

# Project Overview

Suzette Mr.

Full-Stack Developer

## Name of the App

Money Home

## Type of App

Single page App

## Description

Esta es una aplicación de tipo Single page App y first mobile, esta enfocada en apoyar la gestión del gasto familiar, a todo tipo de persona que deseen administrar sus ingresos, egresos y metas de ahorro.

Los usuarios podrán registrar datos básicos, registrar diferentes ingresos y egresos ya sean fijos o variables, podrán dar de alta fechas de pagos de gastos (la aplicación enviara un mensaje de recordatorio un día antes de fecha de pago), los usuarios podrán establecer metas de ahorro ya sean anuales o por semestres donde se asignaran fechas ya sean semanales, quincenales o mensuales la cantidad ahorrar, la App también permitirá visualizar graficas de sus ingresos y de gastos.

- Registro historico de ingresos y egresos
- Graficas de ingresos y egresos
- Registro de metas de ahorro
- Registro de cantidades de meta de ahorro
- Recordatorio de fecha de pagos via SMS

## Technologies

Category	Name	Points
Planning and Design		
	Project Planning	Required
	Asana	
	Responsive	20
Techs		
	React	
	JavaScript	
	NodeJs <ul style="list-style-type: none"> <li>• Express</li> <li>• Morgan</li> <li>• Mongodb</li> <li>• Mongoose</li> <li>• Postman</li> </ul>	
	ChartJS	20
	contetfull	30
	Nexmo for sms	10
Build Tools		
	Webpack	20
Hosting		
	Heroku	required
	mLab	
Presentation		
	<b>Cool Slides (swiper)</b>	<b>10</b>

## **Web App Features (MVP requirements)**

- Vista dashboard home
- Vista de registro historico de ingreso
- Vista de registro historico de egresos
- Vista de graficas de ingresos
- Vista de graficas de egresos
- Vista de reguistros de ahorros
- Vista de regristo de metas de ahorro
- Vista de cantidades de metas de ahorro
- Vista de recodatorio de fecha de pagos via SMS

## Project Roadmap (future requirements)

[illegible]

## User Stories

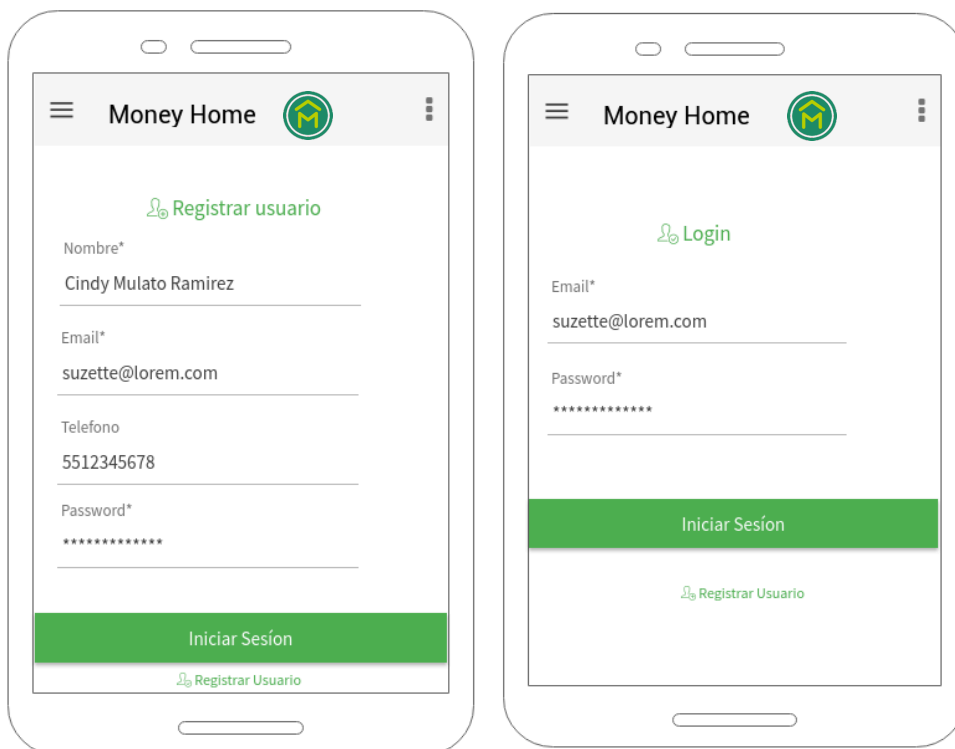
User story: Como usuario requiero una vista de registro o de inicio de sesión para que yo pueda registrarme si aun no cuento con una cuenta.

Suposiciones: El usuario que aun no tiene cuenta podrá darse de alta. En la parte del Back-end tiene autenticación que aceptara solicitudes **post** si aun el email no se encuentra en la base se podran guardar los datos del usuario, en la parte del Front-end tiene la vista de un formulario donde podra ingresar datos del usuario.

### Funcionalidad:

- Usuario tendrá la vista de registro
- Usuario ingresara la informacion en los campos Nombre, Email, Telefono, password
- El Usuario despues de registrar puede ver la vista de inicio de sesión
- Usuario ingresará en los campos el Email y password
- Usuario iniciara sesión si lo datos son correctos

## Wireframes



**User story:** Como usuario quiero visualizar los datos de ingresos, egresos, metas de ahorro y en cuenta general, visualizar la grafica de ingreso, egreso y por gastos.

**Suposiciones:** El usuario que creo la cuenta podrá visualizar el dashboard sobre sus ingreso, egresos metas de ahorro y en cuenta general. Back-end tiene un método que acepta las solicitudes **get** para poder visualizar la información del usuario. En el Front-end tiene una vista donde podrá ver los datos de ingresos y egresos del usuario.

### Funcionalidad

- El usuario puede ver la vista de los campos de ingresos, egresos, meta ahorro y en cuenta general.
- El usuario puede ver la vista de la gráficas de ingresos, egresos y gráficas por gastos.

### Wireframes



User story: Como usuario quiero registrar el concepto, cantidad, fecha, el tipo de mis ingresos.

Suposiciones: El usuario podrá agregar el concepto, cantidad, fecha, el tipo de sus ingresos. Back-end tiene un método de llenado que acepte solicitudes de post para guardar la información del usuario. En Front-end un formulario para guardar los ingresos del usuario.

### Funcionalidad :

- El usuario puede ver la vista de registrar el concepto y el ingreso.
- El usuario puede seleccionar la fecha que fue recibido.
- El usuario puede seleccionar el tipo del ingreso ya sea fijo o variable

### Wireframes

The wireframe shows a mobile application interface for recording income. At the top, there is a header bar with a hamburger menu icon on the left, the text "Money Home" in the center, and a green circular icon with a white 'M' on the right. Below the header, the text "\$ Ingresos" is displayed in green, followed by "\$ 20,000.00" in black. The main form area contains three input fields: "Concepto\*" with the text "Sueldo" entered, "Cantidad\*" with "\$ 20,000.000" entered, and "Fecha" with a calendar picker showing August 2017. Below the calendar, there is a dropdown menu labeled "Tipo por" with a downward arrow, and a list of options: "Variable" and "Fijo". At the bottom of the form, there is a green button labeled "Guardar Gasto".

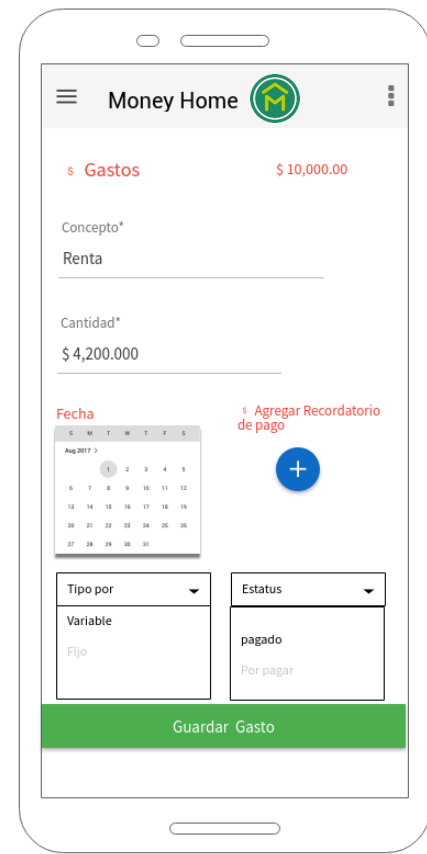
User story: Como usuario quiero registrar el concepto, cantidad, fecha, el tipo, el estatus de mis egresos y poder crear recordatorio antes de la fecha de pago.

Suposiciones: El usuario podrá agregar el concepto, cantidad, fecha, el tipo, el estatus de sus egresos. Back-end tiene un método de llenado que acepta solicitudes post para guardar información del usuario. Front-end tiene un formulario para guardar los datos de egreso del usuario.

## Funcionalidad

- El usuario poder ver la vista de registrar el concepto y el egreso.
- El usuario puede seleccionar la fecha que fue asignado.
- El usuario puede seleccionar el tipo de egreso ya sea fijo o variable
- El usuario puede seleccionar el estatus del egreso
- El usuario poder seleccionar fecha de recordatorio del pago

## Wireframes





User story: Como usuario quiero registrar meta de ahorro como su cantidad, concepto, tipo de ahorro (semanal, quincenal, mensual), fecha de inicio, duración (día, mensual, año).

Suposiciones: El usuario podrá agregar la cantidad, concepto, tipo de ahorro (semanal, quincenal, mensual) fecha de inicio, duración (día, mes, año). Back-end tiene un método de llenado que acepta solicitudes post para guardar información del usuario. Front-end tiene un formulario para guardar los datos de meta de ahorro del usuario.

## Funcionalidad

- El usuario puede ver la vista de registrar cantidad y el concepto.
- El usuario puede seleccionar el tipo de ahorro (semanal, quincenal, mensual)
- El usuario puede seleccionar la fecha de inicio.
- El usuario puede seleccionar la duración (día, mes, año)

## Wireframes

The wireframe shows a mobile app interface for registering a savings goal. The screen is titled "Money Home" with a hamburger menu icon on the left and a coin icon on the right. Below the title is a section labeled "Meta de ahorro". The form includes the following fields and controls:

- Cantidad\***: A text input field containing "\$ 100,000.00".
- Concepto\***: A text input field containing "Enganche carro".
- Ahorro por**: A dropdown menu with three options: "Semanal", "Quincenal", and "Mensual".
- Inicio**: A calendar widget showing the month of August 2017, with the 1st day highlighted.
- Duración\***: A text input field containing "1".
- Ahorro por**: A second dropdown menu with three options: "Dia(s)", "Mes(es)", and "Año(s)".
- Guardar Meta de ahorro**: A green button at the bottom of the form.

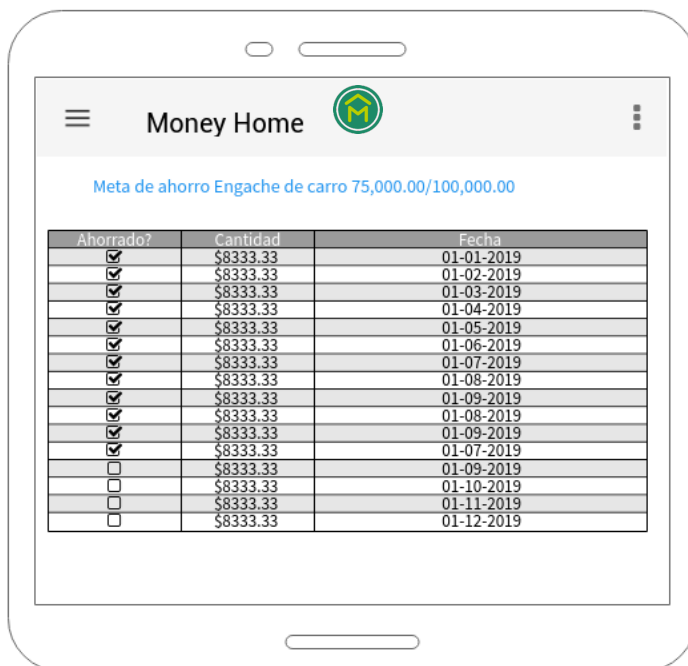
User story: Como usuario quiero ver el historial de meta ahorro donde muestre lo ahorrado, cantidad y fecha que se ingreso anteriormente.

Suposiciones: El usuario podrá ver el historial de su meta de ahorro. Back-end tiene un método de solicitudes get para mostrar información de la meta de ahorro del usuario. Front-end crea una vista que puede mostrar el ahorro, cantidad y fecha de la meta de ahorro del usuario.

### Funcionalidad

- El usuario poder ver la vista historial de la meta de ahorro
- El usuario puede seleccionar si esta ahorrado

### Wireframes



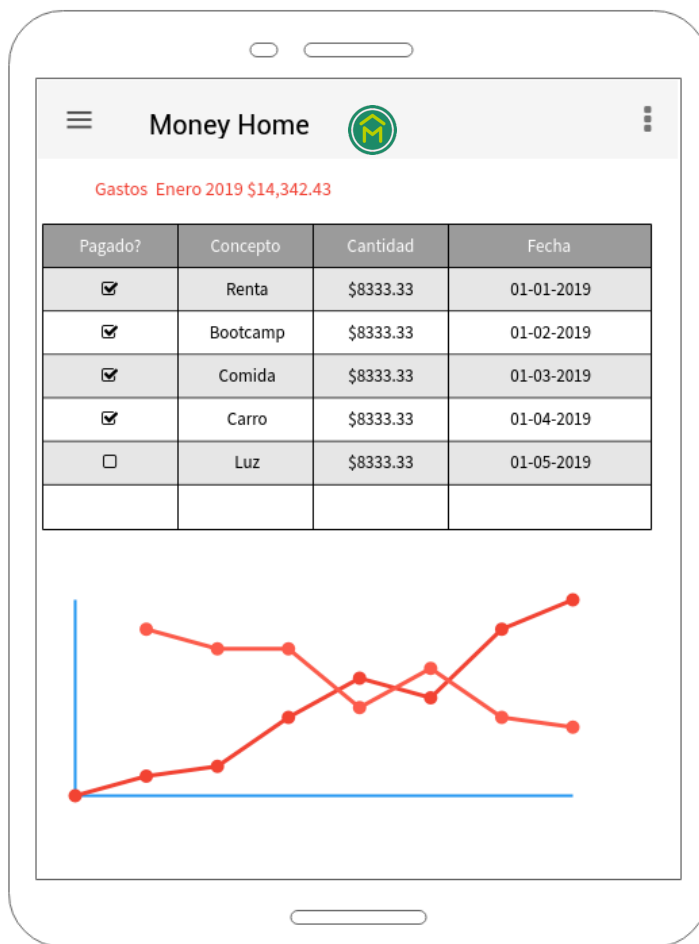
User story: Como usuario quiero visualizar una tabla con los gastos del mes tambien puedo editar y eliminar un dato erroneo.

Suposiciones: El usuario podrá ver el historial de sus gastos por mes. Back-end tiene un método de solicitudes **get** para mostrar información de los gastos del usuario **put** podrá modificar si un dato o **delete** para eliminar un dato erroneo. Front-end crea una vista que puede mostrar el gasto, estatus, concepto, cantidad y fecha de los gastos del usuario, actualizar los campos o eliminarlos.

### Funcionalidad

- El usuario poder ver la vista historial de los gasto
- El usuario puede seleccionar si esta pagado
- El usuario puede actualizar los datos
- El usuario puede eliminar los datos erroneos

### Wireframes

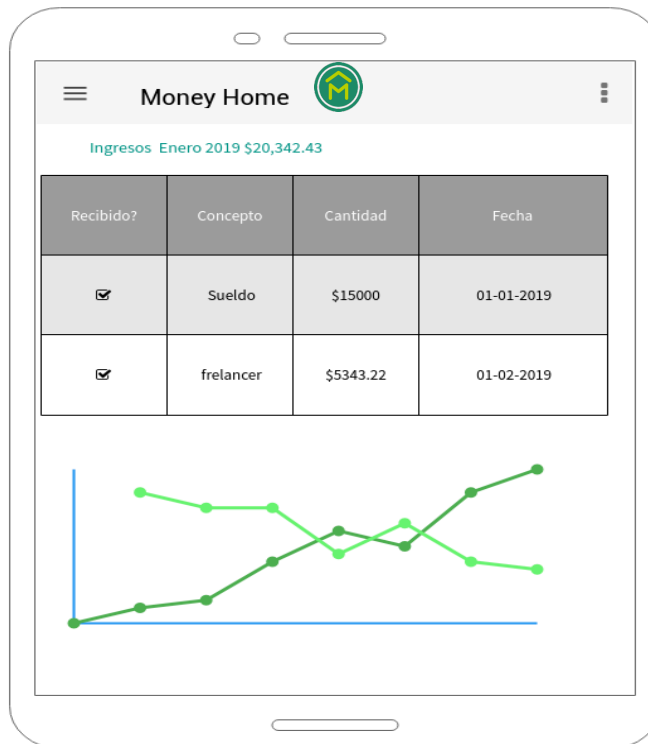


User story: Como usuario quiero visualizar una tabla con los ingresos del mes tambien puedo editar y eliminar un dato erroneo.

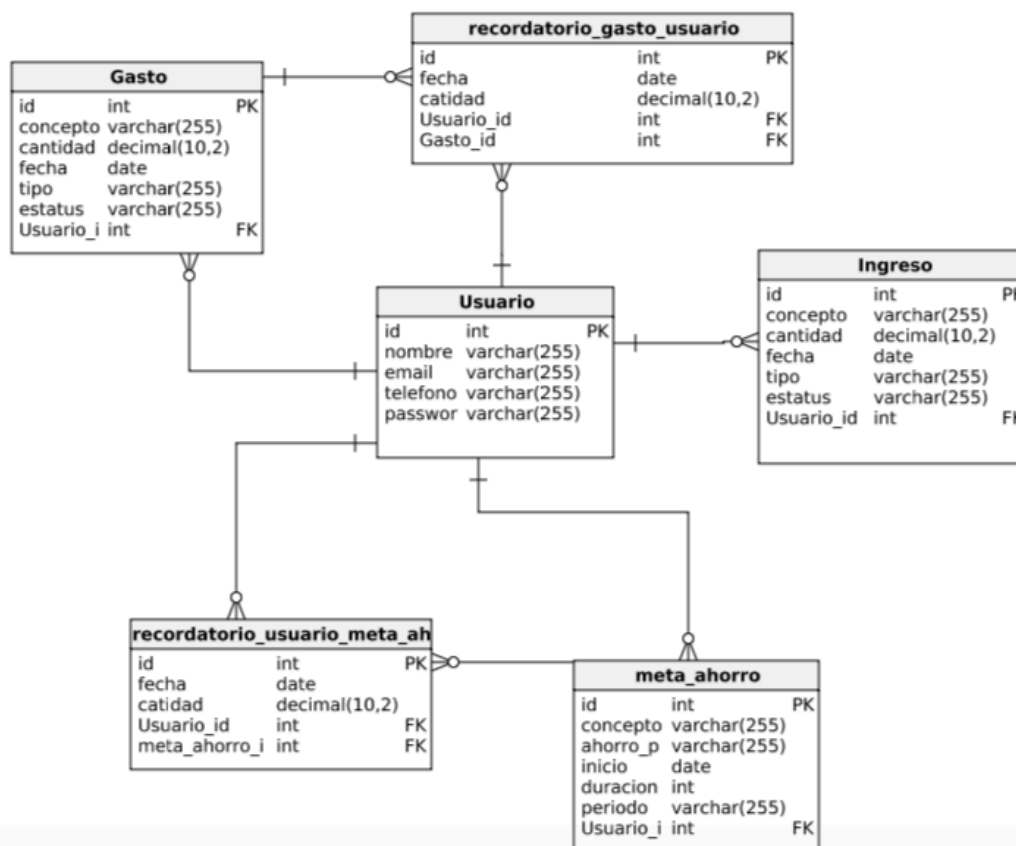
Suposiciones: El usuario podrá ver el historial de sus ingresos por mes. Back-end tiene un método de solicitudes **get** para mostrar información de los ingreso del usuario **put** podrá modificar si un dato o **delete** para eliminar un dato erroneo. Front-end crea una vista que puede mostrar el ingreso, recibido, concepto, cantidad y fecha de los gastos del usuario, actualizar los campos o eliminarlos.

### Funcionalidad

- El usuario puerder ver la vista historial de los ingresos
- El usuario puede seleccionar si esta recibido
- El usuario puede actualizar los datos
- El usuario puede eliminar los datos erroneos



## Data Model



## Routes

### Client Routes

GET “ / ”

GET “ /users”

POST “/users”

PUT “/users/userId”

DELETE “/users/userId”

GET “/users/userId/ingresos”

POST “/users/userId/ingresos”

PUT “/users/userId/ingresos/ingresold”

DELETE “/users/userId/ingresos/ingresold”

GET “/users/userId/egresos”

POST “/users/userId/egresos”

PUT “/users/userId/egresos/egresold”

DELETE “/users/userId/egresos/egresold”

GET “/users/userId/metaAhorro”

POST “/users/userId/metaAhorro”

PUT “/users/userId/ metaAhorro/metaAhorrold”

DELETE “/users/userId/metaAhorro/metaAhorrold”