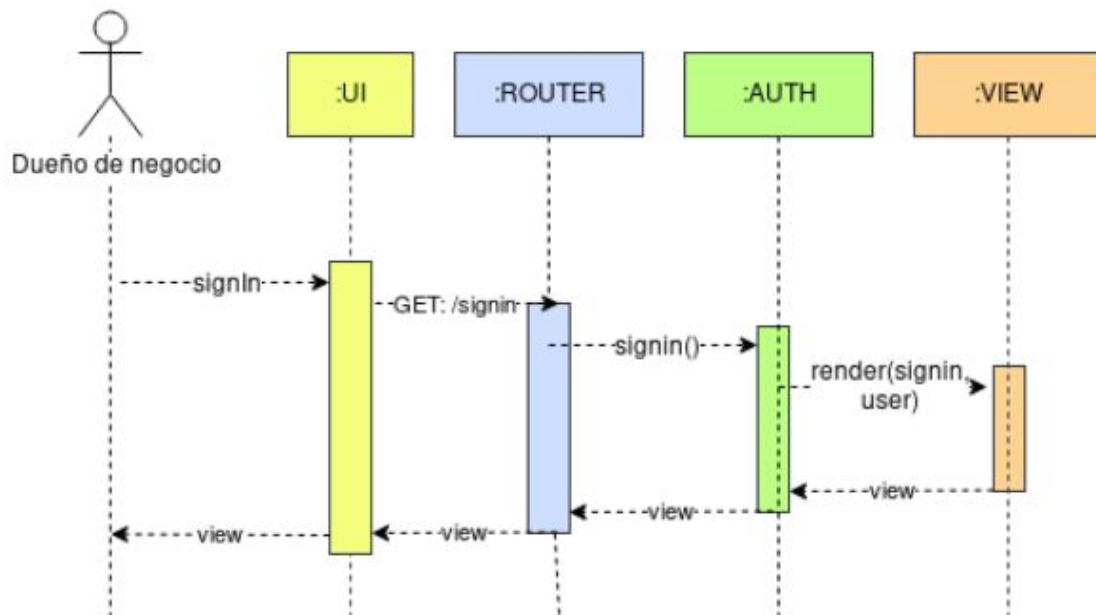


ii. Architecture Diagram

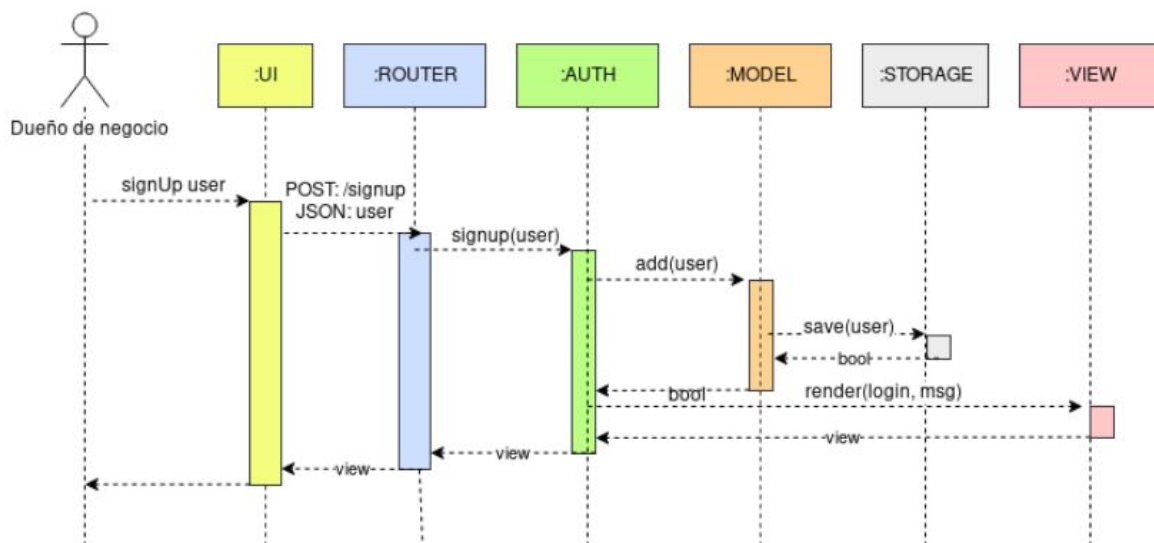


iii. Sequence diagrams

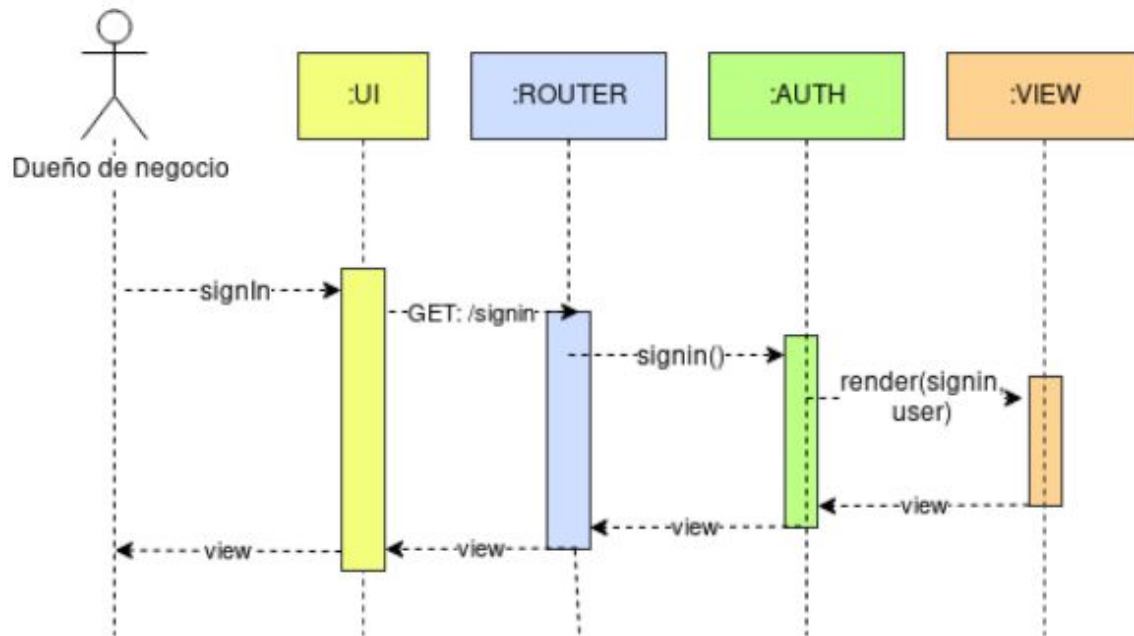
1. Post Sign in



2. Post Sign up



3. Sign in



iv. Interaction Between Components

