

Programar a ciegas

□ pulsá para escuchar la entrada

Existen personas con discapacidad que no saben que pueden dedicarse al desarrollo de aplicaciones y sitios webs. Hay quienes no saben que pueden contratar personas con discapacidad, y que estas pueden desempeñarse en este trabajo, a pesar de las dificultades. Programar a ciegas. Tarea nada fácil. Pero que sea difícil, no quiere decir que sea imposible. Debido a las repercusiones que ha tenido la entrada anterior, he decidido dejar por aquí, e ir agregando posteriormente, es decir, actualizando la entrada, con herramientas y recursos que me han ayudado a aprender, y a desempeñarme como desarrolladora, durante los últimos 10 años. Si bien, no de forma totalmente autónoma y accesible en muchos casos, en otros sí. Creo que hoy en día, y con las herramientas actuales, puedo decir que es una carrera que podemos cursar, y de la que podemos trabajar, ya sea freelance o para alguna empresa, sin mayores dificultades, claro está, que las que ya hemos mencionado en la charla que dimos en la FemITCONF2021. Aunque ya en la mencionada entrada, he dejado algunos sitios webs de interés, en esta oportunidad, voy a optar por algo más concreto. Aplicaciones y recursos webs, listos para programar, de forma casi, o totalmente accesible. Espero, algo de esto les sirva.

Lo primero que quiero destacar, es que yo, particularmente, no me he especializado en desarrollar apps móviles. Sí he trabajado con apps de escritorio y consola, sitios webs, y bases de datos. Entonces, son de estos temas de los que aquí voy a hablarles. Si quieren información sobre otras cuestiones, deberán dirigirse a páginas de otros desarrolladores ciegos, que sé que las hay. Habiendo dicho esto, ahora sí, comencemos.

Editores accesibles

Un editor, es básicamente un software que te permite escribir código de varios lenguajes de programación. No es muy distinto, de hecho también se puede utilizar, al clásico bloc de notas de windows. Pero a diferencia de este, en los editores de código, nos muestra el número de línea en el que nos encontramos, y además, nos detecta el lenguaje con el que estamos trabajando. Tienen función de autocompletado de código, y la posibilidad de ejecutar el código desde el propio programa, añadiendo complementos/extensiones, e instalando los intérpretes de los lenguajes que sean necesarios. Aquí, dejo los más utilizados. Primero, pueden [descargar el Notepad++](#). Personalmente, es el que más he utilizado, ya desde hace años, incluso mucho antes de que sea totalmente accesible con JAWS y NVDA, los lectores de pantalla por excelencia para windows. También, desde el siguiente enlace pueden [descargar el Visual Studio Code](#). Últimamente se ha convertido en el más utilizado no solo por personas con discapacidad visual, si no en general. No tengo mucha experiencia en este, porque empecé a usarlo hace poco, pero sé que tiene funciones de compatibilidad/accesibilidad con lectores de pantalla. Ambas opciones tienen atajos de teclado con los que su funcionalidad, es incluso más amena y versátil.

Entornos de desarrollo integrado

IDEs, por sus siglas en inglés, son programas mucho más complejos que los mencionados anteriormente. Mientras que los otros permiten escribir código en formato texto, estos traen soluciones integradas con las cuales se pueden además, diseñar aplicaciones gráficas. Son muy utilizados en apps para windows y móviles, entre otros sistemas operativos, e incluso sitios webs. Es cierto que el código en sí, es en formato texto. Pero diseñar ese sitio, redimensionar esa ventana, correr ese botón, o aumentarle el tamaño a alguna sección en particular,

es mucho más fácil, al menos para quienes ven, hacerlo visualmente, que teniendo que tocar código. En lo personal, aunque las funcionalidades sí son accesibles, es difícil hacerlo si nunca tuvimos una representación del contenido de la pantalla en nuestra cabeza. No digo que sea imposible, pero es algo que particularmente a mí, se me dificultó mucho. Los más conocidos, o al menos que yo tuve la posibilidad de probar, y que obvio, son accesibles con lectores de pantalla, son: [Microsoft Visual Studio](#). Fue diseñado originalmente para Windows, pero hoy en día es multiplataforma. Basado en .net, tiene versiones gratuitas, y otras más profesionales. Además de las mencionadas anteriormente, otras de sus características, son una base de datos SQL, y un servidor web. De estas utilidades, hablaremos más adelante. Otro de ellos,, es [Eclipse](#). Basado en el lenguaje de programación Java, a diferencia del anterior, aunque no tiene todo integrado, es posible descargarse versiones distintas dependiendo del lenguaje, y de lo que se desee hacer. Por ejemplo, para crear un servidor web, puede integrarse con software como apache. Lo dicho, esto lo ampliaremos en breve.

Aclaración. Que estén basados en .net y java, quiere decir que necesitan de estos motores de desarrollo. Son en sí, herramientas con pequeñas aplicaciones necesarias, para que tu código funcione. Si no instalamos .net o la máquina virtual de java, nuestra aplicación, no va a ir ni para atrás, ni para adelante.

Gestores de bases de datos

No es mi objetivo dar una explicación pormenorizada de como funciona un programa en su totalidad, si no, como ya mencioné, brindar herramientas accesibles. Pero en sí, creo que sobre esto, sí hace falta fundamentar un poquito más. Una base de datos, es la forma en la que, tus datos se almacenan, para luego poder ser utilizados por aplicaciones o sitios webs. Por ejemplo, podríamos ser dueños de una compañía de teléfono. Y

esta, tendría, a simples rasgos, los siguientes datos. Tu nombre, como titular de la línea, tu identificación (DNI en Argentina) tu número de línea asociada, es decir tu número de teléfono, tu ciudad/país de residencia, entre otros. Después, en otros datos, podemos tener los distintos planes que ofrecemos. si es a abono o prepago, qué clase de recargas se pueden hacer en caso de prepago, que beneficios proporcionamos dependiendo de tu plan si es a abono, etc. Entonces, tendríamos que, mediante un código de identificación, relacionar la primer información, con la segunda, para saber qué tipo de plan, tiene tu celular. Es ahí, donde entran las bases de datos. En aplicaciones pequeñas, estos datos se guardan en archivos específicos, muchas veces en formato texto, o, en formatos más encriptados, pero en fin, archivos que no ocupan mucho espacio. Pero para aplicaciones más grandes o empresas, es necesario utilizar un gestor de bases de datos. Entonces, veamos cuales de estos son accesibles, de acuerdo a mi experiencia, como siempre digo.

Algunas de las herramientas mencionadas más arriba, tienen integración con bases de datos, que pueden descargarse aparte, como en el caso de Visual Studio Code o Eclipse, o que ya vienen dentro de la propia aplicación, como es el caso de Microsoft Visual Studio. Pero si nos vamos a dedicar a gestionar bases de datos únicamente, necesitamos herramientas específicas, que realicen esta tarea. Uno de los más completos, es el [SQL Server](#). Se integra con las aplicaciones de Microsoft, obviamente, y utiliza este lenguaje.

Hay que tener en cuenta también, que SQL es en sí un lenguaje en su mayoría estandarizado, pero que cada vez que se lanza una nueva versión de este, ya sean desarrolladores o compañías, como en el caso de Microsoft, le agregan sus propias características. En sí, en su mayoría es lo mismo. Consultar datos, almacenarlos, actualizarlos, crearlos o combinarlos, para obtener los resultados requeridos. Habiendo dicho esto, continuemos.

Existen también, bases de datos basadas en la nube. Sí, ya sé. Hay una premisa que dice que tu nube, es siempre la computadora de alguien más. Pero si una determinada empresa no tiene que comprar y mantener esas computadoras, aunque tenga que alquilar un servicio de otra, puede almacenar muchísima más cantidad de datos, lo que a la larga, es conveniente. Entre ellas, tenemos [Salesforce Object Query Language \(SOQL\)](#). Como su nombre lo indica, le pertenece a la compañía Salesforce. En este caso, no se puede decir que sea totalmente accesible. De hecho, la mayoría de su sitio no lo es. Aunque, he de destacar que están trabajando en ello de a poco. Podemos realizar consultas de datos, pero requerimos extraerlos en archivos CSV (archivos de valores separados por coma en español), para leerlos correctamente. Otra de ellas, con la cual yo estoy trabajando actualmente, y que sí es muuuy accesible, es [BigQuery de Google](#). Tiene opciones de accesibilidad, autocompletado, se pueden filtrar los resultados, entre muchas otras funciones. Por supuesto, la nube de google tiene muchísimos más recursos, funcionalidades y herramientas que estas. Pero hasta que no las conozca un poco mejor, dejémoslo acá.

Pero claro, eso no es todo, me dirán los expertos. Hay otros tipos de lenguajes y gestores de bases de datos, como [MySQL](#), [MariaDb](#), [PostgreSQL](#), entre otros. Bueno, lo cierto es que, aunque son lenguajes similares a los mencionados, acá nos encontramos con un inconveniente. La mayoría de sus gestores, no son accesibles. ¿Y bueno, entonces, les debe surgir la duda de como hice yo para trabajar con ellos? Como decimos en el mundo de la discapacidad visual, cuando la accesibilidad no existe, hay que rebuscársela, darse maña, para que de una forma u otra, salga. Y es aquí, donde saltamos a la siguiente sección. Pero antes, les dejo para que le vayan dando un vistazo, este [Curso de SQL de IKeydata](#). Me sirvió muchísimo al principio, para ir buscando y conociendo las distintas posibilidades que existen, a la hora de manipular datos.

Servidores webs

Sin entrar al desarrollo web directamente, que eso va a ser un tema aparte, es preciso comentar que, podemos crearnos un servidor web, para manejar una base de datos local en alguno de los lenguajes mencionados, de forma totalmente accesible. Instalando [WAMP \(Windows, Apache, MySQL y PHP\)](#). O [XAMPP \(X de multiplataforma, Apache, MariaDB, PHP y Perl\)](#). Podemos utilizar sus gestores, como [PHPMyAdmin](#), o mi favorito, por su simpleza, accesibilidad y por ser tan intuitivo, [Adminer](#), como si de moverse por un sitio web se tratara. De hecho, es básicamente eso. Una página web en PHP, con la que podemos escribir consultas, eliminar, modificar y crear datos, leerlos en formato de tablas, Entre otras muchísimas cosas, sin usar esos gestores que traen por defecto, y que son inaccesibles.

¿Pero, qué es un servidor web? Microsoft Visual Studio trae el suyo propio, y se puede ejecutar código de ASP.Net, el lenguaje de programación para webs de esta compañía. Un servidor web, es un conjunto de herramientas que nos permiten crear y ejecutar una página web, como si estuviéramos ya en internet. De los de software libre, Apache es el más conocido. Y, si bien, las herramientas mencionadas anteriormente (Apache, PHP, MySQL), pueden instalarse por separado, la posibilidad que ofrecen estos servidores prediseñados, es facilitarnos el trabajo, a la hora de desarrollar nuestro sitio web. Instalamos todo, ejecutamos, y listo. Es más, como ya mencioné, el gestor de bases de datos MySQL o MariaDB, ya se ejecuta al iniciar los servicios, junto con el de apache, que es lo que hace que tu código web, esté disponible como cualquier página de internet, pero local, claro está. Para que esa página esté disponible en internet, bueno, eso ya es otro tema, que no vamos a tocar aquí.

Creación de sitios webs

Es uno de los temas que más me apasionan después de bases de datos, al menos a mí. Y de los que más traté de aprender. Como venimos hablando, el código de páginas webs, se puede escribir en cualquier editor de texto, o IDE. El problema es, transformar eso en algo que podamos ver en funcionamiento en tiempo real. Si está escrito en HTML/CSS/Java Script, no vamos a tener problema alguno. Solo bastará con abrir el archivo html con nuestro navegador, ya sea Chrome, Firefox, etc, y el resultado estará disponible. Para familiarizarse un poco con el código de HTML, les dejo en inglés, la página [All The Tags](#). Una lista de todas las etiquetas HTML, incluso HTML5, para que al hacer click en cada una, nos aparezca su descripción. Ahora, si queremos algo más avanzado, podemos optar por [El gran libro de HTML5, CSS3 y Java Script](#). Nota: este libro ahora es pago. Cuando yo lo leí era gratuito, y bastante ameno. No garantizo su accesibilidad en su versión actual, debido a que no lo compré. Solo enlace a la nueva página del mismo.

Por otro lado, si a ese código HTML y CSS queremos agregarle otro tipo de código diferente a Java Script, y además bases de datos, ahí sí vamos a necesitar las herramientas anteriores, para crear nuestro servidor web. Veamos qué opciones tenemos, para hacer esto. Podemos encontrar algo de lo que ya hemos mencionado. ASP.NET, ya sea C# o Visual Basic de Microsoft. PHP, mi favorito, el más utilizado en toda la web ya sea con WordPress, o en páginas tan populares como facebook. Python, un lenguaje similar a Java o .net, en el sentido de que además de sitios webs, se pueden hacer apps de consola y de escritorio, y cuya sintaxis es muy sencilla de aprender, entre otros. En cuanto a Java Script, lo que tiene de particular, es que a diferencia de los anteriores, se codifica del lado del cliente, es decir, el código se puede ver en el navegador del usuario. En cambio, en los casos anteriores, si el usuario quiere ver el código fuente de una página web, solo va a ver

el HTML y el CSS. Quedando la programación del sitio, en manos solo del desarrollador.

Ya para finalizar este tema, comentar que, sin duda, a mí lo que más me cuesta aprender, incluso hasta el día de hoy, es la parte de diseño. Es decir, los estilos en CSS, en sus distintas versiones. Y es que, aunque tengamos noción de como escribirlo, y qué código poner en cada momento, o qué color usar, es bueno contar con una persona que nos garantice que lo que estamos armando, quede realmente como queremos que quede. Entonces, en mi caso particular me hizo sentir tanta inseguridad, que terminé dejando lo referente al diseño web, un poco de lado. Insisto, al igual que con la parte gráfica de una app de escritorio. Es difícil, sí. Nos encontramos con barreras al realizarlo, también. Pero esto no quiere decir, que sea imposible, y que no haya personas que sepan hacerlo.

Aplicaciones de consola

Dije que aquí solo iba a brindar herramientas accesibles, y claro que estoy cumpliendo con el objetivo. Pero además, me parece necesario añadir algunos consejos a la hora de iniciarse en la programación. Y es que, hay tanta información, que de seguro no debe ser fácil tomar una decisión acertada. Como venimos hablando, hay varios tipos de aplicaciones. Las gráficas, las webs, y las de consola. Ya hablamos de las primeras 2. Pero creo firmemente, y las carreras en todo tipo de instituciones y cursos online están de acuerdo conmigo, que las aplicaciones de consola, son las primeras con las que debemos empezar. Para una app de consola, solo necesitamos el editor de texto, y un compilador o intérprete, dependiendo del lenguaje. No tenemos que preocuparnos por el diseño, si no por la lógica. El algoritmo, el problema y su solución. Es lo primordial a la hora de iniciarse en cualquier lenguaje. Y, excepto los webs y los basados en bases de datos, todos tienen su equivalente de consola. Desde Java, pasando por C#, e incluso Python. Es más, en mi caso en particular, si bien no

he incursionado tanto en otros sistemas operativos además de windows, sí he logrado programar hace un tiempito en la consola de Linux con Python. Para dejar un poco más en claro el concepto de consola, comentar que, es una pequeña ventanita dentro del sistema operativo, en la que se pueden ejecutar tanto comandos propios del sistema, como código de programación, iniciando el intérprete o compilador de algún lenguaje. Se puede navegar con las flechas, o con los atajos específicos, en el caso de Linux.

Lenguajes interpretados o lenguajes compilados

También, veo necesario hacer una pequeña mención de esto, para que se entienda a qué hago referencia. Un lenguaje interpretado, es uno en el que, el código se escribe, se guarda con la extensión propia del mismo, .py para Python por ejemplo, y teniendo instalado python en la computadora, haciendo doble click o presionando enter sobre ese archivo, este ya se ejecuta en la consola. En los lenguajes compilados, el código pasa por un proceso adicional, en el que el compilador, transforma ese código en una aplicación .exe, en el caso de windows, que puede ejecutarse ahora sí, haciendo doble click. Esto es beneficioso en el sentido de que tu app puede correr en cualquier computadora sin instalar otro software aparte. Pero los lenguajes interpretados, te permiten probar el código en el intérprete cuantas veces sea necesario, antes de sacar la versión definitiva de esa aplicación, es decir, antes de hacer el proceso adicional, de transformarlo en un programa ejecutable.

Otros recursos

Antes de finalizar, les dejo unos últimos sitios webs que de seguro les resultarán de interés. Primero, la entrada anterior de la que vengo hablando. [Desarrollo, usabilidad y](#)

[accesibilidad con lectores de pantalla](#). Además, 2 de los sitios que compartí allí, pero, como pretendo que sea este el recursero informático por excelencia del blog, los comparto aquí de nuevo. El primero, es la [Fundación Itgrarte](#). Ofrece mediante becas, capacitaciones en programación, con preferencia para personas con discapacidad. Y el segundo, [Tutoriales Programación Ya](#). Para mí, el mejor sitio online, y casi totalmente accesible, para aprender programación de forma totalmente autodidacta.

Después tenemos [Programando accesible](#). Es una comunidad de personas con y sin discapacidad, que trabajan por el desarrollo de aplicaciones de forma accesible. Se hacen consultas, se comparten recursos, y se generan lindos debates y conversaciones, sobre la programación, con perspectiva de accesibilidad.

Segundo, un sitio de un español, que es desde hace años un gran referente entre los programadores ciegos. Además, es el sitio en el que he decidido basar el título de esta entrada. [Programar a ciegas](#).

Y por último, para quienes sean ciegos y quieran optar por el desarrollo web, les recomiendo usar [Bootstrap](#). Es un framework, es decir una especie de código integrado, en este caso de CSS, con estándares prediseñados para los sitios webs. Eso hace que podamos añadir diseño a nuestro sitio web ya codificado, y con la seguridad de que será compatible con varios tipos de dispositivos, incluyendo móviles.

Conclusiones

En primer lugar, comentar que, decidí escribir esta entrada, gracias a una persona con discapacidad que me escribió mediante la página de contacto del blog, explicándome que quería aprender a programar, y que no sabía por donde empezar. Al igual que esta persona, muchos no saben que teniendo una discapacidad, podemos dedicarnos al desarrollo, como dije al

principio. Entonces, se me ocurrió que podía armar una especie de curso para programadores ciegos. Me especialicé en la discapacidad visual total, es decir en la ceguera, porque es de lo que conozco, en lo que tengo más experiencia, por razones obvias, claro está. Aún así, no creo que esto sea todo, ni mucho menos. Invito a otras personas con discapacidad, a aportar, si así lo desean, sus propios sitios webs, herramientas o recursos para aprender programación, de forma accesible, los cuales, me comprometo añadir aquí.

En segundo lugar, algunas aclaraciones sobre el diseño de apps y sitios webs. Dije que es difícil, que nos encontramos con muchas barreras, y que en particular yo, no sé hacerlo en su totalidad. Pero también dije que hay quienes sí. Y, si no, aún así, hay muchas otras cosas más que sí podemos hacer. Estoy totalmente convencida de que el trabajo en equipo, con una repartición de tareas de forma equitativa, es la verdadera forma del éxito, en todo proyecto de programación. Entonces, si una persona ciega, no puede hacer el diseño de tu página, podés darle como tarea encargarse de la programación, o de la base de datos, y no va a tener inconveniente alguno. Es más, ya yendo a la parte de testeo, incluso puede encargarse de verificar que tu página, sea accesible.

En tercer lugar, sobre las herramientas para desarrollar. Como dije, de seguro no las conozco todas. Pero además, puede que haya muchas que tal vez sean accesibles, y no lo sepamos. Por ejemplo, yo no sabía que la consola de google lo era, hasta que no empecé a utilizarla. Lo importante en estos casos, es probar, probar y probar. Sí, así es. Si no sabemos si podemos usar una app o sitio web, naveguémoslo con los lectores de pantalla. JAWS o NVDA para Windows, u Orca para Linux, entre otros. Solo cuando probamos algo por nuestra cuenta, a menos que tengamos referencias previas de su accesibilidad, sabremos si vamos a poder utilizarla de forma accesible o a medias, o si directamente, es inutilizable. Tanto quienes tienen una discapacidad como quienes no, deben perder ese miedo a lo

desconocido. Solo así, podemos también aportar nuestro granito de arena, para contribuir al maravilloso mundo de la programación, de la usabilidad y la accesibilidad.

En cuarto lugar, sobre el inicio en la programación. El desarrollar apps de consola, es lo más recomendado, y lo más utilizado, al menos según yo sé. Pero si alguien desea iniciarse con el desarrollo web, yo no veo inconveniente alguno. Lo único, es que además de memorizar códigos, y conceptos específicos del lenguaje, ya sean PHP o Java Script, o cualquier otro, se debe ir recordando la estructura de las páginas HTML, lo que conlleva a mi entender, un tiempo adicional, que en un principio, puede ser innecesario. Pero mientras el instructor explique bien los conceptos, o estén claros en algún curso, y por supuesto, mientras este sea accesible, la verdad no me parece mal que empiecen por ahí. Además, surge siempre la duda de si, programación estructurada, u orientada a objetos. Tampoco veo inconveniente en ello. Mientras que la segunda va ganando terreno a medida que los lenguajes integran ese paradigma, la primera aún no ha quedado en desuso, y debido a que la tecnología en sí avanza más rápido que las carreras y cursos, en la mayoría de los casos, aún se sigue enseñando y utilizando en muchas instituciones y cursos. Lo que sí, ya sea que empiecen con el lenguaje y el paradigma que mejor les quede, o que más les guste, yo doy este último consejo. Si empiezan con algo, no lo dejen. Concéntrense en eso, y lleguen hasta el final. Una vez aprendido por completo un lenguaje, y en sí, todos sus conceptos, ya va a ser mucho más fácil aprender cualquier otro.

Por último, he de destacar que yo no soy docente, ni mucho menos. Entonces, puede que no sepa explicar bien algunos conceptos, puede que estos queden incompletos, o resulten poco claros. Pido disculpas por eso. Traté de hacer el artículo lo más ameno posible, centrándome en quienes aún no conocen absolutamente nada de programación, e intentar explicarles

algunas pautas de como esto es posible, aunque al principio, no parezca nada fácil. Espero haber sido clara. Y espero les haya gustado la entrada. Cualquier duda, consulta, sugerencia o idea, no duden escribirme a través de la página de contacto, en los comentarios, o a través de mis redes sociales.

Mis más sinceros agradecimientos

A todos aquellos programadores ciegos que marcaron mi camino. Allá por el 2008, cuando a duras penas conociendo muy por encima JAWS, ellos fueron mi luz, mi guía, quienes me enseñaron qué camino tenía que seguir, ya sea de este, como del otro lado del charco. Sí. Muchas gracias a todos. Nombrar a algunos, haría que tal vez me olvide de otros. Es por eso, que prefiero agradecerles en general. Contarles que desde entonces, mucha agua ha pasado bajo el puente. Dejé la carrera, empecé a trabajar, cambié de género, de cuentas de mail... Y en fin, ya llevo 10 años de dedicarme a esta profesión, la cual, más allá de algunos altibajos, me ha traído grandes satisfacciones. Gracias, muchísimas gracias, por ese abrazo virtual, que en momentos de desesperación y frustración por no poder compilar un código, entender un concepto, encontrar un programa accesible, me dieron a la distancia. Por muchísima más gente que contrate personas con discapacidad, y por muchísimas más personas con discapacidad, que se animen a programar. Para hacer de este mundo, uno mucho más inclusivo y accesible. Gracias, nuevamente, muchísimas gracias a todos ellos.