

EN SÓLO 20 HORAS

Aprende lo
que quieras
de manera

— ***Rápida***



JOSH KAUFMAN

de

«Algún día cuando tenga tiempo lo haré». ¿Lo has dicho? Aprende a usar tu tiempo.

Sacude tu lista de pendientes y obtén las herramientas para desarrollar tu destreza en tiempo récord.

En solo 20 horas ofrece un enfoque sistemático para la rápida adquisición de habilidades: cómo aprender cualquier habilidad nueva lo antes posible. Su método muestra cómo simplificar habilidades complejas, maximizar la práctica productiva, y eliminar las barreras de aprendizaje. Con tan solo 20 horas de práctica enfocada y constante, irás de saber absolutamente nada a realizar algo notablemente bien.

Este método no es teórico: es probado sobre el terreno. Josh Kaufman invita a los lectores a poner en práctica el método, ya sea para aprender programación, practicar algún deporte, tocar un instrumento, estudiar un nuevo idioma, o conocer el juego de mesa más antiguo y más complejo del mundo. Y tú, ¿qué quieres aprender hoy?



Josh Kaufman

En solo 20 horas

Aprende lo que quieras de forma rápida

ePub r1.0

Titivillus 06.11.2019

Título original: *The First 20 Hours*
Josh Kaufman, 2013

Editor digital: Titivillus
ePub base r2.1



Para Lela.

Nota para el lector

La vida es tan corta y hay tanto que aprender.

Geoffrey Chaucer, *Parlement of Foules*, 1374

«Hay tanto que deseo hacer... y tan poco tiempo para hacerlo». He aquí la historia de la vida moderna.

Tómate un momento para considerar cuántas cosas te gustaría aprender y cómo hacerlo. ¿Qué hay en tu lista? ¿Qué te impide comenzar?

Por lo regular se trata de dos factores: tiempo y dificultad.

Ahora una verdad incómoda: las experiencias que más recompensas ofrecen en la vida casi siempre requieren cierto nivel de destreza. Las destrezas requieren de tiempo y esfuerzo: el tiempo no lo tenemos y nos mostramos reticentes a invertir el esfuerzo necesario.

«Algún día, cuando tenga tiempo, lo haré».

Francamente, es más fácil sentarse frente a la televisión o navegar por Internet, de modo que eso es justo lo que la mayoría hacemos, y nuestros deseos permanecen como sueños.

Aquí tenemos otra verdad incómoda: muchas cosas no son divertidas hasta que eres bueno haciéndolas. Toda destreza tiene lo que yo llamo *barrera de frustración*: un periodo en que se es tremendamente malo y uno

está dolorosamente al tanto del hecho. ¿Para qué empezar con algo en lo que, de seguro, serás muy malo?

¿No sería grandioso aprender nuevas habilidades con menos angustia? ¿Te gustaría superar la barrera de la frustración rápidamente, para llegar a la parte gratificante? ¿Quisieras pasar menos tiempo en la confusión y la duda y más tiempo divirtiéndote?

¿Es posible aprender nuevas destrezas con menos esfuerzo y en menos tiempo?

Hablando por experiencia propia, la respuesta es sí. Es posible.

Este libro trata de mi búsqueda personal en el terreno del arte y la ciencia de la *adquisición rápida de destrezas*: cómo aprender cualquier nueva habilidad tan rápido como sea posible. El propósito de esta obra es ayudarte a lograr nuevas destrezas en un tiempo récord.

De acuerdo con mi experiencia, se requieren unas 20 horas de práctica para romper la barrera de la frustración. Me refiero a comenzar sin saber nada de lo que se pretende aprender, para llegar a tener un desempeño notable.

En solo 20 horas es una aproximación sistemática a la adquisición de nuevas habilidades tan rápido como sea posible. El método es universal. No importa si pretendes aprender una lengua, escribir una novela, pintar un cuadro, comenzar un negocio o volar un aeroplano. Si inviertes al menos 20 horas en aprender los elementos básicos de la destreza, te sorprenderá lo bueno que puedes ser.

Sea cual sea la destreza que desees obtener, este libro te ayudará a lograrla en menos tiempo y sin desperdiciar energía. Con un poco de esfuerzo bien dirigido y estratégico, llegarás a hacer las cosas bien y rápidamente, evitando la desagradable frustración.

En este libro, comenzaremos por los principios de la adquisición rápida de destrezas para obtenerlas tan pronto como sea posible. Las ideas y las prácticas no son complicadas, así que no tardarás en aprenderlas.

Luego, explicaré cómo usar estos principios en el mundo real al demostrarte la manera en que adquiriré las siguientes habilidades nuevas en 20 horas o menos para cada una, con no más de 90 minutos de práctica al día:

- Desarrollar una práctica personal de yoga.

- Escribir un programa de computadora para funcionar en la red.
- Reaprender mecanografía sin mirar el teclado.
- Explorar el más viejo y complejo juego de mesa de la historia.
- Tocar un instrumento musical.
- Practicar surf a vela.

Espero que este libro te ayude a sacudirle el polvo a esa lista de pendientes para que puedas replantearla y comprometerte a aprender algo nuevo.

*Josh Kaufman.
Fort Collins, Colorado,.
Estados Unidos de América.*

**Para obtener actualizaciones del material contenido en este libro,
visita <http://first20hours.com/updates>.**

Retrato del autor como un adicto al aprendizaje

*Cada mañana me levanto determinado a cambiar el mundo y a pasármela la
mar de bien.
A veces, esto hace que la planeación de mi día sea difícil.*

E. B. White, ensayista y autor de *La telaraña de Carlota* y de *The Elements of Style*.

Hola. Mi nombre es Josh Kaufman y soy adicto al aprendizaje.

Mi casa y oficina están repletas de libros, herramientas y equipo sin usar de todo tipo. La mayoría de las cosas solo acumulan polvo lentamente.

Mi lista de cosas por aprender contiene cientos de actividades. Mi carrito de compras en Amazon.com contiene en este momento 241 artículos, todos los libros que deseo leer. No puedo entrar a una librería sin comprar tres o cuatro libros nuevos que se sumarán a los 852 que ya poseo.

Cada día tengo una idea para otro proyecto o experimento, y la añado a mi siempre creciente lista de cosas pendientes. Ver la lista de las cosas que quiero aprender a hacer puede ser abrumador, así que no la reviso muy seguido.

Quiero aprender a mejorar mi negocio editorial. Quiero aprender a tomar y editar videos. Quiero producir un programa de audio. Quiero aprender a impartir mejores seminarios y a enseñar mejores cursos.

Tengo una idea para un nuevo producto, pero no sé cómo construirlo. Tengo ideas para nuevos programas de computadora, pero no sé cómo crearlos. Tengo más proyectos de escritura en la cabeza que tiempo y energía para escribirlos.

Quiero aprender a dibujar, a navegar en kayak en los rápidos, a pescar, a escalar en roca. Quiero ser capaz de tocar la guitarra, el ukulele, el piano y el violín eléctrico.

Hay juegos que me han interesado durante años, como el Go, pero no he aprendido a jugarlos. Tengo juegos que ya sé jugar, como el ajedrez, pero no soy muy bueno que digamos, por lo que no me divierten mucho y no suelo jugarlos con regularidad.

Me gusta la idea de jugar golf, pero en cada uno de mis juegos termino avergonzado de mí mismo. (Suelo decir que juego golf-maratón: cuando llego al hoyo 18, ya he recorrido tanta distancia como en un maratón).

Pareciera que cada día agrego una actividad a la lista de cosas que quiero aprender a hacer, y así infinitamente. Tanto que aprender, y tan poco tiempo.

Soy autodidacta por naturaleza. Si tengo que hacer algo, prefiero intentarlo yo a buscar ayuda. Incluso si otra persona puede hacerlo más rápido y mejor, me resisto a privarme de una experiencia de aprendizaje.

Para complicar las cosas todavía más, Kelsey, mi esposa, maneja su propio negocio editorial de educación continua para maestros de yoga. Nos va bien en los negocios, así que siempre estamos ocupados.

Para hacer la vida todavía más interesante, dimos la bienvenida a nuestra hija: Lela tiene nueve meses de edad al momento de redactar estas líneas.

Antes del nacimiento de Lela, Kelsey y yo decidimos que si teníamos hijos, sería prioritario criarlos nosotros mismos. Una de las principales razones por las que dejé mi empleo como gerente de una empresa de las que aparecen en la lista de las 500 más importantes, según la revista *Forbes*, fue tener la flexibilidad de trabajar en casa y disponer así del tiempo necesario para estar con mi familia.

Kelsey y yo compartimos las responsabilidades paternas. Dado que somos un hogar que atiende dos negocios (Kelsey trabaja en la mañana mientras yo cuido a Lela, por la tarde, Kelsey cuida a Lela y yo trabajo hasta la hora de la cena), eso me brinda unas 25 horas de trabajo a la semana, además del tiempo que aprovecho cuando Lela duerme.

Después del nacimiento de Lela, sentí que apenas tenía tiempo de hacer mi trabajo. Hablar de adquirir nuevas habilidades era un sinsentido. Para un adicto al aprendizaje, la situación era enloquecedora.

No quise dejar a un lado el aprendizaje y el crecimiento, incluso ante mis nuevas responsabilidades. No dispongo de mucho tiempo libre, pero deseo invertir el poco que tengo tan sabiamente como sea posible.

Eso me llevó a interesarme en lo que llamo «adquisición rápida de destrezas» (métodos para aprender cosas nuevas rápidamente).

Quiero seguir adquiriendo destrezas, pero no deseo que el proceso se prolongue por siempre. Quiero aprender lo esencial rápidamente para así progresar sin sentir una frustración *constante*.

Te aseguro que me comprendes. ¿De cuánto tiempo dispones cada día cuando ya has trabajado y atendido las necesidades familiares? ¿Tienes la impresión de que el día debería durar unas 36 o 48 horas para aprender algo nuevo?

Hay un viejo dicho que reza: «trabaja mejor y no más duro». Al parecer, el proceso de adquisición de habilidades no tiene relación con las horas invertidas, sino con lo que se incluye en dichas horas.

Maldito seas, Malcolm Gladwell

En 2008, Malcolm Gladwell escribió un libro titulado *Outliers: The Story of Success*. En esa obra intenta explicar qué hace que ciertas personas sean más exitosas que otras.

Una de las ideas que Gladwell repite una y otra vez es la llamada «regla de las 10 000 horas». Con base en una investigación realizada por el doctor K. Anders Ericsson, de la Universidad Estatal de Florida, para alcanzar el nivel de experto se requieren, en promedio, diez mil horas de práctica efectiva^[1].

Diez mil horas equivalen a ocho horas de práctica diaria durante tres años

y medio, sin descansar un solo día, sin fines de semana ni vacaciones. Asumiendo que disponemos de unos 260 días laborales al año, se trataría de una labor de tiempo completo durante cinco años, y eso dando por hecho que pasas 100 por ciento de ese tiempo dando 100 por ciento de tu energía y esfuerzo.

En la práctica, este nivel de atención concentrada es extremadamente agotador. Incluso los mejores del mundo en campos ultracompetitivos (como el ámbito de la música y los deportes profesionales) logran concentrarse unas tres horas y media al día. Esto significa que puede requerirse de una década o más para desarrollar una habilidad hasta alcanzar su maestría.

En esencia, si quieres volverte experto en una actividad, la investigación del doctor Ericsson indica que te llevará muchísimo tiempo lograrlo. Ser el mejor del mundo en algo, incluso durante poco tiempo, requiere de años de práctica constante. Si no estás dispuesto a invertir el tiempo y el esfuerzo, serás opacado por los que sí están dispuestos a hacerlo.

El libro del que hablo ocupó el primer lugar en ventas en el rubro no ficción durante tres meses. De la noche a la mañana, la «regla de las 10 000 horas» estaba en todas partes.

Pareciera que aprender algo nuevo no es ya suficientemente difícil. No solo debes buscar tiempo para practicar, sino encontrar 10 000 horas. La mayoría de nosotros podemos considerarnos afortunados si disponemos de unas horas a la semana. ¿Para qué tomarse la molestia si se requiere de tanto tiempo para ser bueno en algo?

¡Toma en cuenta mi esfuerzo, Dios todopoderoso!

Antes de desanimarte, considera lo siguiente.

Existe un elemento en la investigación del doctor Ericsson que es muy fácil pasar por alto: se trata de un estudio de desempeño *a nivel experto*. Si quieres ser el siguiente Tiger Woods, es muy probable que debas pasar al menos 10 000 horas practicando cada aspecto del golf sistemática y deliberadamente. Casi todo golfista profesional comienza a practicar este deporte a edad muy temprana y lo hacen durante al menos siete años. Para ser competitivo a nivel mundial, se requiere de tiempo.

Por otra parte, ¿qué tal si tu objetivo no consiste en ganar la gira de la PGA? Supongamos que quieres ser lo suficientemente bueno para jugar de manera decente, sin sentir vergüenza, pasándola bien y con oportunidades de ganar el torneo de tu club local. Estamos hablando de algo completamente distinto. La maestría a nivel mundial puede exigir 10 000 horas de esfuerzo concentrado, pero se requiere de mucho menos tiempo para hacer las cosas suficientemente bien y cumplir tus propósitos.

No pretendo discutir el valor de lo que Ericsson llama la «práctica deliberada» (práctica intencional y sistemática necesaria para mejorar en una destreza). La práctica deliberada constituye la esencia de la adquisición de una capacidad. La verdadera cuestión sería averiguar *cuánta* práctica deliberada se requiere para alcanzar tu meta. Por lo regular, mucho menos tiempo del que crees.

Calidad, no cantidad

La clave para la adquisición rápida de una habilidad es abrazar la idea de la *suficiencia*. En este libro, discutiremos el desarrollo de tus capacidades, no cómo llegar a ser uno de los mejores del mundo. Haremos que la curva de aprendizaje sea menos difícil o pronunciada, para ascenderla lo más rápidamente posible.

Deja las 10 000 horas a los profesionales. Nosotros comenzaremos con 20 horas de esfuerzo concentrado, inteligente y dirigido.

Nuestro objetivo es lograr resultados valiosos para nosotros con una fracción de esfuerzo. Puede que nunca obtengas una medalla de oro, pero sí la recompensa que deseas en mucho menos tiempo.

Si después de todo quieres volverte experto en la destreza que te interesa, tus posibilidades de éxito serán mucho mayores si empiezas con 20 horas de adquisición rápida de habilidades. Al saber en qué te metes, al aprender lo fundamental, practicar inteligentemente y desarrollar una rutina de práctica, progresarás más rápida y consistentemente, alcanzando el nivel de experto en tiempo récord.

¿Qué es la rápida adquisición de destrezas?

Es un proceso, una forma de reducir la habilidad que pretendes adquirir a sus partes más pequeñas, identificando cuáles son más importantes, para luego practicarlas deliberadamente. Así de simple.

La rápida adquisición de destrezas está integrada por cuatro etapas principales:

- Deconstruir una habilidad en sus partes más pequeñas (subhabilidades).
- Aprender lo suficiente de cada parte como para practicar inteligentemente y autocorregirte durante la práctica.
- Remover las barreras físicas, mentales y emocionales que se interponen en la práctica.
- Practicar las subhabilidades principales durante al menos 20 horas.

Es todo. La adquisición rápida de habilidades no es física nuclear. Simplemente decides qué practicar, averiguas cuál es la mejor forma de hacerlo, encuentras el tiempo necesario y luego practicas hasta alcanzar el nivel de desempeño al que aspiras.

No hay magia. Se trata más bien de realizar un esfuerzo inteligente y estratégico en algo que te interesa. Con un poco de preparación, obtendrás las nuevas destrezas rápidamente, con menor esfuerzo.

Eso no quiere decir que los resultados sean inmediatos. El deseo de gratificación instantánea constituye una de las principales razones por las que la gente no adquiere nuevas habilidades de manera rápida.

El malentendido «Matrix»

¿Recuerdas la escena de la película *Matrix* en la que Keanu Reeves abre los ojos, parpadea unas cuantas veces y murmura: «Sé kung-fu»?

Lamento decepcionarte, pero la adquisición rápida de habilidades no es *tan* rápida.

Hollywood nos ha afectado en este sentido. Claro que sería bueno aprender a pilotear un helicóptero Bell 212 en 5 segundos, al subir *software* directamente a nuestro cerebro, pero la verdad es que la ciencia va muy por detrás de la ciencia ficción.

Hasta que podamos meter la información al cerebro por medio de

programas instantáneos, «rápido» significa solamente menos tiempo del que normalmente se requiere cuando una persona usa el procedimiento normal (un proceso que suele ser casi a ciegas, desordenado e inconsistente).

Una de las primeras destrezas profesionales que aprendí fue desarrollar páginas web (construir sitios de Internet útiles y funcionales). Habiendo empezado con un sitio web llamado Angelfire.com, en 1996, aprendí cómo leer y escribir HTML y CSS (la lengua franca de la red), cómo usar el programa Adobe Photoshop para editar imágenes y cómo mantener los sistemas que hacen público mi trabajo.

No aprendí nada de esto en la preparatoria o en la universidad. Aunque completé un curso en sistemas de información para negocios, las coincidencias entre lo aprendido en el aula y lo que hago a diario son prácticamente inexistentes.

Adquirí la capacidad de desarrollar sitios web por medio de prueba y error, sobre la marcha. Cada vez que me topaba con una nueva técnica o herramienta que prometía mejorar mi sitio web o reducir mi carga de trabajo, experimentaba con ella. Pasado un largo tiempo, mis capacidades mejoraron.

Mi desordenado sistema de adquisición de habilidades para la red cumplió con su propósito: obtuve un trabajo basado en dichas habilidades y ahora publico información en la red como una forma de vida. Misión cumplida, al menos desde un punto de vista.

Por otra parte, aprendí todo de la manera difícil. Ciertamente, podrías tú alcanzar el nivel de conocimiento que yo alcancé en mucho menos de quince años si haces las cosas de modo sistemático. Si te propusieras practicar estas habilidades inteligentemente, podrías alcanzar mi nivel de competencia general en un mes.

A eso me refiero cuando hablo de rápida adquisición de destrezas. Si lograras aprender la mayoría de lo que sé del diseño en Internet en un solo mes y no en quince años, estarías ante un avance enorme. Es posible hacerlo.

El tiempo necesario para adquirir una nueva habilidad depende de cuánto dedicas a la práctica deliberada y a la experimentación inteligente. También depende de qué tan bueno ha de ser tu desempeño para llegar al nivel deseado.

No esperes resultados de la noche a la mañana. Eso sí: el tiempo total invertido será mucho menor del necesario si entras al proceso careciendo de estrategia.

Antes de explorar el método en detalle, debes saber que la rápida adquisición de habilidades nada tiene que ver con tu forma de aprender en la escuela. El aprendizaje académico tiene una relación casi nula con la adquisición de habilidades, y eso sin hablar de la velocidad a la que se alcanzan los logros.

Adquisición de destrezas vs. aprendizaje

Al igual que muchos preparatorianos en Estados Unidos, estudié una lengua extranjera. Cada día de escuela durante cuatro años me senté en mi clase de español. Mis calificaciones eran altas: puros dieces.

Hoy, aparte de: «Hola, ¿cómo estás?» y «Muy bien», no soy capaz de sostener una conversación con un hispanoparlante. (Ni siquiera sé qué decir si no he tenido un buen día).

Por otra parte, mi amigo Carlos Miceli creció hablando español en Argentina. En la preparatoria, Carlos decidió que quería hablar inglés con fluidez, así que se esforzó por conversar todo lo posible con angloparlantes. En el proceso, descubrió Skype y puso su propio sitio web para practicar su conversación y escribir en inglés regularmente.

Carlos nunca tomó clases. No conoce las reglas formales de la gramática inglesa. Ni siquiera puede decirte cómo aprendió inglés. No es importante. Eso sí: puede hablar y escribir en inglés de modo competente, lo que en verdad importa a fin de cuentas.

El doctor Stephen Krashen, de la Universidad del Sur de California, es experto en el área del aprendizaje de segundas lenguas. Uno de los primeros descubrimientos de Krashen consiste en que la *adquisición* de una lengua es distinta del *aprendizaje* de una lengua.

En la escuela aprendí mucho *sobre* el idioma español: miles de palabras, conjugaciones y reglas gramaticales. Aprendí todo esto para pasar los exámenes con excelentes resultados.

Sin embargo, esos exámenes nada tenían que ver con mi capacidad de hablar español inteligente y comprensiblemente con una persona que hablara

con rapidez. Si mi objetivo era hablar español con fluidez, unas cuantas semanas tratando de conversar con gente en español hubieran dado mejores resultados que cuatro años de escuela.

En esa época, mi objetivo no era hablar español con fluidez. Solo quería sacar diez en el examen final. Por otra parte, Carlos se saltó las clases y simplemente comenzó a practicar. En vez de aprender infinitas conjugaciones de verbos, Carlos participó en lo que realmente importa: comunicarse con otras personas en inglés.

En términos de eficiencia y de valor a largo plazo, la aproximación de Carlos fue muy superior a la mía. Nada que oponer.

El verdadero valor del aprendizaje

Lo anterior no quiere decir que aprender sobre la habilidad no sea importante. Aprender puede ser extremadamente importante, pero no en el sentido que imaginas. Aprender conceptos relacionados con una habilidad te ayuda a *autocorregirte* mientras practicas.

Si sabes conjugar verbos en español, podrás autocorregirte mejor mientras hablas con un hispanoparlante nativo. Si aprendes vocabulario común, serás capaz de comprender lo que un hispanoparlante dice, además de disponer de un cúmulo de información que puede ayudarte cuando te trabes al hablar.

El doctor Krashen llama a esto «hipótesis monitor». Aprender te ayuda a planear, editar y corregirte mientras practicas. Por eso es importante el aprendizaje. El problema viene cuando confundimos el aprendizaje con la adquisición de habilidades.

Si deseas adquirir una nueva destreza, debes practicarla en contexto. El aprendizaje mejora la práctica, pero no la remplaza. Si importa el desempeño, el aprendizaje por sí mismo nunca es suficiente.

Adquisición de destrezas vs. entrenamiento

También existe una gran diferencia entre adquisición de habilidades y entrenamiento. El entrenamiento, en este contexto, significa mejorar una

habilidad mediante la repetición. Es lo que sucede tras adquirir una habilidad básica, si deseas mejorar.

Tomemos como ejemplo correr un maratón. La mayoría de nosotros adquirimos la capacidad de correr durante la infancia. Además de poner un pie delante del otro y mantenerte en equilibrio hasta completar los 42 kilómetros, no hay mucho que aprender en lo que a nuevas habilidades se refiere.

Sin embargo, hay un esfuerzo importante necesario para fortalecer el cuerpo y aclimatarlo para completar un maratón. Ese esfuerzo y el consecuente proceso de fortalecimiento constituyen el entrenamiento. Mientras más entrenas, más te fortaleces y más rápido puedes concluir el maratón.

También existe un elemento de aprendizaje cuando corres un maratón: cómo inscribirte en las carreras, cómo calificar para competencias importantes como el maratón de Boston, saber qué esperar mientras corres, ritmo de la carrera, equipo requerido, etcétera.

Por ejemplo, un asunto menor como la fricción entre la playera y la piel no es muy importante si participas en una carrera de 5 kilómetros, por lo que la mayoría de los corredores no piensa en él. La fricción innecesaria se convierte en algo *muy* importante cuando corres 42 kilómetros.

Si no te preparas lo suficiente, puedes experimentar el infame sangrado de pezones. Es doloroso, penoso... y completamente prevenible. (¿No me crees? Haz una búsqueda en Google).

Entrenamiento y aprendizaje ciertamente te ayudarán a terminar la carrera, pero no constituyen la adquisición de una habilidad o destreza. Sin haber adquirido destreza, el entrenamiento es imposible o inútil. La preparación y el acondicionamiento pueden facilitar algunas formas de adquisición de habilidades, pero jamás remplazarán la práctica.

No obstante, aprender de nuevo a correr para lograr un buen desempeño sí puede considerarse una adquisición de habilidades. Técnicas como la Carrera Chi^[2] ayudan al corredor a moverse de forma que minimice el esfuerzo y la pérdida de movimiento frontal entre las zancadas. Con un poco de práctica, el corredor puede reaprender la habilidad básica de la carrera, lo cual puede reforzar el entrenamiento subsecuente.

Adquisición de habilidades vs. educación y acreditación

A pesar de los sesudos esfuerzos de maestros y profesores de todo el mundo, los métodos educativos modernos y la acreditación casi nada tienen que ver con la adquisición de habilidades.

Esta última requiere practicar la habilidad en cuestión. Se requieren periodos significativos de concentración sostenida y dirigida. Se necesita creatividad, flexibilidad y libertad para determinar qué logros constituirán el éxito.

Desafortunadamente, la mayor parte de los métodos modernos de educación y acreditación solo requieren cumplir. El objetivo primario (no declarado) no es la adquisición de habilidades útiles, sino la certificación de que se completaron criterios arbitrarios, casi siempre establecidos por un comité ajeno a los estudiantes, con el propósito de validar ciertas cualidades que parecen importar a terceros.

La creatividad, la flexibilidad y la libertad de experimentación — elementos esenciales en la rápida adquisición de destrezas— resultan casi antiéticas para el proceso de validación. Si los estándares son demasiado flexibles, no son realmente estándares, ¿cierto?

Para nuestra mala fortuna, la educación rigurosa y la validación de credenciales pueden evitar la adquisición de habilidades. El principal problema es el costo de oportunidad: si los requisitos para obtener la validación son tan intensos que te impiden practicar las habilidades en cuestión, los programas de validación pueden hacer más mal que bien.

Imagina a un individuo listo y motivado que se interesa en echar a andar una empresa de *software*. Completar un grado académico en ciencias de la computación, en una universidad de prestigio, suele tomar al menos cuatro años^[3].

Transcurridos esos cuatro años, nuestro flamante graduado ha pasado miles de horas aprendiendo algoritmos y analizando compiladores. Puede aprobar decenas de exámenes, pero no está más cerca de fundar una empresa de *software* que cuando entró a la universidad. Nuestro desafortunado estudiante ha memorizado muchas cosas sobre la programación, al menos

temporalmente, pero todavía no sabe crear un programa computacional suficientemente útil para la gente y que paguen por él.

Para crear una empresa de *software* se deben adquirir nuevas habilidades: aprender lenguajes de programación, instalación y mantenimiento de sistemas de cómputo, uso de herramientas y programas, creación de prototipos, cómo hallar usuarios inicialmente, cómo obtener financiamiento y realizar las actividades administrativas básicas.

¿Coinciden en algún punto las dos actividades, es decir, la de echar a andar una empresa con obtener un reconocimiento educativo? Claro que sí. Pero fíjate en el énfasis: la mayor parte del esfuerzo realizado para obtener un reconocimiento se dedica a cumplir requisitos. En el mejor de los casos, poco importa si son útiles o no para desempeñarte en el mundo real.

En mi primer libro, *The Personal MBA: Master the Art of Business* (2010), expliqué por qué me salté la educación universitaria para aprender los principios de la práctica comercial moderna e iniciar mi propio negocio. Al no asistir a la escuela de negocios para invertir mi tiempo en construirlos, aprendí muchísimo y me ahorré unos 150 000 dólares en el proceso. De acuerdo con mis objetivos, esta estrategia fue bastante positiva.

Si quieres ser bueno en algo que requiere de la vida real para la preparación, debes practicar esa habilidad en contexto. El estudio, en sí mismo, nunca es suficiente.

Neurofisiología de la destreza: la plasticidad cerebral y la memoria muscular

Un último comentario antes de entrar en materia: debes apreciar plenamente ser capaz de adquirir nuevas habilidades.

Resulta un tanto extraño lo que acabo de decir, pero es fácil caer en el error de pensar que las capacidades son fijas: se es bueno en algo o no.

En su obra *Mindset: The New Psychology of Success* (2007), la psicóloga Carol Dweck cita un amplio cuerpo de investigación que indica que, por lo común, la gente opera con una de las dos perspectivas existentes sobre el funcionamiento de la mente.

Según Dweck, la gente con un esquema mental «fijo» asume que capacidades y talentos son innatos, que se nace con ciertas habilidades que

son como son. Si una persona con este tipo de mentalidad fija no es buena en matemáticas, entonces el esfuerzo extraordinario para practicarlas es una mera pérdida de tiempo. ¿Para qué molestarte si nunca serás bueno en eso?

Por otra parte, la gente con una mentalidad de «crecimiento» asume que las habilidades y destrezas aumentan con práctica y persistencia. Si una de estas personas se equivoca en algunos problemas matemáticos, no piensa que Dios le ha negado atributos matemáticos, sino que le falta práctica. Con persistencia y práctica, solo es cuestión de tiempo para dominar la técnica.

Ahora, las buenas noticias para quienes han caído en la trampa de la mentalidad fija: las investigaciones más recientes indican que todos los cerebros pueden mejorar habilidades y destrezas mediante la práctica. La predisposición genética existe, pero es menor si se le compara con el poder de la práctica enfocada e inteligente. Puedes mejorar cualquier destreza, siempre y cuando estés dispuesto a practicar.

El cerebro humano es *plástico*, término que los neurocientíficos utilizan para indicar que tu cerebro cambia físicamente según tu medio ambiente, actos y consecuencias de dichos actos. Al aprender cualquier destreza nueva, física o mental, las conexiones neurológicas del cerebro cambian conforme practicas.

En palabras del doctor Jon Medina (*Brain Rules*, 2009), «las neuronas que descargan al mismo tiempo se conectan», formando nuevos patrones únicos en los circuitos físicos de tu cerebro. Con el paso del tiempo, tus neuronas comienzan a activarse en patrones más eficientes en respuesta a la retroalimentación que recibes del ambiente al practicar.

Si practicas una *habilidad motora* (es decir, una destreza que implica movimiento físico), siempre te sentirás extraño y lento al principio. Tienes que pensar en todo lo que haces y, por lo regular, se cometen errores frustrantes. Aprender lo básico es una lucha constante.

Al practicar, tu coordinación muscular se torna más automática y sincronizada con los procesos mentales. Obtienes la habilidad de poner más atención a los elementos sutiles de tus actos y aprendes a ajustar tu actitud respecto a la retroalimentación que te da el medio ambiente.

Empiezas a hacer más de lo que funciona y menos de lo que no sirve. Eventualmente, serás capaz de desempeñarte sin prestar atención consciente a

cada detalle.

En la literatura académica, este proceso general se conoce como «modelo de las tres etapas» para la adquisición de destrezas^[4], y aplica para el caso de las destrezas físicas y mentales. Las tres etapas son:

1. Etapa cognitiva (temprana): comprender lo que se trata de hacer, investigar, pensar en el proceso y dividir la destreza en partes más manejables.
2. Etapa asociativa (intermedia): se practica la destreza, se tiene en cuenta la retroalimentación ambiental y se ajusta nuestra forma de comprender la actividad con base en esa retroalimentación.
3. Etapa autónoma (tardía): se realiza la destreza efectiva y eficientemente sin pensar en ella o sin prestar atención innecesaria al proceso.

Este proceso neurofisiológico de adquisición de habilidades tiene lugar todo el tiempo, incluso mientras lees esta oración. No existe la llamada «mente quieta». Tu cerebro aprende, codifica y consolida nuevas habilidades todo el tiempo.

Como escribe Dweck en *Mindset*: «Tu mente es como un músculo: mientras más la usas, más crece». Mientras más practicas, más eficiente, efectiva y automática se vuelve la destreza.

Son buenas noticias para la rápida adquisición de destrezas. Si tu mente y tu cuerpo son capaces de desempeñarse bajo una nueva y mejor óptica, bien podemos ingeniarnos para hacer que el proceso sea *más veloz*.

Los diez principios de la rápida adquisición de destrezas

Me di cuenta de que lograr la maestría en el karate no tiene que ver con aprender 4000 movimientos, sino con repetir algunos movimientos 4000 veces.

Chet Holmes, autor de *The Ultimate Sales Machine*.

Ahora que tenemos claro lo que significa la adquisición de habilidades, examinemos cómo hacerlo rápidamente. La intención de este capítulo es proporcionarte un listado de lo que necesitas para adquirir cualquier nueva habilidad.

Para mí es útil pensar en estos principios como si fueran formas de cultivar una «obsesión temporal». La rápida adquisición de destrezas sucede naturalmente cuando te tornas tan curioso e interesado en algo que todo lo demás no importa, al menos transitoriamente.

Piensa en estos principios como maneras de identificar una destreza por la que vale la pena entrar en obsesión temporal, concentrarse y retirar las distracciones o barreras que te apartan de la práctica efectiva.

He aquí los diez principios mayores para la adquisición rápida de habilidades:

1. Elige un proyecto que ames realmente.
2. Concentra tu energía en una sola habilidad en un momento determinado.
3. Define tu nivel de desempeño.
4. Deconstruye la destreza hasta dar con los elementos que la componen.
5. Obtén las herramientas indispensables.
6. Elimina las barreras que te impiden practicar.
7. Dedica un tiempo a la práctica exclusiva de la habilidad.
8. Crea ciclos de retroalimentación rápida.
9. Practica en periodos breves con ayuda de un reloj.
10. Enfatiza cantidad y velocidad.

Muchos de estos principios pueden parecerte mero sentido común, y es correcto. Recuerda: no basta conocer estos principios, debes ponerlos en práctica para obtener recompensas.

Elige un proyecto que ames realmente

Karl Popper fue uno de los más grandes filósofos del siglo XX. Popularizó la idea de la *falsabilidad científica*. En términos comunes, si no puedes probar que algo es potencialmente erróneo por la observación o la experimentación, no se trata de ciencia.

Popper dijo muchas cosas sabias, pero me parece que la siguiente es de las más destacadas: «Lo mejor que puede pasarle a un ser humano es encontrar un problema, enamorarse de él y vivir tratando de resolverlo, a menos que aparezca otro todavía más atractivo».

Si quieres dar con una fórmula para vivir una vida satisfactoria y productiva, no puedes equivocarte en esta.

La adquisición rápida de habilidades requiere elegir un problema o proyecto absorbente. Mientras más emocionado estés con la destreza que deseas adquirir, más rápido la obtendrás.

En la práctica, encontrar un proyecto semejante es algo muy personal. Por ejemplo, aprender a hablar y escribir chino mandarín no figura en mi actual

lista de habilidades por obtener, ya que no tengo necesidad urgente de aprenderlo en este momento y también porque tengo muchos otros proyectos que emprender. Si en un futuro decido mudarme a la zona de China en la que se habla mandarín, quizá el proyecto me fascine, pero por ahora no.

Por otra parte, estoy inmensamente interesado en aprender a jugar Go, el juego de mesa estratégico más antiguo del mundo, que se originó en China hace más de tres mil años. Es un juego hermoso y he querido aprender a jugarlo desde que me enteré de su existencia, hace varios años.

Aprender a jugar Go requiere estudio. Las reglas son sencillas, pero es un reto leer con precisión los patrones de piedras blancas y negras en el tablero. Hace años que las computadoras son eficientes al jugar ajedrez, pero hasta las más poderosas la pasan mal jugando al Go con un humano experimentado.

Por naturaleza, aprendes más rápido las cosas que te importan. Actualmente, me interesa más aprender a jugar Go, por lo que pienso dejar el mandarín para después.

Si te concentras en adquirir tu *destreza principal* (es decir, alcanzar tu proyecto más deseado) antes que cualquier otra cosa, lo lograrás en mucho menos tiempo.

Concentra tu energía en una sola habilidad en un momento determinado

Uno de los errores más comunes al adquirir nuevas habilidades, es tratar de obtener demasiadas al mismo tiempo.

Es una cuestión aritmética: obtener nuevas destrezas requiere cierto tiempo de concentración y atención dirigida. Si solo dispones de una o dos horas diarias para practicar y aprender, y distribuyes ese tiempo y energía en veinte destrezas diferentes, ninguna en particular recibirá suficiente tiempo y energía para generar una mejoría notable.

Adoptar este principio es más difícil para algunas personas. En lo personal, siempre he tenido un temperamento renacentista: hay cientos de cosas que quiero aprender en un momento dado, en cientos de áreas distintas. Emocionalmente, me resulta difícil aplazar el aprendizaje de las cosas nuevas que descubro^[5].

Sin embargo, cuando trato de aprender todo al mismo tiempo no aprendo nada. En lugar de progresar, intercambio las destrezas a que aspiro,

frustrándome, para luego hacer otra cosa. Es la receta extremadamente lenta para adquirir destrezas.

Escoge una y solo una. Utiliza toda tu concentración y energía en conseguirla y deja que las demás esperen su turno. David Allen, autor de *Getting Things Done* (2002), recomienda hacer una lista «tentativa»: se trata de una lista de cosas que deseas explorar en el futuro, pero que no te resultan tan importantes en el momento. Al añadir un elemento a esa lista, te liberas de la responsabilidad de actuar o de pensar en el asunto hasta que decidas pasar a la acción.

Insisto: concentrarse en una habilidad esencial cada vez es absolutamente necesario para la rápida adquisición de destrezas. No renuncias a las otras actividades de modo permanente, únicamente las reservas para después.

Define tu nivel de desempeño

El nivel de desempeño al que aspiras consiste en una breve descripción del nivel de aptitud que constituye tu meta. ¿Qué tanto te gustaría dominar la destreza que estás adquiriendo?

Imagina que este nivel deseado es una sola oración en que describes lo que tratas de lograr y qué harás con ello al terminar tu propósito. Mientras más específica sea tu descripción, mejor.

Definir el nivel de desempeño al que aspiras te ayuda a imaginar cómo desempeñarte de una manera determinada. Cuando decides exactamente qué tan bueno quieres o necesitas ser, es más fácil averiguar cómo llegar a este punto. En palabras de Charles Kettering, inventor del sistema de encendido eléctrico para los automóviles: «Un problema bien planteado es un problema medio resuelto».

La definición de tu nivel de desempeño depende de los motivos que te llevaron a adquirir la habilidad. Si te interesa divertirme, tu objetivo es llegar al punto en que dejas de sentirte frustrado para comenzar a disfrutar de la práctica misma. Si tu objetivo es el desempeño, ¿cuál es el nivel mínimo que estás dispuesto a aceptar en principio?

Una vez conseguido el nivel deseado, puedes elegir continuar por ese camino. Los mejores niveles suelen dar la impresión de estar fuera de nuestro alcance, pero recuerda que no son imposibles de obtener.

Como regla general, mientras más relajado sea el nivel al que aspiras, más rápido adquirirás la habilidad asociada. Si tu mente aspira a un nivel de desempeño de clase mundial, podría parecer que estás haciendo trampa, pues bajas el nivel de desempeño para «ganar» más pronto, ¿o no?

Eso es exactamente lo que hacemos y no es una trampa. Recuerda: el nivel de clase mundial no es el objetivo de la rápida adquisición de destrezas. Aspiramos a ser capaces y suficientes en el menor tiempo posible, no a la perfección.

Es importante señalar que algunas habilidades implican consideraciones de seguridad, que siempre deben incluirse en el nivel de desempeño al que aspiras. Lastimarte (o morir) adquiriendo una nueva capacidad, está fuera de nuestros objetivos.

Deconstruye la destreza hasta dar con los elementos que la componen

La mayoría de lo que consideramos destrezas son, en realidad, actividades compuestas por destrezas menores. Cuando ya has identificado la habilidad en que has de concentrarte, debes deconstruirla, dividirla en sus partes mínimas. Por ejemplo: jugar golf es una destreza con muchos subcomponentes: elección del palo correcto, golpe de salida, cómo hacer las cosas al estar en un bunker, en el *green*, etcétera.

Una vez dividida la destreza en sus partes más pequeñas, es mucho más fácil decidir qué subdestrezas son más importantes. Al concentrarte primero en las subdestrezas esenciales, progresarás más con menos esfuerzo.

La deconstrucción de una habilidad también hace más difícil sentirse abrumado. No tienes que practicar todos los componentes de una destreza al mismo tiempo. Es más efectivo concentrarte en las subdestrezas que prometen grandes recompensas.

La deconstrucción también te permite identificar las partes de la destreza no esenciales para los principiantes. Al eliminar estas subdestrezas o técnicas accesorias desde el principio, invertirás más tiempo y energía en lograr la maestría de las subdestrezas de importancia mayor.

Obtén las herramientas indispensables

La mayoría de las destrezas incluyen prerequisites para la práctica y el desempeño. Es difícil jugar tenis sin raqueta, o pilotar un helicóptero si no tienes acceso a uno.

¿A qué herramientas, componentes y ambientes debes tener acceso antes de practicar de manera eficiente? ¿Cómo obtener las mejores herramientas que puedas costear?

Si dedicas un momento para identificar las herramientas indispensables antes de practicar, ahorrarás un tiempo precioso. Al asegurarte los recursos necesarios antes de comenzar, podrás maximizar tu tiempo de práctica.

Elimina las barreras que te impiden practicar

Hay muchas cosas que interfieren con la práctica, lo que dificulta obtener una destreza. Estas barreras pueden consistir en cualquier cosa. Veamos los siguientes casos:

- El esfuerzo anterior a la práctica es significativo. Los problemas quizá consistan en disponer mal tus herramientas, comprar las equivocadas o saltarte los requisitos de preparación.
- Disponibilidad intermitente. Sucede cuando usas equipo prestado o dispones de un recurso con horario limitado.
- Distracciones ambientales. Como televisión, teléfonos y correos electrónicos en tu bandeja de entrada.
- Bloqueos emocionales. Como miedo, duda y vergüenza.

Cada uno de estos elementos hace más difícil empezar, por lo que la velocidad de adquisición de la destreza disminuye.

Depender de la fuerza de voluntad para superar estas barreras constituye una estrategia perdedora. Cada día disponemos de cierta fuerza de voluntad; lo mejor es usarla sabiamente.

La mejor manera de aprovecharla fuerza es para eliminar las barreras que dificultan la práctica. Si dispones tu ambiente para facilitar lo más posible la práctica, adquirirás la habilidad en menos tiempo.

Dedica un tiempo a la práctica exclusiva de la habilidad

El tiempo que dedicas a adquirir una nueva destreza ha de venir de alguna parte. Desafortunadamente, tendemos a adquirir una nueva habilidad sin sacrificar muchas cosas que nos gusta hacer, como ver televisión, usar videojuegos y demás.

«Ya lo haré cuando tenga tiempo», solemos decirnos. He aquí la verdad: eso de «encontrar» tiempo es un mito.

Nadie «encuentra» tiempo para nada como si descubriera milagrosamente una dotación extra de horas, como cuando encuentras un billete que dejaste accidentalmente en el bolsillo de una chamarra.

Si dependes de tener o encontrar el tiempo necesario para hacer algo, seguro nunca lo harás. Si quieres «encontrar» tiempo, debes «hacerlo».

Dispones de 24 horas cada día: 1440 minutos, ni más ni menos. Nunca tendrás más tiempo. Si duermes 8 horas al día, te quedan 16. Algunas de estas horas serán utilizadas para cuidar de ti mismo y de los tuyos. Otras para trabajar.

El tiempo que sobra es el que realmente te permite adquirir una destreza. Si quieres mejorar tus habilidades tan rápido como sea posible, dedica la mayor cantidad de tiempo.

La mejor forma de asegurar tiempo para adquirir una habilidad es identificar las cosas de poco valor en que lo invertimos, para eliminarlas. Como experimento, te sugiero apuntar en un cuaderno el tiempo que dedicas a cada actividad durante unos días. Solo necesitas un cuaderno.

Los resultados de esta lista te sorprenderán: si tomas algunas decisiones difíciles para optimizar el uso de tu tiempo, seguro tendrás bastante para adquirir rápidamente la habilidad que te interesa. Mientras más tiempo tengas al día, menos requerirás para adquirir la destreza. Recomendando dedicar por lo menos 90 minutos al día para practicar. Podrás lograrlo al suprimir las actividades que no valen gran cosa.

También recomiendo *comprometerte* de antemano a practicar por lo menos 20 horas. Cuando comiences, debes practicar hasta llegar a esta marca. Si te sientes atascado, sigue adelante: no puedes parar hasta alcanzar el nivel de desempeño esperado o hasta llegar a 20 horas de práctica. Si no estás dispuesto de antemano a invertir al menos 20 horas, escoge otra destreza.

La razón para esto es simple: las primeras etapas de la adquisición de destrezas suelen sentirse más duras de lo que realmente son. Por lo regular hay confusión y te toparás con problemas y barreras inesperadas. En lugar de darte por vencido ante la menor dificultad, comprométete de antemano a practicar un mínimo de 20 horas y será más fácil persistir.

Piensa en esta manera de hacer las cosas como si fuera un ejercicio para mejorar tu determinación: no permitirás que alguna minucia te impida hacer lo que en verdad quieres hacer. O resuelves el problema o haces lo necesario para llegar a la marca de 20 horas. Llegado ese punto, estarás en mejor posición para decidir cómo proceder.

Crea ciclos de retroalimentación rápida

Retroalimentación rápida significa información precisa sobre lo bien que te estás desempeñando tan pronto como puedas. Mientras más tiempo te tome la retroalimentación precisa, más te llevará adquirir la destreza.

Pongamos como ejemplo elaborar quesos. Los sutiles procesos químicos requeridos para crear quesos finos se llevan durante meses o años, y no hay manera de apresurar el resultado sin arruinar las cosas. Si se requieren seis meses para determinar si tu queso es bueno, esta tardanza en la retroalimentación dificultará que adquieras la habilidad rápidamente.

La retroalimentación rápida conduce, naturalmente, a la rápida adquisición de habilidades. Si la retroalimentación llega inmediatamente o en poco tiempo, es mucho más fácil relacionar esa información con las acciones para hacer los ajustes necesarios.

Las mejores formas de retroalimentación son las casi instantáneas. Por eso es que habilidades como la programación pueden convertirse en algo medianamente adictivo: haces un cambio y algunos milisegundos más tarde la computadora te indica si el cambio funcionó o no. Si no te gusta la retroalimentación que recibes (¡el programa se congeló!), realiza otro cambio e intenta de nuevo.

Existen muchas fuentes potenciales de retroalimentación. Como explicó en un artículo para *New Yorker*^[6] Atul Gawande, cirujano veterano y jugador de tenis *amateur*, entrenadores y mentores experimentados pueden

proporcionar retroalimentación inmediata sobre tu desempeño y recomendar los ajustes necesarios.

Los entrenadores no son la única fuente de retroalimentación rápida. Las cámaras de video y otros artículos semejantes pueden ayudarte a observarte mientras haces las cosas. Herramientas como los programas computacionales, los auxiliares del entrenamiento y otros aparatos pueden indicarte inmediatamente si cometes un error o si falta algo.

Mientras más fuentes de retroalimentación integres a tu práctica, más rápida será la adquisición de la habilidad.

Practica en periodos breves con ayuda de un reloj

Nuestras mentes están diseñadas para aprender, registrar patrones, simular cursos de acción potenciales y tratar de saber qué pasará a continuación. El cerebro no está diseñado para estimar el tiempo con precisión: cuánto tiempo se requiere para hacer algo o cuánto se invirtió en hacer algo.

En las etapas tempranas de la práctica de una nueva habilidad, es muy fácil sobrestimar el tiempo que pasaste practicando. Cuando no eres bueno en algo (y lo sabes), parece que el tiempo avanza lentamente y sientes como si hubieras practicado más de lo que en realidad lo hiciste.

La solución a esto es practicar con ayuda de un reloj. Compra un cronómetro decente^[7] y prográmalo para que cuente 20 minutos en cuenta regresiva. Solo hay una regla: una vez que has hecho funcionar el cronómetro, debes practicar hasta que el tiempo se acabe. No hay excepciones.

Esta sencilla técnica te hará más fácil completar largos periodos de práctica sostenida, incluso cuando te sientes cansado o frustrado.

Mientras más periodos de práctica sostenida completes, más rápidamente adquirirás la destreza. Procura disponer tu tiempo para realizar entre tres y cinco sesiones al día. En poco tiempo verás resultados importantes.

Enfatiza cantidad y velocidad

Cuando empiezas a adquirir una nueva habilidad, resulta tentador concentrarte en practicar buscando perfección, una receta para la frustración.

Tu desempeño, por supuesto, no estará ni cerca de la perfección.

En vez de procurar ser perfecto, concéntrate en practicar tanto como puedas en el menor tiempo posible, haciendo las cosas «suficientemente bien».

En *Art & Fear* (2001), los autores, David Bayles y Ted Orland comparten una anécdota interesante sobre el valor del volumen:

El maestro de cerámica anunció el primer día que dividiría el grupo en dos. Los del lado izquierdo del estudio, dijo, serían calificados únicamente tomando en cuenta la cantidad de trabajo producido, mientras los del lado derecho serían evaluados con base en la calidad.

El procedimiento era simple: al final de la clase, sacaba una báscula y pesaba el trabajo del grupo «cuantitativo»: 25 kilos de vasijas darían un 10, veinte kilos darían un 8 y así sucesivamente. Los evaluados por la calidad, debían producir una vasija —perfecta— para obtener un 10.

Pues bien. Al llegar el momento de la evaluación, surgió un hecho curioso: los trabajos de más alta calidad fueron producidos por el grupo que era evaluado por la cantidad. Parece que, en tanto el grupo «cuantitativo» producía pilas de trabajo y aprendía de sus errores, el grupo «cualitativo» teorizó sobre la perfección y, a fin de cuentas, tenía poco que mostrar, a no ser por teorías grandiosas y un montón de barro sin procesar.

La destreza resulta de una práctica consistente, deliberada. En las primeras etapas de la práctica, la cantidad y la velocidad triunfan plenamente sobre la calidad total. Si practicas veloz y constantemente, adquirirás la destreza en menos tiempo.

Esto no significa descuidar las formas al practicar. Algunas destrezas, particularmente las que implican movimientos o acciones físicas, requieren de cierto tiempo para realizarse bien. Si practicas tu técnica pictórica para el retrato, de nada te servirá pintar cien lienzos a la Jackson Pollock. La técnica sí importa.

Primero asegúrate de practicar mediante técnicas suficientemente buenas para alcanzar tu nivel de desempeño deseado. Cuando logras practicar bien

80 o 90 por ciento del tiempo, aumenta la velocidad para obtener la destreza.

Ahí lo tenemos: 10 simples principios que te ayudarán a practicar tu destreza de la manera más eficiente y efectiva posible.

¿Funciona?

¿Servirá este método para adquirir habilidades más rápidamente? Los datos de las investigaciones realizadas dicen que sí, de modo absoluto.

En estudios académicos sobre la adquisición de habilidades motoras y cognitivas, los investigadores han notado un patrón común: cuando los participantes en el estudio empezaron a practicar una nueva destreza, su desempeño mejoró de modo notable en muy poco tiempo. No se requiere de mucha práctica para pasar de «lento incompetente» a alguien «razonablemente rápido y notablemente competente».

En los libros, esto suele llamarse «la regla de la práctica poderosa», y la encontramos una y otra vez. Este efecto es conocido por los investigadores que analizan la adquisición de destrezas desde 1926, por lo menos^[8], y ha sido repetido muchas veces en estudios sobre las habilidades físicas y mentales^[9]. Un estudio llegó incluso a afirmar que «cualquier teoría sobre la adquisición de destrezas que no se avenga a la regla de la práctica poderosa, puede rechazarse inmediatamente^[10]».

Los estudios académicos representan esta ley con una gráfica parecida a esta:



Dado que el tiempo es un factor cuantitativo que se incrementa, la curva se aplana. Con la práctica, toma menos tiempo completar una labor determinada.

Es interesante señalar que, si rebautizamos el eje «y» como «qué tan bueno eres» (esto es, defines el desempeño en términos más generales, comparado con una unidad de tiempo), obtienes la famosa *curva de aprendizaje*:



El patrón general de la curva de aprendizaje luce así: cuando comienzas eres malísimo, pero mejoras rápidamente al aprender las partes más importantes de la destreza. Después de alcanzar cierto nivel de habilidad muy rápido, tu tasa de mejoría disminuye y la subsecuente se hace mucho más lenta.

En contra de lo que la gente cree, las curvas de aprendizaje muy pronunciadas son buenas, no malas. La gráfica aclara la razón para ello: las curvas de aprendizaje pronunciadas indican un ritmo muy acelerado de adquisición de la destreza. Mientras más pronunciada es la curva, más mejoras por unidad de tiempo.

Puedes pensar en los diez pasos antes abordados como si fueran una manera de lograr que tu curva de aprendizaje sea más pronunciada. Los principios mismos son técnicas simples que hacen que las dos primeras etapas teóricas del proceso de adquisición de destrezas (cognición y asociación) resulten más sencillas en la práctica.

Cuando empiezas a practicar algo nuevo, tus habilidades mejorarán natural y notablemente en un periodo muy corto. El truco consiste en

practicar lo antes posible. No se trata de *pensar* en la práctica o de *preocuparte* por ella, sino de practicar y punto.

Es fácil creer que estás invirtiendo mucho tiempo en una destreza cuando en verdad no la practicas. Si desde hace mucho tiempo deseas aprender algo, sueñas con ser bueno en ello, pero si dudas en iniciar las cosas, toda esta energía mental y emocional puede desperdiciarse sin mejoría. Si no sabes a dónde vas y no tienes una estrategia sólida para llegar allí, puedes desperdiciar una cantidad enorme de energía en un deambular improductivo.

Los diez principios están diseñados para ayudarte a eliminar esta actitud improductiva y remplazarla con actividades fundamentales para la adquisición de destrezas. Si inviertes más tiempo y energía en las primeras dos fases del proceso, y menos en las actividades que no te ayudan, lograrás adquirir la habilidad en un plazo menor. Así de simple.

¿Y qué hay de la inmersión?

Esta no es la única vía para adquirir nuevas destrezas, pero sí la más flexible. Otros métodos pueden producir resultados similares, pero implican sacrificios más significativos.

El método más conocido para adquirir rápidamente habilidades nuevas es la *inmersión*, es decir, cambiar completamente tu ambiente de modo que las cosas resulten en una práctica deliberada constante. Si quieres aprender francés, por ejemplo, hacerlo por inmersión equivale a vivir en Francia durante semanas o meses.

En general, la inmersión funciona. Si vas a Francia, te verás obligado a practicar tus habilidades de comunicación oral todo el día. Tras unos cuantos días de frustrante adaptación, te percatarás de que tus destrezas mejoran con celeridad.

La inmersión funciona porque asegura completar las primeras horas cruciales de práctica sin fallar: no puedes escapar al medio ambiente, así que la práctica se realiza de modo automático.

La desventaja de la inmersión es que, por lo regular, es necesario que la destreza sea tu principal centro de atención por mucho tiempo. Si dejar todos tus compromisos para ir a Francia es opción, aprender francés vía la inmersión es una buena estrategia.

Por desgracia, la mayoría de nosotros tenemos compromisos que no podemos (o no queremos) cancelar: familia, trabajo, pagos de la hipoteca y demás. En estos casos, la inmersión puede ser difícil o imposible.

En el peor de los escenarios, la idea de la inmersión se convierte en una barrera activa: si sigues esperando una oportunidad para la inmersión antes de comprometerte a aprender una nueva destreza, puedes desperdiciar años de valioso tiempo.

Toma las oportunidades para la inmersión conforme lleguen, pero no estés seguro de ellas. Las técnicas que yo te ofrezco están diseñadas para ayudarte a adquirir nuevas destrezas, incluso si solo dispones de una o dos horas libres al día.

Reactivar viejas habilidades

También es importante observar que estos principios son útiles incluso cuando la destreza que tratas de adquirir no es completamente nueva para ti. Es posible usar estas técnicas para volver a adquirir viejas habilidades en un tiempo récord.

Por ejemplo, aprendí a tocar trompeta en la preparatoria. Practiqué hasta ser bastante bueno. A partir de que salí de la prepa para ir a la universidad, dejé de practicar por completo.

Si me decidiera a retomar la trompeta, no me tomaría mucho tiempo reactivar la destreza. Ya conozco cuáles son las subdestrezas requeridas, así que puedo concentrarme en la técnica de embocadura (el control de los músculos alrededor de los labios mientras se sopla en la boquilla), en la lectura de notas, en la posición de los dedos o en el repaso de la teoría musical básica (la dinámica, el tempo, los *beats* y la expresividad).

Con solo unas horas de práctica podría readquirir la destreza. La reactivación solo requeriría de mi tiempo, eliminar las barreras que dificultan la práctica y una práctica cronometrada.

Un buen comienzo constituye la mitad del éxito

A veces querrás dejar la guitarra. Odiarás la guitarra. Pero si persistes, tendrás la recompensa.

Jimi Hendrix, reconocido guitarrista.

No necesitarás usar cada uno de los principios para cada destreza que adquieras, pero algunos te serán útiles y hasta esenciales en todo momento.

Para mí es útil pensar en estos principios como si fueran una lista de verificación. Cuando decidas aprender algo nuevo, solo repasa la lista y decide qué principios aplicar a tu proyecto particular.

He aquí la lista para lograr la rápida adquisición de destrezas:

1. Elige un proyecto que ames realmente.
2. Concentra tu energía en una sola habilidad en un momento determinado.
3. Define tu nivel de desempeño.
4. Deconstruye la destreza hasta dar con los elementos que la componen.
5. Obtén las herramientas indispensables.
6. Elimina las barreras que te impiden practicar.
7. Dedica un tiempo a la práctica exclusiva de la habilidad.
8. Crea ciclos de retroalimentación rápida.
9. Practica en periodos breves con ayuda de un reloj.
10. Enfatiza cantidad y velocidad.

Eso es todo. Prueba esta lista con la destreza que actualmente te ocupa y tu práctica será más efectiva y eficiente, lo que te permitirá adquirir la habilidad con más rapidez.

Como ya he dicho, este método no es física nuclear. Es mero sentido común, estrategia y preparación, todo ello aplicado a una destreza que quieres mejorar. Nada más y nada menos.

Ahora, examinemos cómo el aprendizaje y la investigación pueden hacer que tu proceso de adquisición de habilidades sea todavía más efectivo.

Los diez principios del aprendizaje efectivo

Ningún problema puede resistir el asalto del pensamiento sostenido.

Voltaire.

Como vimos en el primer capítulo, aprender no es lo mismo que adquirir una destreza. Sin embargo, eso no significa que el aprendizaje no sea importante. Investigar un poco antes de dar paso a la práctica puede ahorrarte tiempo, energía y desgaste emocional.

Aprender hace que la práctica sea más eficiente, lo que te permite utilizar más tiempo de tu práctica trabajando primero en las subdestrezas más importantes.

He aquí los diez principios del aprendizaje efectivo:

1. Investiga sobre la destreza y los temas relacionados.
2. Lánzate de cabeza.
3. Identifica modelos y ganchos mentales.
4. Imagina lo opuesto a lo que deseas.
5. Habla con practicantes de la destreza para definir tus expectativas.
6. Elimina las distracciones ambientales.
7. Usa la repetición espaciada y el reforzamiento para memorizar.

8. Crea rutinas y listas de verificación.
9. Haz predicciones y somételas a prueba.
10. Honra tu biología.

Investiga sobre la destreza y los temas relacionados

Pasa veinte minutos en la red, en una librería o en los anaqueles de tu biblioteca local, buscando información sobre la destreza. La meta es identificar al menos tres libros, tres DVD, cursos y demás materiales que parezcan relacionados con la habilidad que tratas de adquirir.

Antes de entrar en pánico, comprende que no tienes por qué pasar horas memorizando estos materiales. Por el contrario: el tiempo invertido en leer y observar no cuenta como tiempo de práctica.

No se trata de atiborrarte de conocimientos, como haces para los exámenes. La intención en esta investigación inicial es identificar las subdestrezas más importantes, los componentes críticos y las herramientas necesarias para practicar tan pronto como puedas. Mientras más sepas de antemano sobre la destreza, más diligente será tu preparación. La meta es reunir un amplio cuerpo de información sobre la habilidad tan rápido como puedas, creando así una panorámica que te dará una buena idea del proceso que comenzarás.

Para la rápida adquisición de destrezas, echar una ojeada es mejor que la lectura profunda. Al detectar ideas y herramientas que aparecen una y otra vez en diferentes textos, obtendrás patrones confiables que te ayudarán a preparar la práctica.

Si quieres cocinar el *croissant* perfecto, consigue algunos buenos libros relacionados con la panadería. En lugar de reinventar el proceso, encontrarás técnicas preexistentes perfeccionadas durante muchos años por los maestros panaderos. Si ves que la misma técnica o proceso es descrita en varias fuentes, hay buenas posibilidades de que sea algo que debes conocer.

Cuando encuentres las técnicas más útiles, experimenta con ellas en tu propia cocina, con lo que te ahorrarás una buena cantidad de prueba y error.

Lánzate de cabeza

Parte de tu investigación inicial contendrá conceptos, técnicas e ideas que no entenderás. Por lo regular, encontrarás algo que da la impresión de ser importante, pero no tendrás idea de qué significa. Leerás palabras que no reconoces y verás que los practicantes hacen cosas que te resultan extrañas.

No temas. La confusión inicial es normal. De hecho, es grandiosa. Dirígete a la confusión.

La investigación temprana es una de las mejores maneras de identificar las subdestrezas esenciales y las ideas, pero no es muy probable que sepas de qué se tratan todavía. El significado llegará después, cuando comiences a practicar.

El doctor Stephen Krashen, el experto en aprendizaje de idiomas que mencioné antes, llama a esto *ruta de la comprensión*. Desde luego, la nueva información no es muy comprensible, dado que no está relacionada con algo conocido o experimentado antes. Con el paso del tiempo, la misma información será comprensible gracias a la experiencia. En palabras del renombrado maestro de yoga T. K. V. Desikachar: «El reconocimiento de la confusión es, en sí mismo, una forma de claridad».

Darte cuenta de que estás confundido es valioso. Reconocer la confusión puede ayudarte a definir con exactitud qué te confunde, lo que sirve para determinar qué debes investigar o hacer para resolver la confusión.

Si no llega la confusión por lo menos a la mitad de tu investigación, no estás aprendiendo tan rápido como puedes hacerlo. Si comienzas a sentirte intimidado o dudoso respecto al ritmo de trabajo que pones en práctica, estás en el camino correcto. Asumiendo que trabajas en un problema o proyecto fascinante, mientras más confundido estés al principio mayor presión interna sentirás para aclarar las cosas. Y aprenderás más rápido.

Rechazar el deseo de lanzarte de cabeza constituye la más grande barrera emocional que impide la rápida adquisición de destrezas. Sentirse estúpido no es divertido, pero si recuerdas que con la práctica entenderás, pasarás de la confusión a la claridad en poco tiempo.

Identifica modelos y ganchos mentales

Al investigar, naturalmente notarás patrones: ideas y técnicas que surgen una y otra vez.

Estos conceptos son llamados *modelos mentales*, y son muy importantes. Los modelos mentales son la unidad más básica del aprendizaje: una manera de entender y de etiquetar un objeto o relación en el mundo. Conforme reúnas modelos mentales precisos, será más sencillo anticipar lo que sucederá si actúas en un sentido determinado. Los modelos mentales también facilitan la discusión de tus experiencias con otras personas.

He aquí un ejemplo: hace poco, ayudaba a mi padre a poner un sitio web. Mientras trabajaba, traté de explicarle qué estaba haciendo. Al principio, el asunto resultó frustrante para ambos: yo usaba palabras como «servidor» y él no tenía la menor idea de qué hablaba.

Cuando mi padre aprendió que un servidor es una computadora especial que hace llegar una página web a la gente que la solicita, y que ese servidor era una máquina distinta a la que usábamos, le resultó mucho más fácil comprender qué hacíamos. En este caso, el servidor es un modelo mental: cuando se está familiarizado con el término, es más fácil comprender el proceso de publicación de un sitio web.

También notarás algunas cosas que se parecen a las ya familiares. Se trata de *ganchos mentales*: analogías y metáforas que puedes utilizar para recordar nuevos conceptos.

En el caso de los servidores para la red, imagina a un bibliotecario. Cuando vas a la biblioteca y pides un libro específico, un bibliotecario buscará en repisas que contienen miles de libros para encontrar el que buscas. Cuando encuentra el libro, él o ella te lo llevan. Si no lo encuentra, te dirá: «No puedo encontrar el libro que usted busca».

Así justamente funcionan los servidores de la red. Cuando pides una página web específica, el servidor buscará esa página en la memoria. Si la encuentra, te la enviará. Si el servidor no la encuentra, te enviará un mensaje que dice: «Error 404: Página no encontrada». Pensar en el servidor como si se tratara de un «bibliotecario cibernético», es muy útil cuando se enseña a alguien el funcionamiento del sistema.

Mientras más modelos y ganchos mentales identifiques en tu investigación temprana, más fácil te será utilizarlos en la práctica.

Imagina lo opuesto a lo que deseas

Una forma en apariencia poco intuitiva de conocer alguna nueva actividad es imaginar el desastre y no la perfección.

¿Qué pasaría si hicieras todo mal? ¿Qué tal si el resultado es el peor posible?

Estamos ante una técnica para la resolución de problemas llamada «inversión», y es útil para aprender lo esencial de casi cualquier actividad. Al estudiar lo opuesto a lo que deseas, identificas elementos importantes que no resultan obvios en primera instancia.

Pensemos en el descenso de rápidos en kayak. ¿Qué debería saber para ser capaz de descender en kayak en aguas turbulentas y rocosas?

He aquí la inversión: ¿cómo serían las cosas si todo saliera mal?

- Mi kayak volcaría y no podría volver a la posición correcta.
- La canoa se inundaría para después hundirse, lo que implicaría la pérdida total del kayak.
- Podría perder el remo, lo que me impediría maniobrar.
- Podría golpearme la cabeza con una roca.
- Saldría expulsado del kayak, quedaría atorado en una turbulencia (un punto en el río en que el agua gira, como si se tratara de una lavadora) y no podría salir de ella.

Si me las arreglara para cometer todos estos errores en un río de aguas furiosas, probablemente moriría: el peor escenario.

Esta línea de pensamiento negativa es útil porque se refiere a habilidades para la navegación de los rápidos que, probablemente, sean muy importantes:

- Aprender a rodar el kayak para volver a la posición correcta sin salir de la canoa.
- Aprender a evitar que se hunda el kayak si es necesario salir de este.
- Aprender a no perder el remo en aguas turbulentas.
- Aprender a dar la vuelta a grandes rocas con cuidado.
- Inspeccionar el río antes de lanzarte al agua para evitar los peligros por entero.

Esta simulación mental también me proporciona una lista de compras a realizar: necesito invertir en un chaleco salvavidas, un casco y otros

elementos de seguridad.

Ahora, en lugar de pensar en 1: sobrevivir 2: los rápidos 3: divertirme, tenemos una lista de subdestrezas que debemos practicar para asegurarnos de que en verdad nos divertiremos, conservando el equipo y sobreviviendo a la aventura.

El sistema de inversión funciona.

Habla con practicantes de la destreza para definir tus expectativas

El aprendizaje inicial te ayuda a tener expectativas apropiadas: ¿cuál es el desempeño típico del principiante?

Cuando te lanzas a adquirir una nueva destreza, es muy común subestimar la complejidad de la empresa, o el número de elementos involucrados que deben considerarse para tener un buen desempeño. Si la habilidad implica prestigio social, la mística asociada puede ocultar las expectativas tempranas.

Muchos aspirantes a estrellas de *rock* han conseguido una guitarra eléctrica para darse cuenta de que es extremadamente difícil tocarla bien. También se percatan de que no es tan fácil cantar entonado mientras se luce maravillosamente bien. Parte del problema radica en que «ser una estrella de *rock*» no significa una sola habilidad, implica muchas subdestrezas relacionadas y que cada una requerirá de práctica para dominarla.

Hablar con gente que ya tiene la habilidad antes de comenzar te ayudará a liberarte de mitos antes de invertir tiempo y esfuerzo en el asunto. Al estar consciente de qué puedes esperar en el proceso, te será mucho más sencillo mantener el interés en la práctica y evitar el desencanto inicial.

Elimina las distracciones ambientales

Las distracciones son el enemigo número uno de la rápida adquisición de destrezas. Las distracciones matan la práctica concentrada, y la falta de práctica concentrada lleva a una lenta (o nula) adquisición de la habilidad. Puedes anticipar este problema dedicando unos minutos a anticipar y eliminar (o reducir) tantas distracciones como puedas antes de practicar.

Las principales fuentes de distracción aparecen en dos modalidades: electrónicas y biológicas.

Tu televisión, tu teléfono e Internet son distracciones electrónicas. Apágalas, desconéctalas, bloquéalas o elimínalas de tu ambiente mientras practicas, a menos que sean absolutamente necesarias para la práctica misma.

Los familiares y colegas bienintencionados, y hasta las mascotas, constituyen distracciones biológicas. No puedes desconectar a la gente, pero puedes hacerles saber que no estás disponible mientras practicas, lo que da mayores posibilidades de que respeten tu tiempo de práctica sin interrumpir.

Mientras menos distracciones tengas al practicar, más rápido adquirirás la destreza.

Usa la repetición espaciada y el reforzamiento para memorizar

Para usar el material aprendido en la práctica, debes recordar las cosas rápidamente. Muchas destrezas requieren, cuando menos, algún tipo de memorización.

Aquí está el detalle: tu memoria no es perfecta. Cada vez que aprendes algo nuevo, es muy probable que lo olvides, a menos que repases el concepto durante un periodo determinado. Esta repetición refuerza la idea y ayuda a que tu cerebro consolide la habilidad para integrarla a la memoria de largo plazo.

Los investigadores han descubierto que la memoria sigue una curva descendente: los nuevos conceptos deben ser reforzados regularmente; mientras más tiempo hayas conocido un concepto, menos tendrás que repasarlo para que el recuerdo sea preciso.

La *repetición espaciada* y el *reforzamiento* son una técnica de memorización que te ayuda a revisar sistemática y regularmente conceptos e información importantes. Las ideas difíciles de recordar se repasan regularmente, en tanto que las más sencillas o las que ya se aprendieron hace tiempo son revisadas con menor frecuencia.

Los programas que despliegan tarjetas para mejorar la memoria, como Anki^[11], SuperMemo^[12] y Smarttr^[13], hacen que la repetición espaciada y el reforzamiento resulten muy simples. Los sistemas de repetición espaciada se basan en «cartas» digitales desplegadas; tú mismo debes crear las cartas o

tarjetas. Al hacerlo, deconstruyes la habilidad, matando así dos pájaros de un tiro.

Cuando ya creaste las tarjetas digitales, solo debes invertir unos minutos al día en repasarlas. Al sistematizar el proceso de repaso y recuerdo, estos métodos pueden ayudarte a aprender nuevas ideas, técnicas y procesos en tiempo récord. Si repasas tus tarjetas de manera consistente, memorizarás los conceptos necesarios y las ideas a gran velocidad.

Es importante hacer notar que la adquisición de habilidades suele requerir de mucho más involucramiento que el aprendizaje académico. Si tu objetivo primario es memorizar conceptos, ideas o vocabulario para pasar un examen, no necesitarás mucho más que la repetición espaciada para lograrlo^[14].

Para que esta técnica sea realmente eficiente, procura recordar la información cada determinado tiempo. Si aprendes palabras comunes para integrar tu vocabulario en otra lengua, la repetición espaciada (realizada cada determinado tiempo) y el reforzamiento serán muy valiosos. En asuntos en los que recordar rápidamente no es crucial, suele ser mejor idea dejar a un lado las tarjetas para centrarse en el tiempo de práctica y la experimentación.

Crea rutinas y listas de verificación

Muchas destrezas implican algún tipo de rutina, preparación, mantenimiento, etcétera. Crear un sistema simple es la mejor manera de asegurar que estos importantes elementos tengan un lugar con el menor esfuerzo posible.

Las *listas de verificación* son útiles para recordar cosas que deben hacerse cada vez que practicas. Son una forma de sistematización del proceso, lo que te permite concentrar tu atención en cuestiones más importantes.

Las *rutinas* son estructuras que te aseguran hacer siempre las cosas de la misma forma. Piensa en el jugador de basquetbol que establece una rutina antes de realizar un tiro libre: se seca las manos en los *shorts*, afloja los hombros, recibe la pelota del árbitro, la rebota tres veces, espera tres segundos y dispara. Eso es una rutina.

Crear rutinas y listas de verificación hace que tu práctica sea más eficiente. También permiten que sea más fácil visualizar tu práctica, lo que te permite ensayar mentalmente, y esto es muy útil en algunas prácticas físicas.

Haz predicciones y somételas a prueba

Parte del proceso de adquisición de habilidades implica experimentar: poner en práctica cosas nuevas para verificar su funcionamiento.

La verdadera prueba para el aprendizaje útil es la predicción. Con base en lo que sabes, ¿puedes anticipar los resultados de un cambio o experimento? Realizar y poner a prueba predicciones te ayudará a adquirir habilidades más rápidamente. Es una variante del método científico y está integrada por cuatro elementos clave:

- Observaciones: ¿qué observas en este momento?
- Conocimientos: ¿qué sabes ya sobre el tema?
- Hipótesis: ¿qué crees que puede mejorar tu desempeño?
- Pruebas: ¿qué piensas intentar a continuación?

Recomiendo usar un cuaderno para llevar un registro de las hipótesis que pones a prueba mientras practicas. Al llevar un recuento de tus predicciones y de las nuevas ideas generadas, tendrás más experimentos fructíferos que realizar.

Honra tu biología

Tu cerebro y tu cuerpo son sistemas con necesidades biológicas: comida, agua, ejercicio, descanso y sueño. Es muy fácil extralimitarte, lo cual es contraproducente. Sin los insumos adecuados, tu cuerpo y tu mente no podrán dar productos útiles.

Según Tony Schwartz, autor de *The Power of Full Engagement* (2004), y de *Be Excellent at Anything* (2011), el ciclo óptimo de aprendizaje parece consistir en 90 minutos de atención concentrada. Después, tu mente y tu cuerpo necesitarán un descanso. Usa ese tiempo para ejercitarte, descansar, comer, tomar una siesta o hacer otra cosa.

Este principio coincide con la práctica cronometrada. Al poner tu cronómetro entre 60 y 90 minutos de práctica o investigación, te será más fácil acordarte de descansar llegado el momento.

También puedes dividir tu tiempo de práctica en partes más pequeñas, con un descanso breve a mitad de la práctica si es necesario: 20 minutos de

práctica, diez de descanso, 20 de práctica, diez de descanso, etcétera.

Preparar las cosas

Aquí no hay reglas. Estamos tratando de lograr algo.

Thomas Alva Edison, inventor.

No tendrás que usar todos los principios en cada habilidad adquirida, pero siempre encontrarás que algunos son esenciales para hacer las cosas bien.

En mi caso, ha sido muy útil pensar en estos principios como una lista de verificación secundaria. Siempre que decidas adquirir una nueva destreza, pasa revista a esta lista y decide qué principios aplicar a tu proyecto.

He aquí una lista de verificación con los elementos que conforman el aprendizaje efectivo:

1. Investiga sobre la destreza y los temas relacionados.
2. Lánzate de cabeza.
3. Identifica modelos y ganchos mentales.
4. Imagina lo opuesto a lo que deseas.
5. Habla con practicantes de la destreza para definir tus expectativas.
6. Elimina las distracciones ambientales.
7. Usa la repetición espaciada y el reforzamiento para memorizar.
8. Crea rutinas y listas de verificación.
9. Haz predicciones y somételas a prueba.
10. Honra tu biología.

Eso es todo: usa esta lista para adquirir la habilidad que ahora te interesa y aprenderás lo necesario para practicar de modo eficiente y efectivo.

Llevar a la práctica la teoría

Qué vano resulta sentarse a escribir cuando no se ha estado de pie para vivir.

Henry David Thoreau.

Ya estuvo bien de la teoría. Ha llegado la hora de practicar.

Ya abordamos los puntos básicos de la rápida adquisición de destrezas, pero saber cómo hacer las cosas es muchísimo menos importante que hacerlas. Recuerda: sin práctica no hay adquisición de destrezas.

En lugar de hablar y hablar sobre la teoría de la adquisición de habilidades, te demostraré cómo hacerlo. Pienso utilizar estos principios para adquirir varias habilidades nuevas y tú tendrás un asiento de primera fila.

He aquí las destrezas que pretendo adquirir:

- Yoga: desarrollar una práctica casera de las posturas.
- Programación: crear una aplicación funcional para la red.
- Mecanografía: reaprender a teclear sin mirar el teclado, usando una disposición especial de las teclas.
- Estrategia: jugar Go, el juego más antiguo y complejo del mundo.
- Música: tocar el ukulele.
- Surf: navegar a vela y maniobrar en aguas tranquilas.

No tengo experiencia en ninguna de estas actividades. Usando los métodos y técnicas que acabo de describir, pretendo adquirir cada una de estas habilidades en 30 días o menos. El tiempo estimado para cada una es de aproximadamente 20 horas, para lo cual se requieren entre 60 y 90 minutos de práctica por día.

Sobre estos ejemplos

Estas destrezas en particular son por completo personales. Se trata de cosas que me interesa aprender por varios motivos, que explicaré en detalle más adelante. Tal vez las destrezas que te interesan a ti sean distintas, pero el proceso esencial de adquisición de la habilidad será semejante.

Espero que, al enseñarte cómo utilicé este proceso para adquirir muchas habilidades diferentes y en diversas áreas, tengas una comprensión más cabal de cómo usar estas técnicas o de cómo mejorar las destrezas que te importan.

Si por naturaleza te interesan más uno o dos ejemplos que el resto, está bien. Lee primero esos capítulos. Si empiezas a leer uno y la habilidad te resulta aburrida o ajena a tu situación, siéntete libre de saltártelo. Uso el

mismo método esencial para cada actividad, así no te perderás partes importantes del sistema.

Escribo estos capítulos como tutor, asumiendo que nunca antes tuviste contacto con estas prácticas, así que no tenemos experiencia previa que pueda guiarnos. Además de ilustrar el método, espero que aprendas cosas valiosas sobre estas seis interesantes destrezas que vale la pena practicar.

Si tienes mucha experiencia en alguno de estos temas, es probable que adviertas algún error en mi comprensión, en el lenguaje o que de plano estés en desacuerdo con mi método. Está bien. Recuerda que estoy comenzando y que *soy un completo novato que (aún) no sabe qué hacer*.

Lo que estás a punto de leer es un repaso de mi proceso de aprendizaje para cada una de estas destrezas. Me he esforzado en asegurarme que la información contenida en este libro sea precisa y completa, pero no estoy libre de cometer errores. En cualquier caso, lo importante en realidad es el método de investigación y la práctica deliberada.

Empecemos por el yoga.

4

Yoga

*Lección: no lo hagas más difícil
de lo necesario.*

No mido mi progreso en el yoga según lo mucho que pueda doblarme o torcerme, sino de acuerdo con lo bien que trato a mi esposa e hijos.

T. K. V. Desikachar, reconocido maestro de yoga.

Para obtener imágenes, videos y comentarios adicionales sobre este capítulo, visita <http://first20hours.com/yoga>.

Me estoy haciendo viejo.

Aclaro: todavía soy joven, relativamente hablando, pero comienzo a notar algunas cosas en mi cuerpo que me preocupan.

Cuando me despierto por la mañana, suelo tener un dolor a media espalda. Después de un largo día sentado ante la computadora, mi cuello y hombros se sienten tensos y duelen. No es una sensación placentera.

Hasta este momento, no he pensado mucho en mi cuerpo. Ha funcionado bien. Tras mi salida de la preparatoria, dejé de practicar deportes y no me ejercito en modo alguno. Cuando entré a la universidad, traté a mi cuerpo como si fuera un simple vehículo cuyo único propósito fuera llevar mi cerebro de una clase a otra.

Las cosas no han cambiado mucho desde entonces, pero hasta este momento me he sentido bien. Ahora, esos años de negligencia física me pasan la factura.

No estoy en forma y lo sé.

«En verdad creo que debes intentar con el yoga...»

Kelsey, mi esposa, ha practicado yoga desde la universidad. Casi a diario, camina por la calle hasta el estudio de yoga de la localidad y pasa una hora o más contorsionándose en varias posiciones en un cuarto muy cálido. Ama hacerlo, y suele regresar a casa feliz y relajada.

Cada semana, Kelsey solía decirme: «En verdad creo que debes intentar con el yoga. Es grandioso. Te gustaría mucho».

Yo dudaba. Para mí, el yoga no era muy atractivo.

Y nada tenían que ver los estiramientos. Yo practicaba atletismo en la preparatoria, así que los ejercicios de estiramiento no eran problema. Mi prueba favorita era la de 110 metros con obstáculos, que requiere de un buen nivel de flexibilidad^[15].

El entrenamiento para la carrera con obstáculos implica muchos estiramientos y aún conservo buena parte de la flexibilidad desarrollada esos años, especialmente en los tendones de las corvas. Incluso hoy, una década después, puedo poner las palmas de las manos en el suelo sin doblar las rodillas.

Lo que me hacía dudar respecto al yoga era, lisa y llanamente, lo extraño que me resultaba.

Chakras, auras y kundalinis... ¡Dios mío!

Shit Yogis Say^[16], un video producido por lululemon^[17], una de las empresas más importantes del mundo en lo que a ropa deportiva se refiere, es un buen

ejemplo de cómo le suena el yoga a quien no está iniciado en sus secretos. He aquí algunos aspectos destacados:

- «Me preocupa tu aura...»
- «¿Cómo se dice eso en sánscrito?»
- «Mis chakras están *tan* alineados.»
- «¿Quieres ver hasta dónde levanto la pierna?»

El video ha sido visto más de 2 millones de veces desde que lo subieron a Youtube, en diciembre de 2011, lo que me llamó la atención.

Me considero bastante racional y práctico. Estoy dispuesto a creer que el estiramiento regular es bueno para la salud. Aprender una secuencia de estiramientos me parece razonable.

Pero la verdad es que todo ese asunto de las auras, los chakras y demás devocionarios esotérico-espirituales, me disgusta. No sé qué tiene que ver todo esto con el ejercicio y, francamente, la onda místico-jipiosa hace que el yoga parezca un tipo de culto raro.

Hay más. La práctica del yoga en Estados Unidos ha evolucionado hasta convertirse en el centro de un negocio que representa unos 8 mil millones de dólares al año. Demostrar qué tan «yogui» eres se convirtió en una señal de estatus social tan molesta como usar la última bolsa de mano de Louis Vuitton.

No, gracias. Mis «chakras» están bien. Muchas gracias.

«Relaja tu rostro»

Con el tiempo, Kelsey me convenció de practicar yoga en cumplimiento a una de esas promesas que se hacen el último día del año. Yo trabajaba en casa, por lo que la actividad serviría además para sacarme del apartamento y conocer gente. Yo estaba escéptico, pero decidí seguirle la corriente con humor.

Nos hicimos miembros de Pure Yoga, un estudio nuevo, de alto nivel, en la zona este de Nueva York. Muchas clases eran impartidas por Marco Rojas, el maestro favorito de Kelsey y uno de los mejores instructores de la ciudad.

Imagina una versión más joven y esbelta de Antonio Banderas vestida con ropa para yoga y ahí tienes a Marco. Es guapo, carismático y tiene una voz magnífica con acento hispano. Sus clases siempre están atiborradas, debido a que es un maestro genuinamente excepcional y a que 99 por ciento de las practicantes de la ciudad están enamoradas de él. (¿Quién puede culparlas?).

Dado que era novato en el yoga, hice mi mejor esfuerzo al seguir la clase de Marco. No conocía las posturas, ni lo que cada una requería o significaba, así que miré lo que los demás hacían y traté de imitarlo. Se requería de muchísima concentración. Podía hacer las cosas, pero hacerlo era una lucha. Algunas posturas, como los parados de cabeza, estaban más allá de mis capacidades.

Procuré quedarme al fondo del salón para ver lo que todos hacían, y así también quedaba bajo control el daño colateral en caso de que me cayera, lo que sucedió más veces de lo que estoy dispuesto a aceptar.

Disfrutaba las clases de Marco, pero no lograba asimilar lo que pasaba. Marco se acercaba regularmente a mi tapete de práctica para corregir mis posturas.

En una ocasión memorable, Marco apoyó mis pies con los suyos, de modo que quedaron perpendiculares al suelo. Luego, tomó mis manos y jaló con fuerza.

«Usa tus cuádriceps... pivotea con las caderas. Bien. Ahora... relaja tu rostro».

Yo gesticulaba y tenía buenas razones para ello: mis muñecas tocaban los dedos de mis pies. Relajé mi rostro.

El dilema del «amo de casa»

Por mucho que me gustaran los cursos de Marco, me resultaba difícil reservar el tiempo necesario para ellos. Cada clase duraba hora y media, sin incluir el tiempo que me tomaba caminar hasta el estudio, cambiarme, bañarme después de la clase y regresar a casa.

Aunque el estudio estaba solo a una cuadras de distancia, necesitaba en realidad dos horas sólidas para las clases. En aquel tiempo, yo trabajaba en casa para un cliente muy exigente, una empresa de las que figuran en la lista

de las más importantes, según la revista *Fortune*, y no era raro que se agendaran reuniones que coincidían con la clase. No ayudaba el hecho de que las clases se daban a media mañana o a media tarde, lo que me obligaba a dejar trabajo pendiente. Hablando en general, mis compromisos de trabajo eran flexibles, por lo que se trataba más de un problema psicológico: me costaba trabajo convencerme de que era aceptable ir a contorsionarme por dos horas cuando tenía trabajo pendiente.

A fin de cuentas, asistí a unas 15 clases ese año. Aunque me sentía muy bien después de cada una, simplemente no podía justificar las dos horas necesarias.

Además, no estaba practicando en casa. No sabía cómo hacerlo. Durante la clase, imitaba a los otros estudiantes. Podía desenrollar mi tapete en casa y hacer estiramientos, pero en mi mente eso no era yoga.

Cuando el estudiante está listo, aparece el maestro

Después de mudarnos a Colorado, Kelsey mencionó una noche que extrañaba Nueva York. Extrañaba las clases de Marco y el curso de anatomía del yoga de Leslie Kaminoff, al que acudió cuando terminó con su certificación como maestra de yoga. Leslie es coautora de *Yoga Anatomy* (primera edición, 2007; segunda edición, 2011), uno de los libros de yoga más vendidos de todos los tiempos. Su libro se utiliza en los programas de entrenamiento en todo el mundo. Desafortunadamente, para tomar el curso de Leslie, debes vivir en Nueva York, de modo que Kelsey tuvo que abandonar esas clases cuando nos mudamos.

«Alguien debería poner el curso de Leslie en línea», dijo ella.

He aquí la ironía. Unas semanas antes, Kelsey me había ayudado a lanzar mi primer curso en línea, de manera que sabía cómo hacerlo. También tenía razón en cuanto al curso de Leslie: dado que solo estaba al alcance de los alumnos residentes en Nueva York y no de los estudiantes interesados de todo el mundo, tendría mucho sentido crear el curso en línea.

Yo señalé lo obvio: Kelsey era perfectamente capaz de crear el curso, siempre y cuando Leslie autorizara la producción. Esa misma noche, Kelsey le mandó a Leslie una propuesta.

Dos días después Kelsey y Leslie estuvieron de acuerdo en seguir adelante con el proyecto y así fue como Kelsey fundó su primer negocio, una empresa en línea para la producción de cursos de yoga.

Debido a que los dos trabajábamos en casa, de pronto comencé a escuchar mucho sobre yoga. Parte del proceso de producción implica ver las filmaciones de cada clase y elaborar notas detalladas para el editor y para quienes transcriben el texto. Como resultado, terminé aprendiendo fragmentos del curso de Leslie mientras ella lo producía.

Cuando decidí tomar el yoga en serio

Una de las primeras cosas que me llamó la atención fue el tono de Leslie. No sonaba como un jipi alivianado, como yo esperaba. Por el contrario, con sus tres décadas de experiencia en anatomía, fisiología y medicina deportiva, era poco tolerante con la terminología rimbombante.

Desde la perspectiva de Leslie, el yoga es útil porque, como práctica, funciona muy bien para dar fuerza, para aumentar la flexibilidad y mantener el rango de movimiento.

El yoga es aún más eficiente cuando te concentras en la respiración. Muchos piensan que el yoga es una extraña combinación de aeróbics, gimnasia y contorsionismo, pero esa descripción no es adecuada. El yoga es el yoga porque combina respiración, movimiento y estado mental alerta.

Leslie también explica algunas rarezas relacionadas con la terminología de esta disciplina. Los primeros practicantes experimentaban con la mente y el cuerpo. Trataban de hacer ciencia antes de formalizarse con disciplinas como anatomía, biología y psicología cognitiva.

Como resultado, cuando estos primeros practicantes encontraron algo interesante, su única opción consistió en explicar el hallazgo usando metáforas. Por ejemplo, cuando los practicantes descubrieron que ciertas posturas y técnicas de respiración tendían a provocar determinadas respuestas emocionales, explicaron lo descubierto como mejor pudieron.

Tal es el origen de conceptos como «chakras»: desde el punto de vista científico, no existen las concentraciones de energía en vientre, sacro, plexo solar, corazón, garganta, frente y coronilla. Sin embargo, como metáforas,

ayudaron a los primeros practicantes a hablar de algo que experimentaban internamente. Las metáforas fueron tan útiles que se conservaron.

El último elemento llegó bajo la forma de un video de Youtube que Leslie presentó durante una de sus clases en línea. En el video, Gil Hedley^[18], maestro de anatomía humana, conduce una autopsia con fines educativos ante un grupo de estudiantes^[19]. En un momento dado, Gil habla de un aspecto interesante de la fascia, capas de tejido fibroso que envuelven nuestros músculos.

Al igual que sucede con ligamentos y tendones, la fascia es tejido conectivo: nos mantiene unidos. Al envolver grupos musculares, nos ayuda a movernos al permitir que los músculos se deslicen uno sobre otro con mayor facilidad.

Pero hay un detalle importante: cuando estas capas de fascia están en reposo, algunas hebras de tejido conector (que semejan vagamente la apariencia y consistencia del algodón de azúcar) comienzan a crecer entre ellas.

Normalmente, esto no representa problema alguno. Las hebras son muy delgadas, así que los músculos las rompen fácilmente al deslizarse entre sí. El problema comienza cuando no te mueves lo suficiente. Si no mueves los músculos por un tiempo, las hebras permanecen en su lugar y se acumulan. Cuando este tejido es muy grueso, puede solidificarse, limitando así el rango normal de movimiento.

Sé que no me muevo lo suficiente, así que este video me llamó la atención. ¿Será que mi cuerpo me advierte que los músculos se están llenando de estas hebras y por eso me siento poco ágil? (¡Es estremecedor!).

Es una explicación posible, pero no se trata de la única opción. Según Leslie, el dolor de cuello y espalda también puede ser causado por una falta de oxigenación de los músculos de la zona, un mal llamado isquemia. La falta de movimiento puede causar una reducción del flujo sanguíneo a los músculos. Cuando un músculo no dispone de suficiente oxígeno, sus receptores del dolor comienzan a activarse creando ese dolor sordo y persistente. Mientras más dure la enfermedad, más dolor experimentarás.

El estrés crónico y la limitación de los patrones de respiración también contribuyen a la isquemia. Estas causas suelen actuar en conjunto: es común

que la gente contenga el aliento cuando experimenta estrés. Si ayudas a que tus músculos obtengan más oxígeno moviéndote, respirando y reduciendo tu nivel general de estrés, los dolores terminarán^[20].

Además, el movimiento frecuente y la ingesta de oxígeno son importantes para aliviar el dolor muscular y mantener el rango de movimiento. El yoga combina movimiento, respiración y meditación, porque resulta muy eficaz en reducir o eliminar el dolor muscular crónico.

Eso es suficiente evidencia para mí. Aprenderé yoga.

Ahora, ¿cómo empiezo?

¿Qué es el yoga realmente?

Si voy a practicar yoga, debo tener una idea de qué estoy haciendo. Si el yoga solo consistiera en estiramientos, la gente simplemente le llamaría «estiramientos».

Resulta que la clave para identificar la esencia del yoga está en comprender de dónde viene y cómo conocemos esta práctica con ese nombre en nuestros días^[21].

El yoga ha existido en varias formas durante miles de años. La evidencia más temprana proviene de artefactos recuperados en excavaciones arqueológicas relacionadas con la civilización del valle del Indo, que se asentó en el actual Paquistán y en el noroeste de India, entre los años 3300 a 1300 a. C.

El yoga, como filosofía y como práctica, era un conjunto de técnicas no religiosas relacionadas con tradiciones religiosas y filosóficas de la región. Los antiguos sacerdotes védicos completaban los elaborados rituales para conectar el mundo físico con el divino en busca de Bramah, «el último recinto del ser». La palabra «yoga» proviene del término en sánscrito antiguo *yoke*. Así como un buey es atado a un arado para trabajar la tierra, por medio de sus rituales los sacerdotes trataban de unir el mundo espiritual con el mundo físico.

Con el paso del tiempo, la tradición védica palideció. Los sacerdotes comenzaron a explorar la idea del «atman», la esencia de lo humano. Los rituales físicos externos se convirtieron en meditación introspectiva, y los

practicantes espirituales comenzaron a renunciar al mundo material y a iniciar un ritual complejo para recorrer los bosques meditando, asumir votos de pobreza y ascetismo en busca de la verdad última.

Yājñavalkya, sabio que vivió en lo que hoy conocemos como el periodo Upanishad, propuso que el anhelante ego humano evita conocer nuestro verdadero yo. Al destruir el ego, nos hacemos uno mismo con nuestro atman, nuestra «alma» o yo esencial. Yājñavalkya también introdujo la idea de «karma», que definió como las acciones que tienen lugar para lograr la liberación del ego. Al separar el cuerpo del atman, es posible llegar a lo que Buda llamaría después «iluminación».

Te presento al asana

Las ideas filosóficas de Yājñavalkya llevaron al desarrollo de lo que la mayoría de los practicantes considera la esencia de la práctica contemporánea de yoga: posturas físicas combinadas con respiración controlada y meditación. Como explica la historiadora de las religiones Karen Armstrong en su obra *The Great Transformation*^[22]:

El yoga es uno de los grandes logros de India y, en su forma más evolucionada, es casi seguro que fue diseñado en principio [por los practicantes] para liberar el *purusha* [yo esencial] de los enredos de la naturaleza.

Este yoga clásico era muy distinto a la versión que hoy se enseña en occidente. No era un ejercicio aeróbico y no ayudaba a que la gente se sintiera bien con sus vidas: más bien, lo contrario. El yoga era un asalto sistemático al ego, un régimen estricto que, pasado un largo periodo, enseñaba al aspirante a abolir su conciencia normal, con sus errores y engaños, para remplazarla con el descubrimiento extático de su *purusha*.

El yoga era una actividad de tiempo completo. Y no era apta para los débiles de corazón: era una disciplina espiritual demandante, dominio exclusivo de monjes y gurús. Los yoguis clásicos no practicaban para fortalecerse o ser

más flexibles. Intentaban romper el vínculo entre su cuerpo y su atman. Un asunto complicado.

La codificación del yoga

En el siglo II a. C., un estudioso llamado Patañjali empezó a compilar y ordenar las filosofías y las prácticas del yoga en esa época. El resumen de las mejores prácticas (*raja* o «regias»), realizado por Patañjali, se convirtió en *Los yoga-sutras de Patañjali*, colección de aforismos que se hizo el texto fundacional de los practicantes del yoga.

En esta obra, Patañjali señala ocho aspectos fundamentales, o «brazos» de la práctica del yoga *raja*:

- Yama: moralidad.
- Niyama: autopurificación.
- Asana: postura.
- Pranayama: control de la respiración.
- Pratyahara: control de los sentidos.
- Dharana: intención.
- Dhyana: meditación.
- Samadhi: absorción/contemplación.

Patañjali llamó a este sistema «ashtanga» o «yoga de los ocho brazos». Al practicar diligentemente los ocho brazos del yoga, el practicante experimentaría la *kaivalya*: un desapego perfecto del alma del practicante del mundo material, lo que conduce a una felicidad eterna.

Lo que la mayoría de la gente piensa cuando escucha la palabra «yoga» —contorsiones en extrañas posturas— constituye solo uno de los brazos, *asana*. Por lo regular, el asana se practica con ayuda de técnicas de respiración llamadas *pranayama*. El objetivo de la práctica es preparar para la meditación (*dhyana*).

El sistema de Patañjali se concentró en el yoga como filosofía, no como un sistema de ejercicio. Las asanas estaban limitadas a algunas posturas extáticas muy lejanas de las contorsiones gimnásticas con que los practicantes modernos están familiarizados.

Esto no significa que no existieran poses intensas. Las posturas *hatha* (forzadas) habían sido conocidas desde la época de Yajñavalkya, pero no les preocupaba mucho el atletismo.

La compilación más famosa de las prácticas del yoga hatha es el *Hatha-yoga Pradipika*, de Maharishi Swatmarama, sabio del siglo XV. Swatmarama consideraba que el hatha yoga era la vía ideal para llegar al raja yoga de Patanjali. Swatmarama creía que al purificar el cuerpo por medio del ejercicio y al practicar pranayama e inhalar humo ritual mientras se paran de cabeza, los practicantes podían acceder a niveles superiores de conciencia.

Las versiones tempranas del hatha yoga no fueron muy populares. La práctica de posturas físicas esforzadas siguió siendo una forma oscura de *raja yoga* hasta que un brahmán de Mysore, India, revolucionó la práctica cuatro siglos después.

El hombre que «inventó» el yoga moderno

Tirumalai Krishnamacharya nació en 1888, siendo el hijo mayor de Sri Tatacharya, sacerdote brahmán muy conocido^[23]. Krishnamacharya fue introducido a la práctica del yoga hatha por su padre a la edad de 5 años, y siguió un aprendizaje estricto de esta disciplina durante toda su educación formal. En 1919, viajó al Monte Kailash, en el Tíbet, para estudiar con uno de los últimos maestros de hatha yoga que quedaban, Sri Ramamohana Brahmachari.

Krishnamacharya estudió con Brahmachari por siete años y medio, memorizando los sutras de Patañjali y practicando asanas. Según cuenta la historia, también le dio por realizar actos asombrosos para llamar la atención, como disminuir el ritmo de su respiración y detener su pulso por periodos extendidos de tiempo. Después, Krishnamacharya usaría estas habilidades con gran pericia, demostrándolas en público para popularizar la práctica del hatha.

La petición del gurú

La relación gurú-estudiante tiene una larga historia en India. Al final de la estancia de un estudiante con su gurú, se acostumbraba que el estudiante

ofreciera un pago al maestro, una práctica que recibía el nombre de *guru dakshina*. Se trataba de un gesto de agradecimiento y de profundo respeto por las enseñanzas del gurú. Por lo regular, el pago consistía en dinero o en bienes materiales, pero a veces el gurú pedía el cumplimiento de una labor especial. Cuando fueron completados los estudios de Krishnamacharya, la petición de Brahmachari fue simple: debía casarse, tener hijos y enseñar el yoga que había aprendido en el Monte Kailash.

La petición de Brahmachari fue estremecedora, pues Krishnamacharya ya había sido designado para varias posiciones religiosas y educativas de alto estatus. Además, se esperaba que asumiera las funciones de su bisabuelo como líder religioso principal del sur de India. Para darnos una idea del impacto que la propuesta tuvo, imaginemos que somos nombrados directores generales de una empresa prestigiosa, para descubrir que en realidad trabajaremos como encargados de la limpieza. La *dakshina* de Brahmachari constituía una sentencia vitalicia a una vida dura y de bajo estatus.

Krishnamacharya honró la petición de su gurú y volvió a Mysore, viviendo en la pobreza y buscando oportunidades para enseñar el hatha yoga.

En 1931, Krishnamacharya fue invitado a enseñar hatha en el Colegio de Sánscrito, en Mysore. Ahí conoció a Krishna Raja Wadiyar IV, el maharajá gobernante de Mysore. Impresionado por las habilidades y la preparación de Krishnamacharya, y agradecido por la ayuda que este le había brindado para controlar su diabetes, el gobernante propuso que Krishnamacharya abriera una escuela de yoga en el palacio, bajo su patronazgo.

Dado que la mayoría de los estudiantes de *yoga shala* en Mysore eran jóvenes llenos de energía, Krishnamacharya desarrolló una nueva forma de práctica de hatha yoga que enfatizaba el fortalecimiento y la flexibilidad. Combinó los hatha asanas y los pranayama tradicionales con movimientos de la gimnasia británica, adaptando y modificando las secuencias para cada estudiante en particular^[24].

Un nuevo yoga

Ese fue el origen de lo que hoy se conoce como yoga ashtanga vinyasa. Si asistes a una clase de yoga en un gimnasio o si tomas cualquier libro que

contenga las palabras *ashtanga*, *vinyasa*, «poder», «flujo», «esencia», *vinnyoga* o *Iyengar*, estarás practicando la tradición de Krishnamacharya.

Él adaptó sus enseñanzas a las necesidades de sus estudiantes. El yoga ashtanga fue popularizado por uno de los primeros discípulos de Krishnamacharya, K. Pattabhi Jois, quien estudió con él en Mysore. El ashtanga se basa en seis secuencias de posturas vigorosas, comenzando con la «Serie Primaria», que Krishnamacharya enseñó en el yoga shala. La actual práctica de ashtanga conserva el énfasis en la secuencia y en el atletismo.

Cuando enseñó a su cuñado, B. K. S. Iyengar, Krishnamacharya enfatizó la alineación del cuerpo en asana, debido a la frágil salud de Iyengar en esa época. Iyengar desarrolló esta aproximación hasta dar con el estilo de yoga orientado a la alineación que se practica en nuestros días.

Después, cuando Krishnamacharya pasaba la mayor parte de su tiempo trabajando como sanador, enseñó a su hijo, T. K. V. Desikachar, los aspectos terapéuticos de la práctica asana. Hoy, los discípulos de Desikachar conciben el yoga como una herramienta para el bienestar y exploran un amplio rango de aplicaciones para la salud, que van desde la terapia física hasta el alivio de la ansiedad. Como resultado, prácticamente cada variante mayor del yoga moderno ha sido muy influenciada por las enseñanzas de Krishnamacharya.

Lo que no cambió fue el énfasis en la gente normal y con todo tipo de responsabilidades. El yoga de Krishnamacharya no era para el monje o el asceta: el hecho de que se concentrara en una práctica personalizada para la gente común hizo que fuera accesible a millones de personas ocupadas en todo el mundo. En palabras de T. K. V. Desikachar: «Haces yoga para vivir tu vida, no a la inversa».

Yoga = respiración + movimiento + meditación

Volvamos al presente. ¿Qué tiene que ver esta historia con el yoga moderno?

He aquí lo que me sorprendió: la moderna práctica asana, hablando históricamente, es una invención reciente. Claro que los aspectos filosóficos del yoga han estado presentes durante algunos milenios, pero la práctica actual de asumir posturas mientras se respira y medita es, históricamente hablando, nueva.

El énfasis en la fuerza y en la flexibilidad también es nuevo. Muy pocos de nosotros tratamos de disociar nuestra alma del cuerpo en busca del atman. La mayor parte de los practicantes de yoga modernos no son ascetas ni han renunciado a nada. En general, los practicantes actuales se preocupan sobre todo por estar en forma, mejorar la flexibilidad y liberarse del estrés.

En el fondo, queda claro que la práctica moderna del yoga consiste en combinar movimiento, respiración y meditación. Las posturas mismas no son mágicas. Han cambiado mucho con el paso de los siglos, y Krishnamacharya no se andaba con cuentos cuando se trataba de agregar elementos o modificar la práctica para adaptarla al estudiante.

También resulta claro que el yoga no se trata de movimientos vistosos. Hay un trasfondo filosófico: un reconocimiento de que se invierte tiempo para convertirnos en mejores personas. Aunque el asana moderno es relativamente nuevo, la filosofía del yoga ha estado presente por miles de años y es interesante en sí misma. El yoga tiene una profundidad, lo que no siempre sucede con los actuales fanáticos del ejercicio.

Tampoco es necesario salirte de lo cotidiano para practicar posturas locas. Ser un verdadero yogui nada tiene que ver con ejecutar posturas extrañas: se trata de practicar para tener una vida mejor. ¿No puedes realizar una postura porque no eres suficientemente flexible, fuerte o por miedo a lastimarte? No hay problema: adapta la postura o realiza otra. Si te concentras en la respiración al moverte y si pones atención en lo que te comunica tu cuerpo, estás haciendo bien las cosas.

En esencia, la práctica moderna del yoga integra respiración, movimiento y meditación. No se trata de gimnasia, acrobacia o de ser superflexible. Ser superflexible o acróbata no te hace mejor yogui.

Aclarar los malentendidos

He aquí algunos malentendidos respecto al yoga:

- No tienes que saber miles de posturas para hacer yoga. Conocer muchas no te convierte en mejor yogui. Krishnamacharya comenzó con solo 24 posturas.

- No necesitas saber el nombre en sánscrito de las posturas, y ni siquiera en español. Conocer muchas posturas y saberte los nombres no te convierte en mejor yogui.
- Algunos practicantes adoptan dietas vegetarianas, el veganismo o la práctica Ayurveda, pero ninguna dieta es prerequisite para practicar asana.
- Algunos practicantes también se interesan en la medicina alternativa, la astrología y otros temas similares, pero nada de esto es necesario para la práctica del asana.

Las anteriores son malinterpretaciones en que yo mismo caí cuando escuché hablar del yoga por primera vez. Bastó investigar un poco qué es el yoga en realidad para aliviar mis preocupaciones iniciales y para no perder el tiempo en cuestiones sin importancia.

¿Es peligroso el yoga?

Un último asunto que me preocupa: ¿puedo lastimarme haciendo yoga?

En la época en que investigaba el asana, se publicó un artículo incendiario en el *New York Times*, titulado «Cómo puede afectar tu cuerpo el yoga^[25]», escrito por el encargado de temas científicos, William J. Broad. En esencia, el artículo es un extracto del libro de Broad, *The Science of Yoga: The Risks and the Rewards* (2012), que circuló unas semanas antes que el artículo.

La publicación del *New York Times* citaba muchos chismes del mundo del yoga. Broad citó bastantes estudios sobre los beneficios de la práctica asana, casi todos relacionados con el movimiento y el equilibrio emocional. Luego pasaba al ataque sugiriendo que la práctica del yoga es responsable de un amplio rango de lesiones, desde esguinces hasta infartos. En algunos casos, Broad sugiere que el yoga fue el responsable de que ciertos practicantes cayeran en coma.

El tema es un buen ejemplo de una *inversión* útil. ¿Cómo puedes hacer yoga de manera incorrecta y cuáles son los riesgos más significativos de la práctica asana?

La comunidad de maestros de yoga respondió con fiereza y rapidez al artículo de Broad. Dada la evidencia, Broad exageró. El yoga, como

cualquier actividad física, tiene riesgos, pero difícilmente te dañarás si haces las cosas de modo seguro.

En tanto que es posible lastimarte haciendo yoga, los problemas de salud mayores como infartos y estados de coma son raros. No existe evidencia de que la práctica del asana aumente el riesgo de problemas mayores. Comparado con otras actividades físicas, incluyendo los deportes en los que no hay contacto, como el golf, el yoga es muy seguro desde el punto de vista estadístico.

En cuanto a lesiones musculares como desgarres, ciertamente es posible lastimarse si fuerzas a tu cuerpo a adoptar posiciones ignorando el dolor y la incomodidad. Si crees que los «verdaderos» yoguis deben tocar sus rodillas con la nariz, y te obligas a adoptar semejante postura, efectivamente corres el riesgo de lesionar tus ligamentos o desgarrar los tendones de las corvas. Igualmente, si un maestro de yoga ajusta tu postura y excedes los límites de tu cuerpo, es probable que termines lesionado.

Un área potencial para lesiones son las vértebras cervicales. Algunas posturas, como los parados de cabeza, ejercen presión en la cabeza y el cuello, y si llegas a caer mal mientras te paras de cabeza, puedes lastimarte el cuello o la espina dorsal.

Con base en esta información, no fuerzo mi cuerpo para adoptar posturas y evito cargar peso innecesario sobre mis vértebras cervicales. Así, mientras atiende lo que dice mi cuerpo, puedo obtener beneficios de la práctica sin correr riesgos de lesionarme. Es bueno saberlo.

Un mínimo de asana viable

Estoy listo para empezar a practicar, así que... ¿por dónde empiezo?

Queda claro que existen muchos tipos, formas y filosofías que dictan cómo «debe» verse la práctica del yoga. Todo eso está bien, pero necesito dar un paso atrás para decidir cuál es el propósito del yoga *para mí*.

¿Por qué hago esto?

Quiero ser más activo desde el punto de vista físico, pues está claro que necesito moverme más. Mi médico me recomendó ejercicios para ampliar mi capacidad de movimiento y mencionó el yoga. He leído sobre los beneficios tanto del ejercicio como de la meditación, y el yoga combina ambos. No

tengo tiempo de asistir a un estudio para tomar largas clases, así que me gustaría practicar en casa, de manera efectiva y segura.

Ya tenemos suficientes detalles para establecer una meta de desempeño útil. He aquí la mía:

1. Aprender una secuencia de posturas físicas que puedo practicar para aumentar el ejercicio, la fuerza y la flexibilidad que practico cada día.
2. Combinar ejercicio con respiración y meditación, obteniendo beneficios físicos y mentales al mismo tiempo.
3. Practicar seguro y de memoria durante 20 o 30 minutos, ajustando duración, fuerza, flexibilidad y nivel general de energía.

El equipo para practicar yoga

Antes de practicar, debo conseguir equipo básico. En primerísimo lugar, un tapete que amortigüe y evite que manos y pies resbalen durante la práctica de las posturas. La sesión de yoga provoca mucho sudor; si te resbalas con él puedes lastimarte, así que es importante un tapete especial.

Cuando practiqué en Pure Yoga me dieron el tapete. Sin embargo, noté que resultaba un poco corto para mí. No era raro que casi me saliera del tapete, por lo que debía detener el ejercicio y ajustar la postura. Es mejor un tapete más largo.

Tras investigar un poco, decidí comprar uno marca Manduka PRO^[26], con 2,20 metros de largo. Tiene buena adherencia, es durable y de hermoso color verde.

Además, compré una bolsa para transportarlo y algunas toallas de mano para el sudor. Ya tengo *shorts*, playeras y un par de pants para yoga; nada falta en cuanto a ropa se refiere. El asana se practica descalzo, así que no necesito calzado especial.

Por sugerencia de Kelsey, compré una larga correa de algodón y un bloque de bambú, para que, según ella, me ayuden a modificar ciertas posturas.

Es todo el equipo que necesito, lo cual es bueno. No se necesita gran cosa para hacer yoga.

Aprender las posiciones

He aquí mi estrategia inicial: Kelsey es una maestra calificada, así que tiene sentido comenzar con su instrucción. Apartamos 90 minutos de una tarde para cubrir los temas básicos.

Si no tuviera una maestra de yoga en la familia, conseguiría una clase particular. Estas clases son muy útiles para este entrenamiento, puesto que es difícil aprender los movimientos de un libro.

Eso no significa que me salto lo que los libros ofrecen. En realidad, ya elegí varias fuentes de consulta:

- *Yoga Anatomy*, de Leslie Kaminoff y Amy Matthews (2011).
- *Yoga for Wellness*, de Gary Kraftsow (1999).
- *Ashtanga Yoga: Practice and Philosophy*, de Gregor Maehle (2007).
- *Breath-Centered Yoga with Leslie Kaminoff (DVD)*, por Leslie Kaminoff (2010).

Uso estos textos como referencia. Contienen fotografías, ilustraciones de las posturas e instrucciones detalladas de cómo hacer cada una correctamente. Consultaré la información conforme la necesite. Sabré qué otras cosas es importante aprender cuando empiece a practicar.

Kelsey y yo disponemos nuestros tapetes y comenzamos. Primero: el Saludo al Sol.

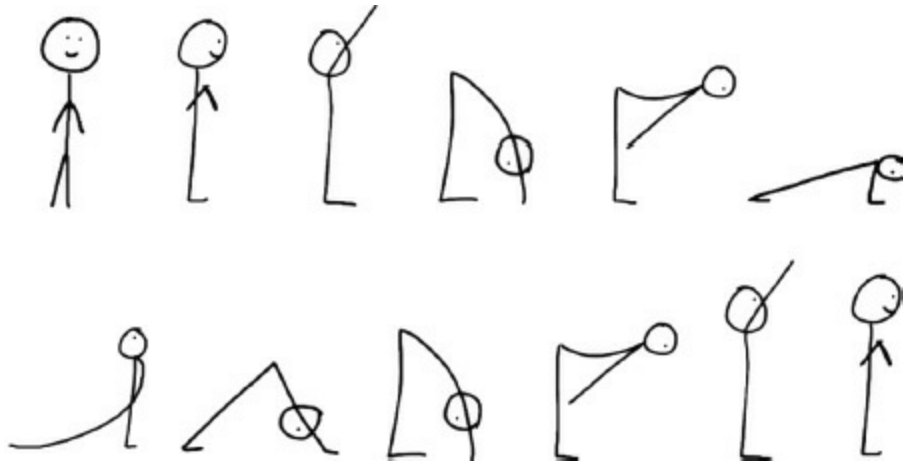
La secuencia del Saludo al Sol

El saludo al sol es una secuencia de posturas diseñadas para calentar los músculos, de modo que casi todas las prácticas asana comienzan con algunas rondas de estos ejercicios.

Una breve nota por si decides intentar esto: todos somos diferentes, y la forma «correcta» de hacer cada postura depende de los límites de tu cuerpo. Si prestas atención a lo que tu cuerpo

te dice, evitas la tensión exagerada y ajustas cada postura a tus límites personales, podrás practicar asana con seguridad.

Debes hacerlo así:



1. Inhala. Párate derecho en el borde del tapete, con los brazos relajados pegados a tus muslos y las rodillas también relajadas.
2. Exhala. Lleva tus manos a la postura universal de «oración», sobre tu pecho.
3. Inhala mientras subes los brazos completamente extendidos sobre tu cabeza. Luego, mira las palmas de tus manos. (Si te resulta incómodo, mira al frente.)
4. Exhala e inclínate al frente, doblando tu cuerpo por la cintura hasta que las manos se acerquen al suelo para tocar los dedos de tus pies. Estira tanto como puedas sin exagerar; está bien si doblas las rodillas. Relaja los hombros y el cuello.
5. Inhala y mira al frente mientras tus manos mantienen contacto con el suelo. (Intenta estirar las piernas mientras conservas la cabeza y el cuello relajados.)
6. Exhala mientras retornas a la posición en que te inclinas al frente.
7. Inhala mientras vuelves a poner los pies en posición parecida a la que adoptas cuando haces lagartijas, de manera que tu cuerpo esté por completo extendido y soportes tu peso con los brazos y las plantas de los pies. Para salir de esta postura, puedes caminar de regreso o saltar. (Esta postura recibe el nombre de «Tabla».)

8. Exhala y baja como si estuvieras haciendo una lagartija. No descanses en el suelo; detente a una o dos pulgadas del tapete y haz una pausa. Asegúrate de mantener un ángulo de 90 grados en tus hombros, codos y muñecas para evitar lesiones. Si esto te resulta muy difícil al principio, puedes apoyar las rodillas en el suelo. (Esta postura se llama Chaturanga Danda-sana, normalmente abreviada como «Chaturanga».)
9. Inhala mientras empujas hacia delante, arqueando la espalda mientras empujas, hasta que tu torso esté derecho. Al mismo tiempo, «voltea» tus pies, de modo que la parte superior de cada pie toque el piso y tus piernas queden a unos cinco centímetros del suelo. (Esta postura se llama «Perro boca arriba», dado que al hacerla se parece uno a un perro que se estira.)
10. Exhala y pon tus pies en la postura normal, de modo que las plantas toquen el tapete. Simultáneamente, mueve el coxis hacia arriba y atrás, hasta que tu cuerpo adopte una forma de «V» invertida. Relaja el cuello y los hombros. Si sientes que los tendones de las corvas se estiran, ayuda imaginar que «respiras» en ellos. Mantén esta posición durante unas cuatro o cinco respiraciones. (A esta postura se le suele denominar «Perro boca abajo».)
11. Inhala y luego exhala conforme caminas o saltas al frente, hasta que tus pies estén bajo tus hombros.
12. Repite los pasos 5 y 6.
13. Inhala conforme subes los brazos hasta quedar completamente extendidos sobre tu cabeza, como hiciste en el paso número 3.
14. Baja los brazos frente a ti hasta que las manos queden en la postura universal de oración, frente a tu pecho, como hiciste en el paso 2.

Nota que la secuencia termina en la misma postura del inicio. Cuando la hayas completado repítela tantas veces como quieras, o cambia a otra postura de pie.

También fíjate en que cada paso comienza inhalando o exhalando. Se trata de algo intencional: cada movimiento corresponde a una respiración. El hecho de concentrarte en esta diferencia al yoga de los aeróbics.

Puedes pensar en la secuencia de Saludo al Sol como si fuera un «mínimo asana viable»: si en verdad quisieras, toda tu práctica podría consistir en

hacer salutations al sol una y otra vez. La secuencia pone en acción todos los músculos mayores del cuerpo, por lo que se requiere de una buena combinación de fuerza y flexibilidad. Además, es un reto que no es complicado. Krishnamacharya sabía lo que hacía.

En la práctica ashtanga tradicional, el Saludo al Sol (también llamado *vinyasa*) se realiza entre cada postura. Eso significa que pasas la mayor parte de las dos horas de la Serie Primaria haciendo esta secuencia completa. Es un buen ejemplo del poder en acción: una minoría de posturas conforman la mayoría de la práctica.

Aun así, el cuerpo humano tiende a lastimarse al repetir los movimientos: si haces demasiados saludos al Sol, puedes lastimarte^[27]. Por eso es buena idea hacer unos cuantos Saludos al Sol para calentar, antes de pasar a otras posiciones.

Acuérdate de respirar como Darth Vader

Al practicar asana, es común utilizar una técnica de respiración especial llamada *ujjayi*. Se trata de la más simple de las técnicas de respiración (pranayama) usadas en el yoga, y está diseñada para que nos sea más fácil concentrarnos en la respiración.

En la parte trasera de tu garganta hay una estructura carnosa llamada «glotis». Está más o menos en la zona de las amígdalas o de las cuerdas vocales. Si te concentras en esa zona por un momento, puedes contraer la glotis conscientemente, lo que constriñe el paso del aire hacia y desde tus pulmones.

El efecto de esta constricción es notable: puedes oír (y sentir) el aire frío al entrar a tus pulmones cuando inhalas, y notar su tibieza al ser exhalado.

La obra *Yoga Sutras* describe el sonido del *ujjayi* como «semejante al del océano», pero yo prefiero valerme de una analogía moderna: el *ujjayi* suena como Darth Vader, el personaje de *La guerra de las galaxias*. Si imitas el infame patrón de respiración del Oscuro Señor de Sith y luego cierras la boca, estarás haciendo un *ujjayi*.

Los textos antiguos sobre yoga atribuyen cualidades místicas al *ujjayi*, así como también a otras técnicas pranayama. Sin importar si el *ujjayi*

«proporciona calor al cuerpo» o si «alienta el flujo del prana (la fuerza vital universal)», ciertamente facilita poner atención a tu respiración.

El principal propósito del ujjayi es mantener la respiración en calma, relajada y regular. Al mantener el ujjayi cuando practicas, es mucho más fácil tener la mente concentrada en la respiración, lo que evita distracciones. Si sigues tu respiración hasta excluir todo lo demás, meditarás sin proponértelo.

Aprendí la secuencia más utilizada en el asana y el patrón de respiración primario en 20 minutos. ¡Vamos bien!

Ahora, aprendamos algunas posturas comunes de pie.

Posturas de pie

Guerrero 1.



Las posturas del guerrero son como la materia prima del ashtanga y del vinyasa. Aparecen después (o en medio) de la secuencia de Saludo al Sol, y conforman una secuencia en sí mismas.

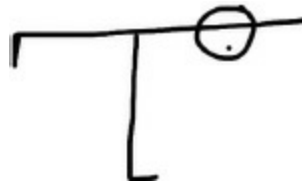
He aquí cómo funciona la primera postura del Guerrero: ponte de pie. Inhala y da un paso atrás con una pierna, dejando que los dedos del pie que hiciste hacia atrás apunten unos 45 grados hacia tu costado. Levanta los brazos en paralelo sobre la cabeza. La pierna al frente debe flexionar la rodilla, el pie completamente plantado contra el suelo y tu barbilla casi perpendicular al suelo. El pie de atrás debe estar completamente apoyado en el suelo. Mantén la posición de tres a cinco respiraciones.

Guerrero 2.



Desde la postura del Guerrero 1, abre los brazos y gira la cadera hacia un costado. (Si el pie derecho está atrás, gira la cadera a la derecha y viceversa). Mantén los brazos paralelos al piso, «abriéndose» a la altura del pecho. Mantén la postura de tres a cinco respiraciones.

Guerrero 3.

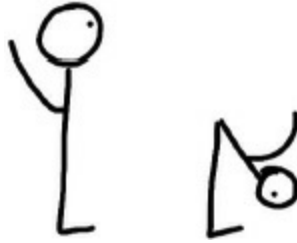


Esta posición bien podría llamarse la «postura de Superman». La meta es mover el torso hacia delante mientras mantienes el equilibrio en una sola pierna, con el resto del cuerpo paralelo al piso.

Estando de pie en posición neutral (el parado del Guerrero 1), eleva los brazos por encima de la cabeza. Luego separa ligeramente del piso uno de los pies. Muévelo hacia atrás mientras adelantas el torso, doblándote por la cintura hasta llegar a la postura de Superman. Concéntrate en lograr el equilibrio con los músculos de tu pie y mantén la rodilla relativamente firme, pero no tensa.

Mantén la postura entre tres y cinco respiraciones y luego vuelve a la posición de parado neutral.

La espiral Kaminoff.



Desde la posición de parado neutral, inhala y deja que tus brazos se muevan en espiral hacia fuera y arriba, como si comenzaras un Saludo al Sol. Desde esta posición, exhala y baja tus brazos en espiral, mientras te inclinas hacia delante por la cintura hasta que el pecho esté cerca de las rodillas. Repite esto algunas veces inhalando conforme te incorporas y exhalando cuando bajas.

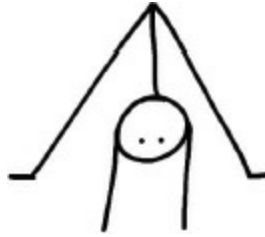
Esta es mi postura favorita de pie: se siente tan bien hacerla cuando has salido de la cama o si has estado horas sentado frente a la computadora. El beneficio de esta variante es que no carga peso alguno en brazos y manos, lo que es bueno si se padece el síndrome del túnel carpiano, te recuperas de alguna lesión o necesitas fortalecer los brazos antes de intentar posturas más difíciles.

Doblado al frente.



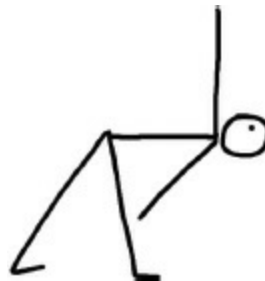
Desde la posición de parado neutral, inclínate al frente doblando la cintura (que tus rodillas se flexionen si es necesario). Relaja el cuello y los hombros y permite que la cabeza cuelgue libremente hacia el piso mientras intentas tocarlo con los dedos. Haz pruebas flexionando un poco y estirando las rodillas, y fíjate cuando sientas el estiramiento. (Si tus tendones de las corvas son muy elásticos, podrás apoyar las palmas de las manos en el piso, pero no fuerces tus capacidades). Después de unas cinco respiraciones, vuelve a la posición inicial.

Amplio descenso al frente.



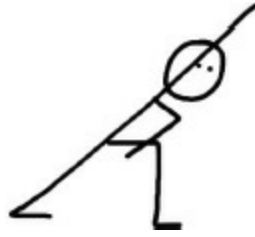
Desde la posición de parado neutral, separa las piernas creando una «V» invertida bastante amplia. Luego, baja el torso hacia delante doblando la cintura. Relaja el cuello y los hombros y permite que tu cabeza cuelgue libremente, mientras intentas alcanzar el suelo con las palmas de las manos. Después de tres a cinco respiraciones, incorpórate de nuevo.

El triángulo.



Desde la posición neutral de pie, abre las piernas hasta tener una postura de «V» invertida, con los pies apuntando al frente. Luego, gira el torso de modo que uno de tus pies apunte hacia el costado del tapete. Dobla esa rodilla y permite que tu codo descansa en ella. Eleva tu mano libre hacia el techo. Finalmente, estira la pierna flexionada y deja que tu mano de abajo caiga hasta la pierna o el suelo. Mantén la postura entre tres y cinco respiraciones, vuelve al parado con las piernas abiertas y repite el movimiento hacia el otro costado.

Extensión lateral en ángulo.



Desde la posición de parado neutral, separa las piernas hasta que estas adopten la forma de «V» invertida, con los pies apuntando al frente. Después, gira la cintura de modo que uno de tus pies apunte al costado del tapete. Dobla esa rodilla y deja que tu codo se apoye en ella. Eleva tu mano libre hacia el techo y luego (si te sientes cómodo), lleva la palma de la mano de abajo hasta el suelo. Tus brazos deben quedar perpendiculares al piso, formando una línea recta.

Existe una variante de esta postura que me gusta mucho: en lugar de extender los brazos hacia el suelo y el techo, puedes llevar ambas manos a la posición universal de oración. Después de mantenerla entre tres y cinco respiraciones, regresa al parado con las piernas abiertas y repite el movimiento hacia el otro costado.

La silla.



Párate derecho con las piernas juntas y los brazos pegados a tus costados. Inhala y estira los brazos sobre tu cabeza. Luego exhala y flexiona las rodillas manteniéndolas juntas, hasta llegar a una posición parecida a la de estar sentado. Puedes imaginar que usas una silla invisible. (Esto es difícil, y más aún conforme sostienes la postura). Mantén la posición entre tres y cinco respiraciones y luego vuelve a ponerte de pie, como al inicio.

El árbol.



Párate derecho, con las piernas juntas y los brazos pegados a tus costados. Levanta un pie y toma tu tobillo con la mano del lado correspondiente. Manteniendo el equilibrio con el otro pie, guía el pie libre hasta la parte interna superior del muslo, con la rodilla apuntando a un costado. Probablemente notarás que los músculos del pie que te sostiene trabajan como locos hasta hallar el equilibrio.

Cuando tu pierna esté en la posición correcta, suelta tu tobillo y junta las palmas de las manos en tu pecho, en la postura universal de oración. Mantén esa pose de tres a cinco respiraciones y luego repite todo el proceso con el otro pie.

Posturas de piso

El gato/la vaca.



Apóyate en el suelo con rodillas y manos (como si estuvieras en cuatro patas), con los brazos y las piernas bien abiertos. Mantén el cuello y los hombros relajados y mira al suelo.

Exhala mientras arqueas la espalda, como si metieras la panza en dirección a tu espalda. Te encontrarás mirando hacia atrás, hacia tus piernas. Esta es la postura del gato.

Luego, inhala mientras vuelves a enderezar la espalda y retornas a la postura neutral. Después arquea la espalda en la dirección contraria,

acercando la panza al suelo, como hiciste al principio del Saludo al Sol. Te encontrarás viendo hacia delante y tu espalda estará cóncava. Esta es la postura de la vaca.

Respira y alterna entre estas dos posturas al menos cinco veces.

Doblado al frente en posición de sentado.



Siéntate en el tapete con las piernas extendidas frente a ti. Mantén los pies flexionados y perpendiculares al piso. Sin doblar las rodillas, pivotea desde la cintura poniendo a trabajar los cuádriceps (los músculos en la parte frontal de los muslos), y estira las manos al frente, como si quisieras llegar a tus pies. Llega tan lejos como puedas sin que te duela y sin sentir incomodidad. Mantén la postura entre tres y cinco respiraciones.

Doblado al frente con una pierna estirada.



Piensa en esta postura como si fuera una combinación del árbol y del doblado al frente. Flexiona la rodilla para subir un pie y luego rota la pierna a un costado. Coloca la planta del pie sobre la parte interna de tu otro muslo y flexiona la cintura hacia delante, como hiciste en el doblado al frente. Llega tan lejos como puedas sin sentir dolor o molestia. Mantén la postura entre tres y cinco respiraciones.

El bastón.



Esta postura es más difícil de lo que parece. Siéntate en el tapete con las piernas extendidas frente a ti. Mantén los pies flexionados, perpendiculares al cuerpo, mientras procuras orientar las puntas hacia arriba. Mantén la postura entre tres y cinco respiraciones.

Ángulo cerrado.



Siéntate en el suelo y pega las rodillas al pecho. Luego, permite que se abran a los costados hasta que estén cerca del suelo y las plantas de tus pies se toquen. Toma tus pies y mantén la postura entre tres y cinco respiraciones.

Torsión espinal en posición de sentado.



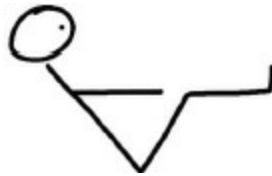
Siéntate en el tapete con las piernas extendidas. Lleva una de tus rodillas al pecho, hasta que el pie quede cerca de la rodilla de la otra pierna. Toma tu pie colócalo en el otro lado de tu pierna extendida. Pon las palmas de las manos en el piso, a la altura de tus caderas, como hiciste en la postura del bastón.

Cuando estés en posición, mueve el brazo opuesto hacia el otro lado de tu cuerpo hasta que el codo toque la parte externa de la rodilla. (Si tienes doblada la pierna derecha, moverás el brazo izquierdo). Dobla el brazo por el codo y mantén los dedos apuntando al techo. Continúa con la torsión al empujar la pierna suavemente. Completa la torsión rotando la cabeza en la misma dirección de la torsión.

Lo bueno de esta postura es que notas la diferencia al girar las diversas partes de tu espina dorsal. Según Leslie Kaminoff, la espina lumbar solo tiene 5 grados de rotación. La parte del tórax tiene 35 grados y la cervical entre 80 y 90 grados. Así, cuando has rotado completamente, debes poder ver atrás de ti.

Mantén la posición entre tres y cinco respiraciones, vuelve a la postura inicial y repite del otro lado.

El bote.



Esta es difícil. Siéntate en el tapete con las piernas extendidas al frente. Dobla las rodillas de modo que te equilibres con las nalgas y que los dedos de tus pies toquen ligeramente el piso. Extiende los brazos frente a ti y luego mantén juntas las piernas mientras las levantas del suelo. Conserva recta la espalda y haz los ajustes necesarios para no perder el equilibrio. Aguanta durante tres y cinco respiraciones.

El cuervo.



Ponte en cuclillas sobre el tapete, apoyando las palmas de las manos en el suelo. Coloca los codos en la parte interna de cada rodilla y luego inclínate hacia delante hasta que tu centro de gravedad se mueva en ese sentido y tus pies se separen del suelo. Balancéate con los brazos tanto como puedas, procurando que tu respiración sea regular.

La rueda.



Acuéstate sobre la espalda en el tapete. Eleva las rodillas hasta apoyar las palmas de los pies sobre el piso, cerca de tus caderas. Al mismo tiempo, apoya las palmas de las manos en el piso, a la altura de tus orejas. Mientras inhalas, empuja con piernas y manos para que tu espalda se arquee separándose del tapete. Aguanta la posición entre tres y cinco respiraciones y luego baja lentamente el torso hasta apoyarlo en el piso de nuevo.

Ajuste de la dificultad

¿Te fijaste en que todas estas posturas deben mantenerse entre tres y cinco respiraciones? Se trata de algo intencional. La respiración es lo que distingue al yoga, así que es parte trascendental de cada postura. Entre tres y cinco respiraciones ujjayi son un buen comienzo para principiantes.

Los practicantes de asana avanzados no necesariamente hacen posturas más complejas: las hacen más lentamente, con mayor control y las mantienen más tiempo. Conforme vaya mejorando con la práctica, mantendré cada postura durante más respiraciones.

Shavasana.



Shavasana es, en la tradición yóguica, el propósito último de todas estas contorsiones y respiraciones. Recuerda que la orientación inicial del asana era preparar para la meditación. Esta tradición se ha mantenido en la práctica moderna y por una razón muy convincente: es muy agradable.

Shavasana es la liberación de toda la tensión muscular, combinada con la meditación profunda. Es común que la gente se refiera a esta postura con el nombre de «el cadáver», pero no permitas que semejante nombre te desanime. La postura misma es sencilla: acuéstate sobre tu espalda en el tapete, con las piernas extendidas y los brazos pegados a tus costados. Cierra los ojos y libera la tensión del cuerpo. Tus músculos deben estar completamente inactivos.

Revisa mentalmente tu cuerpo, de la cabeza a los pies y, cuando detectes cualquier tensión o músculo que se contrae, relájalo. Esto incluye a los grupos musculares en que solemos poner poca atención, como cuello e incluso lengua. En este momento, deja el ujjayi y respira normalmente.

Tendrás la sensación de estar fundiéndote con el piso. Después de tanto tiempo en que los músculos se han movido constantemente, el shavasana constituye un alivio. Parece que al cansar el cuerpo, la mente se relaja. Es más fácil meditar cuando se alcanza el shavasana.

Permanece en shavasana durante 5 minutos y luego levántate. Tu práctica de asana ha terminado.

Un repaso del método

Llegado este punto, completé el desempeño que constituía mi meta. Después de unas 3 horas de instrucción, puedo practicar el asana en casa de manera segura y eficiente.

Repasemos la esencia del método que utilicé para aprender yoga:

- Adquirí el equipo necesario: tapete, correa y bloque.
- Pasé algunas horas con un instructor experimentado que me enseñó lo básico y corregí malinterpretaciones mayores.
- Aprendí que la práctica del yoga asana implica adoptar posturas mientras se concentra uno en la respiración y se mantiene un estado de conciencia amplia.

- Aprendí una secuencia vinyasa básica de posturas (el Saludo al Sol), 10 posiciones de pie comunes y 10 posturas sentado o inclinado.
- Aprendí técnicas básicas de respiración (pranayama), lo que facilita concentrarse en la respiración mientras se practica.
- Aprendí a prevenir lesiones al ponerle atención a mi cuerpo, modificando las posturas según sea necesario, evitando el exceso de ejercicio y el mucho, e innecesario, peso en la zona cervical.
- Practiqué hasta que pude completar la secuencia de memoria, lo que toma unos 25 minutos.

¿Qué rumbo seguir?

Hasta el momento, me siento feliz practicando asana. Después de 3 horas de investigación e instrucción, puedo practicar yoga cuando lo necesite y no dependo de una clase formal en estudio. Veinticinco minutos por la mañana o por la noche y listo.

Me gusta que el yoga combine ejercicio con meditación. Después de practicar, es obvio lo bien que se siente mi cuerpo y lo bien que me siento desde el punto de vista emocional. La práctica continua me asegurará los beneficios de la meditación y del ejercicio regular.

¡Ese es el truco! Necesito practicar regularmente para obtener la recompensa. Reservar tiempo en la mañana y en la noche para mi práctica es clave. En vez de comenzar el día entrando a Internet para revisar mi correo o irme directamente al trabajo, cambio mi agenda para practicar antes de cualquier otra cosa.

Cuidar de mi cuerpo debe ser mi principal prioridad, no la última. El yoga me ayudará a mantener una buena condición física para las décadas venideras.

No debo convertirme en monje o contorsionista para disfrutar los beneficios del yoga. Tampoco conocer cada postura ni poner la pierna detrás de la cabeza. Solo necesito practicar lo básico, que no fue tan difícil de aprender.

Mejor aún: aprendí lo que quería en solo 3 horas, lo que es mucho mejor de lo que imaginaba. Comencé este proyecto esperando estudiar una larga serie de posturas pero resultó que no era necesario. Pude lograr lo que

deseaba con solo simplificar, lo que me permitió obtener los resultados deseados en menos tiempo.

Cuando se trata de aprender algo nuevo, no tiene sentido hacerlo más difícil de lo necesario.

5

Programar

Lección: las cosas complejas son simples cuando las divides en sus partes constituyentes.

Si el debugging es el proceso mediante el cual se eliminan los errores de los programas, entonces programar debe ser el proceso mediante el cual se cometen errores de programación.

Edsger Dijkstra, reconocido experto en ciencias informáticas

Para obtener imágenes, videos y comentarios adicionales sobre este capítulo, visita <http://first20hours.com/programming>.

Vivo de la red desde 2007, fecha en que dejé mi trabajo de tiempo completo como gerente de mercadeo en Procter & Gamble para inaugurar mi propia empresa editorial y de consultoría.

Mi sitio web primario, PersonalMBA.com, es mi fuente de ingresos: efectivamente, soy profesor en el área de negocios, pero no trabajo en la universidad. Cada año, actualizo mi lista de los mejores libros de negocios

para los lectores autodidactas que desean conocer los fundamentos de esta materia^[28].

La lista de recomendaciones de mi sitio web ha sido uno de los elementos favoritos desde que la hice por vez primera, en 2005. Las actualizaciones de la lista hacen llegar a mi sitio oleadas de visitantes de todas partes del mundo. Desde 2005, PersonalMBA.com ha sido visitado por unos 2 millones de lectores.

Mi primer libro, *The Personal MBA: Master the Art of Business* (2010), fue una extensión natural de PersonalMBA.com y se convirtió en un *best seller* internacional. Parte del proceso de escribir un libro consiste en averiguar cómo difundir la palabra, de modo que, a lo largo de los años, he trabajado para atraer nuevos lectores.

Como resultado, *The Personal MBA* ha aparecido en el *New York Times*, en el *Wall Street Journal*, *Fortune*, *Forbes* y *FastCompany*, en muchos sitios web y en blogs muy populares. Cada vez que mi libro o mi sitio son citados en algún medio, miles de lectores visitan PersonalMBA.com en muy poco tiempo.

El precio del progreso

Tener miles de visitantes simultáneamente en tu sitio web es algo grandioso, siempre y cuando esos visitantes puedan acceder al sitio. Ese fue mi problema: siempre que mi sitio web tenía una cantidad considerable de tráfico, el servidor se caía y dejaba a los visitantes con un críptico mensaje de error.

He aquí un ejemplo típico: Lifehacker.com, blog popular dedicado a la productividad, ha dado tres veces un lugar protagónico a mi libro en los últimos siete años. Cada vez, miles de personas trataban de visitar PersonalMBA.com al mismo tiempo, abrumando a mi servidor (la computadora que envía la página web cada vez que esta es solicitada por un visitante). En lugar de enviar la información solicitada, mi servidor regresaba un mensaje de error que decía: «Error al establecer la conexión con la base de datos», o «Error 503». En otras palabras, el servidor pedía ayuda a gritos.

Cada vez que mi servidor se colapsaba por incremento en el tráfico, una parte de mi alma se colapsaba con él. Todo el tiempo invertido en la

promoción de mi sitio web era un desperdicio. Miles de lectores curiosos se interesaban en saber qué les ofrecía, pero con la caída del servidor, quedaban decepcionados y con las manos vacías. Mi estrategia de mercadeo funcionaba muy bien, pero mis sistemas no podían con la demanda repentina.

Examen del problema

Al principio, mi solución consistió en aumentar el poder del servidor añadiendo poder de procesamiento y memoria. Eso ayudó, pero hasta cierto punto. Más allá de eso, el servidor seguía cayéndose, lo que solía coincidir con cada victoria que lográbamos en mercadeo.

En esa época, PersonalMBA.com corría en un popular sistema de administración de sitios web llamado WordPress^[29]. Este programa ha sido optimizado para ser fácil de instalar y usar, no para resistir cargas de tráfico intensas. En la configuración de fábrica del programa, cada petición de una página web da inicio a un mundo de actividad del servidor, generando cientos de peticiones ocultas para entregar cada página web al lector.

Eso hace que cada petición individual de una página resulte «cara». Es decir, cada una requiere de una importante cantidad de memoria y de un poder de procesamiento mayor para ser completada. Si un visitante mira cinco páginas de mi sitio, ese usuario genera cinco peticiones caras. Si mil visitantes solicitan exactamente la misma página en el mismo momento, el servidor tratará de echar a andar mil procesos caros idénticos de modo simultáneo.

El sistema se ha caído

En esta situación, el pobre servidor tratará de atender todas las solicitudes, pero dado que cada una demanda muchos recursos, el servidor agotará la memoria antes de satisfacer cada petición. En ese punto, el servidor levanta la bandera blanca de rendición y los visitantes se quedan sin suerte.

Para combatir este problema, cambié la empresa que albergaba mi página web cinco veces y gasté cientos de horas aprendiendo a configurar los servidores WordPress para permanecer en línea bajo fuertes cargas de

trabajo. Cada nueva configuración del servidor aumentaba la complejidad, y cada nueva instalación requería más y más mantenimiento sobre la marcha.

Eventualmente, opté por hacer mis propias configuraciones de servidor, invocando órdenes secretas del sistema para instalar, configurar y modificar aplicaciones complejas del servidor que apenas entendía. Cada error o problema en la instalación o el mantenimiento implicaba horas de investigación para resolver el conflicto.

Por si fuera poco, la popularidad y la relativa inseguridad de WordPress lo convertían en un objetivo popular entre los *hackers* y los *spammers*. Cada semana o algo así, un oscuro programador encuentra una nueva vulnerabilidad y la utiliza para apoderarse de cuentas de usuarios o para llenar de publicidad no solicitada millones de blogs de WordPress. Asegurar una instalación de WordPress y mantenerla actualizada puede convertirse en un trabajo de tiempo completo, particularmente si te ocupas de varios sitios web. (En aquella época, yo mantenía doce sitios y lo hacía muy mal).

En un momento dado, me di cuenta de que pasaba más tiempo manteniendo mis sitios web que investigando y escribiendo para mis lectores. Eso no tenía sentido. No solo estaba desperdiciando capacidad productiva, sino que ni siquiera aprendía a programar. Sabía darle vuelta a problemas específicos que solo sucedían en WordPress. No estaba nada bien^[30].

Decidí buscar otra manera de mantener mis sitios web y no tardé en encontrar una alternativa prometedora.

Una posible solución

Un día, di con un ensayo sobre Jekyll^[31], programa de administración de sitios web creado por Tom Preston-Werner, mejor conocido por ser el fundador del código abierto para repositorio digital llamado GitHub^[32]. Jekyll está diseñado para remplazar sistemas como WordPress al facilitar la administración de sitios web que no se basan en peticiones caras.

Imagina que tienes cientos de documentos de procesador de textos que contienen información importante y debes dar a todos el mismo aspecto —la misma fuente, el mismo estilo de encabezado y demás—. Si escribieras un programa que pudiera aplicar un diseño de página determinado a cada

archivo de modo automático (en lugar de actualizar cada archivo a mano), ahorrarías mucho tiempo.

Esto es esencialmente lo que Jekyll hace por las páginas web. Da un solo comando y Jekyll produce un sitio web completo usando los archivos de tu computadora con información del sitio y una plantilla de diseño. Si haces algún cambio al diseño o al contenido de la página, solo corres el programa Jekyll y todo tu sitio se actualizará ahorrándote horas de trabajo.

Jekyll ofrecía una oportunidad promisorio. En teoría, remplazaba mi WordPress con un simple folder de archivos de texto en mi computadora. Mi sitio web sería rapidísimo, ultraestable y me ahorraría cientos de horas de mantenimiento al servidor por año.

Sin embargo, había un problemita: Jekyll está escrito en Ruby, lenguaje de programación que desconozco. No tengo ni idea de cómo escribir un código en Ruby o correr aplicaciones Ruby que sirvan a los usuarios de carne y hueso. Algunas cosas que necesito hacer para el funcionamiento de PersonalMBA.com requieren más que un simple formateo.

Para operar mis sitios web usando Jekyll, debía aprender a programar y desplegar aplicaciones de red Ruby.

Parece que encontré un «adorable problema».

Aprendiendo a codificar

Hacía tiempo que deseaba aprender a programar, pero siempre di prioridad a otros proyectos. Si pudiera aprender a programar, mis oportunidades de negocio y editoriales aumentarían considerablemente, ya que todo lo que hago por mi negocio corre en la web.

Es importante señalar que no todo lo hecho hasta la fecha es programación. HTML y CSS, los lenguajes que utilizo para construir páginas web, se conocen como lenguajes de marcado. Los códigos HTML y CSS no son muy inteligentes: simplemente indican a la computadora que muestre al usuario un archivo de texto de un modo específico (por ejemplo, «haz que este texto aparezca en negritas» o «el encabezado de esta sección tiene una letra de 24 puntos»).

Lo mismo sucede con la configuración de mi loco servidor. A pesar de que reunía *software*, no programaba en realidad. Más bien, instalaba

programas preescritos y luego cambiaba parte de su configuración. Los programadores habían escrito las aplicaciones que yo usaba, pero no requería de ningún conocimiento de programación para utilizarlas. La configuración y administración del servidor es una herramienta útil, pero no es programación.

¿Y qué es la programación?

Lo primero que pensé es que programar es decirle a una computadora que haga cosas, pero eso no resulta muy esclarecedor que digamos. ¿Hacer qué? ¿Qué «cosas»?

Hace 10 años tomé dos cursos básicos de programación en la escuela, así que francamente resultaba un poco vergonzoso no dar con una definición útil. Recuerdo algunos términos básicos, como variable, bucle, *input*, *output*, función, orientación del objeto y algo llamado ordenamiento de burbuja, pero no mucho más.

El curso requería aprender un lenguaje de programación llamado C++. Me recuerdo frustrado cuando invertí horas para encontrar un punto y coma faltante que congelaba mi programa. Recuerdo que mi profesor decía cosas como: «Nunca usarás un ordenamiento de burbuja en una aplicación real, pero lo aprenderemos de todos modos».

Escribí algunos programas básicos para el curso porque estaba obligado a hacerlo: la meta era aprobar con una buena calificación y lo hice. Desafortunadamente, los programas que escribía no tenían utilidad real fuera del salón de clases y no había usado esos conceptos en mucho tiempo. A pesar de que recuerdo algunas palabras, debo empezar otra vez para comprender ideas clave.

Dado que no puedo definir lo que trato de hacer cuando digo «quiero aprender a programar», tendré problemas para definir el nivel de desempeño objetivo que será mi meta. «Crear un programa de computadora» es un poco más específico, pero todavía no es muy útil que digamos.

Llegó la hora de poner sobre la mesa todo lo que sé sobre programación:

- Sé que los programadores «escriben» programas, lo que sugiere una labor creativa que puede realizarse de muchas maneras.

- A los programas se les suele llamar «aplicaciones»; ambas palabras se usan indistintamente.
- Cuando los programas de computadora «corren» o «se ejecutan», hacen lo que les dijeron hacer cuando fueron escritos, sea lo que sea.
- «*Input*» y «*output*» son términos fáciles de recordar, pues se utilizan en ámbitos tecnológicos con frecuencia. *Input* es información o datos que el programa usa, y *output* es lo que obtienes cuando el programa termina de correr.
- Una «variable» es básicamente un marcador de posición para algo que cambia. Puedes crear tantas variables como quieras y hacer que representen lo que tú quieras.
- El «programa» mismo es en esencia un conjunto detallado de instrucciones y reglas que indican a la computadora qué hacer exactamente con el *input*. Cuando el programa termina de correr, te da el *output*.
- Los programas se «congelan» o despliegan mensajes de error cuando algo marcha mal o la computadora no sabe qué hacer a continuación.

Ahora sí estamos llegando a alguna parte. Esta es una deconstrucción muy básica: en lugar de «programar», tenemos ahora tres subconceptos para trabajar:

- *Input*: información que usas para ejecutar un proceso.
- Proceso: una serie de pasos que el programa realiza, de acuerdo con el *input*.
- *Output*: el resultado final que ofrece el programa.

Esta subdivisión del concepto es mucho más útil. «Escribir un programa computacional» significa determinar con qué información cuentas para empezar, definiendo pasos que describen exactamente lo que la computadora hará con ese *input*, y definiendo el *output* que la computadora retornará cuando el programa termine de funcionar.

Piensa en un diagrama de flujo, el cual parece ser un gancho mental bastante útil para entender cómo funcionan los programas. Comienzas el proceso con determinados *inputs*. Sobre la marcha, realizas ciertas acciones cuando las condiciones específicas son verdaderas o falsas. El proceso

termina cuando llegas al final del diagrama de flujo y te quedas con el *output*: el resultado final del proceso completo que el diagrama de flujo describe.

Crear un programa computacional da la impresión de ser una forma distinta de lo que haces cuando creas un diagrama de flujo. En ambos casos se hace el mismo tipo de preguntas:

- ¿Con qué cuento para empezar?
- ¿Qué sucede al principio del proceso?
- ¿Qué sucede después de eso? ¿Y luego?
- ¿Cuándo termina el proceso?
- ¿Qué obtengo cuando el proceso ha terminado?

Los diagramas de flujo describen las respuestas a estas preguntas en un formato visual, y los programas lo describen usando texto, pero el proceso de pensamiento es el mismo.

La analogía del diagrama de flujo es útil porque da algunas claves sobre otros conceptos quizá importantes.

Los *condicionales* son afirmaciones como:

- «Si X es verdadero/falso, entonces haz Y.»
- «Si X es/no es Y, entonces haz Z.»
- «Cuando X es verdadero/falso, entonces haz Y.»
- «Cuando X es Y, entonces haz Z.»
- «En tanto que X es verdadero/falso, entonces haz Y.»
- «En tanto que X es Y, entonces haz Z.»

En este caso, X, Y y Z son *variables* que pueden representar cualquier cosa, como números (como en álgebra básica) o palabras. Algunas variables consisten en una sola letra o símbolo; otras son palabras. Sea cual sea el caso, las variables representan aquello con lo que estamos trabajando.

Las condicionales (las partículas como SI, ENTONCES, CUÁNDO y EN TANTO) son como las flechas con preguntas en un diagrama de flujo. Piensa en una actividad como conducir un auto: SI la luz del semáforo es roja, ENTONCES detente. Cuando la luz del semáforo cambie al verde, ENTONCES avanza. CUANDO la luz del semáforo esté en amarillo, desacelera y prepárate para detenerte.

Vale la pena ahondar un poco más en estas afirmaciones condicionales, ya que existen algunos patrones comunes. Verdadero/falso aparece mucho, y EN TANTO parece implicar que seguimos haciendo algo en lugar de realizar una sola acción.

En todos los casos, la condicional contiene alguna afirmación que define si se actuará o no. Esta afirmación recibe el nombre de *condición*, y puede adoptar muchas formas. A veces, consiste en un simple verdadero/falso comparativo (¿Es roja la luz?). En otras ocasiones, se trata de una comparación matemática (¿Es X mayor que 100?), y a veces contiene elementos de lógica (¿NO es roja la luz del semáforo?).

El propósito de la condición es definir si debe o no tener lugar el procedimiento asociado. Si la condición es verdadera o válida, el programa procesa la instrucción. Si no lo es, la instrucción asociada es saltada y el programa avanza a la siguiente instrucción.

Las variables verdaderas/falsas se llaman *variables booleanas*, nombre rimbombante para algo que ofrece solo dos opciones. Sí/no y prendido/apagado son variables booleanas. Estas son muy importantes en los programas de computación, pues son la unidad básica tanto de los procesos simples (como el verdadero/falso en un diagrama de flujo), como del encendido/apagado de los diminutos interruptores que se encuentran en el alma misma de la computadora.

EN TANTO, en este caso, es un tipo de condicional especial llamada *bucle*. Los bucles hacen que el proceso en cuestión se repita hasta que se cumpla la condición. Volviendo a la analogía de la conducción automovilística: EN TANTO la luz del semáforo sea roja, no te muevas.

Bastante fácil, ¿no? Si pensamos en escribir un programa computacional como si se tratara de dibujar un diagrama de flujo, la visualización del proceso es más sencilla.

Un último detalle: ¿qué pasa si la computadora se atasca, si no sabe qué hacer o si las instrucciones que trata de poner en acción no funcionan o no tienen sentido? ¿Qué sucede entonces?

El programa se «congela»: deja de funcionar completamente y suele enviar un mensaje de error en lugar del resultado deseado. Todos hemos experimentado la temible «pantalla azul de la muerte» en Windows, o el

típico mensaje «Error 404: Web Page Not Found» cuando navegamos en la red. Algo inesperado sucede, la computadora se confunde y el programa se congela.

Como programador de computadoras, tu trabajo consiste en evitar estas situaciones. La mejor manera de hacerlo es asegurarte de que el programa siempre tenga la información que necesita para completar el proceso planeado, pero no siempre es posible hacerlo. En los casos en que la incertidumbre es inevitable, es útil disponer de una manera de recuperarse cuando el programa falla al tratar de completar un proceso.

Estas afirmaciones para la recuperación de los errores se llaman *excepciones*, y son muy útiles. La mayor parte del tiempo el generador está allí sin hacer nada. No obstante, si se va la luz, el generador se enciende y el hospital usa su energía en lugar de quedar completamente a oscuras. Eso es algo muy bueno si en el hospital hay pacientes que requieren de máquinas eléctricas para mantenerse con vida. Este tipo de imponderables son peligrosos, así que es de vital importancia disponer de planes de emergencia.

Esa es la programación básica. En serio. Definir *inputs*. Disponer variables. Crear procesos que lleven a los *outputs* deseados. Pensar en estos procesos como si fueran un diagrama de flujo. Añadir las condicionales y las excepciones necesarias. Si todo marcha bien, provees los *inputs*, corres el programa y obtienes el *output* deseado.

Por supuesto, esta es una simplificación enorme de una actividad muy compleja, pero contiene los detalles suficientes para alguien nuevo en la programación. Al deconstruir la programación de este modo, es más fácil saber por dónde empezar.

Los lenguajes de programación

Aquí viene la parte complicada: las computadoras no hablan lenguas como lo hacemos los humanos. En esencia, las computadoras activan o desactivan pequeños interruptores de maneras muy específicas. A menos de que disponga de algún medio para traducir nuestras instrucciones en lenguaje humano en apagado y encendido de interruptores, la computadora no podrá hacer lo que le ordenamos.

Eso hacen los lenguajes de programación: ofrecen al programador humano una forma específica de decir a la computadora cuándo empezar, qué hacer y cuándo parar. También permiten definir los *inputs*, los procesos y los *outputs*, además de indicar al programa qué hacer cuando ha terminado de correr.

Cada lenguaje de programación tiene una manera específica de escribir los comandos. Se llama *sintaxis*. La sintaxis del lenguaje contiene las reglas que la computadora utiliza para traducir las órdenes que le da el programa en encendido y apagado de diminutos interruptores electrónicos.

Prácticamente cada lenguaje tiene su forma de definir variables, condicionales y excepciones. Los detalles varían, pero los conceptos fundamentales son los mismos.

Piensa como programador

Los programadores suelen abordar los problemas en *pseudocódigo*: un lenguaje que luce o suena como código, pero que no es suficientemente específico para que una computadora lo ejecute. Piensa en esto como si fuera un boceto. El pseudocódigo te ayuda a pensar en el proceso de resolución de problemas.

El ejemplo de la conducción de un auto es un pseudocódigo. Si subo a mi coche y digo en voz alta: «SI doy vuelta a la llave, ENTONCES enciende el motor», nada sucederá. Esto no significa que las afirmaciones sean inútiles. La afirmación es solamente una manera de pensar en los pasos para solucionar un problema u obtener el resultado deseado.

Puedes utilizar los conceptos de programación básicos discutidos para esbozar algunos programas sencillos que realicen tareas comunes.

He aquí una idea divertida para poner esto en práctica. Pide a un amigo que te ayude realizar una tarea sencilla como «hacer un sándwich». La única regla es que tu amigo solo hará exactamente lo que le dices, nada más y nada menos. No se le permite aportar nada y debe interpretar toda petición literalmente.

En minutos, tendrás una conversación como esta:

Tú: Toma el pan.

Amigo: No entiendo «toma».

Tú: Mueve tu mano adónde yo digo y agárralo.

Amigo: No entiendo «mano».

Tú: (Suspiro resignado). La cosa esta (señalas la mano de tu amigo).

Amigo: Entendido.

Tú: Mueve la mano adónde yo digo y agárralo.

Amigo: No entiendo «agárralo».

Tú: Flexiona los dedos así. (Demuestras la acción, flexionando y estirando los dedos).

Amigo: Entendido.

Tú: Mueve tu mano hacia el pan y agárralo.

Amigo: No entiendo «pan».

Tú: ¡Es la cosa que está ahí! (Señalas el pan).

Amigo: Entendido.

Tú: Mueve tu mano hasta el pan y agárralo.

Amigo: (Mueve la mano hasta el pan, flexiona los dedos y los estira. El pan no se mueve).

Tú: ¡Vaya juego estúpido!

Es un ejemplo tonto, pero se parece mucho al aspecto y tono de la programación, particularmente al principio.

La computadora, al igual que tu amigo, no entiende nada que no definas explícitamente. Cualquier proceso complejo que trates de definir necesita ser especificado por completo y sin ambigüedades.

Eso es lo que hace difícil la programación: una sola ambigüedad u orden equívoca ocasiona que el programa entero falle. Programar es una actividad que no perdona, tu código es correcto o incorrecto, y debe expresarse con detalles exactos.

Las computadoras no se impresionan con el encanto o el ingenio. Si tu código está incompleto o mal formado, tu aplicación se congelará, perderás datos y/o crearás un error: un pedazo de código malicioso que produce consecuencias inesperadas o imprevistas. Como sucede con las matemáticas, la lógica funciona o no funciona. No se dan puntos extra por el esfuerzo.

Habiendo dicho esto, en programación no existe una sola solución universal a un problema dado, lo que también sucede en las matemáticas. Existen miles de formas de producir el *output* deseado, según los *inputs* especificados. Como programador, debes elegir tu forma de hacer las cosas con base en las herramientas con las que dispones.

Cuando te acostumbras al hecho de que la computadora no puede leer tu mente, aprendes a dar órdenes que el sistema comprende de la misma manera en que aprendiste a definir explícitamente los términos para tu amigo, en el «juego del sándwich», antes de dar órdenes complejas.

¿Qué hace distinta a la programación de aplicaciones para la red?

Ya tenemos una definición viable de la programación. Está muy simplificada, pero por ahora nos da una comprensión suficiente de lo que pretendemos hacer.

No me interesa todo tipo de programación: quiero escribir programas que puedan correr en los sitios web. Si alguna vez utilizaste un programa para el correo electrónico como Gmail, Hotmail, Yahoo mail, etc., sabes de qué estoy hablando. Estos programas pueden correr en tu navegador de Internet. No tienes que bajar de la red a tu computadora un *software* para utilizarlo. Solo indicas a tu navegador que vaya a un sitio web y todo está listo.

Es importante distinguir entre programas que corren *local* y *remotamente*, en un servidor de la red. Para desarrollar una aplicación para la red, primero debes escribir la aplicación y luego probarla y ver si funciona. Todo el desarrollo y las pruebas tienen lugar en tu computadora.

Una vez que el programa funciona, lo envías al servidor de «producción» de la red para que otras personas accedan a él. La gente en Internet no puede conectarse con tu computadora personal desde la red, así que es necesario subir el programa a un servidor de acceso público para que otros lo utilicen.

Eso significa que el proceso de desarrollo tiene dos fases mayores: programación y pruebas locales, y envío del programa terminado a un servidor remoto de producción para su uso. Debo averiguar cómo funcionan estas dos fases.

He aquí algo que sé gracias a mi experiencia con HTML y CSS: son «tontos» en cuanto a que no puedes pedir a una página web básica que

almacene información alguna.

Digamos que tienes un archivo de página web que dice: «¡Hola, mundo!»^[33] y quieres remplazar la palabra «mundo» con el nombre de la persona que visita la página. No es mala idea, pero los sitios web más básicos no permiten almacenar información para uso posterior. Solo despliegan el texto del archivo y no acepta que se actualice a sí mismo.

El término técnico para referirse a esta situación es *estado*. Las páginas de la red básicas formadas mediante HTML y CSS no tienen estado, de modo que se les llama recursos «sin estado». Puedes agregar a una página web un pequeño formulario que pida el nombre del usuario, poniendo junto un botón que diga «salvar», pero el botón no salvará ninguna información a menos que crees un lugar aparte para guardarla.

Por esta razón, las aplicaciones para la red usan dos herramientas comunes para salvar los datos y usarlos más tarde: las *bases de datos* y las *cookies*.

La mejor manera para entender qué hace una base de datos es imaginar un montón de tarjetas de clasificación. Digamos que deseas crear una agenda y capturar nombre, número telefónico, correo electrónico, género y edad de cada uno de tus amigos.

Cada amigo tiene su tarjeta y la información se guarda en ella. Si uno de tus amigos cambia su dirección de correo electrónico, por ejemplo, puedes borrar la vieja información y actualizar la tarjeta. Cuando ves la tarjeta de un amigo en particular, puedes acceder a toda su información de inmediato.

Consideremos que la pila de tarjetas es una base de datos. Cada tarjeta de la pila se llama *record*. Puedes tener tantos récords como quieras en la base de datos, pero en algún punto, la pila de tarjetas se vuelve difícil de manejar. Suele tener sentido dividir la pila en montones menores: los amigos y la familia en una pila y los colegas del trabajo en otra, por ejemplo.

¿Todo claro hasta aquí? Ahora las cosas se ponen buenas: imagina que tu pila de tarjetas de clasificación es mágica. Puedes ordenarle que te muestre las tarjetas que se ajustan a condiciones, como:

- «Muestra la tarjeta de John Smith.»
- «Muestra todas las tarjetas en que el género sea femenino.»

- «Muestra todas las tarjetas en que la edad sea superior a 50 años.»

Bastante útil, ¿no? En esencia, es lo que hacen las bases de datos: te proporcionan una manera de almacenar información estructurada y de recuperar dicha información cuando y como quieras.

Cada uno de los datos que ponemos en las tarjetas se llama *campo*. Mientras más campos tengas en tu base de datos, dispondrás de más formas potenciales para recuperarlos cuando quieras.

Las bases de datos son la forma más común de almacenamiento de datos en las aplicaciones en red. Si quieres guardar información como el nombre de usuario, la dirección de correo electrónico y demás, el uso de una base de datos es casi natural. Cuando John Smith se conecte a tu aplicación, puedes hacer que esta recupere el nombre de John de la base de datos, para después desplegar el saludo: «¡Hola, John Smith!».

La otra forma común de guardar información en una aplicación web es por medio de *cookies*: un archivo de texto muy pequeño que se guarda en la computadora del usuario. Las *cookies* sirven para guardar pequeñas cantidades de datos que no se conservan por mucho tiempo.

En el caso de nuestra agenda digital, sería apropiado guardar una *cookie* cuando John Smith se conecte. El archivo *cookie* de John contendría información como nombre de usuario = johnsmith y loggedin = verdadero. Si Smith sale de la aplicación pero regresa más tarde, la aplicación reconocerá la *cookie* y le dará acceso sin pedirle que vuelva a ingresar los datos. Las *cookies* pueden programarse para expirar en un tiempo determinado, lo que las hace muy prácticas para este tipo de programación. (Si alguna vez has estado en un sitio web que dice: «recordar la contraseña», es así como funciona).

Lo que estamos haciendo aquí es una deconstrucción básica. No es una lista exhaustiva de lo que hace única a la programación para la red, pero es lo suficientemente completa para darte una idea de lo que debes aprender: variables, condicionales, excepciones, ambientes locales/de producción, bases de datos y *cookies*.

¿Te fijas cuán útil es la deconstrucción? Comencé con una idea muy vaga de lo que quería hacer y ahora dispongo de una lista específica de

subdestrezas importantes que aprender.

Todavía no estoy listo para comenzar. ¿Recuerdas que dije que las computadoras no pueden comprender el lenguaje humano? Debo elegir uno de programación para escribir las instrucciones de mi programa, lo que requiere un poco más de investigación.

Elegir un lenguaje para programar aplicaciones de la red

Existen miles de lenguajes disponibles, y cada día se inventan nuevos. La sintaxis de cada uno es diferente y está muy influida por lo que el lenguaje debe lograr. Algunos están más optimizados para unas tareas que para otras.

Antes de lanzarme de cabeza, decidí dedicar una hora para deambular por los principales sitios web de programación para saber qué lenguajes de programación recomendaban los programadores. Esta investigación temprana me ayudará a decidir qué lenguaje aprender y qué habilidades practicar.

Dos de los destinos más populares para los programadores son Stack Overflow^[34] y Hacker News^[35]. Aquí empieza mi búsqueda del lenguaje ideal.

Stack Overflow es un sitio web de preguntas y respuestas, un lugar para preguntar: «¿Cómo puedo hacer X?». Los programadores con más conocimientos y experiencia responden a las preguntas con sugerencias específicas, haciendo que Stack Overflow sea el mejor destino para quien necesita ayuda para resolver problemas complicados de programación.

Hacker News es un sitio web de noticias sociales: una colección de vínculos con discusiones asociadas. Los temas de Hacker News cambian minuto a minuto, pero usualmente giran alrededor de nuevos desarrollos en programación, tecnología y negocios, haciendo que este sitio sea ideal para buscar opiniones medio informadas sobre los nuevos desarrollos.

Nuevos lenguajes, librerías y técnicas son creados por programadores de todo el mundo cada día. Algunas combinaciones de tecnologías e ideas son útiles para resolver ciertos problemas y no otros. Por lo regular, no puedes saberlo hasta ponerlas a prueba.

En términos de programación, «lo mejor» depende del problema que intentas resolver y de tus prioridades específicas. En general, el consejo es 1.

elegir herramientas que te permitan resolver el problema con eficiencia, y 2. si tienes opción, elegir las herramientas que te gusta usar. Me parece justo.

Revisar los archivos de Stack Overflow y Hacker News me produjo una sobrecarga informativa: hay demasiada información para procesarla de inmediato, particularmente si no conoces la terminología. Debía reducir datos para encontrar consejos más específicos.

Ahora te mostraré un truco táctico de investigación que muy poca gente conoce: los buscadores populares te permiten limitar la búsqueda a un sitio web específico y evitar la búsqueda en toda la red. El código para hacer esto en Google es parecido a este:

`“search phrase” site:example.com`

Remplaza «search phrase» con el término que deseas buscar y «example.com» con el sitio web en que deseas buscar. Las comillas significan buscar coincidencias exactas del texto entrecomillado. Sin las comillas, Google te mostrará páginas que contienen todas las palabras de la frase, pero no necesariamente en ese orden.

Usando esta técnica, hice búsquedas de muchas variantes de las frases «programación de aplicaciones para la web», «aprenda a codificar» y «programación para principiantes», y luego pasé una hora leyendo los resultados.

Esto encontré: los desarrolladores experimentados de la red actualmente recomiendan comenzar con uno o dos lenguajes comunes. Ruby o Python tienen la reputación de ser fáciles de aprender, poderosos y también dan buenas bases sobre conceptos importantes de la programación. Ruby es un poco más popular entre los programadores dedicados a las aplicaciones para la red, en tanto que Python es más popular entre científicos y matemáticos debido a su amplia variedad de librerías en ciencias, matemáticas y gráficos.

Ruby y Python tienen comunidades activas de desarrolladores, enormes cantidades de recursos gratuitos, libros bien escritos sobre ellos, y programas y herramientas preexistentes que hacen que las características más importantes sean fáciles de elaborar. La elección de uno de estos programas para trabajar parece cuestión de gustos.

Después de leer unos cuantos ejemplos de códigos escritos en cada uno de estos lenguajes, decidí aprender Ruby. Para mi ojo no entrenado, este código parece más limpio, legible y fácil de entender. Puesto que las principales ideas y técnicas aprendidas usando Ruby serán transferidas a otros lenguajes que podría aprender después, pensé que era mejor hacer el proceso de aprendizaje disfrutable desde el principio.

Además, hay programas y herramientas que me gusta usar y requieren Ruby. En particular, Jekyll está escrito en Ruby, de modo que aprenderlo me ayudará a resolver un problema apremiante. Existen herramientas similares para Python, pero es más difícil usarlas.

Elegir un *framework* o infraestructura digital

Los programadores activos de aplicaciones para la red tienen opiniones muy firmes sobre *frameworks*: librerías de código que facilitan las cosas que la mayor parte de las aplicaciones necesitan hacer.

Las librerías son importantes porque las computadoras hacen exactamente lo que les dices. Ni más ni menos.

Eso es difícil porque el código que proveas es todo lo que existe desde el punto de vista de la computadora. Es como la frase de Carl Sagan, el famoso físico: «Si quieres hacer un pay de manzana de cero, primero debes inventar el universo».

El «universo» de tu programa se define por: 1. las instrucciones y comandos en el código base, 2. las librerías que tu programa importa, y 3. el sistema en que corre el programa. Si el código necesario para completar una operación determinada no existe en ninguna parte del sistema, tu programa se congelará o enviará un mensaje de error.

La mayoría de los lenguajes de programación contienen muchas librerías comunes que la mayor parte de los programas requieren, pero pocas herramientas especializadas. Es aquí donde entran en juego los frameworks. En vez de codificar todo desde cero, lo que llevaría mucho tiempo, usar un framework te permite importar y usar librerías probadas y confiables para tareas especializadas, y así te concentras en la esencia de tu aplicación en lugar de reinventar el universo.

Los frameworks pueden ser grandes o pequeños. Algunos incluyen muchas funciones y comandos en un intento por ahorrar esfuerzo al programador, y otros son más pequeños, cubriendo únicamente algunas funciones esenciales.

Hasta este momento, Ruby ofrece varios frameworks para el desarrollo de aplicaciones de la red, entre los que destacan dos por su popularidad: Ruby on Rails^[36] y Sinatra^[37].

Ruby on Rails (el cual suele abreviarse como Rails) es uno de los primeros frameworks importantes para aplicaciones de la red desarrollado para Ruby. Creado por David Heinemeier Hansson en 2004 es, por mucho, el más popular de Ruby y se utilizó para desarrollar varias aplicaciones exitosas en 37signals^[38], empresa especializada en aplicaciones en que Hansson es socio. A la fecha, miles de negocios han desarrollado grandes e importantes aplicaciones de negocios para la red usando Rails.

Rails se apoya fuertemente en «generadores»: programas incorporados que crean grandes cantidades del llamado código *boilerplate* (fragmentos de código repetidos en varias partes del programas sin cambios) con un solo comando. El código *boilerplate* es modificado con base en las necesidades únicas del programador. En vez de pasar horas creando una aplicación de cero, Rails permite a los programadores crear una aplicación funcional sin tanto esfuerzo, siempre y cuando sepan qué hacen.

Por otra parte, Sinatra es un framework mínimo diseñado y desarrollado por Blake Mizerany. En lugar de generadores, Sinatra se concentra en dar al programador unas cuantas funciones simples que la mayoría de las aplicaciones requieren y nada más.

Las aplicaciones hechas con Sinatra se ven y se sienten simples comparadas con las aplicaciones hechas con Rails. Un solo comando de Rails puede generar 10 o más fólderes y 20 o más archivos. En contraste, no es raro que una aplicación hecha con Sinatra quede contenida en un solo archivo. En lugar de generar toneladas de código, que probablemente tenga que ser retirado después, el desarrollo con Sinatra mantiene el proyecto limpio y simple, pues se añade únicamente el código necesario para el trabajo.

Al igual que en el caso de la elección de un lenguaje, elegir un framework es cuestión de gustos y de elegir la herramienta que más convenga al trabajo.

Rails tiende a ser favorecida por los grandes proyectos en que participan varios programadores, y Sinatra se adapta mejor a los proyectos pequeños. Existen muchas características comunes a ambos programas, por lo que un análisis reciente en RubySource.com concluye que, a fin de cuentas, la elección es subjetiva^[39].

GitHub es un código abierto que muchos programadores utilizan para lanzar y mantener sus proyectos. Es fácil hallar ejemplos de aplicaciones escritas con Rails y Sinatra para tener una mejor idea de cada framework, así que pasé otra hora investigando en proyectos públicos.

Existe un riesgo nada trivial: para avanzar en la programación, debes comprometerte con algo. Cuando ya elegiste un lenguaje y un framework, es mucho más fácil comenzar a aprender todo lo que necesitas para escribir un programa. Si no tomas una decisión, puedes pasar años tratando de encontrar el ambiente de programación «perfecto».

Es mejor escoger un lenguaje y un framework que te parezcan atractivos, comprometerte a explorarlos por un tiempo y aceptar las inevitables ventajas y desventajas, en lugar de pasar años «investigando» sin progreso alguno. Pasarse el día navegando en Stack Overflow y en Hacker News *no* es programar.

Al final, decidí empezar con Sinatra. Aunque los generadores de Rails ahorran mucho tiempo si sabes lo que haces, yo *no sé* lo que estoy haciendo.

La intuición me dice que, en este momento, Sinatra es la mejor opción. La sintaxis es clara, simple y fácil de entender. El único código en una aplicación de Sinatra es el que agrega el programador. El framework está bien documentado y es fácil encontrar ejemplos de aplicaciones funcionales en GitHub, así como obtener ayuda en Stack Overflow.

Quizá experimente con Rails algún día. Por lo pronto, empiezo con Sinatra.

Deconstruir el resultado final

Aproximadamente 5 horas de investigación preliminar me han brindado todo lo que necesito para empezar: una deconstrucción de la programación para la red, un lenguaje, un framework y un proyecto específico. Manos a la obra.

«Codificar una aplicación Sinatra que sirva a un sitio web Jekyll» es una

buena meta de desempeño, pero necesito deconstruir lo que integra esa afirmación para determinar qué hacer a continuación. ¿Qué debo hacer?

Otra hora de investigación me lleva a las siguientes conclusiones:

1. Al usar Jekyll se crea el sitio web terminado a partir de archivos locales estáticos. Debo crear una plantilla para el sitio web con etiquetas de formato especiales, y exportar mis archivos desde PersonalMBA.com, lo que se describe en un tutorial creado por Paul Stamatiou^[40].
2. La aplicación hecha con Sinatra puede solventar las peticiones de los visitantes del sitio, entregando los archivos solicitados. Debo escribir esta aplicación desde cero.
3. Tanto el sitio web Jekyll como la aplicación de servidor Sinatra tienen que ser subidas a un servidor anfitrión.
4. Para completar estas tareas, debo averiguar cómo instalar la última versión de Ruby en mi máquina, así como Sinatra y otros programas que necesito.

Este último requisito es un buen ejemplo de la necesidad de obtener las herramientas clave. Si no me las arreglo para instalar Ruby en mi máquina, no llevaré a cabo los siguientes pasos, así que debo empezar por ahí.

Es importante comentar que la tecnología de la red cambia todos los días. Es muy probable que las series de comandos específicos comentados en esta sección sean obsoletos cuando leas esto. No te preocupes: lo importante es el método, no los comandos.

Igualmente, sentirás la tentación de saltarte las secciones de código de este capítulo. Es natural: el código se ve complicado y no reconocerás su significado de inmediato.

Te pido que resistas este impulso. Estos nombres, comandos y símbolos son tan extraños para mí como para ti. Este capítulo trata del proceso por medio del cual averiguarás qué significa todo esto y cómo usarlo. Si tratas de leer el código, sacarás mucho más provecho de este capítulo.

¡Adelante!

Actualizar Ruby

Ya tengo una computadora, lo que es un buen comienzo: no puedes programar sin una.

Hasta este momento, uso una MacBook Air de Apple que corre el sistema operativo OS X 10.6. Una búsqueda rápida indica que el sistema operativo viene con la versión 1.8.7 de Ruby preinstalada, lo cual es una buena noticia: teóricamente, puedo empezar a correr programas Ruby en mi computadora de inmediato.

El problema es que Ruby 1.8.7 no es su última versión. Cuando trato de instalar Jekyll, el sistema me dice que el programa requiere al menos la versión 1.9.1, lo que significa que debo actualizar. De vuelta a Google.

Búsquedas básicas me indicaron que hay dos programas diseñados para manejar las instalaciones de Ruby más fácilmente: rbenv y ruby-build. Ambos programas son mantenidos por Sam Stephenson, desarrollador de Ruby en 37signals. Estos te ayudan a instalar las nuevas versiones de Ruby y le dicen a tu computadora qué versión usar.

Hay un tutorial en la página de documentación de rbenv^[41] que muestra cómo instalar los programas en tu máquina. Así se ven los comandos de instalación:

```
$ cd
$ git clone git://github.com/sstephenson/ruby-build.git
$ cd ruby-build
$ sudo ./install.sh
$ cd...
$ git clone git://github.com/sstephenson/rbenv.git.rbenv
$ mkdir -p /.rbenv/plugins
$ cd /.rbenv/plugins
$ git clone git://github.com/sstephenson/ruby-build.git
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> /.bashprofile
```

```
$ echo 'eval "$(rbenv init -)"' >> ~/.bashprofile
$ exec $SHELL
$ rbenv install 1.9.3-p125
$ rbenv rehash
$ rbenv global 1.9.3-p125
```

Esto atemoriza, pero se trata solo de una lista de comandos. Analicemos las cosas paso a paso.

Estos comandos se introducen en un programa llamado Terminal, que viene preinstalado en las computadoras Apple. Si alguna vez has visto a los *hackers* de las películas tecleando frenéticamente en computadoras que muestran largas líneas de texto, esas computadoras están corriendo un programa Terminal^[42].

Abro Terminal e introduzco el primer comando:

```
$ cd
```

Este comando es fácil de entender. El signo \$ es lo que Terminal despliega cuando está listo para recibir una nueva orden, así que ya estaba allí desde antes; `cd` es una abreviatura que significa «change directory» (cambiar directorio); «directorio» es otra forma de llamar a los *fólderes*. Una búsqueda rápida me indica que el signo representa el «*fólder principal del usuario*», es decir, el de mi computadora en que se almacena mi perfil de usuario.

Escribo el comando y presiono la tecla Enter. Ahora, Terminal despliega esto:

```
joshkaufman $
```

Eso es bueno: este es mi directorio principal. Hasta ahora, todo va bien. Tecleo el segundo comando:

```
$ git clone git://github.com/sstephenson/ruby-build.git
```

La computadora responde:

```
git: command not found
```

Parece que el programa git no está instalado en mi máquina. Tendré que instalarlo.

¿Qué es un «git»?

Después de buscar instrucciones para instalar git en el sistema operativo Mac OS X, encuentro Heroku Toolbelt^[43]. Heroku^[44] es una empresa dedicada a albergar en sus servidores aplicaciones web, de modo que tienen un interés especial en facilitar las cosas para que los desarrolladores creen aplicaciones para la red.

Heroku Toolbelt instala unos cuantos programas comunes con herramientas para el desarrollo que los programadores necesitan para elaborar aplicaciones en Heroku. Así que git^[45] es uno de esos programas.

Bajo el paquete de instalación, lo corro y recibo una confirmación de que todo se ha instalado correctamente. Llega el momento de volver a dar la orden:

```
$ git clone git://github.com/sstephenson/ruby-build.git
```

Obtengo el siguiente *output*:

```
Cloning into Ruby build...
remote: Counting objects: 1004, done.
remote: Compressing objects: 100% (453/453), done.
remote: Total 1004 (delta 490), reused 937 (delta 431)
Receiving objects: 100% (1004/1004), 108.77 KiB,
done.
Resolving deltas: 100% (490/490), done.
```

¡Éxito! La palabra «done» (hecho) es una buena señal. Además, no recibí ningún mensaje de error. Adelante.

Sigo con el resto de los comandos. Basado en la información del tutorial de instalación, solo estoy bajando los archivos necesarios, usando un comando llamado «echo» para agregar, de modo automático, un pedacito de texto a los archivos de configuración de mi computadora; luego reanudo un programa de mi computadora llamado «shell», que guarda los cambios. Una vez que shell reinicia mi computadora, rbenv y ruby-build quedan instalados. ¡Perfecto!

Ahora ha llegado el momento de instalar la última versión de Ruby:

```
$ rbenv install 1.9.3-p125
```

El programa baja automáticamente el código fuente de Ruby y lo instala, emitiendo una increíble cantidad de información en el programa Terminal en el proceso. (Ahora comienzo a sentirme como un verdadero programador de las películas de Hollywood^[46]).

```
$ rbenv rehas
```

Esta orden, según la documentación, ayuda a que la computadora reconozca la nueva versión de Ruby instalada.

```
$ rbenv global 1.9.3-p125
```

Este comando define la versión 1.9.3-p125 como la versión de Ruby que se usará en esta computadora. El tutorial me indica que corra este comando para asegurarme de que mi computadora usa la nueva versión:

```
$ ruby -v
```

He aquí lo que obtengo:

ruby 1.9.3-p125 (2012-02-16 revision 34643) [x8664-darwin11.3.0]

¡Éxito! Es justo lo que debía obtener, según el tutorial.

De acuerdo con la documentación, si quiero instalar una nueva versión de Ruby en mi computadora, solo tengo que correr de nuevo `rbenv install`, `rbenv rehash` y `rbenv global`. Fácil.

Aunque las órdenes me intimidaban al principio, son en realidad bastante simples: lo que parece «bla-bla-bla» son abreviaturas. Cuando te enteras de qué representan, los comandos mismos se vuelven fáciles de comprender.

Recuerda que nadie nace sabiendo esto. La mayor parte del tiempo, lo único que necesitas es pasar algunos minutos leyendo la documentación y poner en práctica lo leído^[47].

Instalar las librerías de Ruby (gems)

Ahora que ya está instalada la última versión de Ruby, llega la hora de instalar las librerías que necesito, incluyendo Sinatra.

Las librerías de Ruby se llaman *gems*, y son fáciles de instalar. Aquí está el comando que permite instalar la gem Sinatra:

```
$ gem install sinatra
```

Para actualizar Sinatra, debo teclear el siguiente comando:

```
$ gem update sinatra
```

No obstante, antes de instalar demasiadas librerías debo asegurarme de que el programa `gem` está actualizado. Dado que mi computadora llegó a casa con una versión vieja de Ruby, parece probable que el *software* también requiera actualizarse.

Después de investigar un poco, encontré el comando que permite actualizar el programa `gem` de Ruby:

\$ gem update—system

Fácil.

Al correr el comando de instalación de gem, noto que el comando también instala gems adicionales, como rack, rack-protection y tilt. Estas librerías extra se llaman *dependencias*. Sinatra depende de ellas para correr, así que la orden las instala automáticamente.

Usar los libros

Ahora que estoy listo para correr las aplicaciones de Ruby, decidí comprar dos libros de referencia general sobre Ruby. Ambos eran recomendados en Stack Overflow: *The Well-Grounded Rubyist* (2009), de David A. Black, y *Eloquent Ruby* (2011), de Russ Olsen. Estos libros están diseñados para presentar a los lectores los conceptos y técnicas comunes de Ruby, así como algunos textos de referencia básicos.

También compré *Sinatra: Up and Running* (2011), de Alan Harris y Konstantin Hasse. Este libro es una introducción al framework Sinatra. Aunque Sinatra está muy bien documentado en línea, este libro contiene muchos ejemplos, lo que te facilitará el uso de sus funciones más comunes.

Al buscar los libros, encontré también varios sitios web de referencia sobre la sintaxis básica de Ruby:

- La guía oficial para principiantes en Ruby^[48].
- El actualización en Ruby^[49].
- La guía para la seguridad en Ruby^[50].

Es bastante material de referencia para empezar, así que dedico 90 minutos a revisar todo lo reunido hasta ahora.

Al repasar los libros miré atentamente sus índices para advertir qué ideas y términos parecen importantes. También me fijé en conceptos que aparecen una y otra vez, así como en el orden en que los presentaban. Leí los encabezados y algunos recuadros. Cuando terminé con los libros, hice lo mismo con los sitios web.

He aquí lo que aprendí: además de las variables, condicionales, excepciones y otros conceptos básicos de la programación, Ruby está construido alrededor de dos ideas esenciales: *objetos* y *métodos*.

Los objetos son los sustantivos del mundo de la programación: son cosas a las que podemos hacerles algo o con las que podemos hacer algo. Digamos que queremos crear una nueva variable en Ruby llamada *firstname*, y que esta variable contenga mi nombre. En Ruby, esa instrucción se ve así:

```
firstname = "Josh"
```

Bastante simple. Al poner mi nombre entre comillas, le digo a Ruby que *firstname* es un *string* o cadena de caracteres: una secuencia de caracteres alfanuméricos. Eso convierte a *firstname* en un objeto clase «*string*». (Una clase es solo un tipo específico de objeto con ciertas características).

Los *strings* no son la única clase de objetos. Aquí presentamos un objeto que pertenece a la clase de enteros:

```
million = 1000000
```

Si en programación los objetos son como los sustantivos, los métodos son los verbos: son cosas que podemos hacer a (o con) un objeto.

Digamos que tengo dos objetos *string* que contienen mi nombre y apellido:

```
firstname = "Josh"  
lastname = "Kaufman"
```

Puedo usar un signo de más (+) para *concatenar* estos *strings*, lo cual es un término pomposo para decir «juntarlos»:

```
fullname = firstanme + lastname
```

Pregunta: ¿qué contiene *fullname*? Si contestaste «Josh Kaufman», estás en un error.

Recuerda, la computadora hará exactamente lo que le pidas. No le dijimos que añadiera un espacio entre «Josh» y «Kaufman», así que no lo hizo. *Fullname* equivale a «JoshKaufman».

Si deseamos corregir este pequeño error, debemos cambiar el código para agregar un espacio:

```
fullname = firstname + " " + lastname
```

El signo + es un método y su modo de funcionar depende de los objetos con que lo usemos. Si lo usamos con enteros, suma en lugar de juntar:

```
sum = million + million
```

¿Cuánto representa la suma?: 2000000.

Los métodos integrados a Ruby te permiten hacer muchas cosas de inmediato. Digamos que quieres ver cómo luce mi nombre a la inversa. En lugar de hacerlo a mano o escribir mi propio programa para revertir las letras, puedo usar el método de reversión disponible para cada objeto *string*:

```
fullname.reverse
```

Aquí está el *output*: namfuaKhsoJ.

También puedo usar más de un método a la vez. Si quiero revertir las letras de mi nombre y convertir todos los caracteres a minúsculas al mismo tiempo, puedo correr el siguiente comando:

```
fullname.reverse.downcase
```

Output: namfuakhsoj. ¡Perfecto!

Una buena parte del aprendizaje del código Ruby parece implicar el uso, creación y manipulación de objetos, clases y métodos. El lenguaje tiene muchas de estas cosas incorporadas. Ruby te permite crear, modificar o eliminar objetos, clases y métodos a voluntad, lo que da a este lenguaje mucho poder y flexibilidad^[51].

La documentación oficial de Ruby^[52] contiene el canon de todos los objetos y métodos disponibles. Ver la vista por encima es desolador, pero nos ayuda a darnos cuenta de que no tenemos que usarlos todos. Por el contrario, es seguro para nosotros ignorar la mayoría de estos objetos y métodos por ahora. Son meras opciones listas para ser usadas cuando las necesites.

La documentación sirve para otra cosa: cuando tratas de hacer algo que Ruby no entiende, el mensaje de error correspondiente te dirá en dónde tronó el programa.

Digamos que tratamos de correr un programa como este:

```
animal = "Zebra"  
number = 7  
puts animal + number
```

El comando «puts» es otra forma de decir «imprime». Solo queremos que el programa despliegue lo que `animal + number` significa.

He aquí lo que obtengo cuando corro el programa:

```
TypeError: can't convert Fixnum into String from  
program.  
rb:3:in '+'
```

En términos no computacionales: no puedes utilizar la aritmética para añadir un número a una palabra de modo que tenga sentido, así que la computadora despliega el error. Es como tratar de dividir entre cero: no se puede, así que el programa se detiene.

Para arreglarlo necesitamos convertir el número en un *string* para que el método `+` encadene dos variables en lugar de usar la aritmética, o modificar

el programa para que haga otra cosa.

A continuación, muestro un ejemplo del programa revisado:

```
animal = "Zebra"  
number = 7.to_s  
puts animal + number
```

Cuando corremos el programa, obtenemos el *output* «Zebra7». El método integrado `.to_s` convierte al número 7 en un *string*, de modo que Ruby puede juntar.

También podríamos hacer algo completamente distinto, como esto:

```
animal = "Zebra"  
number = 7  
number.times { puts "#{animal}" }
```

Este es el *output*:

```
Zebra  
Zebra  
Zebra  
Zebra  
Zebra  
Zebra  
Zebra
```

Solo usamos un bucle básico condicional, ya incorporado a Ruby: «`number.times`» significa «haz esto X veces, en donde X es igual al valor del número variable». Si cambiamos el valor de `animal` o de `número`, cambiaremos el *output*. (Sí, puedes modificar este programa para imprimir «vombatidae» mil millones de veces si lo deseas).

Comentarios y eliminación de errores (*debugging*)

Al leer, capté otro concepto básico en Ruby: *comentar*. Siempre que das inicio a una línea de programa con el signo # (llamado por lo regular «signo de libra», o también conocido con los nombres de «almohadilla» «numeral» o «gato»), Ruby interpreta esa línea como un comentario y se la salta.

Añadir comentarios a un programa lo hace mucho más fácil de seguir, ya que puedes explicar en lenguaje normal lo que tratas de hacer. Aquí el ejemplo de cómo se vería mi programa de impresión de animales con comentarios:

```
# Asign variables
animal = "Wombat"
number = 1000000000
# Print loop
number.times { puts "#{animal}" }
```

Comentar es también una técnica para solucionar problemas: puedes comentar algunas líneas del código para aislarlos. Si se combina con afirmaciones tipo print o puts, puedes seguir la ejecución de un programa paso a paso para asegurarte de que todo trabaja como debe ser.

Después de unas 8 horas de investigación e instalación, estoy corriendo la última versión de Ruby: puedo instalar cualquier librería que necesite y tengo una idea básica de cómo funciona el programa.

Es importante destacar que, en realidad, no he programado nada sustancial todavía. Hasta el momento, el tiempo se ha utilizado para investigar, instalar Ruby y darme una idea de cómo es escribir un programa en Ruby.

Exploremos programas más complejos.

Todo a prueba con IRB

Durante mi investigación en Stack Overflow, encontré un tutorial en línea de Ruby llamado *Learn Ruby the Hard Way*^[53], de Rod Sobers y Zed Shaw. El tutorial ilustra el funcionamiento con ejemplos de programas simples ya

hechos. Te pide que los modifiques y corras para obtener resultados específicos. Si no obtienes el resultado correcto, tu trabajo consistirá en experimentar modificando el programa hasta obtener lo que desees.

Este método de codificar, probar, correr y corregir es un buen ejemplo de *bucle de retroalimentación rápida*. Cuando corres un programa, la computadora te hará saber en milisegundos si funcionó o no. Si hay un error en el código, puedes corregirlo y correr el programa de nuevo, probando muchas variantes en solo un minuto.

Los primeros capítulos de *Learn Ruby the Hard Way* tienen que ver con la instalación de un programa de edición de texto básico y con cómo aprender a usar IRB, que corre los programas Ruby en tu propia computadora.

Funciona de la siguiente manera: escribes tu programa en el editor de textos y lo salvas en un archivo. (Asumamos que su nombre es `program.rb`.) Cuando estés listo para correrlo, teclea esto en tu máquina:

```
$ irb program.rb
```

IRB correrá el programa y te dará el resultado. También te mostrará los pasos de la computadora para llegar a ese resultado, lo que es útil a la hora de corregir errores. Si el programa está mal, IRB nos mostrará un mensaje de error detallado.

Learn Ruby the Hard Way comienza con la asignación de variables, aritmética básica, manipulación de *strings* y la conformación de afirmaciones condicionales básicas, muy parecidas a los ejemplos que acabo de utilizar. Se trata de una forma muy bien estructurada y lógica para aprender lo básico.

Mi plan original era leer *Eloquent Ruby* y *The Well-Grounded Rubyist*, para después completar los ejercicios de *Learn Ruby the Hard Way* antes de escribir mi primer programa «real». Sin embargo, cuando iba por la lección número 10, me percaté de algo importante: me estaba desesperando y perdía interés.

He aquí la esencia del problema: estoy copiando programas que otros crearon y resolviendo problemas definidos también por otros. Estos programas a veces son interesantes, pero no resuelven mis problemas. Programar comienza a sentirse como un ejercicio académico y no como una

habilidad útil. Necesito salir del modo de investigación para entrar en el de conformación.

No necesito leer todos los libros, cursos, tutoriales y demás recursos descubiertos antes de programar. Necesito comenzar de inmediato a escribir programas reales, para luego consultar con mis fuentes cuando tenga problemas.

Hora de ensuciarse las manos...

Aplicación #1: Un sitio web estático en Sinatra

Ya tengo una idea de cuál será mi primera aplicación para la web: se trata de una aplicación Sinatra que sirve a un sitio básico HTML. Mi nivel de desempeño objetivo para esta aplicación será:

1. Crear una aplicación funcional Sinatra capaz de llevar un sitio web simple a un usuario final (lector).
2. Probar la aplicación en mi computadora para asegurarme de que funciona.
3. Desplegar esa aplicación para ser producida en Heroku, dándole «vida» para que lectores de verdad puedan usarla.

Eso es todo. Nada rimbombante, solo un programa Sinatra muy simple corriendo en un servidor público.

¿Por dónde comenzar? Repasemos mi lista de verificación para la práctica:

1. Elige un proyecto que ames realmente.
2. Concentra tu energía en una sola habilidad en un momento determinado.
3. Define tu nivel de desempeño.
4. Deconstruye la destreza hasta dar con los elementos que la componen.
5. Obtén las herramientas indispensables.
6. Elimina las barreras que te impiden practicar.
7. Dedica un tiempo para la práctica exclusiva de la habilidad.
8. Crea ciclos de retroalimentación rápida.
9. Practica en periodos breves con ayuda de un reloj.
10. Enfatiza cantidad y velocidad.

Tengo un solo proyecto bien definido. He deconstruido la habilidad y sé cómo se verá este programa cuando termine. Eso me lleva a las herramientas básicas: ¿necesito algo que todavía no tengo para completar este proyecto?

Resulta que sí: no tengo cuenta de Heroku. Eso es fácil de arreglar: visito Heroku.com hago clic en el botón que dice Sign Up (darse de alta), verifico mi dirección de correo electrónico y creo una contraseña.

Dado que ya bajé Heroku Toolbelt (el programa para instalar git), el gem Heroku ya está en mi computadora, así que estoy listo.

Con base en las instrucciones, solo hay una última cosa que debo hacer para que mi máquina hable con Heroku: generar algo llamado «clave SSH», archivo especial que sirve como contraseña. Una vez que lo tenga, se supone que debo subirlo a Heroku para que el sistema reconozca mi computadora.

Por fortuna, Heroku tiene un tutorial sobre cómo hacer esto^[54]. Ingreso el siguiente comando para generar la clave:

```
$ ssh-keygen -t rsa
```

... y el siguiente comando para entrar a Heroku:

```
$ heroku login
```

... y este comando para agregar la clave a mi cuenta Heroku:

```
$ heroku keys:add
```

Bien, ya estoy adentro. ¿Cómo empiezo a escribir la aplicación?

Crear la aplicación básica

Llegó la hora de revisar la documentación de Heroku. Genial, hay dos guías que parecen útiles:

- «Cómo empezar con Ruby en Heroku.»^[55]

- «Cómo desplegar aplicaciones basadas en Rack.»^[56]

Con base en estas guías, parece que necesito:

1. Crear los archivos del programa en mi computadora.
2. Añadirlos a un «depósito git» (quién sabe qué sea eso).
3. Usar el comando `git push heroku master` para enviar la aplicación terminada a Heroku.

Afortunadamente, el tutorial incluye un ejemplo, ¡y se trata de una aplicación hecha en Sinatra! Esto será más fácil de lo que pensé.

Creo un folder nuevo en mi computadora. Se llama «raíz», y todo archivo del programa debe guardarse allí.

Luego, abro el editor de texto (uso TextMate^[57]) y creo tres archivos siguiendo las instrucciones:

```
application.rb
config.ru
Gemfile
```

La esencia del programa irá a `application.rb`. Las aplicaciones Ruby siempre terminan en `.rb`.

`config.ru` es adónde van los datos de configuración Rack. Recuerda que Sinatra está construido sobre Rack, de modo que debe tener un archivo de configuración por separado. Los archivos «rackup» terminan en «`.ru`».

Gemfile es el lugar en donde especificaremos qué gems usará el programa. El tuyo solo tendrá un gemfile, así que siempre se llama «Gemfile». Parece bastante sencillo.

Después de crear los archivos, la documentación de Heroku sugiere escribir un programa básico tipo «Hola, mundo» para probar la instalación. He aquí lo que va en `application.rb`:

```
require 'sinatra'
get '/' do
```

```
“Hello world”  
end
```

He aquí lo que va en config.ru:

```
require './application.rb'  
run Sinatra::Application
```

Y aquí está lo que va en el gemfile:

```
source 'http://rubygems.org'  
gem 'sinatra'
```

No es mucho código y resulta bastante fácil de comprender.

- El gemfile dice al servidor que incluya la gem Sinatra para correr la aplicación. El gem se bajará de RubyGems.org.
- El archivo config.ru instala la aplicación principal y luego ejecuta el programa.
- Una vez que el programa corre, mostrará «¡Hola, mundo!» a cualquiera que visite «/», forma abreviada para referirnos a la página principal de un sitio web.

¿En verdad es así de fácil?

El tutorial me dice que falta algo: guardar los archivos en un depósito git. No estoy muy seguro de qué significa eso, pero sé que ya está instalado git y ellos me dicen qué comandos usar:

```
$ git init  
$ git add -A  
$ git commit -m “Initial Commit”
```

El primer comando crea un nuevo depósito git en el folder raíz. El comando «add -A» añade todos los archivos del folder al depósito. La orden «commit

–m» lleva los archivos al depósito, junto con un mensaje del programador en que se indica qué se deposita. (Todavía no conozco la diferencia entre «add» y «commit», así que tomo nota para investigarlo después).

Después de teclear estos comandos, la computadora me dice:

```
[master (root-commit) 8ed1099] Initial commit
3 files changed, 9 insertions(+), 0 deletions(-)
create mode 100644 Gemfile
create mode 100644 application.rb
create mode 100644 config.ru
```

¡Parece que funcionó! Solo hay algo más por hacer: crear un nuevo servidor vacío en Heroku para luego «empujar» mi programa a ese servidor.

Para crear el servidor, introduzco la siguiente orden:

```
$ heroku create
```

Y obtengo la siguiente respuesta:

```
Creating shielded-springs-2049... done, stack is stack is bamboo-ree-
1.8.7
http://shielded-springs-2049.herokuapp.com/ | git@
heroku.com:shielded-springs-2049.git
Git remote heroku added
```

¡Éxito! El *output* detalla el *software* en el que corre el servidor y el mensaje me da la dirección URL pública del servidor. Ahora el comando final:

```
$ git push heroku master
```

Si todo sale bien, habré «empujado» mi primera aplicación para su producción en Heroku.

He aquí lo que obtengo:

```
Heroku receiving push
Ruby/Sinatra app detected
Gemfile detected, running Bundler version 1.0.7
Unresolved dependencies detected; Installing...
Using-without development:test
! Gemfile.lock will soon be required
! Check Gemfile.lock into git with `git add Gemfile.
lock`
! See http://devcenter.heroku.com/articles/bundler
Fetching source index for http://rubygems.org/
Installing rack (1.4.1)
Installing rack-protection (1.2.0)
Installing tilt (1.3.3)
Installing sinatra (1.3.3)
Using bundler (1.0.7)
Your bundle is complete! It was installed into ./bundle/gems/
Compiled slug size 500k
Launching... done, v4
http://shielded-springs-2049.heroku.com deployed to Heroku
```

Ahora, llega el momento de la verdad... abro el navegador de Internet, voy hasta <http://shielded-springs-2049.heroku.com>, y esto es lo que veo:

«¡Hola, mundo!».

¡Victoria!

¡Advertencia! ¡Advertencia!

El programa funcionó, pero también me llegó un mensaje de advertencia. ¿Qué es Gemfile.lock?

La documentación Ruby de Heroku muestra que el sistema usa una librería llamada bundler^[58] para instalar las gems en Heroku. Es un gem, así que puedo instalarlo en mi computadora con el siguiente comando:

```
$ gem install bundler
```

Es necesaria la librería bundler porque Heroku no instala sin más gems en tu aplicación. Por razones de seguridad, Heroku no me da el mismo nivel de acceso computacional de mi máquina, así que no hay manera de correr la orden `gem install sinatra` directamente en mi cuenta.

En lugar de darme (o a cualquier otro usuario) niveles peligrosos de acceso al sistema, Heroku usa bundler para instalar los gems especificados en Gemfile. Cuando identifico qué gems quiero instalar en mi aplicación, escribo el siguiente comando:

```
$ bundle install
```

Este comando crea un nuevo archivo llamado «Gemfile.lock» en tu programa. Cuando subes tus archivos a Heroku, el sistema revisa el Gemfile y el Gemfile.lock para verificar que, por seguridad, sean iguales, baja los gems y los instala.

Si te fijaste en el *output* obtenido cuando empujé el programa a Heroku, puedes ver que el sistema instaló bundler automáticamente como un elemento subordinado. En lugar de mandarme un mensaje de error, los ingenieros de Heroku agregaron una excepción al programa para instalarlo automáticamente y mostrar una advertencia en lugar de congelarse.

El sistema funcionó esta vez, pero en el futuro tendré que añadir Gemfile.lock al depósito git antes de empujar la aplicación. ¡Es bueno saberlo!

Sinatra toma el escenario

Ahora que mi sencilla aplicación corre sin problemas, finalmente averiguaré cómo funciona Sinatra. La documentación de Sinatra^[59] es muy clara y está

llena de ejemplos, de modo que decidí comenzar por ahí.

La esencia de las aplicaciones Sinatra se llama *ruta*. La mejor manera de entender este concepto es por medio del siguiente ejemplo.

Nuestra aplicación Sinatra básica tiene una sola ruta, que contiene la «raíz» de nuestro pequeño sitio web. Por lo general, los usuarios de Internet se refieren a la raíz del sitio web como página de inicio de un sitio web.

Si visitas `google.com` o `yahoo.com`, tu navegador envía una solicitud a los servidores de Google y Yahoo. Esta solicitud se llama *petición GET*, y le pide al servidor que muestre qué hay en el directorio raíz del sitio. El *protocolo* o formato que la computadora utiliza para enviar la solicitud se llama HTTP, siglas de «*hypertext transfer protocol*» (protocolo de transferencia de hipertexto). Eso significa «`http://`», lo que sueles ver al principio de las direcciones de la red.

GET es el tipo más común de solicitud HTTP, pero no es el único. Existen otros tres «verbos» HTTP:

- POST: envía un recurso al servidor.
- PUT: actualiza un recurso en el servidor.
- DELETE: remueve un recurso del servidor.

Si alguna vez publicaste un texto en un sitio web, tus ingeniosos comentarios fueron enviados al servidor usando un comando *POST*. Si cometiste un error y editaste tu comentario, la actualización se envió mediante una orden *PUT*. Si decidiste que el comentario era tonto y preferiste borrarlo, el navegador envió al servidor el comando *DELETE*.

Las rutas que contienen *GET*, *POST*, *PUT* y *DELETE* son la esencia del funcionamiento de Sinatra. Cada ruta que creas es condicional: «Haz X si se recibe un comando *GET/POST/PUT/DELETE* en la ruta Y».

Las rutas de Sinatra también contienen variables, llamadas *parámetros*. Por lo regular, las aplicaciones Sinatra los usan como *inputs* para el proceso contenido en cada ruta.

Modifiquemos nuestro sencillo programa «¡Hola, mundo!» para saludar al usuario por nombre. He aquí una ruta que hará eso:

```
get '/hello/:name' do
```

```
“Hello, #{params[:name]}!”  
end
```

Puedes ver esto en acción en <http://first20hours.com/hello/name>. Siéntete libre de sustituir «name» con tu nombre. ¡Funciona!

Esta aplicación trabaja permitiendo que el servidor busque todo lo que esté en la parte del nombre («name») de la ruta y luego lo utiliza en la aplicación. El comando en la ruta es una simple instrucción que despliega el parámetro «name» al usuario.

Sinatra permite que le pongas nombres a los parámetros (como: name), pero también tiene un parámetro aleatorio (también llamado «splat») que puede contener cualquier cosa. A continuación mostramos cómo utilizarlo en nuestra aplicación «Hola» modificada:

```
get '/hello/ *' do  
  “Hello, #{params[:splat]}!”  
end
```

Muy bien. Entre los parámetros nominados y los aleatorios, puedes crear algunas rutas inteligentes. Las que creas determinan cómo funciona tu aplicación Sinatra.

Suficiente para saber cómo escribir un programa que satisface mi meta de nivel de desempeño. Dado que Jekyll crea los archivos que la aplicación entregará en respuesta a la solicitud GET del usuario, solo necesito escribir algunas rutas que acepten esas peticiones, encontrar el archivo correcto en el sistema y entregarlo al lector.

Con base en la documentación de Jekyll, el programa coloca las páginas web terminadas en un folder llamado «site», en el directorio raíz. La ruta a esa página es generada automáticamente por Jekyll. Si queremos que la página explicativa sobre nuestro sitio web esté disponible en <http://example.com/about>, disponemos una ruta de /about en nuestros archivos Jekyll, y el programa creará el archivo en: `_site/about/index.html`, en el directorio raíz de nuestro sitio web.

Eso significa que debo crear una nueva ruta en Sinatra para leer un archivo en respuesta a la solicitud de usuario GET. He aquí cómo lucen las cosas:

```
# Index handler
get '/' do
  ## File.read("_site/index.html")
end

# Post handler
get '/ * /?' do
  ## File.read("_site/#{params[:splat]}/index.html")
end
```

`File.read ("")` es un comando integrado a Ruby. `File` es un objeto, y `.read` es un método. El uso es bastante sencillo: lo que va entre las comillas ("") es la localización del archivo que quieres que el programa lea, en relación con el folder raíz de la aplicación. Fácil.

¿Qué pasa si el archivo no existe? Eso crea una excepción, y Sinatra tiene dos rutas básicas para los errores integradas: *not_found* y *error*. Hagamos que ambas rutas nos lleven de vuelta a la misma página de error:

```
not_found do
  ## File.read("_site/error/index.html")
end

error do
  File.read("_site/error/index.html")
end
```

Todo lo demás permanece igual. No haré cambios a `config.ru` o a nuestro Gemfile. Solo añado las nuevas rutas al depósito Git del programa,

introduzco los cambios y luego empujo el programa actualizado a Heroku. Listo.

¿Quieres ver cómo se ve nuestro programa actualizado en acción? Visita PersonalMBA.com: el sitio corre ahora Jekyll en lugar de WordPress usando este mismo programa. Al usar uno de prueba de carga llamado seige, mi sitio web puede ahora servir a unos dos mil lectores concurrentes sin problemas. La mayor parte de las solicitudes son entregadas en 18 o 25 milisegundos, así que ahora mi sitio web está perfectamente protegido contra el congelamiento por tráfico pesado.

Mi primera aplicación web funcional está completa. Me tomó cerca de una hora encontrar la forma de realizar el proceso anterior, y otra averiguar cómo transferir la información de mi sitio web fuera de WordPress para meterla en Jekyll.

Tiempo total para lograr el objetivo: unas 10 horas, lo que incluye mi investigación y el repaso de los conceptos de programación. ¡No está mal!

Aplicación #2: Codex, una base de datos de notas personales

Mi primera aplicación funciona y funciona bien. Es sencilla, pero en este caso es una ventaja. Mientras más partes móviles existan, mayores serán las probabilidades de que el programa falle.

Ahora veamos algo un poco más complejo.

¿Recuerdas que hablábamos de las bases de datos? Las páginas web básicas no pueden actualizarse solas, de modo que no pueden almacenar información. La primera aplicación funcionaba porque los archivos eran estáticos: no cambian. Cualquier cambio a los archivos se hace vía Jekyll, programa aparte. La aplicación es rápida y estable porque no depende de una base de datos.

¿Qué sucede con las aplicaciones que usan bases de datos? Las bases son una parte muy importante de las aplicaciones para la red en general, de modo que debo entender cómo funcionan. Para lograrlo, necesito trabajar en un proyecto que dependa de una base de datos.

Una de las aplicaciones que uso a diario se llama Backpack^[60], y fue desarrollada por 37signals. El mayor beneficio de Backpack es crear

«páginas» que contienen prácticamente cualquier cosa: texto, listas, imágenes, archivos, etcétera. Cuando guardas información en una página de Backpack, puedes acceder a ella más tarde desde cualquier computadora, dado que toda la información se guarda en la base de datos de la aplicación.

Me pregunto: ¿puedo crear algo semejante? Vale la pena intentarlo. ¿Pero cómo empiezo?

Mientras investigaba Jekyll, leí un ensayo de Tom Preston-Werner, el programador que creó Jekyll, titulado «Readme Driven Development^[61]». La esencia del texto consiste en que, contra las modas en administración de proyectos que suelen arrasar con la industria del *software* cada pocos años, la mejor manera de crear una aplicación es escribir un documento «readme» antes de hacer cualquier otra cosa.

Un documento tipo readme es un archivo que los programadores incluyen en el folder raíz de la aplicación junto con el código. Este archivo contiene información sobre cómo instalar, configurar y usar el programa.

Los archivos readme son importantes porque muchos programas no se explican por sí mismos. Sin un poco de documentación, suele ser muy difícil averiguar cómo se usa el programa. Meterse en el código para averiguarlo no es tan eficiente como leer una explicación detallada escrita por el programador original.

Tom piensa que es mejor escribir el readme primero, antes de escribir el código. La mayoría de los programadores codifican primero y luego (tal vez) escriben el readme. Eso implica desperdiciar una oportunidad: escribir primero la documentación te ayuda a darte cuenta de cómo funcionará exactamente el programa. El readme puede ser tanto una herramienta de diseño como una herramienta para la documentación.

Eso tiene sentido para mí. Una de las técnicas para el desarrollo de productos que aprendí mientras trabajaba en *The Personal MBA* fue escribir el prospecto de venta antes de crear la oferta misma, no después. Al pensar en qué pueden querer los compradores e incorporarlo en tu prospecto de ventas, obtienes una comprensión más completa de lo que ha de ser un producto para atraer a los clientes. La investigación de mercado informa al desarrollo mismo de la oferta.

Tomé un cuaderno y escribí una lista de lo que quería que la aplicación hiciera, así como de las cualidades que debería tener:

- El programa es una aplicación sencilla, de referencia, para tomar notas.
- La aplicación está diseñada para un solo usuario.
- La aplicación utiliza Sinatra y una base de datos para crear, salvar, actualizar y borrar registros de páginas.
- La aplicación permite al usuario crear páginas con formato sofisticado, como tipos de letra en negrita, cursiva, encabezados, etcétera.
- La aplicación requiere de una contraseña para el acceso, y mantiene la información de la base de datos tan segura como es posible.
- La aplicación tiene un buen aspecto.
- La aplicación puede ser fácilmente desplegable en Heroku o en cualquier otro servidor semejante.

Mi aplicación se llamará Codex, viejo término que significa «libro», pues la aplicación se usará en principio para guardar información de referencia, listas y cosas semejantes.

En programación, el término utilizado para referirse a este tipo de aplicación es CRUD (Create, Read, Update, Delete: crear, leer, actualizar, borrar). Vale la pena mencionar que estas funciones son básicamente lo mismo que los comandos *GET*, *POST*, *PUT*, *DELETE*, así que construir este tipo de aplicación es posible usando rutas de Sinatra. La gran diferencia es la introducción de una base de datos.

¿Qué tipos de bases de datos están disponibles para Heroku? No lo sé: de vuelta a la documentación.

Heroku siempre usa una base de datos llamada Postgres^[62]. A cada nueva aplicación se le asigna de modo invariable una pequeña base de datos para el desarrollo. Eso funciona para mí, pero ¿cómo la uso y qué hago para probar el programa en mi propia computadora?

Te presento DataMapper

Para responder estas preguntas, decidí acudir a Stack Overflow. El consenso indica una librería llamada DataMapper^[63], pues facilita de sobremanera el

desarrollo.

DataMapper es un tipo de programa conocido como «mapeo objeto-relacional» (abreviado normalmente como ORM^[64], por sus siglas en inglés). Los ORM resuelven un importante problema de los programadores: las bases de datos suelen usar su propio lenguaje, distinto del utilizado por el programador para crear la aplicación. El lenguaje más común entre las bases de datos es SQL^[65], pero existen cientos.

Asumamos que somos programadores de Amazon.com y que deseamos desplegar una lista de los libros de J. K. Rowling, autora de la serie Harry Potter. He aquí cómo se vería el comando escrito en SQL:

```
SELECT * FROM Book WHERE author = "J.K. Rowling" ORDER  
BY title;
```

Este comando recupera todos los registros de la base de datos de libros que tienen un campo de autor que contiene «J. K. Rowling», y los devuelve ordenados alfabéticamente por el título.

Desafortunadamente, lograr que SQL, o cualquier otro lenguaje para bases de datos, se lleve bien con Ruby, puede ser complicado. Ya es bastante difícil programar en un lenguaje, y ni hablar de lo que resultaría hacerlo en dos al mismo tiempo.

Eso pasaba antes de que llegaran a la escena los ORM: permiten al programador escribir el código en otro lenguaje, que el ORM traduce luego al lenguaje de la base de datos. Mucho más fácil.

Entonces, DataMapper es una librería que hace mucho más fácil la comunicación con las bases de datos usando Ruby. De entrada, DataMapper provee muchas funciones útiles para crear, leer, actualizar y borrar registros de bases de datos. Puesto que DataMapper existe desde hace tiempo y ha sido probado exhaustivamente, en la mayoría de los casos es más confiable que tratar de escribir tu propio código para la base de datos.

DataMapper está disponible como gem, por lo que se instala así:

```
$ gem install data_mapper
```

Dado que DataMapper es una librería muy grande, también es posible instalarla por partes. Es un concepto llamado «modularidad», y es el sello de la buena programación. A continuación verás el comando que instala todos los gems individuales:

```
$ gem install dm-core dm-aggregates dm-constraints dm-migrations  
dm-transactions dm-serializer dm-timestamps dm-validations dm-types
```

En lugar de toda la librería, instalas solo las partes que tu programa utilizará, lo que es más eficiente.

Usar DataMapper

Ahora que DataMapper está instalado, debo saber cómo usarlo para 1. hablar a la base de datos, y 2. configurar esa base para contener y recuperar la información que quiero.

Con base en la documentación Postgres en Heroku, me entero de que la siguiente orden permitirá que mi aplicación Sinatra hable con la base de datos:

```
Datamapper.setup(:default, ENV['DATABASE_URL'] || "sqlite3://#{Dir.pwd}/database.db")
```

En este caso, el signo `||` es otra forma de decir “o”. `ENV['DATABASE_URL']` es una variable que Heroku usa para designar la base de datos de tu aplicación. Si esa base no está disponible, usará la segunda opción, una base de datos llamada Sqlite^[66].

Sqlite se instala siempre en las computadoras Mac, así que está lista. DataMapper puede hablar tanto a Postgres como a Sqlite si instalo ambas gems:

```
$ gem install dm-sqlite-adapter dm-postgres-adapter
```

Esto significa que mi aplicación usará Postgres cuando corra en Heroku, pero Sqlite cuando corra en mi computadora. En cualquier caso, mi código será el mismo, aunque las bases de datos hablen diferentes lenguajes. Eso está muy bien.

Hablando de correr esta aplicación en mi computadora... ¿cómo lo hago?

¡Pow!

Fui a Stack Overflow y a Hacker News para investigar cómo correr localmente este tipo de aplicaciones. Por fortuna, hay bastantes opciones. Parece que puedo instalar librerías (como Foreman o Shotgun) que correrán la aplicación cuando introduzca el comando en Terminal, o puedo instalar un programa que mantiene corriendo el programa todo el tiempo.

Uno de estos programas es Pow^[67], un servidor Rack de configuración cero para Mac OS X. El sitio promete que será fácil instalar un desarrollo huésped en mi computadora en menos de un minuto. ¡Suenan maravillosos!

Instalar Pow toma unos 10 segundos: se requiere una sola orden en Terminal para bajar e instalar la aplicación. Una vez instalada, ejecutas un comando para ligar tu programa a Pow, y te permitirá correrlo en tu máquina.

Hay un gem de Ruby llamado Powder^[68] que simplifica todavía más este proceso:

```
$ gem install powder
```

Cuando el gem está instalado, corres esto para instalar Pow:

```
$ powder install
```

Luego, vas al directorio raíz de tu aplicación y escribes el siguiente comando:

```
$ powder link
```

Eso es todo. Mi directorio raíz se llama Codex, de modo que mi aplicación corre ahora en mi máquina personal en <http://codex.dev>, y así puedo poner a prueba mi trabajo.

Si hago algún cambio, este comando reinicia el programa:

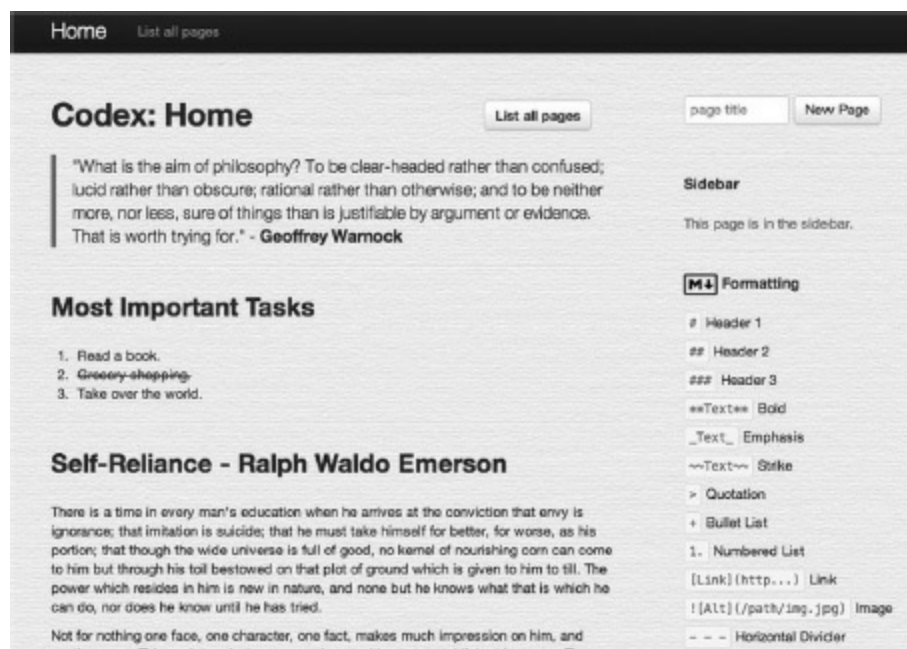
```
$ powder restart
```

Fácil. Ahora estoy listo para programar. Reservaré para ello una hora y media cada noche y seguiré así hasta terminar.

Codificar, probar, revisar

En este punto, describiré lo que estoy haciendo contraponiéndolo a cómo lo hago. Puedes ver el código completo en <https://github.com/first20hours/codex> si quieres seguir esto al detalle.

He aquí cómo luce la aplicación a fin de cuentas:



Seguro adviertes que el diseño tiene tres partes: una barra de navegación en la parte superior, una zona principal de contenido y una barra lateral. Logré este diseño usando Bootstrap^[69], *software* creado por Mark Otto y Jacob Thornton, ambos desarrolladores en Twitter.

En lugar de crear un diseño de cero, Bootstrap contiene una librería con HTML y CSS preescritos y gratuitos. Al usar Bootstrap te ahorras mucho tiempo: puedes realizar un prototipo básico de cómo quieres que luzca una aplicación en minutos en lugar de días.

La unidad básica de esta aplicación es una «página», que despliega un registro guardado en la base de datos. La información de la página va a la zona principal de contenido. Hay un botón que lleva a una pantalla que enlista todas las páginas de la base de datos y que tiene dos botones en la parte inferior. El primero te permite editar la página actual, y el segundo borrarla.

La barra lateral contiene tres características principales. Primero, arriba hay una forma que te permite crear una página nueva, para lo que se requiere un título. Segundo, hay una lista de páginas que el usuario añade a la barra lateral, lo que funciona como sección de consulta rápida. Tercero, hay una referencia de formato, que ayuda a que el usuario recuerde cómo usar los atributos comunes para formatear.

La barra de navegación en la parte superior es muy sencilla. Contiene un vínculo con la página principal, así como un segundo vínculo que va a la pantalla que enlista todas las páginas. Después puedo agregar cosas a la barra de navegación si lo deseo, pero por ahora es todo lo que necesito.

Cada aplicación de la red tiene una página principal, así que debo decidir qué quiero desplegar en esa página. En este caso, solo quiero que se muestre el registro principal de la base de datos.

¿Qué hay en una página? Dado que es un registro de la base de datos, y puesto que los registros tienen campos con información, necesito decirle a DataMapper qué campos alistar. He aquí el código:

```
class Page
  include DataMapper::Resource
  property :id, Serial
  property :title, String
  property :content, Text
  property :lastupdated, DateTime
end
```

Mapper.finalize

Mapper usa este código para crear un nuevo tipo de objeto llamado «página». Ruby puede usar ahora una como si fuera cualquier otro objeto, y yo puedo crear y usar métodos que construyen, modifican y borran páginas. Cuando cambio algo en una página, ese cambio se almacena en la base de datos vía Mapper.

El comando Mapper.finalize indica a la aplicación que instale estos campos en la base de datos si aún no existen.

Ahora que tengo lo necesario para disponer la base de datos, llega la hora de saber a qué rutas ha de servir Sinatra. He aquí mi lista, basada en lo que sé hasta ahora:

```
# Show home page  
get '/'
```

```
# Creates new note from "new page" form  
post '/'
```

```
# Displays requested note  
get('/:url/')
```

```
# Edits requested note  
get('/:url/edit')
```

```
# Saves user edits to note  
post('/:url/edit')
```

```
#Deletes specified note  
delete '/all/'
```

```
# List all pages in database  
get '/all/'
```

```
# Error handling
not_found
error
```

Es una lista bastante completa. Mi aplicación actuará según los comandos creados para cada ruta.

¡Slugs, slugs por todas partes!

¿Recuerdas cuando mencioné que las bases de datos se parecen a una pila de tarjetas de clasificación mágicas, en las que puedes buscar como quieras? Necesitamos una manera de buscar registros específicos de las páginas; por eso ves “:url” en estas rutas. El contenido del parámetro url dirá a la base de datos qué registro recuperar.

Podríamos usar la página de inicio como parámetro, pero eso provoca un problema: a los navegadores de Internet no les gustan espacios, mayúsculas o caracteres especiales (tipo \$ y %) en las direcciones web. Los títulos de las páginas bien pueden incluir esas cosas, así que necesitas quitarlas.

Una cadena de caracteres que identifica una página web se llama *slug*^[70]. El slug de mi página se basará en el título, con las siguientes reglas para hacerlo amigable a los procesos de la red:

1. Todas las letras serán minúsculas.
2. No habrá caracteres especiales: solo se usarán alfanuméricos.
3. No habrá espacios: todos serán remplazados por un guión bajo.

La manera de hacer esto es crear un método que acepte el título de la página, para luego convertirlo a formato slug. Las sustituciones más delicadas se realizan con un tipo de programación llamada *expresiones regulares*, que transforma o encuentra texto en las direcciones o cadenas de caracteres basándose en las reglas establecidas^[71].

Las expresiones regulares pueden ser muy oscuras, pero se usan regularmente, así que procuré encontrar un ejemplo de estas repeticiones. He aquí mi método:


```
# Converts page name into post slug
def slugify(content)
  content.downcase.gsub(/ /, '-').gsub(/[^a-z0-9-]/,
  "").squeeze('-')
end
```

Ahora puedo usar el método slug para transformar una cadena de caracteres como «Page Title» en algo como «page-title», haciéndolo apropiado para utilizarlo como dirección de la red.

Además, si almacenamos el slug junto al título de la página, podemos usarlo para recuperar el registro usando el parámetro :url.

Añadí el siguiente campo a la clase DataMapper:

```
property :slug, String
```

Entonces cada vez que cree una página nueva, puedo «slugificar» su título, guardarlo en la base de datos y usarlo para mostrar la página de nuevo cuando quiera recuperarla. Así sabrá la aplicación qué página debe mostrar.

Crear páginas

Empiezo por trabajar en mi lista de rutas necesarias. La ruta principal es fácil: la redirijo al slug /home/, dado que quiero que la página de inicio muestre el registro de inicio.

La ruta «crear página» está vinculada al pequeño formato en la parte superior de la barra lateral. El usuario introduce el título de página y luego da clic en el botón. El sistema captura la página de título, la slugifica y luego salva el título, el slug y la hora de creación en la base de datos. Luego envía una petición GET conteniendo el slug y desplegando la nueva página.

Hay un detallito muy importante en la ruta «crear página»: ¿Qué pasa si la página ya existe? No quiero sobrescribir la página si ya contiene datos. La pérdida accidental de datos es inaceptable.

Por fortuna, DataMapper resuelve este asunto con el método integrado .first_o_crear. Antes de crear la página, Data Mapper revisará si existe. De

ser así, DataMapper no la sobrescribirá y Sinatra dirigirá el buscador a la página existente. Problema resuelto.

La ruta «mostrar página» lee el slug en el URL, recupera el registro de la base de datos y luego despliega la información en el área principal de contenido. Después añadiré formatos sofisticados, pero ahora solo quiero que esto funcione.

Editar las páginas

Editar una página implica dos rutas distintas. La primera obtiene la página que el usuario desea editar (petición GET) y luego despliega el contenido del registro en un formato que permite editarla.

Para desplegar estas páginas, la aplicación se basa en una plantilla de sintaxis llamada ERB, integrada básicamente por comandos HTML más Ruby. ERB permite a los programadores escribir HTML que incluya elementos que se pueden cambiar. Dado que ERB procesa la página antes de desplegarla ante el usuario, puede cambiar el texto de la página cada vez que esta se carga, con base en las órdenes en Ruby contenidas en la plantilla.

El botón de guardar en la pantalla de edición manda una petición *POST* a la aplicación que actualiza el registro de la página.

Borrar páginas

Se requiere de precaución para borrar páginas: recuerda que la pérdida accidental de datos es inadmisibles. Si vas a borrar una página, debes estar absolutamente seguro de que el usuario efectivamente quiere borrarla.

La forma equivocada de hacer esto sería ligar un botón de «eliminar» directamente a una ruta DELETE en la aplicación. Eso haría que la página se borrara incluso si el usuario lo presionó de modo accidental.

Una mejor estrategia consiste en usar un proceso de dos fases. Hacer clic en el botón de borrado de una página lleva al usuario a una pantalla de confirmación que despliega el título de la página que quiere borrar. Si desea seguir adelante, entonces hace clic en el botón de confirmación, que manda la petición DELETE. Si el usuario hizo clic en el botón de borrado por error,

puede presionar el de cancelación o regresar a la página anterior con el navegador.

Enlistar todas las páginas

«Enlistar todas las páginas» redirige al usuario al slug /all/, que es diferente al resto de las páginas.

En lugar de recuperar un solo registro, DataMapper recupera todos los registros de la base de datos. La plantilla ERB para la página contiene un bucle condicional que crea un elemento para cada página recuperada. Cada elemento contiene el título de la página, desplegado como vínculo que contiene el slug asociado con esa página. Al hacer clic en el vínculo, el usuario es llevado al registro individual de cada página.

Echar a andar la aplicación por vez primera

Todo está en su lugar, pero tengo un problema: cuando pruebo la aplicación al visitar la página de inicio, inmediatamente obtengo un mensaje de error. El programa trata de encontrar la página de inicio en la base de datos, ¡pero esta no existe porque acabo de iniciar la aplicación!

La solución a esto consiste en crear un «proceso administrativo de uso único», mediante un programa llamado Rake. Los programas Rake se guardan en el archivo Rakefile, localizado en el directorio raíz de la aplicación.

Estos archivos funcionan como cualquier otra aplicación normal Ruby, con una excepción: existen fuera de tu programa principal, y debes correr el comando manualmente.

Esto hace que Rake sea muy útil para hacer cosas como añadir información directa a la base de datos antes de correr el programa oficialmente. Copio las partes más importantes de application.rb en Rakefile, y luego creo un comando que haga una nueva página de inicio en la base de datos. Después, solo necesito ejecutar este comando una vez:

```
$ rake setup
```

Rake crea un registro para la página de inicio y mi aplicación deja de mostrar mensajes de error al echarla a andar. Cuando empuje esta aplicación a Heroku, ejecutaré el comando Rake remotamente para instalar la base de datos antes de usar la aplicación.

Llegado este momento, tenemos a todos los elementos principales en su lugar. Ahora es tiempo de añadir cosas divertidas.

Apoyar al usuario en la barra lateral

Me doy cuenta de que no incluí una forma de agregar páginas a la barra lateral como pretendía. Estas características requieren de datos lógicos o booleanos, dado que solo tienen dos valores: la página se despliega en la barra lateral o no.

Agregué esto a la clase `DataMapper`:

```
property :sidebar, Boolean, :default => :no
```

También incluí un espacio pequeño para poner una palomita a la pantalla «editar», junto a «¿Incluir en la barra lateral?», que vinculo al campo de la barra lateral en la base de datos. Luego escribo un bucle simple para encontrar registros en la base de datos en que `:sidebar = true`, y después los despliego como listado, de modo semejante a como lo hice en la página de «Enlistar todo».

Vuelvo a activar la aplicación, edito un registro y la aplicación se congela.

No logro averiguar qué está fallando. Después de combinar la documentación de `DataMapper` con una búsqueda en Stack Overflow, encuentro que las variables booleanas no se llevan muy bien con las formas HTML. Necesito hacer las cosas de otra manera:

```
property :sidebar, Enum[ :yes,:no ], :default => :no
```

Esta es otra manera de hacer básicamente lo mismo. `Enum`, que es abreviatura de «enumerar», crea una lista de opciones, y la forma establece

qué opción guardar en la base de datos.

Agregar apoyo Markdown

Ahora quiero asegurarme de que mis páginas contengan formatos sofisticados, como letras cursivas, negritas y encabezados.

Elegí Markdown como sintaxis de formato. Se trata de un pequeño lenguaje muy popular y útil creado por John Gruber^[72]. Ya estoy familiarizado con el funcionamiento de Markdown, ya que algunas aplicaciones que utilizo en mi computadora lo incluyen. Sin embargo, tengo que ingeniármelas para hacer que mi programa entienda este lenguaje.

Un poco de investigación me indica que hay varios gems Markdown disponibles. Elijo la librería `rdiscount`, y la incluyo en `application.rb`:

```
require 'rdiscount'
```

`Rdiscount` transforma el texto escrito en formato Markdown en HTML: el navegador del usuario despliega entonces ese texto con el formato adecuado. Los archivos Markdown no son especiales en sí mismos: son solo archivos de texto escritos en una determinada manera.

Eso significa que no necesito transformar la información de mi página para Markdown antes de añadirla a la base de datos. Después de todo, es solo texto. Únicamente debo llamar a `rdiscount` cuando quiera desplegar el texto con formato sofisticado.

Este es el comando que hace el trabajo, mismo que agregué a la plantilla ERB responsable de mostrar las páginas:

```
<% markdown(@page.content) %>
```

Este método funciona mal en el campo correspondiente de la página, pero se transforma en HTML y luego despliega el resultado final correcto al usuario. Fácil.

Añadir seguridad

¿Y qué hay de la información para acceder a ella? Si pongo esta aplicación en Heroku sin que se requiera de un nombre de usuario y de una contraseña, cualquiera podrá ver lo que he guardado en la base de datos.

Resulta que los navegadores de red modernos usan un protocolo de seguridad llamado HTTP Basic Authentication^[73], que es una forma sencilla de que el usuario de su nombre y contraseña antes de proceder. Si no proporciona las credenciales adecuadas, se le conduce a una página de error.

Aquí está el código que permite la verificación de identidad básica en Sinatra:

```
use Rack::Auth::Basic, "Restricted Area" do |user-  
  name, password|  
  [username, password] == [ENV['ADMIN_USER'],  
    ENV['ADMIN_PASS']]  
end
```

En este caso, almaceno el nombre de usuario y la contraseña en Heroku como *variables ambientales*, que puedo procesar con un comando de Terminal. ¡Eso me permite usar el mismo código para distintas aplicaciones, así como mostrarlo sin darte mis contraseñas!

Esto es un buen ejemplo de por qué es importante saber que Sinatra está construido sobre Rack. Hay muchas librerías como Rack::Auth::Basic, y puedo usar cualquier de ellas en Sinatra. Mientras menos tenga que reinventar la rueda, mejor.

Solo falta un elemento de seguridad que deseo añadir: el cifrado. Voy a agregar el cifrado SSL —el mismo tipo de seguridad que usan los bancos para asegurar que las sesiones en línea de actividad bancaria sean privadas— a mi aplicación con el gem rack-ssl-enforcer:

```
require 'rack-ssl-enforcer'  
use Rack::SslEnforcer
```

Esta librería obliga a que el navegador use una conexión SSL segura para acceder al sitio. Heroku permite siempre que las aplicaciones usen SSL, así que no hay más que configurar^[74].

Agregar mensajes *Flash*

Hay un último elemento que quisiera agregar: he visto sitios web que te muestran pequeños mensajes como «Tu página ha sido creada/editada/borrada» después de hacer algo. ¿Cómo se logra eso?

Investigo un poco y encuentro una librería llamada *sinatra.flash* que se encarga de estas cosas^[75]. Estos mensajes reciben el nombre de «mensajes *flash*», y funcionan guardando un poco de texto en la *cookie* que se carga en la máquina del usuario antes de ir a la siguiente página. Cuando se carga la siguiente página, la aplicación lee la *cookie* y despliega el mensaje al usuario.

Agregué la librería a Gemfile a `application.rb`, dispuse los mensajes que quería que se mostraran en las rutas apropiadas, luego añadí un poco de código para desplegar esos mensajes en mis plantillas ERB y listo. Mi aplicación está completa.

Código completo

Aquí les muestro el principio de la versión final del archivo `readme` de Codex:

Codex es una sencilla aplicación para la red, para un solo usuario, escrita en Ruby. Codex usa Sinatra y DataMapper para crear, salvar, actualizar y borrar registros de página desde una base de datos simple Postgres. La aplicación está lista para su despliegue inmediato en Heroku.

El formateo Markdown está disponible para todas las páginas, lo cual facilita escribir páginas complejas con un lenguaje de marcado sencillo. La autenticación en HTTP y el SSL obligatorio para todo el tráfico mantienen tu información segura. El framework Bootstrap te permite aplicar estilos que hace que tus páginas luzcan limpias y atractivas.

El readme sigue detallando las instrucciones para su instalación en Heroku. La idea de desarrollar escribiendo primero el archivo readme fue bastante buena.

En total, me tomó 10 horas crear Codex. Eso completa mi inversión total de 20 horas aprendiendo programación. (Me tomó mucho más tiempo escribir este capítulo que programar la aplicación).

Después de crear Codex, asistí a un encuentro local de programadores en Ruby, y me ofrecí para explicar cómo funciona el programa. La retroalimentación fue muy positiva y me felicitaron por lo limpio, compacto y comprensible que era el código. Uno de los participantes destacó que la calidad del código era mejor que la de algunos proyectos escritos por programadores profesionales.

Misión cumplida.

Ira contra la máquina

Quiero aclarar algo: el proceso que describí parece lineal y concreto. Eso se debe a que, hasta ahora, describí lo que funcionó, no lo que falló.

Codificar una aplicación para la red funcional y útil es como armar un rompecabezas que ofrece ciertos retos extra: no sabes qué piezas existen, deberás crear algunas y, si cometes un error, el rompecabezas no se arma.

He aquí cómo fue en realidad mi proceso de programación: tenía una idea de cómo debería trabajar parte del programa. Escribía algo de código, lo probaba y el programa fallaba. Trataba de modificar el código. A veces mis cambios arreglaban el problema, pero no siempre. En otras ocasiones, mis cambios echaban a perder más cosas de las que arreglaban. Si de pronto me sentía atascado, investigaba sobre el mensaje de error recibido o sobre la librería correspondiente en Stack Overflow o en Google.

Cuando todavía estás aprendiendo las funciones de un programa, tu aplicación suele fallar más veces de las que funciona. También se aprende sobre el valor de cosas como llevar el control de las versiones, lo que te permite regresar tu código a una versión funcional previa.

¿Recuerdas que mencioné que no tenía idea de para qué servía un git? Es para esto: si editas archivos y algo falla, quizá no encuentres la falla.

Regresar a una versión anterior que funciona es un regalo de Dios y un alivio. Si no puedes hacerlo, entras en pánico.

Cuando empecé a codificar la característica de agregar páginas a la barra de tareas lateral, mi aplicación falló. Traté de encontrar el error pero no pude. Si tuviera cabello en la cabeza, lo hubiera arrancado de desesperación.

En esos casos se aprende el valor de git branch y de git merge: puedes crear copias experimentales de tu programa y luego hacer modificaciones. Si tu código funciona, puedes integrarlo a la copia maestra. Si la riegas, puedes borrar tu rama experimental sin perder el trabajo previo.

Programar es difícil, y existen millones de formas de equivocarse. La computadora no perdona y no le gusta sufrir por comandos mal hechos. Igualmente, es fácil pasar por alto pequeños detalles que producen resultados inesperados. Yo la estaba pasando muy mal con un error que salvaba un montón de páginas en blanco en la base de datos, mismas que se mostraban en las páginas de mostrar todo lo guardado.

Cada vez que miraba la página, más páginas fantasmas aparecían en la lista y no tenía idea de dónde provenían. Terminé por darme cuenta de que se trataba de un error en la ruta para el comando «list all»: estaba usando un comando erróneo para recuperar los registros de la base de datos.

Cada vez que echaba a perder algo, aprendía algo. Ese es uno de los beneficios ocultos de la programación. La computadora es el bucle que más rápido ofrece una retroalimentación. Si haces algo mal, lo sabes inmediatamente. Si lo haces bien, puedes ver los resultados de tu trabajo sin mayor tardanza. Si resistes el impulso de aventar tu computadora hasta el otro lado del cuarto, esta retroalimentación instantánea puede hacer que programar te resulte bastante adictivo.

Al principio de este experimento, no podía programar en absoluto. Ahora puedo hacerlo. Solo tuve que dedicar el tiempo necesario para someter un código, presionar e investigar un poco cuando me atascaba.

¿Qué obtuve a partir de mis esfuerzos? Una tonelada de cosas. Aprendí qué es la programación, cómo se ve y por qué es útil. Aprendí a crear desde cero aplicaciones reales para la red en Ruby y luego las mandé a los servidores de producción. Aprendí los conceptos básicos de Sinatra, Heroku,

Jekyll, DataMapper, Rake y otras herramientas que puedo utilizar para crear nuevas aplicaciones. Aprendí a encontrar los problemas para solucionarlos.

Un repaso del método

Hagamos un repaso de la esencia del método para aprender programación:

- Pasé algún tiempo aprendiendo qué es la programación y qué son las aplicaciones para la red en general. Luego deconstruí estas habilidades en subdestrezas más pequeñas, fáciles de aprender y poner en práctica.
- Establecí mi meta de desempeño al escoger dos proyectos específicos que quería crear, y después definí cómo serían al terminar.
- Deconstruí esos proyectos en sus partes constitutivas e identifiqué qué partes parecían más importantes.
- Me aseguré de tener las herramientas necesarias para trabajar (como la última versión de Ruby), y también de conseguir cualquier herramienta adicional necesaria.
- Encontré algunas fuentes de información fidedignas sobre programación, pero me salté varios tutoriales con tal de lanzarme de cabeza a escribir programas.
- Trabajé primero en las subdestrezas más importantes, como averiguar la forma de probar los programas en mi computadora, mandar la aplicación terminada a producción, etcétera.
- Usé ejemplos como referencia para empezar y ganar confianza; luego probé varias maneras de programar las características que me interesaban.
- Al cometer un error, el programa se congelaba y me mandaba un mensaje de error, creando una rápida retroalimentación.
- Después de obtener un mensaje de error, experimenté con varias soluciones. Si no lograba arreglar el problema con mis medios, buscaba ayuda.
- Usé el sistema de construir/probar/arreglar hasta que terminé mis programas.

Tiempo total utilizado: aproximadamente 20 horas. Diez fueron para investigación y diez para programar las dos aplicaciones que ahora están terminadas y en producción.

¿Y ahora qué?

He seguido programando aplicaciones para la red desde que terminé estos dos proyectos específicos, concentrándome en programas que facilitan la operación de mi negocio.

Me siento orgulloso al decir que mi negocio entero funciona con *software* que yo mismo hice. Mis aplicaciones pueden hacer cargos a las tarjetas de crédito, manejar suscripciones, mandar correo electrónico y manejar el acceso de los usuarios al sitio. Al aprender a codificar, armé mi ejército de programas para facilitarme la vida.

¿Cuánto me tomó escribir todos estos programas? Unas 90 horas en total, incluyendo las 20 detalladas en este capítulo.

Hay un beneficio adicional: siempre que una actividad de mi negocio me parece repetitiva o frustrante, empiezo a pensar en código. ¿Cómo funcionaría un programa que pudiera resolver esto? Casi siempre hay una manera de sistematizar el proceso de modo que la vida cotidiana sea más sencilla.

También aprendo algunos trucos nuevos, como adaptar mi computadora para que la programación sea más veloz. Aprendo atajos de mi editor de texto que me ahorran tiempo y he actualizado mi programa Terminal a iTerm2 y Z-Shell para hacer que la codificación sea un poco más fácil.

Sigo practicando y de ningún modo soy un experto. Todo lo debo investigar y me lleva bastante tiempo resolver problemas, errores y demás. A veces es frustrante.

Sin embargo, sigo creando programas que resuelven problemas reales de modo práctico y confiable. Es lo que cuenta en realidad.

Lucho con la computadora. Y gano.

6

Mecanografía

Lección: los viejos hábitos no necesariamente se resisten a morir.

Todo lo que vale la pena hacer bien, vale la pena hacerlo mal al principio.

Ray Congdon.

Para obtener imágenes, videos y comentarios adicionales sobre este capítulo, visita <http://first20hours.com/typing>.

Hasta este momento, he aprendido habilidades nuevas en áreas en las que tengo poca experiencia. En principio, esto constituye un obstáculo pero, en lo que a mí respecta, mi mente no *interfiere* activamente con el proceso de aprendizaje.

¿Qué pasa cuando practicas algo nuevo y tu cerebro se rebela?

Ciertamente, existen ejemplos dramáticos de lo que sucede al aprender una actividad. Tiger Woods es famoso por haber cambiado hasta en tres ocasiones su entrenamiento para hacer su famoso *swing* de alto

rendimiento^[76]. A veces vale la pena aprender a hacer mejor algo importante, incluso si el costo a corto plazo consiste en ineficiencia y frustración.

El ritmo de la nueva tecnología hace que el reaprendizaje sea muy común: ¿qué pasa si cambia un programa que necesitas para hacer tu trabajo? ¿Y qué tal si un nuevo programa se convierte en la forma más efectiva para completar una labor? ¿Qué tal si entras a un nuevo trabajo que requiere nuevas herramientas? Reaprender destrezas rápidamente suele ser tan importante como aprender algo nuevo.

Por curiosidad: ¿qué se siente volver a entrenarte para hacer algo que en verdad importa?

Empecé a hacer una lista de cosas que ya sé hacer y luego la peiné buscando habilidades que 1. ya pudiera hacer bien y, 2. tuvieran distintos métodos para lograr el mismo resultado. No me tomó mucho tiempo encontrar la actividad indicada.

La mecanografía.

Una vida detrás del teclado

Según David Allen, autor del *best seller* de productividad *Getting Things Done* (2002), si tu trabajo requiere de una computadora, aprender a escribir sin mirar el teclado es lo más importante para mejorar tu productividad.

El nivel de velocidad crítico parece ser de 60 palabras por minuto (PPM). Si no puedes teclear al menos 60 PPM con un bajo nivel de errores, afectas seriamente tu habilidad para trabajar productivamente. Mientras menos esfuerzo desarrolles para escribir con el teclado, más tiempo y energía tendrás para tareas mayores.

He mecanografiado durante unos 17 años (probablemente más). En la escuela, se exigía que todo estudiante de séptimo grado tomara un curso de mecanografía, y recuerdo que me aburrí mucho. Ya sabía mecanografía, así que terminaba las tareas rápidamente y pasaba el resto de la clase preguntándome cómo lograr que el procesador de palabras produjera caracteres de apariencia extraña (¶ o §, por ejemplo).

Aprendí a escribir en el teclado al pasar mucho de mi tiempo libre usando computadoras. Mi práctica era ambiental, no deliberada: no trabajaba

conscientemente en mejorar mi velocidad o precisión. Solo usaba la computadora y, en el proceso, aprendí a teclear.

Eso no quiere decir que mi técnica fuera grandiosa. Mis manos se movían por todo el teclado en lugar de pasar la mayor parte del tiempo cerca de las teclas centrales. Mi método no era correcto si se toman en cuenta los libros, pero me servía para hacer las cosas, que era lo que me importaba.

Sin embargo, cada determinado tiempo me encontraba con algún artículo sobre diversas disposiciones del teclado: arreglos de teclas que son diferentes a la tradicional disposición QWERTY, la llamada disposición universal, que es la base de la mayoría de los teclados de habla inglesa^[77].



Se dice que en realidad el teclado QWERTY es muy ineficiente y que tiene un horrible diseño. Existen otras maneras de disponer las teclas para que el usuario teclee más rápido, con mayor precisión y menos esfuerzo.

El esfuerzo acumulativo es muy importante: las lesiones por tensión repetitiva y el síndrome del túnel carpiano son enfermedades de las manos y las muñecas causadas por escribir en un teclado. Aunque todavía no tengo síntomas, varios amigos cercanos ya las padecen. No quiero enfrentar estos males si puedo evitarlo.

Planeo escribir y programar durante los años venideros, de modo que, a menos de que el reconocimiento de voz o la transmisión del pensamiento a la computadora lleguen a ser métodos primarios, seguiré mecanografiando en el futuro cercano. Probablemente me convenga de sobremanera aprender a mecanografiar de la mejor manera posible, incluso si eso implica padecer confusión o incomodidad a corto plazo.

Adiós, QWERTY: volveré a aprender a escribir con el teclado.

Cómo la disposición QWERTY del teclado se convirtió en estándar universal

Contra la creencia popular, el formato de teclado QWERTY no fue diseñado para que la mecanografía fuera más lenta: más bien era una solución a un problema de ingeniería mecánica.

En los viejos tiempos, antes de que aparecieran los procesadores de palabras y las computadoras, las máquinas de escribir hacían que los caracteres aparecieran en las páginas apretando una tecla metálica pequeña que levantaba una barra metálica con la letra (tipo) y la impulsaba contra el papel, bien apretado alrededor de un rodillo. Una cinta entintada se interponía entre la tecla y el papel. La barra de metal con la letra golpeaba la cinta, que hacía contacto con el papel dejando una letra en la hoja. La tecla de retorno hacía girar el rodillo, moviendo el papel hacia arriba y permitiendo que el mecanógrafo continuara escribiendo en una línea nueva.

C. L. Sholes, a quien se atribuye el invento de la disposición QWERTY del teclado, construyó su primer prototipo en 1868. No fue la primera persona en crear una máquina de escribir: antes hubo 51 inventores que lo intentaron. Sholes estudió su trabajo e incorporó muchas de sus aportaciones a su diseño^[78].

Las teclas del primer prototipo de Sholes fueron dispuestas en orden alfabético, lo que tenía sentido. En esa época, nadie imaginaba que la gente querría o podría escribir con los diez dedos. Al disponer las teclas alfabéticamente se aseguraban de que los usuarios no entrenados encontrarán las letras apropiadas con los dedos índice.

Sin embargo, el prototipo presentaba un problema importante: las barras metálicas que golpeaban la cinta tenían el horroroso hábito de atascarse cuando se presionaban letras adyacentes en rápida sucesión. Piensa en las teclas que más se usan al escribir en inglés: las vocales AEIOU y las consonantes DHTNS. En un teclado mecánico dispuesto alfabéticamente, la S y la T están juntas. Si un tipo subía mientras el otro bajaba, se atoraban, haciendo que el mecanógrafo batallara para destrabar los tipos manualmente.

Para corregir esta molestia crónica, Sholes buscó la ayuda del maestro Amos Densmore, que realizó un estudio improvisado sobre la frecuencia con

que aparecían las letras en la lengua inglesa. Eso lo aprovechó Sholes para disponer las combinaciones más comunes, como TH, en lados opuestos del banco de tipos, evitando así las colisiones.

Esta estrategia no resolvió por completo el problema, pero mejoró tanto las cosas que Sholes solicitó la patente del diseño en 1872^[79]. E. Remington & Sons, empresa entonces conocida por fabricar armas de fuego, compró la patente de Sholes en 1873.

Después de realizar mejoras mecánicas, como añadir la tecla Shift para permitir que los mecanógrafos alternaran entre letras mayúsculas y minúsculas, Remington produjo máquinas de escribir en masa, en 1874, todas con la disposición QWERTY en el teclado, para venderlas a las empresas.

Remington no era la única que vendía máquinas de escribir. Otras empresas como Hammond y Blickensderfer ofrecían aparatos semejantes y cada marca tenía su propia disposición de teclado.

Entonces los negocios dependían básicamente de los memorándums escritos para registros y correspondencia. Las máquinas de escribir ahorraban mucho esfuerzo manual, pero solo si los operarios sabían usarlas. Para cerrar una venta, las empresas debían superar la necesidad de entrenar a los mecanógrafos para usar el extraño artilugio.

Eso llevó a una dinámica de mercado interesante: las empresas que fabricaban máquinas de escribir reclutaban y entrenaban empleados, operando como eficientes agencias de colocaciones. Si un hombre de negocios quería contratar a alguien que supiera escribir a máquina, llamaba a Remington, que vendía al negocio tanto la máquina como los servicios de quien pudiera usarla.

Con el paso del tiempo, conforme más y más negocios adoptaban las máquinas de escribir, el teclado QWERTY se hizo estándar. No hubo un momento definido, no hubo ley o comité de estándares burocráticos que ordenara esta adopción. Fue un movimiento de mercado que adoptó una solución suficientemente buena.

Los negocios necesitaban máquinas de escribir y mecanógrafos. Remington proveía ambos elementos. Además, cuando un negocio necesitaba una nueva máquina u otro mecanógrafo, era más práctico comprar una máquina QWERTY y contratar un operario que supiera usar el teclado

QWERTY. Pasados 60 años, el teclado QWERTY se convirtió en estándar y los competidores desaparecieron. Al paso de las décadas, QWERTY se adueñó del mercado.

Aparece la competencia: Dvorak

En 1932, August Dvorak, profesor de la Universidad de Washington, recibió un estímulo de 130 000 dólares, por la Comisión Carnegie para la Educación, para investigar el diseño del teclado. Uno de los factores que motivaron esta investigación fue darse cuenta de que los teclados QWERTY habían sido diseñados para resolver un problema de ingeniería mecánica que ya no existía. ¿Había una mejor manera de diseñar un teclado?

Cuatro años después, en 1936, Dvorak solicitó una patente para el Teclado Dvorak Simplificado, que parecía muy superior a otras disposiciones de teclas. La base de tal afirmación era sencilla: la disposición Dvorak colocaba los caracteres de uso más común directamente bajo los dedos del usuario, en la fila central o de inicio.



Las consonantes más comunes estaban colocadas en el lado derecho de la fila central, y las cinco vocales al lado izquierdo. Esta división de vocales y consonantes equilibraba la carga de tecleo entre las manos del mecanógrafo, reduciendo teóricamente la fatiga y mejorando la velocidad.

Dvorak también decía que su disposición de teclado era más fácil de aprender y se emprendieron estudios en organizaciones que entrenaban un número significativo de mecanógrafos, principalmente en el rubro militar. Los resultados fueron poco concluyentes: dado que el mismo doctor Dvorak condujo dichos estudios, muchos resultados favorables fueron cuestionados, pues había mucho dinero en juego.

Un estudio independiente, realizado por la Administración de Servicios Generales, responsable de entrenar a los mecanógrafos del gobierno estadounidense, encontró que los mecanógrafos QWERTY requerían más de 100 horas de reentrenamiento para llegar a la velocidad de escritura en Dvorak. Era demasiado tiempo. Por lo tanto, el estudio recomendó estandarizar a los mecanógrafos del gobierno en la práctica de QWERTY. Los fabricantes de máquinas de escribir y los negocios se alinearon con la decisión.

Como disposición, Dvorak siguió existiendo marginalmente, pero fue un fracaso como estándar. Aunque tenía varias pretensiones válidas comparado con QWERTY, no pudo remplazar al uso habitual. Dvorak languideció durante décadas en un muy distante segundo lugar, después de QWERTY.

Aparece un nuevo aspirante: Colemak

QWERTY y Dvorak no son los únicos teclados disponibles. Aunque la gran mayoría de los mecanógrafos aprenden QWERTY, los inventores y los aficionados han procurado innovaciones durante décadas. La mayoría de esas disposiciones nunca destacaron, pero en 2006 un nuevo teclado comenzó a llamar la atención de los mecanógrafos aventureros, gracias en buena medida a Internet.

La historia es la siguiente: Shai Coleman, un programador, decidió probar suerte con un nuevo teclado. Su meta era sencilla: crear una disposición que tuviera la relativa eficiencia de Dvorak pero más fácil de aprender.

Uno de los mayores problemas de Dvorak era que cambiaba todo. Cada letra está en un lugar diferente en Dvorak comparado con QWERTY. Si originalmente aprendiste QWERTY, la magnitud del cambio hacía que Dvorak fuera una pesadilla.

Peor aún, ahora que las computadoras son la principal herramienta para la captura de texto, es común depender de atajos para realizar tareas repetitivas, como salvar un archivo, cortar y pegar un texto, etcétera. Si estás acostumbrado a usar determinadas combinaciones de teclas, Dvorak es particularmente odioso: no solo cambian todas las letras, sino también los atajos.

La estrategia de Coleman consistió en combinar el análisis computacional de una enorme cantidad de documentos, con la idea de conservar todas las teclas posibles en la posición original para que los atajos cambiaran menos. El lado izquierdo del teclado, así como la mayor parte de las teclas de la fila inferior, quedaron casi como en QWERTY, solo cambiaron las teclas más significativas. El algoritmo de Coleman recomendaba la mayoría de los cambios, dejando al usuario la ubicación de algunas teclas ambiguas.

Al final, la nueva disposición de Coleman (a la que llamó Colemak) cambió 17 teclas de la disposición QWERTY. Con base en los números, el teclado era mucho más eficiente que QWERTY y ligeramente mejor que Dvorak. Lo más prometedor era que, debido a que Colemak cambiaba pocas teclas, resultaba más fácil que los usuarios QWERTY aprendieran a usarlo.

Coleman creó un sitio web (colemak.com) con detalles de la nueva disposición, así como instrucciones para instalarla y aprenderla. Comparado con los caros esfuerzos que hizo Dvorak para popularizar su teclado en encuentros personales con grandes empresas, Internet hizo barato lo caro al difundir la existencia de Colemak con un costo muy reducido.

Como resultado, Colemak es ahora la tercera disposición de teclado más popular en inglés, después de QWERTY y Dvorak. La disposición está ya integrada en los nuevos sistemas operativos y tiene miles de usuarios en todo el mundo. No está mal para una disposición que tiene menos de una década y que fue desarrollada por un solo aficionado.

Cuando dudes, prueba

Con todas estas disposiciones de teclado en los alrededores, ¿cómo decidir cuál es mejor?

Es sencillo: pruébalos.

La tecnología moderna en programación ha hecho mucho más fácil reunir información sobre varias maneras de ver la distribución del teclado. En vez de probar y reunir datos manualmente, es posible usar un programa que analiza las disposiciones de teclado para ti. Uno de estos programas se llama *carpalx*^[80], creado por Martin Krzywinski, programador del Centro de Ciencias Genómicas de Columbia Británica.

Carpalx está diseñado para «realizar una simulación para encontrar una

disposición de teclado que minimice el grado de esfuerzo de acuerdo con parámetros modelo». En otras palabras, carpalx puede probar automáticamente la eficiencia de varias disposiciones de teclado al correrlas contra textos de prueba.

Kyrzwinski usó carpalx para diseñar su propia disposición, pero también para comparar Colemak con QWERTY y Dvorak, produciendo una gran cantidad de datos comparativos. Aquí les muestro lo que encontró^[81]:

QWERTY es el gran perdedor, con un enorme incremento en el esfuerzo base (+193 %), un gran aumento en la ruta de tecleo (+36 %) y un aumento significativo de las penalizaciones (+16 %). Dvorak ya mejora a QWERTY, así que la diferencia entre este y Colemak es menor.

Colemak usa más la fila central (74 %) si se compara con Dvorak (71 %). Esto deja muy atrás al 34 por ciento de QWERTY. El uso de la fila inferior es bajo en Colemak e igual al de Dvorak, con 9 por ciento. Colemak es más equilibrado en cuanto al uso de cada mano, con 6 por ciento de preferencia para la mano derecha (Dvorak presenta 14 por ciento para la derecha y QWERTY 10 por ciento de preferencia por la mano izquierda).

Colemak es muy bueno para mantener la alternancia de las manos. Tanto Colemak como Dvorak hacen mayor uso del dedo meñique. Colemak usa el meñique en 16 por ciento de las situaciones, en tanto que Dvorak lo hace en 18 por ciento y QWERTY en 10 por ciento. Colemak hace las cosas bien cuando se trata de usar los dedos más fuertes, índice y medio, pues lo hace 67 por ciento del tiempo. En eso es mejor que Dvorak, que usa estos dedos 60 por ciento del tiempo, pero no tan bueno como QWERTY, que los usa 69 por ciento.

Con base en los datos de Kyrzwinski, se requiere casi el doble de esfuerzo físico para teclear en QWERTY, si se le compara con Colemak. Este también vence por poco margen a Dvorak, lo cual es impresionante, puesto que Colemak cambia solamente 17 teclas de la disposición QWERTY, contra las 24 que cambia Dvorak.

De acuerdo con los datos, pienso aprender a usar el teclado Colemak. Me gusta que reduzca el esfuerzo sin reacomodar por completo los atajos comunes del teclado (los uso bastante).

¿Qué hago a continuación?

¿Cómo se ve Colemak?

Te presento una ilustración que te permitirá saber cómo se ve un teclado Colemak^[82]:



Además de cambiar la posición de las teclas con los caracteres más utilizados, Colemak convierte la tecla que fija las mayúsculas (Caps Lock) en una segunda tecla de borrado. Este cambio es una de las más grandes innovaciones de Colemak. Nadie (a no ser por algunos usuarios maleducados de los foros) usa la tecla Caps Lock regularmente, pero esta ocupa un lugar principalísimo: está al alcance del dedo meñique de la mano izquierda, junto a la tecla con la letra A, en la fila central.

Al cambiar la Caps Lock por una tecla de borrado (Delete), los usuarios de Colemak pueden corregir errores sin quitar la mano de la fila central para presionar la tecla de borrado —que en el QWERTY está en la parte superior derecha del teclado—, con lo que se ahorra un esfuerzo considerable. Ese cambio reduce de 15 a 20 por ciento la distancia entre dedo y tecla, comparado con QWERTY. Mientras más errores corrijas con el meñique izquierdo, más eficiente se torna Colemak.

¿Cómo entro en la modalidad Colemak?

Es fácil cambiar la disposición de teclado en las computadoras: solo debes decir a la máquina que use un archivo de disposición diferente, y esto se logra de la misma manera en que cambias el lenguaje de tu teclado. Esta configuración suele encontrarse en el panel primario de la computadora, el de «sistemas y preferencias». Colemak está incluido en el sistema operativo Mac OS X desde la versión 10.5 en adelante, por lo que no tendrías que instalar nada^[83].

Además, algunos teclados como el TypeMatrix 2030^[84], aceptan Colemak en el teclado mismo. En lugar de configurar la computadora para reconocer Colemak, el teclado traduce los teclazos Colemak en QWERTY, así que los caracteres correctos se muestran en la pantalla. Como resultado, puedes teclear en Colemak sin cambiar nada en la computadora, siempre y cuando estés dispuesto a gastar unos 100 dólares en un buen teclado ergonómico^[85].

Cambiar la configuración de la computadora es la parte sencilla, pero no se trata de la única barrera. Dado que la mayoría de los teclados en inglés vienen con QWERTY de fábrica, cambiar a Colemak implica que las letras impresas en las teclas no coincidirán con los caracteres que aparecerán en la pantalla. Si sueles teclear raras cadenas de caracteres, como contraseñas complejas, las cosas pueden ponerse confusas. ¿Cómo consigues un teclado físico Colemak?

Hay dos maneras de hacerlo: puedes comprar un teclado en blanco o convertir un teclado QWERTY en uno de formato Colemak.

Algunos teclados, como el TypeMatrix 2030, están disponibles en versiones Colemak, lo cual proporciona una solución sencilla si estás dispuesto a comprar un teclado nuevo. Eso no me sirve, pues tecleo en una laptop, por lo que el teclado está incorporado.

Modificar mi máquina

Por fortuna, es fácil sacar las teclas de los teclados Apple para reordenarlas sin dañar la computadora. Dado que Apple usa teclas planas con las mismas dimensiones, convertir el teclado a Colemak fue un proyecto de cinco minutos^[86].



Levanté ligeramente el costado superior derecho de cada tecla, inserté con cuidado la punta de un desarmador pequeño y lo moví por el costado derecho de la tecla. Cuando el desarmador llega a la mitad, la tecla se suelta gracias a un mecanismo de tijera que la regresa a su lugar después de presionarla.

Cuando entiendes cómo hacerlo, es muy fácil quitar las teclas que necesitan ser cambiadas. Luego usé un diagrama de la disposición Colemak para volver a colocar las teclas en el lugar correcto. Con una suave presión, las teclas producen un sonido de clic, quedando en su lugar.

Es un procedimiento simple, pero vale la pena señalar que esto podría anular la garantía de mi laptop. Por seguridad, hice primero la prueba con un teclado Apple extra y luego cambié las teclas de mi MacBook Air cuando me sentí cómodo con el proceso.

Ahora tengo un teclado Colemak en mi laptop. ¿Bien, no?

Es un ejemplo de cómo invertir un poco de tiempo cambiando el ambiente para alentar la práctica. Ahora que mi teclado es Colemak, será más fácil cambiar. Si tengo problemas para saber cuál tecla es cuál, puedo mirar si es necesario.

¿Qué tan rápido puedo teclear?

Ahora que está listo mi teclado, estoy cerca de realizar el gran cambio. Sin embargo, antes quiero tener una idea de qué tan rápido tecleo, lo que me ayudará a determinar mi meta de desempeño.

En este caso, mi meta es sencilla: quiero teclear en Colemak tan rápido como lo hago en QWERTY. No quiero superar mi velocidad, pues no ha sido un impedimento en mi trabajo. Solo quiero teclear tan bien como lo hago ahora pero con menos esfuerzo.

Lo único que necesito saber es mi velocidad actual de escritura, así que busqué en la red un programa para medirla^[87].

La prueba es bastante sencilla: cuando oprimes el botón de inicio, el programa te muestra cien palabras al azar tomadas de un viejo libro^[88]. Tu trabajo consiste en teclear el texto de muestra tan rápido como puedas y con la menor cantidad de errores. Al terminar, presionas el botón «*stop*» y el programa te da tu velocidad de escritura y la tasa de error.

Mi plan consiste en hacer la prueba primero con QWERTY y luego repetirla con Colemak, para saber dónde comienzo. Presiono el botón de inicio y escribo.

Al terminar, presioné el botón *stop* y recibí mis resultados: 61 palabras por minuto, con 100 por ciento de precisión, cero errores. No está mal: tecleo a la velocidad mínima que David Allen recomienda. No soy un demonio de la velocidad, pero puedo escribir lo suficientemente bien para realizar el trabajo.

Tengo todo lo necesario: sé cuán rápido puedo escribir, mi teclado está en la modalidad Colemak y sé cómo activar el teclado en el sistema operativo. No hay nada más que preparar.

Bien. ¿Estoy listo para dejar atrás a QWERTY?

Cambiar el interruptor

Ahora o nunca. Cambio mi computadora al modo Colemak y luego cierro el panel de control. De ahora en adelante, no podré teclear nada en QWERTY hasta completar el experimento.

Mi navegador aún despliega la prueba de velocidad. Echo a andar el cronómetro con el *mouse* y comienzo a teclear.

Aquí está mi primera impresión: #%&@.

Corro el riesgo de sonar melodramático, *pero siento como si me hubieran quitado una parte del cerebro.*

Estoy habituado a que las palabras fluyan sin esfuerzo desde mi cabeza hasta la computadora. Ahora, no sé siquiera dónde están las teclas. Tengo que buscar cada una, incluso las teclas que no cambiaron de lugar, lo que no tiene sentido.

Veo algo en la pantalla y mis dedos se mueven inconscientemente, de modo que escribo cosas absurdas que debo borrar. Me toma varios segundos escribir palabras sencillas y me preocupa desgastar la tecla de borrado.

Cada palabra es una nueva lucha. Miro el reloj y descubro que me ha tomado varios minutos escribir dos oraciones. No voy ni siquiera a la mitad. En verdad estoy considerando la posibilidad de dejar el proyecto, pero sigo adelante. Con el paso del tiempo estuve a punto de claudicar al menos 10 veces.

Al final, me tomó casi 20 minutos teclear 100 palabras. Mi nueva velocidad: cinco palabras por minuto.

Mátenme.

Escribo para ganarme la vida y apenas aprendí a programar. Ahora no puedo hacer ninguna de las dos cosas. ¿Cómo responderé los correos electrónicos? ¿Cómo trabajaré? ¿Qué hice?

He visto al enemigo: soy yo

Es la barrera primaria: que la gente aprenda a manejar una nueva disposición del teclado. Los aspectos técnicos del cambio son triviales si se les compara con los emocionales.

Cuando estás habituado a cierto nivel de velocidad o te sientes cómodo al realizar una tarea, cualquier otra cosa resulta terrible. Esto es particularmente cierto en la mecanografía: si estás acostumbrado a que la escritura en el teclado no conlleve esfuerzo y de pronto esta te obliga a esforzarte mucho, puedes sentirte verdaderamente mal. Lo peor es saber que si vuelves a hacer las cosas como siempre, todo mejorará de inmediato.

La mente no es de mucha ayuda en este caso: nuestros cerebros tienen una terca tendencia a asumir que lo experimentado en un momento seguirá así en el futuro. Ahora mismo, mi mente entra en pánico. Si solo puedo

teclear cinco palabras por minuto, ¡ni siquiera podré trabajar! ¡Mi carrera estará terminada! ¡Mi familia se morirá de hambre!

Por supuesto, eso no es verdad, pero se siente como si lo fuera en ese momento. La experiencia emocional es el mayor obstáculo para el aprendizaje.

Apago la computadora, busco cuaderno y pluma. Necesito un plan para alcanzar una velocidad de escritura funcional en Colemak, y lo necesito *ya*.

Reestructurar mi cerebro

El principal problema es que no sé dónde están las teclas. Claro que imprimí una imagen del teclado como referencia, y las teclas físicas están en la disposición Colemak, pero mi cerebro no es capaz por ahora de canalizar el deseo de escribir una letra y ligar este deseo con el movimiento de un dedo específico.

Lo primero es ayudar a mi cerebro a hacer un mapa de la nueva disposición para convertirlo en habilidades motoras. Necesito ser funcional tan rápido como sea posible.

Afortunadamente, la gente lleva décadas aprendiendo mecanografía, así que hay herramientas bastante buenas que pueden ayudarme. Los tutoriales para escribir en el teclado, como Mavis Beacon^[89] y Typing Trainer^[90], han estado disponibles desde hace mucho tiempo y no cuestan gran cosa.

Desafortunadamente, estos programas asumen que quieres aprender a teclear en QWERTY, ya que la mayor parte de los clientes quieren aprender utilizando el teclado convencional. Estos programas suelen empezar por las teclas de la fila central. La fila central de QWERTY es diferente a la de Colemak, así que los tutoriales para QWERTY no me ayudarán.

Dvorak ha estado presente el tiempo suficiente para que los programas lo incluyan, pero Colemak es relativamente nuevo. Quizá sea buena idea usar algún tipo de tutorial para mecanografía, pero necesito uno capaz de vérselas con Colemak.

Por suerte, tengo una pista. Cuando vagaba por Hacker News durante la investigación para aprender a programar, encontré una publicación relativa a Keyzen^[91], un entrenador para mecanografía con código abierto creado por

Rye Terrell, que desarrolló Keyzen para ayudar a los programadores a teclear más rápido. El programa corre en los navegadores de Internet estándar e incluye signos de puntuación poco comunes como paréntesis, corchetes y diagonales: caracteres que los programadores usan frecuentemente pero que no reciben suficiente atención cuando se trata de los programas normales. Terrell publicó el código entero de Keyzen en GitHub, e invitó a que otros programadores lo usaran o modificaran.

El programa mismo es bastante simple: despliega un grupo de siete caracteres, comenzando por las letras de la fila central. Tu labor es teclear los caracteres en el mismo orden.

Al teclear, el programa emite sonido. Una letra correcta emite uno como el de las teclas normales, pero cambia cuando se comete un error y la letra se pone roja.

Cuando terminas una serie, el programa te pone una nueva. Si presionas la tecla incorrecta, incluirá dicha tecla en las series siguientes. Si tecleas todos los caracteres correctamente por tres series seguidas, escucharás un agradable «¡ding!», que significa que has mejorado. Keyzen introduce entonces un nuevo carácter y así continúa el entrenamiento.

Modificar Keyzen

Keyzen, al igual que la mayoría de los entrenadores mecanográficos, se basa en QWERTY. Dado que el programa es de código abierto, puedo modificarlo para aprender Colemak.

Mi práctica en programación está a punto de resultarme bastante útil.

Hice una copia del código fuente de Keyzen y abrí el programa en mi computadora. El programa mismo es sencillo, así que fue fácil encontrar la sección que controlaba qué caracteres eran desplegados, y también fue fácil reordenarlos como me convenía.

El programa original presentaba primero letras de la fila central del teclado QWERTY, comenzando por los dedos índices para agregar más caracteres hasta que la fila central se agotaba. Luego seguía la fila superior, pasando del centro del teclado a la periferia y luego a la fila inferior, siguiendo el mismo patrón.

Usando como modelo el patrón de entrenamiento de Keyzen, edité el programa para utilizarlo con la disposición de teclado Colemak. El patrón de las teclas físicas era el mismo, pero la secuencia enseñaba ahora Colemak en lugar de QWERTY.

Cuando cerré el programa, tenía mi propio tutorial de mecanografía Colemak. ¡Éxito!

Si te interesa, puedes probar el programa en <http://first20hours.com/keyzen-colemak>.

Habilidades motoras finas

Tengo mi primer tutorial de mecanografía y mi primer objetivo es simple: aprender dónde está cada letra en el teclado.

Corro Keyzen y comienzo a teclear (_ representa un espacio):

```
nn_nnn_n
n_nnn_n_
nnn_nnn_
tntt_tn
t_tntt_
ttttnt
```

Este tipo de práctica no será muy elegante, pero sí es necesaria. Al insistir en la localización de cada letra casi al azar, ayudo a que mi cerebro traduzca lo visto (o pensado) en un carácter con un movimiento motor fino de mis dedos.

Además, Keyzen me ayuda a aprender eficientemente por muchas razones. Primero, Colemak está diseñado para ubicar las teclas de uso más común en la fila central, bajo los dedos más fuertes. Keyzen introduce esos caracteres primero, así que aprendo ARSTDHNEIO antes de llegar a QWZX YM.

Cuando cometo un error, Keyzen reintroduce el carácter en la secuencia de práctica. Como resultado, paso la mayor parte del tiempo practicando las letras que más problemas me dan. Es una forma de aprendizaje espaciado, que se combina con la retroalimentación instantánea, haciendo que la practica sea muy eficiente.

Defino un horario de práctica: al menos dos sesiones de 20 minutos, con un corto periodo de descanso entre ellas, al final del día, justo antes de irme a la cama.

Mi locura tiene un método: para adquirir cualquier tipo de habilidad motora se requiere cambiar la estructura física de tu cerebro y resulta que el sueño es fundamental en ese proceso.

¡Aprende mientras duermes!

Durante las pasadas cuatro décadas, la psicología ha investigado muy activamente la adquisición de habilidades motoras. Si acudes a cualquier biblioteca bien dotada, podrás encontrar varios libros sobre la adquisición de habilidades en general, y de las motoras en particular.

Acabo de visitar mi biblioteca. Revisé la sección de psicología cognitiva de la biblioteca de la Universidad Estatal de Colorado buscando información útil sobre la adquisición de destrezas. No tuve ningún problema en la localización de los materiales. Desafortunadamente, la mayoría de estas investigaciones me resultan casi incomprensibles debido a la jerga académica.

Por suerte, no todo es incomprensible. He aquí un extracto de un estudio que me llamó la atención:

Etapas del aprendizaje de habilidades motoras (2005)^[92].

El aprendizaje exitoso de las habilidades motoras requiere entrenamiento repetitivo... Este artículo aborda la creciente evidencia de que los avances en el aprendizaje de las habilidades motoras se dan en etapas, y en cada etapa predominan mecanismos de almacenamiento distintos. La fase de adquisición se caracteriza por el aprendizaje rápido (en la misma sesión) y el lento (en más sesiones). Durante un corto periodo que sigue a las sesiones de entrenamiento inicial, la habilidad tiende a sufrir interferencias de otras habilidades y la inhibición en la síntesis proteínica, indicando que tienen lugar procesos de consolidación durante los periodos de descanso, entre sesiones.

Durante el entrenamiento y también durante los periodos de descanso, la activación de diferentes regiones cerebrales cambia dinámicamente.

«Interferencia» y «consolidación» son las palabras clave en este caso. Al practicar una destreza, tu cerebro está sumamente activo, pues trabaja en hallar patrones y almacenarlos en la memoria. En el caso de las habilidades motoras, esos patrones implican asociar lo que ves, oyes, sientes y tocas con la activación de las neuronas que controlan los músculos de tu cuerpo. Mientras más fuertes sean estas conexiones neuronales, mejor será tu desempeño.

Durante la práctica, tu cerebro está ocupado haciendo estas conexiones y asociaciones, pero eso no significa que estas se almacenan instantáneamente en la estructura neuronal. Toma tiempo para que estos patrones se fijen, lo que sucede durante el proceso que conocemos como consolidación. La consolidación sucede todo el tiempo, pero es particularmente efectiva cuando duermes.

Aquí te muestro el primer estudio que hallé que establece un nexo directo entre el sueño y la adquisición de destrezas:

La práctica y el sueño, logran la perfección: Implicaciones del aprendizaje dependiente del sueño y la plasticidad para la mejora de habilidades (2005)^[93].

Suele creerse que solo de la práctica depende la mejoría. Aunque realizar una labor nueva repetidamente suele resultar en aprendizaje, lo que ha llevado al conocido adagio de «la práctica hace al maestro», un conjunto de estudios realizados durante la pasada década ha comenzado a cambiar este concepto. Los reportes sugieren que después del entrenamiento inicial, el cerebro humano sigue aprendiendo aunque ya no practique, y que esta mejoría diferida se desarrolla durante el sueño.

He aquí la conclusión de ese estudio:

Aunque las funciones del cerebro dormido siguen siendo inciertas, muchos estudios apoyan ahora el papel del sueño en la modificación y

el mejoramiento de la memoria. Estos reportes dan una abundante evidencia convergente de que los mecanismos de la plasticidad neural dependientes del sueño llevan a la consolidación de la memoria y a la consecuente mejora diferida del desempeño. Diferentes formas de memoria simple y compleja parecen requerir de tipos sutilmente distintos de sueño para mejorar la memoria pasada la noche. Muchos estudios indican que el sueño acaecido dentro de las 24 horas transcurridas tras el fin de la práctica es esencial para que tenga lugar la consolidación.

La adquisición efectiva de habilidades, particularmente de las motoras, parece requerir del sueño, que juega un papel importante en la consolidación de la destreza en la memoria de largo plazo. Investigaciones recientes sugieren que, para un mayor efecto, es mejor dormir dentro de las cuatro horas siguientes a la práctica de la habilidad: incluso una corta siesta es mejor que nada. Si pasa más tiempo, disminuye la habilidad de tu cerebro para consolidar la información obtenida en la práctica.

Por eso practico la mecanografía antes de ir a la cama. Si me duermo pasada una hora desde la práctica, ayudo a que mi cerebro consolide los movimientos motores con mayor efectividad.

Lo raro es que puedo *ver cómo pasa esto*. Mi primera sesión de práctica fue horrible: no logré hacer nada bien, cometí errores constantemente y apenas pasé de los caracteres ubicados en la fila central. Después de dormir una noche entera, al sentarme frente a la computadora me di cuenta de que estaba cometiendo menos errores. El sueño había consolidado lo aprendido la noche anterior.

Nuestros cerebros son en verdad maravillosos.

Interferencia cognitiva

La *interferencia* es lo opuesto a la consolidación: se trata de la interrupción del proceso de consolidación. Si utilizas o practicas una segunda habilidad, semejante a la primera, poco después de practicar esta última, esa práctica

puede interferir con la capacidad de tu cerebro para consolidar la nueva información.

El periodo crítico de interferencia parece ser también de unas cuatro horas. Si esperas para practicar una habilidad conflictiva, dejando que la consolidación tenga lugar, es menos probable que se interfiera con la mejoría obtenida en la destreza primaria.

Por eso es que no practico QWERTY inmediatamente después de practicar Colemak. Eso afectaría la habilidad de mi cerebro para consolidar la práctica de Colemak, reduciendo mi ritmo de aprendizaje.

También es interesante notar que después de siete horas de práctica de Colemak, me cuesta trabajo escribir en QWERTY, a pesar de escribir con este teclado por muchísimo tiempo. Mi cerebro está haciendo un mapa para los movimientos necesarios en Colemak, lo que parece dificultar el acceso a QWERTY al menos por ahora.

Después de que mi cerebro se acostumbre a Colemak, puedo readquirir QWERTY si lo deseo: con base en lo leído de otros mecanógrafos Colemak, es posible ser «bilingüe» en el teclado y pasar de una a otra disposición según convenga. Sin embargo, por ahora me concentro exclusivamente en Colemak para minimizar la interferencia. Puedo esperar para reactivar QWERTY.

Romper con el hábito de mirar

Ya he practicado deliberadamente por 7 horas, con un promedio de 45 minutos cada noche. Completo la secuencia Keyzen sin muchos problemas. Todavía cometo errores, pero con menor frecuencia. Cuando vuelvo a tomar la prueba de velocidad de escritura, mis resultados son mucho mejores: 20 palabras por minuto.

Durante el día, obtengo algo de práctica ambiental bajo la forma de correos electrónicos que requieren respuesta urgente. Cuando me siento a escribir, no es tan desagradable. Soy lento, pero puedo expresarme. ¡A eso le llamo progresar!

Sin embargo, noto que miro mucho el teclado. Tener las teclas correctas en el teclado es muy práctico para teclear cadenas extrañas de caracteres, como contraseñas por ejemplo, pero también es una desventaja: cada vez que me siento inseguro, miro. Si quiero teclear sin ver, debo romper con ese

hábito tan pronto como pueda, pero es difícil dado que se requiere muy poco esfuerzo para bajar la vista.

El teclado vacío

Para eliminar mi hábito de espiar el teclado, decidí adoptar una nueva herramienta de aprendizaje: un teclado completamente blanco.

Das Keyboard es el mejor teclado que encontrarás. El modelo «Ultimate S»^[94] no tiene marca alguna. Si no puedes escribir sin ver el teclado, no puedes usar Das Keyboard. Punto. Es divertido ver cómo reacciona la gente cuando lo ve: hasta los mecanógrafos expertos se sienten intimidados.

Conecté el Das Keyboard y cubrí el teclado de mi laptop con una hoja de papel para no ver las teclas. La sensación fue parecida a la que tuve cuando cambié a Colemak por primera vez: me sentía confundido y frustrado, pero solo por un momento. Las habilidades motoras captadas en las primeras siete horas de práctica funcionaron de pronto y resultó que pude teclear razonablemente bien. Cuando olvidaba una letra, tenía que cazarla durante algunos segundos con prueba y error, pero la libraba.

Como no había nada que mirar, me concentré en la pantalla. Das Keyboard estaba cumpliendo su propósito. Al cambiar el teclado, mi conducta cambió de modo automático.

Junto con Das Keyboard, modifiqué mi método de entrenamiento. Teclear caracteres al azar cansa pronto, así que cambié a un programa llamado Type Fu^[95]. Además de las palabras y letras al azar, Type Fu contienen una base de datos de proverbios y citas, lo que hace que la práctica sea un poco más entretenida. El programa también detecta los caracteres con los que más te equivocas, lo que es útil. Al momento, me cuesta trabajo encontrar la J, la U, la V y la B.

Cada noche practico durante 45 minutos. Después de un total de 14 horas de práctica deliberada, puedo teclear unas 40 palabras por minuto.

En este punto, soy perfectamente funcional: puedo usar el correo y navegar por la red con relativa normalidad. Incluso pude redactar una propuesta de cinco páginas sin demasiados problemas. Me tomó más tiempo de lo usual, pero no ha sido lo más frustrante que he hecho.

Práctica deliberada vs. práctica ambiental

Dado que soy funcional, quiero constatar algo: ¿cuánta importancia tiene la práctica deliberada?

En la actualidad, practico de dos maneras: mis sesiones de Keyzen y de Type Fu son práctica deliberada, dado que me concentro en la destreza y trabajo activamente por mejorar su práctica. El tecleo realizado durante el día es práctica ambiental: cuando escribo un correo electrónico o un ensayo, estoy tecleando en Colemak, aunque me concentro más en el contenido del mensaje que en mi técnica.

Me pregunto: ¿qué pasaría si dejara la práctica deliberada por un tiempo y solo siguiera escribiendo correos y navegando en la red? Ya solo falta 30 por ciento de desempeño para alcanzar la meta de 60 palabras por minuto después de solo 14 horas de práctica deliberada. ¿Podrá ayudarme la práctica ambiental a recorrer el resto del camino sin necesidad de realizar más esfuerzo concentrado?

Decidí hacer un experimento: suspenderé mi práctica deliberada 30 días y veré qué sucede. Seguiré usando Colemak normalmente, sin cambiar a QWERTY. Paso tanto tiempo frente a la computadora que bien puedo llegar a 60 palabras por minuto con práctica ambiental, ¿no?

Después de 30 días, retomé mi prueba de mecanografía. ¿Quieres saber cuál fue mi velocidad de escritura?

Cuarenta palabras por minuto. Ninguna mejoría.

Aunque tecleaba bastante, no estaba concentrado activamente en mejorar mis habilidades. La práctica ambiental no bastó para mejorar.

Si quieres mejorar en una destreza, necesitarás práctica deliberada, al menos en las etapas tempranas de la adquisición de destrezas. Aprendí la lección.

El último esfuerzo

De vuelta a la práctica deliberada, pero hay otra cosa que quiero probar.

Los lenguajes humanos, incluyendo al inglés, siguen la ley de la curva de poder llamada ley Zipf: pocas palabras hacen posible la mayor parte de uso real de una lengua. De acuerdo con un análisis de *The Brown Corpus* (1964)

—una colección de 500 documentos modernos en inglés que suman un millón de palabras— 135 palabras bastan para 50 por ciento del uso total del inglés^[96]. La palabra *the* es responsable de 7,5 por ciento del uso, en tanto que *of* lo fue por 3,5 por ciento.

Puedes llevar esta idea aún más lejos: entre las palabras más comunes en inglés, hay subgrupos de dos y tres caracteres que aparecen una y otra vez, como TH, AN, ING y NCE. Estos grupos se llaman subsecuencias *n-grama*: «n» es una variable que representa el número de caracteres agrupados.

Pude hallar una lista de las n-gramas más comunes en un libro titulado *Cryptological Mathematics*, de Robert Edward Lewand (2000). Las n-gramas son un área mayor de estudio en campos como la criptografía. Si puedes identificar patrones en un mensaje codificado, eso te da una pista sobre el contenido. Al comparar las subsecuencias n-gramas en un mensaje codificado con las n-gramas más utilizadas en el lenguaje que nos interesa, los criptógrafos pueden resolver cifrados complejos.

Pongamos a trabajar esta teoría. He aquí la lista de los más comunes 2-gramas del inglés (digramas) en un orden descendente en relación con la frecuencia de uso:

Th, he, in, en, nt, re, er, an, ti, es, on, at, se, nd, or, ar, al, te, co, de, to, ra, et, ed, it, sa, em, ro.

Y ahora te presento los 3-gramas (trigramas), en orden descendente de acuerdo con la frecuencia de uso:

The, and, tha, ent, ing, ion, tio, for, nde, has, nce, edt, tis, oft, sth, men.

Estas listas son muy útiles. Mientras mejor pueda teclear estas secuencias de caracteres, más rápido podré escribir en general.

Nuestros cerebros son muy buenos para este tipo de cosas: *memoria procedimental* es el concepto que los científicos cognitivos usan para las destrezas motoras que se realizan en un orden determinado. Al practicar los

n-gramas más comunes, puedo entrenar a la memoria procedimental involucrada directamente con la mecanografía.

Para lograrlo, bajé un programa gratuito llamado Amphetamine^[97], diseñado para este tipo de práctica. El programa te permite crear sesiones de entrenamiento a la medida, así como disponer algunos niveles de desempeño, como palabras por minuto o tasa de error.

Cuando comienzas una sesión en Amphetamine, puedes hacer que el programa genere tu práctica de varias maneras. Yo configuro el programa para que despliegue cada n-grama tres veces, en tres secuencias y repitiendo la secuencia tres veces.

Como resultado, mi primera práctica lucía así:

Th he in th he in th he in en nt re en nt re en nt re er an ti er an ti er an ti.

Mi objetivo era teclear la secuencia entera a un promedio de 60 palabras por minuto, con un mínimo de 95 por ciento de precisión. Si no superaba esos criterios, tenía que repetir la secuencia entera.

Practicar así no era entretenido, pero sí fue muy, pero muy efectivo. Repetí cada secuencia una y otra vez hasta que se me grababa y luego continuaba con la siguiente.

Todos los días, cuando me sentaba a practicar, empezaba desde el principio. La mejoría era notoria día tras día: secuencias que el día anterior me habían tomado cinco intentos requerían de solo dos o tres al siguiente. En poco tiempo terminé con todas las secuencias.

Cuando dominé los digramas, pasé a los trigramas. Luego encontré una lista con las palabras más comunes del inglés, compilada por el doctor Peter Norvig.

Norvig es director de investigación en Google. Hace unos cuantos años, publicó las palabras más comunes de la lengua inglesa basado en el trillón que componen el grueso de Google, con todas las palabras independientes que el motor de búsqueda de Google ha indexado^[98].

La intención de Norvig al publicar los datos era ayudar a los programadores a construir utilerías, como por ejemplo revisores de

ortografía, pero en mi caso, esto era el set de entrenamiento perfecto. Tomé las 100 palabras más utilizadas, las añadí a Amphetamine y seguí practicando.

Después de ocho horas adicionales de práctica deliberada, tomé otra prueba de velocidad mecanográfica. El resultado, después de varias pruebas para asegurarme de que no era un error: 60 palabras por minuto, con picos de 70 a 80 palabras y 98 por ciento de precisión. Tiempo total de práctica deliberada: 22 horas.

¡Misión cumplida!

Impresión de las 60 palabras por minuto

Me gusta mucho teclear en Colemak. Ya no estoy frustrado. Más bien, me pregunto cómo aguanté tanto tiempo a QWERTY.

Al teclear en QWERTY sientes que tus manos vuelan por todo el teclado: muchos movimientos en todas direcciones. La disposición no tiene ninguna lógica: los caracteres que usas todo el tiempo están en las zonas más difíciles de alcanzar del teclado.

En contraste, Colemak te hace sentir como si el movimiento de tus dedos hiciera que las palabras aparecieran en la pantalla. Las manos se mueven menos y pasas mucho menos tiempo llegando a caracteres que están en las filas superior e inferior. Es un gran cambio si se compara con QWERTY. No imagino por qué podría volver a utilizarlo.

Repaso del método

Repasemos la esencia del método que utilicé para reaprender mecanografía:

- Aprendí a cambiar la disposición de mi teclado a Colemak.
- Creé un bucle de retroalimentación rápida al reordenar las teclas físicamente en el teclado, de modo que si olvidaba dónde se localizaba un carácter, lo podía encontrar fácilmente.
- Utilicé el tutorial para mecanografía Keyzen para aprender la ubicación de los caracteres sin ver, aprendiendo los más utilizados primero. Keizen reintroducía los caracteres en que me equivocaba, para pasar más tiempo practicando los caracteres difíciles hasta que mi precisión mejoró.

- Practiqué durante 45 minutos cada noche, justo antes de ir a la cama, de modo que mi cerebro pudiera consolidar las habilidades motoras en la memoria de largo plazo de modo más eficiente.
- Cuando alcancé una velocidad funcional (20 palabras por minuto), cambié a Type Fu, concentrándome en teclear las oraciones lo más rápido posible con una precisión mínima de 99 por ciento.
- Cuando llegué a 40 palabras por minuto, usé Amphetype para entrenarme en usar los digramas y trigramas más comunes de la lengua inglesa, aumentando mi velocidad y precisión.
- Cuando dominé las secuencias de n-gramas, cambié el entrenamiento para usar las palabras de uso más común en el inglés con ayuda de Amphetype, hasta llegar a 60 palabras por minuto sostenidas con 98 por ciento de precisión, lo que sucedió en la marca de las 22 horas.

¿Y ahora qué?

Ahora que puedo teclear 60 palabras por minuto en Colemak, no tengo necesidad de seguir entrenando para adquirir más velocidad. En este nivel, puedo escribir tan rápido como necesito. Dado que la velocidad de escritura ya no es una limitante, el entrenamiento de velocidad ya no es una prioridad.

El tecleo a velocidad es una habilidad por sí misma. Algunos de los mecanógrafos más rápidos del mundo escriben alrededor de 180 palabras por minuto en teclados normales, pero mejorar en pruebas de velocidad no necesariamente implica mejorar en velocidad de escritura o programación. Estas pruebas siempre tienen que ver con escribir lo que ves en la pantalla, así que la habilidad primaria que los mecanógrafos veloces practican implica mirar más adelante en el texto y procurar que este permanezca en la memoria de corto plazo el tiempo suficiente para que los dedos lo escriban.

Aunque me encantaría poder escribir prosa corrida a más de 180 palabras por minuto, eso está fuera del alcance incluso de los mecanógrafos más veloces. Mecanografiar es más difícil y lento cuando tienes que crear lo que pones en la página.

No obstante, puedo beneficiarme si reduzco mi margen de error. Con esa idea, sigo practicando las palabras comunes, los digramas y trigramas con Amphetype y el uso de mi dedo meñique de la mano izquierda para corregir errores. Al reducir el margen de error, mi eficiencia y velocidad aumentarán.

Lo que más me sorprendió al aprender Colemak fue cuán fácil resultó reescribir 20 años de experiencia previa en el uso de QWERTY. Pensaba que dos décadas de memoria muscular necesitarían más de 20 horas para ser remplazadas. Estaba equivocado.

Nuestros cerebros cambian con mayor facilidad de lo que pensamos.

7

Go

Lección: explorar y luego decidir.

El Go usa los conceptos y materiales más elementales: línea y círculo, madera y piedra, blanco y negro, todo combinado con reglas sencillas que generan estrategias sutiles y tácticas complejas que estremecen la imaginación.

Iwamoto Kaoru, jugador profesional de Go, noveno Dan

Para obtener imágenes, videos y comentarios adicionales sobre este capítulo, visita <http://first20hours.com/go>.

Soy la espada en la oscuridad, el vigía en la pared.

La noche es fría: cae la nieve. Detrás de mí, las antorchas bañan con su luz trémula a mis compatriotas. Estamos en la muralla del castillo, observando. Esperando.

En el extremo más lejano del puente del castillo, los soldados enemigos se reúnen en un punto fuera del alcance de las catapultas y preparan el ataque.

Guerreros, ladrones, exploradores, todos afilan sus armas mortales. Los nigromantes convocan armadas de desagradables criaturas no muertas. Los ingenieros preparan los morteros y los arietes flamígeros, armas poderosas diseñadas para abatir nuestros portones.

Esta horda intenta robar la posesión más preciada de mi reino: un globo relumbrante de gran poder. No lo lograrán. No mientras yo viva.

Estamos preparados. A mis costados, los soldados surten las ballestas y los carros de flechas, quedando así listos para mandar sus cargas mortales en dirección a la fuerza invasora. Entre las defensas, están los magos que preparan una lluvia de fuego y rayos para los enemigos. Nuestros guardianes e hipnotizadores han dispuesto runas y espejos reflejantes por todo el perímetro, brindándonos protección ante las flechas y los hechizos, al menos por un tiempo.

Esperamos. Cae la nieve y las antorchas hacen danzar su luz en mi armadura de acero pulido.

No nos derrotarán. No nos romperemos. Pelearemos y ganaremos.

Una piedra enorme surge de la oscuridad. Vuela sobre las líneas enemigas. La roca se estrella contra el portón del castillo cimbrándolo y astillando su madera.

El enemigo ruge y luego se lanza a la carga cruzando el puente.

En un instante, tomo mi espada y mi escudo. Mi grito de guerra hace eco en cada piedra, llevando el poder y la furia a cada defensor de la muralla.

«¡Por la gran justicia!».

Guarda tu arma; no pretendo hacerte daño

Dirigir un ejército durante una gran operación de sitio es una buena manera de pasar una noche de sábado.

No soy fanático de mirar la televisión, películas o deportes. Más bien, podrás encontrarme jugando videojuegos como *World of Warcraft* o *Guild Wars 2* durante una hora o dos.

Disfruto los juegos, en especial los que me presentan retos que involucran hechizos, monstruos que luchan y obligan a ser más astuto que los rivales. Desde joven, me gustaron las historias épicas sobre magos, guerreros y logros

heroicos. Los videojuegos me permitieron ser un personaje en estas aventuras, al menos por un tiempo^[99].

Mi generación fue la primera que creció con juegos de rol para computadora. Todo empezó con las primeras consolas Atari y Nintendo; los juegos han crecido en detalle y complejidad. Ahora, es posible luchar vía Internet con dragones y contra jugadores de todo el mundo.

Parte de lo que me gusta de estos juegos tiene que ver con la destreza necesaria para jugar bien. Cualquiera puede crear un personaje, pero para entrar en batalla y salir victorioso debes saber bien qué haces.

Ha pasado mucho tiempo desde que Mario brincaba sobre sus enemigos y lanzaba su ocasional bola de fuego. Ahora, es común que los personajes de los videojuegos tengan cerca de cien habilidades potenciales. Para jugar bien, debes conocerlas, saber cuándo usarlas y cómo adaptar a tu personaje para tener el mayor efecto posible en el juego.

Hay mucho que aprender. ¿Qué habilidades hacen más daño o te protegen mejor? ¿Qué pueden hacer los monstruos enemigos? ¿Cuál es la mejor estrategia para luchar con otros jugadores humanos?

Mientras más juego, mejor es mi desempeño. Por eso es divertido.

Históricamente hablando, los videojuegos son bastante nuevos. Sin embargo, los juegos de destreza y azar han formado parte de la experiencia humana durante miles de años.

El juego estratégico de mesa más antiguo del mundo

Hace poco me topé con un juego interesante. Comparado con los que uso, este es silencioso, casi sereno. Sin embargo, bajo la superficie hay un gran drama. El tablero es un mapa de guerra y los jugadores son generales que luchan entre sí para obtener la supremacía definitiva.

El Go es el juego más antiguo del mundo y aún se juega en su modo original. Con base en registros históricos, se puede afirmar que el Go se originó en la antigua China, y que ha existido con las reglas actuales al menos 3000 años, aunque algunos historiadores opinan que hablamos más bien de unos 4000 años. Si la edad es indicador de calidad, el Go tiene muchísimo que ofrecer.

El nombre chino para el Go es *weiqi*. *Wei* (圍) significa «rodear» y *qi*

(棋) significa «juego de mesa». Juntas, estas partículas se convierten en *weiqi* (圍棋), en chino simplificado: (圍棋), que significa «juego en que se rodea». Es una buena y sencilla descripción de la condición victoriosa del juego: rodear al oponente.

El Go fue introducido en Europa y América vía Japón, así que el término Go es una simplificación de la palabra japonesa que denota al juego: *igo* (囲碁). No importa si te refieres al juego con el término Go, *weiqi*, *igo*, *baduk* o cualquier otro, el juego es el mismo.

El arte de la guerra

En Occidente, el ajedrez es el juego de mesa estratégico más popular, así que podemos utilizarlo para comparar. En la superficie, los juegos guardan muchas similitudes.

En cada uno de los juegos participan dos jugadores al mando de las blancas y las negras. El juego tiene lugar en un tablero cuadrado. Las negras mueven primero (las blancas en el ajedrez). Los jugadores participan por turnos hasta que el juego termina. Se puede pensar como un conflicto bélico en el que los jugadores hacen las veces de los generales enemigos.

Hasta ahí las semejanzas.

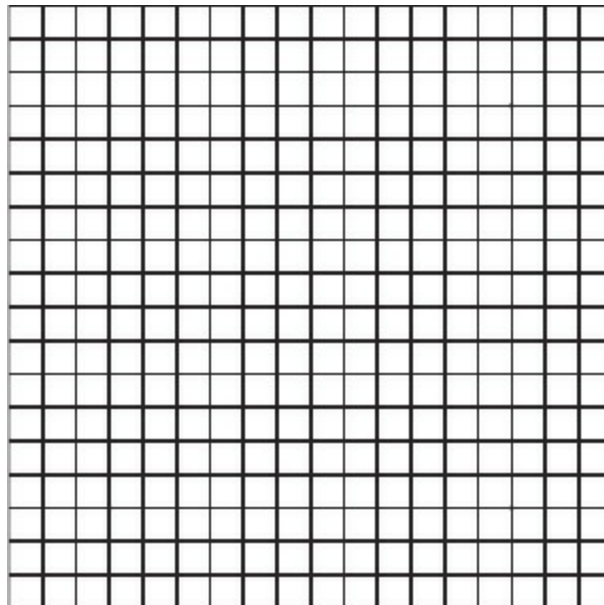
El ajedrez se juega en un tablero de 8×8 cuadros (escaques), lo que da un total de 64. Cada pieza del ajedrez ocupa un solo cuadro. El tablero se parece a un campo de batalla y las piezas son los soldados.

He aquí cómo se ve un tablero de ajedrez al principio del juego:



Por otra parte, el Go se juega en un tablero formado por 19 líneas verticales que forman intersección con otras 19 horizontales. Las piedras se colocan en las intersecciones, no en los cuadros. Como resultado, hay 360 intersecciones disponibles para el juego en el tablero del Go, 5625 más veces que en el ajedrez.

Ahora te presento cómo luce un tablero de Go al principio de un juego:



¿Notas la diferencia? No solo el tablero es mucho más grande, sino que el juego comienza sin que haya piedras en este. En el Go, las piedras se añaden

durante el juego. En el ajedrez, las piezas se retiran del tablero conforme son capturadas.

Date cuenta de que el *goban* (tablero) es cuadrado. Los puntitos en el tablero (llamados «puntos estrella») son perfectamente simétricos. Esos puntos son importantes y regresaremos a ellos dentro de poco.

En el ajedrez, existen seis tipos de piezas y cada una tiene reglas y capacidades especiales. Los peones siempre se mueven hacia el frente, excepto cuando capturan otra pieza. Los alfiles se mueven en diagonal, pero no lo hacen vertical u horizontalmente. Las torres se mueven horizontal y verticalmente, pero no en diagonal. Los caballos pueden saltar sobre otras piezas, pero deben moverse dos escaques horizontales y uno vertical, o dos escaques verticales y uno horizontal. La dama puede moverse horizontalmente, verticalmente y en diagonal, pero no puede saltar otras piezas. El rey puede moverse en cualquier dirección, pero solo un escaque en cada ocasión, a menos que salte a la torre en un movimiento especial que se llama «enroque». Hay mucho para recordar.

En contraste, cada movimiento de Go es igual: se coloca una piedra en una intersección y, al menos que sea capturada más tarde en el juego, una piedra puesta en el tablero es una piedra jugada.

Las piezas de ajedrez son capturadas por alguna de las piezas del oponente, como si se tratara de combate cuerpo a cuerpo en el campo de batalla. La torre parece dar un mazazo al alfil en la cabeza y el alfil sale del juego.

Las piedras de Go son capturadas cuando las del oponente logran rodearlas por los cuatro costados. Cuando el ejército negro es rodeado por el blanco, las negras se rinden y se hacen prisioneras.

Los juegos profesionales de ajedrez suelen contar con alrededor de 40 movimientos. En el Go, se considera que los primeros 30 movimientos, más o menos, son de apertura: el final del juego comienza alrededor del movimiento 100. Los juegos de Go suelen consistir en 250 movimientos.

Como sea, la escala del Go es mucho mayor que la del ajedrez. Si cada juego de ajedrez es una batalla, el *goban* es una batalla masiva.

¿Cómo se juega el Go? Veamos.

Las reglas del juego

Aunque no lo creas, el Go tiene solamente siete reglas principales y ya hemos visto dos de ellas:

1. Las piedras se colocan en las intersecciones.
2. Las negras y las blancas toman turnos para poner piedras en el *goban*.

Las siguientes cinco reglas definen la evolución del juego y las condiciones para la victoria:

1. Las piedras son «capturadas» cuando están rodeadas por las piedras del oponente en todos los costados.
2. Está prohibido jugar con una piedra que se captura inmediatamente (una especie de suicidio).
3. Está prohibida la repetición continua de la misma secuencia de movimientos (situación llamada *ko*).
4. El juego termina cuando a los jugadores se les terminan las piedras, cuando uno se rinde, o ambos pasan.
5. Gana el jugador que logre rodear mayor territorio en el tablero al final del juego.

En las competencias oficiales, se introducen otras reglas para eliminar las ambigüedades (como qué constituye un ciclo de repetición *ko*), se especifica el método de anotación y se evitan los empates. Fuera de lo anterior, las reglas son las que ya hemos visto.

Bastante simple, ¿no?

Aprender las reglas del Go es fácil: solo toma unos minutos. Las reglas en sí no son complicadas.

No obstante, hay un antiguo proverbio relativo al Go que dice: «Se necesita muy poco tiempo para aprenderlo y una vida entera para lograr la maestría». Combina estas reglas simples con un tablero sencillo y piedras también sencillas y tendrás una complejidad tremenda.

El tamaño del universo

En tanto que las barrocas leyes del ajedrez solo pudieron ser creadas por humanos, las reglas del Go son tan elegantes, orgánicas y rigurosamente lógicas que, si la vida inteligente existe en otro lugar del universo, casi seguramente jugarían al Go.

Edward Lasker, campeón mundial de ajedrez y autor de *Go and Go Moku*

Digamos que queremos crear una computadora que juegue Go de modo inteligente, como Deep Blue, famoso programa de inteligencia artificial que derrotó a Garry Kasparov en 1996, quien entonces era el campeón mundial de ajedrez.

Por lo regular, las computadoras vencen a los humanos por medio de su fuerza bruta de cómputo: calculan todos los movimientos legales y luego eligen la jugada con mayores probabilidades matemáticas de éxito, con base en una biblioteca conformada con datos de juegos pasados.

En un tablero de ajedrez, este tipo de cómputo no es sencillo, pero es posible. Hay 64 escaques y el movimiento de cada pieza está constreñido por reglas. Dado que cada pieza puede moverse de cierta manera, el programa solo necesita considerar algunas opciones.

En el Go, el jugador puede colocar una piedra en cualquier intersección disponible en el tablero. El juego comienza con 360 posibles ubicaciones, así que, desde el principio, el programa debe analizar muchas más probabilidades.

Hagamos cálculos matemáticos rápidos. ¿Cuántas secuencias de cinco movimientos pueden jugarse en un tablero de Go, asumiendo que se trata del inicio del juego y que ningún jugador captura las piedras del oponente?

Aquí está el cálculo:

$$360 \times 359 \times 358 \times 357 \times 356 = 5\,880\,282\,488\,640$$

Estamos hablando de 5,8 billones de secuencias posibles, solo durante los primeros 5 movimientos.

Las matemáticas enloquecen rápidamente. ¿Recuerdas que mencioné que no era raro que los juegos de Go tuvieran 250 movimientos? Dependiendo de los parámetros que se supongan, hay aproximadamente 2.08×10^{170} secuencias de movimientos legales en un tablero de Go de 19×19 cuadros.

Si los cálculos matemáticos son correctos, *existen más jugadas legales de Go que partículas subatómicas en el universo conocido*.

Es una certeza matemática que todo el juego de Go no ha sido desarrollado en la historia del universo, incluso si aventuras la posibilidad de que hubiera billones de civilizaciones extraterrestres por ahí jugando al Go^[100].

Con la tecnología disponible, se necesitaría que las computadoras más sofisticadas corrieran los algoritmos más eficientes en procesamiento por fuerza bruta durante unos 400 años para calcular una sola jugada óptima, asumiendo que el programa completara un cálculo cada pocos milisegundos.

Estoy estupefacto. Este juego es *enorme*.

¿Cómo juegan al Go los jugadores (y las computadoras)?

Si los humanos tuvieran que analizar por fuerza bruta para jugar al Go, se volverían locos. Es obvio que no lo hacemos así: los buenos jugadores humanos pueden identificar los mejores movimientos en el tablero en pocos segundos. ¿Cómo lo logran?

Los jugadores de Go dependen del reconocimiento de patrones para identificar los movimientos de alto valor. Los jugadores suelen utilizar muchas palabras que aluden a la intuición, como «forma», y *sente* (iniciativa). Los mejores jugadores parecen dar la misma importancia a la geometría, la belleza estética y la emoción, que a la lógica rigurosa y al análisis.

Eso tiene sentido: el cerebro humano no está muy bien equipado para el análisis por fuerza bruta, pero es excelente para reconocer patrones. Al notarlos en las piedras en el tablero, así como los patrones de juego en las piedras que se están jugando, los jugadores avanzados de Go pueden leer la situación actual para encontrar el mejor movimiento: todo en menos de 400 años.

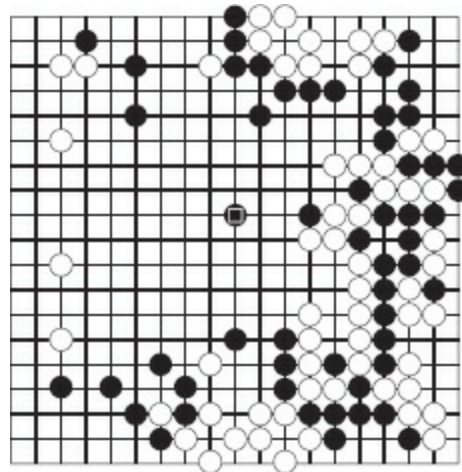
Más impresionante aún es que los mejores jugadores pueden anticipar qué piedras se jugarán en el futuro, con entre 30 y 40 movimientos de anticipación. Si alguna vez juegas con un profesional, probablemente sentirás que te está leyendo la mente.

Un juego de verdad

La diferencia entre una piedra colocada en una intersección y no en la vecina es insignificante para el no iniciado. Sin embargo, para los maestros de Go, existe entre las dos jugadas tanta diferencia como la que encontramos entre una flor y un pedazo de carbón.

Dave Lowry, autor de *The Challenge of Go: Esoteric Granddaddy of Board Games*

Aquí les muestro cómo se ve un juego de Go en progreso:



Este es un diagrama de un juego real. De hecho, se trata de uno de los juegos más famosos en la historia del Go^[101].

En 1846, Shusaku de la Casa Honinbo, un cuarto dan de 17 años, fue invitado a jugar con Gennan Inseki, un octavo dan. Inseki era la cabeza de la casa Inoue, quienes dirigían una de las cuatro principales escuelas profesionales de Go en el Japón de mediados del siglo XIX.

Shusaku aceptó el honor y el juego atrajo a un auditorio considerable. Nadie esperaba que Shusaku ganara, pero su juego era lo suficientemente fuerte para dar la pelea.

Después de un juego corto en que Inseki concedió a Shusaku una ventaja de dos piedras, quedó claro que Shusaku no necesitaba dicha ventaja. Inseki aceptó comenzar un segundo juego sin ventajas de ningún tipo para nadie.

Shusaku, siendo el retador, jugó con las piedras negras. La apertura del juego fue perfecta, excepto por un pequeño error en un intercambio en el extremo inferior derecho del tablero. Ciento veintiséis turnos después, Inseki conservaba la ventaja, tal como se esperaba. Después de todo, ¿quién podía vencer al maestro?

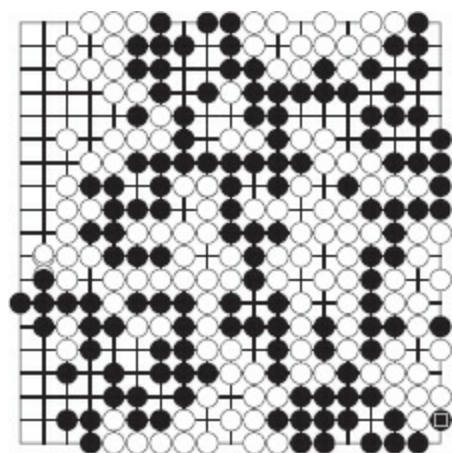
La siguiente jugada de Shusaku cambió el juego. Puedes ver la piedra tú mismo: se trata de la piedra negra que está justo encima del centro del tablero marcada con un cuadro.

Después del movimiento de Shusaku, un espectador notó algo raro: Inseki tenía las orejas rojas. El maestro estaba enojado.

El movimiento 127 de Shusaku fue bueno, muy muy bueno. Esa sola piedra central apoyaba simultáneamente sus piedras en la parte superior derecha e inferior del tablero, en tanto que ejercía nueva influencia en el costado izquierdo. La jugada mostraba un equilibrio perfecto entre ofensiva y defensiva.

Inseki estaba en problemas y lo sabía. Con una sola piedra, Shusaku ejerció su influencia en todo el tablero. El juego continuó, pero Inseki luchaba ahora por recuperar el control.

Esta era la posición en el movimiento 325:



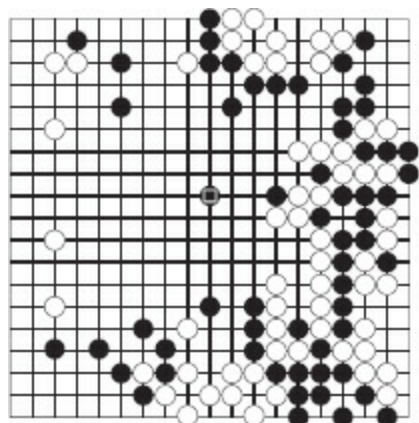
Tómate un minuto para estudiar el tablero. ¿Qué zonas están rodeadas por las negras y cuáles por las blancas? ¿Quién ha rodeado más territorio en el goban?

Inseki aceptó la derrota y Shusaku ganó por dos puntos. El juego pasaría a la historia como el de las Orejas rojas, y el movimiento 127 se conoce como el Enrojecedor de orejas.

«El invencible Shusaku» se convirtió en uno de los jugadores más celebrados en la historia del Go, reconocido por sus 19 victorias consecutivas en el castillo, en un certamen celebrado anualmente bajo el patrocinio del Shogun. Murió el 7 de septiembre de 1862, a la edad de 33 años, tras atender enfermos en un brote de cólera.

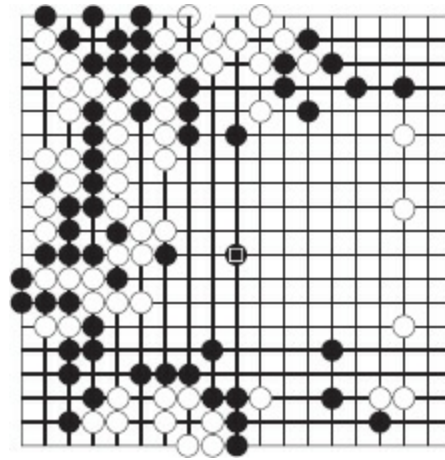
Reconocimiento de patrones

Podemos aprender de estos diagramas. Te muestro de nuevo el movimiento Enrojecedor de orejas:



Los juegos de Go suelen presentarse desde la perspectiva de las negras. Vemos el tablero tal y como lo hacía Shusaku durante el juego.

Aquí vemos el tablero desde la perspectiva de Inseki (blancas), rotándolo 180 grados:



En el ajedrez, solo existen dos perspectivas primarias: las negras están enfrente de las blancas y ambos jugadores comienzan la partida con sus respectivas piezas colocadas en su lado del tablero. El ajedrez nunca se juega con las piezas blancas y negras dispuestas en los costados izquierdo y derecho del tablero. Además, las jugadas de ajedrez en general se mueven en dirección al oponente.

Como resultado, los jugadores de ajedrez pueden aprender a reconocer patrones específicos en el tablero. A continuación les muestro un famoso patrón llamado Gambito de rey:



Una buena parte del entrenamiento para llegar a ser gran maestro de ajedrez implica memorizar patrones comunes que aparecen en los juegos. Los mejores jugadores de ajedrez ven cómo se desarrolla un patrón en el tablero en un instante, dado que los patrones se ven iguales. Si tratas de jugar un Gambito de rey, un gran maestro lo notará de inmediato y sabrá exactamente cómo responder.

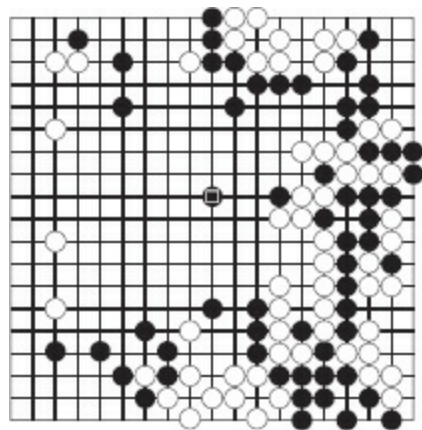
En tanto que el reconocimiento de patrones ajedrecísticos no es fácil ayuda el hecho de que los jugadores nunca deben rotar el tablero, lo que hace que los patrones sean más fáciles de aprender.

En contraste, dado que el *goban* es perfectamente simétrico, ningún lado del tablero tiene una significación especial. Como resultado, puedes ver el *goban* desde cualquiera de sus cuatro costados.

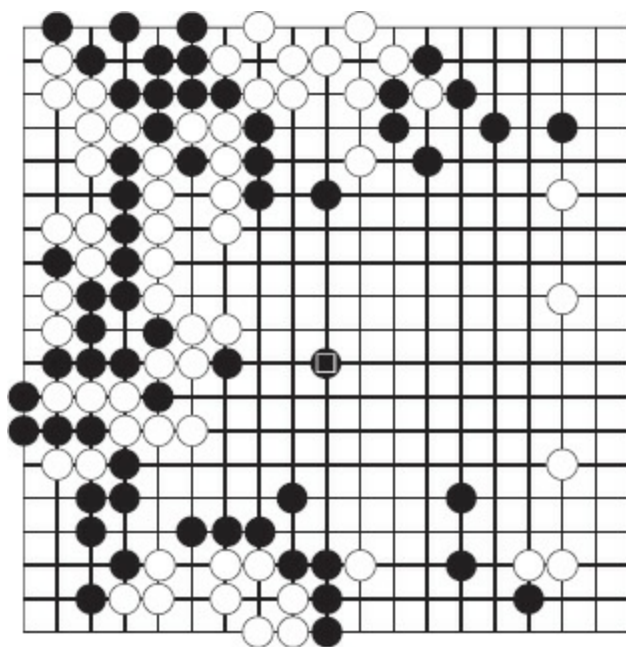
Al principio de un juego de Go, las piedras de ventaja son colocadas por las negras en los puntos estrella, esos pequeños puntos negros, en un patrón simétrico. Por lo demás, el tablero permanece vacío.

Las piedras del Go se colocan en cualquier intersección vacía en cualquier momento. Un jugador puede jugar en un turno en la parte superior del tablero, cambiando a la parte inferior para el siguiente turno. Pueden jugar en el lado derecho y en el izquierdo. No existe una tendencia universal predecible, como sucede en el ajedrez.

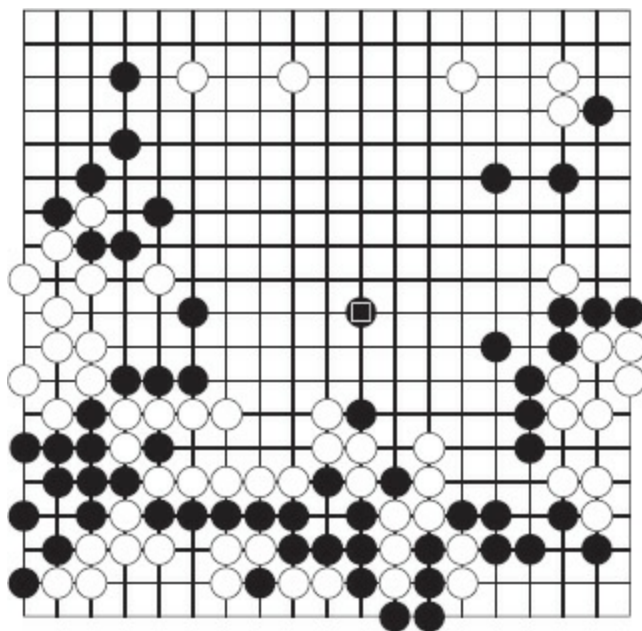
Volvamos al movimiento Enrojecedor de orejas. He aquí la perspectiva de Shusaku:



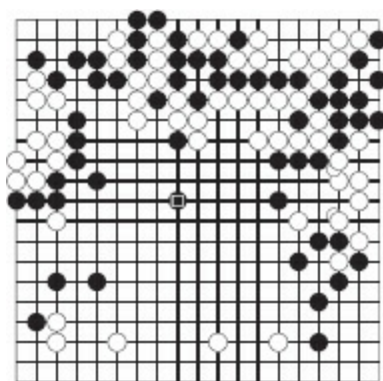
De nuevo, veamos la perspectiva de Inseki:



Ahora, veamos la perspectiva desde la derecha de Shusaku, o desde la izquierda de Inseki:



Y la perspectiva desde la izquierda de Shusaku, o desde la derecha de Inseki:



La naturaleza simétrica del *goban* hace que el reconocimiento de patrones sea mucho más difícil. Estos cuatro diagramas del movimiento Enrojecedor de orejas son *exactamente iguales* desde el punto de vista estratégico, aunque lucen muy distintos para el ojo no entrenado. Si lo desearas, podrías rotar noventa grados un tablero de Go tras cada movimiento sin afectar de manera notable el juego.

Los jugadores de Go no pueden confiar en técnicas de memorización directa, como los ajedrecistas, para identificar patrones en desarrollo durante un juego. Cada patrón lo aprenden los jugadores de Go desde cuatro perspectivas, no desde una sola.

Por eso, los jugadores de Go dependen tanto de la intuición. El juego es demasiado grande para memorizar o calcular, así que los jugadores destacados se entrenan para detectar patrones a un nivel muy alto: miran formas generales, direcciones de movimiento e impresiones de fuerza y debilidad.

Sigue tus corazonadas...

[El Go] requiere la táctica del soldado, la exactitud del matemático, la imaginación del artista, la inspiración del poeta, la calma del filósofo y la mayor inteligencia.

Zhang Yunqi, en un documento interno del Instituto Chino Weiqi, 1991

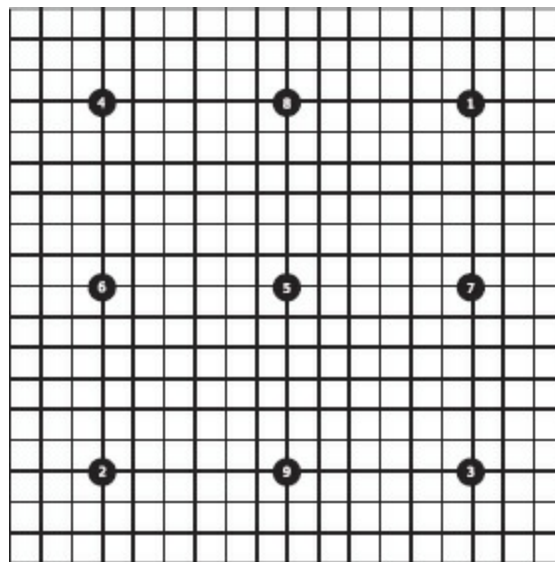
Uno de los primeros artículos que leí sobre el Go exponía un argumento interesante: la maestría en el juego no tiene que ver en realidad con la

competición o el dominio del oponente. Se trata de lograr la maestría con uno mismo.

Los juegos de azar han estado presentes desde el alba de la civilización. Al arrojar dados, la física decide quién gana. La destreza no es un factor determinante.

En contraste, los juegos de destreza suelen concentrarse en superar en ingenio a otros jugadores. ¿Quién demuestra tener mayor aptitud y control? ¿Quién puede hallar las mejores oportunidades primero? ¿Quién es mejor para explotar las debilidades del rival? El ajedrez cae en esta categoría: para ser muy bueno se requiere la maestría de los factores tácticos y la correcta lectura de la intención del oponente.

El Go es un juego único en el siguiente aspecto: la introducción de piedras de ventaja es una forma de debilitar intencionalmente a los jugadores fuertes. Si tu oponente comienza el juego con algunas piedras en posiciones estratégicas del tablero, se influye mucho en el juego. He aquí los lugares en que se colocan las piedras que dan ventaja:



En juegos de Go en que la fuerza de ambos jugadores es semejante, cada uno ha de ganar la mitad de las veces, más o menos. Si un jugador domina al otro, en el siguiente juego el que perdió comienza con una o dos piedras adicionales en el tablero.

Debido a este factor de equilibrio, puedes pensar en el Go como si se tratara de un juego contra ti mismo. Seguro que tomas las decisiones con base en lo que hace tu oponente, pero ganar el juego no es el único objetivo.

Conforme aumenta tu destreza, disminuye el número de piedras de ventaja que necesitas. Cuando juegas contra oponentes fuertes, podrás dar batalla en igualdad. Eventualmente, darás algunas piedras a rivales más débiles.

La historia del Go lleva el aspecto de la maestría autodidacta hasta los extremos: para aprender a jugar bien se requiere dominar tus pensamientos y emociones. Los jugadores avanzados aprenden a ver en las piedras signos que revelan el estado mental y emocional del oponente.

Según las leyendas, los maestros antiguos podían leer la transcripción de un juego e identificar los momentos en que un jugador sentía ira, confusión, envidia o codicia, así como también señalar «el instante en que la sirvienta entró para servirles el té».

Las piedras ninja

A lo largo de los siglos, el sistema de ventajas del Go evolucionó hasta convertirse en un sistema de rangos: la diferencia en rango determina cuántas piedras puede colocar el jugador de menor rango al principio.

Aunque no lo creas, el sistema usado para los cinturones de las artes marciales se deriva de las jerarquías del Go. Conforme aumentan su destreza, ascienden en el *ranking*.

Los principiantes empiezan en el nivel 35 kyu (equivalente a la cinta blanca). Cuando los jugadores se hacen más fuertes, ascienden hasta llegar a 1 kyu.

El siguiente nivel después de 1 kyu es el primer dan, equivalente a la cinta negra de las artes marciales. De ahí, los jugadores avanzan hasta llegar al noveno dan, el nivel oficial más alto; el décimo dan es un título honorario reservado para los ganadores de los torneos más importantes del mundo.

El nivel de un jugador se establece de varias maneras. La primera es la competencia: si un jugador puede ganar la mayoría de las partidas, en condiciones de igualdad, contra jugadores del nivel 12 kyu, pero pierde la mayoría de los juegos al enfrentarse a jugadores 8 kyu, su fuerza probable

estará en el rango del 10 kyu. Mientras más juegos complete un jugador en competencia, más preciso será su *ranking*.

La otra forma de estimar el rango se basa en la resolución de problemas de Go: se trata de acertijos estructurados que presentan una situación determinada; se le pide al jugador que determine cuál es la mejor jugada para obtener un resultado específico, como «capturar un grupo negro» o «salvar la posición de las blancas». Conforme los jugadores ascienden de rango, su habilidad para resolver problemas correctamente también se incrementa.

Dado que el Go ha estado presente por tanto tiempo, existe una enorme biblioteca de problemas sobre este juego para su estudio. Están organizados por grado de dificultad: un jugador 20 kyu luchará para resolver los problemas diseñados para 10 kyu, mientras otro de primer dan pensará que los problemas para los 10 kyu son fáciles y obvios. Kiseido, editor especializado en libros de Go, publica una serie muy popular de problemas divididos por niveles llamada *Graded Go Problems for Beginners*, muy útil.

La existencia de problemas con grado de dificultad establecido hacen del Go un juego que puedes estudiar y jugar. Solucionar los problemas contenidos en estos libros es una buena manera de practicar las destrezas del Go por cuenta propia, y también sirven para estimar tu fortaleza de juego relativa.

Estoy listo para empezar. ¿Qué necesito para practicar?

El equipo necesario

Es difícil jugar Go sin tablero ni piedras, así que compré un buen Go de Yellow Mountain Imports^[102], compañía que distribuye productos de Go de todo el mundo en Estados Unidos.

Los tableros y las piedras de Go varían mucho en precio y calidad. Los tableros baratos y las piedras de cristal sencillas pueden adquirirse por unos cuantos dólares, así que un juego básico es fácil de obtener.

Al otro lado del espectro, los tableros de calidad superior hechos con fina madera de *kaya*, el material usado tradicionalmente en Japón, se venden por decenas de miles de dólares. Igualmente, puedes comprar piedras hechas con verdadera pizarra y concha, pero te costarán bastante.

Después de investigar un poco, me decidí por un bonito tablero de piso

shin-kaya (imitación *kaya*; normalmente se trata de picea blanca) con un juego de piedras Yunzi, hechas en China por una empresa del mismo nombre. La compañía considera que la composición de sus piedras es un secreto profesional: las piedras se sienten bien, sólidas, y suenan muy agradablemente cuando las colocas en el tablero. El precio es razonable. Las piedras vienen con cuencos tradicionales de madera, mismos que se colocan junto al *goban* durante el juego.

Además de comprar el *goban* y las piedras, también escogí varios libros de Go para principiantes:

- *Go: A Complete Introduction to the Game*, por Cho Chikun (2010).
- *The Second Book of Go: What You Need to Know After You've Learned the Rules*, por Richard Bozulich (1998).
- *How Not to Play Go*, por Yuan Zhou (2009).
- *Lessons in the Fundamentals of Go*, por Toshiro Kageyama (1996).
- *Opening Theory Made Easy*, por Otake Hideo (1992).

Encontré estos libros por medio de un sitio de Internet bastante útil llamado Sensei's Library^[103], que tiene cientos de páginas de información y comentarios sobre la historia y la técnica del Go. En una hay una enorme lista de opiniones sobre los mejores libros de Go disponibles. La lista fue realizada por jugadores avanzados, lo que es muy importante en esta etapa del aprendizaje. Leí toda la página, que contenía comentarios sobre cerca de cien libros.

He aquí cómo elijo qué libros leer primero. *Go: A Complete Introduction to the game era*, por mucho, el más recomendado para los principiantes, así que no hubo problema. Igualmente, *The Second Book of Go* fue muy recomendado como primer texto relativo a la estrategia. El libro asume que ya se saben las reglas básicas y se dedica a explorar técnicas fundamentales.

How Not To Play Go es un ejemplo de inversión que me dio mucho gusto encontrar: aprendes mucho sobre cualquier cosa estudiando los errores más comunes. Para la mayoría de las destrezas, uno mismo debe realizar el proceso de invertir los pasos, pero en este caso hay un libro escrito por un experto. ¡Maravilloso!

Lessons in the Fundamentals of Go, también llamado «El libro amarillo», es el que la mayoría de los jugadores avanzados citan como el que mejoró su juego de manera notable. Parece bastante avanzado para un principiante, de modo que lo abordaré después de atender las cuestiones básicas.

Finalmente, elegí *Opening Theory Made Easy* porque es claro para los principiantes y la apertura del juego es extremadamente importante. Dado que al principio el *goban* está vacío, a no ser por algunas piedras de ventaja, las primeras 30 o 40 piedras que se juegan crean una estructura que influye de modo profundo en el resto del juego. Si no sabes cómo jugar la apertura correctamente y tu oponente sí, probablemente perderás, por lo que vale la pena estudiar la apertura desde el principio.

Sigue al conejo blanco

Además, la tecnología ha hecho que la práctica del Go sea un poco más sencilla. SmartGo, un programa disponible para iPhone y iPad, incluye un muy buen programa de juego, una base de datos con problemas de Go organizado por niveles y juegos importantes de la historia para analizar.

SmartGo está diseñado para aprovechar las ventajas de la pantalla táctil, así que puedes ubicar tus piedras «directamente en el tablero» tocando la intersección indicada. Esto hace mucho más fácil resolver los problemas de Go. En lugar de imaginar la solución para después buscarla en la parte posterior de los libros impresos, el programa te da una retroalimentación inmediata. Este rápido círculo virtuoso de la retroalimentación inmediata hace mucho más sencilla la práctica con problemas, particularmente en los que se requiere hacer más de una jugada para llegar a la solución correcta.

Estoy listo. Tengo todo lo necesario para aprender a jugar. Solo hay una cosa más que debo hacer... asegurarme de que paso el tiempo jugando Go y no otra cosa.

Eliminación de distracciones

El Go no es tan visceralmente atractivo como los juegos de video ricos en acción. Aprender Go requerirá de tiempo y concentración. Ya estoy jugando

otros juegos, pero tengo poco tiempo disponible. Si quiero progresar a mi mejor ritmo, necesito concentrarme.

Eso quiere decir que debo eliminar las distracciones. La amenaza inmediata está en los otros juegos: el tiempo invertido en derrotar monstruos digitales *no* es tiempo dedicado al aprendizaje del Go. Si sigo con mis videojuegos, no tendré tiempo para aprender.

Recuerda: el tiempo nunca se encuentra, se produce.

En ese sentido, suspendo otros juegos hasta invertir al menos 20 horas en el Go. No puedo permitirme distracciones si quiero aprender rápidamente.

Esta es una práctica bastante útil: la mejor manera de cambiar tu conducta es modificar la estructura de tu medio ambiente. Si no quieres hacer algo, haz que sea imposible seguir haciéndolo. Si no puedes hacer que la conducta indeseada sea imposible, al menos haz que su práctica resulte tan difícil, cara y prohibitiva como puedas. Mientras más esfuerzo se requiera, más difícil será que vuelvas a la conducta que deseas eliminar.

Adiós, *World of Warcraft*... mucho gusto en haberte conocido.

Antes de embarcarme en la aventura del Go, cancelé mi cuenta de *World of Warcraft* y borré el juego de la computadora. Si el juego no está instalado, no puedo jugarlo aunque quiera. Bajo ninguna circunstancia jugaré con videojuegos hasta que mi experimento con el Go finalice.

Las reglas del juego, reexaminadas

La mejor opción para comenzar es un nuevo examen de las reglas del juego. Ya las leí, pero necesito saber cómo se aplican en el tablero.

Una característica bastante agradable que facilita el aprendizaje del Go es que se puede cambiar el tamaño del tablero. Dado que el *goban* es simétrico, puedes agrandarlo o reducirlo, siempre y cuando elijas un número impar de líneas. Aplican las mismas reglas y los mismos principios tácticos.

Los tableros de competencia tienen 19×19 líneas pero, para aprender, es mejor comenzar con tableros de 7×7 . Es un buen tamaño para aprender las ideas clave, y lo bastante pequeño para que no resulte abrumador.

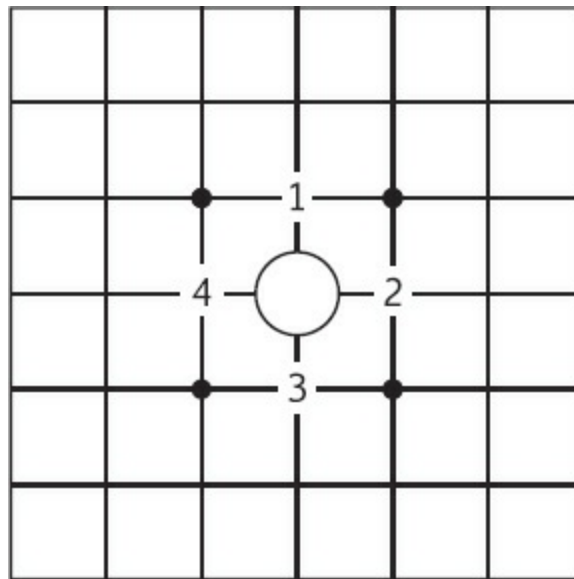
Veamos cómo se aplican estas reglas en el tablero.

Dame libertad o dame la muerte

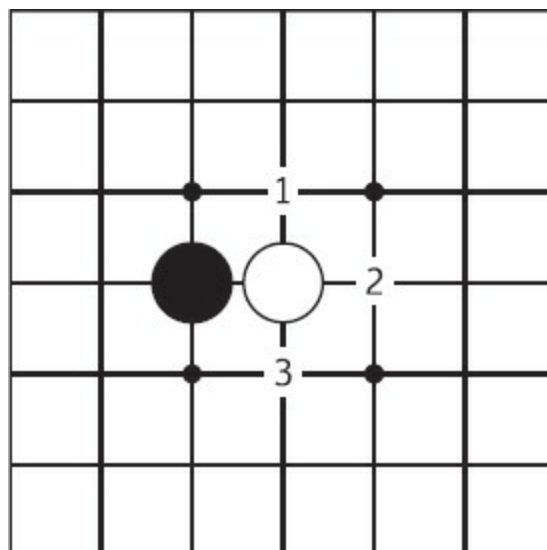
Empecemos con la tercera regla del Go: las piedras son «capturadas» cuando las rodean por todos los flancos las del oponente.

Cuando colocas una piedra en una intersección, las intersecciones adyacentes a esa piedra se llaman libertades. Imagina a soldados en un campo de batalla: si en las cercanías hay un lugar seguro para refugiarse, los soldados no pueden ser capturados.

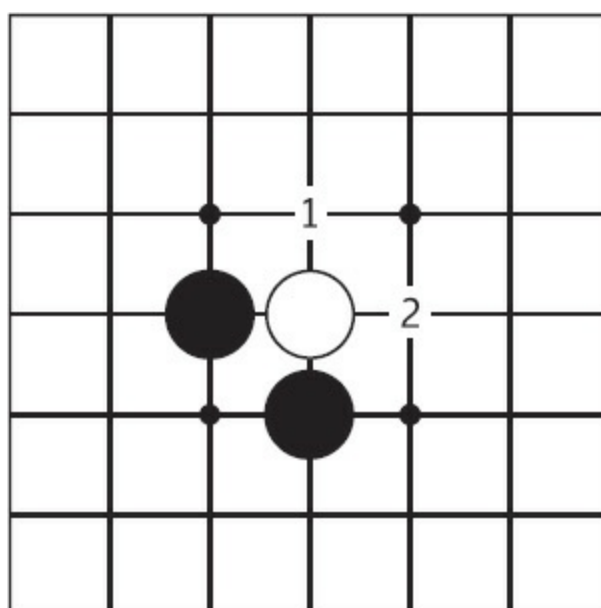
El máximo número de libertades que una piedra puede tener es cuatro:



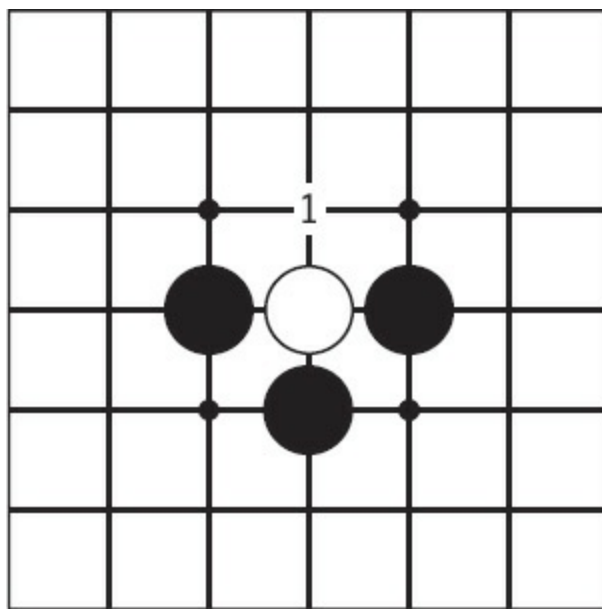
Las libertades se reducen cuando el oponente ataca al colocar alguna piedra inmediatamente adyacente a la tuya. Este ataque reduce a la piedra blanca a tres libertades:



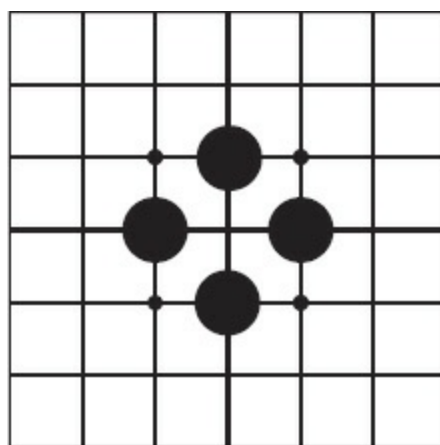
Otro ataque reduce las libertades de la piedra blanca a dos:



¡Cuidado, blancas! Solo hay una libertad restante:



Esta situación se conoce con el nombre de *atari*^[104]. Si las negras eliminan la última libertad de las blancas, la piedra blanca es hecha prisionera y se retira del tablero:

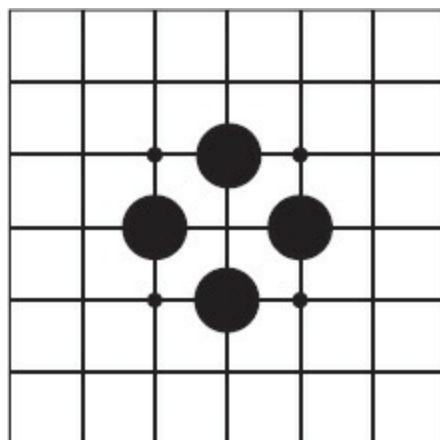


Hasta ahora todo es bastante sencillo. Si no quieres que tus piedras sean capturadas, necesitas asegurarte de tener tantas libertades como sea posible. Si deseas capturar las piedras de tu oponente, acaba con sus libertades.

Nada de suicidios, por favor

Esta es una buena oportunidad para abordar la cuarta regla: está prohibido jugar una piedra que será inmediatamente capturada («suicidio»).

Veamos de nuevo la última situación:



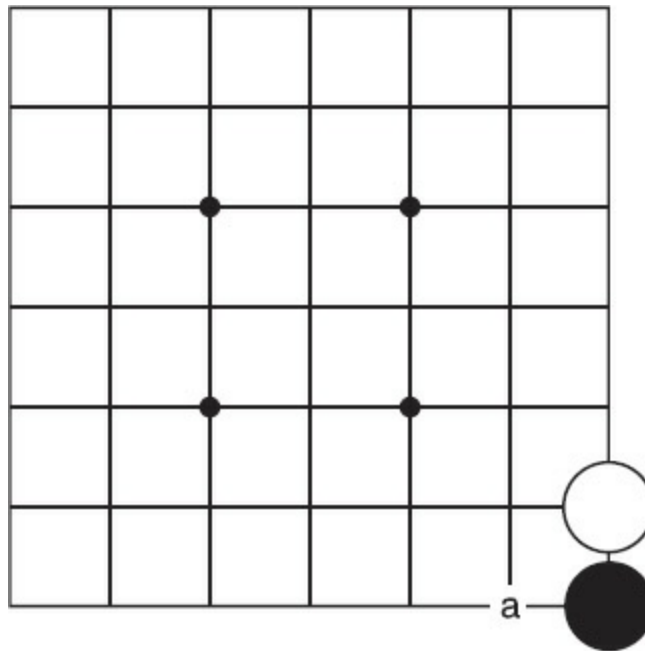
Esta forma se llama *ponnuki*, y es muy común en el Go. El espacio entre las piedras negras no tiene libertades, de modo que si las blancas juegan ahí, la piedra sería capturada inmediatamente. La cuarta regla prohíbe movimientos «suicidas» como este.

Esta regla cobra importancia al analizar grupos mayores de piedras, así que volveremos pronto a este asunto.

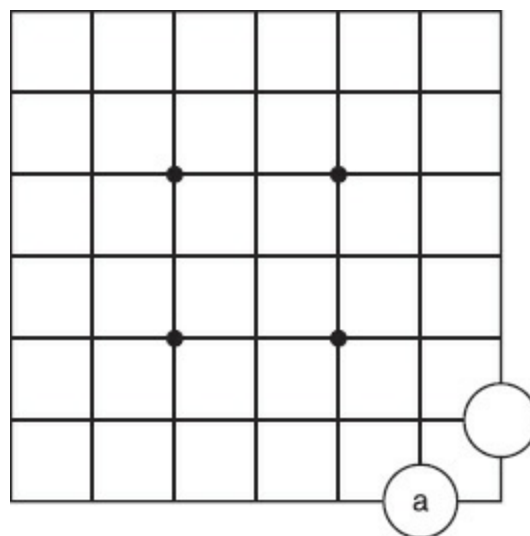
Arrinconarse

En medio del *goban*, las piedras suelen tener muchas libertades hasta ser atacadas. En los costados y en las esquinas del tablero hay menos libertades puesto que hay menos intersecciones en los alrededores.

En el siguiente diagrama, las negras tienen solo una libertad restante: la intersección marcada con la letra «a»:



Las negras están en *atari*. Si las blancas juegan en el punto «a», las negras serán tomadas prisioneras:



Este ejemplo ilustra la diferencia entre las esquinas y el centro del tablero. En las esquinas solo se requieren dos piedras para capturar. En medio, se requieren por lo menos cuatro. En los costados, se requieren tres piedras.

Como resultado, suele ser más fácil establecer y defender territorio en las esquinas del tablero. Los bordes son un poco más difíciles y la parte central

es la más difícil de todas, puesto que se requiere de muchas piedras para atrapar al oponente.

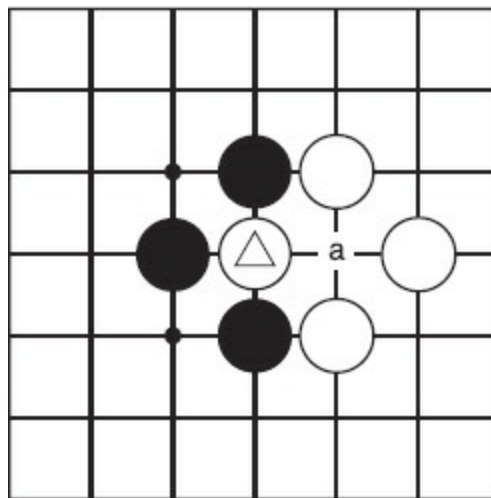
Es por eso que, si miras el principio de los juegos de Go, los jugadores avanzados siguen un patrón general. Primero, establecen posiciones en la esquina del tablero. Una vez que ese territorio está seguro, se ramifican a los costados. El centro del tablero es la última prioridad y se reserva para las partes posteriores del juego. Por eso Shusaku esperó hasta el movimiento 127 para jugar el Enrojecedor de orejas.

Al infinito... pero no más allá

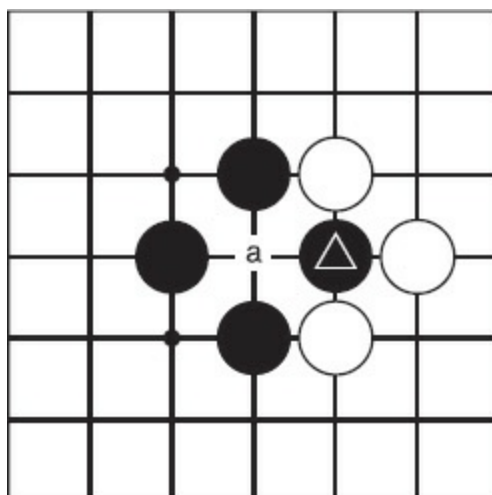
Ahora que conocemos las reglas de captura, examinemos la quinta regla: repetir la misma secuencia de movimientos una y otra vez en un ciclo infinito está prohibido.

Esta situación se conoce con el nombre de *ko*, término japonés para denotar «infinitud». Los ciclos o bucles infinitos son posibles en el Go, y si se permitieran arruinarían la mayor parte de los juegos.

Aquí les presento una ilustración de *ko*. Las negras pueden capturar la piedra blanca, marcada con un triángulo, si juegan en el punto «a»:



¿Te fijaste qué sucedió? Tenemos el mismo patrón, solo que invertido. Ahora, si las blancas juegan en «a», pueden recuperar la piedra:



Sin una regla que evitara el *ko*, las blancas y las negras podrían luchar por esta piedra hasta el fin de los tiempos. Por eso existe la regla: en una lucha *ko*, el jugador que pierde la primera piedra no puede recuperarla hasta pasado al menos un turno, previniendo el bucle infinito.

En los torneos, suele haber una regla adicional llamada «súper ko». Imagina que hay dos o más luchas *ko* en el tablero. Los jugadores pueden pasarla en batallas *ko*, inmersos en un ciclo infinito mayor. Una vez que se ha establecido un patrón, entra en juego el «súper ko», y los jugadores tienen que hacer otra cosa o el juego termina.

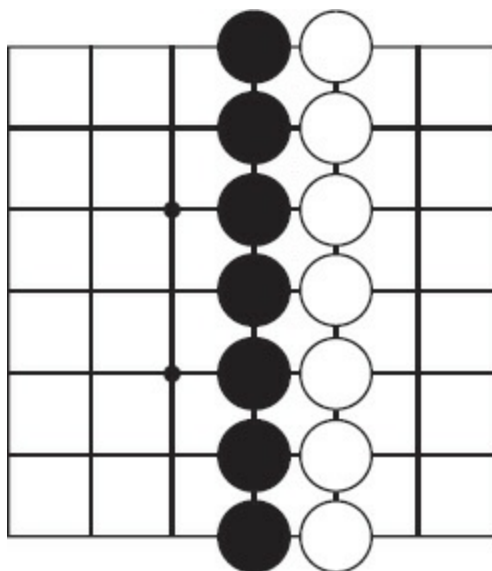
Condiciones para la victoria

Ahora veamos cómo termina el juego. He aquí la regla 6: el juego termina cuando los jugadores se quedan sin piedras, cuando uno concede el juego al rival o ambos pasan.

Sencillo. ¿Cómo determinamos quién gana?

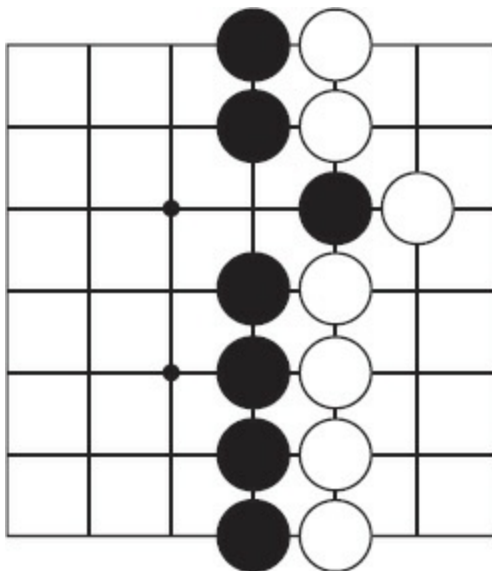
Esa es la regla siete: gana el jugador que rodea mayor territorio en el tablero al final del juego.

Veamos un ejemplo. En un tablero de 7×7 hay 49 intersecciones. Supongamos que el tablero está repartido de la siguiente manera:

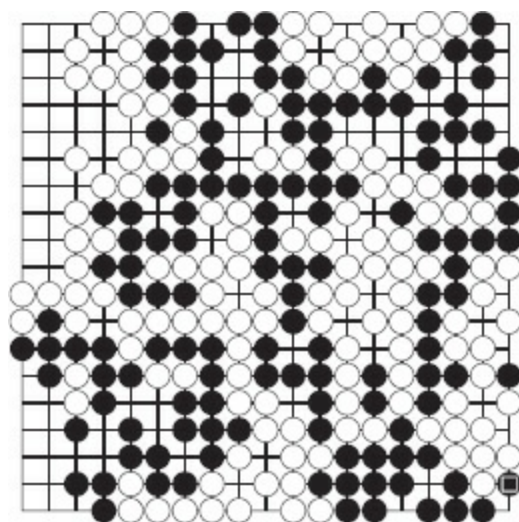


Las negras controlan el lado izquierdo del tablero, lo que da un total de 28 espacios territoriales. Las blancas controlan el lado derecho, lo que equivale a 21 espacios. Ganan las negras.

He aquí una variante: las negras se las han arreglado para invadir un poco el territorio blanco. Las negras tienen ahora 29 espacios y las blancas 20:



El conteo se complica cuando el tablero es mayor y los grupos se distribuyen por toda su extensión, como al final del Juego de las Orejas rojas:



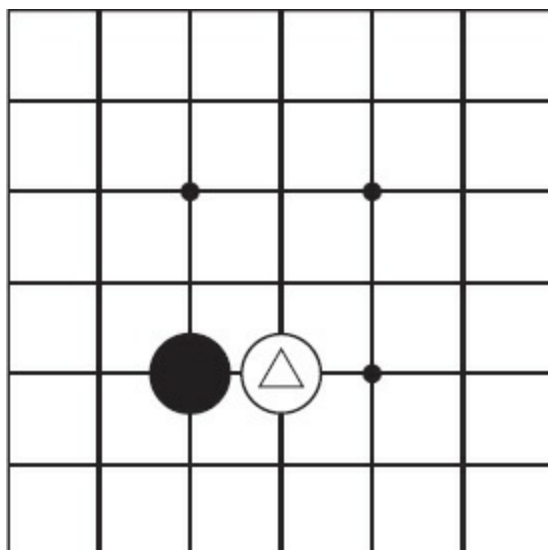
Este es el principio general para contabilizar: si un territorio está en disputa, ningún bando obtiene puntos por él. En ocasiones, los jugadores avanzados concederán territorio al oponente sin jugar a tope, de modo que cuente como capturado. Si hay disputa sobre quién controla tal o cual territorio al final del juego, este se reanuda hasta resolver la disputa.

Estimar el territorio es una habilidad que requiere práctica.

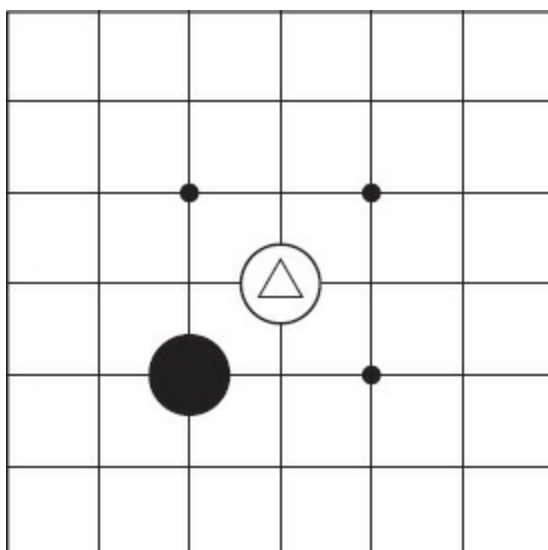
¡En guardia!

Eso cubre las reglas básicas. Ahora, ¿qué hay de las tácticas más comunes?

Lo primero es saber cómo luce un ataque. Los ataques directos quitan una libertad a alguna piedra de tu oponente. En este caso, la negra ataca a la piedra blanca marcada con un triángulo:



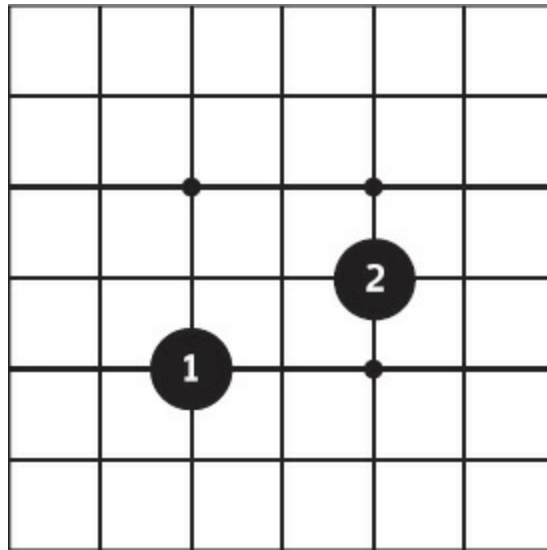
También puedes atacar indirectamente al colocar una piedra cerca de tu oponente. Los ataques indirectos son una manera de fortalecer tu posición en una zona antes de comenzar el asalto directo. En este caso, la negra amenaza a la piedra blanca marcada con un triángulo:



La carga de los caballos

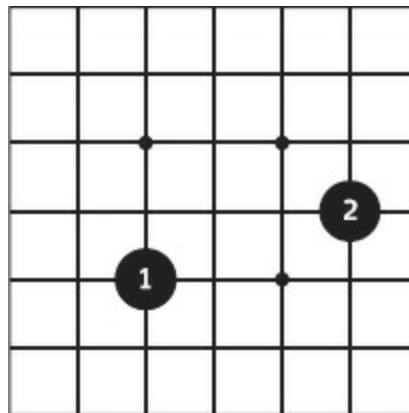
Los ataques son el elemento básico de la ofensiva. Estos elementos se llaman *conexiones* y *uniones*. Las jugadas que equilibran la ofensiva y la defensa se llaman *aproximaciones*.

Una aproximación común es el llamado «movimiento de caballo», que deriva su nombre del movimiento del caballo en el ajedrez:



El movimiento de caballo equilibra la ofensiva y la defensa: se está cerca de otras piedras, así que es fácil conectar de ser necesario, pero extiende tu influencia más que un movimiento sencillo.

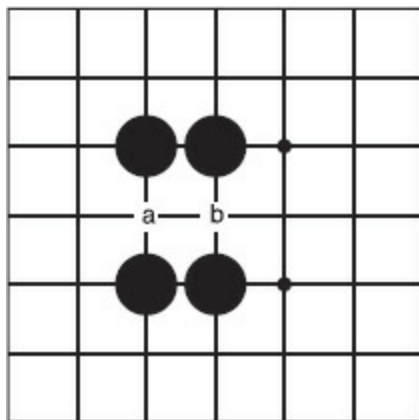
También existe un movimiento de caballo que resulta aún más agresivo:



Este movimiento cambia la defensa por influencia: extiende el dominio del jugador, pero será más difícil defender la piedra en caso de ser atacada.

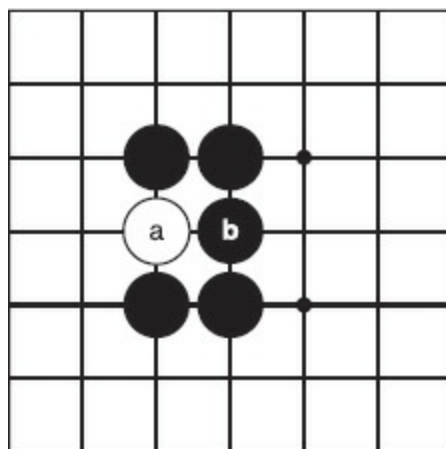
El bambú es más fuerte que el acero

En lo que a defensa estratégica se refiere, hay una disposición llamada unión bambú, y es muy sólida. Se ve así:

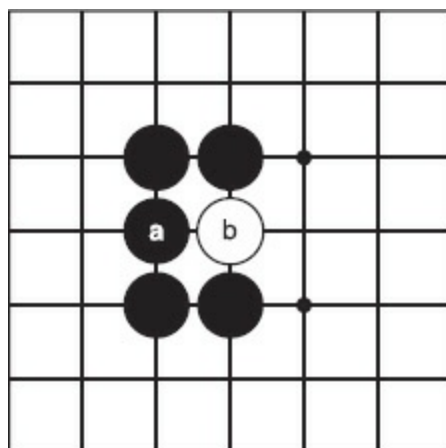


Las uniones bambú son muy fuertes porque, sin importar qué tan agresivo sea el ataque de tu oponente, es fácil conectar tus piedras para conformar una formación fuerte.

En este caso, si las blancas atacan en «a», las negras pueden conectar en «b»:



Igualmente, si las blancas atacan en «b», las negras pueden conectar en «a»:



Una vez que las negras conectan en la unión bambú, se requieren nueve piedras más para que las blancas capturen el grupo. Es difícil vencer este nivel de protección.

Bastante fácil: atacar, defender y capturar parece relativamente sencillo. Además, los movimientos de caballo y las uniones bambú son tácticas fáciles de recordar y comprender.

¿Templado en mil batallas?

Ahora que conozco las reglas y los fundamentos estratégicos, elijo. ¿Debo mejorar mi juego enfrentándome a oponentes humanos, o mejor estudio el juego en libros resolviendo gradualmente problemas?

Los jugadores expertos de Sensei Library tienen opiniones encontradas. Varios de buen nivel invocan a Sun Tzu, autor de *El arte de la guerra*, en el sentido de que «es mejor ser templado en mil batallas». Otros prefieren estudiar problemas de Go y leer libros de táctica y estrategia como la mejor manera de mejorar al principio.

No sé qué pensar. Hay un club local de Go que se reúne cada miércoles, pero lo hace temprano por la tarde, lo que entra en conflicto con mi horario. Me encantaría jugar, pero por el momento no es opción.

Existe otra posibilidad: jugar Go en los servidores de Internet. Jugadores de todo el mundo se congregan en línea para jugar, y los servidores están disponibles 24 horas al día, siete días a la semana. Mejor todavía, estos juegos se califican, así que puedes estimar tu fuerza si juegas lo suficiente. No se trata de un *ranking* oficial de torneo, pero es algo.

Decidí checar los dos destinos más populares de Internet relacionados con el Go: IGS (Internet Go Server) y KGS (Kseido Go Server). Jugar es fácil: creas una cuenta con tu dirección de correo electrónico, bajas un programa y te conectas. Después puedes retar a jugadores o ver los juegos de otros en tiempo real.

Abrí cuentas en ambos sistemas y luego busqué a mi primer oponente real. Dado que apenas había realizado mi registro, carecía de categoría, así que invité a alguien que tampoco lo tenía. Aceptó y el juego comenzó.

Puesto que ninguno de los dos la teníamos, no había piedras de ventaja. Jugué con las negras y considero que hice una sólida apertura. Abarcamos por turnos territorio en las esquinas y luego en los costados. Pasadas unas 30 jugadas, me sentía bastante bien.

Entonces las blancas atacaron. No me preocupaba: mi territorio lucía seguro, así que conecté mis piedras para fortalecer mi posición.

Lo que había estudiado hasta entonces no me preparó para lo que vendría. Jugada a jugada, las blancas me destruyeron.

El territorio que creí seguro fue invadido de modo brutal. Las piedras que, según yo, estaban seguras, fueron tomadas por el enemigo.

Traté de contraatacar y mis avances fueron bloqueados; intenté defenderme, pero fallé.

Busqué ganar terreno pero, tras una hora de juego, mi situación era desesperada. Cedí la partida.

Pasado el juego, pregunté a mi oponente cuánto tiempo llevaba jugando.

«Años», dijo. «Acabo de crear una cuenta en el servidor, y por eso no tengo *ranking*».

¡Ahora sé por qué me destruyó! Le comenté que era nuevo en este juego y le pedí consejo para mejorar.

«Mira los juegos de 10 a 20 kyu. Aprenderás mucho».

Agradecí a mi oponente haber jugado conmigo y me desconecté.

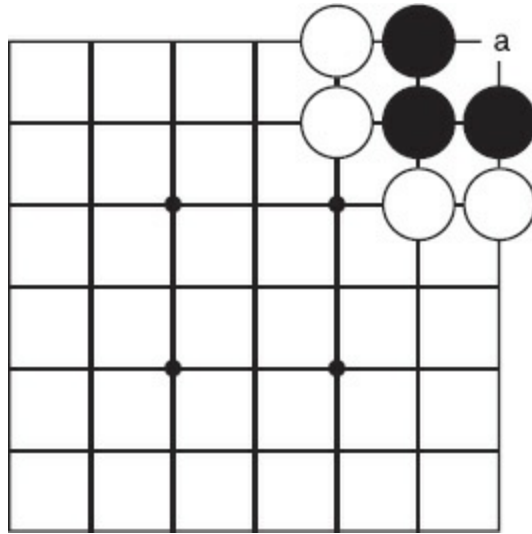
¿Qué había marchado mal?

Al revisar el juego, pensé que estaba creando grupos de piedras defendibles, pero me equivoqué. Para entender por qué habían capturado a mis conjuntos de piedras, debía estudiar cómo se atacan los grandes grupos.

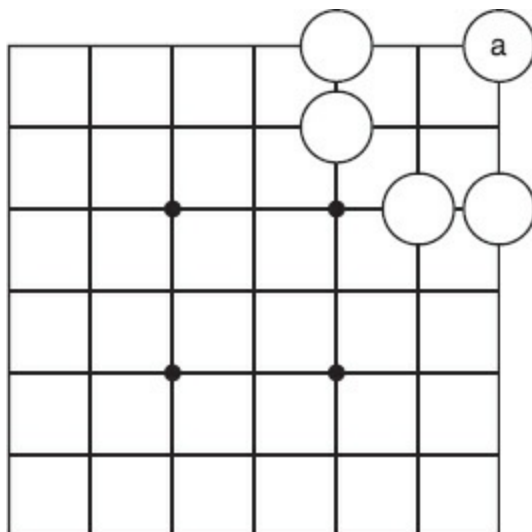
¡Un ojo malo!

¿Recuerdas la regla contra los movimientos suicidas? Contiene un detalle importante: si una jugada suicida captura una o más piedras del oponente, de manera que se creen nuevas libertades, el movimiento se permite.

Veamos este ejemplo. La intersección «a» tiene cero libertades para las blancas, puesto que está rodeada por las negras:

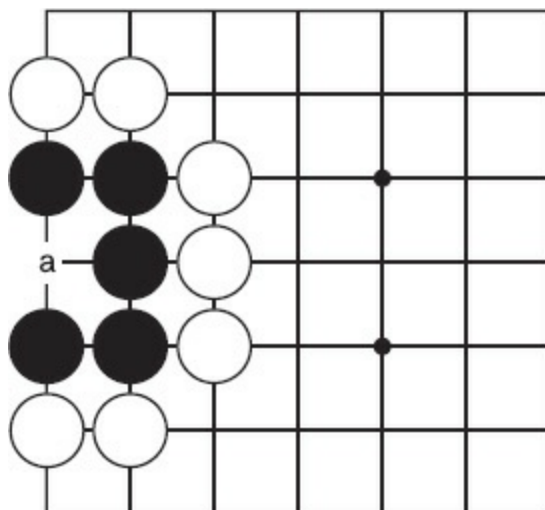


Normalmente, no se permitiría que las blancas jugaran en «a» por la regla del suicidio, pero en este caso, tienen rodeadas a las negras. Sucede que la última libertad de las blancas es «a». Cuando jueguen en «a», el grupo negro es capturado:

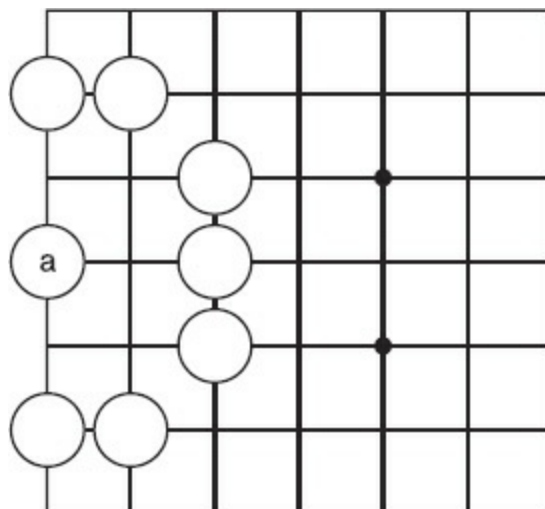


En el Go, a este tipo de espacios se les llama *ojos*. Dependiendo de la situación, pueden ser fuente de fortaleza o debilidad. Si un grupo tiene solo un ojo, es débil.

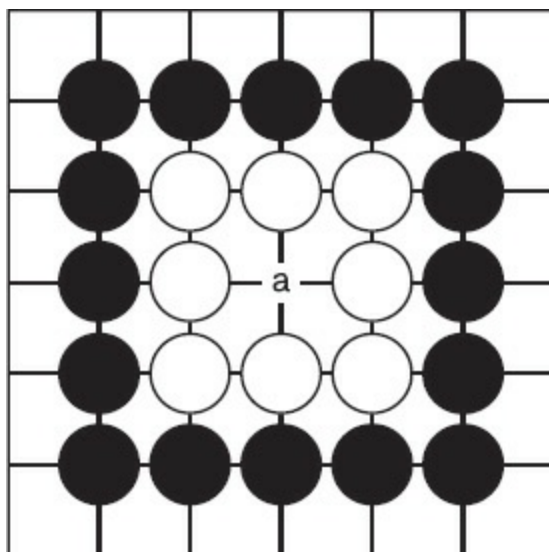
En el siguiente caso, las negras están en problemas:



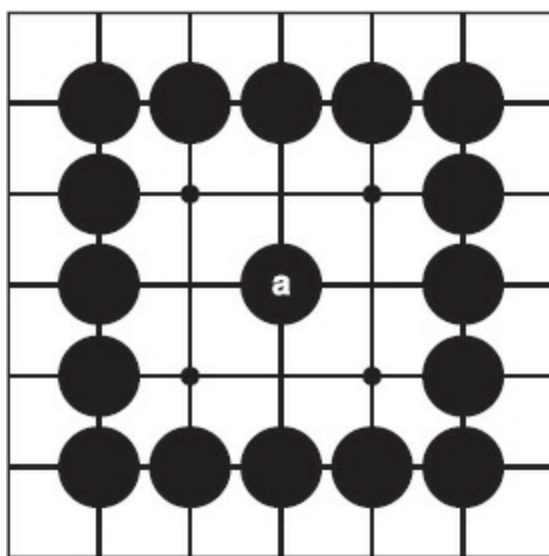
Las blancas capturan en «a»:



Ahora las blancas están en problemas:



Las negras capturan en «a»:



Cuesta un poco de trabajo acostumbrarse a esto. Los grupos con un solo ojo son fáciles de capturar, así que podemos considerarlos fundamentalmente inseguros.

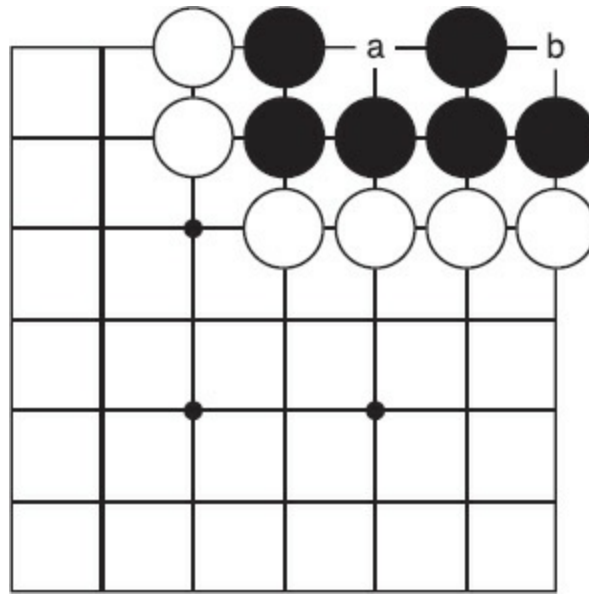
¡Dos ojos buenos!

Esto no significa que los ojos sean malos: son necesarios. Parece raro hasta que analizamos la regla del suicidio otra vez.

La excepción a la regla contra el suicidio solo aplica si la jugada tiene como resultado la captura del grupo opositor. De no ser así, el movimiento queda prohibido bajo la regla del suicidio.

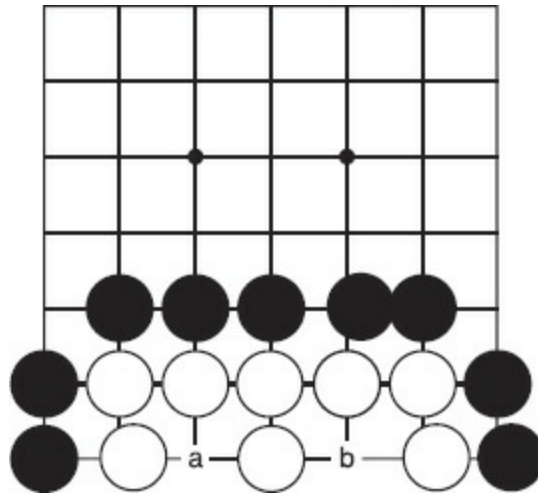
Eso significa que los grupos con dos ojos son invencibles. Es como construir una fortaleza en medio del territorio enemigo: no importa con cuántas piedras ataque el oponente, no podrá capturar al grupo.

En el siguiente ejemplo, las negras tienen dos ojos:

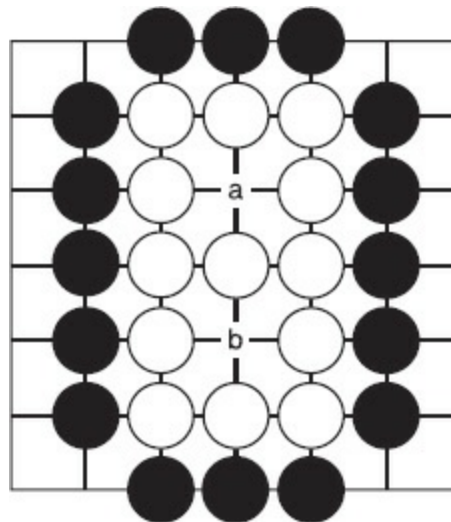


Si las blancas juegan en «a», se trata de suicidio. Si juegan en «b», también. No pueden jugar en «a» o en «b» simultáneamente, así que las negras están seguras, incluso si el grupo está rodeado por todos los costados.

En el siguiente diagrama, un grupo de piedras blancas tiene dos ojos junto al borde:



Aquí, un grupo de piedras blancas tiene dos ojos en el centro:



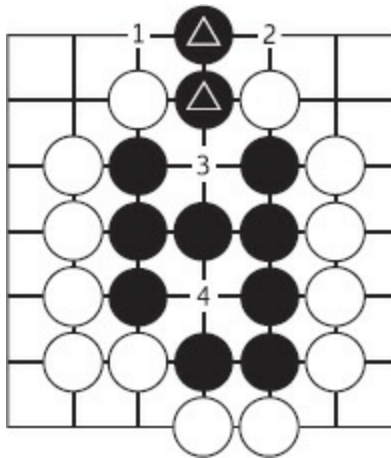
Fíjate en cómo se requirieron seis piedras para formar dos ojos en una esquina, ocho para el costado y 13 para la parte central. Por eso es más fácil defender las esquinas que la parte central del tablero, misma que siempre es difícil asegurar.

El ojo falso (¿de Sauron?)

Si tener dos o más ojos en un grupo lo hace invencible, formarlos tan rápido como sea posible se convierte en una prioridad. Desafortunadamente, no siempre es fácil determinar si una disposición que *parece* que forma un ojo en realidad es un ojo.

Es común que los grupos que forman ojos se vean sólidos, pero a fin de cuentas resultan vulnerables. Estas formaciones se conocen como *falsos ojos* y son mortales.

Aquí va un ejemplo. Parece que el grupo negro tiene dos ojos, pero las piedras de la parte superior, marcadas con un triángulo, son vulnerables:



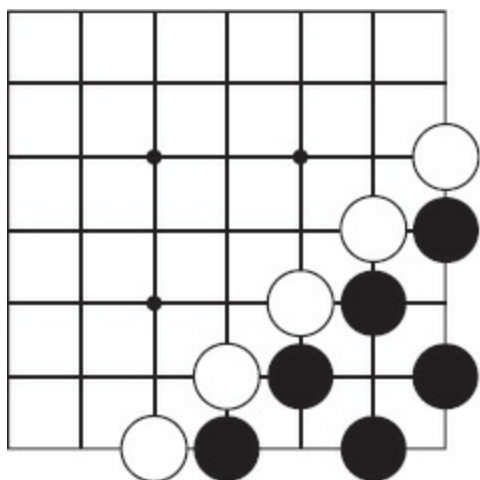
Digamos que toca jugar a las blancas. Estas pueden atacar en 1 o en 2, quitando una libertad. Las negras pueden tratar de escapar por la parte superior, pero en la siguiente jugada las blancas cortarán la ruta de escape. No importa la respuesta de las negras, las blancas capturarán las piedras marcadas jugando en 3. Eso deja vulnerable al resto del grupo negro, que será capturado cuando las blancas jueguen en 4. ¡Uff!

Una buena parte de la estrategia del Go implica crear grupos con dos ojos, evitando los falsos y previniendo que el rival construya ojos reales. También se debe destruir al oponente cuando este cree ojos falsos. Se requiere de mucha experiencia y habilidad para crear verdaderos y para notar cuando son falsos.

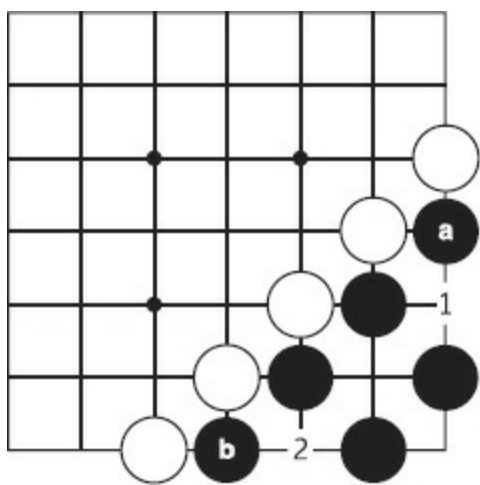
El «desmantelamiento»

Esto nos acerca a comprender la causa de mi primera derrota mayor. Pensé crear territorio seguro, pero en realidad era vulnerable ante el ataque y no me di cuenta a tiempo.

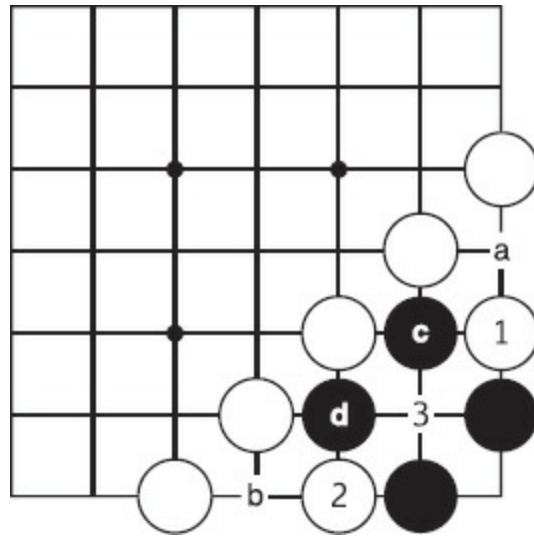
A continuación presento un ejemplo simplificado: fíjate en los ojos creados por las negras. Con cuatro, ese grupo debe ser invencible, ¿correcto?



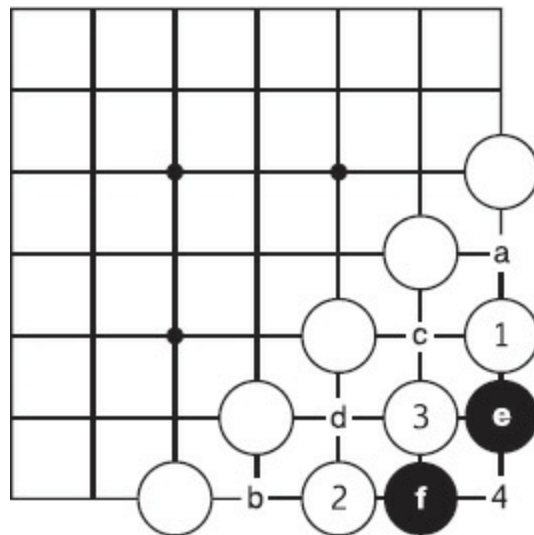
Desafortunadamente para las negras, no es el caso. Si las blancas juegan en 1 o en 2, «a» o «b» serán capturadas:



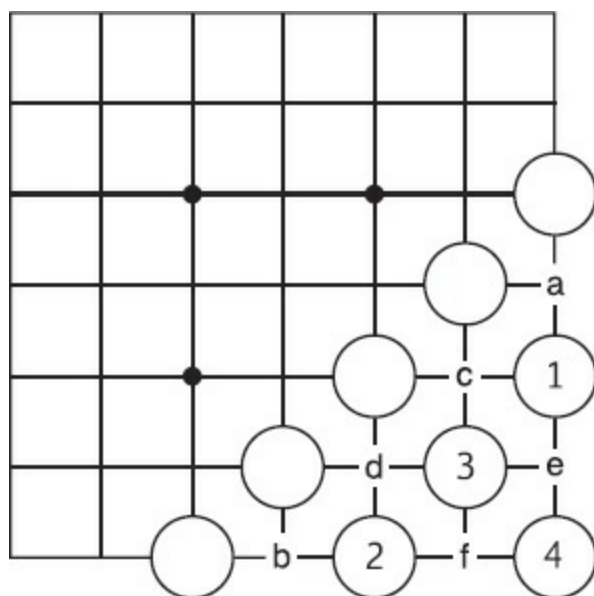
Luego, si las blancas juegan en 3, «c» y «d» serán capturadas:



Finalmente, si las blancas juegan en 4, «e» y «f» serán capturadas:



Como resultado, las blancas desmantelan el «invencible» grupo negro. Nunca se trató de ojos verdaderos y las negras pagaron el precio:



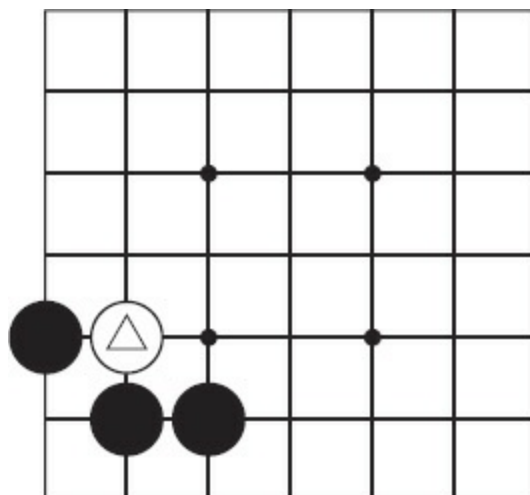
Esto es, en esencia, lo que me sucedió. Pensé que establecía territorio seguro en las esquinas y en los costados, pero no construí dos ojos reales. En consecuencia, mi oponente desmanteló todo lo construido. Lección aprendida.

Serpientes y escaleras

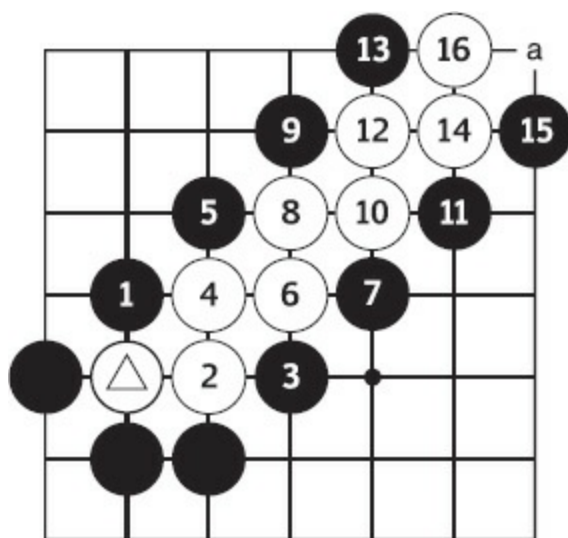
Hay algunos otros conceptos estratégicos que no aprecié hasta verlos en acción.

El primero se conoce como «escalera». Cuando tiene lugar un ataque, el oponente elige contraatacar o huir. Por lo regular, escapar es la mejor opción. Si el oponente va tras de ti, comienza una carrera por la captura.

Veamos un ejemplo. Aquí, la piedra blanca (marcada con un triángulo) está a punto de ser atacada:



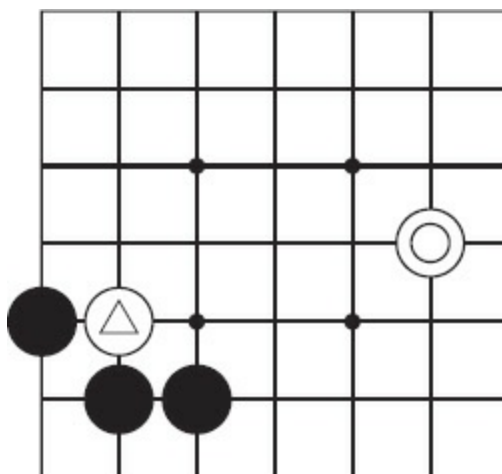
Las negras atacan en 1 y las blancas escapan en 2. Las negras atacan en 3 y las blancas escapan en 4. Las negras atacan en 5 y las blancas escapan en 6 y así. Eventualmente, llegamos al siguiente diagrama:



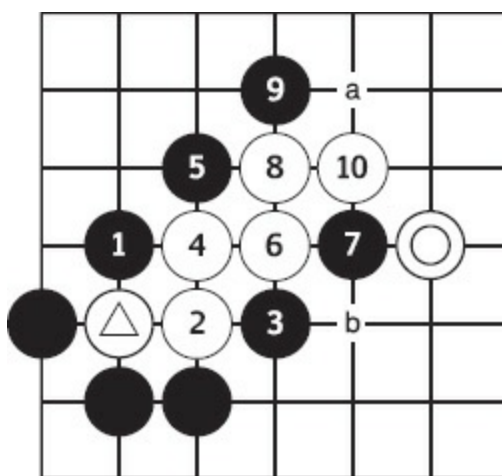
Es el turno de las negras. Las blancas no tienen por dónde huir. Cuando las negras jueguen en «a», todas las piedras blancas serán capturadas y la negras controlarán todo el tablero. ¡El juego terminó!

Hay un viejo proverbio del Go que reza: «Si no sabes qué es una escalera, no juegues». Esta situación ilustra bien por qué el refrán tiene razón.

Ahora, volvamos a examinar la misma situación pero con un pequeño cambio: las blancas tienen una piedra (marcada con un círculo) en la parte derecha del tablero. ¿Qué sucede?



La primera parte es igual: las negras atacan en 1 y las blancas escapan en 2. Las negras atacan en 3 y las blancas escapan en 4. El patrón continúa hasta que las blancas escapan en 10. Llegado ese momento, la situación luce así:



Si las negras continúan el ataque en «a», siguiendo el patrón, las blancas capturarán a las negras en 7, jugando en «b». Si las negras tratan de defender «b», las blancas escapan a la amenaza jugando en «a».

Una sola piedra en el lado opuesto del tablero salvó a las blancas de la aniquilación total. Esto es lo que hace que el Go sea interesante: una piedra que parece desconectada y lejana de la acción principal puede cambiar el curso del juego.

Leer las escaleras es una destreza en sí misma. En un tablero de 19×19 hay muchas maneras de comenzar una carrera de captura. También es fácil encontrarte con que eres objeto de un ataque del que quieres escapar creando

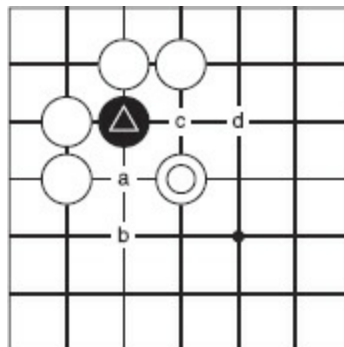
una escalera. En estos casos, necesitas anticipar mentalmente quién ganará la carrera.

Si no piensas que puedes ganar, es mejor sacrificar una o dos piedras para salvar al grupo entero. Si crees que puedes ganar, debes hacer que tu oponente llegue a un juego en que no pueda ganar, dándote una gran ventaja a fin de cuentas.

Lanzar redes

Lanzar una piedra en la ruta de una escalera potencial puede resultar en una muy buena jugada. Igualmente, atrapar a tu oponente al colocar una piedra en una posición que rodea a la del rival puede ser buena estrategia.

Aquí, las blancas quieren capturar la piedra negra (marcada con un triángulo). En lugar de atacarla directamente, las blancas tienden una red que rodea a las negras de manera indirecta.



Si las negras tratan de escapar por «a», las blancas bloquean en «b». Si las negras tratan de escapar en «c», las blancas bloquean en «d». No importa cuánto luchen las negras, jamás escaparán a la red tendida por las blancas.

Aparte del asunto de los ojos, mi oponente fue mucho mejor al provocar que yo me extendiera demasiado. Me encontré en escaleras que no podía ganar o atrapado en redes que ni siquiera noté hasta que fue demasiado tarde.

Todas estas cosas conforman a un buen jugador. Si no conoces las formas básicas como ojos, escaleras y redes, perderás al jugar con personas que saben crearlas y tomar ventaja de ellas.

Maximizar el tiempo de práctica

Después de mi derrota, acepté el consejo de mi rival y traté de ver juegos entre jugadores rankeados entre los kyu 10 y 20. Desafortunadamente, no había muchos que pudiera observar.

Los servidores de Go en Internet no son los sitios más amigables para los principiantes. Los servidores y programas mayores han operado por mucho tiempo, así que las interfaces son de los años noventa, por lo que lleva tiempo aprender a utilizarlas.

Tampoco parece haber muchos jugadores de bajo nivel. Y si no sabes jugar muy bien, ser arrollado una y otra vez no es una experiencia muy agradable. Muchos de los juegos en curso que encontré eran entre jugadores de 8 kyu para arriba.

Traté de observar algunos, pero mi ritmo de aprendizaje era muy lento. Podía seguir las jugadas, pero no las razones por las que los jugadores las hacían. A veces las secuencias de movimientos eran demasiado rápidas y yo no lograba mantener el ritmo del análisis. Sabía que existía un motivo detrás de cada piedra colocada, pero era demasiado inexperto para saber de cuál se trataba.

Los juegos de Go pueden durar más de una hora, lo que constituye otra barrera. Varias veces observé un juego para distraerme mientras ayudaba a Lela o a Kelsey con alguna tarea. Dado que los juegos tenían lugar en tiempo real, no lograba seguir el juego hasta la resolución.

Traté con la idea de «las cien batallas» y, con base en mi experiencia, no fue la mejor manera de mejorar como principiante. Decidí cambiar de estrategia, sustituyendo los devaneos ociosos con estudio estructurado.

El hombre contra la máquina

Cada noche, pasé 45 minutos estudiando problemas de Go; primero, usando el programa SmartGo en mi iPad, y luego trabajando con la serie de libros con problemas en progresión de Kiseido: *Graded Go Problems*.

Después de estudiar los problemas, jugaba una partida contra el programa incorporado en SmartGo, usando un tablero de 9×9 . En principio, SmartGo me arrolló, incluso con ocho piedras de ventaja. ¡Vergonzoso!

Al estudiar, mis juegos comenzaron a mejorar rápidamente. Aprendí a aprovechar mis piedras de ventaja al jugar ofensivamente, atacando a las

blancas rápidamente y formando fuertes conexiones cuando me era posible.

En poco tiempo logré reducir las piedras de ventaja que se me otorgaban a seis, y después a cinco, conforme ganaba juegos contra el programa. Después de 30 juegos me quedan unas cuatro piedras de ventaja.

Estimar el marcador es difícil: a veces pienso que hago las cosas bien pero, cuando verifico el marcador, resulta que estoy perdiendo por un margen significativo. El problema es que para ganar un juego se requiere hacer jugadas redituables que incrementen la cantidad de territorio controlado. Por lo regular, las jugadas que según yo amplían mi territorio, no lo hacen, convirtiendo mi jugada en turno desperdiciado y en una oportunidad para el oponente.

Modos de falla

Además de estudiar problemas de Go, leo los libros que elegí en Sensei's Library. Lo aprendido me ayudó considerablemente.

Estudiar *How Not to Play Go* resultó extremadamente provechoso. De acuerdo con Yuan Zhou, el autor, los principiantes cometen varios errores comunes que les cuestan bastante:

- Los jugadores sin experiencia siguen a su oponente ciegamente. Por ejemplo, cuando son atacados, los jugadores malos responden defendiéndose, escapando o contraatacando en la misma zona, en lugar de considerar todas las opciones.
- Los jugadores novatos no ponen atención al tablero entero. A veces, la mejor jugada no se realiza en el sitio de la acción, sino en otra parte del tablero, lejos de todo. La batalla es emocionante y los jugadores principiantes pueden pasar por alto las oportunidades presentes en zonas más tranquilas del *goban*.
- Los jugadores inexpertos no realizan la mejor jugada posible. El Go requiere pensar en términos de ganancia y costos: cada jugada cuesta una piedra, y siempre querrás obtener lo más posible a cambio de ella. Por lo regular, esto implica sacrificar una o dos piedras, siempre y cuando obtengas un mayor beneficio en otra parte.
- Los jugadores con poca experiencia no valoran el *sente*. En el Go, la iniciativa es un asunto de gran importancia: uno intenta que las

respuestas del oponente sean determinadas por nuestras jugadas tanto como sea posible. En lugar de pensar en la agenda del rival, quieres que tu oponente esté tan preocupado por prevenir pérdidas potenciales que incluso deje para después las ganancias territoriales. Quien obtiene la iniciativa y la mantiene, casi siempre gana el juego.

- Los jugadores inexpertos no son buenos al calcular el territorio. Como resultado, pasan mucho tiempo en batallas locales o asegurando una esquina del tablero, mientras su oponente se apodera de una gran cantidad de territorio no disputado.
- Los jugadores inexpertos envidian la posición del oponente. Zhou llama a este problema «el problema de los ojos rojos» y, por lo regular, resulta en que el jugador hace movimientos malos porque siente que el oponente adquiere demasiado territorio.
- Los jugadores principiantes alimentan grandes esperanzas. Es fácil ser atraído por un grupo de piedras que puedes capturar con dos de las tuyas, olvidando que no se pueden hacer dos movimientos seguidos: tu oponente responderá a la primera jugada. Como resultado, desperdicias piedras preciosas con la vana esperanza de que tu oponente sea lo bastante tonto como para no darse cuenta de lo que haces.

Estos puntos dieron en el clavo en mi caso, particularmente en lo que a las grandes esperanzas se refiere. ¿Cuántas veces he atacado un grupo esperando que mi oponente no se percate de mi plan? ¿Cuántas veces me he metido en peleas locales ignorando el panorama general?

Después de leer *How Not to Play Go*, mi juego mejoró mucho. Una de las cosas que corregí fue la sensación visceral de que capturar las piedras del oponente era el camino más rápido a la victoria. En el ajedrez y en las damas en ocasiones así funciona, en el Go, no.

Capturar piedras es útil, pero no es una condición para la victoria. Asegurar territorio es el objetivo del Go y es común que esto se logre sin capturar piedras enemigas. Me tomó un rato asimilar la lección: atrapar piedras hace sentirse bien, incluso cuando hacerlo resulta contraproducente.

Sabiduría proverbial

Los aforismos son herramientas de enseñanza muy útiles en el Go. Dado que el juego es tan complejo, hay reglas que ayudan a los jugadores a recordar lo conveniente en situaciones complejas.

Uno de los ejemplos más tempranos de esto son «Las diez reglas de oro del Go», que se atribuyen a Wang Jixin, quien jugó en tiempos de la dinastía Tang en China del año 600 al 900 de nuestra era, aproximadamente^[105]:

1. «Los codiciosos no tienen éxito.»
2. «No tengas prisa por entrar en el territorio del oponente.»
3. «Cuídate a ti mismo cuando ataques al rival.»
4. «Sacrifica una piedra con tal de obtener la *sente*.»
5. «Abandona lo pequeño para salvar lo grande.»
6. «Cuando estés en peligro, sacrifica.»
7. «Realiza formas sólidas y evita los movimientos apresurados.»
8. «Cada jugada debe responder a la del rival.»
9. «Juega a la segura cuando enfrentes posiciones sólidas.»
10. «Busca la paz y evita luchar en una situación de debilidad o aislamiento.»

La sabiduría del Go se parece mucho a la lista anterior. Existen incontables proverbios sobre el Go:

- El punto clave del enemigo es también tu punto clave.
- Cuando dudes, *tenuki* (juega en otra zona).
- El *hane* (una piedra que rodea para atacar) contiene la muerte.
- Los principiantes juegan *atari*.
- Un *ponnuki* vale 30 puntos.
- Las redes son mejores que las escaleras.
- Cierra el puño antes de golpear.
- Sacrifica las ciruelas para obtener duraznos.
- Un hombre rico no debe entrar en peleas.
- Si montas un tigre, es difícil elevarse (no empieces algo que no llevarás hasta sus últimas consecuencias).

Hay mucho por aprender. No solo es sabio aprender proverbios y saber qué significan, también debes saber cuándo ignorarlos. Después de todo, uno de los proverbios dice: «No sigas ciegamente los proverbios».

Algunos proverbios se aplican a los principiantes. Mi favorito es: «Solo después del décimo golpe, podrás ver el puño». A este suele seguir el de: «Solo después del vigésimo golpe podrás desviarlos». En mi experiencia eso es muy cierto: los jugadores expertos pueden atacarme de maneras que ni siquiera reconozco.

Tengo mucho que aprender.

Las preguntas de las «cinco piedras»

Tras casi un mes de aprender Go, encontré un ensayo de Sam Bleckley que describía sus experiencias^[106], Sam refiere el consejo de un amigo, quien le sugería hacerse cuatro preguntas sobre la situación del juego antes de colocar una piedra:

1. ¿Puedo asegurar la vida de alguno de mis grupos? (Formando dos ojos, conectando, etcétera.)
2. ¿Puedo aniquilar alguno de los grupos de mi oponente?
3. ¿Puedo asegurar mi territorio?
4. ¿Puedo invadir el territorio de mi rival?

Estas preguntas captan la esencia de las cuatro prioridades del Go: vivir, matar, asegurar e invadir. Al repasar las preguntas en orden antes de realizar un movimiento, ganarás de inmediato «cinco piedras» en fuerza de juego, pues tomarás mejores decisiones y evitarás intercambios poco ventajosos.

Esta es una técnica bastante efectiva. Es fácil memorizar y recordar una lista de cuatro prioridades. Al formularte estas sencillas preguntas antes de colocar una piedra, y al considerar los movimientos que más valor aportan antes que los de ganancia menor, te aseguras de que las decisiones del momento correspondan a las verdaderas prioridades.

El Go requiere adoptar decisiones con información incompleta en ambientes cambiantes, así que las técnicas de este tipo son muy valiosas. Preguntas como estas dan estructura y sentido lógico en ciertas circunstancias. Mejor aún, también identificas y usas las preguntas de las «cinco piedras» en otras áreas de tu vida. Cuando te encuentres en una situación que requiera tomar decisiones con información incompleta, asigna

prioridades y acepta intercambios. Verás que es muy posible que las preguntas de «cinco piedras» te ayuden a tomar la decisión correcta.

Llegando a la marca de las 20 horas

Con base en mis notas, después de casi un mes de estudio, he llegado a la marca de las 20 horas. ¿En dónde estoy y qué logré?

De acuerdo con la base de datos de SmartGo, he resuelto unos 150 problemas clasificados y he progresado hasta los del nivel 18 kyu. He enfrentado 33 juegos y actualmente tengo una ventaja de cuatro piedras contra la computadora.

También terminé de leer el primer volumen de la serie *Graded Go Problems*, clasificado en el rango 30-25 kyu. Voy a la mitad del segundo libro, clasificado entre 25-15 kyu, así que puedo asegurar que mi fuerza de juego ronda los 18-20 kyu. No está mal.

Repaso del método

Hagamos un repaso de la esencia del método utilizado para aprender a jugar Go:

- Aprendí las reglas básicas del juego.
- Compré un goban, piedras, y varios libros sobre la estrategia en el Go para principiantes.
- Encontré varias formas de estudiar los principios estratégicos más importantes al trabajar en problemas clasificados de Go, tanto de la computadora como impresos en libros.
- Concentré mi práctica inicial en resolver problemas y repetir los ejercicios que no podía resolver o explicar en el primer intento.
- Practiqué y jugué hasta llegar a las 20 horas. En ese punto, logré resolver cómodamente problemas diseñados para el 20 kyu, una mejora de 15 estratos a partir del grado de principiante, 35 kyu.

¿Y ahora qué?

Tengo sentimientos encontrados con respecto del Go. He aprendido lo suficiente para apreciar lo complejo que es y lo lejos que estoy de dominarlo.

He progresado bastante desde que empecé, pero para ser en verdad bueno en el juego tengo muchísimo camino por recorrer.

Por otra parte, mi tiempo libre es limitado y el Go parece requerir de la misma intensa concentración que demandan la escritura y la programación. En tanto que el Go puede ser divertido al final de un largo día, se parece demasiado al trabajo.

Además, los juegos en tablero completo de 19×19 pueden requerir más de una hora para completarse, y no es muy correcto abandonar una partida contra un rival para atender asuntos familiares. Hasta ahora, la mayoría de mis juegos los he realizado contra la computadora, puesto que siempre está disponible y no le importa si interrumpo el juego para hacer otra cosa.

He llegado a apreciar las dificultades intelectuales de los problemas del Go, así que probablemente siga resolviéndolos cuando esté de humor para escrutar acertijos. Me gustan mucho más los ejercicios del Go que los crucigramas, el sudoku u otros juegos semejantes, así que mis programas de Go permanecerán en mi teléfono, listos para cuando tenga algunos ratos libres.

Planeo enseñar a Lela cómo jugar en cuanto sea un poco más grande. Las reglas son lo suficientemente simples para que las comprenda una criatura de cuatro años, y aún conservo gratos recuerdo de las ocasiones en que jugaba ajedrez con mi papá. El Go es una buena manera de enseñar destrezas importantes como análisis, estrategia, intercambios, y pienso que nos divertiremos mucho jugando Go en años venideros.

Fuera de eso, ya no tengo deseos de invertir más tiempo tratando de alcanzar maestría en el Go. Ya aprendí lo suficiente para satisfacer mi curiosidad y progresar en él no es prioritario.

Eso está bien. No existe ninguna ley universal que diga que debes alcanzar maestría en todo lo que aprendes. La vida requiere de compensaciones y nada malo tiene explorar algo nuevo, aprender mucho y luego decidir explorar otra cosa.

No tienes por qué ser cinta negra en todo para tener una vida satisfactoria.

8

Ukulele

Lección: aislar, practicar, repetir.

*Hoy, al igual que en cualquier otro día, nos despertamos vacíos y temerosos.
No abras la puerta del estudio para empezar a leer. Toma un instrumento
musical. Deja que la belleza que amamos se convierta en lo que hacemos.
Existen cientos de formas de arrodillarse y besar el suelo.*

Rumi, poeta persa del siglo XIII.

Para obtener imágenes, videos y comentarios adicionales sobre este capítulo, visita <http://first20hours.com/ukulele>.

Hace unos cuantos años, leí dos libros muy interesantes escritos por Tom Hodgkinson. Se llamaban *How to Be Idle* (2005) y *The Freedom Manifesto* (2007). En pocas palabras, el autor afirma que trabajamos demasiado y que nos beneficiaríamos si recortáramos un poco la actividad laboral. En lugar de llevarnos al límite, aprendamos a relajarnos y a no tomar todo con demasiada seriedad.

Los libros son, en principio, una llamada de atención para ignorar la cultura del consumo y el «carrerismo». En lugar de perseguir la sobreabundancia material, Hodgkinson recomienda contentarnos con medios modestos y realizar las actividades de tiempos más sencillos, como jardinería, cocina y tocar música.

En *The Freedom Manifesto*, Hodgkinson recomienda aprender a tocar el ukulele:

Esta maravilla de cuatro cuerdas es barata, portátil y fácil de tocar. En estos sentidos, es todavía más práctica que la guitarra. He aquí los tres acordes que debes tocar en la mayoría de las canciones: do, fa y sol. Compra un ukulele y nunca estarás aburrido.

El consejo de Hodgkinson encontró eco en mí. En esa época, Kelsey y yo vivíamos en un estudio-apartamento de 31 metros cuadrados, en la zona noreste de Nueva York. Trabajábamos mucho tratando de progresar en nuestras carreras y pagando esas locas rentas de Manhattan. Cualquier diversión era bienvenida, y aprender a tocar el ukulele sonaba muy divertido.

En la preparatoria, me involucré en todas las actividades musicales que la escuela ofrecía. Ya mencioné que aprendí a tocar la trompeta, pero también canté en el coro y me gustó bastante. Sin embargo, cuando me gradué para asistir a la universidad, dejé toda actividad musical y el libro de Hodgkinson me recordó lo mucho que la extrañaba.

Parte del problema es que las trompetas *suenan muy fuerte* y no son divertidas tocadas a solas. Estaba bastante ocupado con otras cosas y no me interesaba unirme a una banda. Además, no se puede cantar y tocar la trompeta al mismo tiempo, como se puede hacer en el caso del piano, la guitarra y el ukulele.

Después de investigar un poco, terminé comprando un ukulele de concierto OU5, marca Washburn Oscar Schmidt. Está hecho con madera de Koa hawaiana, suena bien y el precio era razonable: entre el ukulele y el estuche, gasté unos 175 dólares.

Cuando lo tuve entre mis manos, jugué un poco con las cuerdas, pero la verdad es que nunca había tocado un instrumento de cuerdas, a no ser por un

breve experimento con una guitarra barata en la universidad. Sin una buena idea de cómo comenzar, otros asuntos cobraron importancia y el ukulele quedó en el armario acumulando polvo junto a la guitarra.

Un héroe del ukulele

Mi interés por tocarlo revivió cuando Kelsey y yo dimos la bienvenida al mundo a Lela. Es sorprendente cómo la llegada de un niño puede cambiar tus prioridades en un abrir y cerrar de ojos. De pronto, trabajaba menos para pasar más tiempo en casa y volví a pensar en la música.

Uno de mis mejores amigos, Nate Siebert, toca la guitarra para sus hijos Jackson y Finley cada noche antes de ir a dormir. Los hijos adoran esta práctica y, con el paso de los años, llegó a convertirse en un ritual tranquilizador.

Como beneficio adicional, es muy probable que la música resulte buena para ellos: la exposición temprana a la música se ha relacionado con el desarrollo cognitivo. No está muy claro cómo ayuda la música a que los pequeños cerebros se desarrollen, pero es una realidad^[107].

Me gustaría que Lela creciera en un hogar en que se toca música regularmente, y haré todo lo posible para convertirla en una niñita feliz e inteligente. De pronto, aprender ukulele me pareció una muy buena manera de invertir mi tiempo.

Les presento mi nuevo instrumento

Ya tenía un ukulele, pero otro instrumento me llamó la atención. Resulta que el primo de Kelsey, Erik Smith, es un laudero autodidacta que construye instrumentos de calidad. La empresa de Erik, Crow Hill Guitars^[108], se especializa en guitarras eléctricas y acústicas hechas a mano al gusto del cliente. Erik las exhibe en festivales musicales por todo Estados Unidos y decidió probar suerte construyendo su primer ukulele tenor. Les presento al Grizzly:



Este ukulele es único: el cuerpo está hecho con wenge y caoba africanos, con incrustaciones de cedro español, palo de rosa indio, secoya rizada y maple, también rizado. En el diapasón se incrustó madreperla blanca y también concha de abulón para adornar el agujero frontal. Quise tener el Grizzly en cuanto lo vi.

Pero había un problemita: los instrumentos de Erik cuestan miles de dólares. Valen cada centavo en términos de calidad, pero debía decidir si mi interés era tan serio como para justificar el gasto.

Decidí comprar el Grizzly por dos razones. Primero, tocar un instrumento de alta calidad, incluso como principiante, te da una importante ventaja psicológica: suenas mejor desde el principio. Comparado con el Grizzly, mi viejo ukulele suena bastante aburrido. Como resultado, tocar el Grizzly es mucho más divertido.

En segundo lugar, sería muy penoso tener en casa un instrumento de esa calidad sin tocarlo. Además, si no aprendo a tocar, habré desperdiciado una buena cantidad de dinero. Habiendo invertido tanto en el probable resultado, tengo más posibilidades de reservar tiempo para practicar, haciendo que la compra se convierta en un buen ejemplo de cómo usar el compromiso para cambiar una conducta.

El equipo adicional

Junto con el Grizzly, compré otras dos piezas importantes de equipo:

- Un afinador para ukulele Snark SN-6. Este económico afinador electrónico se prende a la cabeza del ukulele, logrando que cada cuerda quede afinada en el tono correcto.
- Cuerdas para ukulele tenor marca D'Addario J71 Pro-Arte. Se trata de cuerdas de nylon extraduras para guitarra clásica que dan al ukulele un sonido más pleno y hacen más fácil el rasgueo rápido^[109]. Si se me rompe una cuerda, deberé reemplazarla antes de continuar, así que vale la pena tener varios juegos extra, por si acaso.

Además del equipo, compré tres libros sobre ukulele:

- *Absolute Beginners Ukulele, Omnibus Edition*, por Steven Sproat (2012), un tutorial para principiantes.
- *Fretboard Roadmaps-ukulele*, por Jim Beloff y Fred Sokolow (2006), que cubre técnicas avanzadas.
- *The Daily Ukulele*, por Jim Beloff y Liz Beloff (2012), libro de referencia que contiene una amplia variedad de canciones que van desde lo tradicional hasta el *rock* clásico.

Un reto intrigante

Poco después de comprar el Grizzly, recibí un correo de mi amigo Chris Guillebeau, autor de *The \$100 Startup* y *The Art of Non-Conformity*. Cada año, Chris organiza una conferencia muy divertida llamada Encuentro para la Dominación Mundial. Imagina lo que es reunir a unas mil personas que están haciendo cosas raras/novedosas/interesantes en los mismos rubros. Eso es el EDM.

Chris originalmente me había invitado a hablar de mi libro *The Personal MBA*, pero 10 días antes del encuentro otro orador canceló su participación intempestivamente. Chris sabía que yo estaba trabajando en este libro, así que me envió una nota: ¿podría yo dar una segunda conferencia sobre la rápida adquisición de destrezas?

Claro... ¿Por qué no?

El Encuentro para la Dominación Mundial estaba por convertirse en el lugar de estreno de estas ideas en público.

Esbozaba yo la nueva presentación (que se trataba de las ideas abordadas en los capítulos 1 a 3), cuando recibí otro correo electrónico de Chris que enviaba a todos los oradores antes de la reunión. Entre otras cosas, su nota contenía el siguiente consejo: «Si puedes hacer algo especial, hazlo... Todo lo que puedas hacer para sorprender [a los participantes] destacará y llegará lejos».

Eso me puso a pensar... tocar el ukulele sería una buena forma de demostrar cómo funcionan estas ideas. ¿Podría aprender a tocar en 10 días?

Sería difícil, pero pensé que podría hacerlo. Respondí aceptando la invitación.

Oficialmente tengo 10 días para aprender a tocar el ukulele.

Loco por ti

Faltando solo 10 días para mi primera presentación en público, debía elaborar un plan y rápido. No es el tipo de compromiso del cual puedo disculparme fácilmente, y será la primera vez en 12 años que haga algo musical en público.

Después de aceptar, la parte de mi cerebro que desea que yo quede bien frente a los demás, naturalmente entra en acción. ¿Cómo se me ocurre? ¿Estoy loco?

Lo que hace a esta idea todavía más estimulante es que estaré solo: solo yo con el Grizzly. No tendré una banda que me apoye, y si cometo un error, todos lo sabrán. Si lo hago horriblemente, todos verán y escucharán lo malo que soy.

Pensar en esa posibilidad me puso nervioso. Estoy considerando hablar, usar un instrumento que jamás he tocado antes y hasta quizá cantar frente a desconocidos. ¡Vaya!

Razones me sobran para saber qué he de hacer antes de trepar al escenario. Tengo mucho por hacer y el reloj no se detiene.

Definir el éxito

La primera parte de mi objetivo en cuanto a nivel de desempeño está definida: tocaré el ukulele en público en 10 días. Es un buen comienzo, pero ¿cómo ser más específico?

Saqué un cuaderno y esbocé cómo me gustaría que se viera la sesión. Tengo una hora para hablar. Si logro exponer las ideas principales en unos 30 minutos, eso me deja 20 para tocar y 10 para responder preguntas.

Haré planes para tocar unos 20 minutos. ¿En qué debo concentrarme?

La conferencia no es un concierto. Se trata más bien de una enseñanza, así que quiero concentrarme en usar lo que toque para ilustrar algunas ideas interesantes e importantes sobre la rápida adquisición de destrezas. También me gustaría destacar algunas cuestiones interesantes sobre música.

Al esbozar, comienzo a tener ideas... esto va a ser *divertido*.

La «Canción de los cuatro acordes»

Mientras pensaba en qué tocaría, recordé un video gracioso de una banda que vi varios meses antes.

Axis of Awesome^[110] es una banda humorística de *rock* en Sídney, Australia. El grupo es conocido mayormente por la «Canción de los cuatro acordes», un popurrí de éxitos del pop pasado y presente, visto 24 millones de veces en YouTube.

La gracia que demuestra la banda es que prácticamente cualquier canción popular de décadas pasadas está compuesta por los mismos cuatro acordes. Toca los acordes una y otra vez, sobrepón algunas letras al azar y logras componer un hit de la música pop.

Ningún artista se salva: Elton John, Los Beatles, John Denver, U2, los Red Hot Chili Peppers, Bob Marley, Beyonce, Lady Gaga. Hasta la venerable canción tradicional australiana «Waltzing Matilda» aparece aquí. La canción dura cinco minutos, introduciendo un nuevo éxito cada cinco a 10 segundos, acompañado por risas y aplausos del público.

La «Canción de los cuatro acordes» es simultáneamente histórica y alucinante. Al mostrar la estructura de canciones conocidas, muestra que la música popular tiene una fórmula. Existen muchas variantes posibles del tema, pero la mente humana parece atraída a ese método en particular para construir una tonada pegajosa.

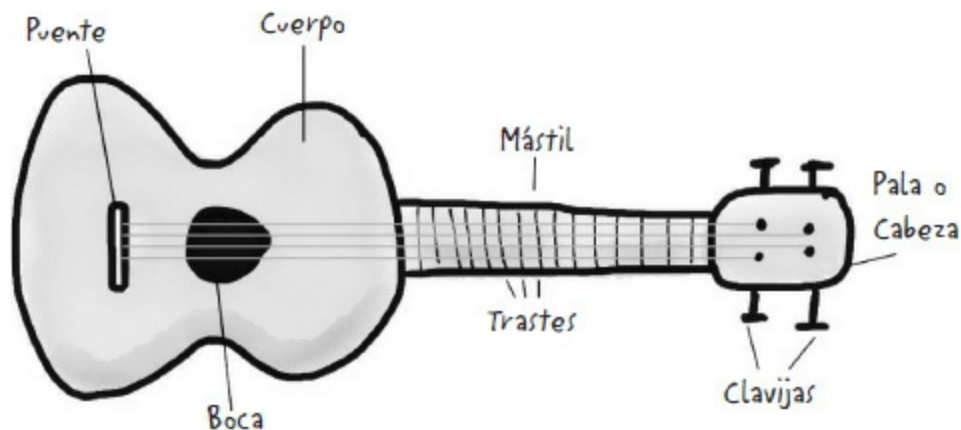
Ahora te presento los acordes de la «Canción de los cuatro acordes»:

Sol/Re/Mi menor/Do

Aprenderé a tocar la «Canción de los cuatro acordes». Si el asunto es cierto, aprenderé simultáneamente a tocar cualquier canción pop. ¡Parece un comienzo eficiente!

Resulta que estos cuatro acordes son bastante fáciles de tocar cuando se sabe cómo funciona el instrumento. Examinemos el ukulele para saber de qué se trata el asunto.

Anatomía de un ukulele



Los ukuleles tienen cuatro cuerdas que se extienden a lo largo del cuerpo del instrumento. La pieza larga que sobresale del cuerpo del instrumento se llama *mástil*.

Justo al final del mástil, hay una pieza de madera llamada *pala* o *cabeza*. Las *clavijas* afinadoras están incorporadas a la pala, y sostienen con seguridad el extremo de las cuatro cuerdas del instrumento. Estas cuerdas van de la pala, pasando por el mástil, sobre la *boca* hasta llegar al *puente*, que está cerca de la parte inferior del ukulele. El puente mantiene a las cuerdas en su lugar, bien estiradas sobre la boca y el mástil.

Tradicionalmente, el ukulele se toca envolviendo el mástil con la mano izquierda y usando el brazo derecho para abrazar el instrumento pegándolo al pecho. El ejecutante rasguea las cuerdas con los dedos de la mano derecha

(usualmente con el dedo índice o con el pulgar), en tanto que los dedos de su mano izquierda presionan sobre el mástil para cambiar el tono de las cuerdas.

Afinación

Cada cuerda tiene una *afinación*: un tono que se produce cuando se pulsa una cuerda sin que se presione esta en ningún sitio del mástil. El tono de cada cuerda se ajusta al girar las clavijas afinadoras. Si una clavija es girada para agregar tensión a la cuerda, el tono sube. Si una clavija se gira para reducir la tensión de una cuerda, el tono baja.

A continuación, te presento la afinación estándar para el ukulele tenor^[111]:

- Cuarta cuerda (parte inferior) = Sol (196 Hz)
- Tercera cuerda = Do (261.6 Hz)
- Segunda cuerda = Mi (329.6 Hz)
- Primera cuerda (parte superior) = La (440 Hz)

Afinar el ukulele es muy fácil si tienes un buen aparato afinador. Ahí es donde el Snark SN-9 entra en juego. Lo pones en la pala, lo enciendes, pulsas una cuerda, el afinador detecta el tono de la cuerda, convierte la frecuencia en una nota y te dice si el tono es bemol (ligeramente por debajo de la nota) o sostenido (ligeramente por encima de la nota). Cuando el tono es correcto, se enciende una luz y se pasa a la siguiente cuerda. Cuando las cuatro cuerdas están afinadas, estás listo para tocar.

Una afinación apropiada es muy importante: si las cuerdas no están correctamente afinadas, el ukulele suena tan mal que hasta los que dicen no tener buen oído lo notan. Mientras más precisa sea la afinación, mejor sonará el instrumento.

Notas y acordes

Si quieres tocar una sola nota en el ukulele, solo debes rasguear una cuerda, lo que provoca que vibre sobre la boca del ukulele, en el centro del cuerpo, creando sonido. Si presionas una cuerda en la zona del mástil al tiempo que la pulsas, la cuerda se torna temporalmente más corta y vibra más rápido al ser

rasgueada. Mientras más alta sea la frecuencia vibratoria, más alto será el tono del sonido.

Lo que hace interesante al ukulele y a otros instrumentos de cuerda, como las guitarras y los pianos, es que son capaces de tocar varias notas a la vez. Si un ejecutante de ukulele rasguea cada cuerda con la mano derecha, se tocarán cuatro notas de modo simultáneo. Un buen ejecutante puede presionar las cuerdas en el mástil de modo que el tono que cada cuerda genera complementa al de otra, creando un *acorde*.

Gracias a los acordes, los instrumentos de cuerdas tienen un sonido rico y vibrante. Al tocar combinaciones de notas en lugar de notas aisladas, el ejecutante puede crear melodía y armonía al mismo tiempo. En cierto sentido, los acordes permiten al ejecutante convertirse en su propio acompañante.

Mejor aún, a diferencia de la trompeta, el clarinete, la flauta o el saxofón, los instrumentos de cuerdas solo requieren que el ejecutante utilice sus manos. Con suficiente práctica, es posible tocar acordes complejos y cantar al mismo tiempo, razón por la cual los instrumentos como la guitarra y el piano son populares entre los vocalistas.

Tocar acordes

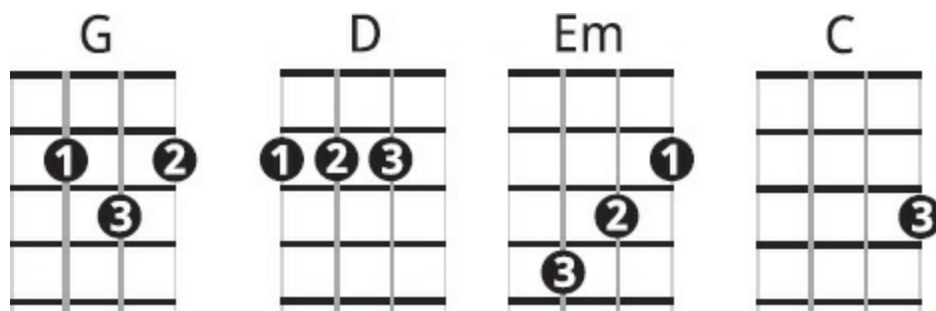
Dado que estoy empezando de cero, debo elegir: puedo concentrarme en aprender a tocar notas individuales leyendo partituras, o practicar acordes. Dado que son nuevos para mí y quiero cantar y tocar al mismo tiempo, me inclino por los acordes.

Los diagramas de los acordes son representaciones visuales de qué cuerdas presionar sobre el mástil con los dedos de la mano izquierda. Si presionas la combinación de cuerdas mostrada en el diagrama, tocarás el acorde.

Aquí presento los cuatro de la «Canción de los cuatro acordes»:

Sol/Re/Mi menor/Do

Ahora te presento cómo se ven esos cuatro acordes en el ukulele:



Si sostienes el ukulele verticalmente, se verá igual que los diagramas. La línea gruesa que está arriba del diagrama representa el borde superior del mástil del ukulele, que se llama nuez. Cada línea vertical debajo es un traste, que corresponde a las piezas metálicas incrustadas en el mástil y que sobresalen de este.

Los trastes hacen que sea mucho más fácil producir notas limpias cuando se presionan las cuerdas contra el mástil. Cuando tocas, no se presionan los trastes: presionas las cuerdas en el espacio entre los trastes. Al hacerlo, el traste bajo tu dedo también presiona la cuerda, acortándola y alterando el tono correspondiente.

Los diagramas de acordes indican qué cuerdas debes presionar y en dónde debes hacerlo para tocar acordes determinados. Las cuerdas están representadas por las líneas verticales: la cuerda de la izquierda es la cuarta, que sería la de arriba en la posición del ukulele. La cuerda más a la derecha es la de abajo, la primera.

Los círculos negros indican dónde presionar, y los números representan los dedos a utilizar:

- 1 = dedo índice
- 2 = dedo medio
- 3 = dedo anular
- 4 = dedo meñique

De acuerdo con el diagrama, para tocar un acorde en Sol, necesito presionar las tres cuerdas inferiores con mis dedos índice, medio y anular. Mi índice debe presionar en la tercera cuerda, justo encima del segundo traste; mi dedo

medio debe estar en la misma posición en la primera cuerda y mi dedo anular debe presionar la segunda cuerda encima del tercer traste.

Suena muy complicado, pero en la práctica no lo es tanto. Una vez que queda claro dónde va cada dedo, tocar acordes se convierte en una destreza motora estándar: aprender la posición de los dedos y en dónde presionar para producir un acorde determinado.

Para practicar acordes, es útil tener un juego de diagramas de referencia para mostrarte cómo tocar los acordes comunes. Una breve búsqueda en Internet me llevó a la «Gráfica Kiwi de acordes indispensables para ukulele^[112]», guía que es posible bajar para tocar acordes en el ukulele. La guía fue creada por Mike Dickison, autor de *Kiwi Ukulele: The New Zealand Ukulele Companion* (2008).

Hay muchos lugares en los que puedes encontrar gráficas de acordes: prácticamente todos los libros sobre ukulele contienen un juego de diagramas de acordes. Lo que hace única a la gráfica de Mike es su organización por frecuencia de uso. Los acordes comunes, como los que aparecen en la «Canción de los cuatro acordes», se enlistan primero. Los acordes raros se enlistan al final, puesto que rara vez se usarán.

Este tipo de análisis de frecuencia es muy útil: la ley del poder aplica también a la música. Para tocar canciones, no debes saber cientos de acordes. Con 10 o 12 abarcarás la mayor parte de las canciones.

La práctica de los acordes

Elijo los diagramas que se usan en la «Canción...» y paso algún tiempo aprendiendo dónde va cada dedo. Cuando logro conformar la posición correcta con los dedos, rasqueo las cuatro cuerdas. Funciona: ¡puedo producir acordes que suenan bien!

En esta etapa temprana, el mayor reto es presionar la cuerda de modo que el traste haga bien su trabajo. Si no se presiona con suficiente fuerza, la cuerda no producirá un tono claro. En lugar de ello, se obtiene un zumbido poco atractivo junto con la nota que se desea, lo cual distrae.

Una vez aprendidos los cuatro acordes, paso una hora tocando cada uno de ellos y repitiendo la secuencia una y otra vez:

Sol... Re... Mi menor... Do...
Sol... Re... Mi menor... Do...
Sol... Re... Mi menor... Do...

Este es el primer paso en la adquisición de habilidades motoras: debo ayudar a que mi cerebro asocie ciertos pensamientos, como el nombre de un acorde, con el patrón de movimientos musculares de mi mano izquierda. Luego, podré tocar un acorde en Mi menor cuando quiera, pero para ello necesito practicar la correcta disposición de los dedos, presionando lo suficientemente fuerte el mástil y cambiando de acordes en secuencia.

También practico antes de irme a la cama, usando la misma estrategia que para reaprender mecanografía. Al practicar antes de dormir, me aseguro de que mi cerebro pueda consolidar estos movimientos con toda la eficiencia posible.

Patrones de rasgueado

Al día siguiente, puedo tocar bien los cuatro acordes y pasar de uno a otro a voluntad. No está mal para haber practicado solo una hora. Ahora que tengo una idea de lo que mi mano izquierda hace, necesito concentrarme en la derecha.

En tanto que mi mano izquierda está ocupada tocando acordes, la derecha será responsable por el rasgueo de las cuerdas para producir sonido. Hasta este punto, he rasgueado con un patrón muy básico: contando hasta cuatro y rasgueando con cada número:

1 2 3 4
1 2 3 4
1 2 3 4

La mayoría de las canciones, cuando son transcritas al papel en notación musical, se escriben en lo que se llama tiempo de 4/4. Sin entrar en muchos detalles, las canciones 4/4 se organizan alrededor de cuatro sonidos fuertes.

La siguiente vez que escuches una canción, trata de contar a ritmo del uno al cuatro en tu cabeza. Casi siempre notarás que el patrón funciona.

Obviamente, si cada canción consistiera en tocar notas o acordes en un patrón simple y predecible una y otra vez, la música se tornaría aburrida. Para añadir interés y variantes, los músicos rompen intencionalmente este sencillo patrón de conteo usando una técnica llamada *síncopa*.

Síncopa significa tocar de modo que nos alejemos de la progresión normal de pulsos. He aquí cómo suena: con una de tus manos, toca un ritmo sencillo en una superficie lisa, una mesa, un escritorio, etc. Mantén los golpes fuertes y regulares. Mientras lo haces, cuenta del uno al cuatro una y otra vez:

1 2 3 4

1 2 3 4

1 2 3 4

Bastante sencillo, ¿no?

Ahora, sin dejar de tocar el ritmo, añade una variante. Usa tu otra mano para tocar *entre* cada golpe, así:

1 y 2 y 3 y 4 y

1 y 2 y 3 y 4 y

1 y 2 y 3 y 4 y

Se trata de una *síncopa* muy básica. Los golpes que coinciden con los «y» están fuera del ritmo normal, lo que añade variedad y hace que el ritmo sea más interesante.

Los músicos habilidosos pasan mucho tiempo experimentando con la *síncopa*. Cada pulso o *beat* puede dividirse en muchas partes. Es muy común que los *beats* se dividan en cuatro secciones, que se cuentan así:

1oia 2oia 3oia 4oia

Suena «un-o-i-a, dos-o-i-a, tre-o-i-a, cua-o-i-a». La «o» es larga y la «a» es corta.

Al acentuar o enfatizar ciertos pulsos del ritmo, al tiempo que se dejan fuera otros, el músico puede crear todo tipo de ritmos interesantes que le van a la canción.

Es importante saber esto porque en el ukulele la mano derecha lleva el ritmo. Si quiero hacer cualquier cosa además de rasguear con el pulso del ritmo, lo que es aburrido, necesito pasar tiempo descubriendo patrones de rasgueo interesantes.

Después de experimentar un poco, me quedo con el siguiente patrón:

1 ia oia3 ia oia

Me toma algún tiempo entrenar mi mano derecha para rasguear el ritmo, pero mejoro poco a poco. Primero, toco un simple acorde en Do mientras rasqueo. Cuando me siento cómodo, comienzo a moverme por la secuencia de los cuatro acordes, cambiando los acordes en los pulsos o *beats* 1 y 3:

Sol	Re
1 ia oia	3 ia oia
Mi	Do
1 ia oia	3 ia oia

En este punto, estoy tocando la canción tal cual: toco los acordes en el orden correcto con la mano izquierda mientras mantengo el patrón de rasgueo con la derecha.

Por primera vez, toco una canción en el ukulele. Es sencillo, pero funciona.

Repito la secuencia una y otra vez. Cuando me siento cómodo con ella, cierro los ojos y trato de tocar sin mirar el mástil o las cuerdas. Cometo más errores, pero soy capaz de mantener la canción en movimiento y vuelvo al camino correcto después de cometerlos.

No está mal para haber practicado solo dos horas.

Tocar y cantar al mismo tiempo

Ahora que puedo tocar los acordes y mantener el patrón de rasgueo, debo añadir algo. ¿Podré arreglármelas para cantar y tocar al mismo tiempo?

La «Canción de los cuatro acordes» es chistosa porque es fácil reconocer las canciones. Sin las letras, parece que estoy tocando lo mismo una y otra vez. Por supuesto, ese es el punto, pero sin las letras la canción no tiene sentido.

A lo largo de los años, mi trabajo me ha llevado a conocer personas muy agradables. Entre ellas se encuentra Derek Sivers, fundador de CDbaby.com. Derek es un guitarrista y cantante exitoso, así que le pedí consejo sobre cómo proceder. He aquí lo que me recomendó:

Primero, ¡memoriza la canción solo cantándola! Es importante separar el conocimiento del instrumento del conocimiento de la canción. Debes cantar toda la canción, sin instrumento alguno. (La calidad de la voz no importa: canta, tararea, silba, lo que sea).

Cuando ya memorizaste la canción, canta el nombre de las notas en lugar de la letra. Memoriza la canción así, con los ojos cerrados.

Finalmente, añade el instrumento cantando el nombre de las notas conforme las tocas en las cuerdas.

Estamos ante un ejemplo brillante de deconstrucción: dividir un proceso complejo en sus partes más simples. Para cantar y tocar al mismo tiempo, debes conocer la letra y la melodía de una canción lo suficientemente bien para recordarla mientras tocas los acordes y mantienes el patrón de rasgueo. Al separar la canción del instrumento en principio, te aseguras de saberla bastante bien para no tener que pensar mucho en ella cuando agregues el instrumento.

Cuando ya aprendiste la letra, cámbiala por los nombres de las notas para sentir cuándo cambiar de acordes al estar tocando. Dado que la letra suele

tener un ritmo propio, puedes usarla para recordar cuándo cambiar: ¡una cosa menos en qué pensar!

Finalmente, al cantar el nombre de los acordes al ritmo de los cambios de acordes y del patrón de rasgueo, ayudas a tu cerebro a unir las piezas. Si puedes mantener todo junto, cambiar del nombre de los acordes a las letras es simple, puesto que ya las memorizaste.

Me tomé muy en serio el consejo de Derek y comencé a memorizar la letra de la «Canción de los cuatro acordes» escuchándola una y otra vez para aprenderme la tonada. Luego practiqué la letra escribiéndola en un cuaderno para repetirla una y otra vez de memoria.

Después, toqué la canción en el ukulele mientras silbaba la melodía, manteniendo el nombre de los acordes en la cabeza. Era fácil saber cuándo se debía cambiar el acorde. De pronto, empecé a tocar y a silbar al mismo tiempo.

Agregar la letra fue más fácil de lo que pensaba. De vez en cuando olvidaba la letra o me equivocaba en algún acorde, así que me detenía y comenzaba de nuevo.

Tras cinco horas de práctica, puedo tocar y cantar mi primera canción completa. Hay momentos en que no suena muy bien, pero las piezas del rompecabezas están ahí.

Me quedan cinco días para la hora de la verdad. Puedo desempeñarme bastante bien a solas, pero ¿cómo serán las cosas cuando la gente me esté mirando?

Automatización

Para estar seguro de que podré tocar en el evento, el resto de mi práctica consistirá solamente en la repetición, siempre antes de ir a la cama. La meta es sencilla: quiero hacer que cambiar de acordes, mantener el rasgueo y cantar la letra requiera de tan poca atención mental como sea posible.

Esta parte de la práctica no es glamorosa, pero sí es muy importante. Hasta este punto, todo ha tenido que ver con la fase cognitiva de la adquisición de habilidades motoras: analizar, deconstruir y experimentar. Ahora, necesito fortalecer la fase asociativa y permitir que mis músculos asuman la tarea sin pensar conscientemente.

Cada día, toco y canto la canción una y otra vez: ya perdí la cuenta de cuántas veces lo he hecho. Mis cambios de acordes están siendo más suaves, el patrón de rasgueo se hace más consistente y recuerdo la mayoría de las palabras. Algunas de las transiciones entre canciones son truculentas, así que pasé más tiempo practicándolas.

Antes de darme cuenta, llegó la hora de viajar a Portland. No viajaré solo: esta vez me acompañará el Grizzly.

Llegó la hora

Mi participación está programada para el último día de la conferencia. Lo bueno es que el horario me da un par de horas más para practicar. Lo malo es que me quedan dos días de angustia por saber cómo irán las cosas.

El día de la conferencia afiné el Grizzly y practiqué por última vez una hora antes de mi conferencia. Luego caminé hasta el lugar indicado, ukulele en mano.

No tenía idea de cuánta gente asistiría, así que me tranquilizó ver que se trataba de un salón relativamente pequeño. No sé cómo me las arreglaría con un auditorio entero a estas alturas.

El lugar se llena. En el salón caben 40 personas sentadas, pero llega más gente y algunos se quedan parados en los costados del salón. La energía es alta.

Después de explicar mi forma de investigar y el método, saco el ukulele y todos se emocionan visiblemente. Hago mi mejor esfuerzo porque no se me noten los nervios.

Vamos. Comienzo a tocar y luego a cantar.

Salió bastante bien: mucho mejor de lo que yo esperaba. Seguro que no fue una presentación digna del Carnegie Hall, pero toqué la canción entera sin equivocarme con los acordes, sin perder el ritmo y sin olvidar las palabras. El auditorio sonrió, me acompañó llevando el ritmo con los pies, se rio con las letras y aplaudió al final.

Mi entrenamiento cumplió: en lugar de mirar las cuerdas o el mástil todo el tiempo, pude mirar a la gente y disfrutar del momento.

Pulsar cuerda por cuerda

Al terminar la canción, mostré al auditorio otro truco que se me ocurrió al practicar: puedes tocar canciones que suenan complicadas si pulsas las cuerdas en lo individual en una secuencia, en lugar de rasguitarlas todas al mismo tiempo. Los acordes son exactamente los mismos.

Esta técnica se llama pulsar *cuerda por cuerda*, y hay muchas formas de hacerlo. El patrón que demuestro es muy simple: pulsar la primera cuerda, la cuarta, la segunda y la tercera en el *beat*, en ese orden. Al repetir este patrón mientras la mano izquierda toca acordes, cada nota complementa naturalmente a la siguiente. El resultado ofrece un sonido agradable y complejo, aunque la técnica no es más difícil que el rasgueo.

La conferencia fue un éxito. Además de divertirme, me da gusto haber aceptado el riesgo, presionarme y practicar bien. No está nada mal ir de cero a la primera presentación en público en 10 horas.

Uno-cuatro-cinco

Por casualidad, me encontré con otro participante en la convención, Melissa Dinwiddle, justo afuera del auditorio principal. Melissa es artista y casualmente llevaba consigo su ukulele, así que sacamos nuestros instrumentos y tocamos juntos un rato.

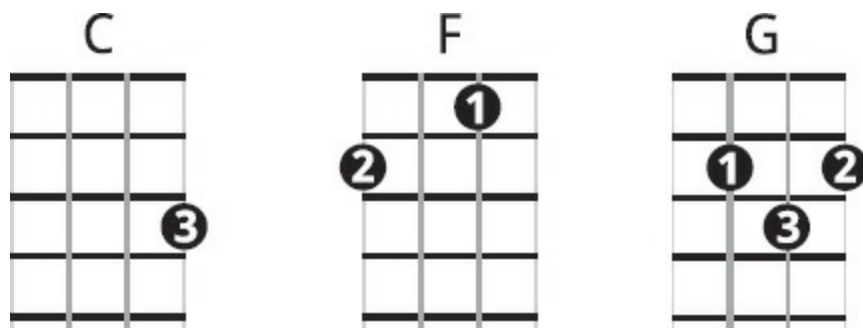
Una de las cosas que Melissa me enseñó ese día fue una estructura de canción llamada «*Blues de los 12 compases*», que se basa en una secuencia de acordes conocida como uno-cuatrocinco.

Sin entrar en demasiada teoría musical, cada nota tiene un acorde asociado, así como una familia de acordes que complementan al acorde raíz. Si puedes tocar este, así como los acordes de la familia asociada, de pronto resulta que puedes tocar miles de canciones comunes.

Tomemos como ejemplo «Twinkle, Twinkle, Little Star». La canción entera puede tocarse usando tres acordes, que siguen este orden:

Do/Fa/Sol

Aquí te presento los diagramas de los acordes:



En este caso, Do es el acorde raíz o acorde I. Fa y Sol son otros dos acordes de esta familia. Fa es el acorde IV y Sol el V.

Usando solamente esos tres acordes, se pueden tocar toda suerte de canciones divertidas, como:

- «La canción ABC.»
- «Frosty el hombre de nieve.»
- «La canción de Elmo.»

Eso significa que me las puedo arreglar para tocar a Lela canciones infantiles cuando llegue a casa, así como algunos *blues* divertidos.

Mis siguientes 10 horas de práctica estarán dedicadas a explorar las canciones que puedo tocar. Seguiré trabajando con el libro *The Daily Ukulele* (Belloff, 2010), aprenderé nuevos patrones de rasgueo con el libro *Absolute Beginners Ukulele* (2009), y aprenderé más teoría musical básica de la obra *Fretboard roadmaps-ukulele* (2006). ¡Las horas vuelan!

Tom Hodgkinson tenía razón: si tomas un ukulele, nunca volverás a sentirte aburrido.

Repaso del método

Hagamos un repaso del método que utilicé para aprender a tocar el ukulele:

- Conseguí un ukulele, cuerdas, un afinador y demás equipo necesario.
- Decidí empezar concentrándome en aprender las secuencias de acordes y los patrones de rasgueo más comunes, lo que me permitió tocar miles de canciones populares.
- En cuanto toqué los acordes y pude rasguear sin mirar el mástil o las cuerdas, practiqué hasta lograr mantener el patrón y cambiar de

acordes sin pensar en ello.

- Cuando me sentí cómodo con los acordes básicos y los patrones de rasgueo, empecé a aprender canciones tarareando mientras tocaba, aprendiendo en qué momentos se sobreponían los acordes y las palabras. Luego añadí la letra hasta que pude cantar y tocar simultáneamente.
- Experimenté con la pulsación individual de las cuerdas en la misma secuencia de acordes, en lugar del rasgueo común, lo que me permitió tocar piezas más complejas.
- Aprendí la secuencia de acordes I, IV, V, que me ayudó a poder tocar aún más canciones.

¿Y ahora qué?

Me encanta tocar el ukulele. Entre el cancionero llamado *The Daily Ukulele Songbook* y las canciones que encuentro en línea, puedo tocar casi cualquier canción. Algunos acordes son más difíciles que otros, y existen miles de patrones de rasgueo y de pulsaciones de cuerda individual por aprender, lo que me mantendrá ocupado por largo tiempo.

Una de las mejores cosas de tocar el ukulele es que me provoca poca presión. Aparte de mi experimento inicial, no me preparo para dar una presentación ni pretendo ser músico profesional en el futuro. Cuando tomo mi ukulele, me puedo relajar y jugar con él, mientras aprendo una nueva canción, técnica o patrón de rasgueo.

Lela ya ha crecido lo suficiente como para recordar las canciones, por lo que mi viejo y polvoso ukulele Oscar Schmidt es muy útil: ahora es el ukulele de Lela; sirve para distraerla cuando yo practico o como instrumento principal cuando ella decide sentarse en mis piernas para tocar las cuerdas mientras yo cambio los acordes. Ahora que soy capaz de tocar canciones que ella reconoce. Suele pedirme que toque el «ukuyaya» antes de ir a dormir.

Es agradable que la música forme parte de nuestra vida.

El surf a vela o *Windsurf*

Lección: el ambiente cuenta.

No luchamos con la carne y la sangre, sino con la resina epoxi, la fibra de carbón, el monofilm para reparar velas y con nuestros egos, sueños y voluntades indómitas.

Chris Zeitvogel, surfista a vela.

Para obtener imágenes, videos y comentarios adicionales sobre este capítulo, visita <http://first20hours.com/windsurfing>.

Estoy en medio del lago y trato desesperadamente de levantar mi vela. Es difícil: los músculos de mis brazos, piernas y espalda gritan por el esfuerzo.

Entra un fuerte viento del noroeste y produce pequeñas olas crestadas que balancean mi tabla. Dejo que la vela vuelva a caer al agua y hago una pausa, aprovechando el momento para descansar y para acostumbrarme al balanceo de la tabla bajo mis pies.

Miro al oeste: se forma una tormenta que empuja los nubarrones oscuros sobre las montañas. Ha llegado la hora de intentarlo otra vez.

Me estiro, tomo la driza, jalo y la vela sale del agua por trigésima ocasión en los últimos 30 minutos. Espero regresar a tierra antes de que llegue la tormenta.

Ya casi... no soy muy bueno que digamos para dirigir este artefacto, pero tengo que aprender. Comienza a cansarme remar hasta la costa.

La vida en el agua

Tengo muy buenos recuerdos de los veranos que pasé en zonas con agua. Al crecer, estuve muy involucrado con los Boy Scouts de América, y disfruté de muchos veranos como miembro del equipo anfitrión en dos campamentos al norte de Ohio: el Campamento Firelands y el Avery Hand.

Durante mi último campamento de verano, tuve el honor de servir como director asistente de actividades acuáticas, bajo la dirección del legendario Bob «Aqua» Sliney. Bob suele estar a cargo del programa de alberca: nado, actividades de salvamento y nado de la milla. Yo era responsable de las demás actividades acuáticas: canotaje, remo, vela y lanchas con motor.

Amé cada minuto en ese campamento. Pasaba la mayor parte de cada día en el agua enseñando a jóvenes *scouts* cómo disfrutar con seguridad de las actividades acuáticas. Era un trabajo lleno de retos y cansado, siempre desde una perspectiva positiva.

Cuando asistí a la universidad, dejé atrás el agua. Mi primer trabajo en una empresa de Cincinnati ocupaba todo mi tiempo. El río Ohio, el cuerpo de agua más cercano, no es ideal para los paseos recreativos. Luego, cuando Kelsey y yo nos mudamos a la ciudad de Nueva York, lo más cerca que estuvimos de un bote fue cuando paseamos en góndola en Central Park. Cuando nos avecindamos en Colorado, estuve seguro de que mis días en bote habían terminado; después de todo, esta parte del mundo no se distingue exactamente por su abundancia de agua.

Sin embargo, recientemente pude negociar mi acceso a un pequeño lago privado, justo al norte de la ciudad, a unas cuantas millas de mi casa. Me siento mareado: es la primera vez en años que pasaré algo de tiempo en el agua.

¿Qué debo hacer primero?

El contagio del virus del *windsurf*

Mi primera idea fue intentar con el remo: el lago es lo suficientemente grande para el canotaje. Pero los botes de remos comunes son grandes y pesados; hasta los más pequeños son largos y relativamente frágiles, lo cual es una molestia, dado que no puedo guardar con seguridad embarcaciones grandes o largas cerca del lago.

Tengo el mismo problema con los veleros, a lo que debemos sumar la desventaja del precio. Hasta el equipo básico suele costar un ojo de la cara. Además de las complicaciones, preferiría no tener que gastar 10 mil dólares en equipo.

El canotaje es más sencillo y menos caro, pero no es tan divertido hacerlo solo: la longitud de las canoas es ideal para dos remeros. Las lanchas de motor y los *Jet Sky* están fuera del juego, pues los dueños del lago limitan los motores permitidos a los que tienen menos de 9 caballos de fuerza; además, prefiero las embarcaciones impulsadas por el hombre.

Al investigar sobre mis posibilidades reales, di con un video tomado por James Douglass, un profesor de biología marina de la Universidad de la Costa del Golfo de Florida. Además de su experiencia en el área de la biología, Jim es un surfista a vela bastante competente que publica muy buena información sobre este deporte en su blog personal^[113].

En el video, Jim equipa una tabla de surf profesional con una cámara de video de alta definición, fija al extremo de la vela para capturar toda la acción al mismo tiempo^[114]. Él *vuela* sobre el agua, saltando las pequeñas olas y dando vueltas rápidas e impresionantes.

Quedé prendado y vi el video varias veces seguidas. Nunca antes había estado en una tabla de surf y de pronto quise intentarlo.

El *windsurf* se ajusta a todos mis criterios: lo puedo hacer solo y el equipo necesario no es enorme o estorboso. Según mis investigaciones incipientes, el equipo no es tan caro: tendré que comprar equipo, pero algo básico no tiene por qué terminar con mi cuenta bancaria.

El *windsurf* parece divertido, pero antes de lanzarme quiero constatar que sea seguro. No soy adicto a la adrenalina por naturaleza ni me interesa buscar emociones a costa de todo. Si resulta que es peligroso, me contentaré con realizar otra actividad.

Si voy a practicarlo debo conocer los riesgos antes de comprometerme. ¿Qué puede salir mal?

¿Es peligroso el *windsurf*?

Es común clasificarlo como deporte «extremo», pero el grado de peligrosidad depende de qué tipo de vela en tabla quieres hacer. En aguas tranquilas y cerradas, al estilo de las que usaré para aprender, es más intenso que pilotear un pequeño velero, pero no llega a ser una locura. Por otra parte, hacerlo en el mar conlleva muchos más riesgos. Las olas que alcanzan la orilla hacen más difícil el comienzo; perder el aparejo lejos de la costa es un problema bastante grave. El mar añade elementos divertidos, como saltar olas, pero también incrementa riesgos. Comparado con el *windsurf* en el mar, en un lago es muy seguro.

Sin importar el tipo que practiques, existen riesgos significativos, como ahogarte o sufrir hipotermia. No hay vuelta de hoja: cuando flotas en un pedazo de poliestireno en medio de las aguas, sosteniendo una vela grande en un día ventoso, pueden pasar cosas malas si no estás preparado.

El riesgo de ahogamiento está compuesto por dos factores: la lesión y el cansancio. Si llegas a caer de modo que te lastimes algún miembro o padeciendo una conmoción, puedes perder el sentido en el agua. Si te cansas tanto que no puedes volver a la tabla ni llegar a la orilla, el ahogamiento es posible.

La mejor manera de prevenir el ahogamiento es: 1. usar aparatos de flotación (salvavidas), y 2. siempre estar acompañado, ya sea por alguien en el agua o atento desde la orilla. Si llegan los problemas, es deseable que alguien esté cerca para ayudar^[115].

La hipotermia puede ser igualmente mortífera. En condiciones de frío, humedad y viento, el cuerpo pierde calor con gran rapidez. Una vez que tu temperatura corporal cae por debajo de 35 grados centígrados, tu corazón, pulmones y sistema nervioso comienzan a colapsar, llevando a la muerte a

menos que la temperatura corporal se restaure a niveles operativos, que se encuentran entre los 36,6 y los 37,7 grados centígrados.

Lo que hace particularmente peligrosa a la hipotermia es que sucede gradualmente. Al bajar la temperatura corporal, comienzan síntomas como temblor, confusión, pérdida de coordinación y fatiga, lo que afecta la destreza física y el juicio. Si dependes del buen juicio o de la destreza física para volver a casa, la llegada de la hipotermia es una amenaza mayor que es muy fácil pasar por alto.

En el agua, cuando la temperatura sube de los 18,3 grados, la hipotermia no es un gran factor de riesgo. Por debajo de ese punto, es necesario tener aislamiento antes de entrar al agua. Ahí entran en juego los trajes especiales.

Este tipo de prendas están fabricadas con materiales delgados y flexibles que aíslan el agua, como el neopreno. Los trajes térmicos modernos están clasificados de acuerdo con los milímetros de grosor, y suelen diseñarse para que las partes más gruesas cubran el torso. Este diseño responde a dos propósitos: más aislamiento en el torso preserva el calor de modo más eficiente, en tanto que menos material en los brazos y piernas facilita el movimiento. Para practicar *windsurf* en condiciones de frío, se requieren aislamiento y flexibilidad.

La mejor protección en condiciones comunes se obtiene combinando dos trajes de aislamiento. Se usa una versión de perneras y mangas cortas, que expone brazos y piernas, para temperaturas más altas (basta con 3 milímetros de grosor en el torso y 2 milímetros en los miembros [3/2]). En aguas con temperatura por debajo de los 18,3 grados, es mejor usar un traje completo 5/4 con botas, guantes y capucha de neopreno^[116].

Las zonas costeras de California son destinos populares para este deporte, así que el departamento estatal encargado de las aguas preparó una lista bastante práctica con consejos de seguridad^[117]:

1. Toma en cuenta los pronósticos del clima y del oleaje.
2. Siempre avisa a alguien sobre el sitio al que planeas ir y cuándo pretendes regresar.
3. Usa ropa que se ajuste a las condiciones climatológicas.
4. Usa un chaleco salvavidas aprobado por la Guardia Costera de los Estados Unidos y también procura que este chaleco tenga un silbato

incorporado.

5. En condiciones de calor, sol y humedad, toma suficientes líquidos.
6. Revisa tu equipo en busca de indicadores de daño o fatiga de materiales.
7. Velea con un compañero.
8. Cuando los vientos lleven mar adentro, deja de velear.
9. El frío puede matar. Cuando tiembles por primera vez, regresa a la orilla y entra en calor.
10. Permanece sobre tu tabla y nunca trates de nadar hasta la orilla.

El estado también ofrece una lista que debes revisar antes de lanzarte al agua:

Antes del chapuzón...

1. Revisa doblemente tu correa de seguridad.
2. Debes estar atento a las nubes oscuras en el horizonte: las tormentas golpean rápido.
3. Si tienes dudas, no entres al agua.
4. Un marinero inteligente intenta primero con el curso de acción más seguro, antes de que el rescate sea la última posibilidad.

Se trata de puro sentido común, pero es importante. El *windsurf* es divertido. Morir no lo es. Con un poco de preparación y planeación, los grandes riesgos pueden minimizarse.

¿Por dónde comenzar?

Como era de esperarse, el norte de Colorado no es precisamente un sitio ideal para la práctica de este deporte. Hay suficiente viento para intentarlo si tienes un lugar para practicar, pero en contraste con los destinos principales como la garganta del río Columbia, en Washington, los vientos locales son variables e impredecibles, no fuertes ni sostenidos.

Además, los grandes cuerpos de agua no son muy comunes al norte de Colorado. Dado que buena parte de la cordillera Front es una estepa semiárida de altura, los grandes cuerpos de agua, como los lagos, no se forman naturalmente. La mayoría de los «lagos» de la zona son reservas

artificiales diseñadas para contener agua para los granjeros locales; son alimentadas por ríos como el turbulento y bello Cache La Poudre, que lleva agua de deshielo de las montañas a las planicies.

Como resultado, la geografía local sirve de escenario a una práctica muy bien establecida de navegación en rápidos, pero no al *windsurf*. Yo no conozco a nadie que lo practique regularmente, y no hay muchos comerciantes locales que vendan tablas o equipo. Igualmente, después de buscar instructores en la zona, tengo la impresión de que tendré que valerme por mis propios medios. El lugar más cercano donde se imparten clases es Denver, a una hora de camino. Eso dificulta la conformación de una agenda de práctica, pues algunos días habrá bastante viento, mientras en otros será casi nulo, de modo que cualquier reservación conlleva un riesgo significativo de ser cancelada.

Aunque sería ideal comenzar con algo de instrucción personal, las responsabilidades familiares y laborales me obligan a permanecer en casa por ahora. Confío en que, si puedo conseguir el equipo correcto, podré aprender la técnica ya estando en el agua.

Pero ese asunto de conseguir el equipo no es tan sencillo cuando no hay tiendas cercanas. A mi entender, las tablas para el surf a vela son más grandes que los mayores paquetes que empresas como Fedex y UPS transportan regularmente. Si no puedo encontrar el equipo adecuado, este proyecto estará muerto antes de empezar siquiera.

La compra de equipo

Mi primera fuente de información es el sitio web de Jim, que contiene muchos consejos para principiantes. Uno de los primeros ensayos que leo es un *post* titulado «Respuesta a las 18 principales preguntas sobre *windsurf*^[118]», que incluye también un tutorial sobre el equipo necesario.

A partir de estas preguntas y respuestas, parece que necesitaré, mínimo, una tabla, una vela, un a PFD y un traje térmico. Leí todos los *posts* sobre el surf a vela del sitio de Jim, hice una lista con el equipo que parecía útil, traté de hacer una selección más rigurosa hasta quedar con las dos mejores opciones y luego le escribí un correo a Jim para darle las gracias y también para pedir su consejo.

De vital importancia son el tipo de tabla y de vela que debía comprar. Era impresionante el video de Jim en el que volaba en una tabla de competencia (conocidas como tablas Formula), y me encantaría llegar a ese nivel de desempeño, pero no estoy seguro si es buena idea comenzar con una tabla así. De acuerdo con mi investigación, la otra opción es una tabla muy versátil llamada Rio, que es fabricada por una empresa llamada Starboard. La Rio tiene la reputación de ser muy amigable con los principiantes y de ser muy buena tabla en todo tipo de condiciones: no irá tan rápido como una Formula, pero se desempeñará bien en una amplia gama de condiciones.

No sabía qué hacer, así que le pedí consejo a Jim:

Soy completamente nuevo en el *windsurf*, ni siquiera me he parado sobre una tabla todavía. Espero pueda darme un consejo para empezar. Quiero comprar un juego completo de equipo y trato de decidir entre la Rio mediana de Starboard y la Formula 167, también de Starboard.

La Rio parece una buena tabla para principiantes, pero me gustaría aprender tan rápido como sea posible. Solo surfearé en aguas tranquilas, y quisiera hacerlo con menos viento del ideal (4-5 nudos). Las tablas Formula, según sé, hidroplanean en el agua más rápido y funcionan mejor con menos viento.

He aquí mi pregunta: ¿es una locura tratar de aprender en una tabla Formula? Tengo mucha tolerancia para la frustración en las etapas tempranas y preferiría comprar una sola tabla a tener que comprar más de una. Si aprendo en una Formula, ¿corro algún riesgo significativo de afectar al equipo o lastimarme?

Gracias por cualquier consejo que pueda darme. ¡Aprecio su ayuda!

Una hora más tarde, Jim me responde:

1. La cuestión de que las tablas Formula son mejores en el caso de viento ligero es engañosa. Tienen, más que ninguna otra tabla, el potencial de planear con vientos más ligeros (7-8 nudos), pero tienes que ser muy buen marinero para controlar una enorme vela de 11-12 metros cuadrados de superficie y explotar así el potencial. Además, cuando

una tabla Formula no planea es como un perro, por ser tan corta, ancha y por carecer de orza o quilla. Para condiciones ajenas al planeo (siendo realista, para cualquier condición con menos de 10 nudos), una tabla más larga y menos ancha (como la Rio M) es mucho más rápida y fácil de navegar que una tabla Formula.

2. Es poco probable que te lastimes tratando de aprender en una tabla Formula, a menos que te duela la espalda al lidiar con una vela tan grande. La razón de que sea poco probable lastimarte en una tabla Formula ¡es que hay altas posibilidades de que ni siquiera la puedas levantar! Sin embargo, puedes afectar la tabla, porque las Formula tienen una cubierta delgada que se raya fácilmente cuando tiras la vela durante una caída.
3. La Rio M es, probablemente, la mejor tabla de todos los tiempos para aprender y para sentirse cómodo planeando y usando cinchas de sujeción para los pies, aunque la GO 171 también es una buena opción.
4. Mi estrategia consistiría en tener velas de distintos tamaños, para llevar más allá tus límites respecto de cuánto poder logras controlar, independientemente de la velocidad del viento. Para empezar y para los vientos fuertes después, debes adquirir una vela menor a seis metros cuadrados... Tengo una calculadora para el *windsurf* que te da una idea del tamaño de la vela que necesitarás para planear a determinada velocidad de viento^[119].
5. Tus mejores posibilidades de planear las tendrás con velas más pequeñas cuando el viento sople fuerte, porque se requiere menos técnica para planear con una vela chica en viento fuerte que con una vela grande en viento ligero.

Esta información es oro sólido. Jim me aclaró varias dudas sobre la tabla y velas para comenzar. La amabilidad de Jim, al compartir sus consejos, me ahorró varios miles de dólares y bastante frustración.

Jim también me habló de Isthmus Sailboards^[120], tienda ubicada en Madison, Wisconsin, que vende equipo para surf a vela en línea. Llamé a Isthmus y Gary Stone, uno de los dueños, me ayudó a integrar la lista del equipo necesario para empezar.

He aquí lo que terminé ordenando:

- Una tabla Starboard Rio M.
- Una vela Chinook Powerglide, de 4,7 metros cuadrados.
- Un *boom* Chinook Sport AL.
- Un mástil Chinook de 400 centímetros.
- Una extensión de mástil Chinook US.
- Una base de mástil de una tuerca Chinook US.
- Un bungee tiravelas Chinook.
- Un traje entero Mystic Crossfire 5/4, más botas, guantes y capucha de neopreno.
- Un traje corto Mystic Crossfire 3/2, más guantes de neopreno Vibram.
- Un salvavidas personal Dakine Surface.

Inversión total: cerca de tres mil dólares, incluyendo el envío. El *windsurf* no es barato, pero si cuida el equipo, debe durar mucho tiempo.

Al principio, me inclinaba por la compra de una vela más grande, pero Gary ha enseñado a la gente durante dos décadas y pronto me hizo cambiar de idea. Las velas grandes, dijo, pueden ser buenas para condiciones de poco viento, pero solo si sabes usarlas.

La desventaja de las velas grandes es que son más pesadas y difíciles de sacar del agua. Si empiezo con una vela grande cuando aún trato de equilibrarme y de aprender a virar, odiaré mi vida, según palabras de Gary.

Para no odiarla, acepté su consejo y opté por la vela de menor tamaño. De nuevo, conviene seguir el consejo de los mentores más experimentados antes de tomar decisiones. Después de pasar media hora en el teléfono con Gary, envié la orden a Isthmus.

Resulta que las empresas de transporte de carga son bastante capaces y están más que dispuestas a llevar objetos del tamaño de una tabla de surf a donde uno quiera, así que obtener el equipo fue mucho más fácil de lo esperado. Gary estimó que mi pedido tardaría una semana en llegar, así que opté por empezar a educarme sobre cómo utilizar las cosas.

¡Apártense, marineros de agua dulce!

Un poco de investigación en Internet me condujo a varias fuentes de información sobre *windsurf*:

- *A Beginner's Guide to Zen and the Art of Windsurfing*, por Frank Fox (1988).
- *Windsurfing*, por Peter Hart (2005).
- *Learn Windsurfing in a Weekend*, de Phil Jones (1992).
- *Beginner to Winner* (DVD), por Jem Hall (2006).

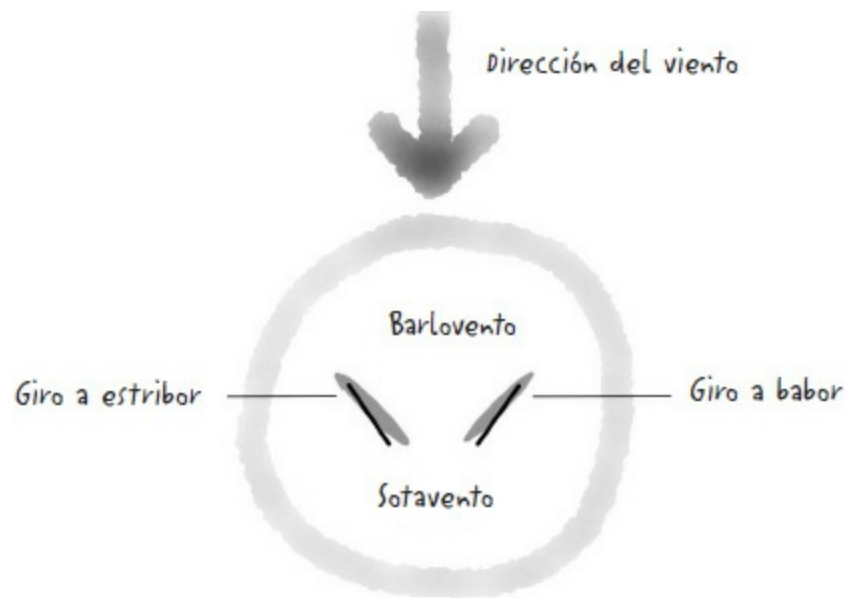
Estas fuentes hacen un buen trabajo al explicar los términos de *windsurf*, la teoría y la técnica básica. El DVD de Jem Hall es particularmente bueno al explicar cómo manejar la tabla en el agua: es más fácil explicar movimientos complejos demostrándolos. En este tema, los libros están en desventaja, porque incluso las mejores ilustraciones pueden resultar confusas comparadas con un video detallado.

Algo que me impactó de inmediato fue la mucha terminología de la marinería tradicional que se usa en el *windsurf*. Tiene sentido, pues una tabla es básicamente un velero con tabla en lugar de casco, pero aún así me sorprendió. A los surfistas se les llama «marineros», y los términos utilizados en la instrucción provienen directamente de la marinería.

Han pasado años desde que navegué a vela, así que necesito un repaso. Si no conoces los términos, leer libros de instrucción puede resultar frustrante, así que es importante conocer las ideas clave. Aquí te presento algunas:

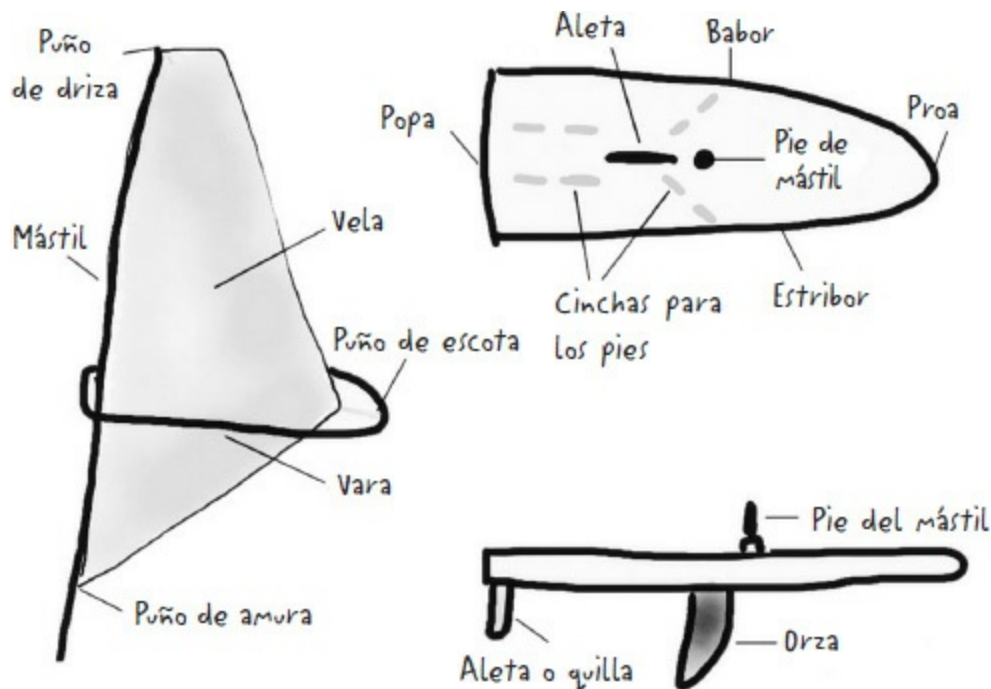
- Dirección del viento: direcciones como norte, sur, este y oeste no resultan muy útiles para navegar, dado que los vientos cambian de dirección. Las direcciones se dan respecto a la del viento: *barlovento* significa adentrarse en el viento prevaleciente y *sotavento* significa viajar con el viento.
- Dirección de la nave: igualmente, «derecha» e «izquierda» son términos relativos a la orientación de la nave. *Babor* significa el costado izquierdo de la nave y *estribor* el derecho.
- Frente y parte trasera: el frente de la nave se llama *proa* y la parte de atrás *popa*. Cuando algo está en dirección a la proa, se dice «de proa»; cuando algo pertenece a la popa, se dice «de popa».
- Cambios de dirección: los términos para girar la tabla son *virar* y *plantarse*. La diferencia entre ellos tiene que ver con qué parte de la tabla atraviesa primero el viento. Si te adentras en el viento (a

barlovento) y giras, estás virando. Si te alejas del viento (a sotavento) y giras, estarás plantándote.



Las cosas empiezan a ponerse difíciles al combinar esos términos. Si vas en contra del viento y giras la proa de la tabla a la derecha (a estribor), estás a babor, pues el viento viene de la izquierda de la nave (de babor).

Es confuso, así que me toma algunas horas decodificar las instrucciones del libro e imaginarlas en el agua. Este proceso de estímulo mental será útil cuando esté en el agua: al imaginar los conceptos aplicados a la nave real, me será más fácil recordarlos cuando sea necesario.



Las partes de la tabla también tienen nombres distintivos:

- La tabla: ya hemos visto la proa, la popa, babor y estribor, pero eso no es todo lo que debemos recordar. Las tablas suelen tener piezas adicionales, como una larga aleta en el centro (*orza*) y una aleta más pequeña en la popa (*aleta o quilla*). Las dos se adentran en el agua. En la parte superior de la tabla hay un lugar en el que la vela se une a la tabla (*pie de mástil*), asegurado a una base mecánica (*base del mástil*). También hay correas para los pies en la parte superior de la tabla, lo que ayuda a que el marinero mantenga el equilibrio en la posición correcta cuando la tabla se mueve realmente.
- La vela: las velas son apenas triangulares y se unen a la tabla por medio de un poste largo llamado *mástil*. Este se inserta en una especie de bolsillo de la vela (*funda de mástil*) que se extiende desde la base del mástil hasta arriba (*cabeza*). Una vez el mástil en su lugar, un asidero horizontal llamado *botavara* se coloca en posición apenas perpendicular al mástil y se asegura con una abrazadera. Los tres costados de la vela se atan a la botavara con soga de nailon (*línea*), que se ajusta envolviéndola alrededor del final de la botavara, para luego tejerla con una soga especial a un artefacto llamado *puño de*

amura. Cuando la vela está completamente enjarciada, el mástil se fija en el pie de mástil y la tabla está lista para navegar.

Para unir correctamente la tabla y el mástil se requiere de buen juicio y experiencia. Hay muchas formas de unir la base del mástil a la tabla, y cada variante cambia el centro de gravedad de la tabla. Dependiendo de la tabla, también puede haber distintas maneras de fijar la aleta y las correas para los pies, lo cual afecta el desempeño de la tabla en el agua.

Igualmente, la forma de enjarciar la vela influye en el comportamiento de la tabla. Las dos mayores variables son el puño de amura y el puño de escota, lo que se relaciona con qué tanto se aprietan las cuerdas que fijan la vela a la parte baja del mástil y a la parte más lejana de la botavara. Con menos tensión en estas partes, se forma un «bolsillo» en la vela, lo que significa que capta más viento, produciendo mayor poder de desplazamiento. Al tensar más estos cordajes, se reduce el tamaño del bolsillo y el poder potencial, lo que hace que la vela sea más manejable en condiciones muy ventosas.

¡Atrás, que estoy a punto de hacer cálculos físicos!

Las velas son artilugios fascinantes. En contra de lo que la gente cree, las tablas de *windsurf* (y los veleros en general) no se mueven porque el viento empuje la vela. La historia completa es un poco más compleja que eso.

Las velas funcionan creando diferencias en la presión del aire entre el frente y la parte posterior de la vela. En la mayoría de los casos, cuando el aire fluye alrededor de la vela, el flujo crea una zona de baja presión en el frente (hacia la proa), y una zona de alta presión atrás (hacia la popa). El efecto combinado de estas dos zonas de presión diferentes crea una fuerza que mueve la nave en la dirección de la zona de baja presión.

Como resultado, el viento *jala* la vela tanto como la *empuja*. Las alas de los aviones funcionan casi de la misma manera.

Es importante saber estas cosas cuando se está en el agua tratando de moverse. Si tu modelo mental sobre el funcionamiento de la vela es «aumenta la cantidad de vela sobre la que puede soplar el viento», la pasarás mal tratando de añadir poder, así como también al navegar en cualquier otra dirección que no sea aquella en la que sopla el viento.

Es difícil pensar en toda esta información, pero me alegra tener un poco de tiempo para aprender la teoría básica antes de navegar. Este deporte es, principalmente, una destreza motora, pero si no entiendes cómo funciona la nave, se corre el riesgo de pasar mucho tiempo practicando los movimientos equivocados.

Observar el viento

Además de leer libros, puse mucha atención al clima. Antes de interesarme en el surf a vela, no notaba el viento a menos que este soplara inusualmente fuerte. Ahora me fijo en el viento constantemente, y miro las puntas de los árboles para estimar su velocidad.

La tecnología viene en nuestra ayuda en este caso: sitios web como Weather Underground^[121], Windfinder^[122] y iWindsurf^[123] reúnen información sobre lugares de todo el mundo, haciendo mucho más fácil conocer la velocidad del viento en la zona que se requiera.

Por supuesto, el único que nos interesa es el que sopla sobre el agua en la que intentas deslizarte. Para revisar la velocidad del viento y la dirección de su movimiento en el lago, compré un anemómetro de bolsillo Kestral 3000^[124]. Este conveniente aparatito me permite conocer la velocidad actual del viento en el lago, así que es más preciso que la información disponible en línea. También mide la temperatura del agua y del aire, gracias a lo cual puedo asegurarme de no salir si el riesgo de padecer hipotermia es alto.

Me fijo en el viento durante unos días y comienzo a notar patrones. En el lago, los vientos ligeros tienden a soplar desde el este en las mañanas, luego aumenta la velocidad y sopla desde el nornoroeste por la tarde, alrededor de las 4 P. M. Dado que pienso hacerme al agua en la cara este del lago, será mejor hacerlo por la tarde. Las 4 es una hora que se aviene bien a mi horario en general: si planeo bien mi día de trabajo, puedo ir al lago durante una hora sin mayores distracciones.

No puedes practicar *windsurf* sin viento

También me percaté de que aquí los vientos no son muy predecibles: hay mucha variabilidad. Algunos días sopla el viento y otros no. Con base en lo

que he leído, si el viento corre a menos de 5 o 6 millas por hora, no vale la pena practicar.

Eso significa que debo contar con un plan alternativo. Por fortuna, tengo una idea: he visto gente en el lago con tablas para remar, lo cual también parece ser una actividad divertida que se ajusta a mis gustos. Si yo mismo produzco mi poder de desplazamiento con un remo, no necesito viento, así que este deporte de tabla y remo será una buena alternativa para los días en que no pueda practicar.

Ya tengo el equipo de flotación y dos trajes especiales, así que solo necesito una tabla decente y un remo largo. Después de investigar un poco, me decido por el siguiente equipo:

- Tabla Ocean Kayak Nalu 11.
- Remo Quick Blade Kahana Elite, de 80 pulgadas.

Me entrega la tabla el mismo camión de carga que me llevó la de *windsurf*. Entre mi tabla de remo y la otra puedo pasar el tiempo en el agua sin importar las condiciones del viento.

Reunir las piezas

Ahora que tengo todo el equipo, necesito unir la tabla y la vela, proceso que se llama *enjarciar*.

Por fortuna, Isthmus me ayuda en este problema. Enjarciar suele ser problemático para los nuevos dueños de este tipo de tabla, puesto que se trata de un proceso complicado. Cuando tratas de determinar dónde va cada cuerda o jarcia, es fácil confundirse.

Isthmus resolvió este problema con videos detallados sobre cómo enjarciar^[125], así que con este método puedes ver a un profesional enjarcando su nave en tiempo real. El video instructivo es ideal para este tipo de aprendizaje: miré las instrucciones varias veces y tomé notas antes de tratar de enjarciar mi equipo.

Armado con mis notas, pude enjarciar mi nave en media hora: no está mal para ser la primera vez. Ya tengo el traje térmico y la tabla está lista. Además, el viento sopla. Hora de zarpar.

El viaje inaugural

Llevo mi tabla y mi vela hasta la orilla del lago, uno el mástil con la base y meto todo al agua hasta que esta me llega a la cintura: suficiente profundidad para desplegar la orza. El viento sopla desde el noroeste a unas 12 millas por hora, creando algunos rizos en la superficie del lago. No importa. Estoy preparado.

Me subo a la tabla, me pongo de pie y luego trato de tomar el tiravelas, una cuerda atada al mástil. Al tirar de ella levanto la vela del agua hasta que está derecha, casi perpendicular a la tabla.

Cuando la vela deja atrás el agua, el viento me mueve naturalmente a sotavento, alejándome del viento. Eso me permite levantar la vela sin usar más fuerza, así que permanezco en el mismo lugar sosteniendo el mástil con ambas manos. Esta posición neutra es la postura de inicio, por lo que me siento bastante bien. Hasta ahora, lo estoy haciendo muy bien.

Lo que sigue es mover y soltar el mástil con una de mis manos para tomar la botavara y jalar. Al hacerlo, agrego poder y comienzo a moverme.

Nada pasa...

Al amparo de la madre naturaleza

No estaba preparado para lo que sucedió después.

Al jalar la botavara hacia mí, varias cosas sucedieron a la vez. La nave empezó a moverse más rápido de lo esperado y esto, combinado con el balanceo de la tabla bajo mis pies, me hizo entrar en pánico y perder el equilibrio. Mi centro de gravedad se movió hacia atrás y, antes de saber siquiera qué sucedía, estaba de vuelta en el agua.

Mis manos seguían sosteniendo la botavara, así que lo último que logré ver antes de sumergirme fue el mástil moviéndose muy rápidamente en dirección a mi cabeza.

¿Mencioné que no usaba casco?

No estoy seguro de cuál era mi idea sobre una caída al agua, pero ciertamente no tan violenta. Por fortuna, el mástil pasó a 30 centímetros de mi cabeza, pero la caída me sacó el aire y tragué bastante agua del lago. No sabía dónde estaba la superficie cuando me sumergí.

Por suerte, no estuve bajo el agua mucho tiempo, gracias al sistema de flotación de seguridad, que me llevó a la superficie tosiendo agua. Entonces detecté el siguiente problema: no podía ver nada.

Uso lentes. Los lentes de contacto no me funcionan bien, así que los abandoné hace años. Antes de entrar al agua, no pensé en mis lentes de sol graduados y, por supuesto, no usaba ningún tipo de correa de seguridad para mantenerlos atados a mi cabeza. Como resultado, la primera caída me arrancó los lentes de la cara. Actualmente, adornan el fondo del lago y jamás los volveré a ver.

Habían pasado cinco minutos desde mi estreno en el *windsurf* y ya estaba mojado, cegado y estremecido después de evitar por poco una conmoción.

Las cosas no marchan bien.

Aprieto los dientes y nado hacia la tabla. Cuando logro reorientarme, el viento la ha alejado a unos cinco metros de mí. Llego a la tabla, me subo y jalo la vela una vez más.

Autorrescate

Durante los 40 minutos que estuve en el agua ese día, me caí de todas las formas posibles: de espaldas, de frente, de lado. Tragué agua hasta causarme náuseas; las piernas, los brazos y la espalda me dolían por jalar la vela una y otra vez.

Decidí que era suficiente. Tenía frío, me sentía mal y estaba exhausto. Hora de ir a casa.

Desafortunadamente, no sé dirigir esta maldita tabla, y cada vez que alzo la vela, termino tragando más agua. El viento me lleva al sur, alejándome del sitio en que me hice al agua. No podría maniobrar para regresar allí aunque mi vida dependiera de ello.

Llegado ese momento, decidí poner en práctica una técnica de autorrescate que aprendí de uno de los libros. Me acosté en la tabla y dispuse la vela de manera que estuviera descansando sobre mí, con la punta de la vela orientada hacia la popa. Luego, comencé a remar con las manos, llevándome lentamente hacia la orilla.

Avanzaba muy lentamente. No estaba lejos, pero me tomó diez minutos de duro remar el regreso hasta la orilla. Los brazos no eran muy buenos

remos, ni siquiera en las mejores circunstancias, y estas no eran las ideales.

Cuando por fin llegué a tierra, mis pruebas se habían terminado. Dado que estaba bastante retirado del lugar en que había entrado al agua, mi primer intento terminó con una larga «caminata de la vergüenza». Tuve que hacer que mi cuerpo exhausto cargara el equipo de vuelta al coche. Conduje a casa y colapsé.

No fue un principio muy prometedor.

Análisis postraumático del suceso

Después de descansar, analicé mi salida. ¿Qué pasó? ¿Qué estuvo mal?

Primero, el viento era demasiado fuerte para un principiante. Para alguien con experiencia, probablemente estaba bien, pero siendo mi primera vez sobre la tabla, fue demasiado rápido. Lección aprendida.

Segundo, no tengo experiencia manteniendo el equilibrio sobre ningún tipo de tabla. Nunca he surfado, andado en patineta ni nada que requiera equilibrio sobre una tabla en movimiento. Las olas que creaba el viento la balanceaban y eso me asustó.

Tercero, añadir poder a la vela cambia el centro de gravedad de la tabla. Si no muevo mi cuerpo en la dirección correcta para compensar, es probable que me caiga. Si no suelto la vela a tiempo cuando pierdo el control, tendré que echarme un clavado. Debo entrenar para estar cómodo al cambiar de postura sobre la tabla, particularmente en las rachas de viento.

Cuarto, no tenía todo el equipo de seguridad necesario. Me quedaba clara la necesidad de un casco para protegerme del mástil en caso de caer. Ya había estado cerca una vez y había tenido suerte. No cometeré el mismo error dos veces: compraré un casco inmediatamente.

Aprender del pasado

Hay varias cosas que puedo hacer para que mi segunda experiencia no se parezca a la primera.

Primero, me aseguraré de que mi siguiente experiencia tenga lugar en un día con menos viento. Puede no ser tan emocionante, pero me permitirá practicar sin sentirme sobrepasado.

Segundo, puedo adaptarme a la sensación de estar parado en la tabla al usarla en el agua sin la vela, todo con la ayuda de mi remo de 80 pulgadas para maniobrar. Al restar la variable de la vela, puedo darme una idea de qué es mantener el equilibrio en la tabla y aprender qué tanto puedo inclinarme en cualquier dirección sin caer. No será una prueba perfecta, pues la vela cambiará el centro de gravedad de la tabla cuando esté instalada, pero es mejor que caer constantemente.

Tercero, puedo concentrar mi siguiente sesión de práctica en aprender a sentir el equilibrio de la vela. Si pongo atención, sentiré en qué dirección se mueve la vela y cómo afecta a la tabla. Puedo aprender a mover mi peso para equilibrar la fuerza de la vela y evitar caer de espaldas; también a soltar la vela a tiempo cuando una racha de viento fuerte la haga incontrolable.

Los siguientes días hay calma, lo que me permite tratar de aislar el entrenamiento ocupándome de mantener el equilibrio sobre la tabla. Éxito: al final de mi sesión de práctica ya no tengo tanto miedo y luego, en un día ventoso, tengo la oportunidad de practicar el equilibrio sobre la tabla cuando el agua no está tan tranquila. Aún tengo que acostumbrarme a mirar abajo y ver que el agua se mueve bajo mis pies, pero el simple hecho de experimentar la sensación ayuda mucho a calmar mis nervios.

Luego, se presenta un día soleado y cálido con una brisa aceptable. Enjarcio, me hago al agua y jalo la vela hasta la posición neutral. ¿Estaré a punto de repetir el día uno?

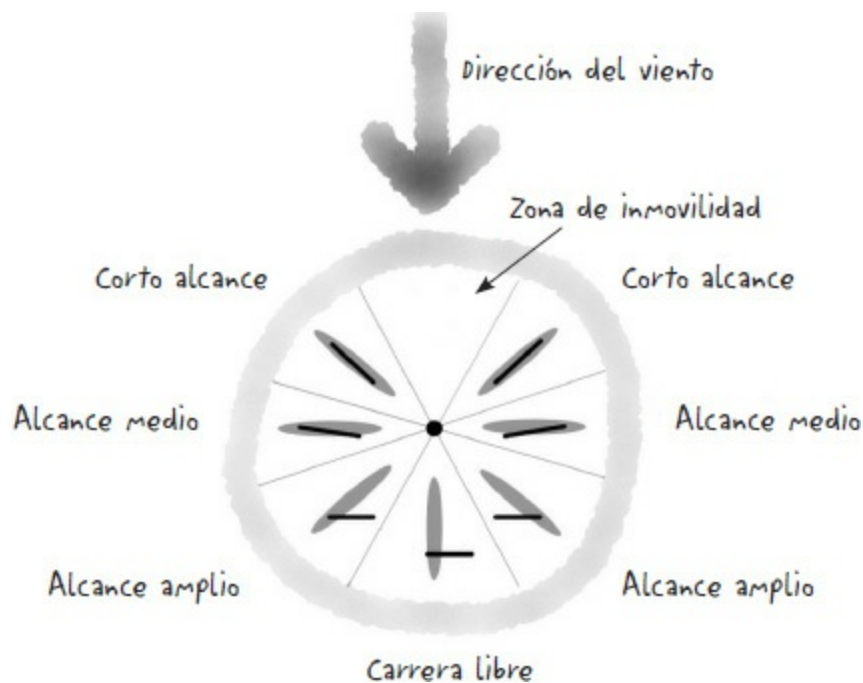
Para nada: solo me caí dos veces ese día. Mi práctica de equilibrio sobre la tabla ayudó mucho y no caí cuando la tabla se salió de control. Usé mucho el tiravelas ese día, pero está bien: es mejor que tragar agua del lago. Practiqué añadir poder a la vela y, de pronto, me estaba moviendo.

¿Cómo das la vuelta a esta cosa?

Hacer que la tabla se mueva sin caermes es una victoria, pero me lleva a tener que resolver un problema importante: ¿cómo dar la vuelta?

Experimenté con lo aprendido de libros y DVD: girar la tabla bajo mis pies mientras sostengo la vela en posición neutral ayudó mucho. También me ayudó a recordar que las tablas de navegación (y los veleros en general) no se pueden adentrar directamente contra el viento.

Existe una zona «muerta» o de «exclusión» que se extiende unos 45 grados a cada costado del barlovento. Trata de moverte en esa zona y te encontrarás detenido e incluso comenzarás a moverte hacia atrás. Para adentrarte en el viento, debes hacer que la tabla vaya a unos 45 grados en cualquier dirección lateral, para luego moverte en el otro sentido tras desplazarte un rato. Al moverte hacia un costado y luego hacia el otro, puedes avanzar en zigzag hasta tu destino, incluso si el viento sopla en contra.



También es importante virar la tabla cuando se está bajo el poder de la vela. Cuando estás en movimiento, el cambio de dirección de esta a la derecha o a la izquierda modifica el *centro de esfuerzo*, el punto central de toda la fuerza que el viento genera en la vela, en relación con el *centro de resistencia lateral* de la tabla, en foco de resistencia entre la tabla y el agua. Al mover la vela en cualquier dirección, la relación entre estos puntos focales cambia y la tabla gira para compensar.

Cuando te subes de nuevo a la tabla después de una caída, es común encontrarte en una posición extraña. Idealmente, la tabla debe estar perpendicular al viento, con la vela apuntando en su dirección. En esta posición puedes jalar la vela para sacarla del agua sin demasiados problemas.

No obstante, si la vela apunta contra el viento, estás en peligro de que la botavara te dé en la cara cuando la levantes y la vela capture el viento. Igualmente, puedes encontrarte con la tabla apuntando a la zona de no avance, justo contra el viento.

Para compensar, es mejor usar los pies para rotar la vela, de manera que esté perpendicular respecto de la tabla sin importar la dirección del viento. Una vez que vela y tabla están perpendiculares entre sí, puedes jalar la vela lentamente, permitiendo que el viento rote la tabla hasta la posición correcta. Ya que la vela apunta a la dirección del viento, puedes levantarla completamente y avanzar en la dirección que quieras^[126].

Después de algunas horas de práctica, empiezo a entender las cosas. No soy muy ágil o rápido, y mis vueltas no son hermosas, pero puedo echar al agua la tabla, navegar hasta la mitad del lago y regresar más o menos a mi punto de partida. Comparada con mi primera experiencia, es una gran mejora.

Para todo hay una época

El otoño llega rápido y fuerte a la cordillera Front de Colorado, por lo regular a fines de septiembre. Un día estamos a 23,8 grados bajo un sol radiante, y al siguiente día tenemos una temperatura de menos un grado y a punto de nevar.

He estado en el agua tanto como he podido cada tarde, tratando de maximizar el tiempo de práctica antes de que cambie el clima. La temperatura del agua permanece estable en 20 grados centígrados: un tanto fría, pero no en la zona de peligro por hipotermia. Con un traje especial, el golpe inicial de entrar en el agua es la peor parte: después de eso, las caídas no son tan desagradables.

En los días en que el viento supera las 6 millas por hora, enjarcio la tabla. Cuando el viento está calmo, mejor remo sobre la tabla.

He estado remando más sobre la tabla que practicando *windsurf*: es difícil navegar cuando no hay viento suficiente. Es frustrante, porque quiero practicar tanto como pueda, pero no se puede negociar con la madre naturaleza. O el viento sopla con la fuerza suficiente como para moverme en el agua, o no lo hace y punto.

Una mañana, la temperatura baja a 4,4 grados. En tres días, la temperatura

del lago baja a 10 grados centígrados. Con mi traje térmico 5/4 pude sobrevivir por un tiempo, pero el riesgo de padecer hipotermia o ahogarme no está entre mis prioridades. Para mí, la temporada de *windsurf* ha terminado.

Sumo las horas de práctica y resulta que me quedé un poco corto de mi objetivo: en total practiqué nueve horas, muchas menos de las 20 que debería haber entrenado llegado este momento. Sin embargo, pasé más de 20 horas remando sobre la tabla.

Me encuentro con una importante lección: el medio ambiente importa. Sería fácil sentirme desilusionado por no haber logrado llegar a mi objetivo de práctica, pero mi deseo de aprender no puede hacer que sople el viento. Incluso si esto fuera posible, no es sabio pasar mucho tiempo en el agua, pues el cansancio es un gran factor de riesgo. Como resultado, cada día pude pasar entre 30 y 40 minutos en el agua.

Aunque me faltó para llegar a las 20 horas, aprendí muchísimo. Sé enjarcar, aprendí a ensamblar la tabla, aprendí a maniobrar con variadas condiciones de viento y practiqué en incontables ocasiones volver a subirme a la tabla después de caer. Puedo levantar la vela, ir en la dirección deseada y controlar la tabla con poder en la vela. Puedo virar cuando quiero y retornar a la orilla con seguridad cerca de la zona en que entré al agua. Es mucho progreso para tan poco tiempo.

Además, aprendí una minidestreza: remar en tabla. La disposición del equipo y la técnica no se acercan siquiera a la complejidad y dificultad del surf a vela, pero es divertido. Incluso cuando el viento no sopla, disfruto deslizarme sobre el agua persiguiendo patos de un lado al otro del lago. Es agradable remar mientras el sol se oculta, y entre el ejercicio y la belleza del paisaje, esta es una gran manera de relajarme.

Repaso del método

Hagamos un repaso del método esencial para aprender *windsurf*:

- Adquirí el equipo necesario: una tabla, una vela, trajes térmicos, salvavidas, casco y otras cosas importantes para la seguridad.
- Aprendí a enjarcar la tabla y la vela, así como a desensamblar, mantener, transportar y guardar el aparejo con seguridad.

- Aprendí a protegerme de los riesgos mayores, incluyendo ahogamiento, conmoción e hipotermia.
- Aprendí a hacerme al agua desde la orilla, a levantar la vela en una posición neutral y añadirle poder para comenzar a moverme.
- Aprendí a virar y a ubicar la vela para moverme en diversas condiciones de viento.
- Aprendí a no caer de la tabla, a subirme de nuevo y a levantar la vela cuando me caía.

¿Y ahora qué?

Cuando llegue el momento en que estés leyendo esto, habrá llegado de nuevo la temporada de *windsurf*. Estaré allí en cuanto el clima y las condiciones del agua se alejen de la zona de hipotermia. Volveré a practicar el enjarcado, el equilibrio y las vueltas.

Volveré a adquirir rápidamente el nivel básico que tenía al término de la temporada anterior, así que me preparo para el siguiente reto: hacer que la tabla pueda hidroplanear, lo que aumenta bastante la velocidad. El «planeo» solo es posible en condiciones de viento fuerte, y mayor velocidad aumenta también el riesgo de accidente y lesión, así que debo sentirme muy cómodo con las cuestiones básicas antes de intentarlo.

También usaré una vela más grande. Las condiciones de viento generalmente bajo y variable hacen que el uso de una vela mayor sea una buena idea. Así, cuando me sienta cómodo utilizando la vela de 4,7 metros cuadrados, comenzaré a practicar con la de 7,5. Entre las dos debo cubrir todas las posibilidades: utilizaré la grande para los días con poco viento y la pequeña para los días ventosos.

En suma, este deporte es muy divertido. Disfruto el tiempo que paso en el agua, en particular ahora que no me caigo tanto. Espero impaciente la siguiente estación y procuraré sacar ventaja de los días con viento cuando lleguen.

No puedo controlar a la madre naturaleza, pero sí controlar mi manera de practicar cuando las condiciones son favorables. Con eso basta.

Epílogo

El logro parece estar conectado con la acción. Los hombres y las mujeres exitosos se mantienen en movimiento. Cometen errores, pero no se dan por vencidos.

Conrad Hilton, fundador de la cadena hotelera Hilton.

En menos de un año, aprendí seis destrezas complejas.

No soy un genio o un fenómeno de la naturaleza. No tengo talentos especiales. No dejé mi trabajo. No boté todo para irme al otro lado del mundo. No ignoré a mi familia.

Me limité a reservar una hora o algo así de cada día para practicar. Y practiqué de modo inteligente. Las destrezas que empezaron siendo un completo misterio se tornaron comprensibles en cuestión de días, y hasta en horas. Solo tuve que investigar un poco para después reservarme unas 20 horas de práctica consistente, concentrada y deliberada.

Mejor todavía, mi práctica se convirtió en rutina: estas destrezas son ahora parte de mi vida diaria. Aprender primero las subdestrezas principales permite un fácil progreso. Cuando leas estas líneas, seré todavía *mejor* en dichas áreas. Cuánto mejoraré dependerá del tiempo que pase practicando.

Si quieres adquirir una nueva destreza, debes practicar. No hay otra opción.

Puedes prepararte. Puedes investigar. Puedes eliminar las distracciones y alterar el ambiente para facilitar la práctica. Puedes encontrar maneras para que tu práctica sea más efectiva o eficiente. Pero, a fin de cuentas, debes practicar.

El que parece ser el camino largo, termina siendo el más corto. No existen los atajos que permitan progresar sin practicar. Si no hay práctica, no hay adquisición de destreza. Tan simple como eso.

¿Por qué no practicamos? Es simple: estamos ocupados y nos da miedo. Shakespeare lo dijo bien hace tiempo, en una obra titulada *Medida por medida*: «Nuestras dudas son traidoras, y nos hacen perder lo que podríamos ganar por el miedo a intentar».

La mayor barrera para la rápida adquisición de destrezas no es ni física ni intelectual: es emocional. Hacer algo nuevo siempre es incómodo al principio, y es fácil desperdiciar una tonelada de tiempo y energía pensando en practicar en vez de hacerlo.

Por fortuna la barrera de la frustración es fácil de superar: la adquisición de destrezas siempre se siente mejor de lo que es en realidad. Al reservar tiempo para practicar, al investigar un poco y soportar la incomodidad inicial, siempre progresarás de manera importante durante las primeras 10 a 20 horas de práctica. Para obtener la recompensa solo hace falta un poco de esfuerzo, persistencia y firmeza de carácter.

No necesitas escoger muchas destrezas por adquirir: elige solo una. Toma una destreza de tu lista y comprométete a intentarlo. Aprende ese idioma, toca ese instrumento, explora el juego, trabaja en ese proyecto, cocina ese platillo, crea ese arte. Es más fácil de lo que parece.

Comprométete de antemano a practicar esa habilidad durante una hora al día a lo largo del siguiente mes. Cuando comiences a practicar, siempre mejorarás más rápido de lo esperado. Divide la destreza, reserva el tiempo, intenta cosas nuevas, y tu cerebro comenzará a captar la técnica automáticamente: eso es justo lo que hacen los cerebros. Cuando te atasques o confundas, intenta poner a prueba una nueva forma de hacer las cosas.

Recuerda: una vez que has comenzado, no puedes detenerte hasta alcanzar el nivel de desempeño que constituye tu meta o la marca de las 20

horas. Lucha si es necesario, pero no te detengas. Demuestra tu fuerza de voluntad y sigue intentando. Lo lograrás: solo debes practicar.

Una última consideración: el único momento en que puedes elegir practicar es hoy.

No mañana, no la semana entrante, no el mes siguiente o el año venidero. Hoy.

Cuando te levantas por la mañana, tienes la oportunidad de elegir. Puedes dedicar tu tiempo a adquirir destrezas que harán que tu vida sea más exitosa, disfrutable y provechosa... o puedes desperdiciar el tiempo haciendo otra cosa.

¿Qué harás hoy?

Reconocimientos

Para Kelsey y Lela: las amo. Gracias por todo.

Para Dave, Sheri y Zulema: gracias por las discusiones sobre Lela que hicieron posible este libro.

Para Leslie Kaminoff, Derek Sivers, Jim Douglass y Gary Stone: gracias por su generosidad y ayuda. He aprendido mucho de ustedes.

Para Lauren Baker: gracias por mejorar las ilustraciones de este libro para ser impresas.

Para Lisa DiMona: soy el más afortunado al tener tu ayuda y apoyo.

Para Adrian Zackheim y Joel Rickett: ¡viva Portfolio!

Para Emily Angell: usas la pluma roja con clase. Gracias por tu visión y diligencia.

Para Will Weisser, Margot Stamas, Richard Lennon y Allison McLean: los libros sin lectores no son muy divertidos. Gracias por todo lo que hacen para difundir la palabra.

Para Joe Perez y Dan Donohue: cuando la gente juzga los libros por su cubierta, siempre hacen grandiosos a los autores. Aprecio su duro trabajo para dar vida a mi idea.

Para Bria Sandford, Samantha LaBue, Sarah Katie Coe, Thomas Dussel, y para todo el equipo de producción y distribución de Penguin: gracias por todo lo que han hecho y por todo lo que siguen haciendo para llevar este libro a los lectores de todo el mundo.

Para ti, querido lector: tú eres la razón por la que hago lo que hago. Espero disfrutes este libro y que la información te resulte útil en los años venideros. ¡Buena suerte!



JOSH KAUFMAN es uno de los 100 autores de negocios más importantes del mundo y ha aparecido como el autor número uno en ventas en Business and Money, según el *ranking* de Amazon.com.

Josh lleva a cabo una investigación independiente sobre cursos de negocios, emprendimiento, adquisición de habilidades, productividad, psicología del comportamiento, diseño de sistemas y optimización, y se centra en ayudar a las personas a aprender cómo ganar más dinero, hacer más y divertirse más en la vida diaria.

El enfoque único y multidisciplinario de Josh para el dominio comercial y la rápida adquisición de habilidades ha ayudado a millones de lectores en todo el mundo a aprender conceptos y habilidades esenciales en sus propios términos.

La investigación de Josh ha sido presentada por *The New York Times*, The BBC, *The Wall Street Journal*, *The Atlantic*, *Fortune*, *Forbes*, *Time*, *BusinessWeek*, *Wired*, *Fast Company*, *Financial Times*, HarvardBusiness.org, *Inside Higher Ed*, *Lifehacker*, MarketWatch, *The Independent*, Bloomberg TV, PBS Next Avenue, CCTV y el Dr. Sanjay Gupta de CNN.

Josh ha sido ponente destacado en The Aspen Ideas Festival, Universidad de Stanford, World Domination Summit, Pioneer Nation, Microconf, BaconBiz, Google e IBM.

Notas



[1] Ericsson, K. Anders, Neil Charness, Paul J., Feltovich y Robert R. Hoffman (editores), *The Cambridge Handbook of Expertise and Expert Performance*, Cambridge University Press, 2006. <<

[2] <http://www.chirunning.com>. <<

[3] Este tipo de programas universitarios suelen durar al menos cuatro años por mero convencionalismo y porque así se benefician las escuelas: obtienen más dinero por cada año que el alumno se encuentre inscrito. Esto no quiere decir que sea imposible completar hasta los programas más demandantes en menos tiempo: Scott H. Young completó el curso de ciencias de la computación del Tecnológico de Massachusetts en menos de un año. Ver <http://www.scotthyoung.com/blog/mit-challenge>. <<

[4] VanLehn, Kurt, «Cognitive Skill Acquisition», *Annual Review of Psychology* 47, número 1, 1996, páginas 513-539. <<

[5] Si puedes comprender mi frustración, te recomiendo que leas el libro *The Renaissance Soul: Life Design for People with Too Many Passions to Pick Just One*, de Margaret Lobenstine (Nueva York, Harmony, 2006). Esta obra me ayudó a darme cuenta de que mis intereses diversos constituyen una fortaleza y de que es posible ordenar mis esfuerzos para sacar ventaja de mi inclinación natural a aprender muchas cosas de manera simultánea. <<

[6]

[http://www.newyorker.com/reporting/2011/10/03/111003fa_fact_gawande.](http://www.newyorker.com/reporting/2011/10/03/111003fa_fact_gawande)

<<

[7] Yo utilizo un cronómetro de intervalos programables de la marca Enso Pearl: <http://www.salubrion.com/products/ensoparl/>. Puedes usar cualquier cronómetro, incluyendo los básicos que se usan en la cocina, pero el Enso dispone de intervalos incorporados y tiene sonidos atractivos, lo cual es más importante de lo que parece. Cuando practicas mucho, los timbres electrónicos aburren pronto. <<

[8] Snoddy, George S., «Learning and Stability: A Psychophysiological Analysis of a Case of Motor Learning with Clinical Applications», *Journal of Applied Psychology* 10, número 1, 1926. <<

[9] Newell, Allen y Paul S. Rosenbloom, «Mechanisms of Skill Acquisition and the Law of Practice», *Cognitive Skills and Their Acquisition*, 1981, páginas 1-55. <<

[10] Logan, Gordon D., «Toward an Instance Theory of Automatization», *Psychological Review* 95, número 4, 1988, página 492. <<

[11] <http://ankisrs.net>. <<

[12] <http://www.supermemo.com>. <<

[13] <http://smartr.be>. <<

[14] Este no es un libro sobre técnicas de estudio académico, pero ya que te has esforzado en leer las notas, te recomiendo un método de estudio engañosamente simple y que funciona de maravilla: toma una idea, saca una hoja de papel en blanco y luego escribe la explicación completa de la idea usando un lenguaje muy simple, como si enseñaras a un principiante. Los huecos en tu formación serán muy evidentes, lo que te facilitará repasar. Scott H. Young, investigador de habilidades para el estudio, llama a esto la «técnica Feynman» (en honor del reconocido físico Richard Feynman), y es muy efectiva. Para más detalles, te recomiendo visitar: <http://www.scottyoung.com/learnonsteroids/grab/TranscriptFeynman.pdf>. <<

[15] Hay algo muy divertido en echarte a correr a toda velocidad en dirección a una barrera que te llega hasta la cintura, para saltarla sin romper el ritmo de la carrera. Lo más cerca que estuve de sentirme Superman fue el momento en que mi pie se apoyó sólidamente en el centro del obstáculo durante una carrera. El madero del obstáculo se cimbró y yo lo salté sin perder velocidad. Fue fantástico. <<

[16] http://www.youtube.com/watch?v=IMCl_RH_b3k. <<

[17] <http://www.nytimes.com/2009/07/26/magazine/26FOB-consumed-t.html>.

<<

[18] <http://www.gilhedley.com>. <<

[19] Puedes ver el famoso discurso de Gill aquí: <http://www.youtube.com/watch?v=FtSPtkSug>. Debes saber que en el video aparece un cadáver humano, así que tal vez quieras adelantar esa parte si eres muy sensible. <<

[20] Sarno, John E., *Healing Back Pain: The Mind-Body Connection*, Grand Central Life and Style, Nueva York, 2010. <<

[21] Intentaré explicar miles de años de historia muy compleja en unos cuantos párrafos. Los libros sobre este tema pueden llenar bibliotecas enteras, así que, por favor disculpa mi brevedad. Si estás interesado en un repaso más detallado, te recomiendo la lectura de *The Great Transformation: the Beginning of our Religious Traditions*, Anchor, Nueva York, 2007. <<

[22] Armstrong, Karen, *The Great Transformation: The Beginning of Our Religious Traditions*, Anchor, Nueva York, 2007. <<

[23] Desikachar, T. V. K. y R. H. Cravens, *Health, Healing, and Beyond: Yoga and the Living Tradition of Krishnamacharya*, North Point Press, Nueva York, 1998. También existe una breve biografía disponible en <http://www.yogajournal.com/wisdom/465>. <<

[24] Aún se debate cuánto de la doctrina de Krishnamacharya pertenece a fuentes no hatha, como la gimnasia británica. Como mínimo, muchas de las posturas y movimientos que incluyó en sus secuencias tienen una fuerte semejanza con la gimnasia y el entrenamiento militar de la época. <<

[25] http://www.nytimes.com/2012/01/08/magazine/how-yoga-can-wreck-your-body.html?_r=3&pagewanted=all. <<

[26] <http://www.manduka.com/us/shop/categories/products/gear/manduka-pro-black-sage>. <<

[27] <http://orthoinfo.aaos.org/topic.cfm?topic=A00063>. <<

[28] <http://personalmba.com/best-business-books>. <<

[29] <http://wordpress.org>. <<

[30] Para los curiosos: mi pila informática estándar consistía en PHP% con PHPFastCGI, NGINX, APC, MSMTP, y WP-Supercache en un Slicehost.com VPS corriendo un Ubuntu 8.04 LTS, todo con archivos de configuración establecidos por el usuario. <<

[31] <http://jekyllrb.com>. <<

[32] <http://github.com>. <<

[33] Por alguna razón, casi todos los tutoriales de programación comienzan mostrándote cómo desplegar o imprimir la frase «¡Hola, mundo!». <<

[34] <http://stackoverflow.com>. <<

[35] <http://news.ycombinator.com>. <<

[36] <http://rubyonrails.org>. <<

[37] <http://sinatrarb.com>. <<

[38] <http://37signals.com>. <<

[39] <http://rubysource.com/rails-or-sinatra-the-best-of-both-worlds>. <<

[40] <http://paulstamatiou.com/how-to-wordpress-to-jekyll>. <<

[41] <https://github.com/sstephenson/rbenv>. <<

[42] En la práctica, el «hackeo» no se parece nada a lo que muestran en las películas, lo que me resulta muy decepcionante. <<

[43] <https://toolbelt.heroku.com>. <<

[44] <http://www.heroku.com>. <<

[45] <http://git-scm.com>. <<

[46] Las versiones de Ruby anteriores a la 1.9.3-pl25 requerían un programa llamado GCC para completar la instalación. GCC está disponible en <https://github.com/kennethreitz/osx-gcc-installer>. <<

[47] Los programadores acuñaron el acrónimo «RTFM», que significa «*read the (freaking) manual*» (lee el maldito manual), respuesta estándar a todas las preguntas relacionadas con temas que se abordan en la documentación oficial del programa. <<

[48] <http://www.ruby-lang.org/en/documentation>. <<

[49] <http://0xfe.muthanna.com/rubyrefresher>. <<

[50] <https://code.google.com/p/ruby-security/wiki/Guide>. <<

[51] El término usado para referirse a la modificación avanzada de los principales objetos, clases y métodos de Ruby es *metaprogramación*. Consulté un libro llamado *Metaprogramming Ruby: Program Like the Ruby Pros*, de Paolo Perrotta (Raleigh, Carolina del Norte, Pragmatic Bookshelf, 2010), y estaba muy por encima de mi nivel por el momento. Lo primero es lo primero. <<

[52] <http://www.ruby-doc.org/core-1.9.3/index.html>. <<

[53] <http://ruby.learncodethehardway.org>. <<

[54] <https://devcenter.heroku.com/articles/keys>. <<

[55] <https://devcenter.heroku.com/articles/ruby>. <<

[56] <https://devcenter.heroku.com/articles/rack>. <<

[57] <http://macromates.com>. <<

[58] <https://devcenter.heroku.com/articles/bundler>. <<

[59] <http://www.sinatrarb.com/intro>. <<

[60] <http://backpackit.com>. <<

[61] <http://tom.preston-werner.com/2010/08/23readme-driven-development.html>. <<

[62] <http://www.postgresql.org>. <<

[63] <http://datamapper.org>. <<

[64] <http://stackoverflow.com/questions/1152299/what-is-an-object-relational-mapping-framework>. <<

[65] <http://en.wikipedia.org/wiki/SQL>. <<

[66] <http://www.sqlite.org>. <<

[67] <http://pow.cx>. <<

[68] <https://github.com/rodreegez/powder>. <<

[69] <http://twitter.github.com/bootstrap>. <<

[70] No tengo idea de por qué se llama así, y claro que me parece raro. En inglés, el término *slug* significa «babosa» [N. del T.]. <<

[71] Ver <http://www.regular-expressions.info> para obtener ejemplos de expresiones regulares comunes. <<

[72] <http://daringfireball.net/projects/markdown>. <<

[73] <http://www.httpwatch.com/httpgallery/authetification>. <<

[74] Usar SSL en un dominio a la medida, fuera de Heroku, es más complicado: debes pasar por un largo proceso de verificación de identidad y obtener un «certificado» que asegura cada sesión de usuario. <<

[75] <https://github.com/SFEley/sinatra-flash>. <<

[76] Un hecho divertido: Tiger Woods es el único golfista profesional en la historia que ganó el Torneo de Maestros de Estados Unidos con tres diferentes *swings*. Tiger reentrenó con Butch Harmon después de su victoria en el torneo de 1997, ganó de nuevo en 2001 y 2002, y reentrenó nuevamente en 2002 con Hank Haney, luego ganó en 2005. Más recientemente, Tiger comenzó su tercer reentrenamiento para el *swing* de salida con Sean Foley en 2011, el cuarto desde que es golfista profesional. <<

[77] Imagen de teclado en vectores vía <http://wowvectors.com/object/mac-keybord-vector>. Distribuida bajo licencia de Creative Commons Attribution 3.0. <<

[78] Para una historia muy interesante del desarrollo de la máquina de escribir QWERTY, ver «The Fable of the Keys», por S. J.Liebowitz y Stephen E. Margolis, disponible en <http://www.utdallas.edu/liebowit/keys1.html>. <<

[79] <http://www.google.com/patents?id=qSVdAAAAEBAJ>. <<

[80] <http://mkweb.bcgsc.ca/carpalx>. <<

[81] <http://mkweb.bcgsc.ca/carpalx/?colemak>. <<

[82] <http://colemak.com>. <<

[83] Otros sistemas operativos pueden requerir la instalación de un pequeño paquete de *software* que permite el uso de esta disposición de teclado. Los paquetes para los sistemas más populares se pueden conseguir en <http://colemak.com>. <<

[84] <http://www.typematrix.com/2030/features.php>. <<

[85] En los teclados TypeMatrix puedes activar el modo de *hardware* de Colemak presionando la tecla de función+F5. <<

[86] Esto no funcionará con todos los teclados, ya que muchos modelos tienen teclas de diferente tamaño en las filas superior e inferior. Si ese es el caso, probablemente sea más fácil conseguir otro teclado u ordenar las calcomanías que pueden pegarse en cada tecla. <<

[87] <http://www.typeonline.co.uk/typingspeed.php>. <<

[88] El material de la prueba viene de libros que son del dominio público y se distribuyen gratuitamente en línea vía el Proyecto Gutenberg. Ver <http://www.gutenberg.org>. <<

[89] <http://www.mavisbeacon.com>. <<

[90] <http://typingtrainer.sourceforge.net>. <<

[91] <http://github.com/wwwtyro/keyzen>. <<

[92] Luft, Andreas R. y Manuel M. Buitrago, «Stages of Motor Skill Learning», *Molecular Neurobiology* 32, número 3, 2005, páginas 205-216.
<<

[93] Walker, Matthew P. y Robert Stickgold, «*It's Practice, with Sleep, That Makes Perfect: Implications of Sleep-Dependent Learning and Plasticity for Skill Performance*», *Clinics in Sports Medicine* 24, número 2, 2005, páginas 301-317. <<

[94] <http://www.daskeyboard.com/model-s-ultimate-silent>. <<

[95] <http://type-fu.com>. <<

[96] Milton, James, *Measuring Second Language Vocabulary Acquisition*, Bristol, Reino Unido, Multilingual Matters, 2009. <<

[97] <http://code.google.com/p/amphetype>. <<

[98] <http://norvig.com/ngrams>. <<

[99] En mi opinión, el aspecto más desilusionante de la realidad es la falta de magia: si esto fuera de otra manera, yo sería al menos un mago de nivel 80. Pero desafortunadamente todavía no puedo manipular el universo. <<

[100] Otro hecho divertido: si revuelves un mazo de cartas a conciencia, la secuencia de cartas resultante tampoco ha sido vista jamás en la historia del universo. Un factor de 52 es un número muy grande: 8065 veces 10^{67} , o más de 80 undecillones de combinaciones posibles. <<

[101] Otros juegos famosos incluyen al Juego del Vómito con sangre, pues uno de los maestros que jugaban murió tras la partida, y el Juego de la Bomba atómica, ya que el juego fue interrumpido por la explosión de la bomba atómica en Hiroshima. Cuando la explosión terminó, los jugadores hicieron una pausa para comer, remplazaron las piedras en el tablero y continuaron con el juego. Ver <http://senseis.xmp.net/?FamousGoGames> para saber de más juegos históricos. <<

[102] <http://www.ymimports.com>. <<

[103] <http://senseis.xmp.net>. <<

[104] Sí, ese es el origen del nombre de la consola de juego Atari. <<

[105] <http://senseis.xmp.net/?TheTenGoldenRulesList>. <<

[106] http://diiq.org/five_stone_questions.html. <<

[107] <http://cdp.sagepub.com/content/14/6/317.short>. <<

[108] <http://www.crowhillguitars.com>. <<

[109] <http://www.daddario.com/DADProductDetail.page?ActiveID=3769&productid=264>. <<

[110] <http://www.axisofawesome.net>. <<

[111] [http:// www.ukuleles.com/Technology/strings.html](http://www.ukuleles.com/Technology/strings.html). <<

[112] <http://www.kiwiukulele.co.uz/Kiwi-Ukulele-Chord-Chart-pdf>. <<

[113] <http://jimbodouglass.blogspot.com>. <<

[114] Si quieres saber cómo se ven las cosas al ir verdaderamente rápido en una tabla de competencia, checa el video de Jim: <http://jimbodouglass.blogspot.com/2010/01/formula-windsurfing-boom-mount-video.html>. <<

[115] Los Boy Scouts llaman a esto el «sistema amigo», y constituye la regla cardinal para la seguridad en el agua. <<

[116] Este tipo de trajes captan agua entre el traje y el cuerpo; el agua es calentada por el cuerpo y actúa como capa aislante. Con temperaturas por debajo de los 10 grados, es mejor usar un traje seco, que evita que el agua toque la piel. <<

[117] <http://www.dbw.ca.gov/Pubs/Windsurf/index.htm>. <<

[118] <http://jimbodouglass.blogspot.com/2008/02/top-16-winsurfing-questions-answered.html>. <<

[119] <http://jimbodouglass.blogspot.com/2010/11updated-windsurf-calculator-online-html>. <<

[120] <http://www.isthmussailboards.com>. <<

[121] <http://www.wunderground.com>. <<

[122] <http://www.windfinder.com>. <<

[123] <http://iwindsurf.com>. <<

[124] <http://kestrelmeters.com/products/kestrel-3000-wind-meter>. <<

[125] http://www.isthmussailboards.com/info_technical_help.asp. <<

[126] Para obtener una guía detallada sobre cómo hacer esto, incluyendo diagramas, ver <http://jimbodouglass.blogspot.com/2012/10/beginner-windsurfing-how-to-reorient-html>. <<