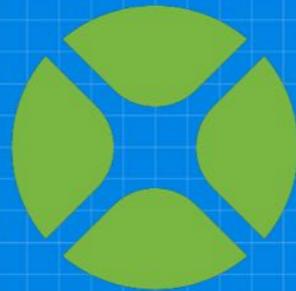


4TH EDITION

INTRODUCTION
TO PROGRAMMING
WITH XOJO



**Teacher
Guide**

RHINE, LEFEBVRE, PERLMAN

Introducción



Introducción

Bienvenido a la Guía del Profesor para el libro [Introduction to Programming with Xojo](#). Se ha diseñado esta guía para ayudarte a la hora de enseñar a tus alumnas cómo programar, incluso si tu mismo tienes una experiencia limitada.

Para cada uno de los capítulos de *Introducción a la Programación con Xojo*, encontrarás un capítulo asociado en esta guía. Los capítulos en esta guía están divididos en cuatro secciones:

CONCEPTOS Y VOCABULARIO

No encontrarás aquí “definiciones de diccionario”, pero esta sección te presentará las ideas básicas y términos del capítulo que has de conocer. Cuando es posible, los términos se explican en castellano plano.

ENLACES Y REFERENCIAS

Cuando estés listo para ir más allá de los términos y definiciones, en esta sección encontrarás algunos enlaces que te pueden proporcionar más información sobre los temas del capítulo.

PREGUNTAS DE REVISIÓN

No hay preguntas por las que exista necesariamente una respuesta correcta o incorrecta. Estas preguntas tienen como objetivo el diálogo con tus estudiantes.

RETO DE PROGRAMACIÓN

Si quieres retar a tus estudiantes más allá de lo que se encuentra en el libro de texto de Xojo, estos retos de programación están diseñados para ampliar los ejercicios sobre los cuales se trabaja en cada capítulo. No se proporcionan las soluciones porque para un problema dado de programación puede haber incontables modos de solucionarlo. Más importante que el código esté “bien”, es el comportamiento y salida proporcionados por el programa. Para promover la exploración, los retos pueden involucrar el uso de controles o técnicas que no hayan sido presentados en el libro de texto de Xojo.

Te animamos a que tu también leas cada capítulo del libro de texto Xojo, haciendo los ejercicios y consultando esta guía a medida que lo hagas.

Capítulo 1: ¡Hola, Mundo!



1.1 Conceptos y Vocabulario

1) IDE

IDE son las siglas de Integrated Development Environment en inglés (Entorno de Desarrollo Integrado). Un IDE es un programa informático utilizado para introducir y editar código, además de incluir otras características como autocompletado de código y depuración (el IDE de Xojo ofrece ambas). En resumen, es el programa que usas para crear otros programas.

2) Bug

Un bug es un fallo en un programa informático. Cuando un programa no se comporta como se espera, es un bug. Algunas veces el bug se debe a un problema con el código introducido. A menudo es un problema relacionado con las suposiciones realizadas por el programador.

3) Depurar

Hablando estrictamente, depurar significa eliminar bugs o errores de un programa. En términos más prácticos, también es el término utilizado para describir el proceso de analizar un programa en funcionamiento para comprobar defectos en su comportamiento.

4) Código

De forma simple, el código son las “palabras” del lenguaje de programación que se introduce en el IDE para crear programas. Como verás después, el código puede estar compuesto por palabras clave, nombres de función, variables, etc. Pero por ahora, piensa en el código como las palabras escritas en el IDE.

5) Lenguaje

Si el código son las palabras que escribes, el lenguaje es el conjunto de palabras que puedes utilizar. Xojo es un lenguaje de programación; al igual que lo son Swift, C#, PHP, Objective-C, Java y Python. Si bien muchos lenguajes comparten

muchas similitudes en su sintaxis y también palabras clave, cada lenguaje es único a su manera.

6) Ejecutar y Compilar

Cada vez que ejecutas una app en tu ordenador, tablet o teléfono inteligente, estás ejecutando un programa. Para ejecutar un programa, el IDE (o a veces un compilador incorporado en muchos IDE) necesita convertir tu código en algo que el ordenador pueda entender y usar como instrucciones para saber cómo ha de actuar. Cuando el resultado es una app sobre la que puedes hacer tap o doble clic para ejecutarla, dicho proceso se llama compilación. Xojo y algunos otros IDE también pueden tomar tu código y ejecutarlo sin necesidad de crear una app que puedas usar posteriormente. Con Xojo también puedes compilar una aplicación Web que se ejecute en el navegador web de tu ordenador.

7) Excepción

Una excepción tiene lugar cuando ocurre algo en el programa que se supone que nunca debería de ocurrir. No es necesariamente lo mismo que un bug. Un bug es un error en tu código; es algo que has hecho mal al crear tu app. Una excepción ocurre cuando el programa final está ejecutándose, y si bien puede apuntar a un error en tu código, también puede suceder que hayas escrito bien el código... pero haciendo suposiciones incorrectas. La gestión de excepciones de Xojo te puede indicar qué tipo de excepción ha ocurrido. Recuerda que la palabra “excepción” significa que algo que se supone que no podría ocurrir, ha ocurrido; como por ejemplo quedarse sin memoria para trabajar con una imagen o intentar acceder a un archivo que no existe. La clave para tratar con excepciones es preparar el código para gestionar dichas situaciones.

1.2 Enlaces y Referencias

1. Programación:

https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software

2. Preguntas y Respuestas de Programación:
<http://programmers.stackexchange.com/>
3. Depuración:
https://es.wikipedia.org/wiki/Error_de_software
4. Excepciones:
<http://docs.xojo.com/index.php/Exception>

1.3 Preguntas de Revisión

1. ¿Cuál es la diferencia entre ejecutar y compilar una aplicación?
2. Una de las opciones que puedes definir en el StageCode de tu aplicación en beta. ¿Cuán importante es hacer betatesting de tus aplicaciones? ¿Cuáles podrían ser algunas buenas formas de probar tus aplicaciones?
3. Una excepción es algo que se supone que no debería de ocurrir, pero a veces ocurre. Un buen ejemplo podría ser cuando falla el arranque en tu coche al girar la llave de encendido. ¿Qué cosas podrías comprobar para esta excepción en particular?
4. ¿Cuáles son algunas cosas “excepcionales” del mundo real que te vienen a la cabeza? ¿Cómo te prepararías para lidiar con ellas?

1.4 Reto de Programación

Como reto sencillo, cambia el mensaje que aparece al hacer clic en el botón. Como reto más avanzado, utiliza un BevelButton en vez de un PushButton para lanzar el mensaje.

Capítulo 2: Preséntate a ti Mismo



2.1 Conceptos y Vocabulario

1) Variable

Una variable es un valor al que se asigna una etiqueta en código. El valor puede cambiar, pero el nombre continúa siendo el mismo. Puedes tener una variable llamada PageCount que mantenga el número de páginas impresas en el programa. Cada vez que tu programa imprime una nueva página, se incrementa el valor de PageCount. Las variables pueden ser de diferentes tipos. La mayoría de estos tipos se explican en la *Introducción a la Programación con Xojo*, mientras que otros son los detallados a continuación.

2) Asignar

El proceso de indicarle a tu programa cuál es el valor de una variable se llama asignación. Estás asignando un valor cuando escribes algo como esto:

```
x = 42
```

La asignación también puede hacer uso de la autoreferencia a la variable, como por ejemplo:

```
x = x + 1
```

3) String / Cadena

Uno de los tipos más comunes de variables es la cadena (string), siendo simplemente un fragmento de texto. Cada vez que tu programa toma un nombre, una dirección web o un párrafo de información, esta se almacena en una string. Si la palabra “string” no tiene mucho significado para ti, piensa en ello como varios caracteres (números, letras, puntuación, etc.) que se “encadenan” juntos para su visualización.

4) Integer / Entero

Por otra parte, un Entero es un valor numérico (un Integer es siempre un número entero, Xojo también incluye números de coma flotante — o Doubles —, en los que se puede manejar información decimal). Es importante distinguir entre cadenas y enteros: puedes almacenar “1” como cadena o como entero. La diferencia

básica es que deberías almacenar variables como enteros (o Doubles) si vas a realizar cualquier tipo de manipulación matemática, incluso tan simple como la suma.

5) Boolean / Booleano

Un booleano es un tipo de variable que puede almacenar uno de estos dos valores: True o False. Son más útiles cuando necesitas mantener un registro del estado o condición de algo, como por ejemplo cuando se ha guardado un documento o se han definido las preferencias de tu programa. Un booleano siempre toma el valor por defecto False y no cambia hasta que el código así lo indique.

6) Comentario

Los comentarios son la forma de documentar lo que hace el código. Mientras que forma parte del código, no son parte de la app compilada final. Están ahí como referencia para cualquiera que quiera leer, depurar o modificar este código en cualquier momento.

2.2 Enlaces y Referencias

1. Variable:

[https://es.wikipedia.org/wiki/Variable_\(programación\)](https://es.wikipedia.org/wiki/Variable_(programación))

2. Booleano:

https://es.wikipedia.org/wiki/Tipo_de_dato_lógico

3. String / Cadena:

https://es.wikipedia.org/wiki/Cadena_de_caracteres

4. Integer / Entero:

https://es.wikipedia.org/wiki/Número_entero

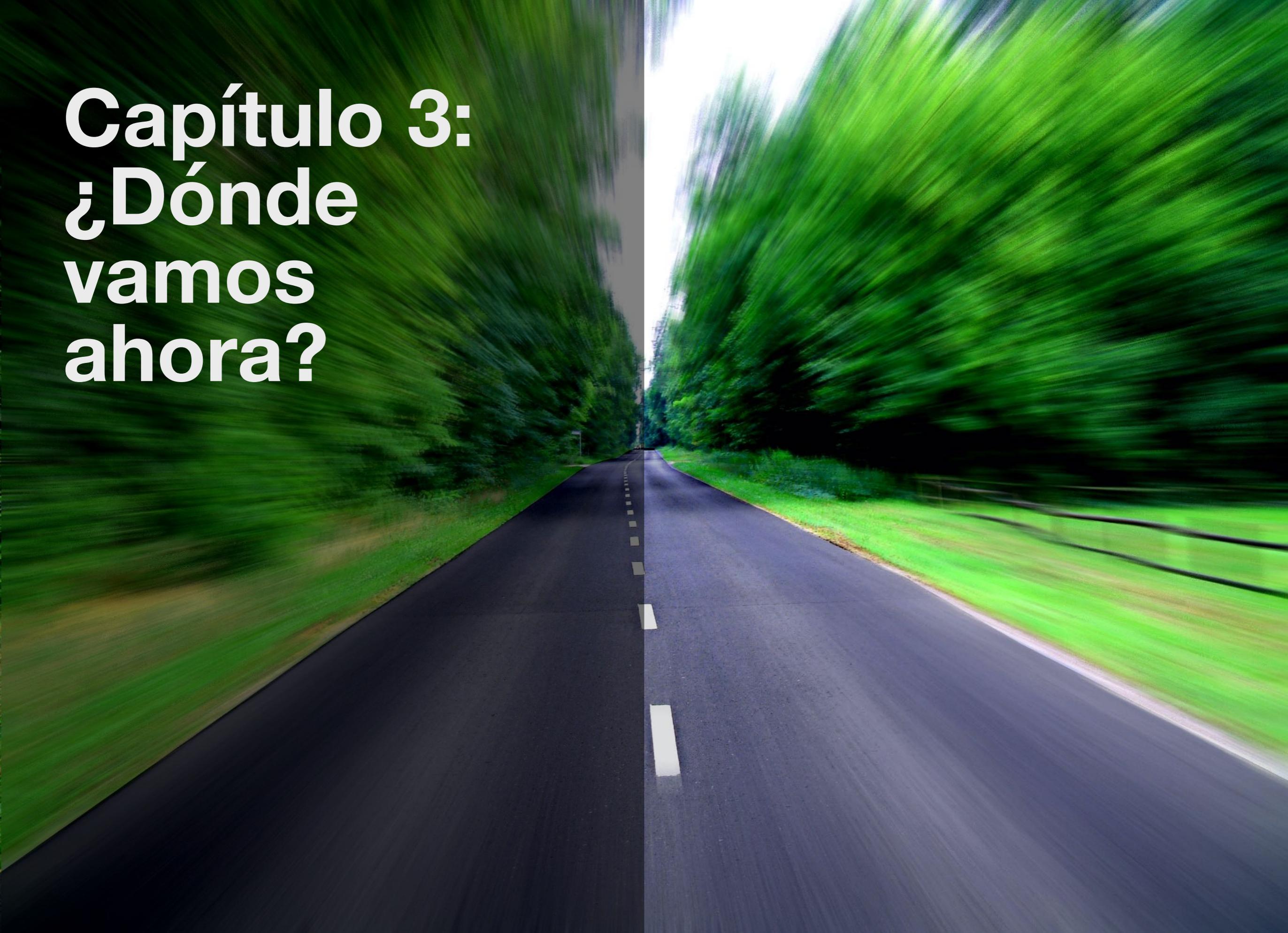
2.3 Preguntas de Revisión

1. ¿Cuáles son algunos de los motivos por los que es importante añadir comentarios al código?
2. ¿Cómo explicarías la diferencia entre “12” (la cadena) y 12 (el entero) a alguien sin experiencia de programación?
3. ¿Por qué es tan importante ser consistente con las estrategias a la hora de dar nombres a las variables?

2.4 Reto de Programación

En vez de calcular la edad del usuario en años, calcúlalo en días o meses.

Capítulo 3: ¿Dónde vamos ahora?



3.1 Conceptos y Vocabulario

1) Control de flujo y Lógica Booleana

El control de flujo y la lógica booleana van de la mano. La lógica booleana es un término técnico para un escenario del tipo “si esto, entonces haz tal cosa”. Involucra comprobar el valor de una variable y responder a continuación en consecuencia. El control de flujo es el nombre dado a dicho proceso en su conjunto (piensa en una gráfica de flujo con varias ramas, y te harás una idea). Comprueba los ejemplos del libro de Xojo para ver algunos ejemplos sobre el aspecto de estas instrucciones en código.

2) Operador

Un operador es algo que actúa sobre una variable. Algunos ejemplos típicos pueden ser los operadores matemáticos como +, - y *. Xojo también incluye otros operadores. Aprenderás más sobre estos en los siguientes capítulos.

3) Case

En algunas ocasiones, las situaciones “si esto, entonces haz tal cosa” pueden tornarse complejas en tu código. Si tienes una situación en la que una variable puede tener más de dos valores y cada valor requiere de una acción distinta, entonces es un buen punto en el cual usar la instrucción Case. Una instrucción Cas es muy similar a la instrucción If, pero ofrece una serie de opciones. Una vez más, el libro de texto de Xojo incluye algunos ejemplos.

4) Loop / Bucle

Un bucle es un modo de iterar una lista de elementos. En el ejemplo de código de este capítulo, el programa itera por todas las fuentes del usuario en el ordenador. En cada una de las iteraciones, el programa puede realizar una serie de acciones, incluyendo la de salir del bucle en caso de que sea necesario.

3.2 Enlaces y Referencias

1. Control de Flujo:

https://es.wikipedia.org/wiki/Estructuras_de_control

2. Lógica Booleana:

https://es.wikipedia.org/wiki/Álgebra_de_Boole

3. Bucle:

[https://es.wikipedia.org/wiki/Bucle_\(programación\)](https://es.wikipedia.org/wiki/Bucle_(programación))

3.3 Preguntas de Revisión

1. ¿De qué forma es una instrucción Select más eficiente en comparación con una instrucción If?
2. ¿Cuáles son las diferencias entre los bucles For...Next loop y Do...Loop? ¿Cuándo usarías uno en vez del otro?

3.4 Reto de Programación

Añade botones de estilo a la app Font Previewer para cambiar negrita, cursiva y/o subrayado. Como reto más avanzado, muestra los nombres de las fuentes en el ListBox usando las propias fuentes.

Capítulo 4: Hacer las Cosas



4.1 Conceptos y Vocabulario

1) Método/Función

Un método o función, de la forma más simple, es un conjunto de instrucciones para tu programa. Si tu programa realiza una tarea específica más de una vez, entonces es buena idea convertir dicho código en un método en vez de escribir el mismo código en múltiples sitios de tu programa. Tanto los métodos como las funciones son conjuntos de código. La diferencia clave es que una función devuelve un valor, mientras que un método no.

2) Parámetro

Algunos métodos y funciones pueden comportarse de forma distinta basándose en los valores que les puedas proporcionar. Dichos valores se denominan parámetros. No todos los métodos o funciones han de recibir parámetros, pero pueden hacerlo.

3) Return / Retorno (Valor Devuelto)

Algunos métodos y funciones realizarán una tarea y saldrán sin más. Otros están diseñados para proporcionar un valor de vuelta al código que llama a la función. Estos valores se denominan valores devueltos.

4) Scope / Ámbito

Las variables definidas en una función no pueden usarse fuera de la función. Esto se conoce como ámbito, y que limita las partes del código que pueden acceder a otras partes del código. Por ejemplo, puedes tener un método que necesite modificar una cadena, pero no quieres que otra parte del código tenga acceso a ella. El ámbito de dicha string podría estar restringido al método.

5) Valor por defecto (Default)

Puedes especificar que algunos de los parámetros de tu función tengan valores por defecto en el caso de que no se les asigne uno. Estos se denominan valores por defecto.

4.2 Enlaces y Referencias

1. **Ámbito:**
[https://es.wikipedia.org/wiki/Ámbito_\(programación\)](https://es.wikipedia.org/wiki/Ámbito_(programación))
2. **Funciones:**
<https://es.wikipedia.org/wiki/Subrutina>
3. **Return:**
<https://es.wikipedia.org/wiki/Subrutina>

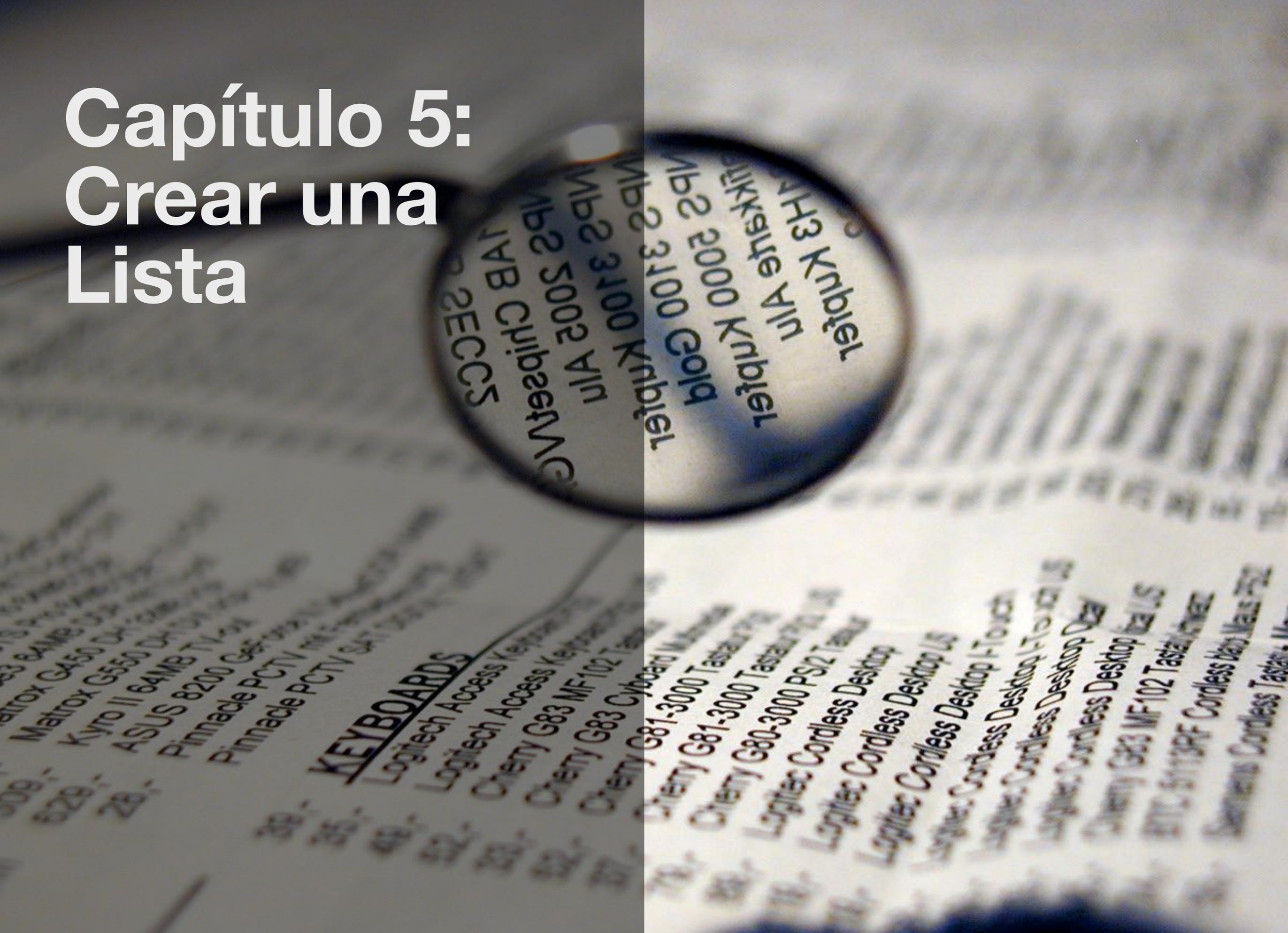
4.3 Preguntas de Revisión

1. ¿Cuándo deberías de convertir un fragmento de código en un método o función?
2. Los parámetros por defecto son muy convenientes; pero, ¿puedes pensar en una situación en la que podrían introducir fallos en tu aplicación?
3. ¿En qué casos sería inteligente proporcionar un ámbito de Privado a un método? ¿Por qué?

4.4 Reto de Programación

En la app Font Previewer, añade la capacidad de que el usuario final seleccione un tamaño de fuente en la previsualización. Como reto más avanzado “duplica” la interfaz de la app, de modo que los usuarios puedan seleccionar cómo se ven dos fuentes distintas en pantalla.

Capítulo 5: Crear una Lista



5.1 Conceptos y Vocabulario

1) Array

En su nivel más básico, un array es una lista de elementos. Estos elementos pueden ser algo tan simple como enteros o algo más complejo como tipos de datos definidos por ti. Incluso puedes tener arrays de arrays. Cada fila de un array tiene algo denominado índice, y que indica su posición en el array (estas posiciones empiezan en cero, de modo que la primera fila tiene el índice 0, la segunda el índice 1, etc.)

2) Añadir filas

Añadir una fila a un array es lo mismo que añadir algo al final de la lista. También puedes añadir una fila al array en una posición concreta indicando a Xojo el índice en el que quieres que esté el nuevo elemento.

3) Ordenar

Por defecto, los ítems del array permanecerán en el orden en el que fueron creados. Sin embargo, puedes ordenarlos. Los métodos de ordenación más comunes son el alfabético o el numérico.

4) Desordenar

Al igual que al barajar cartas, el método Shuffle hace que las filas de un array se ordenen aleatoriamente.

5) Delimitador

Un delimitador es un caracter usado para “marcar” secciones de datos. Por ejemplo, es posible que hayas utilizado un archivo .csv (valores separados con coma) en el pasado. De ser así, habrás advertido que cada valor está separado, o delimitado, mediante una coma.

5.2 Enlaces y Referencias

1. Arrays:

[https://es.wikipedia.org/wiki/Vector_\(informática\)](https://es.wikipedia.org/wiki/Vector_(informática))

2. Delimitadores:

https://es.wikipedia.org/wiki/Valores_separados_por_comas

3. Ordenamiento:

https://es.wikipedia.org/wiki/Algoritmo_de_ordenamiento

4. Barajado:

<http://www.codinghorror.com/blog/2007/12/shuffling.html>

5.3 Preguntas de Revisión

1. Más allá del ejemplo de juego de cartas, ¿en qué otras situaciones podrías querer desordenar un array?
2. Explica tu clase schedule desde la perspectiva de un array.
3. El ejemplo de lista de tareas es una aplicación web. ¿Qué hace que la web sea una plataforma especialmente buena para una app de tareas?
4. ¿Qué otras apps podrían llevarse a versiones web?

5.4 Reto de Programación

Añade la capacidad de asignar una prioridad a cada tarea y permitir que el usuario pueda ordenarlas.

Capítulo 6: ¿Puedo tomar su pedido?



6.1 Conceptos y Vocabulario

1) Evento

Cuando tu programa realiza ciertas tareas, este ha de realizar necesariamente otras. Por ejemplo, si necesitas mostrar un botón en pantalla, entonces tu programa ha de crear el botón y dibujarlo. Tu programa también ha de responder al usuario. Cuando el usuario hace clic en un botón, es posible que quieras que se ejecute código como respuesta a dicha acción. Cada acción tomada por el usuario se denomina Evento. Puedes añadir código a diferentes eventos en tu programa para cambiar el modo en el se comporta e interactúa con los usuarios.

2) Control

También llamado a veces widget, un control es un objeto de interfaz de usuario que muestra información o reacciona al usuario, como pueda ser un botón, una etiqueta, un listbox o una barra de desplazamiento.

3) Label / Etiqueta

Una etiqueta es (generalmente) un control no interactivo que indica el propósito o nombre de otro control. A menudo se utiliza como una mera etiqueta.

4) Botón

Un botón es un control interactivo que realiza una acción (indicada por ti) cuando el usuario hace clic sobre él. Los botones pueden tener diferentes tamaños, aspectos o formas.

5) Ventana

Una ventana es el agrupamiento físico de controles en tu pantalla. Habitualmente pensamos en cada ventana como una pantalla, hoja o vista.

6.2 Enlaces y Referencias

1. Eventos:

https://es.wikipedia.org/wiki/Programación_dirigida_por_eventos

2. Controles:

<https://es.wikipedia.org/wiki/Widget>

6.3 Preguntas de Revisión

1. En general es mejor utilizar los controles nativos del sistema (como los `PushButton`) en vez de los diseñados por ti. ¿Cuáles son los motivos de que esto sea así? ¿En qué situaciones querrías crear tus propios botones desde cero?
2. ¿Cuáles son algunas ventajas de usar selectores, como un `Slider` o `RadioButton` en vez de permitir que el usuario introduzca texto?

6.4 Reto de Programación

En la app de pedidos de comida, añade un botón de radio para el salchichón, pero asegúrate de que sólo esté activo cuando se seleccione Pizza como plato principal.

Capítulo 7: Navegando

HHTH :|||

7.1 Conceptos y Vocabulario

1) Tabla

Una tabla es similar visualmente a una hoja de cálculo, con filas y columnas de datos. En Xojo, la tabla se denomina ListBox. Un ListBox puede tener cabeceras, encargarse de ordenar los elementos y también ofrecer capacidades de edición en línea. Cuando tu app necesita mostrar datos tabulados de algún tipo, el ListBox probablemente sea el control que debas de usar.

2) Grupo de Controles

Un grupo de controles es exactamente lo que indica su nombre: un agrupamiento lógico de controles. Es posible actuar sobre todos los controles en un grupo de controles como uno.

3) Barra de Progreso

Una barra de progreso es un control que indica el avance en el procesamiento de algo. Algunos ejemplos típicos pueden ser la descarga de un archivo o la reproducción de una película. La barra de progreso se utiliza cuando se conoce el valor final. Cuando es desconocido, debería de utilizarse el control ProgressWheel (también puedes usar la ProgressBar con un valor de Indeterminado para obtener una “barra de barbería”, indicando así al usuario que se está realizando una acción pero se desconoce la cantidad de tiempo que va a requerir).

4) TabPanel

Las pestañas (o Tabs) son otra forma de agrupar controles. Probablemente te hayas encontrado con pestañas en tu navegador web. Al usar un control TabPanel puedes organizar tu interfaz en grupos más lógicos, dado que cada pestaña contendrá su propio grupo de controles.

7.2 Enlaces y Referencias

1. Barra de Progreso:

https://www.glosarioit.com/Barra_de_progreso

2. Pestañas:

[https://es.wikipedia.org/wiki/Pestaña_\(informática\)](https://es.wikipedia.org/wiki/Pestaña_(informática))

7.3 Preguntas de Revisión

1. ¿En qué situaciones sería buena idea usar un PagePanel o TabPanel? ¿Cuándo usarías una u otra?
2. Un navegador web es evidentemente más complejo que el proyecto de ejemplo de este capítulo. ¿Cuáles son algunas de las características fundamentales que no están implementadas? ¿Qué otras funciones no estándar añadirías a un navegador web?
3. Explica como podrías usar PagePanels para crear una interfaz de “asistente”.

7.4 Reto de Programación

Añade un listado de marcadores preseleccionados entre los que pueda escoger el usuario para navegar con un clic. Como reto más avanzado, ofrece al usuario la posibilidad de añadir sus propios marcadores (no te preocupes de guardarlos al salir de la aplicación).

Capítulo 8: Házlo tu Mismo



8.1 Conceptos y Vocabulario

1) Clase

Una clase es un tipo personalizado de variable que puedes diseñar tu mismo. Como se indica en el libro de texto de Xojo, generalmente representa un objeto del mundo real (como una persona o coche) o bien una idea abstracta (como una reserva en un hotel o la inscripción a un curso). Una clase puede incluir sus propios datos, propiedades y métodos. Una clase Persona puede incluir propiedades como el nombre y edad, así como métodos como Saltar, Correr y Hablar.

2) Objeto

Piensa en la clase como la definición de tu dato personalizado, y en el objeto como la instancia de dicho dato en tu código. La clase es la idea y el objeto es aquello sobre lo que actúa tu código. Por ejemplo, tu serías un objeto específico creado a partir de la clase "Persona", con tus propios valores en cada una de las propiedades.

3) Propiedad

Una clase puede tener propiedades definidas por ti. Estos son atributos del objeto o la idea que este representa. Por ejemplo, una clase que representa una Persona podría tener propiedades como Apellido como String, Nombre como String, y EdadEnAños como Integer.

4) Variante

Una variante es un tipo de dato intrínseco en Xojo que puede contener cualquier tipo de dato, ya sea string, integer, boolean o bien un objeto personalizado diseñado por ti. Las Variantes (tipo de dato Variant) deberían de usarse con precaución, dado que puedes introducir bugs sutiles en tu código con mucha facilidad. En general debería de usarse una variante sólo cuando necesites guardar diferentes tipos de datos (cadenas, enteros u objetos personalizados) en el mismo sitio, como en un CellTag de un ListBox.

5) Módulo

En Xojo, un módulo es similar a una clase, pero sólo puede existir una instancia del mismo en tu app. Es fácil pensar en un módulo como una clase de uso único. A diferencia de la Clase, un módulo existe siempre, lo que hace que resulte útil para tareas como almacenar y cargar preferencias, gestionar datos disponibles para toda la aplicación, y otras funciones globales.

8.2 Enlaces y Referencias

1. Clases:

[https://es.wikipedia.org/wiki/Clase_\(informática\)](https://es.wikipedia.org/wiki/Clase_(informática))

2. Propiedades:

[https://es.wikipedia.org/wiki/Atributo_\(informática\)](https://es.wikipedia.org/wiki/Atributo_(informática))

3. Variantes:

https://es.wikipedia.org/wiki/Tipo_variante

8.3 Preguntas de Revisión

1. Este capítulo utiliza el ejemplo de una clase actuando como plantilla y el objeto como la “casa” creada a partir de ella. ¿Qué otras analogías te ayudarían a explicar la diferencia?
2. Las variantes deberían de usarse con cautela. ¿Cuáles podrían ser buenos usos de las variantes?
3. ¿Cuándo deberías de crear tus propias clases en vez de los tipos incluidos de serie en Xojo?

8.4 Retos de Programación

Añade una propiedad “EmailAddress” a la clase Student. Proporciona alguna vía para que el profesor pueda enviar un email a un estudiante (pista: consulta el método ShowURL).

Capítulo 9: Entrada y Salida



9.1 Conceptos y Vocabulario

1) Archivo

Un archivo es un fragmento de datos que residen físicamente en un disco. Un archivo puede ser cualquier cosa, como un documento de Word, una canción en formato MP3 o un controlador de impresora.

2) Carpeta

También llamado directorio, una carpeta es un contenedor lógico que almacena archivos y otras carpetas. En Xojo, se representan tanto las carpetas como los archivos mediante la clase FolderItem.

3) Stream / Flujo

Los flujos (o stream) son un método de escribir datos sobre un archivo o bien de leer datos de un archivo. La información se envía como un “flujo” de datos desde tu programa a un disco (guardar un archivos), o bien desde un disco a tu programa (abrir y leer un archivo).

4) Diálogo

Un diálogo es una ventana especializada que presenta al usuario información específica (como un mensaje de error o un aviso), o bien que solicita información del usuario (como por ejemplo el archivo a abrir).

5) Estilos de Texto

Probablemente ya estés familiarizado con los estilos de texto de un tratamiento de textos. Los estilos proporcionan añadidos como negrita, cursiva o subrayado. También incluyen información como la fuente seleccionada y el tamaño de fuente.

9.2 Enlaces y Referencias

1. Archivos y Carpetas:

https://es.wikipedia.org/wiki/Sistema_de_archivos

2. Archivos:

http://en.wikipedia.org/wiki/Computer_file

3. Carpetas:

<https://es.wikipedia.org/wiki/Directorio>

4. Estilos de Texto:

https://es.wikipedia.org/wiki/Formato_de_texto

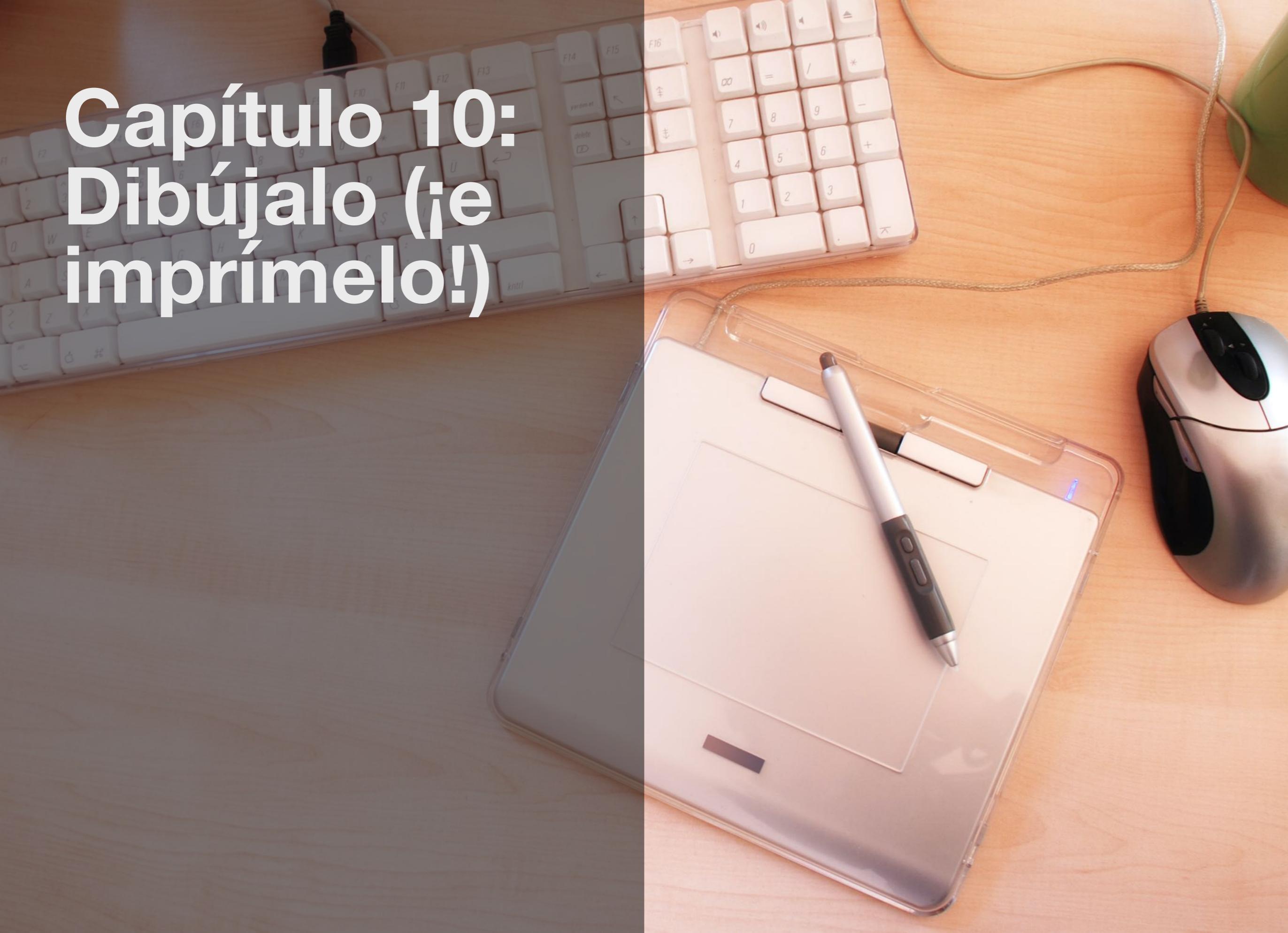
9.3 Preguntas de Revisión

1. ¿Cómo explicarías la diferencia entre un archivo y una carpeta a alguien con poca o ninguna experiencia en ordenadores?
2. ¿Cuándo harías que tu aplicación use archivos de texto en oposición a archivos binarios?

9.4 Reto de Programación

En este capítulo has aprendido sobre los estilos de texto. Crea una nueva app cuyo único propósito sea tomar texto con estilo y devolver la versión de texto plano (las mismas palabras pero sin ningún tipo de estilo aplicado).

Capítulo 10: Dibújalo (¡e imprímelo!)



10.1 Conceptos y Vocabulario

1) Imagen

En Xojo hay diferencias entre imágenes, Pictures y gráficos. Una imagen es aquello en lo que normalmente pensarías como dibujo (Picture). Es una captura de una escena o elemento. Si tienes una fotografía de un perro, esto sería una imagen.

2) Picture

En la terminología de Xojo, una Picture es un objeto que representa una imagen. Tiene propiedades que reflejan los detalles de la imagen, como su altura y ancho.

3) Graphics (Gráficos)

Graphics es el objeto de Xojo que hace la mayor parte del trabajo cuando se trata de trabajar con imágenes. La clase Graphics contiene métodos para dibujar, imprimir y prácticamente cualquier cosa relacionada con el dibujado de gráficos.

4) Canvas

El Canvas es un control especial en Xojo que te proporciona varias opciones para mostrar imágenes o texto. Tiene un objeto Graphics que lo hace extremadamente flexible.

5) Imprimir

Imprimir es el hecho de enviar información, ya sea texto o imágenes, desde tu programa a una hoja de papel.

10.2 Enlaces y Referencias

1. Canvas:
<https://docs.xojo.com/Canvas>
2. Impresión:
<https://docs.xojo.com/UserGuide:Printing>

10.3 Preguntas de Revisión

1. ¿Cuáles son las diferencias entre imagen, dibujo y gráficos? ¿Por qué son diferentes?
2. ¿Por qué es mala idea asumir que conoces las dimensiones de un objeto Graphics antes de enviarlo a la impresora?

10.4 Reto de Programación

Crear una aplicación que se pueda usar para abrir, cambiar el tamaño, recortar y escalar imágenes atendiendo a las especificaciones del usuario.

Capítulo 11: Conexiones



11.1 Conceptos y Vocabulario

1) Red

Una red es un grupo de ordenadores u otros dispositivos conectados. Diferentes tipos de redes tienen diferentes tipos de conexiones y velocidades; pero generalmente cada dispositivo se puede conectar con otro dispositivo.

2) Protocolo

Un protocolo es una descripción detallada sobre como dos o más programas o dispositivos pueden comunicarse en una red. Habrás observado que muchos sitios web comienzan con “http” —esta es la indicación de que el sitio utiliza el protocolo de transmisión hipertextual. Existen protocolos para imprimir, enviar emails o cualquier otra tarea que puedas realizar a través de una red.

3) Puerto

Con todos estos datos fluyendo hacia el ordenador, ¿cómo sabe el ordenador a qué aplicación se dirige cada dato? Bien, cada protocolo se comunica mediante un puerto, representado por un número. Por sí mismos los puertos no significan realmente mucho; se trata simplemente de un valor acordado para que los dispositivos se puedan comunicar entre sí. La web utiliza generalmente el puerto 80, mientras que el envío de email utiliza el puerto 25. En la mayoría de los sistemas informáticos, los puertos inferiores a 1024 están reservados para su uso por parte del sistema. Sólo una aplicación se puede comunicar por un puerto a la vez.

4) Dirección

Cada dispositivo de la red tiene una dirección. En la mayoría de las redes actuales, esta es una dirección IP representada por cuatro números, que van desde el cero al 255, separados mediante un punto entre ellos. En la mayoría de los casos no necesitas comprender como funcionan las redes, simplemente recuerda que cada dispositivo necesita una dirección única.

5) Socket

En Xojo, las tareas de red se realizan usando la clase Socket. El socket tiene funciones y propiedades encargadas de realizar el “trabajo duro” para llevar a cabo las comunicaciones de red.

11.2 Enlaces y Referencias

1. Redes:

https://es.wikipedia.org/wiki/Red_de_computadoras

2. Protocolos:

https://es.wikipedia.org/wiki/Protocolo_de_comunicaciones

3. Puertos

[https://es.wikipedia.org/wiki/Puerto_\(informática\)](https://es.wikipedia.org/wiki/Puerto_(informática))

11.3 Preguntas de Revisión

1. Más allá de recuperar los contenidos de una página web, ¿en qué otras formas podrías usar el `HTTPSocket`?
2. ¿Encuentras más valor en usar (o ampliar) un protocolo existente o bien en desarrollar el tuyo propio? ¿Por qué?

11.4 Reto de Programación

En tu app de email, añade la capacidad de enviar email a múltiples receptores. Como reto más avanzado, añade la capacidad de añadir archivos como adjuntos.

Capítulo 12: Filas y Columnas

STENOGRAPHIC / WITTING BOARD

12.1 Conceptos y Vocabulario

1) Base de datos

Una base de datos es una colección organizada de datos. La información en una base de datos está organizada habitualmente en tablas, explicado a continuación.

2) Tabla

La tabla es el bloque de construcción básico de una base de datos. Una tabla representa un objeto del mundo real (como una Persona o un Coche) o bien una idea abstracta, tal y como ocurre con una clase en Xojo.

3) Columna

Cada tabla en una base de datos tiene como mínimo una columna. Una columna es una pieza de información relacionada en la tabla. Por ejemplo, una tabla que representa coches puede tener columnas que almacenen información como el fabricante, número de modelo y tamaño.

4) Fila (Row)

Una fila es una instancia de una “cosa” representada en la tabla de la base de datos. Por ejemplo, si tu tabla se llama Personas, cada fila representará a una persona.

5) Servidor de Bases de Datos

Algunas bases de datos están alojadas en archivos que residen en tu propio ordenador, mientras que otras requieren de servidores de bases de datos.

6) SQL

SQL significa Structured Query Language (Lenguaje de Consultas Estructuradas), y es el lenguaje que se utiliza para obtener o insertar datos de la base de datos.

7) CRUD

CRUD es un acrónimo de Create (crear), Read (leer), Update (actualizar) y Delete (borrar); siendo las cuatro funciones básicas de cualquier base de datos o aplicación de bases de datos. Si tu app utiliza una base de datos, entonces probablemente la mayoría de cosas que haga esté relacionada con una de estas cuatro operaciones.

8) Fecha ISO

La mayoría de las bases de datos almacenan las fechas en un formato llamado ISO 8601, a menudo referenciado como Fecha ISO. El formato es YYYY-MM-DD: cuatro dígitos para el año, dos para el mes, y dos para el día. Observa que tanto el mes como el día han de incluir ceros a la izquierda para cualquier día inferior a 10.

12.2 Enlaces y Referencias

1. Aprender SQL:

<http://www.sqlcourse.com>

2. Descripción de SQL:

<https://es.wikipedia.org/wiki/SQL>

3. CRUD (Create, Read, Update, Delete):

<https://es.wikipedia.org/wiki/CRUD>

4. ISO Dates:

https://es.wikipedia.org/wiki/ISO_8601

5. Servidor de Bases de Datos:

https://es.wikipedia.org/wiki/Servidor_de_base_de_datos

12.3 Preguntas de Revisión

1. Enumera algunos ejemplos de bases de datos usados en el mundo real, tanto si son informatizadas como si no.
2. ¿Por qué crees que las fechas se almacenan en formato ISO? ¿Cuáles son algunas de las ventajas y desventajas de este formato?

12.4 Reto de Programación

En tu app Agenda, añade la capacidad de editar los contactos (pista: consulta el comando UPDATE de SQL).

Capítulo 13: Todo en Familia: Subclases



13.1 Conceptos y Vocabulario

1) Subclase

Una subclase es el “hijo” de una clase existente. Como hijo, tiene todos los atributos y métodos de sus padres, pero también puede incluir atributos y métodos adicionales.

2) Súperclase

El padre de una subclase también puede ser denominada como su súperclase.

3) Herencia

Cuando una subclase recibe los atributos y métodos de su súperclase, esto se conoce como herencia.

4) Cursor

En los sistemas operativos de escritorio, el cursor es el objeto de la pantalla que se mueve con el ratón o con cualquier otro dispositivo apuntador, mostrando así la posición del usuario en pantalla.

13.2 Enlaces y Referencias

1. Inheritance:
[https://es.wikipedia.org/wiki/Herencia_\(informática\)](https://es.wikipedia.org/wiki/Herencia_(informática))
2. Cursores:
[https://es.wikipedia.org/wiki/Cursor_\(informática\)](https://es.wikipedia.org/wiki/Cursor_(informática))

13.3 Preguntas de Revisión

1. ¿Cómo explicarías las subclases a alguien recién llegado a la programación?
2. En este capítulo has aprendido sobre cursores. ¿Por qué es importante tener diferentes cursores en tu app?

13.4 Reto de Programación

Añade otros tipos de personas a tu app de subclases: ayudantes del profesor, directores y secretarías. Piensa sobre los atributos que necesitan y que sean distintos de los ya disponibles en las clases de un estudiante o profesor.

Capítulo 14: Encerar y Pulir



14.1 Conceptos y Vocabulario

1) Thread (Hilo)

Los hilos son un tema muy complejo pero, por ahora, puedes pensar en un hilo como en un trabajador en tu ordenador. Cada trabajador puede hacer una tarea o parte de una tarea, y mediante el uso de varios trabajadores a la vez, tu app puede ser más rápida o con una respuesta más ágil.

2) Timer (Temporizador)

El Timer es el control de Xojo que ejecuta un evento cuando indiques. Los eventos pueden ser únicos o recurrentes; y los Timer se pueden activar o desactivar cuando necesites.

3) Random (Aleatorio)

La clase Random de Xojo se utiliza para generar un número aleatorio sin saber qué número será (aunque puedes indicar un límite superior e inferior).

14.2 Enlaces y Referencias

1. Gestión de Excepciones:
<https://docs.xojo.com/index.php/Exception>
2. Hilos:
[https://es.wikipedia.org/wiki/Hilo_\(informática\)](https://es.wikipedia.org/wiki/Hilo_(informática))
3. Aleatoriedad:
<https://es.wikipedia.org/wiki/Aleatoriedad>

14.3 Preguntas de Revisión

1. ¿Qué es más importante, la velocidad computacional o la agilidad de respuesta de la interface? ¿Preferirías tener un programa rápido que pareciese estar bloqueado o uno más lento que permaneciese usable e informativo? ¿Habría situaciones en las cuales cambiases tu respuesta?
2. ¿Qué significado tiene una interface poco cuidada sobre el desarrollador? ¿Qué implica una interface moderna y elegante?

14.4 Reto de Programación

Crema una nueva app que utilice un Timer para seguir los cambios en el portapapeles del usuario. Almacena las nuevas entradas del portapapeles en un ListBox, asegurándote de que no existan entradas duplicadas.

Epílogo



CONTENIDOS

1. **Gracias**
2. **Sobre los Autores**

Gracias

Gracias por tomarte el tiempo de leer *Introducción a la Programación con Xojo - La Guía del Profesor*.

Si tienes cualquier pregunta, comentario o sugerencia sobre cualquier cosa de este libro, por favor envía un email a

docs@xojo.com

Sobre los Autores

BRAD RHINE

Brad es un autodenominado geek de los ordenadores y ha estado trabajando como Desarrollador de Ordenadores, Desarrollador Web, Escritor Técnico, Administrador de Bases de Datos, Asistente del Director de Tecnología y, brevemente, como vendedor de árboles de Navidad.

También es un antiguo columnista de XDev Magazine y ha realizado presentaciones en las Xojo Developer Conference sobre diferentes temas.

Brad ha pasado gran parte de su carrera profesional trabajando en el sistema escolar público.

Cuando no está escribiendo código o escribiendo sobre código, puedes encontrar a Brad tocando su guitarra, pasando el tiempo con su familia, o corriendo.

Vive en la Pennsylvania rural con su mujer, sus dos hijos, su perro y u dos gatos inadaptados.

PAUL LEFEBVRE

Paul es un Ingeniero Xojo que también ha contribuido en gran medida a la documentación y ejemplos, entre otras cosas. Ha estado trabajando con ordenadores desde la primera vez que usó un Atari 400 en 1983.