


Este es CS50

Introducción a la informática

OpenCourseWare

Donar  (<https://cs50.harvard.edu/donate>)

David J. Malan (<https://cs.harvard.edu/malan/>)
malan@harvard.edu

 (<https://www.facebook.com/dmalan>)  (<https://github.com/dmalan>)  (<https://www.instagram.com/davidjmalan/>)  (<https://www.linkedin.com/in/malan/>)  (<https://www.reddit.com/user/davidjmalan>)  (<https://www.threads.net/@davidjmalan>)  (<https://twitter.com/davidjmalan>)

Proyecto final

El clímax de este curso es su proyecto final. El proyecto final es su oportunidad de tomar su nueva experiencia con la programación para dar una vuelta y desarrollar su propia pieza de software. Mientras su proyecto se base en las lecciones de este curso, la naturaleza de su proyecto depende enteramente de usted. Puede implementar su proyecto en cualquier idioma (s). Le invitamos a utilizar infraestructura distinta del CS50 Codespace. Todo lo que pedimos es que construyas algo de tu interés, que resuelvas un problema real, que impactes a tu comunidad, o que cambies el mundo. Escántate de crear algo que sobrevive a este curso.

Dado que el desarrollo de software rara vez es un esfuerzo de una sola persona, se le permite la oportunidad de colaborar con uno o dos compañeros de clase para este proyecto final. Hulo sea que se espera que cada estudiante de cualquier grupo de este tipo contribuya igualmente al diseño e implementación de ese proyecto de grupo. Además, se espera que el alcance de un proyecto de grupo de dos o tres personas sea, respectivamente, dos veces o tres veces el de un proyecto típico de una sola persona. Un proyecto unipersonal, dígame la cabeza, debe implicar más tiempo y esfuerzo de lo que requiere cada uno de los conjuntos de problemas del curso.

Tenga en cuenta que el personal de CS50 audita las presentaciones a CS50x incluyendo este proyecto final. Los estudiantes que se encuentren violando la política de Honestidad Académica serán retirados del curso y considerados inelegidos para un certificado. Los estudiantes que ya hayan completado CS50x, si se encuentra en violación, tendrán su Certificado CS50 (y certificado edX, si procede) revocado.

Ideas

- una aplicación basada en la web usando JavaScript, Python y SQL
- una aplicación iOS usando Swift
- un juego con Lua con L-VE
- una aplicación para Android usando Java
- una extensión Chrome utilizando JavaScript
- un programa de línea de comandos usando C
- una aplicación basada en hardware para la que programa algún dispositivo
- ...

Empezar

Crear un proyecto entero puede parecer desatado. Aquí están algunas preguntas que usted debe pensar a medida que empieza:

- ¿Qué hará su software? ¿Qué características tendrá? ¿Cómo se ejecutará?
- ¿Qué nuevas habilidades necesitarás adquirir? ¿Qué temas necesitarás para investigar?
- Si trabajas con uno o dos compañeros de clase, ¿quién hará qué?
- En el mundo del software, la mayoría de todo tarda más en implementarse de lo que esperas. Y así no es raro lograr menos en un tiempo fijo de lo que esperas. ¿Cuál podrías considerar que es un buen resultado para tu proyecto? ¿Un mejor resultado? ¿El mejor resultado?

Considere hacer hitos de meta para mantenerte en el buen camino.

Si se utiliza el CS50 Codespace, cree un directorio llamado `project` para almacenar su código fuente de proyecto y otros archivos. Usted es bienvenido a desarrollar su proyecto fuera del CS50 Codespace.

Para su proyecto final (y su proyecto final solamente) es razonable utilizar software basado en IA que no sea el propio CS50 (por ejemplo, ChatGPT, GitHub Copilot, Bing Chat, et al.), pero la esencia del trabajo debe seguir siendo la suya. Has aprendido lo

suficiente como para usar herramientas como ayudantes. Tratar herramientas como amplificar, no suplantar, su productividad. Pero usted todavía debe citar cualquier uso de tales herramientas en los comentarios de su código.

Cómo Enviar

Debes completar los tres pasos.

Paso 1 de 3

Crea un vídeo corto (no más de 3 minutos de duración) en el que presentes tu proyecto al mundo. Su vídeo **debe** comenzar con una sección de apertura que muestra:

- su título de proyecto;
- tu nombre;
- sus nombres de usuario de GitHub y edX;
- su ciudad y su país;
- y, la fecha en que has grabado este video.

Luego debería continuar con la demostración de su proyecto en acción, como con diapositivas, capturas de pantalla, voz en off y/o acción en vivo. Consulte [howtogeek.com/205742/cómo-a-record-your-windows-mac-linux-android-o-i-i-panscreen](https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/) (<https://www.howtogeek.com/205742/how-to-record-your-windows-mac-linux-android-or-ios-screen/>) para consejos sobre cómo hacer un "screencast", aunque seas bienvenido a usar una cámara real. Sube tu video a YouTube (o, si está bloqueado en tu país, un sitio similar) y toma nota de su URL; está bien para marcarlo como "unlisted", pero no la marcar como "privado".

Envíe [este formulario \(https://forms.cs50.io/d5009db5-ee7d-43f1-8214-87cebc1a554f\)](https://forms.cs50.io/d5009db5-ee7d-43f1-8214-87cebc1a554f).

Paso 2 de 3

Crear un `README.md` archivo de texto (nombre exactamente eso) en su `project` carpeta que explica su proyecto. Este archivo debe incluir su Título de Proyecto, la URL de su vídeo (creado en el paso 1 anterior) y una descripción de su proyecto. Puede utilizar lo siguiente como plantilla.

```
# YOUR PROJECT TITLE
#### Video Demo: <URL HERE>
#### Description:
TODO
```

Si no está familiarizado con la sintaxis de Markdown, es posible que encuentre útil la [escritura básica y la identificación \(https://docs.github.com/en/free-pro-team@latest/github/writing-on-github/basic-writing-and-formatting-syntax\)](https://docs.github.com/en/free-pro-team@latest/github/writing-on-github/basic-writing-and-formatting-syntax) de GitHub. También puedes previsualizar tu `.md` Archivo haciendo clic en el icono de la vista previa, como se explica aquí: [Previsualización de marcado en vscode \(https://code.visualstudio.com/docs/languages/markdown#_markdown-preview\)](https://code.visualstudio.com/docs/languages/markdown#_markdown-preview). Proyecto de software estándar `README`s a menudo puede toparse con los miles o decenas de miles de palabras de longitud; la suya no tiene por qué ser tan larga, sino que por lo menos debe ser varios cientos de palabras que describen las cosas en detalle.

Tu `README.md` El archivo debe ser mínimamente varios párrafos en longitud, y debe explicar cuál es su proyecto, qué contiene y hace cada uno de los archivos que escribió para el proyecto, y si debatió ciertas opciones de diseño, explicando por qué los hizo. Asegúrese de asignar suficiente tiempo y energía a escribir un `README.md` que documenta tu proyecto a fondo. Estad orgullosos de ello. A `README.md` en el vecindario de 750 palabras es probable que sea suficiente para describir su proyecto y todos los aspectos de su funcionalidad. Si no puede alcanzar ese umbral, eso probablemente significa que su proyecto no es suficientemente complejo.

Ejecutar el `submit50` comando a continuación desde el interior de su `project` directorio (o de cualquier directorio contiene `README.md` archivo y su código de proyecto, que también debe ser enviado), iniciando sesión con su nombre de usuario y contraseña de GitHub cuando se les solicite. Para la seguridad, verás asteriscos en lugar de los caracteres reales en tu contraseña.

```
submit50 cs50/problems/2024/x/project
```

▼ Sumisión de problemas?

Si te encuentras con problemas porque tu proyecto es demasiado grande, intenta ZIP todo el contenido de ese directorio (excepto `README.md`) y luego presentarlo en su lugar. Si sigue siendo demasiado grande, intente eliminar ciertos archivos [de \(https://github.com/me50/USERNAME\)](https://github.com/me50/USERNAME) configuración, reduciendo el tamaño de su envío por debajo de 100MB, o intente cargar directamente [usando la interfaz web de GitHub.s \(https://docs.github.com/en/free-pro-team@latest/github/managing-files-in-a-repository/adding-a-file-to-a-repository\)](https://docs.github.com/en/free-pro-team@latest/github/managing-files-in-a-repository/adding-a-file-to-a-repository) visitando [github.com/me50/USERNAME \(https://github.com/me50/USERNAME\)](https://github.com/me50/USERNAME) (donde `USERNAME` es su propio nombre de usuario de GitHub) y arrastrando y soltando manualmente carpetas, asegurando que al cargarlo lo está haciendo a su `cs50/problems/2024/x/project` Branch, de lo contrario el sistema no podrá comprobarlo.

Paso 3 de 3

Asegúrese de visitar su libro de calificaciones en cs50.me/cs50x (<https://cs50.me/cs50x>) unos minutos después de enviar. Es sólo cargando su Gradebook que el sistema puede comprobar para ver si ha completado el curso, y eso también es lo que desencadena la generación (instantánea) de su Certificado CS50 gratuito y la (dentro de los 30 días) de la generación (dentro de 30 días) del Certificado Verified de edX, si ha completado todas las otras asignaciones. Asegúrese de reclamar su certificado gratuito (siguiendo el enlace en la parte superior de su libro de calificaciones) antes del 1 de enero de 2025.

No te saltes el paso anterior. El curso no se considera completo hasta que hagas lo anterior y veas la pancarta verde diciendo que has completado el curso. Si no hace lo anterior antes del 1 de enero de 2025, su estado en el curso estará sujeto a **las reglas de prórroga** en las preguntas frecuentes. El personal no hará ninguna corrección manual a principios de 2025 en función de que esto se salte.

Eso es. Su proyecto debe ser clasificado en pocos minutos. Si no ves ningún resultado en tu libro de calificaciones, mejor para volver a enviar (corriendo lo anterior `submit50` comando) con sólo su archivo README.md esta vez. No hay necesidad de volver a presentar su forma.

Este era CS50x.