

CODIFICACIÓN

Contacta con
nosotros

MENÚ

Por que aprender a ^{nosotros}
codificar es tan difícil

Por Erik Trautman

11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 103

Lo que todo principiante necesita saber absolutamente sobre el viaje que se avecina

Quincy Larson era solo un "tipo con traje en una oficina" y decidió que quería aprender a codificar. Así que preguntó por ahí. Comenzó recogiendo un poco de Ruby y luego se encontró hojeando otros idiomas como Scala, Clojure y Go. Aprendió Emacs, luego Vim e incluso el diseño de teclado Dvorak. Aprendió Linux, incursionó en Lisp y codificó en Python mientras vivía en la línea de comandos durante más de medio año.

Como una hoja en un tornado, el consejo que Quincy recibió lo sacudió primero de una manera y luego de otra y luego de otra hasta que finalmente tomó "todos los programas de cursos en línea imaginables". Al final de todo, a pesar de haber [conseguido](#) finalmente [un trabajo de desarrollo de software](#), Quincy:

... estaba convencido de que los programadores aparentemente normales con los que me encontré eran en realidad sociópatas que habían experimentado, luego reprimido, el trauma de aprender a codificar.

Ay. ¿Te suena familiar?

Aprende a codificar con Thinkful

Tome el camino probado hacia una carrera de altos ingresos con tutoría y apoyo profesional, formas flexibles de pago y aprendizaje basado en proyectos en el mundo real.

[VER NUESTRO PLAN DE ESTUDIOS](#)

Fase I: La Luna De Miel De La Mano

Es muy difícil culpar a alguien por entrar en la industria de la programación con expectativas escandalosas.

Por un lado, has escuchado rumores de lo difícil que es la programación desde que eras joven, como cuentos de viejas esposas destinados a asustar a los niños para que estudien ciencias sociales.

Por otro lado, el movimiento "Aprende a codificar" ha hecho un trabajo fantástico al romper barreras y mostrar a la gente que el código es en realidad bastante inofensivo. Herramientas como Codecademy y Treehouse llegan con los toques más suaves para asegurarte que tú también (¡no, cualquiera!) no solo puede

aprender a codificar, sino también convertirse en un desarrollador de pleno derecho.

De repente, el problema no es el miedo, es una sobreabundancia de esperanzas y altas expectativas.

Y, en su mayor parte, estas herramientas introductorias hacen un gran trabajo al guiarte como un niño en un paso de peatones más allá de las grandes variables aterradoras y las declaraciones condicionales y a través de las primeras fases de la sintaxis de programación. A medida que superas uno tras otro sus desafíos [gamificados](#), tu confianza aumenta. ¡Tal vez puedas hacer esto después de todo! ¿Qué tan difícil puede ser? *¡Básicamente ya eres un desarrollador!*



La luna de miel de la mano

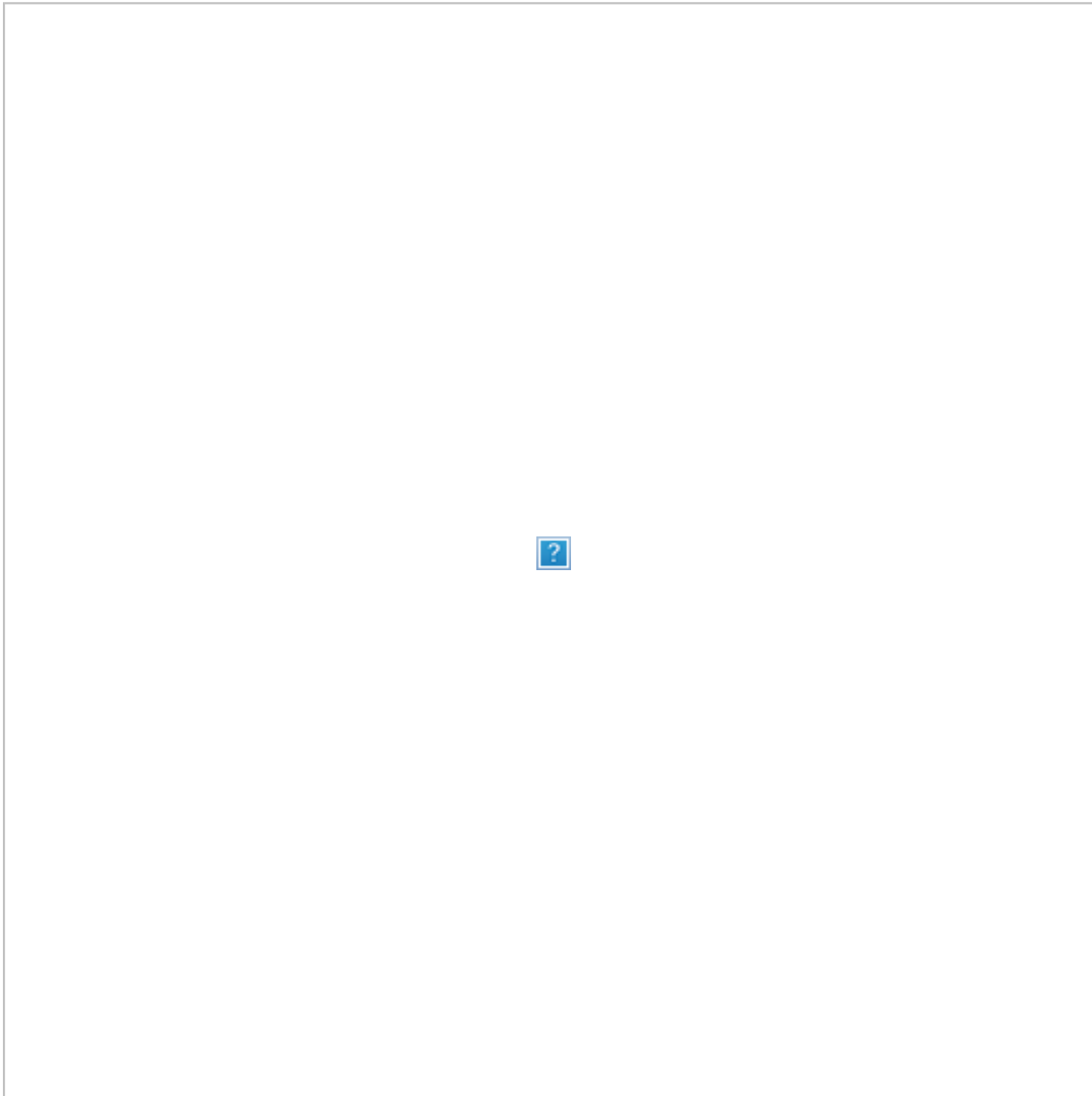
Aquí está el problema: estás en lo que me gusta llamar la fase de "Luna de Miel de Mano". Aunque puedas sentir que el final está a la vuelta de la esquina, solo estás a una fracción del camino. Esto es solo el principio...

Trazando El Camino Por Delante

Antes de sumergirnos en la Fase II, echemos un vistazo al panorama general.

En esta publicación, te guiaré a través de las cuatro fases del viaje típico hacia la codificación y lo que tendrás que hacer para sobrevivir a cada una de ellas. También verá cómo dos factores clave, la densidad de recursos y el alcance de los conocimientos requeridos, definen este viaje.

El viaje hacia la preparación para el trabajo se puede trazar en términos de cómo cambia su nivel de confianza a medida que aumenta su capacidad:



El viaje de aprender a programar -- [Haga clic para ampliar](#)

Esta es una relación relevante porque tu confianza está altamente correlacionada con tu felicidad y porque el punto en el que tu confianza y tus capacidades coinciden es el mejor proxy que tengo para el punto dulce cuando estás oficialmente "listo para el trabajo".

Examos los desafíos únicos de las 3 fases restantes en un momento, pero esto es lo que cada una de ellas implica esencialmente:

1. **The Hand Holding Honeymoon** es el juego lleno de alegría a través de recursos altamente pulidos que te enseñan cosas que parecen complicadas pero que son totalmente factibles con su apoyo intensivo. Aprenderás principalmente la sintaxis básica, pero te sentirás muy bien con tus logros.
2. **El acantilado de la confusión** es la dolorosa comprensión de que es mucho más difícil cuando la toma de la mano termina y se siente como si no pudieras hacer nada por tu cuenta todavía. Tus principales desafíos son la depuración constante y no saber cómo hacer las preguntas correctas mientras luchas hacia cualquier tipo de impulso.
3. **El desierto de la desesperación** es el largo y solitario viaje a través de un paisaje sin caminos donde cada nueva dirección parece correcta, pero con frecuencia estás dando vueltas en círculos y te mueres de hambre de los recursos para superarlo. Cuidado con los "Espejismos de la Manía", como las sirenas del desierto, que te desviarán.
4. **El Upswing de Awesome** es cuando finalmente has encontrado un camino a través del desierto y has reunido una comprensión de cómo crear aplicaciones. Pero tu código sigue siendo aislado y frágil como un castillo de naipes. Ganas confianza porque tus sitios parecen funcionar, has dominado algunos patrones útiles y tus amigos piensan que tus interfaces son geniales, pero te aterra mirar bajo el capó y, en última instancia, no sabes cómo llegar al código "listo para la producción". ¿Cómo se cierra la brecha a un trabajo real?

He entrevistado a cientos de aspirantes a desarrolladores en los últimos años y he escuchado ecos de la misma historia una y otra vez. Mi objetivo para esta publicación es que te acerques al viaje del alumno con los ojos abiertos y suficiente plan para que puedas evitar las trampas comunes de aquellos que te han precesto.

Vamos a volver a la Fase II...

Fase II: El acantilado de la confusión

Entonces, estás en la Fase I, la "Luna de Miel de la Mano", marcando insignias y completando desafíos de codificación mientras tu confianza y capacidades crecen. Esto no es tan malo... ¿de qué se trata todo el alboroto? Has llegado al "Cima de la Exuberancia Irracional"...

¡Ten cuidado! Estás a punto de superar un precipicio que ha roto a muchos aspirantes a estudiantes fuertes y los ha relegado al campo de "la codificación es demasiado difícil". El momento preciso en que ocurre este salto es la primera vez que te sientas en tu teclado, abres tu editor de texto e intentas construir un proyecto desde cero sin ninguno de los editores de lujo en el navegador, código en andamio o consejos útiles.

Mierda.

Podrías estirar esto un poco siguiendo tutoriales, pero nadie ha llegado a los cielos sin dejar el suelo y, en algún momento, vas a tener que crear magia a partir de un archivo de texto en blanco. Acabas de entrar en la segunda fase del aprendizaje, donde la confianza se derrumba en la tierra: el "Acantilado de la Confusión":



El acantilado de la confusión

Así que tú construyes. Luchas y te arañas para encontrar una solución apenas funcional, pero falta algo. Estás en una guerra con bichos que hacen que los [Starship Troopers](#) parezcan benignos. Parece que cada victoria se obtuvo solo con un golpe de suerte en Google y tu confianza en que alguna vez puedas descubrir que estas cosas se desploman.

Ataque de insectos de los soldados de la nave estelar

i i i B u u u u u u u u u u u u u u u u u u g ! ! !

Esta es una fase particularmente frustrante para ver como educador y para todos los participantes en nuestra industria. Puede que la programación no sea perfecta para todos, pero *queremos* que progrese porque a veces las historias más improbables se convierten en los éxitos más grandiosos.

Cuando la toma de la mano termina y los estudiantes son empujados por el acantilado y se les dice que vuelen, demasiadas personas potencialmente impresionantes están cayendo en espiral sobre las rocas de la frustración sin aprender a agitar sus alas.

La parte aterradora es que ni siquiera has llegado a las cosas carnosas todavía. Esta segunda fase, el acantilado de la confusión, todavía es muy temprana. Una vez que

finalmente hayas aplastado suficientes errores para acabar con la [octava plaga de Egipto](#) y hayas terminado un par de proyectos, marcando así el final de la Fase II, todavía estás empezando.

Para aquellos que están realmente listos para hacer una carrera de esto, sobrevivir al acantilado de la confusión es a menudo el punto en el que decides ir a por todas con tu nueva vida. Pero demasiados se quedan atrás. Y, desafortunadamente, estás a punto de entrar en el "Desierto de la desesperación".

Los dos factores clave en juego

Entonces, ¿qué marca realmente la diferencia entre una fase y la siguiente? ¿Por qué la Fase II (el Acantilado de la Confusión) fue tan horrible en comparación con la Fase I (la Luna de Miel de la Mano)? Comprender esto te ayudará a darte cuenta de que no es tu culpa en absoluto si tu viaje se parece a lo que acabamos de describir.

Básicamente, hay dos fuerzas clave en el trabajo en cada fase: la **densidad de recursos** y el **alcance del conocimiento**. Veamos qué son antes de explorar cómo definen la Fase III.

Factor 1: Densidad de recursos

Como dije anteriormente, cuando empiezas, parece que hay un millón de recursos por ahí tratando de tomarte de la mano y meterte en la codificación. ¡Eso es porque hay!

Busque ["Aprende a codificar"](#) y se le encontrará un muro de herramientas, textos, vídeos y tutoriales útiles y útiles. Y, francamente, ¡son geniales! Nunca antes había habido tantas formas de empezar a aprender a codificar.



Desafortunadamente, en fases posteriores la densidad de recursos disminuye rápidamente. Cualquiera que haya dado el salto de principiante a intermedio puede dar fe de que hay una GRAN diferencia entre la cantidad de recursos disponibles cuando comienzas frente a cuando buscas ayuda para construir cosas por tu cuenta sin sostenerte demasiado.

Este problema se exagera a medida que la cantidad de conocimiento aumenta rápidamente al entrar en la Fase III, y es una de las razones por las que llamamos a esa fase el "Desierto de la Desesperación". Una vez que supases esto y empieces a sentirte cómodo con lo que exactamente necesitas buscar, los recursos regresan y podrás trabajar con herramientas más técnicas como blogs de la industria y screencasts. Parte de esto es solo entender qué preguntas hacer.

Así es como se ve la densidad de recursos en cada fase (una mayor densidad de líneas indica más recursos):



Densidad de recursos en cada fase -- [Haga clic para ampliar](#)

Factor 2: Alcance del conocimiento

Ahora hablemos de un tema relacionado: el alcance del conocimiento. Esto representa la amplitud total de nuevos temas que necesitas aprender en cada fase.

Así es como se ve:



El alcance del conocimiento que se requiere en cada fase -- [Haga clic para ampliar](#)

Cuando empiezas a aprender, el conjunto de cosas que necesitas entender es estrecho. Todos, independientemente de los objetivos, el idioma o los antecedentes, necesitan averiguar qué es un bucle `for`, cómo construir lógica condicional y otras estructuras básicas de la sintaxis de programación. En última instancia, ni siquiera hay muchos de estos conceptos fundamentales, por lo que el alcance del conocimiento durante esa fase es muy estrecho.

Tan pronto como te alejas de lo básico, ves una rápida ampliación del alcance del conocimiento, ya que necesitas comenzar a recoger cosas que son más difíciles, como entender los errores y *cuándo* usar el código que sabes usar. Esto es diferente porque no hay una respuesta "correcta" a una pregunta clara... las cosas se ponen confusas.

Cuando avanzas a la tercera fase, el alcance del conocimiento se hincha más amplio. Ahora necesita entender qué herramientas usar, qué idiomas aprender, los fundamentos subyacentes de CS, cómo escribir código modular, orientación a objetos, buen estilo y cómo pedir ayuda (por nombrar solo algunos). Cada viaje a [Google](#) o [Hacker News](#) te lleva por otro conjunto de madrigueras de conejo y te abruma con más cosas que no sabes pero que sientes que deberías.

No sabes lo que no sabes.

Solo cuando finalmente has encontrado algo de tracción y has dejado el desierto, el alcance comienza a estrecharse de nuevo. En ese momento, has encontrado la tecnología que has elegido y su lugar en el ecosistema. Finalmente (más o menos) sabes lo que no sabes y puedes trazar un camino a través de él. Continuarás aumentando la concentración a medida que avanzas y hacia el comienzo de tu carrera.

Fase III: El desierto de la desesperación

Con una comprensión de estos factores, puedes ver que el Acantilado de la Confusión es realmente solo un punto de inflexión. El dolor causado por la combinación tóxica de un alcance de conocimiento en rápido aumento y una densidad de recursos que cae como resultado en lo que yo llamo el "desierto de la desesperación".

En esencia, este desierto es donde sabes que hay un final *en alguna parte*, pero no sabes cómo llegar allí:



El desierto de la desesperación

El desierto es largo y está lleno de peligros. Te encontrarás atraído por "Mirages of Mania" a lo largo del camino, docenas de recursos tentadores que parecen contener las soluciones que estás buscando, pero que te depositarán, una vez más, en un lugar donde la arena solitaria se extiende a cada línea del horizonte.

Tal vez te inscribas en un par de cursos [MOOC](#) de Coursera o Udacity o edX. O encuentres un tutorial que pretende llevarte hasta el final. Pensaste que habías aprendido las lecciones de la luna de miel de la mano, que no hay respuestas fáciles, pero la tentación de buscar la salvación es demasiado grande y caes en la promesa de que *esta* te llevará al final donde los demás no lo hicieron.

No puedes aprender estas cosas en una semana o un mes o una sola clase universitaria, no importa lo que alguien diga, ¡así que deja de caer en eso!

Hay MUCHO más que aprender de lo que probablemente esperabas. Incluso si eres capaz de hacer que algunas aplicaciones funcionen, es difícil no sentirse perdido en el mayor esquema de convertirse en un verdadero profesional. Es difícil medir tu progreso. ¿Cómo sabes lo que necesitas aprender o si incluso estás aprendiendo las cosas correctas?

Incluso si estás apuntando en la dirección correcta, es difícil medir tu progreso. Puede que te sientas totalmente perdido hasta el mismo momento en que finalmente puedas construir algo que se vea y actúe de la manera que esperabas. Pero, con suficiente perseverancia y una buena brújula, eventualmente lanzarás tus primeros proyectos "reales" y te darás cuenta de que finalmente estás empezando *a conseguirlo*.

Claro que ha sido difícil hasta ahora, pero tal vez esto de los desarrolladores web no sean tan malos después de todo... [¡Todo está llegando, Milhouse!](#)

¡Todo Se Acerca, Milhouse!

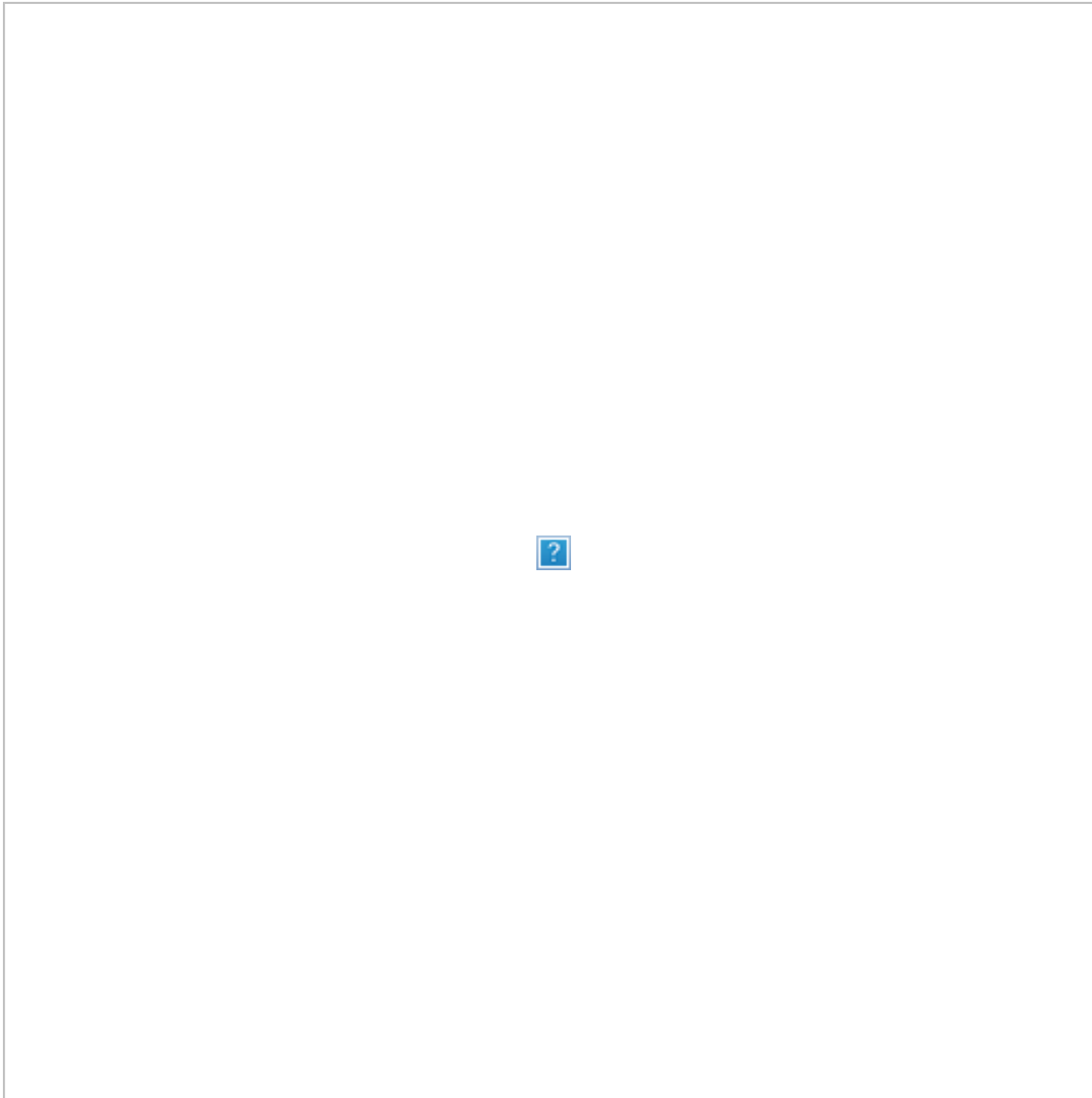


Fase IV: El respeto de lo impresionante

Has atravesado el desierto y tu confianza está creciendo. Tu [Google-fu](#) es excelente y finalmente puedes entender esas publicaciones detalladas de blog y

screencasts de la industria. Tal vez hayas profundizado en un lenguaje o marco en particular y tengas confianza en que puedes construir y lanzar una aplicación que funcione.

Este es el "Recurso de lo impresionante":



El respe de lo impresionante

Todo puede parecer bien por fuera, pero en el fondo sabes que aún no estás allí.

Puedes hacer que esa aplicación funcione, pero ¿qué está pasando debajo de la superficie? Tu código es cinta adhesiva y cuerda y, lo peor de todo, ni siquiera sabes qué partes son terribles y cuáles están bien. Tus destellos periódicos de brillantez se contrarrestan con errores de novato y, peor aún, con una sospecha progresiva de que todavía no tienes ni idea de lo que estás haciendo.

Esta es una fase bipolar. Sientes que la mitad de ti es un desarrollador a prueba de balas y la otra mitad es una fina carilla de efectividad que cubre a un novato de ojos salvajes que está demasiado metido. Cuanto más progresas, más crece una sensación de incertidumbre de que alguien te va a "exprender" como un fraude.

Sientes que ya deberías ser un desarrollador, pero la distancia entre el código que estás escribiendo y un entorno de trabajo "profesional" no podría parecer más lejos...

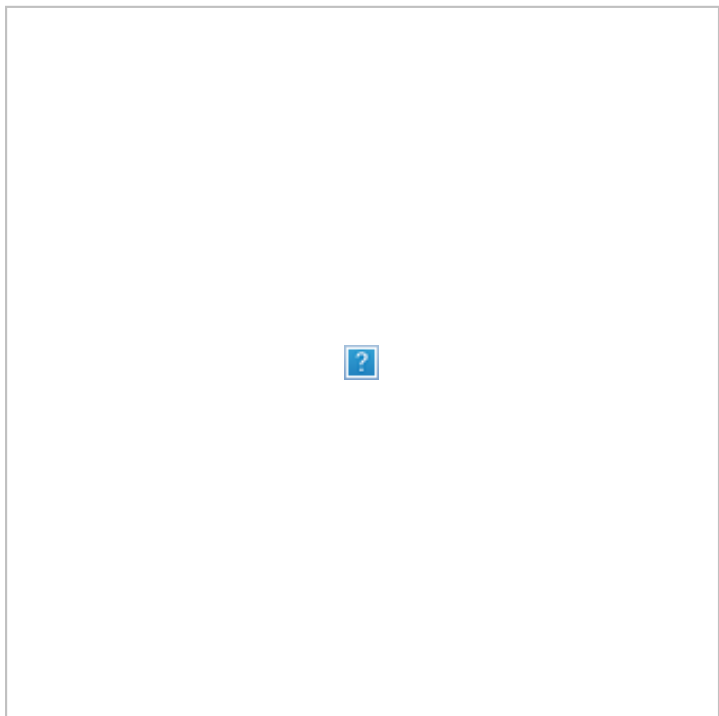
Eventualmente, sin embargo, lo lograrás. ¡Hay demasiado impulso para no hacerlo! El desierto de la desesperación está detrás de ti y el acantilado de la confusión es un recuerdo lejano. Finalmente, de *verdad*, estás en aunque. Estás aprendiendo más rápido y de forma más inteligente que nunca y, eventualmente, habrás absorbido suficientes mejores prácticas para que tu conocimiento del queso suizo se fusione en un conjunto de habilidades de grado de producción.

El Upswing de Awesome siempre lleva más tiempo de lo que esperas y se siente interminable porque estás *tan cerca*... pero lo conseguirás. Si eres lo suficientemente persistente en las formas correctas (el tema de una publicación futura seguro), convencerás a alguien de que te pague por seguir aprendiendo. El trabajo es tuyo.



Cómo se ve todo

Así que ahora has visto el camino por delante y las razones por las que puede ser difícil. Cuando combinas las cuatro fases que acabamos de cubrir con los factores que las definen, se ve algo así como el siguiente gráfico:



Una cosa es conocer el camino y otra es caminar por él. Vamos a empezar con el pie derecho.

Cómo superarlo vivo

El viaje parece intenso y, francamente, a menudo lo es. Es importante que entiendas lo que te estás deparando, especialmente si lo haces solo. *Pero no tienes que hacerlo.* Hay formas de cortocircuitar la mayoría de estos problemas. Aprender a codificar rara vez es tan fácil como la gente lo hace parecer, pero también rara vez es tan difícil como parece en las profundidades de tu desesperación.

En esta sección, presentaré las tácticas clave que puedes usar para mantenerte apuntando en la dirección correcta.



Su progresión a través de las fases -- [Haga clic para ampliar](#)

I: Sobrevivir a la luna de miel de la mano

La plétora de recursos disponibles en la luna de miel de la mano lo hacen muy divertido. Hacen un gran trabajo al facilitarte el tipo de pensamiento lógico que necesitarás cultivar en las próximas fases. Es un buen momento para empezar a aprender a codificar, así que trata de disfrutarlo y ten en cuenta estos dos consejos:

1. **Comience probando diferentes recursos** para encontrar cómo aprende mejor y qué tipo de proyectos son los más interesantes para usted. Tal vez sean los desafíos rápidos de [Khan Academy](#), los ejercicios en el navegador de [Codecademy](#) o la experiencia [de tutoría individual](#) de Thinkful. Sé de mente abierta al principio e ignora cualquier cosa sobre lo que *deberías* aprender... todo

el código es igual en esta fase.

2. **Luego elija un recurso** y quédese con él una vez que haya encontrado su ajuste. Trabaja hasta el final de su arco del curso introductorio, que debería darte todos los conocimientos básicos que necesitas para escribir guiones y aplicaciones básicas. Entonces prepárate para empezar a construir por tu cuenta.

II: Sobreviviendo al acantilado de la confusión

Casi todo el mundo experimentará el acantilado de la confusión porque la única manera de [convertirse en desarrollador](#) es, bueno, desarrollar. Puedes fingir que estás construyendo inscribiéndote en tutoriales (o tutoriales que se disfrazan de cursos "completos"), pero solo estás posponiendo lo inevitable. Los tutoriales son una buena manera de salvar las ofertas introductorias más de alto toque, pero tendrás que dejarte del chupete y enfrentarte al mundo real en algún momento.

Tres consejos para hacer la transición a la construcción por su cuenta:

1. **Trabaja con otra persona**, incluso con otro principiante. Te sorprenderá lo fácil que es depurar un error imposible al compartir dos pares de ojos.
2. **Lee el código de otras personas** para sentirte cómodo con buenos patrones. Intenta entender por qué el autor hizo lo que hizo. No tratarías de convertirte en novelista sin leer libros también, ¿verdad? Nos centraremos en esto en una próxima publicación, pero, por ahora, mantén los ojos abiertos para cualquier pequeño problema o proyecto para el que otras personas hayan escrito soluciones.
3. **Empieza poco a poco y construye constantemente.** Deberías tener grandes proyectos interesantes en mente para el futuro, pero necesitarás sentirte cómodo depurando y buscando recursos con desafíos del tamaño de un bocado. Realmente no hay sustituto para la experiencia.

III: Sobrevivir al desierto de la desesperación

Una vez que te sientas cómodo depurando, tu mayor problema se convierte en la

manguera de incendios del conocimiento requerido y una pérdida total de cómo aprenderlo todo... el Desierto de la Desesperación. En este caso, lo que realmente necesitas es un camino fuerte hacia adelante. Los Mirages of Mania representan todos los interesantes caminos secundarios y madrigueras de conejo y esquemas de habilidades rápidas que finalmente te hacen perder el tiempo.

Así que las claves para salir del desierto de la desesperación son:

Ten un objetivo fuerte para lo que quieres lograr porque de lo contrario terminarás persiguiendo tu cola aprendiendo todo tipo de cosas interesantes pero en última instancia improductivas. Si tienes tiempo de sobra, omite esto por todos los medios...

Encuentre un camino sólido que lo lleve directamente al objetivo que se ha establecido y verifique que realmente lo lleve allí. Aquí es donde necesitas profundizar más que los eslóganes de marketing y las caras sonrientes en los sitios web de cursos o en las cubiertas de libros para preguntar "¿me ayudará esto a lograr el objetivo que me he establecido o no?"

Concéntrate y evita las distracciones porque, si eres el tipo de persona que está interesada en aprender a codificar, también eres el tipo de persona que se interesa aprendiendo *todo tipo* de otras cosas increíbles. Cuando la codificación se vuelve difícil, necesitas ser capaz de seguir adelante en lugar de solo probar la siguiente cosa de aspecto genial.

Si eres capaz de identificar un camino y seguir con él, eventualmente avanzarás a la siguiente fase en lugar de pasar meses o años persiguiendo espejismos a través de las arenas cambiantes de este desierto.

IV: Sobreviviendo al respeo de Awesome

The Upswing of Awesome es una de las transiciones más complicadas. Puedes *desarrollar aplicaciones*, pero realmente quieres *convertirte en un desarrollador web*. Salir de esta fase y entrar en un trabajo requiere que hagas tres cosas:

Busque y siga las mejores prácticas para la programación. Necesitas entender la diferencia entre *una solución* y *la mejor solución*. Las mejores prácticas son una gran diferencia entre piratear por su cuenta y construir código de calidad de producción en un entorno de trabajo real.

Comprueba tus suposiciones porque probablemente hayas patinado con algunos agujeros en tu conocimiento que ni siquiera sabías que tenías. Necesitas diagnosticar y arreglar estos agujeros.

Aborda las habilidades poco atractivas que rara vez se abordan, pero que son muy importantes para la transición a un entorno profesional. Esto incluye cosas como pruebas, modelado de datos, arquitectura e implementación que son realmente fáciles de superar, pero que son totalmente fundamentales para un buen desarrollo.

La clave para lograr estas cosas y superar el Upswing de Awesome es **obtener comentarios**. Los estudiantes que han aprendido completamente por su cuenta pueden ser productivos, pero rara vez tienen el tipo de código legible, modular y mantenible que los hace atractivos en un entorno profesional. Necesitas trabajar con otros humanos que desafiarán tus suposiciones, harán preguntas de seguimiento penetrantes y te obligarán a arreglar las filtraciones en tu cubo de conocimiento.

Así que... ¿Se puede hacer?

Todo esto puede sonar abrumador, pero prometo que muchos otros han perseverado y sobrevivido a este viaje antes que tú. Al entender el camino por delante, ya estás en un buen lugar para asumirlo con un plan enfocado y acceso al tipo correcto de ayuda.

Obviamente, no hay espacio en esta publicación en particular para profundizar tanto en cada fase del viaje como nos gustaría o para proporcionar el tipo de consejos granulares que se merece. Dicho esto, este es un viaje con el que

estamos bastante familiarizados y del que nos apasiona mucho, por lo que queremos ayudar en todo lo que podamos.

Nuestro programa de inmersión en ingeniería está diseñado específicamente para unir todo este proceso, pero, si está interesado en seguirlo por su cuenta, también lo abordaremos de forma pública y en profundidad durante futuras publicaciones de blog.

Un agradecimiento especial a Peter DePaulo, Javier Noris, Michael Alexander, Andy Brown, Saul Costa, Phil Nachum y Quincy Larson por compartir sus experiencias y por ayudar a depurar estos pensamientos.

Aprende a codificar con Thinkful

Tome el camino probado hacia una carrera de altos ingresos con tutoría y apoyo profesional, formas flexibles de pago y aprendizaje basado en proyectos en el mundo real.

VER NUESTRO PLAN DE ESTUDIOS

Carrera en codificación

¿Qué es la codificación?
¿Qué hace un programador?
Salario de codificador
Habilidades de codificador
Conviértete en un

Aprende a codificar

Bootcamps de codificación
Certificados de codificación
Cursos de codificación
Escuelas de codificación
Programas de capacitación en

Guía de codificación para principiantes

Mejores Prácticas De
Codificación

Libros de codificación

Lenguajes de codificación
para aprender

¿Cuánto tiempo se necesita
para aprender a programar?

Boletines de codificación

Codificación vs Programación

Conseguir un trabajo de codificación

Prácticas de codificación

Preguntas de la entrevista
de codificación

Búsqueda de empleo de
codificación

Currículum de codificación

Compartir este artículo



Recomendados

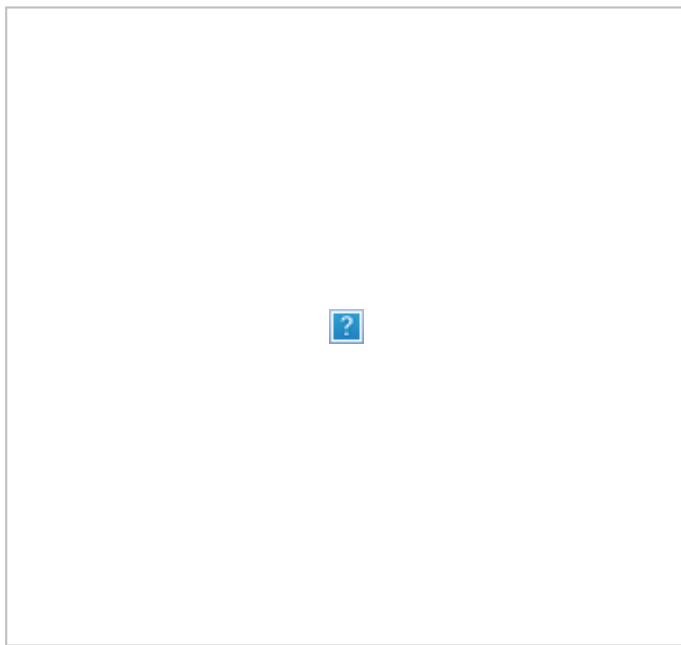
Encuentra más como esta historia



Trucos de estudio: Consejos
para principiantes en
codificación



¿Qué hace un programador?



Cómo convertirse en programador



Cursos

- Ingeniería de software
- Ciencia de datos
- Análisis de datos
- UX / Diseño UI
- Marketing digital
- Gestión técnica de proyectos

información

- Cómo trabajamos
- Resultados
- Contratar a nuestros graduados
- Catálogo del curso
- Seminarios por Internet
- Centro de aprendizaje
- Buscador de Bootcamp

Información reglamentaria

EMPRESA

Sobre nosotros

CONDICIONES DE USO

política de privacidad

carreras profesionales

prensa

Preguntas frecuentes

reseñas

SEGUIR

TWITTER

FACEBOOK

INSTAGRAM

Blog pensato

© 2023 Thinkful, Inc.