

# Frontend Fundamentals

# Responsive design



# Introducción

En el desarrollo web moderno es imprescindible pensar en los diferentes usuarios que consumirán los servicios de tu página, como pueden ser usuarios de una computadora, de un teléfono móvil o una tablet.

Por ello conocerás cómo la filosofía de diseño "mobile-first" o "desktop-first", te permiten estructurar el contenido de tu página para que sea vista en la mayor cantidad disponible de dispositivos, manteniendo un diseño claro y amigable con el usuario.

Después de practicar la "responsividad" en nuestro sitio, iniciaremos la implementación de elementos de uno de los CSS frameworks tipo responsive más utilizado y prepararemos el código para recibir nuevos elementos prediseñados, lo cual te permitirá acelerar significativamente el desarrollo de tu página.

Crear una versión móvil de un proyecto y una versión para escritorio es una idea que carece de funcionamiento y practicidad, por el tiempo y costos que representa para las empresas; la idea de crear una versión que pueda visualizarse en cada dispositivo con acceso a Internet es cada vez menos viable ya que cada dispositivo cuenta con una resolución y orientación diferente.

Con la llegada de los teléfonos móviles con acceso a Internet de alta velocidad, las reglas para el diseño de las páginas y aplicaciones web han sido orientadas a la correcta visualización y optimización para estos.

## RESPONSIVE WEB DESIGN

ADAPTIVE PAGES ON ALL DEVICES



Las primeras páginas web móviles se programaron de manera muy simple, pero con el progreso de las tecnologías como HTML, CSS y JavaScript, la aparición de nuevas terminales y tabletas, las páginas y aplicaciones se fueron haciendo cada vez más complejas.

Una página web móvil se ha convertido hoy en día en un complemento adicional en forma reducida de la versión de escritorio de una web, por lo que es necesario el uso del diseño web receptivo o responsive design que trata sobre la creación de páginas web; la idea es que se muestren correctamente en todos los dispositivos por medio de un ajuste automático dependiendo del tamaño de la pantalla o ventana.

# Introducción

W3Schools define al responsive design como el uso de HTML y CSS para cambiar el tamaño, ocultar, reducir o ampliar automáticamente un sitio web para que se muestre correctamente en todos los dispositivos (W3Schools, 2022).

## ¿Qué es responsive design?

Al tratarse de desarrollo web, no se puede dejar de lado el tema del diseño web responsive ya que se ha convertido en un factor oficial para el ranking de Google. Una página web responsive logra diferenciarse de páginas web “antiguas” en que no existen dos versiones diferentes de una misma página (una versión para escritorio y otra para móvil). Todos los elementos como bloques de texto, imágenes, etc., se integran en una misma plantilla y solo varía su visualización dependiendo del dispositivo en donde se intenta acceder a la página.

**Ventajas:** funcionan en todo tipo de plataformas: una plantilla funciona para todos los dispositivos por medio de una URL.

**Inconvenientes:** su planificación y desarrollo requieren de un presupuesto elevado; “traducir” una web de escritorio para ajustarla al diseño web responsive, suele ser un proceso sumamente complejo.

Todo comienza identificando si desarrollaremos una web desde cero o haremos responsive una web creada previamente.

Existen herramientas online para verificar si un sitio es responsive, tomando en cuenta que muchos desarrollos ya están montados en un host (por ejemplo, Netlify). A continuación, se muestra un listado de herramientas online gratuitas y que no requieren de registro: **Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.**

responsivepx: <http://www.responsivepx.com>

Screencheck de Cyber Crab: <http://responsivetesttool.com>

Demonstrating Responsive Design: <https://jamus.co.uk/demos/rwd-demonstrations/>

Responsinator: <http://www.responsinator.com>

Am I Responsive?: <https://ui.dev/amiresponsive>

En el caso que el desarrollo se encuentre en una etapa temprana y en un servidor local, la herramienta adecuada sería el **inspeccionador de elementos** que viene integrado en tu navegador. Generalmente se accede en el menú de navegación; haz clic en la opción “Desarrollador Web” y luego en “Vista de diseño adaptable”.

Al desarrollar un proyecto desde cero, son pocas las consideraciones que nos permitirán trabajar con diseño responsive; lo primero es indicarle al navegador la configuración de la ventana, esto se hace añadiendo la siguiente etiqueta meta a nuestra página web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Observa la línea 6 del siguiente código:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Esta etiqueta va a configurar la ventana del navegador con las instrucciones para tener control sobre las dimensiones y la escala.



Figura 1. HTML sin la etiqueta meta.

Figura 2. HTML con la etiqueta meta.meta.

Fuente: W3Schools. (2022). *HTML Responsive Web Design*. Recuperado de [https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp) solo con fines educativos.

## Media queries

Las media queries (o “consultas de medios” en español) son muy útiles cuando en función del tipo de dispositivo, la resolución del ancho de la pantalla cambia, esto quiere decir que obtendremos estilos completamente diferentes para diversos tamaños de pantalla o de la ventana del navegador.

Las media queries son elementos escritos en CSS que nos permiten estilos aplicables cuando se cumplan condiciones específicas; se ejecutarán en caso de que se cumpla una condición (generalmente asociada a la resolución de una pantalla o ventana).

### Sintaxis

El portal para developers de Mozilla aclara que el resultado de una consulta será “verdadero” cuando el tipo de medio en caso de especificarlo coincida con el dispositivo en el que se muestra el documento y todas las expresiones en el media query sean “verdaderas”.

En este caso, se aplicarán los estilos que hayamos escrito dentro del media query. Las consultas sobre tipos de medios desconocidos son siempre falsas (mdn, 2022).



```
1 <!-- CSS media query on a link element -->
2 <link rel="stylesheet" media="(max-width: 800px)" href="example.css" />
3
4 <!-- CSS media query within a style sheet -->
5 <style>
6 @media (max-width: 600px) {
7   .facet_sidebar {
8     display: none;
9   }
10 }
11 </style>
```

Fuente: mdn. (2022). CSS media queries. Recuperado de [https://developer.mozilla.org/es/docs/Web/CSS/Media\\_Queries/Using\\_media\\_queries](https://developer.mozilla.org/es/docs/Web/CSS/Media_Queries/Using_media_queries) solo con fines educativos.

Se pueden redactar queries utilizando operadores lógicos como **not**, **and** y **only**. Un query básico con el tipo de medio especificado como all puede lucir así:

```
@media (min-width: 700px) { ... }
```

El operador and es usado para colocar juntas, múltiples funciones multimedia. La siguiente query solo retornará verdadero si la ventana tiene un ancho de 700px o más y la pantalla está en formato horizontal:

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

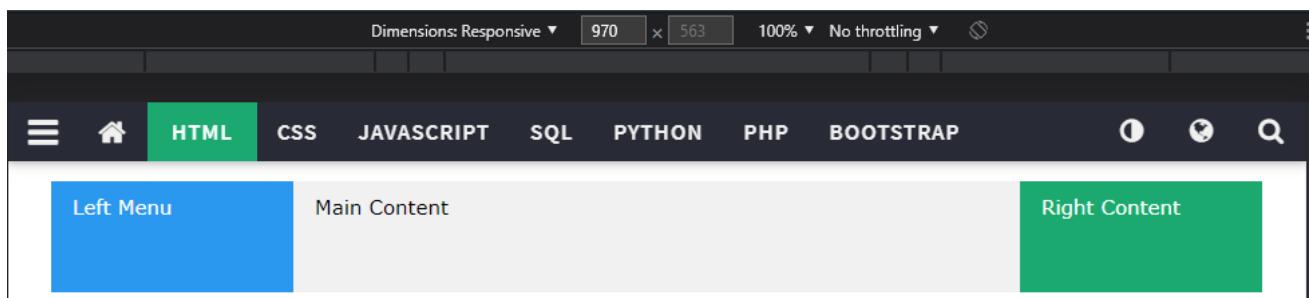
Ahora, si se quiere aplicar esta opción, solo si el tipo de medio es TV se puede utilizar la siguiente cadena:

```
@media tv and (min-width: 700px) and (orientation: landscape) { ... }
```

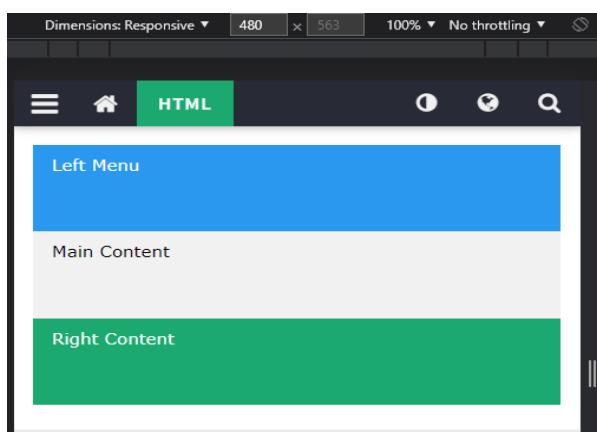
Observa el siguiente ejemplo propuesto por el portal W3schools en donde 3 divs adquieren un comportamiento diferente a partir del ancho de la ventana utilizando la siguiente media query:

```
@media screen and (max-width: 800px) {
    .left, .main, .right {
        width: 100%; /* The width is 100%, when the viewport is 800px or smaller */
    }
}
```

Con una resolución mayor a 800 pixeles, los *divs* se comportan de la siguiente forma:



Con una resolución inferior a 800 pixeles, los *divs* se comportan de la siguiente forma:



Estas pantallas se obtuvieron directamente del software que se está explicando en la computadora, para fines educativos.

# Explicación

Puedes acceder al simulador de W3schools donde se encuentra el código completo y lograrás modificar las reglas. Ingresa mediante el siguiente enlace: **Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.**

[https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_responsive\\_media\\_query](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_responsive_media_query)

## Unidades fijas

Son aquellas que especifican una medida en términos absolutos, por ejemplo, 300px (300 píxeles). La unidad declarada tiene un valor de 300 píxeles y siempre será así a menos que una nueva regla sobrescriba ese valor. Otros ejemplos de unidades fijas son los siguientes:

- **px**: píxeles.
- **in**: pulgadas (1 in es igual a 96px).
- **pt**: puntos (1 pt es igual a 1/72 in).
- **cm**: centímetros.
- **mm**: milímetros.
- **pc**: picas.

## Unidades relativas

Alvarez (2015), menciona que las unidades relativas “*son aquellas que tienen en cuenta el contexto en el que se encuentran; son relativas a las dimensiones del contenedor donde se han definido*”. Por ejemplo, % es una unidad relativa, puesto que el 60% de ancho no será lo mismo para un elemento situado dentro de un *div* con 1000px de ancho o sobre un contenedor con 860px de ancho. A continuación, un listado de unidades relativas:

Unidad	Relativa a
<b>em</b>	Tamaño de letra del elemento padre (para propiedades como font-size) y tamaño de la fuente del elemento en caso de otras propiedades CSS como ‘width’
<b>ex</b>	Altura en el eje x de la fuente del elemento
<b>ch</b>	La fuente, igual que ex pero con la anchura del carácter 0
<b>rem</b>	Tamaño de la letra del elemento raíz. Por default, en Chrome es de 16px. (1rem = 16px)
<b>lh</b>	Altura de la línea del elemento
<b>vw</b>	% (porcentaje) del ancho de la ventana gráfica
<b>vh</b>	% (porcentaje) de la altura de la ventana gráfica
<b>vmin</b>	1% de la dimensión más pequeña de la ventana gráfica
<b>wmax</b>	1% de la dimensión más grande de la ventana gráfica

Tomaremos como ejemplo la unidad em: “em” es una unidad que equivale al tamaño de texto en un contenedor. 1em es el tamaño de la fuente definido en un contenedor. El tamaño del texto predeterminado en un navegador es de 16 píxeles, por lo tanto y a no ser que lo cambiemos, 1em en el body será equivalente a 16px, mientras que 10em será equivalente a  $10 \times 16$  píxeles = 160px.

## rem vs em vs px

Conforme desarrolles tus proyectos de forma responsiva, observarás que existen 3 unidades relativas que se encargan de cambiar tamaños.

## CSS Framework

De manera personal e individual intenta centrar un div y que sea funcional bajo cualquier resolución de la pantalla sin la ayuda de nadie. Algo tan simple puede representar un reto medianamente complicado, ya que simplemente es centrar un cuadro dentro de otro cuadro. La necesidad de un “algo” que nos ayude con esas tareas dio pie a la llegada de los framework.

Un framework es un esquema o marco de trabajo que nos ofrece una estructura base para elaborar un proyecto con objetivos específicos, se podría decir que es una especie de plantilla que funciona como punto de partida para la organización del desarrollo, por lo que su uso nos puede ayudar a simplificar muchas tareas.

A continuación, una lista de algunos frameworks de CSS:

**Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.**

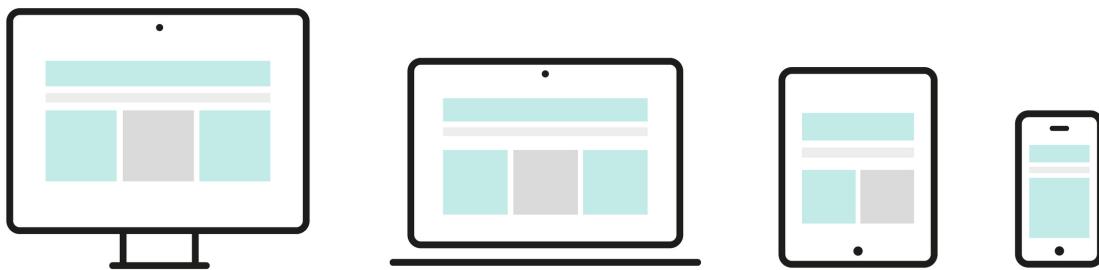
- Bootstrap: <https://getbootstrap.com>
- Foundation: <https://get.foundation>
- Bulma: <https://bulma.io>
- UIkit: <https://getuikit.com>
- Semantic UI: <https://semantic-ui.com>
- Susy: <https://www.oddbird.net/susy/>
- Materialize: <https://materializecss.com>
- Pure: <https://purecss.io>
- Skeleton: <http://getskeleton.com>
- Miligram: <https://milligram.io>
- Tailwindcss: <https://tailwindcss.com>

Para este curso, el framework a utilizar será Bootstrap debido a su amplia popularidad, así como a sus múltiples características tales como la facilidad al crear de forma rápida prototipos, un gran ecosistema, documentación, una gran cantidad de componentes, soporte para LESS y SASS, y por supuesto el respaldo de Twitter.

Puedes acceder a los archivos .txt de código en el apartado “Archivos adjuntos”.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Durante este tema exploramos conceptos importantes a considerar relacionados con el desarrollo web, pero debemos hablar sobre su importancia. Los sitios web responsive proporcionan al usuario facilidades a la hora de navegar y requieren menos tiempo de carga. Además, ofrecen una experiencia de usuario más natural que se traduce en una percepción más positiva del sitio desarrollado.



También hay que remarcar que el diseño responsive ahorra tiempo y esfuerzo a los diseñadores web que ya no tienen que desarrollar y ejecutar dos versiones del sitio web, pudiendo dedicarse a otro tipo de proyectos y aumentando consigo la productividad.

Por otra parte, no debemos olvidar que un proyecto desarrollado con metodologías obsoletas, tarde o temprano tendrá que adoptar un diseño responsivo, saliendo en muchas ocasiones más caro en términos de económicos y de tiempo.

Tú mismo te darás cuenta de que en la mayoría de las ocasiones, es mejor, más fácil y económico realizar nuevamente el proyecto desde cero, en lugar de traducir poco a poco cada parte para que sea responsive.

# Referencias bibliográficas

- Alvarez, M. (2015). *Unidades de medida en CSS: relativas y absolutas*. Recuperado de <https://desarrolloweb.com/articulos/unidades-medida-css-relativas-absolutas.html>
- mdn. (2022). *CSS media queries*. Recuperado de [https://developer.mozilla.org/es/docs/Web/CSS/Media\\_Queries/Using\\_media\\_queries](https://developer.mozilla.org/es/docs/Web/CSS/Media_Queries/Using_media_queries)
- W3Schools. (2022). *HTML Responsive Web Design*. Recuperado de [https://www.w3schools.com/html/html\\_responsive.asp](https://www.w3schools.com/html/html_responsive.asp)

## Para saber más

**Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.**

### Lecturas

Para conocer más acerca de **Media Query**, te sugerimos leer lo siguiente:

- JavierVallejo. (s.f). *Qué es una Media Query (con ejemplos)*. Recuperado de <https://javiervallejo.com/ejemplos-de-uso-media-query/>

Para conocer más acerca de **Unidades Relativas**, te sugerimos revisar lo siguiente:

- LenguajeCSS. (s.f). *Unidades CSS*. Recuperado de <https://lenguajecss.com/css/modelo-de-cajas/unidades-css/>

Para conocer más acerca de **Responsive design**, te sugerimos leer lo siguiente:

- Platzi. (2020). *Diseño responsive: 7 ejemplos magníficos para entenderlo*. Recuperado de <https://platzi.com/blog/diseno-responsive-ejemplos/>

### Videos

Para conocer más acerca de **Responsive Design**, te sugerimos revisar lo siguiente:

- Jesse Showalter. (2020, 3 de enero). *Responsive Web design | 10 Basics*. Recuperado de <https://www.youtube.com/watch?v=zF6VSKy4Slc&t=66s>
- Vida MRR - Diseño y desarrollo web. (2019). *Tutorial crear un sitio web responsive design* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=cfzqMi1SBQ>

Para conocer más acerca de **Media Query**, te sugerimos revisar lo siguiente:

- Vida MRR - Diseño y desarrollo web. (2018). *Introducción a media queries | CSS Responsive Design* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=abelQallu5k>

Para conocer más acerca de **Unidades Relativas**, te sugerimos revisar lo siguiente:

- Vida MRR - Diseño y desarrollo web. (2020). *UNIDADES EN CSS: REM VS EM VS PX* [Archivo de video]. Recuperado de [https://www.youtube.com/watch?v=p\\_DVRE4uZgw](https://www.youtube.com/watch?v=p_DVRE4uZgw)

## Asegúrate de:

- Comprender qué es *Responsive web design*.
- Identificar las ventajas del *Responsive web design*.
- Identificar *Media queries* y saber darles un uso correcto
- Tener claro el panorama respecto a los *frameworks css*

# Requerimientos Técnicos

- Computadora con acceso a Internet.

# Prework

¿Has notado que cada día más dispositivos tienen conexión a Internet? ¿Puedes imaginar la dificultad que representa para diseñadores y programadores, crear, planificar, adaptar y desarrollar contenido que se vea bien en cada dispositivo? ¿Sueles navegar en Internet en su *smartwatch*, tableta, *smart TV*, *smartphone*, *laptop*, *netbook* o PC de escritorio? Durante esta actividad guiada explorarás conceptos y soluciones que te ayudarán a comprender como se abordan estas problemáticas.

Antes de seguir con tu actividad guiada, considera lo siguiente:

Algunos recursos textuales y de video pueden estar en inglés, si no tienes dominio sobre el idioma te recomendamos el uso de un traductor

( <https://translate.google.com.mx/?hl=es> ) y de activar los subtítulos en el reproductor de YouTube:

1. Busca el logo de configuración (⚙️) dentro de la ventana del video.
2. Haz clic ahí y selecciona Subtítulos.
3. Haz clic en Activar.

Bienvenido a tu Prework. A continuación, listaremos algunos materiales y recursos que te serán de ayuda. Ten en cuenta que completar tu Prework te va a preparar para tener conceptos claros.

## Responsive web design

En esta sección explorarás conceptos claves que te ayudarán a entender la forma en cómo las páginas web se adaptan en función al tamaño de la pantalla o la ventana del navegador. Es importante tener claros los siguientes conceptos:

- Cómo configurar la ventana gráfica.
- Dimensiones de las imágenes.
- Flexbox.
- Consultas de medios.
- Breakpoints.
- Optimización del texto.

Todos estos conceptos los vas a explorar en el siguiente documento:

web.dev. (2019). *Conceptos básicos del diseño web responsive*. Recuperado de <https://web.dev/i18n/es/responsive-web-design-basics/>

## Media queries

En este tema vas a trabajar con diferentes resoluciones dentro del navegador; es recomendable que sepas identificar en dónde se encuentran las *DevTools* de tu navegador; revisa el siguiente documento:

Chrome Developers. (2018). *Open Chrome DevTools*. Recuperado de <https://developer.chrome.com/docs/devtools/open/>

Es conveniente que te familiarices en cómo simular el tamaño de ciertos dispositivos; revisa el siguiente documento:

Chrome Developers. (2018). *Simulate mobile devices with Device Mode*. Recuperado de <https://developer.chrome.com/docs/devtools/device-mode/#toggle>

Es importante tener claro un listado de pantallas y *viewports* comunes que manejan dispositivos y computadoras de escritorio; revisa el siguiente documento:

desarrolloweb.com (2015). *Medidas estándar de pantallas para tus media queries*. Recuperado de <https://desarrolloweb.com/articulos/medidas-estandar-pantallas-media-queries.html>

## Unidades relativas

En este apartado explorarás las formas en que se definen lo alto o ancho de elementos, así como tamaños de letra y comportamientos de imágenes o contenedores; revisa el siguiente material:

dis.um.es. (s.f.). *Unidades de medida*. Recuperado de [http://dis.um.es/~lopezquesada/documentos/IES\\_1213/LMSGI/curso/css/css11/Paginas/medidas.html#:~:text=Las%20medidas%20relativas%20definen%20su,es%20directamente%20el%20valor%20indicado](http://dis.um.es/~lopezquesada/documentos/IES_1213/LMSGI/curso/css/css11/Paginas/medidas.html#:~:text=Las%20medidas%20relativas%20definen%20su,es%20directamente%20el%20valor%20indicado).

## CSS Framework

La escritura del código directamente en CSS puede ser un proceso que lleva tiempo y esfuerzo; el uso de un framework puede ser una vía atractiva si el objetivo es desarrollar de forma rápida y adaptándose a los estándares actuales. Usar un framework CSS puede tener una curva de aprendizaje algo elevada (dependiendo del framework), pero el tiempo que se ahorra durante el desarrollo de proyectos medianos o grandes, compensa con creces el tiempo invertido. Debes conocer las ventajas y desventajas que conlleva el uso de un framework:

openwebinars. (2019). *Ventajas y desventajas de usar Framework CSS*. Recuperado de <https://openwebinars.net/blog/ventajas-y-desventajas-de-usar-framework-css/>

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.