Chapter 1

**Exercises**

• Answer the following questions:

• Why is it so important to spend time on your user requirements?

• What is the difference between design requirements and an algorithm?

• What is the difference between a method and a property?

• What is a bug?

• What is state?

• Write an algorithm for how a soda machine works from the time a coin is inserted

until a soda is dispensed. Assume the price of a soda is 80 cents.

• Write the design requirements for an app that will run the soda machine.

Responde las siguientes preguntas:

• ¿Por qué es tan importante dedicar tiempo a los requisitos de sus usuarios?

Es importante conocer y analizar los requerimientos de los usuarios para poder desarrollar aplicaciones conforme a las exigencias de los usuarios.

• ¿Cuál es la diferencia entre los requisitos de diseño y un algoritmo?

Como **algoritmo** denominamos un conjunto ordenado y finito de operaciones simples a través del cual podemos hallar la solución a un problema.

En cambio, los **requisitos** **de diseño** son peticiones que el usuario o cliente desea que tengan su aplicación.

• ¿Cuál es la diferencia entre un método y una propiedad?

una **propiedad** es una característica del objeto mientras que un **método** es una acción.

• ¿Qué es un error?

(también conocido por el inglés ***bug***) es un problema en un programa de computador o sistema de [*software*](https://es.wikipedia.org/wiki/Software) que desencadena un resultado indeseado.

• ¿Qué es el estado?

un **estado** es una configuración única de información en un programa o máquina.

Llamamos estado de un objeto al conjunto de valores de los atributos del objeto en un momento dado.

• Escriba un algoritmo sobre cómo funciona una máquina de refrescos desde el momento en que se inserta una moneda

hasta que se dispense un refresco. Suponga que el precio de un refresco es de 80 centavos.

• Escriba los requisitos de diseño para una aplicación que ejecutará la máquina de refrescos.

Requerimientos

crear una aplicacion que le permita al usuario interactuar con la maquina de Refrescos.

Interfaz de usuario sencilla facil de usar.

La app debe Permitir Reconocer el ingreso de monedas.

Debe permitir verificar lo que el usuario requiera ya sea jugos naturales, gaseosa o Licuados.

Debe verificar que la moneda sea de un dólar.

debe cobrar los 80 centavos que es el precio del refresco y devolver los 20 centavos al usuario.

Sacar el producto seleccionado.

Chapter 2.

**EXERCISE**

Extend your playground by adding a line of code that prints any text of your choosing.

Amplíe su área de playground agregando una línea de código que imprima cualquier texto que elija.

// EJERCICIO 1 CAPITULO 2

import UIKit

var str = "Hello, playground"

str = "Hello World"

print(str)

// agregando una linea de codigo que imprima un texto.

var mensaje = "Completando el ejercicio 1 del Capitulo 2"

print("Imprimiendo mensaje: \(mensaje)")

Chapter 3.

**EXERCISES**

1• Write code within a Swift playground that multiplies two integers and displays the result.

Escriba código dentro de Swift playground que multiplique dos números enteros y muestre el resultado.

// EJERCICIO 1 CAPITULO 3

import UIKit

// Elaborar un programa que multiplique dos numeros enteros y muestre el resultado

var num1:Int=0

var num2:Int=0

var resultado=0

num1=7

num2=2

resultado=num1\*num2

print("El resultado de la Multiplicacion es: \(resultado)")

2• Write code within a Swift playground that squares a float. Display the resulting float.

Escriba código dentro de un patio de recreo Swift que cuadre un flotador. Muestre el flotador resultante.

// EJERCICIO 2 CAPITULO 3

//Ejercicio 2 del Capítulo 3

import UIKit

// Elaborar un programa que cuadre un valor Flotante y muestre el resultado en float

// usando tipo de datos Float.

var v1:Float = 5.5

var cuadrado:Float = 0

cuadrado = v1\*v1

print("Resultado del valor float al cuadrado = \(cuadrado)")

3• Write code within a Swift playground that subtracts two floats, with the result being

stored as an integer. Note that rounding does not occur.

Escriba código dentro de un patio de recreo de Swift que reste dos flotadores, y el resultado sea almacenado como un número entero. Tenga en cuenta que no se produce redondeo.