

Elementos de Programación y Lógica.

Unidad 1 - Clase 2.

Conceptos Generales de Informática.

Lenguajes y Lenguaje de Mercado.

.

Sección 1. Lenguajes.

Lenguajes.

Lenguaje **Vamos a llamar lenguaje a un conjunto de símbolos y reglas que determinan textos que son válidos y textos que no lo son.**

Idiomas.

Los idiomas que hablamos todos los días son lenguajes.

Cada uno tiene su propio conjunto de símbolos (letras, números, signos de puntuación, etc.) y de reglas (palabras válidas, reglas gramaticales, etc.)

Generalmente se denomina “lenguaje natural” al lenguaje que uno emplea para comunicarse habitualmente, es decir, a nuestro idioma.

Descripción de imagen: Banderas de distintos países.

Idiomas Artificiales.

Hay idiomas que consisten en lenguajes artificiales, con distintas finalidades: artísticas, filantrópicas, científicas, etc.

El Klingon, el Esperanto, el Lojban son algunos ejemplos.

Descripción de imagen: Banderas de los lenguajes de idiomas artificiales.

Sintaxis.

Todo lenguaje tiene una sintaxis.

La sintaxis consiste en los símbolos y las reglas que determinan cómo disponer los mismos.

Si un texto contiene solamente símbolos válidos, dispuestos correctamente de forma tal que se cumplen todas las reglas, entonces ese texto es parte del lenguaje, caso contrario no lo será.

Sintaxis, orden y reglas de composición.

Siempre hay un orden, por ejemplo, en español, al hacer una pregunta se deben colocar los signos de interrogación que delimitan la pregunta (¿?).

¿Cuál es tu nombre?

En inglés, una pregunta está delimitada por la forma de la oración, terminando con el signo de interrogación de cierre (?).

What is your name?

Semántica.

Todo lenguaje tiene además una semántica.

La semántica indica qué significa una determinada secuencia de símbolos. Por ejemplo, que representación mental nos genera al leer la palabra “Pelota” o “Correr”.

Es eso que generalmente vemos descrito en el diccionario, aunque la semántica varía de acuerdo al contexto.

Yodalang: Lenguaje de ejemplo.

Por ejemplo, podemos definir un lenguaje que contenga como símbolos solo las letras del idioma español en mayúscula, y el espacio en blanco:

A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P, Q, R, S, T, U, V, W, X, Y, Z.

También vamos a aceptar las letras acentuadas.

Y como reglas, vamos a definir que los símbolos se pueden agrupar uno tras otro, formando palabras como en el español, pero que a diferencia de este idioma, en cada renglón solo puede haber tres palabras (o sea, solo puede haber dos espacios por renglón).

Yodalang: Sintaxis.

En el lenguaje que definimos estas frases son válidas:

- VÁLIDO ESTO ES.
- INVÁLIDO ESTO NO.
- VENCER SITH DEBES.

Estas frases no lo son:

- VÁLIDO ESTO NO ES.
- DARTH VADER TU PADRE ES.

Yodalang: Semántica.

En el lenguaje que definimos esas frases no significan absolutamente nada, pues nunca definimos su semántica.

Deberíamos, para definir el lenguaje, decir que significa cada palabra de nuestro lenguaje para determinar qué es lo que se debe interpretar de cada renglón, y como las frases deben ser interpretadas.

Ojo, cómo usamos los mismos símbolos del español, y las mismas palabras, nuestra mente tiende a asociar automáticamente un significado, pero esto no es válido.

Yodalang: Semántica, continuación.

Sin definir la semántica, es imposible determinar qué significa una frase:
VENCER SITH DEBES.

- ¿Debemos interpretar “**Debes vencer a los Sith**” o “**Sith, debes vencer**”?
- ¿Será que “**VENCER SITH DEBES**” es lo mismo que “**DEBES SITH VENCER**”?

Yodalang: Desambiguación.

Un humano puede entender lenguajes con sintaxis y semánticas complejas gracias a que puede pensar y desambiguar.

Las computadoras no pueden desambiguar ni entender contextos.

Los lenguajes que las computadoras pueden comprender y procesar suelen tener una sintaxis sencilla y una semántica sin ambigüedades.

Sección 2. Lenguajes de Marcado.

Lenguajes de Marcado.

Un lenguaje de marcado es un lenguaje en donde algunos símbolos cobran un significado especial. Son marcas que indican algo, por ejemplo qué función cumple un determinado texto, o cómo debería verse en pantalla con algún visualizador.

Ej. Whatsapp.

Descripción de imagen: Whatsapp mostrando mensajes con estilos y el código de marcado que lo genera

Lenguajes de marcado: Utilidad.

- Un lenguaje de marcado sirve para comunicar contenido de forma semántica.
- Un lenguaje de marcado no es un lenguaje de programación.

- Los lenguajes de marcado son utilizados en la industria editorial y de la comunicación, así como entre autores, editores e impresores.
- Los lenguajes de marcado son ampliamente usado para generar documentación que acompaña a proyectos de software.
- El lenguaje de marcado más popular, HTML, sirve para diseñar y estructurar un sitio web.

Lenguajes de marcado: Características.

Suelen ser **fáciles de leer y escribir por humanos**, pero no contienen ambigüedades de ningún tipo en su semántica, permitiendo que sean **interpretables por una computadora**.

Hay muchos lenguaje de marcado (cada lenguaje tiene su conjunto específico de reglas)

Cada lenguaje de marcado tiene una **utilidad específica**: generar documentos en PDF, escribir documentación de proyectos, describir páginas web, etc.

Vamos a mencionar algunos de ellos.

Parte 1. LaTeX.

LaTeX.

LaTeX es un lenguaje que incluye marcas que **permiten generar documentos listos para imprimir** (PDF o similares).

Es un lenguaje muy complejo, pero muy poderoso.

Es el lenguaje utilizado habitualmente para escribir papers científicos, tesis de licenciatura, libros técnicos, y mucho más.

Lo van a tener que usar en algún punto de la carrera, tanto para entregar TPs como para el trabajo final de la carrera.

LaTeX: Ejemplo.

Descripción de imagen: Código que genera un documento LaTeX

Descripción de imagen: El documento LaTeX compilado en un documento

Parte 2. BBCode.

BBCode.

BBCode es un lenguaje de marcado que es **utilizado en foros de internet** para embellecer los mensajes que envían los usuarios.

El código BBCode permite marcar que texto irá en negrita, subrayado, determinar el color, etc.

El código BBCode permite incluso agregar emoticones y cosas similares escribiendo simplemente una marca en el texto.

BBCode: Ejemplo.

Puede verse un ejemplo completo en: <http://www.roleplayerguild.com/bbcode>

Descripción de imagen: Muestra de las posibilidades de BBCode.

Parte 3. WikiText.

WikiText.

Es un lenguaje de marcado **diseñado específicamente para páginas de wikis**, por ejemplo Wikipedia.

Incluye marcas de estilo, como negrita, resaltado, etc. pero el símbolo usado para esas marcas depende del software que se utilice para la wiki.

Wikipedia ha propuesto un estándar conocido como MediaWiki que ha sido ampliamente adoptado.

WikiText: Ejemplo.

Descripción de imagen: Muestra de código de WikiText en Wikipedia

Parte 4. Markdown.

Markdown.

Es un lenguaje de marcado **diseñado para maximizar la facilidad de lectura**, tanto en su entrada como en su salida.

Se inspira fuertemente en convenciones usadas al momento de enviar correos electrónicos en texto plano.

Muchos sistemas de blogs permiten escribir artículos en código Markdown.

En programación es ampliamente utilizado para escribir documentación de los

programas.

Veremos más de este lenguaje en las próximas clases

Parte 5. HTML

HTML.

HyperText Markup Language (HTML) es un lenguaje de marcado que sirve para **describir páginas web**.

Es el lenguaje de marcado más extendido en su uso.

El principio básico del HTML son las **etiquetas**, que actúan de marcas para indicar distintos tipos de contenidos en un sitio web.

HTML se orienta a describir qué contenidos van a ser presentados en un sitio web, y estructurarlos de forma semántica.

También lo veremos en profundidad en la próxima clase

Sección 3. Markdown.

Parte 1. El lenguaje.

Markdown.

Es un lenguaje de marcado **diseñado para maximizar la facilidad de lectura**, tanto en su entrada como en su salida.

Se inspira fuertemente en convenciones usadas al momento de enviar correos electrónicos en texto plano.

Muchos sistemas de blogs permiten escribir artículos en código Markdown.

En programación es ampliamente utilizado para escribir documentación de los programas.

Su salida suele ser un documento HTML.

Markdown: Editores y Visualizadores.

Se puede utilizar cualquier editor de texto para editar el archivo.

Se debe contar con un visualizador específico para Markdown para poder ver el resultado final.

Como su salida es un HTML, un visualizador de HTML también puede servir.

Markdown: Editores.

Hay editores de texto específicos para Markdown, que permiten ver tanto la entrada como la salida al mismo tiempo.

Hay incluso editores de markdown online, basta contar con conexión a internet.

Vamos a estar utilizando el siguiente:

<http://dillinger.io/>

Pero pueden utilizar también:

Atom, Sublime Text, Visual Studio Code, etc.

(Algunos pueden requerir complementos para visualizar la salida)

Markdown: ¿Por qué lo usamos?.

Para nosotros va a ser de extrema importancia, pues va a ser el formato con el que vamos a enviar mails y documentar nuestros programas (No solo en esta materia sino en toda la carrera).

Es un estándar en la industria, por lo que aprenderlo es fundamental para cualquiera que quiera dedicarse a la programación.

Markdown: Ejemplos de markdown en la industria.

El sitio web **GitHub** que almacena código fuente de muchos programas de **software libre** (*luego veremos de que se trata eso*) utiliza Markdown como lenguaje para documentar los proyectos.

Algunos proyectos en GitHub:

- Apache HTTPD.
- Arara.
- Scala.

Markdown como generador de sitios web.

Markdown está muy ligado a otro lenguaje de marcado, el HTML. De hecho, la mayoría de los visualizadores de Markdown transforman el mismo en un documento HTML, documento que, como veremos en próximas clases, sirve para definir sitios web.

Por ese motivo, muchos visualizadores dan la opción de obtener el archivo HTML a partir de un archivo Markdown. Eso, si bien puede resultar útil en algunos escasos escenarios, no es en absoluto la intención de Markdown.

Es decir, **Markdown no es un lenguaje pensado para armar y describir sitios web**, ni siquiera bosquejos del mismo. Por tanto no debe ser utilizado para ello.

Parte 2. Sintaxis y ejemplos.

Markdown: Párrafos.

- Todo texto escrito, sin marcas aparece tal cual.
- Todo el texto que no tiene saltos de línea conforma un párrafo. Si hay un salto de línea, este aparece en el resultado, pero no se termina el párrafo.
- Si se aplican dos o más saltos de línea consecutivos, entonces se da comienzo a un nuevo párrafo (No importa si hay 20 saltos de línea consecutivos, el resultado no mostrará 20, sino solo el inicio de un nuevo párrafo)

Markdown: Párrafos. Visualización.

Descripción de imagen: Resultado de documento Markdown con párrafos.

Markdown: Títulos.

El texto subrayado con signos de igual (=) es un título, el resaltado con signos de menos (-) es un subtítulo.

Las almohadillas (#) también representan título que van de mayor a menor importancia dependiendo de la cantidad de almohadillas consecutivas (máximo 6)

Markdown: Títulos. Visualización.

Descripción de imagen: Resultado de documento Markdown con títulos.

Markdown: Énfasis.

Colocar texto entre asteriscos (*) o guiones bajos (_) permite dar énfasis a palabras o frases.

```
*Un asterisco da por resultado italica*.
_ Un guion bajo tambien es italica _ .
**Dos asteriscos dan negrita**.
__ Al igual que dos guiones bajos __.
```

Markdown: Énfasis. Visualización.

Descripción de imagen: Resultado de documento Markdown con énfasis.

Markdown: Citas.

En Markdown se puede citar texto utilizando un formato similar al que usan los sistemas correo electrónico, con el carácter de mayor (>).

Markdown: Citas. Visualización.

Descripción de imagen: Resultado de documento Markdown con citas.

Markdown: Listas.

Se pueden generar listas no ordenadas comenzando la línea con asterisco (*), signo de suma (+) o de resta (-) siendo todos equivalentes.
Se pueden generar listas numeradas usando el número para el inicio de la línea seguido de un punto (1.).

Markdown: Listas. Visualización.

Descripción de imagen: Resultado de documento Markdown con listas.

Markdown: Separadores.

Se pueden ingresar líneas de separación horizontales usando al menos tres signos de menos consecutivos (—) o tres asteriscos (***) y luego seguirlo de un salto de línea.

También es posible agregar varios signos de menos o asteriscos más para representar la misma separación.

Markdown: Separadores. Visualización.

Descripción de imagen: Resultado de documento Markdown con separadores.

Markdown: Enlaces.

Se puede agregar enlaces a un sitio web (links) colocando el texto a mostrar en el enlace entre corchetes seguido por la dirección a la cual se va al presionarlo entre paréntesis.

Por ejemplo: *[Ir a Google](http://google.com)*

Markdown: Enlaces. Visualización.

Descripción de imagen: Resultado de documento Markdown con enlaces.

Markdown: Imágenes.

Se pueden insertar imágenes de la misma forma que un link, pero agregando el signo de exclamación antes de los corchetes. En este caso, los corchetes contienen la descripción de la imagen (No deberían quedar vacíos).

Por ejemplo: *![Descripción de la imagen](http://link_a_una_imagen.jpg)*

Markdown: Imágenes. Visualización.

Descripción de imagen: Resultado de documento Markdown con imágenes.

Markdown: Otros elementos.

Hay muchos otros elementos, como citas, inclusión de código de ejemplos, etc. No es el propósito de este curso ni de esta diapositiva volverlos expertos en Markdown.

Si les interesa, siempre se puede encontrar mucha más información en internet. Basta buscar “syntax de markdown” o algo parecido.

Fin de la lección.