



Objetos literales

Como vimos, los objetos son unas de las estructuras más importantes de la programación, tanto que hay toda un área en función de ellos. Para entender un poco más sobre por qué son tan útiles e importantes, vamos a realizar una serie de prácticas para afianzar mejor las particularidades y utilidades que vienen con ellos. Para esta actividad, recordá que podés verificar cada punto ejecutando los métodos y las operaciones que realices para asegurarte de que todo funcione correctamente.

Descripción del problema

Desde un banco nos contactan para crear una aplicación que pueda facilitar el manejo de información y las acciones que el mismo necesita. Nuestro tech leader nos facilita la forma de representar a los usuarios (cuentas bancarias). Cada una de estas cuentas tiene la siguiente información:

- N.º de cuenta (solo números).
- Tipo de cuenta (cuenta corriente en pesos o caja de ahorro en pesos).
- Saldo en pesos (solo la cantidad).
- Titular de la cuenta (nombre completo).

Teniendo en cuenta esta información, se nos pide que hagamos lo siguiente:



1. Descargar y analizar el siguiente [archivo](#), el cual contiene la lista de clientes (cuentas bancarias).
2. Nos solicitan también crear un objeto literal llamado “banco”, el cual tendrá una propiedad llamada “clientes”, que estará compuesta de la lista de objetos obtenidos en el punto anterior.
3. Al objeto “banco”, crearle un método llamado *consultarCliente*, el cual recibirá un nombre (titular) por parámetro, deberá buscarlo en la lista de cuentas y retornar el objeto cliente que corresponda con ese nombre ingresado.

ayuda: let clienteEncontrado = banco.consultarCliente("Ramon Connell");

4. Crear otro método llamado *depósito*, que recibirá dos parámetros, el titular de cuenta y una cantidad de dinero a depositar. El método debe obtener a la cuenta correspondiente y luego sumar la cantidad de dinero a depositar a *saldo en pesos*. Luego, deberá dar un aviso por la consola con el mensaje “Depósito realizado, su nuevo saldo es: xxx”.
5. Crear un último método llamado *extracción*, que recibirá también dos parámetros: el titular de cuenta y el monto a extraer. El método debe obtener la cuenta correspondiente y restar el monto al saldo actual. En caso de que el resultado sea menor a 0, deberá imprimir un mensaje por la consola de “Fondos insuficientes”, de lo contrario deberá imprimir “Extracción realizada correctamente, su nuevo saldo es: xxx”.

Si llegaste hasta acá, felicidades, el equipo de desarrollo y el tech leader están impresionados con tu trabajo.



Bonus extra

Para que no te quedes con las ganas y puedas seguir practicando si así lo deseas, te proponemos algunos ejercicios más. Tené en cuenta que a partir de acá los ejercicios pueden escalar en dificultad. Como siempre decimos, paciencia, ignorá la complejidad y tratá de resolverlo con las herramientas que tengas a tu disposición. También podés buscar información extra en Google o documentaciones que conozcas.

Propiedad única

Crear una función llamada *propiedadUnica*, que reciba un arreglo de objetos como parámetro y un string. Esta deberá retornar un nuevo arreglo de objetos, teniendo como parámetro la propiedad que fue pasada como string.

Ejemplo:

```
let array = [{ nombre: "Lean", edad: 27 }, { nombre: "Eze", edad: 49 }]
```

```
propiedadUnica(array, "edad") debe retornar [ { edad: 27 }, { edad: 49 } ]
```

```
propiedadUnica(array, "nombre") debe retornar [ { nombre: "Lean" }, { nombre: "Eze" } ]
```

Calculador de notas

Crear el objeto "alumno", el cual va a consistir en las siguientes propiedades básicas:

- Nombre
- Número de legajo
- Lista de notas

Nos gustaría calcular el promedio del alumno y si el mismo está aprobado, basado en una nota de aprobación que le va a ser dada. Para este ejercicio, vamos a dejar que



pienses todos los métodos y propiedades que puedan hacer falta para que el programa funcione correctamente de la manera solicitada.