Metodología SCRUM

Alejandro Jiménez Jiménez

20 – 05 – 2024

ICOM

Ingeniería de Software

Contenido

Introducción

La metodología SCRUM es una metodología ágil, lo cual significa que es una metodología que nos permitirá realizar un software de una manera muy volátil y adaptativa, esto gracias a que esta metodología trabaja con “sprints”, los que básicamente son una pequeña división de todo el proyecto, la cual se trabaja por un cierto tiempo por todo el equipo, para que, al finalizar, seguir con otro sprint para así, poco a poco ir terminando el proyecto.

Las fases de la metodología son:

1. Formación del Equipo Scrum: El primer paso es formar un equipo Scrum que consista en un Product Owner, un Scrum Master y el Equipo de Desarrollo.

2. Definición del Backlog del Producto: El Product Owner es responsable de crear y priorizar el Backlog del Producto, que es una lista de todas las características, mejoras y arreglos que se deben realizar en el producto.

3. Planificación del Sprint: El equipo Scrum se reúne para planificar el próximo Sprint. Durante esta reunión, seleccionan un conjunto de elementos del Backlog del Producto que trabajarán durante el Sprint.

4. Desarrollo del Sprint: Durante el Sprint, el equipo trabaja en las tareas seleccionadas. Se llevan a cabo reuniones diarias cortas llamadas Scrum Daily Standups para mantenerse actualizados sobre el progreso y cualquier problema que surja.

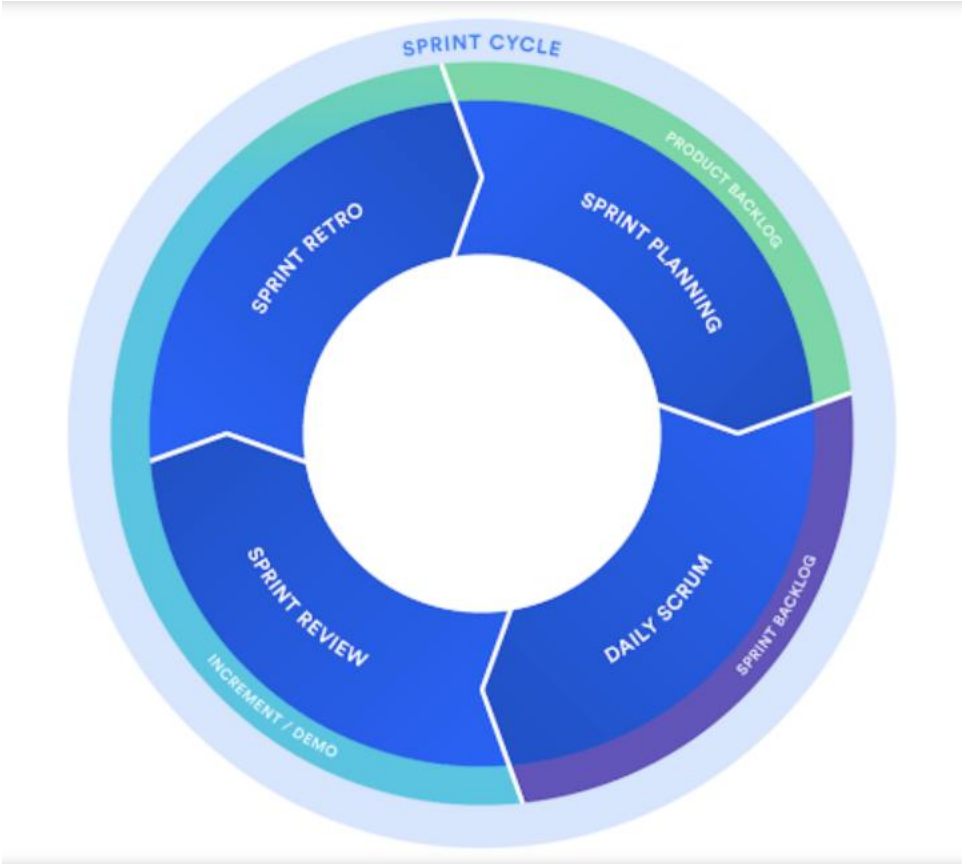
5. Reunión de Revisión del Sprint: Al final del Sprint, el equipo realiza una reunión de revisión del Sprint para demostrar el trabajo completado al Product Owner y a otros interesados. Se recopila retroalimentación y se ajusta el Backlog del Producto según sea necesario.

6. Reunión de Retrospectiva del Sprint: Después de la reunión de revisión del Sprint, el equipo Scrum lleva a cabo una reunión de retrospectiva del Sprint para reflexionar

sobre lo que salió bien, lo que podría mejorarse y cómo mejorar el proceso para el próximo Sprint.

7. Inicio del siguiente Sprint: Con la finalización de la retrospectiva del Sprint, comienza el siguiente Sprint y el ciclo se repite.

Estos son los pasos básicos de la metodología Scrum, pero la implementación puede variar según las necesidades y circunstancias específicas de cada equipo y proyecto, por ejemplo, para este proyecto sólo hay un integrante, por lo tanto, las reuniones para la revisión y planeación de los sprints serán realizados por sólo esta persona (yo), al igual que todos los roles como el Product Owner, Sprint Master y el grupo de desarrollo serán realizados por una persona, así que la primera fase de la metodología será suprimida.



Definición del Backlog del Producto

Se necesita un sistema el cual genere un plan de ahorro a partir de ciertos parámetros que cada usuario va a introducir para cumplir cualquier meta en específico que también cada usuario seleccionará.

Recursos necesarios

- Humanos: El equipo de trabajo estará conformado por una persona (yo).
- Físicos: Sólo se requiere una laptop/computadora capaz de ejecutar los recursos tecnológicos.
- Tecnológicos: Se utilizarán el programa Visual Studio Code, con la extensión “live server” la cual nos ayudará a estar observando los cambios en tiempo real, y un navegador en donde se ejecutará nuestro programa, en este caso utilizaré Google Chrome.
- Informáticos: Se utilizará el lenguaje HTML para el modelado de la página, CSS para ajustar los elementos y JavaScript para hacer los cálculos y funcionalidades del programa.

Diagrama de casos de uso

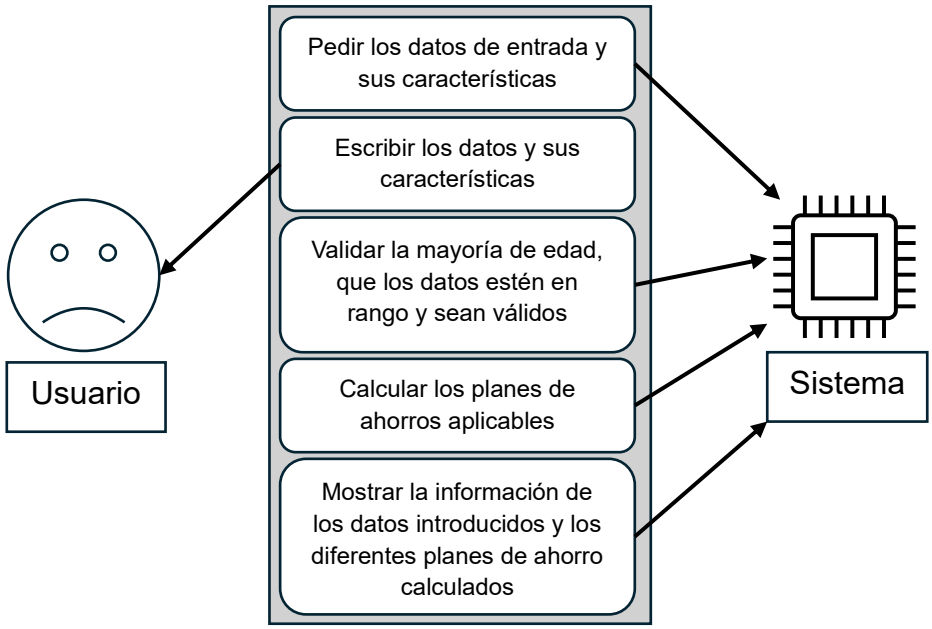


Diagrama de flujo de datos

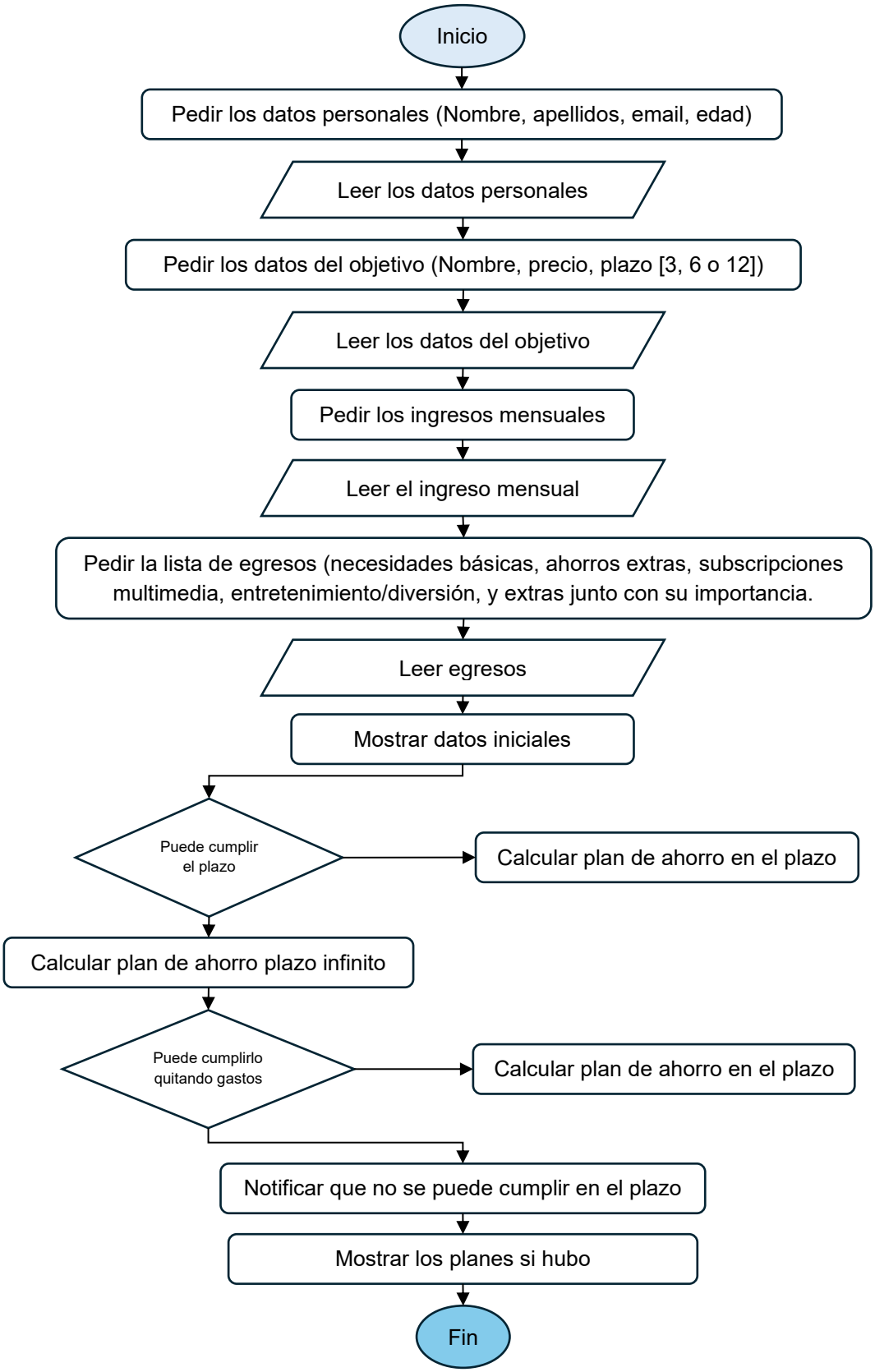


Diagrama de clases

Datos_Personales
str = Nombre str = Apellido email = Correo smallint = Edad

Objetivo
str = Nombre int = Precio list[3,6,12] = plazo

Egresos
int = Básicas int = ahorros int = multimedia int = entretenimiento int = extras list[1,2,3,4,5] = importancia

Planificación del primer sprint

En este sprint se llevará a cabo la recolección de los datos de la persona que solicita el reporte, su objetivo, sus ingresos y egresos.

Para que el programa tenga un buen funcionamiento, debemos de tener todos los datos, por lo tanto, debemos de asegurar que el usuario no pueda solicitar el reporte si no introduce todos sus datos. Hay un riesgo de que el usuario se equivoque al ingresar los datos, así que tendremos que hacer que el formulario sea editable en cualquier momento antes del cálculo.

Los datos para solicitar y las especificaciones de sus variables son los siguientes:

- Nombre del usuario (cadena)
- Apellidos (cadena)
- email del usuario (cadena verificando que sea un email)
- Edad del usuario (numérico, mayor que 0)

- Nombre del objetivo (cadena)
- $ del objetivo (numérico mayor que 0)
- Seleccionar plazo (3, 6 o 12 meses)
- $ de ingresos totales (numérico mayor que 0)
- Egresos:
    - $ de egresos en multimedia (música, multimedia, subscripciones en general) (numérico mayor que 0)
    - $ de egresos en necesidades básicas (renta, agua, luz, comida básica) (numérico mayor que 0)
    - $ de egresos entretenimiento (cine, cafeterías, diversiones, videojuegos) (numérico mayor que 0)
    - $ de egresos que quiera guardar (gastos médicos, inversiones) (numérico mayor que 0)
    - Otros gastos (se preguntará que tan importante son estos gatos, si son más importantes que los de entretenimiento, que las de multimedia que el dinero a guardar o igual de importante que las necesidades básicas) (numérico mayor que 0)

Las validaciones para asegurarnos que los datos son correctos serán hechos con las validaciones de los forms de HTML, serán enviados por el método “GET” hacia otro documento HTML, en el cual, guardaremos los datos en otras variables (será necesario cambiar de tipo de dato a los valores numéricos).

Será necesario imprimir los datos recibidos para poder corroborar que se han recibido los datos correctamente.

Desarrollo del primer Sprint

Primero crearemos el documento HTML con los parámetros generales del documento, le pondremos de título “Plan de ahorro”, el idioma en español, al igual que lo vincularemos al otro documento CSS.

```html
<!DOCTYPE html>
<html lang="es">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Plan de ahorro</title>
    <link rel="stylesheet" type="text/css" href="estilos.css">
</head>
```

Comenzamos poniendo un título, y creando el formulario en una tabla, la cual se dividirá en varios segmentos, el primero son los datos personales.

```html
<tr>
    <th colspan="2"> DATOS PERSONALES </th>
</tr>
<tr>
    <td>Nombre </td>
    <td><input type="text" placeholder="Introduzca su nombre" name="nombre" id="nombre" required
            autocomplete="off"></td>
</tr>
<tr>
    <td>Apellidos </td>
    <td><input type="text" placeholder="Introduzca sus apellidos" name="apellidos" id="apellidos" required
            autocomplete="off">
    </td>
</tr>
<tr>
    <td>Correo electrónico </td>
    <td> <input type="email" name="email" id="email" placeholder="Introduzca su email" required
            autocomplete="off"></td>
</tr>
<tr>
    <td>Edad </td>
    <td> <input type="number" name="edad" id="edad" min="1" required placeholder="¿Cuál es su edad?"></td>
</tr>
<tr>
```

Luego se piden los datos del objetivo

```html
    <th colspan="2">OBJETIVO</th>
</tr>
<tr>
    <td>Nombre</td>
    <td><input type="text" name="nomobjetivo" id="nomobjetivo" placeholder="Nombre del objetivo" required
            autocomplete="off">
    </td>
</tr>
<tr>
    <td>¿Cuánto cuenta tu objetivo?</td>
    <td><input type="number" name="cantobjetivo" id="cantobjetivo" placeholder="Costo del objetivo"
            required></td>
</tr>
<tr>
    <td>¿A qué plazo quiere su objetvo?</td>
    <td> <select name="plazo" id="plazo" required>
            <option value="" disabled selected>Selecciona una opción</option>
            <option value="3">3 meses</option>
            <option value="6">6 meses</option>
            <option value="12">1 año</option>
        </select></td>
</tr>
```

Los ingresos totales mensuales

```html
    <th colspan="2">INGRESOS</th>
</tr>
<tr>
    <td colspan="2" class="centrar"> Favor de ingresar el total de ingresos <strong>mensuales</strong> que
        usted obtiene, contando su salario fijo y otros ingresos que usted pueda generar</td>
</tr>
<tr>
    <td>Ingresos mensuales</td>
    <td><input type="number" name="ingresos" id="ingresos" required min="1"
            placeholder="Ingresos mensuales"></td>
</tr>
```

Luego todos los distintos tipos de egresos

```
    <th colspan="2">EGRESOS</th>
</tr>
<tr>
    <td colspan="2" class="centrar"><em>Favor de leer bien la descripción de cada uno de los diferentes
            tipos de egresos para poder calcular su reporte con mayor exactitud, todos los egresos son
            <strong>mensuales</strong>. Si no tiene ningún gato en alguna categoria, puede ingresar el valor
            "0".</em><br><br></td>
</tr>
<tr>
    <td><strong>a) Necesidades básicas</strong><br>Toma en cuanta todos esos gastos indispesables que no
        puedes dejar pasar cada mes (agua, luz, comida básica) <br><br></td>
    <td><input type="number" name="basicos" id="basicos" required min="0" placeholder="$"></td>
</tr>
<tr>
    <td><strong>b) Ahorros extras</strong><br>Cantidad de dinero que puedes estar ahorrando mensualmente
        para
        otros objetivos (ahorros o seguro médico, inversiones, otros ahorros, etc) <br><br></td>
    <td><input type="number" name="ahorros" id="ahorros" required min="0" placeholder="$"></td>
</tr>
<tr>
    <td><strong>c) Subscripciones multimedia</strong><br>Subscripciones de música, videos, subscripciones en
        general (Spotify, Netflix, Amazon Prime, etc) <br><br></td>
    <td><input type="number" name="subscripciones" id="subscripciones" required min="0" placeholder="$">
    </td>
</tr>
<tr>
    <td><strong>d) Entretenimiento/Diversion</strong><br>Gastos para entretenerse y divertirse que podrian
        reemplazarse con otras cosas (cine, cafeterias, parques de diveriones, etc) <br><br></td>
    <td><input type="number" name="diversion" id="diversion" required min="0" placeholder="$"></td>
</tr>
<tr>
    <td><strong>e) Gastos extras</strong><br>Otros gastos que no entran en las categorias anteriores, deberá
        de
        especificar que tan <strong>importantes</strong> <br>son respecto a las otras categorías, siendo e)
        esta categoria <br></td>
    <td><input type="number" name="extras" id="extras" required min="0" placeholder="$"><br> <br>
        <select name="importanciaextra" id="importanciaextras" required>
            <option value="" disabled selected>Nivel de importancia</option>
            <option value="1">1: e) > a)</option>
            <option value="2">2: a) > e) < b)</option>
            <option value="3">3: b) > e) < c)</option>
            <option value="4">4: c) > e) < d)</option>
            <option value="5">5: d) < e)</option>
        </select>
    </td>
</tr>
```

Por ultimo, en el otro documento recibimos todos los datos enviados por el usuario, verimicamos que sí han llegado mostrandolos por consola.

```
<body>
    <script>
        const PARAMETROS = new URLSearchParams(window.location.search);
        const NOMBRE = (PARAMETROS.get("nombre") + " " + PARAMETROS.get("apellidos"));
        const EMAIL = PARAMETROS.get("email");
        const EDAD = PARAMETROS.get("edad")
        const OBJETIVO = PARAMETROS.get("nomobjetivo");
        var target = PARAMETROS.get("cantobjetivo")
        var plazo = PARAMETROS.get("plazo")
        var ingresos = PARAMETROS.get("ingresos");
        var egreso1 = PARAMETROS.get("basicos");
        var egreso2 = PARAMETROS.get("ahorros");
        var egreso3 = PARAMETROS.get("subscripciones");
        var egreso4 = PARAMETROS.get("diversion");
        var egreso5 = PARAMETROS.get("extras");
        var importancia = PARAMETROS.get("importanciaextra");

        console.log(NOMBRE, EMAIL, EDAD, OBJETIVO, target, plazo, ingresos, egreso1, egreso2, egreso3, egreso4, egreso5, importancia)
    </script>
</body>
```

Revisión y retrospectiva del sprint

Primero se revisa que se haya cumplido el requerimiento del formulario en una tabla, el cual sea entendible, editable y fácil de comprender, también se revisará que se han pedido todos los parámetros y variables establecidas, al igual que se incluyan las leyendas y los requerimientos de los avisos necesarios:

Plan de ahorro

DATOS PERSONALES

Nombre	Alejandro
Apellidos	Jiménez
Correo electrónico	Ejemplo@gmail.com
Edad	20

OBJETIVO

Nombre	ejemplo
¿Cuánto cuesta tu objetivo?	1000
¿A qué plazo quiere su objetvo?	3 meses

INGRESOS

Favor de ingresar el total de ingresos **mensuales** que usted obtiene, contando su salario fijo y otros ingresos que usted pueda generar

Ingresos mensuales	10000

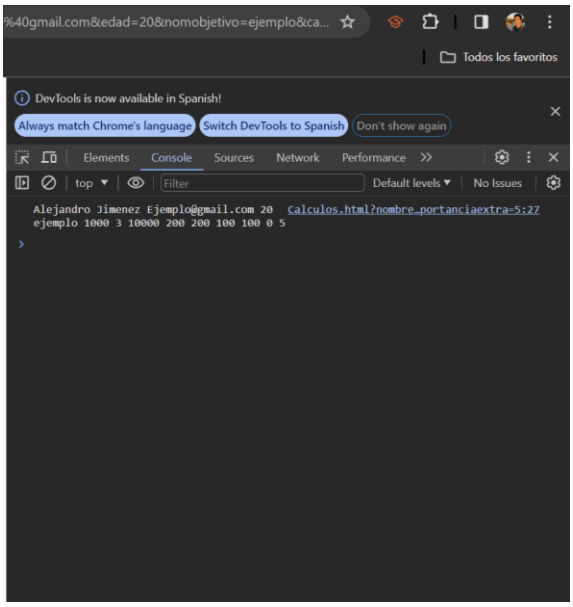
EGRESOS

*Favor de leer bien la descripción de cada uno de los diferentes tipos de egresos para poder calcular su reporte con mayor exactitud, todos los egresos son **mensuales**. Si no tiene ningún gato en alguna categoría, puede ingresar el valor "0".*

a) Necesidades básicas Toma en cuanta todos esos gastos indispesables que no puedes dejar pasar cada mes (agua, luz, comida básica)	200
b) Ahorros extras Cantidad de dinero que puedes estar ahorrando mensualmente para otros objetivos (ahorros o seguro médico, inversiones, otros ahorros, etc)	200
c) Subscripciones multimedia Subscripciones de música, videos, subscripciones en general (Spotify, Netflix, Amazon Prime, etc)	100
d) Entretenimiento/Diversion Gastos para entretenerse y divertirse que podrían reemplazarse con otras cosas (cine, cafeterías, parques de diveriones, etc)	100
e) Gastos extras Otros gastos que no entran en las categorías anteriores, deberá de especificar que tan **importantes** son respecto a las otras categorías, siendo e) esta categoría	0 5 Menor importancia

Calcular reporte

Luego revisaremos que los parámetros sean recibidos correctamente, para esto se han mandado las variables a otra página y son mostrados por consola.



Como pudimos observar, todos los parámetros han llegado y se ha cumplido el objetivo definido, el cual era capturar todos los datos de una manera entendible, editable, explicando cuales eran los aspectos por considerar para cada dato. Podemos concluir que este primer sprint ha sido completado correctamente y sin ningún error gracias a estas sencillas pruebas.

Planificación del segundo sprint

El objetivo de esta vuelta será realizar los cálculos generales para poder hacer el reporte del usuario utilizando los datos previamente recolectados.

Los cálculos deben de ser contundentes y exactos, deben de tener en cuenta que pueden haber puesto cualquier cantidad, por lo tanto, se necesitarán hacer diversas validaciones. Al igual que puede que los resultados de estos cálculos no sean representativos de una forma coherente debido a que los resultados pueden ser muy variados dependiendo si puede cumplirlo cuando dijo, si puede cumplirlo antes, si no puede cumplirlo, en cuanto lo cumpliría, si ese nuevo tiempo cabe en otro plazo y si no, hacer recomendaciones para poder conseguirlo en el plazo más grande (1 año).

Para este sprint se requiere:

Utilizar una función para hacer un reporte, la cual será llamada para recibir los datos del tipo de plan de ahorro aplicable para cada usuario.

1. Primero se calcularán los egresos totales del usuario, sumando todos sus tipos de egresos, luego se hará un balance, en el cual sabremos cuánto dinero puede guardar al mes.
2. Aquí tendremos que hacer una validación si tiene un balance positivo, ya que, si gasta más de lo que gana, no podrá guardar nada, si es positivo, seguirá los siguientes pasos:
    2.1. Se condicionará si se puede cumplir el objetivo en el tiempo que especificó, esto se hará multiplicando el balance por los meses del plazo.
        2.1.1. Si el resultado es igual o mayor a la cantidad que se requiere para el objetivo: se procederá a hacer el reporte con las cantidades que tendrá que guardar cada mes, esto se calculará dividiendo el total que cuesta el objetivo en los meses del plazo, luego lo dividiremos en dos para sacar cuanto es lo que debe de ahorrar cada quincena, se redondeará para arriba, esto para evitar que no nos falte ningún peso.
        2.1.2. Si no, se especificará cuánto tiempo le tomaría en cumplir el objetivo sin hacer ningún cambio en sus egresos, esto se calculará dividiendo el precio del objetivo entre el balance del usuario y redondeándolo hacia arriba, esto nos dará la cantidad de meses en que cumpliría el objetivo. Entonces se hará el reporte utilizando el nuevo plazo
            2.1.2.1. Luego se comenzará a ordenar los egresos de mayor a menor importancia, esto con el objetivo de comenzar a tomar los egresos de menor importancia para sugerir su ahorro y así, completar el objetivo en el plazo seleccionado. Esto se hará primero obteniendo el monto faltante para cumplir el objetivo a tiempo, luego se le sumara el egreso con menos importancia multiplicado por el plazo, para así, saber si ese egreso es suficiente, si no, se continuará a hacer lo mismo con los ingresos un poco más importantes.

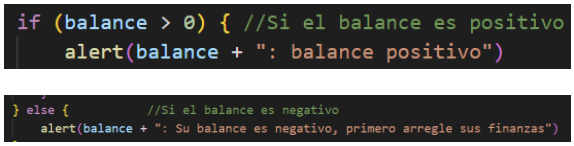
2.1.2.2. Si al terminar con todos los egresos, no es de plano posible terminarlo a tiempo, se le notificará al usuario que es imposible cumplir el plazo, aunque ahorre todos sus ingresos.

2.2. Si no tiene un balance positivo, se mostrará un mensaje el cual le haga saber que su balance es negativo o 0, y que necesita primero arreglar sus finanzas para poder ahorrar algo.

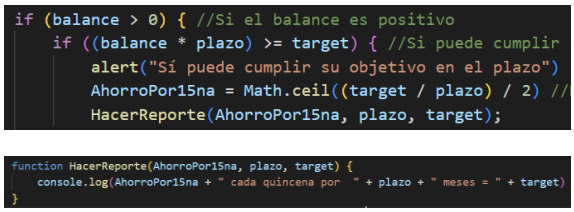
Desarrollo del segundo Sprint

Tendremos que corroborar que los datos se conviertan a número. Luego de hace la suma de todos los egresos y se saca el balance

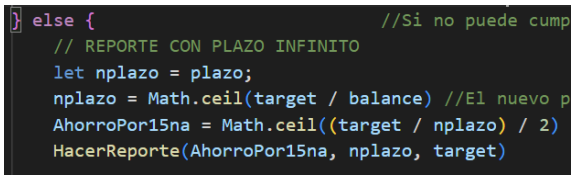
Utilicé una alerta solo para comprobar si el balance es positivo o negativo



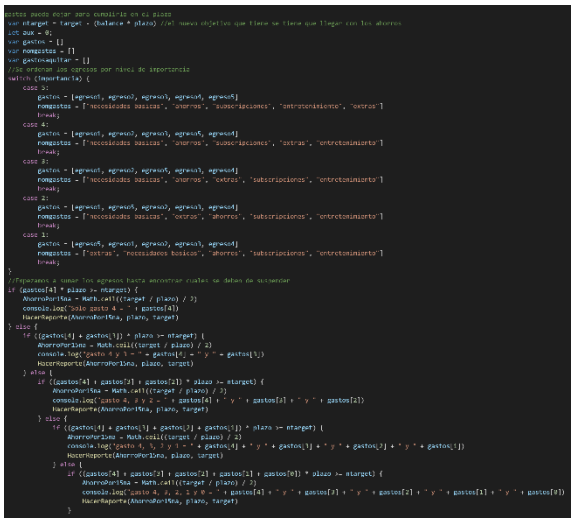
Para hacer el reporte, lo puse en una función para poderla llamar en todos los casos que se pueda hacer el reporte



Utilizamos nuevas variables para no borrar el plazo original

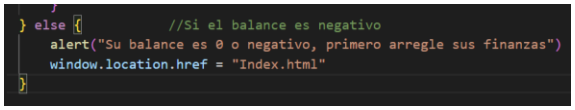


Acomodamos los valores en arreglos al igual que la descripción de estos valores, esto nos ayudará para después.



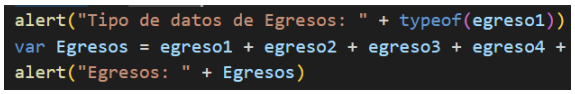
Este else está al final de todos los ifs anidados

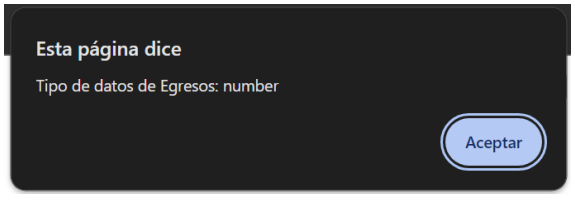
Else del primer if donde condicinabamos si su balance era positivo



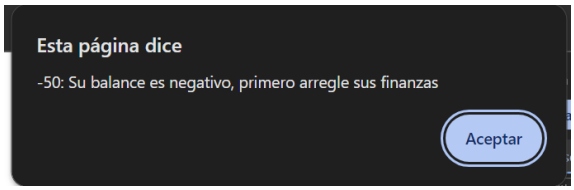
Revisión y retrospectiva del sprint

Para probar si funciona, utilizaremos alertas





Para la comprobación si el balance es positivo o negativo también utilicé unas alertas



Ahora haré una prueba para corroborar que los cálculos de cuando es posible hacer el reporte en tiempo con los siguientes valores:

Objetivo 5,000

Ingresos 2,200

Egresos 1,500 (totales)

Balance 700

Plazo 12

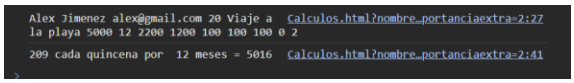
(objetivo 5,000) / (plazo 12) = 416.6667

> balance (sí alcanza)

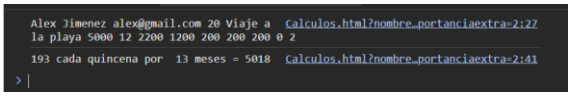
Entonces: 416.6667 / 2 = 208.33334

AhorroPor15na = 209 (redondeado)

Cantidad de dinero al final: 5016



Ahora para el plazo infinito



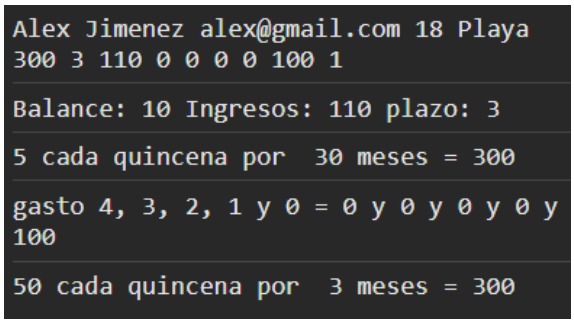
Para los siguientes ejemplos, los valores serán 300 para el objetivo a 3 meses, 110 de ingresos y 100 de egresos, el cual lo vamos a estar cambiando de importancia para corroborar que se estén sumando todos los egresos.

```
Alex Jimenez alex@gmail.com 18 Playa
300 3 110 0 0 0 0 100 5
Balance: 10 Ingresos: 110 plazo: 3
5 cada quincena por  30 meses = 300
Sólo gasto 4 = 100
50 cada quincena por  3 meses = 300
```

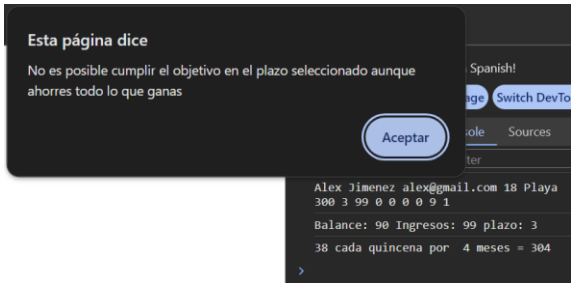
```
Alex Jimenez alex@gmail.com 18 Playa
300 3 110 0 0 0 0 100 4
Balance: 10 Ingresos: 110 plazo: 3
5 cada quincena por  30 meses = 300
gasto 4 y 3 = 0 y 100
50 cada quincena por  3 meses = 300
```

```
Alex Jimenez alex@gmail.com 18 Playa
300 3 110 0 0 0 0 100 3
Balance: 10 Ingresos: 110 plazo: 3
5 cada quincena por  30 meses = 300
gasto 4, 3 y 2 = 0 y 0 y 100
50 cada quincena por  3 meses = 300
```

```
Alex Jimenez alex@gmail.com 18 Playa
300 3 110 0 0 0 0 100 2
Balance: 10 Ingresos: 110 plazo: 3
5 cada quincena por  30 meses = 300
gasto 4, 3, 2 y 1 = 0 y 0 y 0 y 100
50 cada quincena por  3 meses = 300
```



Cuando no es posible cumplir el objetivo, aunque ahorre todo lo que gana, se muestra una alerta



Al igual que si su balance es negativo, no será posible hacer ningún ahorro, por lo tanto, se notificará y después de redirigirá a la página principal

Por lo visto, todas las posibilidades de los balances y cumplimiento de los plazos han sido condicionados correctamente, por lo tanto, podemos concluir que, al menos con los ejemplos que probé, funciona sin ningún problema.

Sólo un cambio que hubo es necesario que el usuario sea mayor de edad, por o tanto, en la planificación del siguiente sprint esto será incluido.

Planeación del tercer sprint

Mostrar el resultado de los cálculos en formato de un reporte.

Condicionar que sea mayor de edad.

Se debe de tener cuidado que se quiera mostrar algún dato que no exista, ya sea nulo o indefinido, sin embargo, si los datos fueron analizados correctamente (lo cual en teoría ya lo hace), no debería de haber ningún problema.

Para condicionar que el usuario es mayor de edad, leeremos la edad, si es menor de 18 años, le mostraremos una alerta para hacerle saber que no puede utilizar nuestro servicio, y lo redireccionará a la página Index.

Para mostrar los datos se hará una tabla en HTML, en el cual, se mostrarán todos los datos constantes que fueron introducidos, para que el usuario pueda corroborar que son los datos que él puso.

Cada diferente tipo de ahorro que pueda realizar el usuario, será adjuntado al final de la tabla como una fila diferente, esto para que queden visibles las diferentes opciones de ahorro que tiene.

Luego se modificará la función de HacerReporte() para que en lugar de mostrar los datos por consola, los muestre en el documento.

Al final de la tabla, como footer, se agregará una leyenda con un agradecimiento por utilizar la aplicación, al igual que los datos e información del dueño y responsable de la aplicación.

Desarrollo del tercer sprint

Para la verificación de mayor de edad simplemente se condiciona que sea mayor de 18 años.

```
<script>
    const PARAMETROS = new URLSearchParams(window.location.search);
    var edad = PARAMETROS.get("edad")
    edad = +edad
    if (edad < 18) {
        alert("Necesitas tener mas de 18 años para usar este programa")
        window.location.href = "Index.html";
    }
</script>
```

Luego para mostrar los datos:

```
<h1>PLAN DE AHORRO</h1>
<table align="center">
    <tr>
        <th colspan="2">Información del usuario</th>
    </tr>
    <tr>
        <td>Nombre:</td>
        <td id="nombre"></td>
    </tr>
    <tr>
        <td>Correo electrónico:</td>
        <td id="email"></td>
    </tr>
    <tr>
        <td>Edad:</td>
        <td id="edad"></td>
    </tr>
    <tr>
        <th colspan="2">Objetivo</th>
    </tr>
    <tr>
        <td id="objetivo"></td>
        <td id="$objetivo">$</td>
    </tr>
    <tr>
        <td>Plazo:</td>
        <td id="plazo"></td>
    </tr>
    <tr>
        <td><br></td>
    </tr>
    <tr>
        <td>Ingresos mensuales:</td>
        <td id="ingresos"></td>
    </tr>
    <tr>
        <td> <br></td>
    </tr>
    <tr>
        <td>Necesidades basicas:</td>
        <td id="e1"></td>
    </tr>
    <tr>
        <td>Ahorros Extras:</td>
        <td id="e2"></td>
    </tr>
    <tr>
        <td>Subscripciones:</td>
        <td id="e3"></td>
    </tr>
    <tr>
        <td>Entretenimiento:</td>
        <td id="e4"></td>
    </tr>
    <tr>
        <td>Gastos Extras:</td>
        <td id="e5"></td>
    </tr>
    <tr>
        <td><strong>Egresos totales:</strong></td>
        <td id="et"></td>
    </tr>
    <tr>
        <td></td>
    </tr>
</table>
```

Hice una tabla donde se especifican todos los datos introducidos, y el campo del valor, se le coloca un “id” donde pondremos el valor correspondiente.

```
document.getElementById("nombre").innerHTML = NOMBRE
document.getElementById("email").innerHTML = EMAIL
document.getElementById("edad").innerHTML = edad + " años"
document.getElementById("objetivo").innerHTML = OBJETIVO
document.getElementById("$objetivo").innerHTML = "$" + target
document.getElementById("plazo").innerHTML = plazo + " meses"
document.getElementById("ingresos").innerHTML = "$" + ingresos
document.getElementById("e1").innerHTML = "$" + egreso1
document.getElementById("e2").innerHTML = "$" + egreso2
document.getElementById("e3").innerHTML = "$" + egreso3
document.getElementById("e4").innerHTML = "$" + egreso4
document.getElementById("e5").innerHTML = "$" + egreso5
```

```
document.getElementById("et").innerHTML = "<strong>$" + Egresos + "</strong>"
```

Mostramos cada variable agregando símbolos y leyendas sobre los datos.

La tabla para mostrar los diferentes tipos de ahorro:

```
<table align="center" id="tabla" class="tabla">
    <tr>
        <th>
            <h2>Opciones de ahorro</h2>
        </th>
    </tr>
</table>
```

Para la tabla de las opciones de los planes de ahorro, como puede haber diversos planes, haremos la tabla con un id para poder ir insertando filas.

La funcion HacerReporte fue dividida en 3 funciones diferentes para poder dar un mensaje específicando los parametros que tenían que cambiar y el contexto en el que se encuentran.

```
if ((gastos[4] + gastos[3]) * plazo >= ntarget) {
    AhorroPor15na = Math.ceil((target / plazo) / 2)
    mensaje = nomgastos[4] + " y " + nomgastos[3] + " ($" + (gastos[4] + gastos[3]) + ") <br>"
    console.log("gasto 4 y 3 = " + gastos[4] + " y " + gastos[3])
    HacerReporte3(AhorroPor15na, plazo, target, mensaje
```

El unico HacerReporte que cambió fue el 3, el cual recibe un tercer parámetro con la lista de los egresos que debería de dejar para alcanzar el objetivo en el plazo deseado.

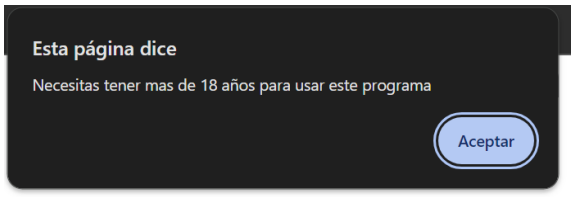
Para el footer

```
<footer>
    <p>Gracias por utilizar esta aplicación</p>
    <p>
        Alejandro Jiménez Jiménez <br>
        218817306 <br>
        ICOM <br>
        Ingeniería de Software <br>
    </p>
</footer>
```

Al final agregué un footer con los datos del equipo del trabajo (yo) junto con un agradecimiento.

Revisión del tercer sprint

Revisamos que funcione el condicionamiento para que sea mayor de edad

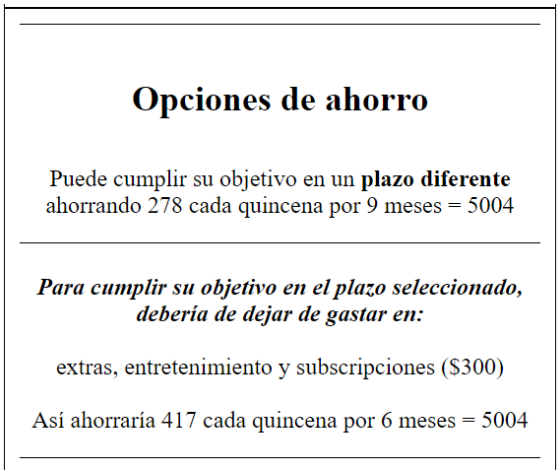


Revisamos la tabla donde se muestran los datos previamente introducidos

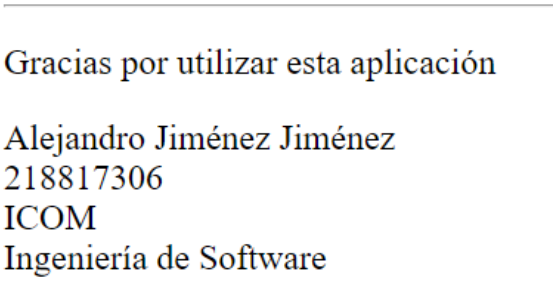
PLAN DE AHORRO

Información del usuario	
Nombre:	Alejandro Jimenez
Correo electrónico:	alex@gmail.com
Edad:	18 años
Objetivo	
Playa	$5000
Plazo:	6 meses
Ingresos mensuales:	$1200
Necesidades basicas:	$200
Ahorros Extras:	$100
Subscripciones:	$100
Entretenimiento:	$200
Gastos Extras:	$0
Egresos totales:	**$600**

Revisamos que sí se muestren los planes de ahorros generados, al igual que los cálculos sean correctos a los generados



Y el footer con los datos



Por lo visto con las pruebas, el programa funciona correctamente, se a hecho un recuento de los requisitos, los cuales han sido satisfechos en su totalidad, cualquier cambio o requisito extra tendrá que pasar por el proceso previo fijando un costo extra y fecha de entrega nueva.

Conclusiones

La implementación de la metodología Scrum en un proyecto de software mejora significativamente la colaboración y la comunicación del equipo, proporciona flexibilidad y adaptabilidad ante cambios, aumenta la transparencia y el control del progreso, asegura la entrega continua de valor y fomenta la mejora continua a través de retrospectivas periódicas. Estos beneficios contribuyen a un desarrollo más eficiente y alineado con las expectativas del cliente, resultando en un producto final de mayor calidad y un entorno de trabajo más dinámico y colaborativo.

Referencias

Atlassian. (n.d.). *¿Qué es scrum? [+ Cómo empezar] | Atlassian*.

https://www.atlassian.com/es/agile/scrum