

Guía 3: Completando el Home Banking

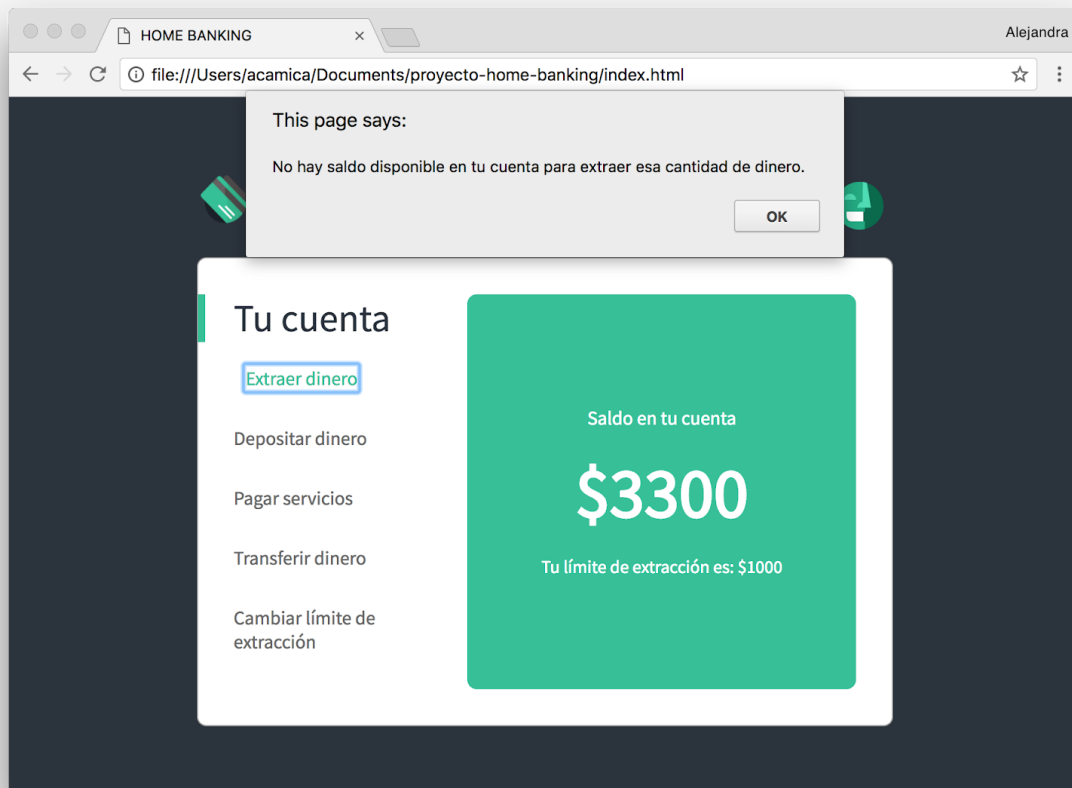
Paso 1: Limitá la extracción de dinero

En este paso vas a modificar la funcionalidad de extracción para que realice algunas verificaciones antes de entregar el dinero. Vas a tener que:

- Verificar que el dinero que se quiere extraer no sea mayor al dinero que hay en la cuenta.
- Verificar que el dinero que se quiere extraer no supere el límite de extracción.
- Como el Home Banking maneja solo billetes de 100, verificar que el monto que se quiere extraer pueda ser entregado solo con esos billetes.

A. Saldo de la cuenta

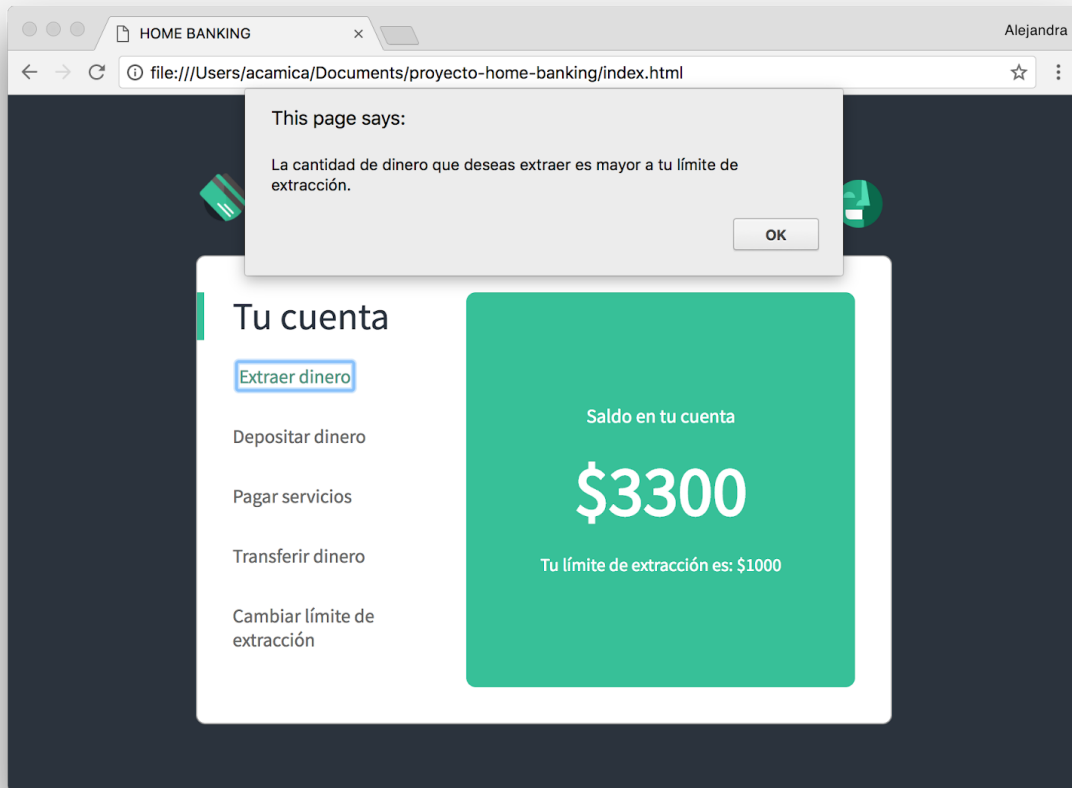
Agregá una validación que, en el caso que el usuario quiera extraer más dinero del que hay en la cuenta, muestre una alerta como la siguiente.



Cuando termines, tu Home Banking no debe permitir que retires más dinero del que hay en tu cuenta. Probá esta funcionalidad y verificá que funcione correctamente. Primero intentá sacar menos dinero del que hay la cuenta, el Home Banking debe permitir extraerlo. Luego, intentá sacar un monto mayor al saldo de la cuenta y chequeá que se muestre una alerta de error y que no se descuente dinero de la cuenta.

B. Límite de extracción

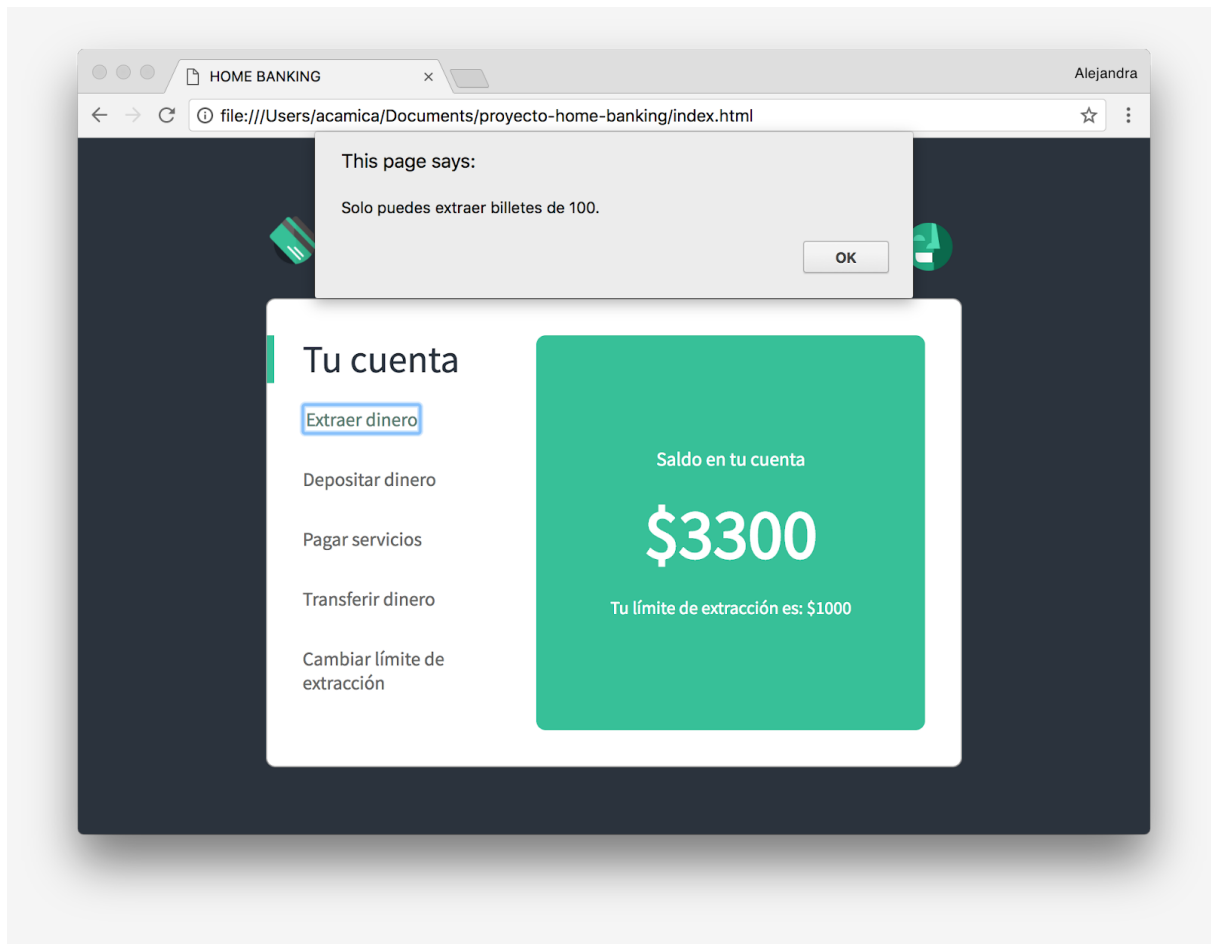
Agregá una segunda validación que, en el caso de que el usuario quiera extraer un monto superior a su límite de extracción, muestre una alerta como la siguiente:



Cuando agregues esta validación, intentá extraer un monto mayor al límite de extracción. El Home Banking debería mostrar una alerta y no debería entregar el dinero.

C. Billetes de 100

Este Home Banking solo entrega billetes de 100, por lo tanto, no se pueden extraer montos como \$150, \$210 o \$335. Agregá una última validación que verifique si el monto que se quiere extraer puede ser entregado solo por billetes de 100. Si no es posible, se debe mostrar una alerta como la siguiente:



Pista: Recordá la función módulo, que devolvía el resto de realizar una división entre dos números. ¿Se te ocurre cómo aplicarla para verificar si el monto que quiere extraerse puede ser entregado con billetes de 100?

Cuando termines de agregar esta última validación, la funcionalidad de extraer dinero está terminada. Para chequear que funcione correctamente, intentá todos los casos y fijate que se me muestre la alerta que corresponde.

Buenas prácticas: La función que se encarga de la extracción de dinero va a ser extensa y va a contener muchas validaciones. Una buena idea es crear funciones más pequeñas que devuelvan un valor boleano y luego consultarlo desde la sentencia `if`. Por ejemplo, podrías crear una función `haySaldoDisponible()` que retorne verdadero o falso, según corresponda. Luego, en la sentencia `if` de la función principal, podes consultar por ese valor y realizar la acción que corresponda.

Paso 2: Desarrollá la funcionalidad de pago de servicios

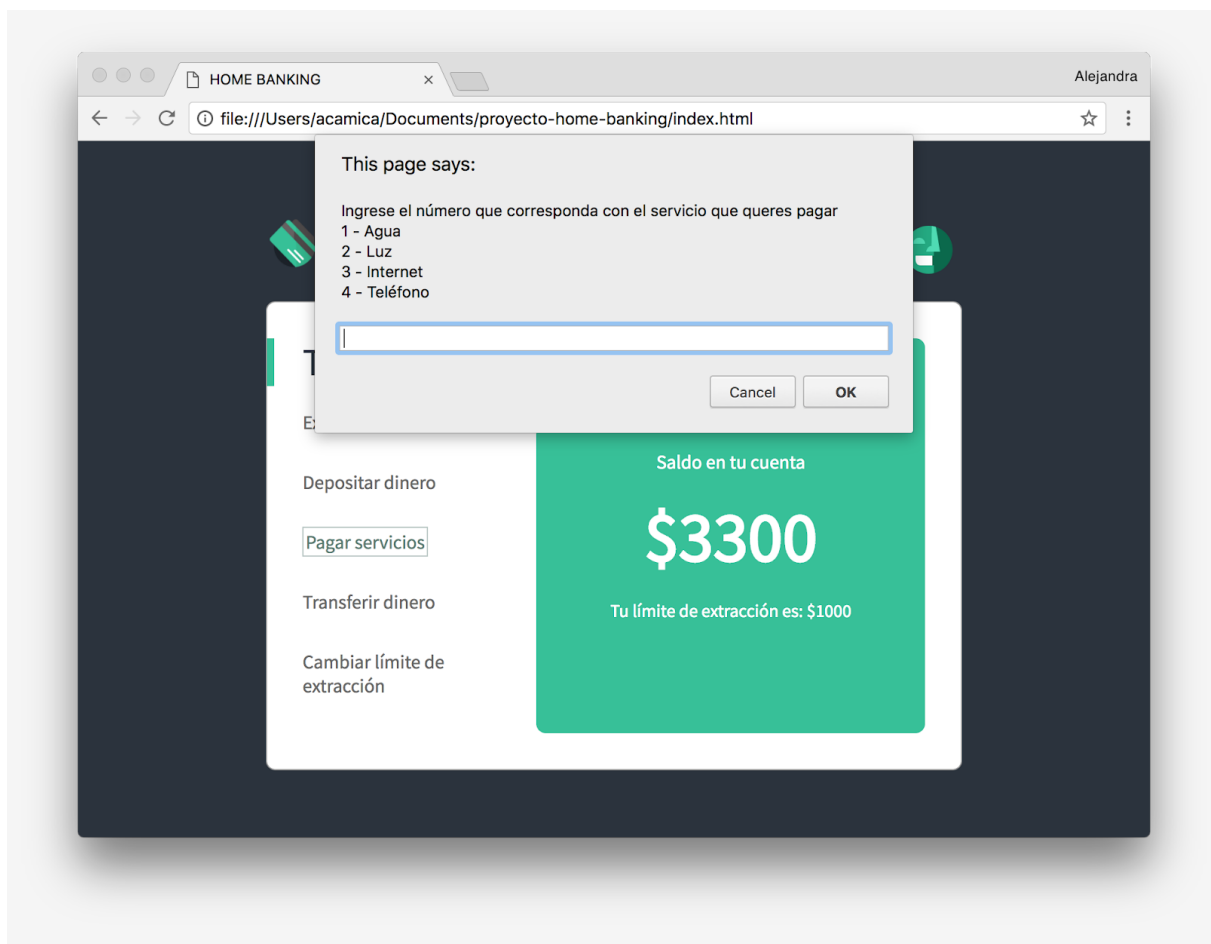
En este paso, vas a crear la funcionalidad de pago de servicios, permitiendo que el usuario pueda seleccionar un servicio de un listado y elija cuál pagar.

Primero, comenzá declarando las variables que contienen los precios de los servicios. En esta tabla se muestran los precios de los servicios:

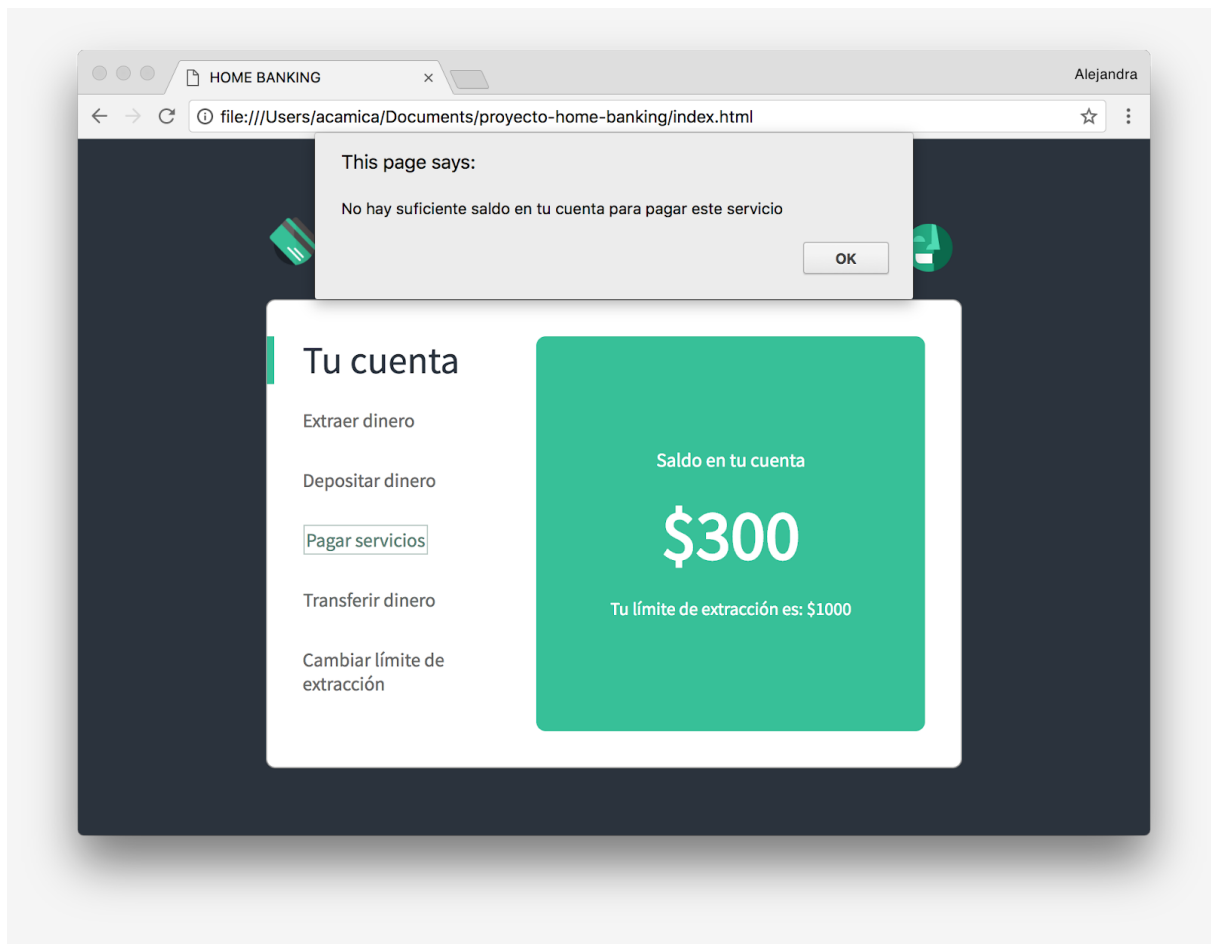
Servicio	Precio
Agua	350
Teléfono	425
Luz	210
Internet	570

Cuando termines de crear las variables, comenzá con el desarrollo de tu función. Los pasos que debe seguir son:

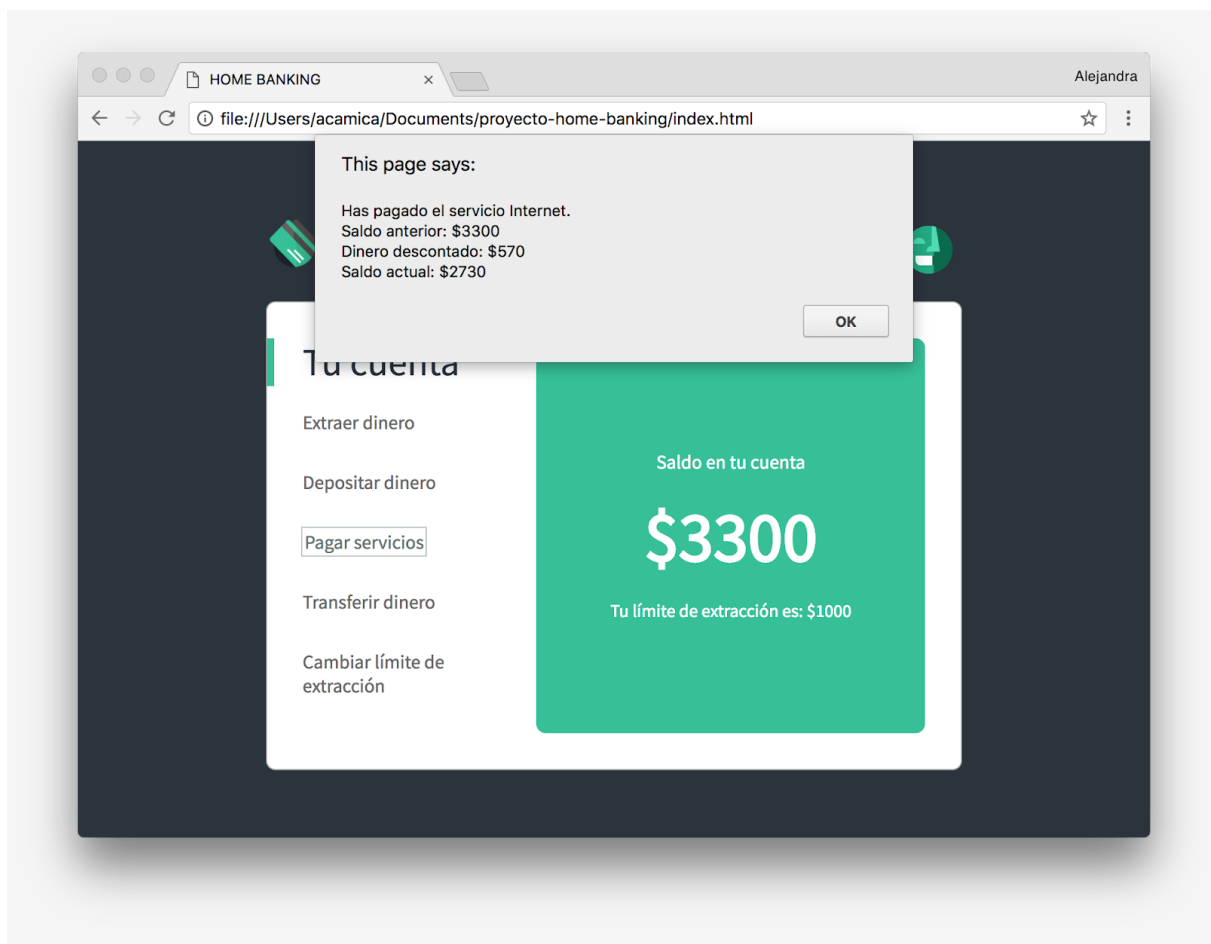
- Mostrar una alerta como la siguiente para que el usuario pueda elegir el número que se corresponde con el servicio que quiere pagar.



- Utilizar la sentencia `switch` determinando la acción que va a realizarse según la opción que elija el usuario. Por ejemplo, si elige la opción 1, se debe realizar el pago del servicio “Agua”. Si el usuario elige una opción que no se encuentra en el listado, se debe mostrar una alerta diciendo que no existe el servicio que se ha seleccionado.
- Verificar que el usuario tenga en su cuenta el dinero suficiente para pagar el servicio que seleccionó.
- Si no posee el dinero que se necesita para pagar el servicio, se debe mostrar una alerta como la siguiente:



- Si hay dinero suficiente en la cuenta para pagar el servicio se debe descontar el dinero correspondiente de la cuenta y mostrar una alerta con la siguiente información:



- Actualizar el nuevo saldo en el HTML.

Una vez que termines de crear esta funcionalidad, comenzá a realizar pagos de servicios verificando que todo funcione correctamente.

Paso 3: Desarrollá la funcionalidad de transferencia del dinero

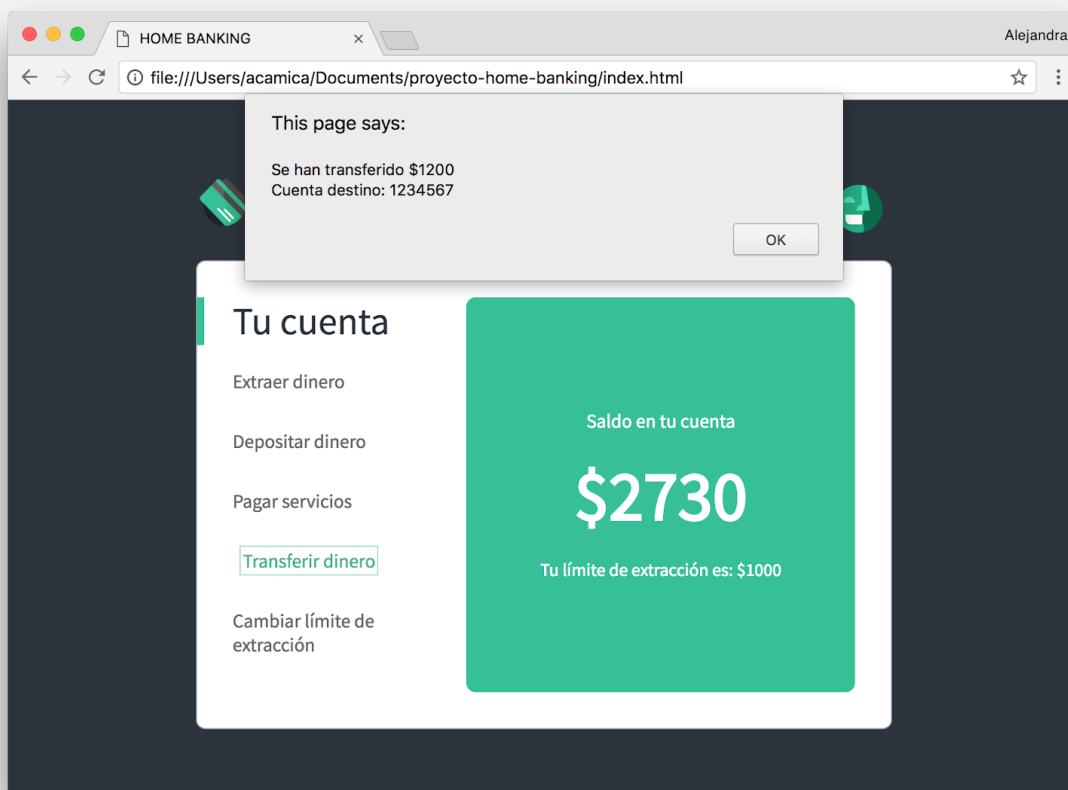
Creá la funcionalidad de transferencia, permitiendo que el usuario transfiera cierto monto de dinero a alguna de sus cuentas amigas.

Primero comenzá declarando las siguientes variables que contienen las cuentas amigas a las que se puede transferir dinero:

Nombre	Número de Cuenta
Cuenta amiga 1	1234567
Cuenta amiga 2	7654321

Luego completá la función `transferirDinero()` . Debe seguir los siguientes pasos:

- Pedirle al usuario que ingrese el monto que desea transferir
- Verificar que el monto que se quiere transferir se encuentre disponible en la cuenta. Si no hay saldo suficiente, se debe mostrar una alerta diciendo que no puede transferirse esa cantidad de dinero.
- Si hay saldo disponible, se debe continuar pidiéndole al usuario que ingrese el número de cuenta al que desea transferir el dinero.
- Verificar que la cuenta ingresada sea alguna de las cuentas amigas (variables declaradas anteriormente). Si la cuenta que el usuario ingresó no es una de las cuentas amigas, mostrar una alerta diciendo que solo puede transferirse dinero a una cuenta amiga.
- Si no hubo ningún problema con la transferencia (había dinero en la cuenta y se ingresó una cuenta amiga correcta) se debe descontar el dinero que se desea transferir de la cuenta.
- Actualizar el nuevo saldo en el HTML.
- Mostrar una alerta como la siguiente:



Una vez que finalices tu función, comenzá a realizar transferencias verificando que todo funcione correctamente.

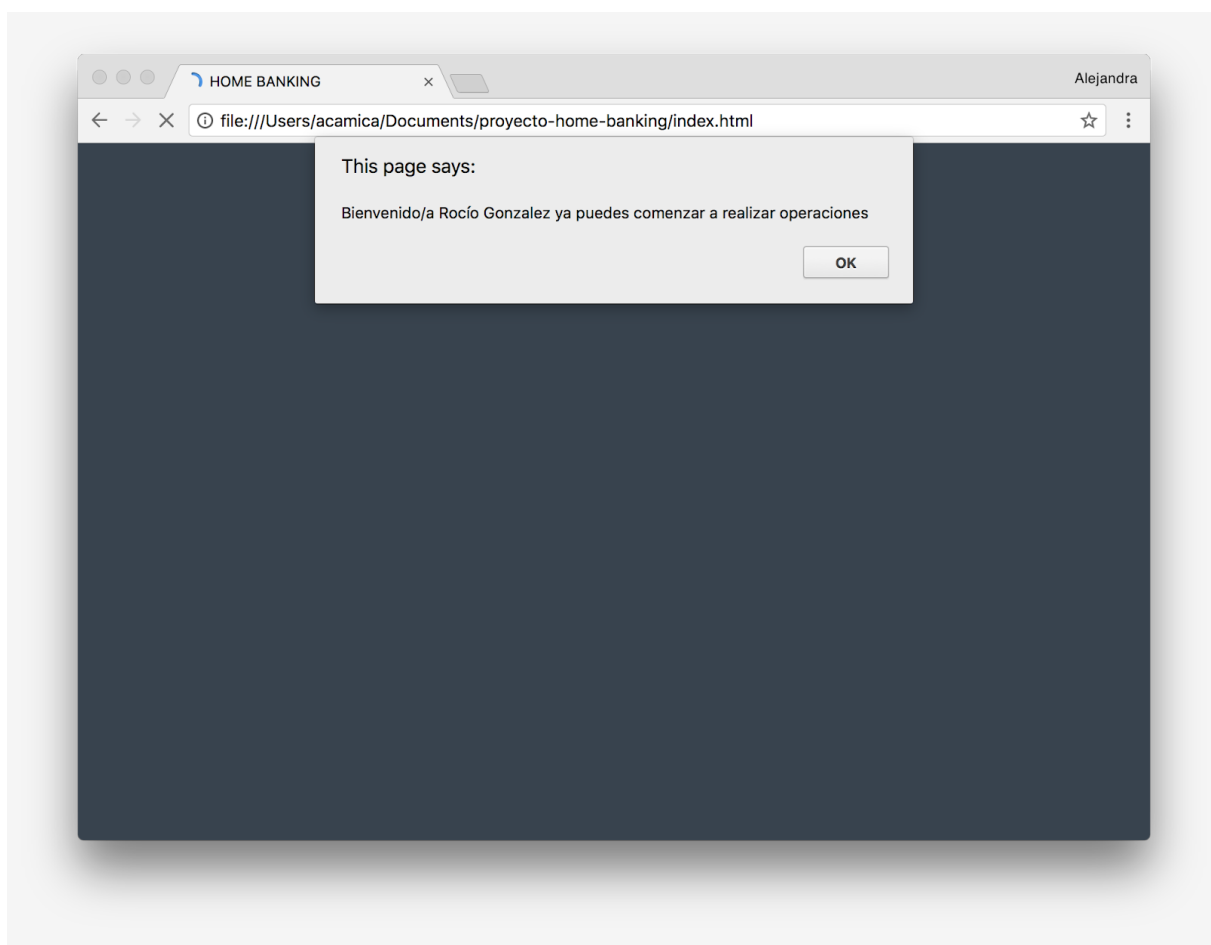
Paso 4: Agregá un inicio de sesión a tu Home Banking

El paso final es agregarle un inicio de sesión a tu Home Banking. Cada vez que se inicie la aplicación, el usuario debe ingresar su código de verificación. Si es correcto puede comenzar a utilizar la aplicación. En caso de que el código ingresado sea incorrecto, el Home Banking va a retener todo el dinero de la cuenta por cuestiones de seguridad (la cuenta del usuario va a quedar con \$0).

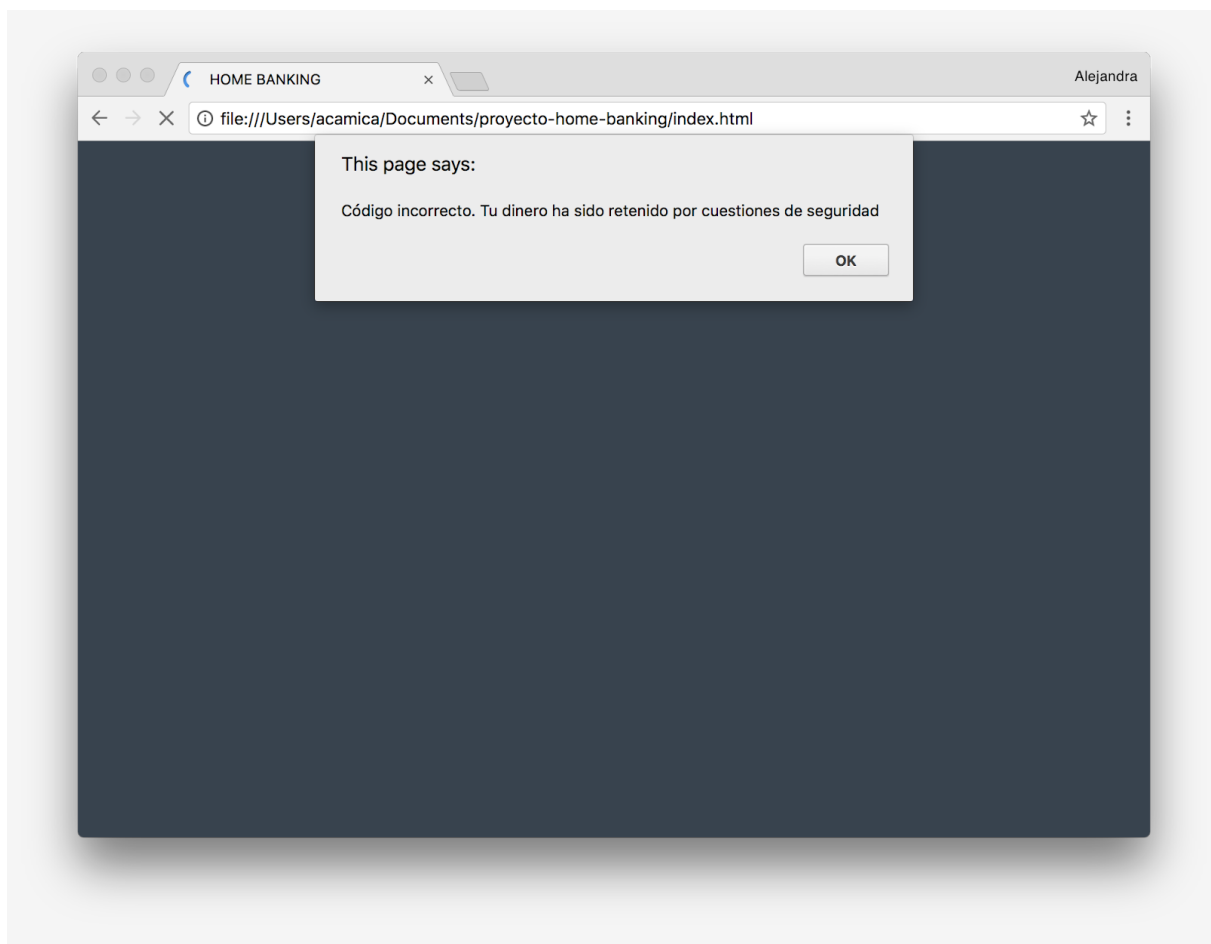
Comenzá este paso declarando una variable con el código de seguridad de tu cuenta. Puede ser el código que vos quieras. En general, son de 4 dígitos numéricos.

Completá la función `iniciarSesion()`. Esta función debe:

- Pedirle al usuario que ingrese el código de su cuenta.
- Verificar que el código ingresado sea correcto.
- Si el código es correcto, se debe mostrar una alerta como la siguiente:



- Si el código ingresado es incorrecto, se debe retener todo el dinero de la cuenta (dejar la cuenta con \$0) y mostrar una alerta como la siguiente:



- Ejecutá esta función en el archivo JS para que cada vez que se ingrese a la aplicación, el usuario tenga que realizar el inicio de sesión.

Cuando finalices, refrescá tu Home Banking. Debería pedirte que inicies sesión. Probá que pasa si ingresas el código correcto y también qué pasa si ingresas mal el código.

Cómo seguir

¿Querés seguir agregándole funcionalidades nuevas a tu Home Banking? Acá vas a encontrar algunas ideas.

Verificá que las respuestas del usuario no sean nulas o vacías

¿Probaste qué pasa cuando en alguna alerta que muestra el Home Banking haces click en el botón cancelar? Hay un problema: cuando se hace click en el botón cancelar, el valor que ingresa el usuario es está vacío. Sin embargo, tu programa no se da cuenta de eso y la función sigue ejecutando, intentando sumar al saldo de la cuenta una variable `null`, es decir, nula.

Lo mismo ocurre cuando el usuario no ingresa ningún valor y hace click en el botón “Ok”. Se intenta sumar una variable vacía al saldo de la cuenta.

Una opción para mejorar tu Home Banking es agregar una validación que verifique que la respuesta no esté vacía y que no sea nula. Si el usuario no ingresa ningún valor o hace click en el botón cancelar, la función debe finalizar.

Si realizas esta modificación, utilizá la consola para ayudarte. Vas a poder ver el valor de la variable que guarda la respuesta del usuario en cada caso.

Verificá que las respuestas ingresadas sean valores numéricos

¿Probaste qué pasa si en vez de ingresar un valor numérico en las alertas, ingresás una letra? Ocurre un problema: tu aplicación está intentando sumar un número con un string. Esto genera un error ya que el operador `+` aplicado a dos números implica una suma pero aplicado a un string, implica concatenación. Al ser dos tipos de datos distintos, JS no sabe como resolverlo.

Para mejorar esta cuestión, puedes agregar un paso que verifique que la variable ingresada sea de tipo numérico. Investigá la función `typeof` que te va a ayudar a resolver este problema.

Agregá nuevas funcionalidades al Home Banking

Podes mejorar tu Home Banking agregandole nuevas funcionalidades como por ejemplo, el depósito de cheques. Este es solo un ejemplo, pero podés agregarle lo que vos quieras.