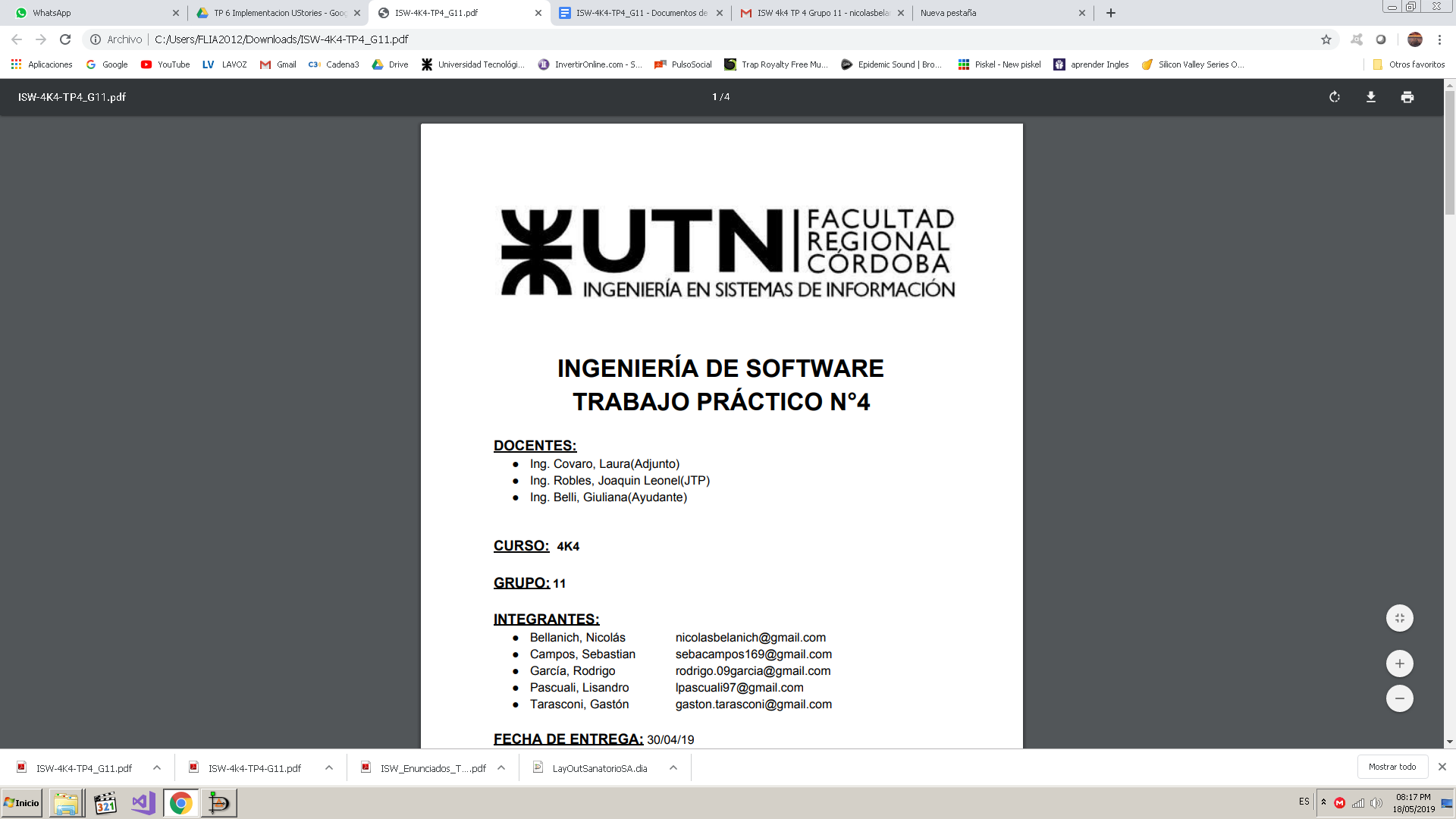
**INGENIERÍA DE SOFTWARE TRABAJO PRÁCTICO N°6** 

**DOCENTES:**

» Ing. Covaro, Laura(Adjunto)

» Ing. Robles, Joaquin Leonel(JTP)

» Ing. Belli, Giuliana(Ayudante)

**CURSO: 4K04**

**GRUPO: 11**

**INTEGRANTES:**

» Bellanich, Nicolás nicolasbelanich@gmail.com

» Campos, Sebastian sebacampos169@gmail.com

» García, Rodrigo rodrigo.09garcia@gmail.com

» Pascuali, Lisandro lpascuali97@gmail.com

» Tarasconi, Gastón gaston.tarasconi@gmail.com

**FECHA DE ENTREGA:** 21/05/19

**1-**  Grupo impar : Realizar pedido de “lo que sea”

**2-** Conjunto de tecnologías:

» HTML y CSS para el esquema de la página web

» Bootstrap

» Bootstrap-notify

» Google-Maps API

» Javascript

» Jquery

» PHP 7 con servidor apache

**3-** Buenas prácticas:

# »Para html [1] :

## DECLARAR EL DOCTYPE CORRECTO

Existen varias versiones de HTML, cada una con sus propias reglas, etiquetas, elementos y atributos. La única manera de que un navegador muestre correctamente una página web es conociendo exactamente la versión de HTML que usa el documento que la contiene.

Para indicarle al navegador la versión de HTML que estamos usando debemos usar la declaración y debe ser la primera línea de nuestro documento.

## RECUERDA CERRAR LAS ETIQUETAS

Un documento HTML está compuesto por muchas etiquetas que sirven para definir encabezados, párrafos e incluso secciones enteras, usualmente vienen en pares, una para abrir y otra para cerrar. Siempre que abras una etiqueta, recuerda cerrarla correctamente.

## NOMBRA LAS ETIQUETAS SIEMPRE EN MINÚSCULAS

Tan importante como cerrarlas, es usar los nombres de las etiquetas siempre en minúsculas.

## UTILIZA FICHEROS EXTERNOS PARA CSS Y JAVASCRIPT

Aunque puedes crear funciones JavaScript y dar estilo a los elementos directamente en el documento HTML, por favor, nunca lo hagas. Utiliza siempre hojas de estilo y scripts externos en archivos separados para estos fines.

## ENLAZA LOS CSS EXTERNOS AL PRINCIPIO

Aunque en teoría puedes enlazar los archivos CSS externos en cualquier parte del documento HTML, lo más recomendable es hacerlo dentro de las etiquetas , en la practica esto hará que las páginas se cargan ligeramente más rápido.

## ENLAZA LOS JAVASCRIPT EXTERNOS AL FINAL

En la medida de lo posible trata de enlazar los archivos JavaScript, justo antes de la etiqueta.

Esto ayudará a que las páginas carguen mucho más rápido.

No es una tontería y existe una explicación para esto: cuando el navegador está cargando un script, se detiene por completo la carga del resto del sitio y continua una vez que finaliza el otro proceso. Si se trata de un script de gran tamaño, el usuario deberá esperar un poco y notará fácilmente que algo sucede en el fondo, algo para nada elegante.

## USA EL ATRIBUTO «ALT» EN TODAS LAS IMÁGENES

Es una regla fácil de olvidar, sin embargo hay que hacer un esfuerzo por recordar incluir el atributo «alt» en todas las imágenes. Aunque parezca una tontería, es necesario incluir este atributo por razones de validación y accesibilidad.

## VALIDA TU CÓDIGO

Validar el código de los documentos varias veces durante el proceso de creación, te ayudará a descubrir errores de manera oportuna y te ahorrará dolores de cabeza innecesarios al final. W3C pone a nuestra disposición su Markup Validation Service que nos permitirá realizar esta tarea de manera sencilla y nos avisará sobre cualquier error en nuestro código.

## MEMORIZA TODAS LAS ETIQUETAS

Resulta muy útil repasar la lista completa de etiquetas HTML disponibles y tratar de memorizar la mayor cantidad de ellas. Siempre puedes acceder a la referencia, pero mantenerlas en la cabeza te ayudará a escribir código mucho más rápido.

Si memorizar no es lo tuyo, te recomiendo buscar un editor con soporte para HTML (hay cientos) que te ayude a agilizar el proceso, autocompletando código y ofreciendo sugerencias.

## LEE CÓDIGO DE OTROS

Para cerrar, aunque no esté ligado únicamente al código HTML (ya que aplica para JavaScript, CSS y cualquier otro lenguaje) al leer código de otros desarrolladores, incluso mirando el código fuente de tus sitios web favoritos puedes aprender mucho. Inspecciona ese elemento particular que llama tu atención y aprende como lo hicieron. Revisa otros estilos y crea uno propio.

# »Para CSS [2] :

## 

## RESETEA LAS PROPIEDADES DEL CSS

Antes de empezar a maquetar te recomendamos que resetees las propiedades de tu código css como paddings, margins, borders, las diferentes etiquetas etc. Esto puedes hacerlo de forma manual, o con alguna de las herramientas existentes, las más populares son: Normalize CSS, HTML5 Doctor Reset CSS, Yahoo! (YUI 3) Reset CSS, Universal Selector \* Reset.

## USA UNA SOLA HOJA DE ESTILOS

De este modo, además de concentrar toda la info en un mismo doc evitando repetir declaraciones de css, nos ayuda a optimizar el tiempo de carga, ya que tener una hoja de estilos por cada página sería cargar la misma información muchas veces innecesariamente.

## NO USES ESTILOS EN LÍNEA

Sobre todo por motivos relacionados con el SEO, Google penaliza estas prácticas. Además, dificulta el flujo de trabajo y hace repetir las mismas declaraciones en distintas partes del html para elementos cuyo estilo es común.

## SIGUE UNA JERARQUÍA

Maqueta siempre con el mismo orden los atributos de las clases. Puedes seguir un orden alfabético o separarlos por tipos de selectores, por aspectos de colocación y diseño, etc.

Como las posibilidades son múltiples, tienes la opción de utilizar el orden con el que te sientas más cómodo.

## USA CLASES QUE TENGAN RELACIÓN SEMÁNTICA CON LO QUE SE ESTÁ MAQUETANDO

No siempre vas a estar tu solo maquetando un proyecto o quizás tengas que retomarlo con el paso del tiempo, por eso es importante que las clases tengan un sentido con lo que se está maquetando, para facilitar la compresión del código. Si usas clases con nombres muy rebuscados o sin sentido, cuando se retome el proyecto o alguien nuevo entre en el proyecto no entenderá el usos de las clases y cómo se aplican de forma rápida, si no que tendrá que repasar todo el proyecto para entender la estructura.

## USA GUIÓN MEDIO EN LUGAR DE GUIÓN BAJO

Mejora la compatibilidad con todos los navegadores y Google lo interpreta mejor. Además, los editores de texto lo interpretan como una sola palabra.

## NO REPITAS CÓDIGO

Si dos elementos diferentes tienen el mismo estilo, no los declares dos veces, usa una sola declaración para ambos elementos separándolos mediante comas.

*ejemplo:*

*h1 {color: white;}*

*h2 {color: white;}*

*h1, h2 {color: white;}*

## USA ESTILOS EN CASCADA SIEMPRE QUE SEA POSIBLE, PARA REDUCIR EL NÚMERO DE CLASES E ID USADOS

Esto ayudar a reducir el número de instancias innecesarias, reduciendo el peso del proyecto. Además, es una buena herramienta de trabajo para ubicar un elemento dentro de la jerarquía html.

## USA SIEMPRE QUE SE PUEDA CLASES EN VEZ DE ID

Los Id son sobre todo utilizados por los desarrolladores, para implementarlos en otros lenguajes de programación como Javascript o Angular. Además, los ID solo se pueden utilizar en un elemento, por lo que si más adelante queremos reciclar esas declaraciones no podremos y deberemos duplicar una de las instancias, aumentando el número de líneas que tenemos en nuestro código.

## NO USES IMPORTANT

Nunca uses important, si necesitas usarlo en algún momento significa que algo no estas haciendo bien, revisa tu código e intenta subsanar el error.

# »Para PHP [3] :

## Commenting & Documentation

Documentar métodos nuevos de clases y funciones

## Consistent Indentation

Mantener un consistente indentación

## Avoid Obvious Comments

Evitar comentarios obvios

## Code Grouping

Agrupar código de acuerdo a funcionalidad

## Consistent Naming Scheme

Mantener consistencia de nombrado, para nuestro caso usamos el guión bajo para separar palabras de funciones o métodos, y camelCase para nombre de archivos y clases.

## Avoid Deep Nesting

Evitar muchos niveles de condicionales

## Limit Line Length

Una línea de código no puede superar los 120 caracteres con tabs incluidos

## File and Folder Organization

Separamos el backend del frontend. Y además utilizamos la estructura que elegimos en gestión de la configuración

[1] Información de www.hipertextual.com : <https://hipertextual.com/archivo/2012/08/buenas-practicas-html/>

[2] <https://www.gobalo.es/blog/diseno-web/buenas-practicas-de-css/>

[3]

<https://code.tutsplus.com/tutorials/top-15-best-practices-for-writing-super-readable-code--net-8118>