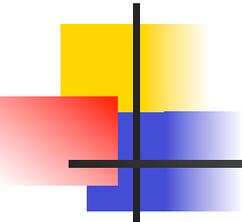


# Una Carta a los Estudiantes de Doctorado

---

Duane A. Bailey  
Department of Computer Science  
Williams College  
Williamstown, Massachusetts 01267  
[bailey@cs.williams.edu](mailto:bailey@cs.williams.edu)



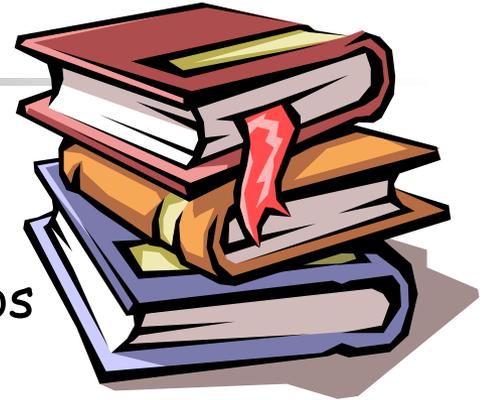
# Introducción

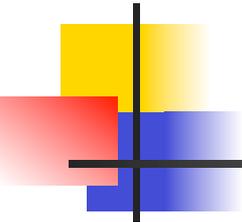
---

- La investigación puede hacerse más agradable y productiva mediante **un poco de organización**.
- Para muchos investigadores en computación, las técnicas de investigación son **un producto de la experiencia** en vez de, desafortunadamente, entrenamiento formal.
- Varias sugerencias organizacionales se resumen aquí, la mayoría de las cuales son útiles para el investigador por graduarse.
- Los artículos en cursiva están dirigidos a los estudiantes de posgrado, investigadores profesionales, y académicos.
- Disfrútalo.

# Lee mis consejos

- La investigación exitosa en ciencias de la computación involucra la coordinación de muchos esfuerzos.
- Para hacer tu tiempo tan productivo e interesante como sea posible he conjuntado unos cuantos "consejos organizacionales" que podrías encontrar útiles.
- Tómame el tiempo de leer estos consejos. Algunos son dolorosamente obvios (iotros son solo dolorosos!), pero cada uno tiene el potencial de mejorar tu vida como investigador.

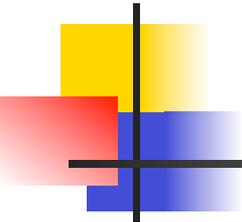




# 1. Leer es fundamental

---

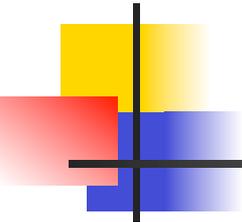
- Encontrar y leer trabajo relacionado es el fundamento de la buena investigación.
- Si eres nuevo en el campo, puede que solo estés familiarizado con el material presentado en libros de texto.
- Dos recursos iniciales e importantes son la *ACM Guide to Computing Literature* [3] y los *Computing Reviews* [2].
- Cuando comiences a investigar tu tópico, comienza con una lista de lecturas fundamentales.
- Tú, como investigador, eres responsable de desarrollar una bibliografía de trabajo relacionado, y de encontrar lecturas futuras.
- Puedes encontrar útiles referencias que otros no han tomado en cuenta.



# 1. Leer es fundamental

---

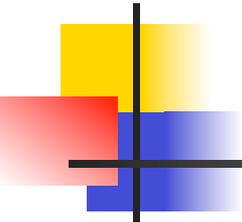
- **Continúa leyendo.**
- Aún cuando la lectura de antecedentes es la tarea inicial en cualquier empresa de investigación, **lecturas importantes suelen aparecer después**, especialmente si el enfoque de la investigación cambia.
- **Asegúrate de identificar las fuentes más importantes en tu campo y sigue las revistas conforme van llegando.** Te toma apenas un par de horas cada mes dar una revisión a los últimos artículos. (Las memorias de conferencias juegan un papel importante en rápidamente emitir resultados de investigación, aún cuando no son revisados tan rigurosamente como los artículos de revista.)
- Ya que la computación cambia rápidamente, **mantenerse al día con las lecturas técnicas recientes es vital.**



# 1. Leer es fundamental

---

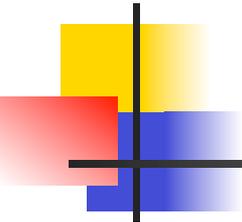
- **Lee los artículos con cuidado.** Una revisión en un curso de un trabajo puede identificar material básico.
- Cuando un artículo hace una contribución, es importante **revisar tal artículo con mayor detalle.**
- Conforme lees, pregúntate lo siguiente:
  1. **¿Cuál es la contribución de esta investigación?**
  2. **¿Cómo esta contribución se relaciona con trabajo hallado previamente?**
  3. **¿Cuáles son las referencias más importantes citadas por el autor de este artículo?**
- Estas preguntas parecen obvias y poco sofisticadas, pero esto no es razón para evitar preguntárselas.



# 1. Leer es fundamental

---

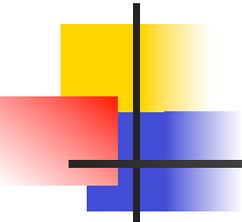
- **Escribe un resumen de cada artículo** que crees pertinente para tu investigación. Los resúmenes son útiles para rápidamente describir los trabajos de otros, y proveen un mecanismo de "arranque" si necesitas releer el material después.
- Durante una carrera de investigación, **resumir trabajo (tuyo y de otros) es una tarea común**; hacerlo concisamente es un arte.
- Después de leer un artículo bien escrito, considera su presentación. Puede ser que esté agradablemente organizado o que subraye conceptos con ejemplos bien pensados.
  1. **¿Qué hace este artículo fácil de leer?**
  2. **¿Qué nivel de detalle se provee?**
  3. **¿Qué ejemplos se usan para demostrar conceptos importantes?**
  4. **¿Qué preguntas deja sin responder?**
  5. **¿Pueden generalizarse los resultados?**



# 1. Leer es fundamental

---

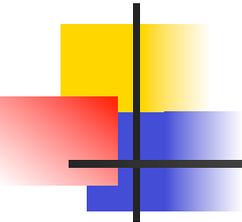
- Ya que aprendemos con ejemplos, **es importante señalar los artículos que están particularmente bien escritos.** Desafortunadamente, es frecuente que la literatura técnica esté obscuramente escrita.
- Ocasionalmente, puede ayudarte **leer un artículo con tu director de tesis, o con un colega interesado.** El ritmo puede ser lento, pero al final el potencial de un artículo bien escrito se hace notar, y tu comprensión de tal artículo mejorará.



# 1. Leer es fundamental

---

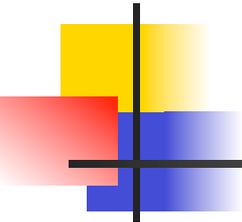
- *En grandes departamentos, un seminario informal o un "club de revistas" provee de una excusa para que los investigadores se reúnan semanalmente para discutir artículos. Se enfoca en un tópico ("El semestre de la recolección de basura"), pero permite que haya desviaciones cuando se desarrolla un interés común. Estas reuniones informales a la semana reservan tiempo en calendarios atestados para la investigación que podría de otro modo hacerse a un lado.*



## 2. Escribir es fundamental

---

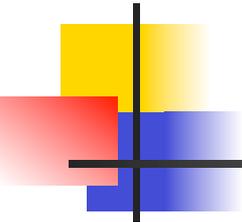
- La buena escritura es el único medio permanente del proceso científico.
- Mucho tiempo después de que las presentaciones se han desvanecido y los programas de estudio han cambiado, el trabajo escrito sirve para llevar conceptos importantes hacia adelante en el futuro.
- Así, necesitas comenzar a escribir tus resultados tan pronto como sea posible. Aún cuando la experimentación es importante, **estos esfuerzos se pierden a menos que sean cuidadosamente registrados.**



## 2. Escribir es fundamental

---

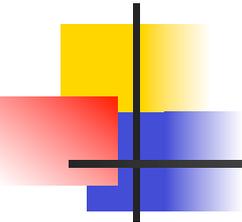
- **Escribe correctamente.** Cuando tengas dudas, la escritura sencilla siempre es preferible [11].
- Si eres un escritor nuevo o infrecuente, **debes preocuparte por ejercitar tus habilidades de escritura.** Si tienes suerte, podrás hallar en tus lecturas ejemplos de escritura exitosa en tu área.
- Generalmente, los artículos más efectivos dejan ver los puntos importantes con una sola lectura. Y discuten ejemplos interesantes.
- Cuando se presentan detalles técnicos directamente, ya sea matemáticas o código, éstos no substituyen al lenguaje natural.
- **Escribir en un estilo que provea la información correcta y concisamente es difícil.**
- Cuando no estés seguro de tu estilo, relee el artículo que más te haya gustado para integrar las cualidades estilísticas a tu escritura.



## 2. Escribir es fundamental

---

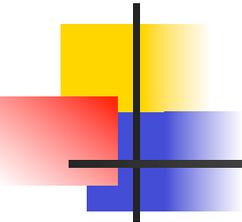
- **Escribe un cuaderno de notas como diario.** En este diario escribe las referencias, propón preguntas, describe los problemas y sus soluciones.
- **Mantén un historial de los experimentos y sus resultados.**
- Como los científicos de laboratorio saben, el diario es una herramienta simple para organizar tu investigación y es un registro valioso de progreso que vas haciendo.
- **Mantén un resumen de los resultados hallados en lecturas relacionadas.** Si esperas presentar formalmente tu investigación (¡siempre sé optimista!), posicionar tu trabajo respecto al de otros es un servicio para tus lectores.
- Reconsiderar investigaciones relacionadas también sirve para **recordarte de contribuciones importantes**, y de alinear tus metas con las de otros en tu área.
- Ya que este documento necesita ser editado frecuentemente, puedes **almacenarlo electrónicamente** (BibTEX [10] es un buen sistema de organización bibliográfica y es un mecanismo portable para intercambiar referencias con otros).



## 2. Escribir es fundamental

---

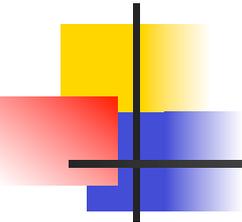
- *Mantén un grupo de pequeños proyectos en papel. Éstas son una o dos páginas de ideas presentadas formalmente sobre un solo tema. Cuando a las ideas no se les permite salir y vagan por ahí, se vuelven malformadas y desaliñadas.*
- *Estas escrituras mantienen el lápiz afilado y sirven para organizar ideas espontáneas. Tener éstas disponibles paga al momento de llamadas a artículos o cuando los estudiantes están buscando proyectos de investigación.*



## 2. Escribir es fundamental

---

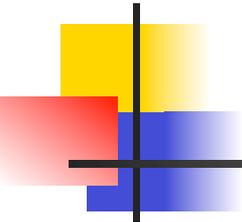
- Reserva una buena porción de tu tiempo para escribir. **Una hora de escritura es una hora considerando un problema en lugar de, digamos, forcejear con la computadora.**
- Debes tener tiempo **sin distracciones** para documentar tu trabajo y enfocar tus esfuerzos. Como un ejercicio regular, necesita hacerse, y llegarás a disfrutarlo.



## 3. Trabajando con otros

---

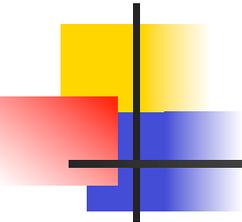
- Para muchos, el éxito proviene de trabajar con otros. **Es importante compartir ideas y permitirles desarrollarse en una atmósfera de grupo** (en departamentos pequeños esto puede ser problemático, especialmente si uno está en un campo aislado; pero los esfuerzos cooperativos siguen siendo importantes).
- Cuanto sea posible, haz uso de tu director de tesis o tus pares de investigación **como un panel de coherencia para tus ideas**.
- Podrías pensar que alguna de tus ideas no vale la pena, pero ponla a prueba de cualquier modo.
- ¡Mantén la conversación! La mayoría de los investigadores están felices de participar en discusiones productivas.



## 3. Trabajando con otros

---

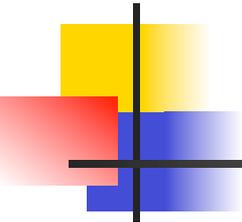
- **Mantén un horario regular de reuniones.**
- Si parece que no hay "nada que discutir", ese es un tópico importante. **Una reunión no atendida o cancelada produce un precedente que es difícil de corregir.**
- Si te sales de tu grupo, **debes re-establecer contacto.** Es poco probable que ellos te busquen, y el interés común tiende a decaer con el tiempo.
- **Las reuniones proveen de puntos de sincronización donde la concentración se reenfoca en el tema a discutir.**
- Los temas importantes se hacen, así que tómate el tiempo de escribir estas ideas (¡trae tu diario!) y distribuye copias.
- Estos minutos "informales" ayudan a mantener el contexto entre las discusiones y las decisiones documentadas que se han hecho.
- Reordenar material usado es generalmente una pérdida de tiempo.



## 3. Trabajando con otros

---

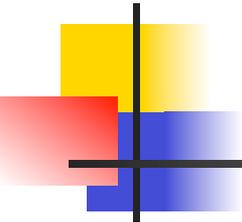
- **Considera cuidadosamente la crítica.**
- **Ya que la revisión por pares es un componente integral del proceso científico, habrá momentos en que tu investigación sea criticada.**
- **No tomes los comentarios críticos de manera personal. En lugar de eso, úsalos como guías para hacer tus argumentos más efectivos.**
- **De igual manera, cuando critiques el trabajo de otros, es importante hacer tus argumentos constructivos; el comentario no-constructivo normalmente se ignora con todo derecho.**



## 4. Las presentaciones no son sencillas

---

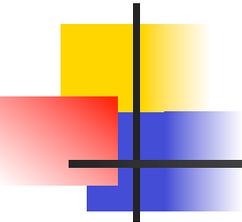
- En cualquier investigación seria, te encontrarás en la posición de **dar una presentación**.
- Las buenas presentaciones requieren de considerable preparación, **de modo que comienza temprano**.
- Las diapositivas deben ser legibles y eliminar todo elemento "no-esencial", incluyendo código.
- **Deben contener ejemplos ilustrativos**, pero solo al nivel de detalle que es apropiado para tu audiencia.
- Las diapositivas deben trabajar en conjunto para **documentar tus contribuciones**.
- **Ejercita y mide tus presentaciones**. Las presentaciones que no han sido preparadas son obvias, y generalmente causan que una audiencia originalmente abierta monte todo un ataque.
- Una presentación que está bien estructurada y ejercitada te prepara mejor para responder a aquella **Pregunta Inesperada**.



## 5. El proyecto

---

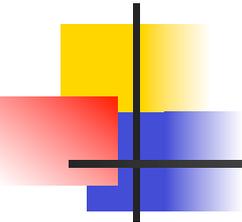
- La investigación en computación frecuentemente lleva al desarrollo de un “proyecto” que involucra programación.
- Es importante recordar que **hacer programación no es hacer investigación en computación**. En lugar de ello, la programación es meramente un mecanismo para realizar un experimento.
- Como cualquier experimento, debe ser cuidadosamente planeado con suficiente previsión.



## 5.1 El proyecto: Establece metas

---

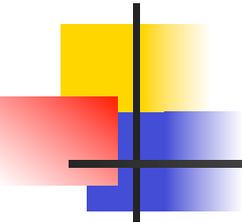
- Debes saber hacia dónde te diriges, y acercarte a la solución sin distracción.
- Desarrolla una lista de metas que demuestren tu progreso, y esfuérzate por cumplirlas.
- Si no puedes formular metas concisas, debes detenerte y reconsiderar las motivaciones para el proyecto.



## 5.2 El proyecto: Piensa simple

---

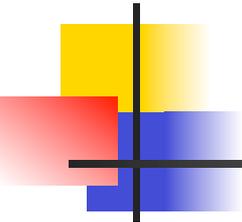
- Diseña tus proyectos de modo que puedan completarse en **un periodo de tiempo razonable**.
- Un programador con experiencia genera **poco menos de cien líneas de código confiable por mes**. Un proyecto que demanda miles de líneas de código, por lo tanto, toma algo más de un par de semestres para implementarse correctamente.
- El tiempo dedicado a reducir el experimento a un tamaño manejable **es un tiempo bien usado**.
- Los proyectos grandes **no necesariamente arrojan grandes resultados**.



## 5.3 El proyecto: Construye prototipos

---

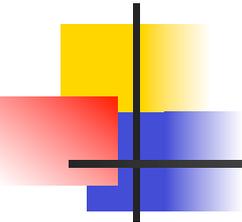
- La mayoría de los proyectos tratan de la construcción de un prototipo.
- Un prototipo bien considerado valida los supuestos, prueba el valor de las abstracciones y motiva reconsiderar ideas débiles.
- Mientras que hay poco *valor de investigación* asociado con pulir un "producto", muchas preguntas de investigación pueden ser respondidas satisfactoriamente mediante maquetas o implementaciones parciales.



## 5.4 El proyecto: Usa herramientas

---

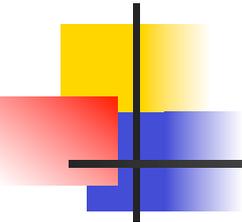
- El desempeño de un programador se mejora dramáticamente mediante el uso de **unas pocas herramientas simples**.
- Los programadores de hoy deben estar al tanto de, por ejemplo, `anim` [6], `awk` [1, 4, 5], `bison`[7], `HyperCard` [8], and `Mathematica` [13].
- En ambientes de ventanas, los constructores de interfaces pueden eliminar muchas de las distracciones del diseño de interfaces.
- Hay por supuesto, muchas otras herramientas importantes y útiles, pero el punto principal es claro: **la correcta selección de herramientas puede reducir el trabajo total en un proyecto**.
- **Encuéntralas, aprende de ellas y úsalas.**



## 5.5 El proyecto: Colabora

---

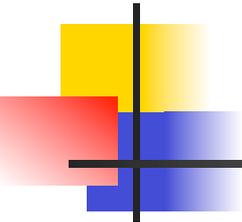
- *Cuando los recursos pueden ser coordinados, los esfuerzos en grupos son frecuentemente más productivos que los esfuerzos individuales.*
- *Si compartes intereses de investigación con otros, **contáctalos y colabora**. Sin duda, ellos habrán resuelto problemas que tu estas actualmente considerando, y sus soluciones pueden influenciar cómo lograr metas comunes.*
- *Un efecto colateral de la colaboración es el **aumento de la disciplina**, que es necesaria para reducir la cantidad de energía que se dedica a sincronizar esfuerzos.*
- *Para los programadores, una variedad de recursos se encuentran actualmente disponibles para administrar proyectos distribuidos. Por ejemplo, un conjunto de estándares de codificación han sido desarrollados por el proyecto GNU, y un estio de codificación común puede ser adecuadamente apoyado por los propios programas (v.gr. Indentación).*
- *Cuando se comparte el acceso a un proyecto, un sistema de control de versiones debe ser considerado (v.gr. SCCS o RCS8).*



## 5.6 El proyecto: Documenta los resultados

---

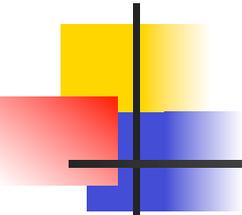
- Los proyectos terminados **deben documentarse**.
- Como mínimo, una revisión técnica del experimento permite a otros ver **qué motiva tu investigación**.
- El documento debe describir el problema, tus supuestos, tu aproximación, y **una evaluación honesta de tus resultados**.
- Cuando documentes software, incluye ejemplos ilustrativos, tutoriales, y cualquier experiencia obtenida de su uso.
- Documentación correctamente escrita incrementa grandemente el impacto del proyecto.



## 6. Resumen

---

- Tu participación en la investigación es parte de un desarrollo incremental.
- Tu trabajo se construye sobre el de otros y produce algo de utilidad.
- Como un participante de un campo que se mueve rápidamente, eres responsable de mantenerte informado, desarrollar y realizar experimentos adecuados para probar tus hipótesis, y presentar tus resultados de manera que los haga accesibles para la comunidad de investigación.
- Las sugerencias que se presentan aquí pueden ayudarte a realizar estas responsabilidades.



# 7. Referencias

---

1. Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger. **The AWK Programming Language**. Addison-Wesley, Reading, Massachusetts, 1988.
2. Association for Computing Machinery. **ACM Computing Reviews**. Association for Computing Machinery, New York, 1960.
3. Association for Computing Machinery. **ACM Guide to Computing Literature**. Association for Computing Literature, New York, 1977.
4. Jon L. Bentley. **Programming Pearls**. Addison-Wesley, Reading, Massachusetts, 1986.
5. Jon L. Bentley. **More Programming Pearls: Confessions of a Coder**. Addison-Wesley, Reading, Massachusetts, 1988.
6. Jon L. Bentley and Brian W. Kernighan. **A system for algorithm animation tutorial and user's manual**. Technical Report 132, AT&T Bell Laboratories, Murray Hill, New Jersey, 1987.
7. Charles Donnelly and Richard M. Stallman. **Bison: The YACC-compatible Parser Generator**. The Free Software Foundation, Cambridge, Massachusetts, 1991. Available via anonymous ftp from prep.ai.mit.edu.
8. Danny Goodman. **The Complete HyperCard 2.0 Handbook**. Bantam Books, Toronto, third edition, 1990.
9. Donald E. Knuth. **Literate Programming**. Center for the Study of Language and Information, Stanford University, 1992.
10. Leslie Lamport. **LATEX: A Document Preparation System**. Addison-Wesley, Reading, Massachusetts, 1986.
11. William Strunk and E. B. White. **Elements of Style**. Macmillan, New York, third edition, 1979.
12. N. Wirth. **The programming language Oberon**. Technical report, ETH, Zurich, January 1990. Available via anonymous ftp from neptune.ethz.ch:Oberon/Docu/OberonReport.ps.
13. Stephen Wolfram. **Mathematica: A System for Doing Mathematics by Computer**. Addison-Wesley, Redwood City, second edition, 1988.