Se busca realizar una aplicación que le permita al usuario optar por la mejor opción para hacer rendir su sueldo en Argentina. Orientado especialmente hacia el joven profesional (23-30 años) que busca comparar diversas alternativas factibles acorde a intereses comunes tales como comprar un auto, viajar al exterior, sacar un credito hipotecario, sacar rédito de la especulación financiera (ahorrar en dólares, comprar bitcoins, comprar oro, etc) ser su propio jefe (abrir una pequeña pyme, poner un quiosco).

El usuario podrá ingresar su sueldo, el objetivo deseado y el tiempo pretendido, la aplicación determinará gastos promedio para el usuario acorde a los rubros de alimentos, servicios, movilidad y ocio, recibirá como respuesta cuanto es su potencial de ahorro.

Se plantea la posibilidad de personalizar el usuario así, por ejemplo, si una persona fuese vegana o celíaca su costo de alimentación aumentará en consecuencia. Si una persona viviese con sus padres su costo de servicios disminuiría, etc.

Ante una posible reticencia del usuario a decir abiertamente su sueldo y que la aplicación determine el dinero potencialmente ahorrable se plantea el hecho inverso, en el que determina un objetivo y la aplicación realizará recomendaciones sobre cuánto debería ahorrar para conseguirlo, haciendo ciertas sugerencias tales como que si desea viajar lo haga en tal o cual fecha en que los pasajes o el hospedaje es más barato.

Se plantea en futuro en caso de lograr una masa crítica de usuarios obtener beneficios para los usuarios por ejemplo 10% de descuento en determinado seguro o estación de servicio, probablemente esto funcione mejor

## Consideraciones

- Contar con acceso a internet para obtener precios en tiempo real que permitan una actualización constante de valores de referencia. (Esto requeriría de crawling, en un principio a fin de realizar un prototipo funcional en poco tiempo, se piensa realizar una base de datos local con una actualizacion periodica).
- Hacer un pequeño estudio de mercado para conocer porcentaje sobre cuáles son los intereses más comunes. No Funcional
- Realizar herramienta que ajuste valor de los bienes de un valor pasado a uno presente y prever de un valor actual uno futuro aproximado.
- ¿Qué comprar? ¿Cuándo comprar? Variación en bienes durables y estacionales por ejemplo conocer fecha de salida de modelos nuevos de autos y celulares, que genera variaciones regularmente en el mercado.

## Monetarismo

Se evalúa acorde a variables históricas el rendimiento de 5 patrones estándar tales como bonos del tesoro de EE UU, valor oro, rendimiento de empresa multinacional (Techint, Arcor, YPF), bien de consumo masivo, acción de alto riesgo o bono país emergente.

## Módulos/Requerimientos

- ¿Donde Ahorrar? ---> analizar gastos, señalar early birds para gastos en entretenimiento , sugerencias de ahorro en gastos del usuario de automóvil(pagar patente anual o mensual; que seguro me conviene), sugerencias de compra en supermercados, etc *Deseable, Funcional*
- Deducción del ahorro actual y potencial acorde al salario que se percibe *Básico*
- Estimador poder adquisitivo valor actual > valor futuro *Básico, Funcional*
- Estimador rendimiento dólar-peso > inversion pesos *Basico, Funcional*
- Compra auto que convieine comprar: comparar autos de una misma gama (gol,clio,etios,fiesta por ej) y acorde a precios y planes de pago determinar una preferencia Básico
- Comprar un objeto en el corto plazo (celular/computadora) Básico
- Compra moto que conviene comprar: comparar motos de una misma gama y acorde a precios y planes de pago determinar una preferencia Básico
- Retener el poder adquisitivo (cambio de divisa,compra de oro) se busca solamente resguardar un valor que en pesos se vería afectado por la devaluación, a diferencia del plazo fijo esto permite el acceso a los fondos en forma casi inmediata *Basico*
- Sugerencia de inversión (ser parte de un fondo de inversión en cocheras o edificios, ser parte de un pool de siembra, acceder a algún tipo de bono/accion, sugerir 3 opciones indicando riesgo y repago Deseable
- Llamador, sin loguear al usuario en el login/inicio de aplicación producir articulo con la inversión recomendada del mes con un repago rápido y un riesgo bajo en forma de que el usuario considere el funcionamiento de la app. *Deseable, No Funcional*
- Módulo de gráficos, que se encargue de presentar las comparación de una forma visualmente más comprensible. *Deseable*
- Módulo de alertas: tanto al alejarse de objetivo previsto; como al realizar un gasto innecesario o exagerado; o aparecer una oportunidad de ahorro digna de aprovechar. *Deseable, No Funcional*
- Módulo de divisas e indicadores macroeconómicos: se trata de mantener actualizadas las variables que dan soporte a comparaciones y permiten el funcionamiento de otros módulos como el comparaciones, gráficos e inversiones.

## Alcance del proyecto

- Billetera de usuario: En ella el usuario podrá almacenar información sobre consumos del mes.
- Analizador de gastos: El analizador hará lo que nosotros nunca queremos hacer, analizar los resúmenes de las tarjetas y decirnos en qué estamos gastando nuestro dinero.
- Red de oportunidades de ahorro y descuento: Esta red colaborativa, le va a permitir a los usuarios contribuir a la plataforma con oportunidades de ahorro o descuentos en diversos rubros basados en experiencias personales.

- Estadisticas y metricas: Evolución de mercados, dólar e inversiones.
- Módulo agro: Cotizaciones de las principales commodities en los distintos mercados mundiales.
- Alertas de alejamiento de objetivo. La aplicación detecta aquellas circunstancias en las que el usuario gasta más de lo debido, modificando la curva de ahorro. Envía notificaciones para evitar dichos casos.

## Escalabilidad

La aplicación será completamente escalable, la modularización de la arquitectura permitirá evolucionar el modelo con el fin de alcanzar los objetivos del proyecto.

## Aspectos técnicos

- Tecnología Java
- Spring MVC (Evitable - no queremos un bloque monolítico)
- Spring boot
- Thymeleaf (front end)
- MySql (probablemente implementemos mongoDb)
- JavaScript / jQuery / Angular (Front end)
- Apache Tomcat

# Objetivo

Realizar una aplicación que le permita al usuario optar por la mejor opción para hacer rendir su sueldo en Argentina, orientado especialmente hacia el joven profesional (23-30 años)

# Definición de Requerimientos

Los requerimientos del sistema especifican lo que el sistema de información deberá hacer o cual propiedad o cualidad debe de tener éste.

Los requerimientos del sistema que especifican lo que el sistema de información debe hacer son frecuentemente llamados requerimientos funcionales. Aquellos que especifican una propiedad o cualidad que el sistema debe tener con frecuencia son llamados requerimientos no funcionales.

definir los requerimientos y categorizarlos (deseable, requerido o básico - al menos).

# Definición de Casos de Uso

- Registrar Usuario
- Validar Usuario (loguear)
- Analizar billetera de usuario.
- Potenciar ahorro (subflujo red de ahorro)
- Estimar valor actual a valor futuro
- Comprar bien acorde a sugerencia de aplicación.
- Invertir

# Desarrollo de Casos de Uso

| Caso de uso | **Registrar usuario** |
|---|---|
| Actores | Usuario, Base de datos de registro |
| Tipo | Básico |
| Propósito | Permitir a un usuario registrarse para hacer uso del sistema EPABO. |
| Resumen | Este caso de uso es iniciado por el usuario. Ofrece la funcionalidad de crear el registro de usuario en el sistema. |
| Precondiciones | Ninguna. |
| Flujo Principal | El usuario en la pantalla de inicio (que cuenta con los botones registrarse e ingresar) selecciona el botón registrarse, el sistema |

| | |
|---|---|
| | despliega una pantalla con una serie de campos a completar: nombre de usuario, email , password y repetición de password, -de carácter obligatorio- e información adicional sobre gastos para optimizar el rendimiento de EPABO -de carácter opcional-.<br>Se cierra la pantalla con los campos, se envía un mensaje al usuario notificando el registro exitoso e invitándolo a ingresar. |
| Subflujos | Ninguno. |
| Excepciones | E1-Información incompleta: faltó completar algún campo obligatorio.<br>E2-Nombre de usuario no disponible.<br>E3-Mismatch:No coinciden contraseña y repetición. |

| Caso de uso | **Validar usuario**. |
|---|---|
| Actores | Usuario, Base de datos de registro |
| Tipo | Inclusión. |
| Propósito | Validar un usuario ya registrado para el uso del sistema. |
| Resumen | Este caso de uso es iniciado por el usuario. Valida el usuario mediante un login y password a ser validado con su respectivo registro de usuario para así poder utilizar la app. |
| Precondiciones | Si el usuario aún no se ha registrado, requerirá ejecutar el caso de uso Registrar usuario. |
| Flujo Principal | |
| Subflujos | Ninguno. |
| Excepciones | E4- No hubo validación: El login/password no se validó correctamente. Se le pide al usuario que vuelva a intentar hasta tres veces, después de lo cual se bloqueara al usuario notificando vía mail para su desbloqueo. |

| Caso de uso | **Analizar billetera de usuario**. |
|---|---|
| Actores | Usuario, Base de datos de registro |
| Tipo | Extensión |
| Propósito | Utilizar la funcionalidad analizar billetera. |

| | |
|---|---|
| Resumen | El sistema compara sus gastos separado en cada rubro con el promedio general del sistema y con un ideal básico ideado por la aplicación para hacer uso de los recursos en forma más eficiente |
| Precondiciones | Si el usuario aún no se ha validado, requerirá ejecutar el caso de uso Validar usuario. |
| Flujo Principal | El usuario ya cargo sus datos en todos los campos opcionales del caso de uso registrar usuario, el sistema compara sus gastos acorde a las reglas de negocio de la aplicación para hacer uso de los recursos en forma más eficiente, se muestra esta información al usuario con diferentes gráficos, uno de barras comparando las tres medidas y otro de torta comparando los gastos acorde al rubro.<br>Reglas de negocio:<br>● Gasto exagerado será aquel que represente más del 5% del ingreso destinado a la categoría ocio/entretenimiento.<br>● Gasto inteligente será aquel que se realice con un ahorro en la compra respecto de su precio usual(precio compra anterior) superior al 30% o esté planeado de forma tal de conseguir ese nivel de ahorro mediante el pago de cuotas sin interés o cuyo interés esté por debajo del nivel inflacionario acumulado al término de pago de la última cuota.<br>● Gastos recomendados,sugeridos: prever compras para realizarlas en forma contraestacional aprovechar las liquidaciones de fin de invierno para comprar un abrigo, determinar el mes en el que comprar pasajes de avión, compras al por mayor de productos no perecederos (ligado al caso de uso red de oportunidades de ahorro).<br>● Gasto esencial sería aquel referido a salud, alimentación o vestimenta sobre los cuales es dificultoso su reducción. Con excepción de los siguientes casos:<br>○ salud → netamente estético.<br>○ vestimenta → productos repetitivos, disfraces, accesorios (anteojos de sol, reloj)<br>○ alimentación → comida rápida, bebidas alcohólicas.<br>Filtros en un primer momento para rubro y monto, con la posibilidad de combinarlo |
| Subflujos | Ninguno. |
| Excepciones | E1-Información incompleta:El usuario no completo todos los campos opcionales en el registro, el sistema otorga la opción de completar el registro recomendando dicha acción, caso contrario en expresa negativa del usuario se le hacen preguntas de tipo |

| | |
|---|---|
| | general para la construcción del gráfico de torta, se construye el gráfico de barras con solo dos métricas. |

| | |
|---|---|
| Caso de uso | **Potenciar ahorro.** |
| Actores | Usuario, BD Crawling |
| Tipo | |
| Propósito | Dar soporte a los casos de uso comprar bien acorde a sugerencia de aplicación. |
| Resumen | Valiéndose de crawling la aplicacion encuentra productos que están a partir del 25% menos de su precio de mercado habitual, y los almacena en una base de datos dinámica. |
| Precondiciones | Estar logueado.Tener una antigüedad que permita determinar gastos habituales (1 año). |
| Flujo Principal | El caso de uso es invocado por el caso de uso analizar gastos, para encontrar aquellos gastos que tienen una potencial mejora. |
| Subflujos | |
| Excepciones | |

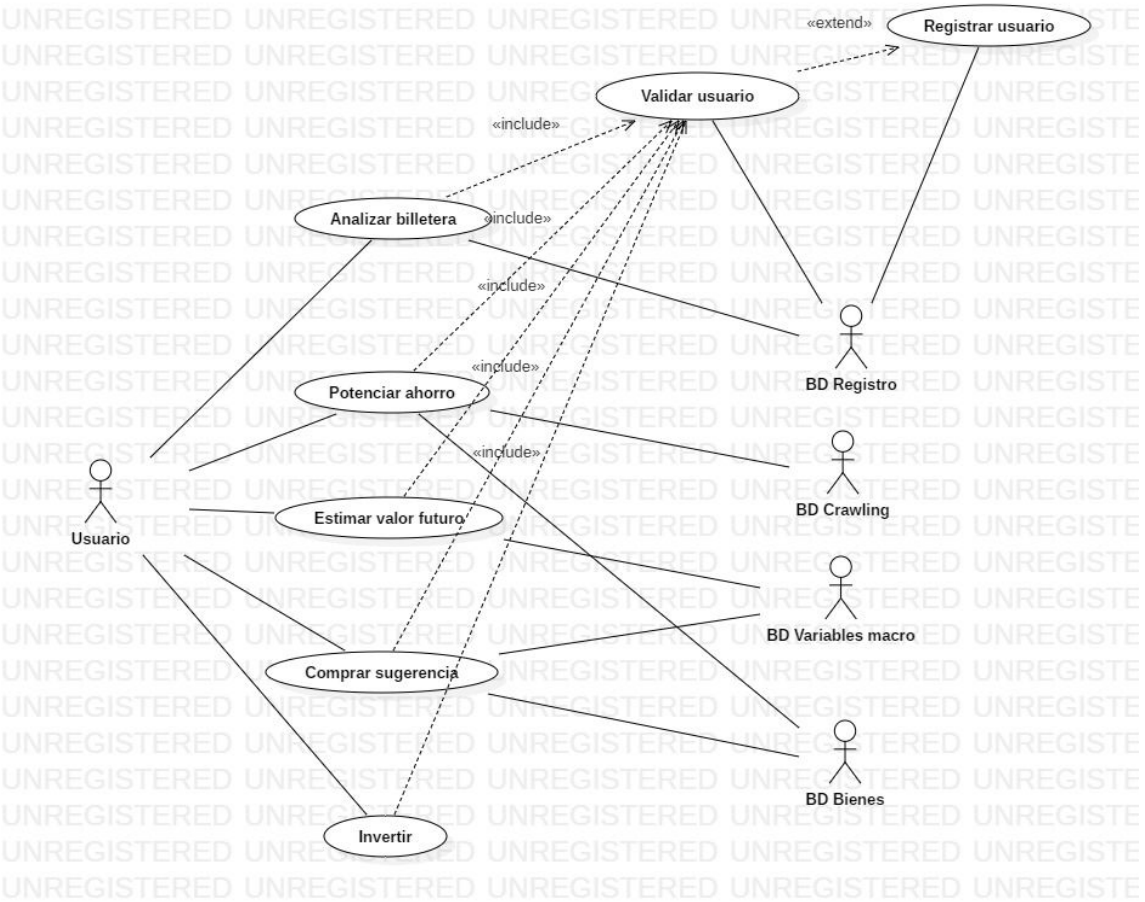
| | |
|---|---|
| Caso de uso | **Estimar valor futuro.** |
| Actores | Usuario, Base de datos variables macroeconómicas. |
| Tipo | Extensión |
| Propósito | Estimar poder adquisitivo equivalente en el tiempo medido en moneda nacional. |
| Resumen | Este caso de uso es iniciado por el usuario.Que quiere estimar su poder adquisitivo en un periodo futuro. |
| Precondiciones | Si el usuario aún no se ha validado, requerirá ejecutar el caso de uso Validar usuario. |
| Flujo Principal | El usuario ingresa el valor a estimar su equivalente futuro y el periodo de tiempo el cual debe ser menor a los dos años, el sistema recurre a la BD para conseguir los parámetros restantes, que se unen a los ingresados por el usuario para realizar la estimación, una vez resuelto el cálculo el sistema devuelve una |

| | |
|---|---|
| | tabla mostrando los dos montos y las fechas. |
| Subflujos | Ninguno. |
| Excepciones | E3- El periodo de tiempo es superior a los dos años por tanto no hay variables en la BD para realizar la estimación, se le pide al usuario que ingrese un periodo de tiempo menor. |

| | |
|---|---|
| Caso de uso | **Comprar bien acorde a la sugerencia de la aplicación.** |
| Actores | Usuario, BD variables macroeconómicas, BD Bienes. |
| Tipo | Extensión |
| Propósito | Decidido un bien deseado determinar que producto específico es más recomendable realizar la compra. |
| Resumen | Este caso de uso es iniciado por el usuario.Que quiere determinar bien de consumo es más recomendable su compra |
| Precondiciones | |
| Flujo Principal | El usuario recibe una buena noticia, consiguió un dinero extra y quiere gastarlo de la mejor manera entonces quiere hacer un gasto pero gastando lo menos posible para lo cual recurre a este caso de uso el cual se sirve de la BD que creo potenciar ahorro para hacer sugerencias acorde a cada usuario.<br>A modo de ejemplo si para el usuario es habitual concurrir a un recital se le notificara con tiempo exacto para que realice la compra en los early birds. |
| Subflujos | |
| Excepciones | |

| | |
|---|---|
| Caso de uso | **Invertir.** |
| Actores | Usuario |
| Tipo | Extension |
| Propósito | Dado un dinero que el usuario no debe hacer uso en forma inmediata desea en primer lugar resguardar su valor, considerando el estado inflacionario del pais, y en caso de ser posible, a un bajo riesgo hacer crecer su poder adquisitivo. |

| | |
|---|---|
| Resumen | El usuario decide el monto por el cual desea realizar su inversion , el sistema le devuelve opciones cercanas y facilita su desarrollo. Por ejemplo le cobra una pequeña comision pero le simplifica la compra de bitcoins y le provee acceso a la evolucion de su compra en comparacion a un valor estandar (dolar).<br>La compra de divisas con una recapitalizacion superior al dolar ligadas a economias emergentes.<br>O ser parte de un fideicomiso reconocido que invertira su dinero por usted dandole una tasa conveniente. |
| Precondiciones | |
| Flujo Principal | |
| Subflujos | |
| Excepciones | |

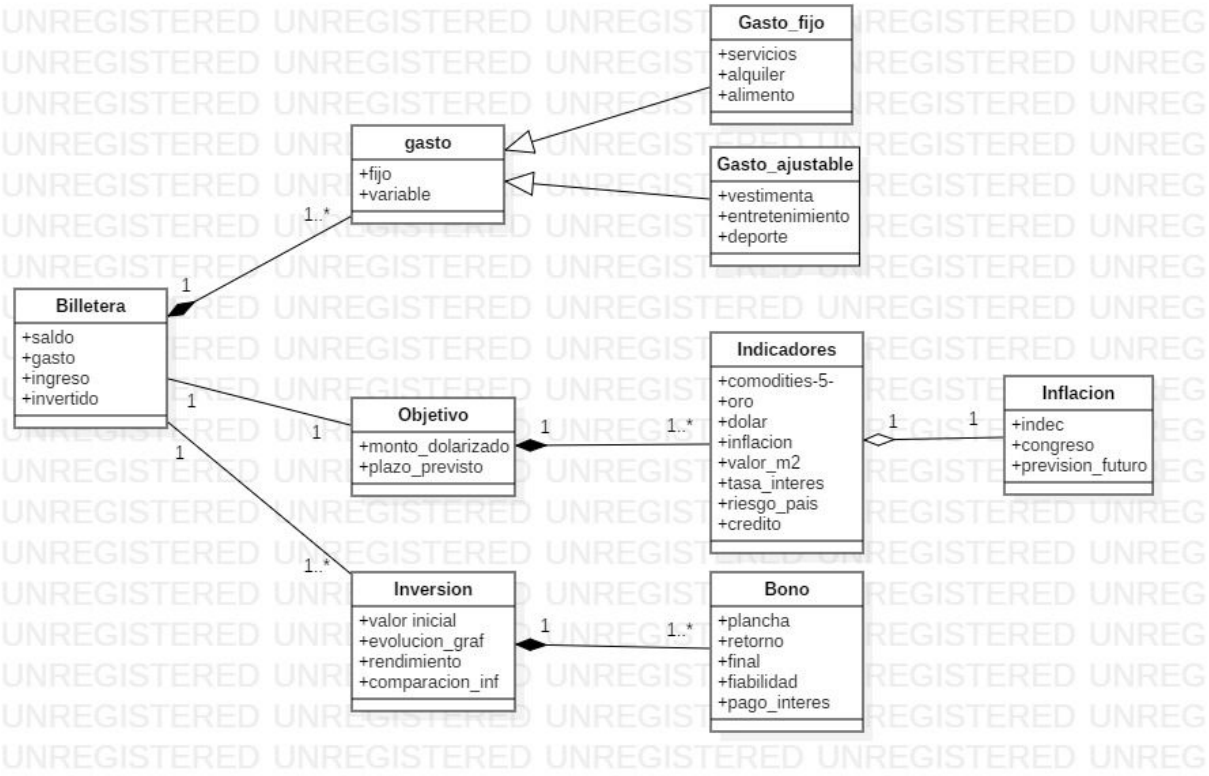



codificar y titular los casos de uso. Hacer un diagrama de casos de uso.

# Seguimiento de Requerimientos

Realizar un cruce entre los requerimientos y los casos de uso, definiendo en cada cruce el tipo de relación que existe (Necesario, parcial, etc)

# Dominio




definir un modelo de dominio. Representarlo en un diagrama UML.

# Diagrama de Estados

En caso de tener un objeto que pase por diferentes estados.
*Entendemos que no tenemos objetos que pasen por más de dos estados, por ejemplo la alarma tendría un activado o desactivado, por tal motivo descartamos por el momento un diagrama de estados.*

# Arquitectura

Dada la velocidad con la que avanzan las nuevas tecnologías, la necesidad de adaptar constantemente el diseño a los nuevos cambios será de vital importancia. Es por eso que optamos por una arquitectura basada en servicios los cuales puedan acoplarse sin causar impacto en el sistema.

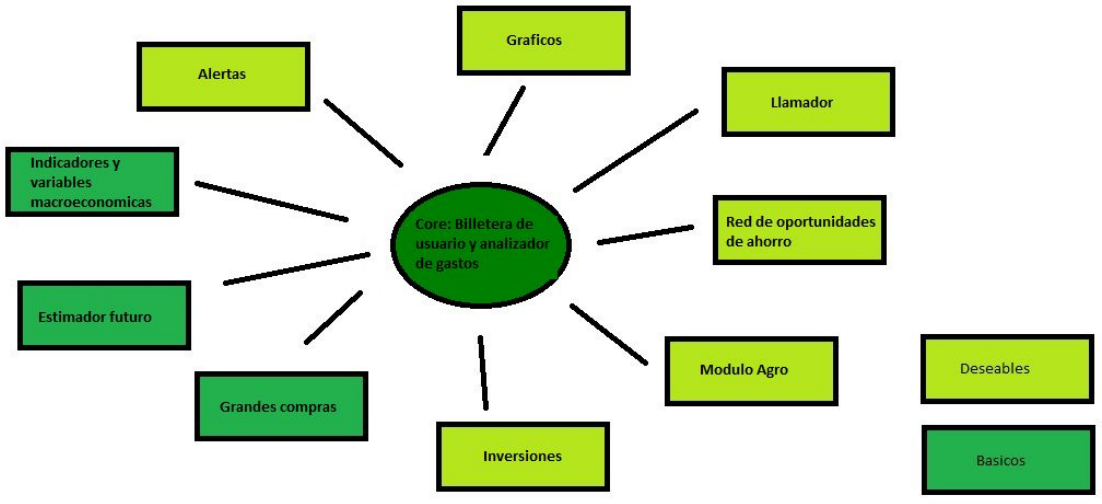
La plataforma será web en su totalidad, apelaremos al uso de servicios web apoyados en el estándar REST.

Permite organizar en forma modulada el desarrollo de microservicios, haciendo que cada integrante del grupo de trabajo pueda codificar un servicio en forma independiente del resto, para luego realizar la integración con el núcleo y los demás microservicios

*La Arquitectura de micro-servicios, conocido por las siglas MSA (del inglés Micro Services Architecture) es una aproximación para el desarrollo de software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros (normalmente una API de recursos HTTP). Cada servicio se encarga de implementar una funcionalidad completa del negocio. Cada servicio es desplegado de forma independiente y puede estar programado en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos. Se suele considerar la arquitectura de microservicios como una forma específica de realizar una arquitectura SOA*

*Los cinco principios del REST son los siguientes:*

- Dar a todas las cosas un identificador
- Vincular las cosas
- Use métodos estándar
- Recursos con múltiples representaciones
- Comunique sin estado




Definir la arquitectura a utilizar y fundamentar la decisión.

# Desarrollo

definir lineamientos para la codificación y para los equipos. Formatos, nombres de variables, cantidad de capas a utilizar, configuraciones, nombres de funciones o procedimientos.

El desarrollo de cada servicio se limitará a su ámbito, dejando a criterio del grupo que desarrolle el módulo las condiciones necesarias para llevar a cabo la implementación. Por otra parte, lo que resulta importante de destacar es que las interfases estarán sujetas a las necesidades del coordinador del grupo, es decir, deberán seguir estrictos lineamientos para poder lograr una integración sin inconvenientes.

# Implementación

La implementación se realizará en un servidor propio al cual se le instaló Ubuntu Server 18.04.1 LTS. Allí se encuentra corriendo Apache 2.4.29, PHP 7.2, MySQL y Apache Tomcat 8. Esto permite que la aplicación esté disponible al usuario en todo momento en cualquier parte.

# Documentación de la API

De momento la API brinda los siguientes endpoints:

Crear nueva gasto en billetera

/v1.0/api/wallets/[Id]/new

| MÉTODO | CREAR GASTO | LISTAR GASTOS |
|---|---|---|
| GET | | /v1.0/api/wallets/{id} |
| POST | /v1.0/api/wallets/[Id]/new | |

| | | |
|---|---|---|
| GET | VER GASTO | /v1.0/api/wallets/{id}/expense/{exId} |

# FormatoJSON:

```
{
        "walletId":ID_BILLETERA,
        "userName":"USR",
        "details":"DETALLLES",
        "amount":DOUBLE,
        "exId":A GENERAR RANDOM
}
```