

1. Descripción general del proyecto: En este proyecto tendrás que construir una aplicación web que le permita a un usuario **validar el número de una tarjeta de crédito**. Además, tendrás que implementar funcionalidad para **ocultar todos los dígitos** de una tarjeta **menos los últimos cuatro** cuando se le indique al usuario si su tarjeta es válida o no. La temática es libre: qué pantallas, explicaciones, mensajes, colores, marca, etc. Puedes guiarte del siguiente ejemplo:



The image shows a web interface for Wannapay, a credit card validation service. The interface is split into two main sections. The left section has a yellow background and contains the Wannapay logo (a yellow 'W' in a white square) and the tagline "Valida tu tarjeta de crédito. Rápido. Fácil. Seguro." Below this, the name "Wannapay" is displayed in bold, followed by a description: "Wannapay es una empresa internacional de verificación de tarjetas de crédito. Ingresar los dígitos de tu tarjeta, validala e instantáneamente tienes acceso a miles de sitios de compras por internet en todo el mundo." The right section has a white background and contains the prompt "Ingresa los dígitos de tu tarjeta de crédito" above a text input field. Below the input field is a dark gray button labeled "VERIFICAR".



The image shows a web interface displaying an error message. At the top center is a large red 'X' icon. Below the icon, the text reads: "Tu tarjeta #####0512 no ha sido reconocida. ¿Quieres intentarlo de nuevo?". At the bottom center is a dark gray button labeled "Intentar de nuevo".

2. Requisitos de entrega:

- a. Debe tener una página o vista de inicio donde le expliques al usuario para qué es esta web app. Puedes incluir una breve descripción de tu marca, slogan, instrucciones, etc.
- b. En esa misma vista de inicio, debe haber un input donde el usuario pueda ingresar los dígitos de su tarjeta y un botón para el envío de la data.
- c. Se deben mostrar mensajes de error al usuario si:
 - i. Intenta ingresar caracteres distintos a números.
 - ii. Intenta ingresar el campo vacío.
- d. Mostrar una pantalla con mensajes diferentes si:
 - i. La tarjeta es válida.
 - ii. La tarjeta no es válida. Adicionalmente, si la tarjeta no es válida, agrega un botón que redirija al usuario a la pantalla de inicio para darle la opción de intentarlo de nuevo, tal como se muestra en el ejemplo.

3. Objetivos de aprendizaje:

El objetivo principal de aprendizaje es adquirir experiencia desarrollando aplicaciones web que interactúen con el usuario a través del navegador y la lógica, utilizando HTML, CSS y JavaScript como herramientas. Además de los anteriores requisitos de entrega, debes cumplir los siguientes objetivos con dichas herramientas. A lo largo del proyecto, reflexiona individualmente sobre los objetivos individuales que has cumplido o que te faltan:

HTML y CSS:

- Uso correcto de HTML semántico.
- Uso de selectores de CSS.

DOM:

- Entender y reconocer los selectores del DOM.
- Manejar eventos del DOM.
- Manipular dinámicamente el DOM.

Javascript:

- Manipular strings.
- Uso de condicionales.
- Uso de bucles.
- Uso de funciones.

4. Recursos sobre el [algoritmo de Luhn](#): también llamado algoritmo de módulo 10, es un método de suma de verificación, se utiliza para validar números de identificación; tales como el IMEI de los celulares, tarjetas de crédito, etc.

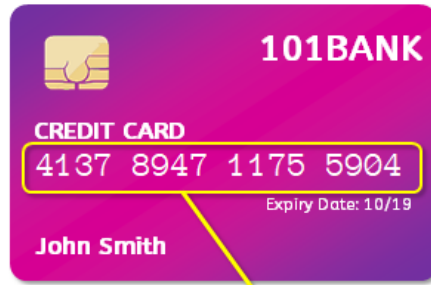
Este algoritmo es simple. Obtenemos la reversa del número a verificar (que solamente contiene dígitos [0-9]); a todos los números que ocupan una posición par se les debe multiplicar por dos, si este número es mayor o igual a 10, debemos sumar los dígitos del resultado; el número a verificar será válido si la suma de sus dígitos finales es un múltiplo de 10.

Aquí hay un artículo muy interesante: <https://www.quobit.mx/asi-funciona-el-algoritmo-de-luhn-para-generar-numeros-de-tarjetas-de-credito.html>

Y aquí, un video muy útil por si tu estilo de aprendizaje es más visual:

https://www.youtube.com/watch?v=f0zL6Ot9y_w&t=0s

Finalmente, en la siguiente página te dejamos una pequeña infografía para que puedas ver el paso a paso:



4	1	3	7	8	9	4	7	1	1	7	5	5	9	0	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



Step 1: Double the value of every second digit.

8	1	6	7	16	9	8	7	2	1	14	5	10	9	0	4
---	---	---	---	----	---	---	---	---	---	----	---	----	---	---	---



Step 2: If the result of this doubling operation is greater than 9 (e.g. 16), then add the digits of the product (e.g. $1 + 6 = 7$).

8	1	6	7	7	9	8	7	2	1	5	5	1	9	0	4
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



Step 3: Take the sum of all the digits.

$$8 + 1 + 6 + 7 + 7 + 9 + 8 + 7 + 1 + 2 + 5 + 5 + 1 + 9 + 0 + 4 = 80$$



Step 4: If the total ends in zero, this is a valid card number, If not, it is an invalid card number.

80 ends with a **0** ➡ **Valid Card Number**

5. ¿Cómo será evaluado? Se espera que cada miércoles, puedas compartir tu pantalla y mostrar tu avance del proyecto a tus compañeros así como pedir feedback o compartir tips sobre cómo resolviste algo del proyecto. Entre semana se dejará un link a sli.do donde podrán escribir preguntas de forma anónima y votar por las que más le interese resolver en el taller grupal del viernes. Al finalizar tu proyecto por completo, tendrás code review individual con uno de los profesores para revisar los requisitos de entrega y los objetivos de aprendizaje individuales.
6. ¿Qué otros tips me recomiendan?
 - a. No te quedes únicamente con la información de este README. Si algo no te queda claro, busca otros recursos en internet! Vivimos en una época dorada donde, si un tutorial no te satisface, puedes buscar un artículo o un foro en stackOverflow. Así como en este README te hemos dado tres recursos diferentes sobre el algoritmo

de Luhn, hay miles de recursos para developers y gente que empieza a aprender a programar; no los desaproveches!

- b. Pide ayuda a tus compañeros. La mayoría de veces varias personas tenemos las mismas dudas, y en eso alguien encuentra una manera muy ingeniosa de resolverlas. Si por más que ya intentaste, aún no encuentras una respuesta, coméntale a tus compañeros qué has hecho y tal vez ellos te pueden ayudar a ver tu error o darte tips para cómo resolver tu problema.
- c. Ayuda a tus compañeros. Existe mucha evidencia científica de que compartir tus conocimientos ayuda a consolidarlos. De igual manera, si tú no sabes algo, puedes intentar buscar junto a tus compañeros la solución, o incluso probar hacer un poco de pair programming! (Pro tip: busca la extensión de Live Share para Visual Studio Code).
- d. Reemplaza la frustración por curiosidad. Siempre que el código no te salga (y créeme, eso pasará mucho xD) no te lo tomes de manera personal. La computadora no quiere herir tus sentimientos, y tampoco quiere decir que no seas bueno para esto. Para ser programador hace falta mucha paciencia y cuando algo no sale, lo más importante es tener mucha curiosidad y querer encontrar el **por qué**: A ver, esta línea hace esto, este argumento viene de esta otra línea, esta variable tiene este valor, y así... como cuando nos toca desenredar nuestros audífonos!
- e. Tómate un descanso. Existen técnicas de productividad como Pomodoro, que son útiles para muchas cosas más que sólo escribir código, y el cerebro funciona mejor cuando nos tomamos un descanso, porque además nos permite asimilar toda la información que acabamos de aprender.
- f. Conoce tu estilo de aprendizaje. Si eres más visual, mira un tutorial. Si eres más pragmático, abre Visual Studio, empieza a escribir código y observa detenidamente qué pasa al correrlo. Si eres más lingüístico, lee un artículo y trata de explicarlo o escribirlo en tus propias palabras. Hasta puedes abrir un blog o canal de youtube sobre todo lo que vas aprendiendo en tu camino como developer, ayudando a otros como tú y llegar a ser un influencer. Nunca se sabe...!
- g. Pregunta siempre. No dejes de preguntar a tus amigos y profesores. Aunque te dé pena, recuerda que nadie aprendió de la noche a la mañana, que esta carrera requiere mucha dedicación, pero sobre todo, que muchas veces es una actividad en la que se disfruta, se aprende más y muchas veces toca: trabajar en equipo. Ni siquiera los profesores tienen todas las respuestas. Prepárate para googlear mucho.

