



# GUÍA DE TRABAJOS PRÁCTICOS 2024

## LABORATORIO DE COMPUTACIÓN II

AGUSTINA EDMÉ ALICIARDI

**ucc** | FACULTAD  
DE INGENIERÍA

# Trabajos Prácticos

---

## Índice

TRABAJO PRÁCTICO Nº 1 Características básicas de los SO	4
TRABAJO PRÁCTICO Nº 2 Drivers	8
TRABAJO PRÁCTICO Nº 3 Herramientas de Desarrollo	10
TRABAJO PRÁCTICO Nº 4 Prototipado	14
TRABAJO PRÁCTICO Nº 5 Aplicaciones Web	19
TRABAJO PRÁCTICO Nº 6 Orientación a Eventos	29
TRABAJO PRÁCTICO Nº 7 Aplicaciones Móviles	33

## ENCUADRAMIENTO DE LA FORMACIÓN PRÁCTICA

La formación práctica que se desarrolla en esta GTP incluye la resolución de problemas tipo o rutinarios de problemas de Ingeniería.

### 2. CONSIGNAS GENERALES

Los problemas propuestos en el presente TP deben ser resueltos en forma individual, y el código debe ser subido a github de manera pública.

De acuerdo a la complejidad creciente que esta asignatura presenta, el planteo o formulación de los problemas debe ser realizado basándose en lo desarrollado en las clases teóricas referidas a la unidad correspondiente y a todas las unidades anteriores ya desarrolladas, como también de aquellas asignaturas que son correlativas de esta.

Es fundamental para un correcto aprendizaje que el alumno controle sus soluciones con la herramienta de desarrollo elegida ([WebStorm](#), [VisualStudio](#), etc).

### 3. PRESENTACIÓN DE LOS TRABAJOS PRÁCTICOS

#### 3.1 Ejercicios de clase

Todos los ejercicios realizados en clase emplearán código base disponible en la plataforma de github: <https://github.com/UCC-LabCompu2/ejercicios2024>

#### 3.2 Proyecto

De manera evolutiva, es decir, clase a clase se irá realizando un proyecto con tecnologías web: HTML, CSS y JavaScript. El mismo debe ser presentado en la plataforma de GitHub [https://github.com/UCC-LabCompu2/proyecto2024-\[Alumno1\]-\[Alumno2\]](https://github.com/UCC-LabCompu2/proyecto2024-[Alumno1]-[Alumno2])

El trabajo debe ser realizado de manera individual, ó en grupo de hasta 2 personas. El primer parcial consistirá en código html y css (se evaluará igualmente el buen uso de git).

El segundo parcial consistirá en código js, además de las correcciones y observaciones realizadas durante el primer parcial.

# UNIDAD I: Instalación y configuración de Sistemas Operativos

En el área de la informática no sólo es necesario poseer conocimientos de programación, sino también conocer las diversas herramientas que posibilitan el desarrollo de esta tarea. Es por esto que la instalación y configuración de sistemas operativos es de suma importancia: sustenta el resto de las actividades.

## TRABAJO PRÁCTICO Nº 1 Características básicas de los SO

### Objetivos de Aprendizaje

- Conocer las características básicas de diferentes sistemas operativos (Windows y Linux).
- Adquirir habilidad en la instalación y configuración de Windows.

### Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad Nº 1 de la programación de la asignatura.

### Consignas a desarrollar en el trabajo práctico



**Sugerencia:** muchos libros están disponibles digitalmente a través de la plataforma e-libro. Consulta en biblioteca.



El teórico puede ser consultado en el libro [“Instalación y configuración de sistemas operativos”](#) de Enrique Bellido Quintero, Unidad 1, Tema 5:



### Preguntas sobre Sistemas Operativos:

- 1) ¿Qué es un SO (Sistema Operativo)? De ejemplos de SO para diferentes dispositivos.
- 2) ¿Qué tareas realiza un sistema operativo?
- 3) ¿Qué criterios o características definen un SO?
- 4) Complete la siguiente tabla acorde a las características de los SO:

SO	Año	Tipo de Interfaz		Tareas		Arquitectura		Licencia	
		GUI	CLI	Mono	Multi	Cliente	Servidor	Libre	Propietario
Windows 3.11									
Windows NT									
Windows 95									
Windows 10									
Windows 11									
Ubuntu 20									
Ubuntu Server									
Red Hat									
Mac OS X									

- 5) ¿Qué recursos de hardware emplean los sistemas operativos?
- 6) ¿Qué diferencia existe entre los requisitos mínimos y los recomendados de hardware especificados por los SO?
- 7) Complete la siguiente tabla comparativa con requisitos mínimos de diferentes versiones de Windows. ¿Observa algún patrón?

Requisitos	3.11	NT	95	98	XP	7	10	11
Año								
Procesador								
Memoria Ram								
Disco Duro								
Tarjeta Gráfica								

- 8) ¿Cuáles son las ventajas y desventajas de mantener los SO actualizados?
- 9) ¿Qué es una máquina virtual? ¿Para qué sirve? ¿Qué características y ventajas posee? De ejemplos de aplicaciones o programas para crear máquinas virtuales.
- 10) ¿Qué es docker? ¿Qué diferencia tienen los contenedores y las máquinas virtuales?
- 11) ¿Qué es la BIOS? ¿Qué función cumple?
- 12) ¿Qué es la UEFI? ¿Qué características tiene?
- 13) ¿Qué sucede al formatear un disco duro? ¿Qué es el formato rápido?
- 14) ¿Qué es una partición? ¿Qué ventajas tiene particionar un disco duro? ¿Qué son las particiones primarias y secundarias?

- 15) ¿Qué es Rufus? ¿Para qué sirve? ¿Qué opciones presenta?
- 16) ¿Cuáles son los pasos de instalación de un sistema operativo?



### Ejercicios en Windows:

- 1) Acceda a la configuración de la BIOS. ¿Es legacy o UEFI?. ¿Qué opciones presenta?
- 2) Saque fotos de las diversas pantallas de la BIOS y adjúntelas. ¿El diseño de las pantallas, de qué depende?
- 3) Entre a la sección del “BOOT Loader” dentro de la BIOS y ordene adecuadamente los dispositivos de arranque del sistema.
- 4) Configure una máquina virtual para realizar una instalación de Windows 10. Detalle la configuración realizada.

Procesador:

Memoria Ram:

Disco Duro:

Tarjeta Gráfica:

- 5) Instale Windows 10. Describe muy brevemente los pasos que realizó para la instalación del SO.
- 6) Cree un pendrive autobootable empleando Rufus. ¿Cuáles son los pasos para crearlo? ¿Qué opciones ofrece el programa?



### Para aprender más...

- 1) Busque al menos tres sistemas operativos aparte de MacOS, Windows y Linux y realice una breve descripción de ellos.
- 2) Clasifique los sistemas operativos que conoce según:
  - a) Número de tareas: multitarea o monotarea. Distinga si es del tipo: conmutación de contexto, multitarea cooperativa ó de tiempo compartido.
  - b) Administración de recursos: monousuario, multiusuario.
  - c) Forma en que se ofrecen y emplean los recursos: Centralizado, Distribuido.
  - d) Número de procesadores: multiproceso, monoproceso.
- 3) ¿Qué sistemas de archivos pueden utilizarse en windows? ¿Qué sistema de archivos pueden utilizarse en linux? Describa las características de cada uno.
- 4) ¿Qué es el formateo a bajo nivel? ¿Qué diferencia tiene con un formateo común?
- 5) ¿Qué es GRUB y LILO?


**Ejercicios sobre Linux:**

- 1) Complete la siguiente tabla comparativa con requisitos mínimos de diferentes versiones de Linux. ¿Observa algún patrón?

Requisitos	Ubuntu 11.10	Debian 6	Fedora 33	CentOS	RedHat 3.0.3
<b>Año</b>					
<b>Procesador</b>					
<b>Memoria Ram</b>					
<b>Disco Duro</b>					
<b>Tarjeta Gráfica</b>					

- 2) Configure una máquina virtual para realizar una instalación de Linux. Detalle la configuración realizada.

Procesador:

Memoria Ram:

Disco Duro:

Tarjeta Gráfica:

- 3) Instale Linux. Describa muy brevemente los pasos que realizó para la instalación del SO.


**Para aprender más...**

- 1) ¿Qué es una distribución de linux? nombre tres diferentes
- 2) ¿Qué es UNIX? ¿Qué es GNU?
- 3) ¿Qué similitudes y diferencias hay entre UNIX y Linux? ¿Qué relación hay entre MacOS y UNIX?
- 4) ¿Qué es el Kernel de un SO?

# Drivers

Seguro que has leído esa palabra un montón de veces. Pero ¿qué son? Sin drivers no podrías jugar, imprimir documentos o reproducir música. “Pero eso lo hace el hardware”. Sí, pero el hardware no se controla solo: necesita órdenes. Esas órdenes las da el sistema operativo, que las envía en un idioma abstracto, que los dispositivos no entienden. Para que cada aparato sepa qué se le pide, hacen falta programas que traducen cada comando del SO. Esos programas son los drivers.

## TRABAJO PRÁCTICO Nº 2

### Drivers

#### Objetivos de Aprendizaje

- Adquirir conocimientos sobre Drivers.

#### Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad Nº 1 de la programación de la asignatura.

#### Consignas a desarrollar en el trabajo práctico



El teórico puede ser consultado en el libro “Instalación y configuración de sistemas operativos” de Enrique Bellido Quintero, Unidad 2, Tema 4



#### Preguntas sobre Drivers:

- 1) ¿Qué es un driver o controlador de dispositivo? ¿Para qué sirve?
- 2) ¿Qué dispositivos necesitan controladores?
- 3) ¿Qué tipos de drivers existen?
- 4) ¿Qué diferencia hay entre los drivers genéricos o universales y los drivers específicos?
- 5) ¿Por qué es recomendable mantener los drivers actualizados?
- 6) ¿Cómo obtener los drivers?



#### Preguntas sobre Drivers en Windows:

- 1) ¿Qué es el administrador de dispositivos?
- 2) ¿Qué puede hacer el usuario a través del Administrador de Dispositivos?
- 3) ¿Cómo se distribuyen los Drivers en este sistema operativo?



**Preguntas sobre Drivers en Linux:**

- 1) ¿Qué muestra la 'Información del sistema'? ¿Cómo veo los recursos de hardware del equipo?
- 2) ¿Cómo se llama a los drivers en linux?

**Ejercicios sobre Windows:**

- 1) En Windows entre a la opción "Panel de Control" >> "Administrador de Dispositivos" o desde "Equipo" >> "Administrar" >> "Administrador de Dispositivos" y verifique que drivers tiene instalados y cuales le faltan.

**Ejercicios sobre Linux:**

- 1) En Linux verifique que todos los drivers estén instalados.



Puedes leer más en el libro "Desarrollo de componentes software para el manejo de dispositivos: UF1287" de María Josefa Díaz Coca, Capítulo 2.

- 1) ¿Cuáles son las 3 interfaces que proporciona un controlador de dispositivo?
- 2) Realice un listado de dispositivos y clasifíquelos según:
  - a) Forma en que transfieren la información a los dispositivos.
  - b) Acceso secuencial o aleatorio.
  - c) Transferencia de forma síncrona o asíncrona.
  - d) Dispositivo compartido o dedicado.
  - e) Velocidad del dispositivo.
  - f) Forma de acceso: lectura/escritura, solo lectura, solo escritura.
- 3) ¿Qué es el spooling?

## UNIDAD II: Herramientas de desarrollo

Existen diversas herramientas que nos ayudan en la tarea de desarrollar software y hacen a las buenas prácticas del programador.

La disponibilidad de herramientas para desarrolladores es muy amplia. Hay unas más completas y complejas que otras, todo dependiendo de las necesidades individuales o de conjunto de una organización. Es en ese sentido han surgido varias herramientas para desarrollo en equipos con el fin de organizar y facilitar un poco su flujo de trabajo.

### TRABAJO PRÁCTICO Nº 3 Herramientas de Desarrollo

#### Objetivos de Aprendizaje

- Conocer diferentes entornos de desarrollo
- Adquirir experiencia en la utilización de sistemas de gestión de versiones

#### Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad Nº 2 de la programación de la asignatura.

#### Consignas a desarrollar en el trabajo práctico



##### Preguntas

- 1) ¿Qué son las herramientas de desarrollo? ¿Qué tipo de herramientas existen?
- 1) ¿Qué significa IDE? De ejemplos de IDE's para desarrollo web ampliamente usados.
- 2) ¿Qué herramientas suele integrar un entorno de programación? Describa cada una.
- 3) ¿Qué es un sistema de gestión de versiones o VCS? ¿Qué ventajas tiene?
- 4) ¿Qué es Git? ¿Qué servidores se pueden utilizar para ello?
- 5) ¿Por qué es importante la documentación de Código?
- 6) ¿Qué es JsDoc? ¿Qué otras herramientas similares existen?
- 7) ¿Cuáles son las líneas de comando de GIT para:
  - a) Clonar un repositorio vs. Inicializar un repositorio
  - b) Actualizar tu repositorio local al commit más nuevo
  - c) Registrar cambios (añadirlos al Index)
  - d) Hacer un commit
  - e) Enviar cambios a repositorio remoto

8) Escriba los comandos de git correspondientes en cada flecha



9) ¿Qué función poseen los Bug Trackers? De ejemplos de los más empleados.

10) ¿Qué son las pruebas de desempeño? ¿Qué tipo de pruebas se pueden realizar y para qué sirve cada una?

11) ¿Qué es la integración continua? ¿Qué herramientas se pueden emplear para ello?



## Ejercicios

1) Configurar usuario de git por terminal de manera global.

2) Desde la consola de comandos, resuelva el siguiente ejercicio:

\*Recuerde instalar git en su computadora. Puede descargarlo de <https://git-scm.com/>

Proyecto Local:

1	Inicializar un repositorio	git ____
	Crear un archivo y guardarlo*	(manualmente)
2	Mostrar situación actual del repositorio	git _____
3	Registrar los cambios (añadir el archivo)	git _____
4	Incluir el archivo en el repositorio local	git _____
5	Mostrar los commits realizados	git ____
	Modificar nuevamente el archivo	(manualmente)
7	Registrar los cambios e incluirlos	git _____ "____"
8	Mostrar los commits realizados	git _____

\*El archivo creado debe ser una página web (archivo HTML), y en el <head> incluir el título, autor, descripción de la página y palabras principales.

- 3) Cree una cuenta de Github, entrando a <https://github.com/> (si ya tienes una cuenta en github, este paso no es necesario)
- 4) Empleando Github:
  - a) Crear un repositorio compartido sin inicializarlo
  - b) Clonar el repositorio
  - c) Agregar un archivo y comitearlo de manera local
  - d) Guardar el commit en el repositorio remoto
  - e) En el navegador, visualizar los commits realizados
- 5) Empleando GitHub (si utiliza comandos de git, especifique cuales son):
  - a) Crear un repositorio e Inicializarlo. Agregar una descripción.
  - b) Configurar el repositorio para trabajar de manera colaborativa con un compañero. (Setting->Collaborators).
  - c) Clonar el repositorio.
  - d) Crear un archivo dentro del repositorio.
  - e) Verificar el estado del repositorio. De ser necesario, agregar el archivo y guardar los cambios de forma local.
  - f) Guardar los archivos en el repositorio remoto (GitHub).
  - g) Si el compañero ha realizado un cambio (si está trabajando de manera individual, realizar un cambio desde el navegador, y realizar un commit desde el mismo), actualizar el repositorio local.
- 6) Escribir un Readme.md con la siguiente información:
  - a) Títulos secundarios
  - b) Autor del repositorio
  - c) Contenido/Descripción
  - d) Emplear negrita como estilo
  - e) Agregar link del aula virtual
  - f) Agregar info de contacto en una tabla
  - g) Agregar emoji
- 7) Empleando el repositorio del ejercicio 3:
  - a) Cree una nueva rama
  - b) Vea el listado de todas las **branches** existentes en el proyecto y la branch actual
  - c) Cambie de rama

- d) Realice algún cambio en el código y push al repositorio remoto
  - e) Visualice en Github los commits de las **branches**
  - f) Merge de la nueva rama con Master
  - g) Borrar una **branch**
- 8) Empleando el repositorio del ejercicio 1:
- a) Volver a un commit anterior (sin borrar el posterior o último)
  - b) Volver al último commit
  - c) Volver a un commit anterior (borrando el último, usando checkout)
  - d) Descartar el último commit (usado reset hard)
- 9) Empleando el repositorio del ejercicio 3:
- a) Eliminar un archivo
  - b) Eliminar una carpeta
  - c) Guardar los cambios en repositorio remoto.
  - d) Verificar que en el remoto, los archivos no estén presentes.
- 10) Desde GitHub realizar un Fork de las filminas usadas en clases.  
<https://github.com/UCC-LabCompu2/filminas.git>
- 11) Configure WebStorm para clonar el repositorio al que se le realizó el Fork.
- 12) Crear un .gitignore
- a) Incluir el nombre de carpeta/archivo que se desea ignorar.
  - b) Subir cambios al repositorio remoto



**Recomendación:** Puedes utilizar alguna plataforma como Test Automation University para hacer el curso de [Source Control](#) (gratuito), DevTalles para [Git+GitHub](#). También puedes buscar cursos en otras plataformas como Coder House, Udemy, Codecademy, Digital House, Code School o Platzi para aprender y practicar sobre Git.



### Para aprender más...

- 1) De los diferentes IDEs nombrados en clase, detalle sus principales características. ¿Qué ventajas y desventajas presentan? ¿Cuál se adapta mejor a tu forma de trabajo?
- 2) ¿Por qué no es recomendable usar IDE's Wysiwyg?
- 3) Nombre diferentes tipos de herramientas y cuál es su utilidad. Ejemplifique.
- 4) ¿Cuáles son las principales características de Webstorm?

## UNIDAD III: Interfaces de usuario

Un diseño cuidadoso de la interfaz de usuario es parte fundamental del proceso de diseño general del software. Si un sistema de software debe alcanzar su máximo potencial, es fundamental que su interfaz de usuario sea diseñada para ajustarse a las habilidades, experiencia y expectativas de los usuarios previstos. Un buen diseño de interfaz es crítico para la confiabilidad del sistema.

### TRABAJO PRÁCTICO N° 4

#### Prototipado

#### Objetivos de Aprendizaje

- Aprender a desarrollar el maquetado de una interfaz de usuario
- Adquirir habilidad en el diseño de interfaces de usuarios

#### Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad N°3 de la programación de la asignatura.

#### Consignas a desarrollar en el trabajo práctico



Teniendo en cuenta la unidad teórica N°16 del libro “Ingeniería del software” séptima edición de Ian Sommerville y el diseño del sistema que se realizó en el TPN°4, responda las siguientes preguntas:



#### Preguntas

- 1) ¿Qué es UI y UX? ¿Qué diferencias tienen? ¿Quién lo diseña? ¿Por qué es importante?
- 2) ¿Cuáles son los pasos del proceso de diseño de UI? Describa brevemente cada uno.
- 3) ¿Qué factores se deben considerar para realizar el diseño de una UI?
- 4) ¿Cuáles son los principios de diseño? Explique cada uno.
- 5) ¿Cuáles son los estilos de interacción? ¿Qué ventajas y desventajas tienen? De ejemplos de aplicaciones.
- 6) ¿Qué es el modelo de Seeheim?
- 7) ¿Cuáles son las pautas de Shneiderman para la utilización efectiva del color?
- 8) ¿Qué factores se deben tener en cuenta para presentar mensajes al usuario? Explique cada uno.
- 9) ¿Qué son los mensajes de error orientados al usuario? ¿y los orientados al sistema?

- 10) ¿Qué es la usabilidad? Explique los atributos de usabilidad.
- 11) ¿Cuáles son las técnicas para realizar un análisis de los usuarios? Explique cada una.
- 12) ¿Qué diferencia existe entre los diagramas de organización, funcionamiento y presentación?
- 13) ¿Qué diferencia existe entre un Sketch, un wireframe, un mockup y un prototipo? Detalle características.
- 14) ¿Qué es Thumbnail Sketches? ¿Qué es StoryBoard?
- 15) ¿Cuáles son las ventajas de emplear técnicas de diseño?
- 16) ¿Qué es la evaluación de interfaz?

### Ejercicios



Sistema: Piense en algún tema de su interés para desarrollar una página web. Piensa que la misma debe incluir: utilización de campos de formulario, realización de cálculos, funciones javascript, renderizado mediante la utilización de canvas. Si no sabes que tema elegir, puedes pensar en algún tema de física, matemática, etc.

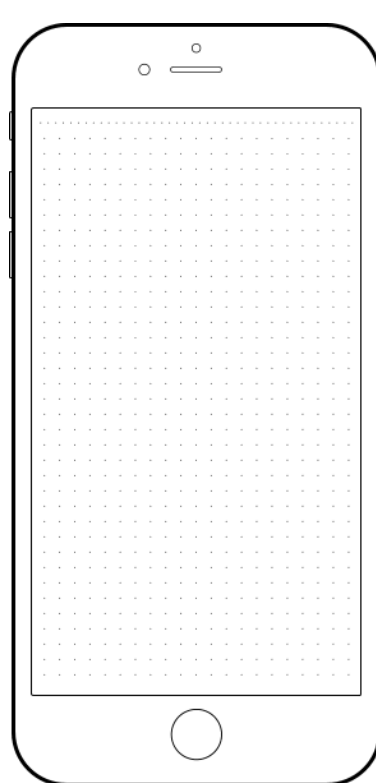
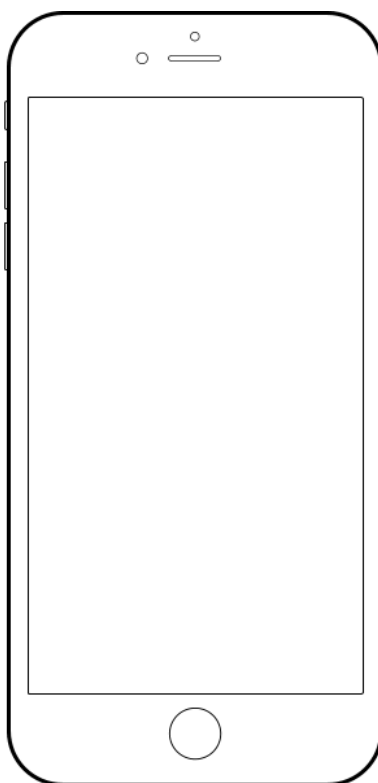
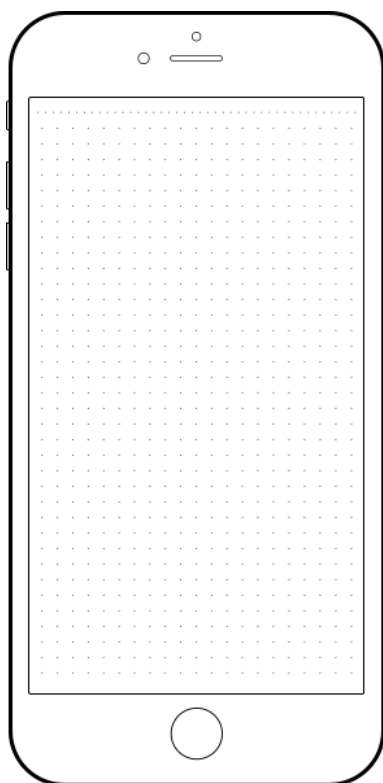
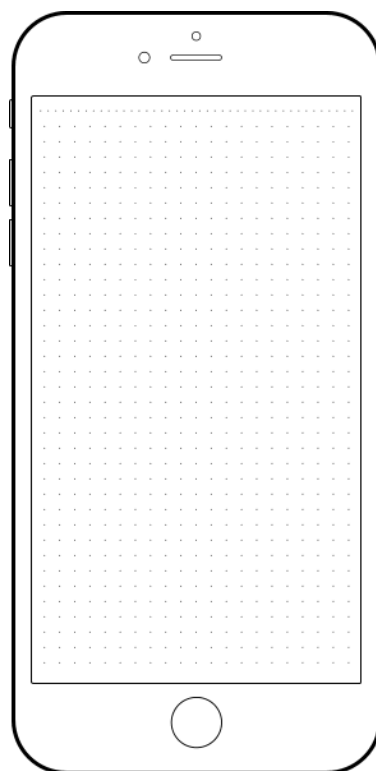
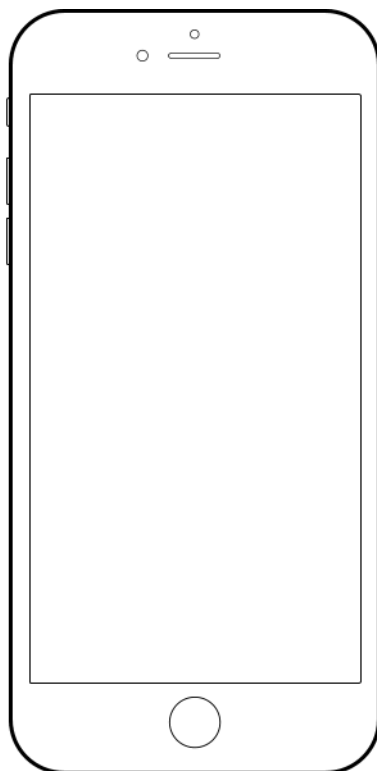
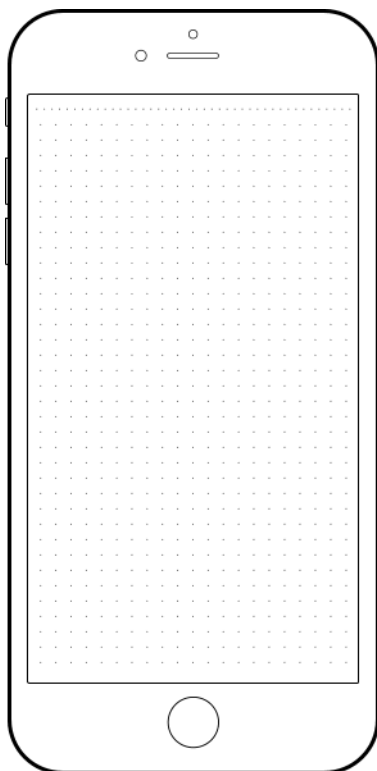
En grupos de hasta 2 integrantes:

- 1) Realice sketches de la UI del sistema para generar un 'Thumbnail Sketchs' empleando el template para celulares de la hoja siguiente. Debe tener en cuenta ¿Qué opciones debe ofrecer al usuario? ¿Qué campos existen para ingresar datos? ¿Hay ejercicios de ejemplo? ¿Hay una introducción teórica? ¿Qué acciones le permiten al usuario realizar? ¿Qué mensajes de error mostrará al usuario? Recuerde realizar los pasos del "Proceso de diseño de Interfaz de usuario". Si conoce sistemas similares puede utilizarlos de ejemplo para plantear su interfaz.

(Se recomienda que en la misma página donde se piden ingresar los datos, muestre los resultados y el gráfico, y permita la conversión de unidades; pero se pueden diseñar pantallas adicionales con ejemplos, ejercicios o contenido teórico)

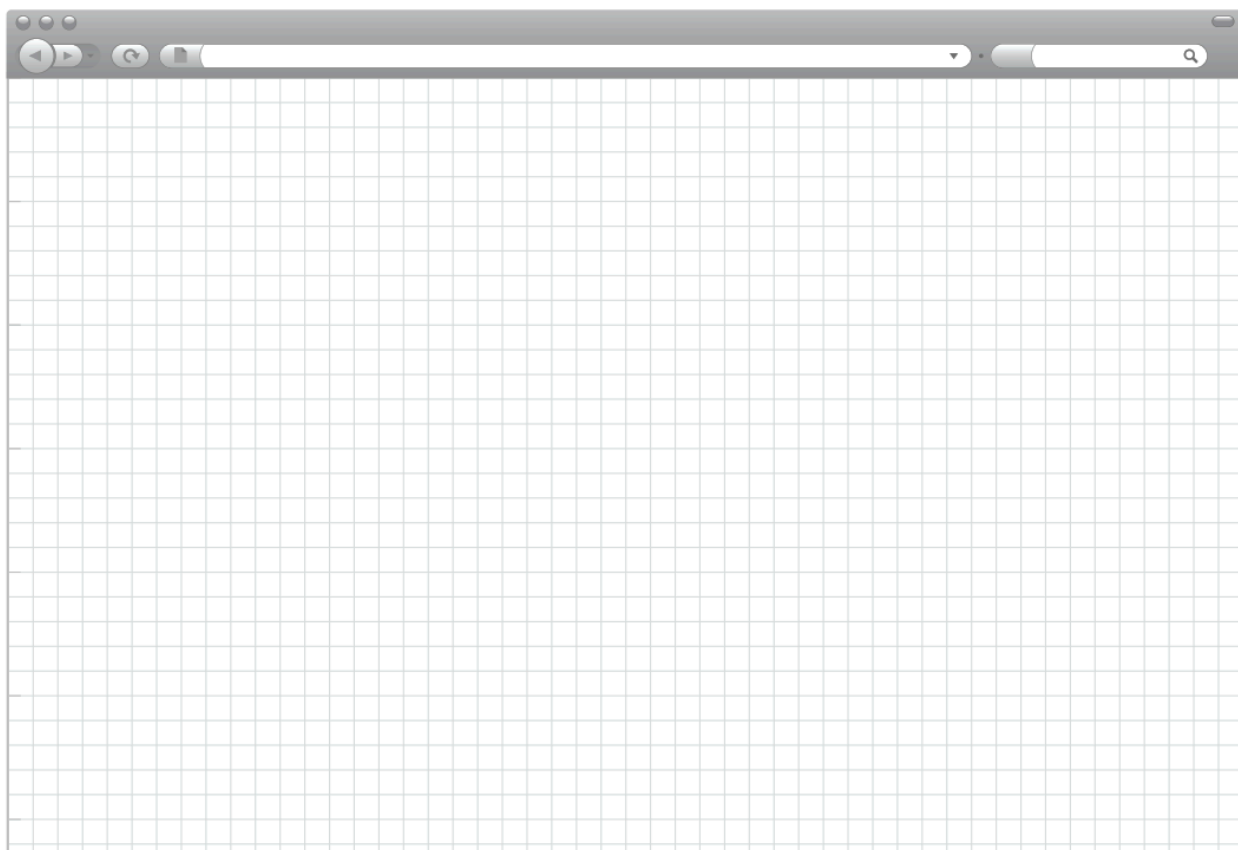
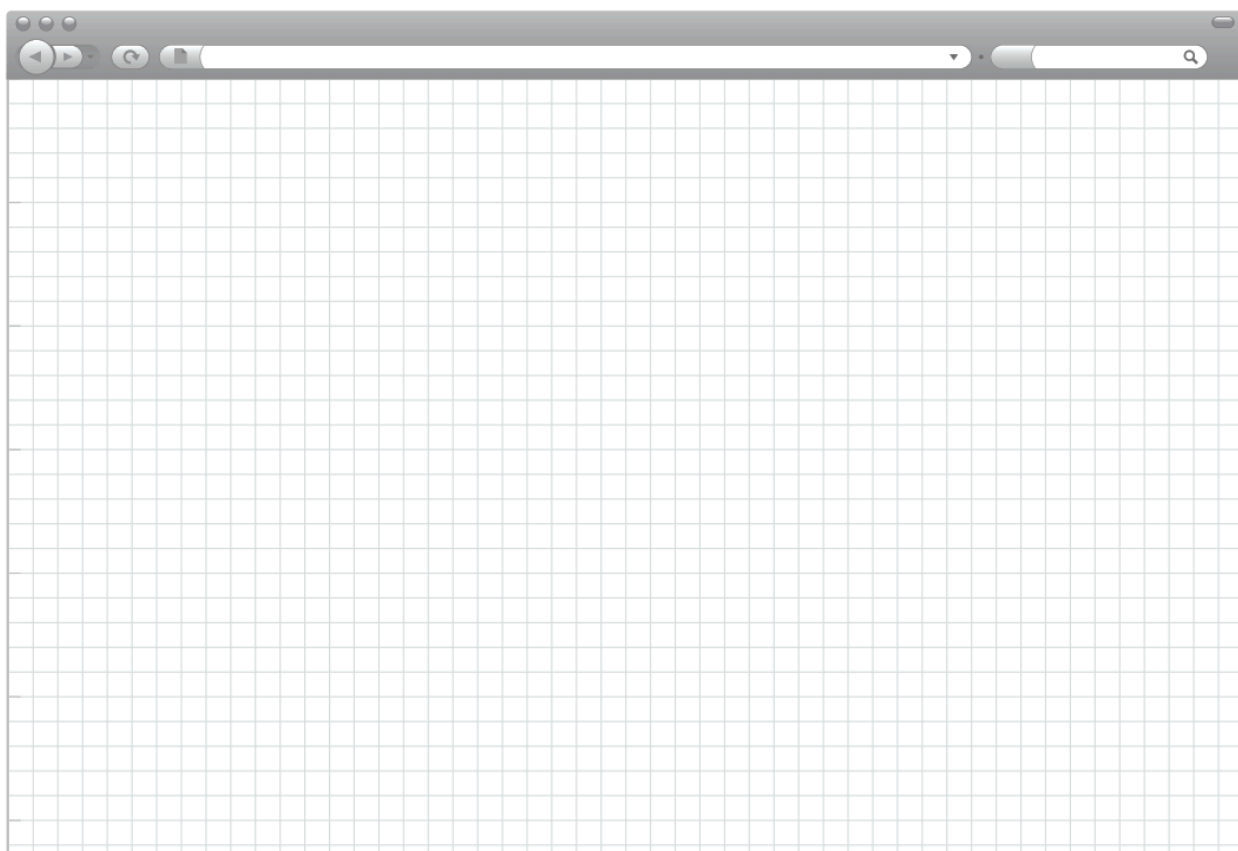
- 2) Adapte el diseño previamente realizado a la web empleando el template de la hoja siguiente. Debe tener en cuenta que los contenidos que en el celular se muestran uno bajo el otro, pueden mostrarse en columnas, o en otras proporciones.
- 3) Intercambie su diseño de interfaz de sistema con otro grupo, y considere: ¿Qué opciones tiene este sistema que no estén disponibles en el propio? ¿Es deseable introducir en un futuro rediseño del sistema estas características? ¿Qué ventajas y desventajas presenta el otro sistema con respecto al propio? Presente un breve documento al otro equipo con el análisis realizado. (Puede emplear de ejemplo la tabla existente posterior a los templete).
- 4) Realice un rediseño de su sistema teniendo en cuenta el documento de sugerencias recibido.
- 5) Emplee algún programa para confeccionar un Mockup. (Ej. [Figma](#), [AdobeXD](#), [Draw.io](#), [Pencil project](#), Mockups, NinjaMoc, o similares).

**Nombre y Apellido:**





**Nombre y Apellido:**



Emplee la siguiente tabla para realizar un análisis de su diseño:

Tema	
Alumnos	
Menú principal y navegación	
Contenido	
Funcionalidad	
Familiaridad	
Otras Sugerencias o Comentarios	



### Para aprender más...

- 1) ¿Cuáles son las reglas de oro que determinó Theo Mandel para el diseño de interfaz de Usuario?
- 2) ¿En qué consiste el HTA? ¿Qué ventajas y desventajas tiene esta técnica? ¿Cómo se obtiene la información?
- 3) ¿Cómo se realiza la evaluación de la interfaz en ámbitos profesionales?

## UNIDAD IV: Aplicaciones Web

La red de redes no sería lo que es hoy sin la ayuda de HTML. HTML5 es un conjunto de lenguajes que posibilitan tener aplicaciones Web con grandes funcionalidades en nuestros navegadores.

La nueva versión de 5 de HTML nos da la libertad como programadores de realizar aplicaciones completas, juegos, animaciones, o simples artículos de Blog. En la actualidad, HTML5 no se limita solamente a navegadores web, sino que se pueden realizar aplicaciones móviles y de escritorio sin necesidad de estar conectados.

### TRABAJO PRÁCTICO Nº 5

#### Aplicaciones Web

#### Objetivos de Aprendizaje

- Adquirir conceptos básicos sobre aplicaciones web.
- Adquirir habilidad en el desarrollo web utilizando HTML y CSS.

#### Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad Nº 4 de la programación de la asignatura.

#### Consignas a desarrollar en el trabajo práctico



Teniendo en cuenta la unidad teórica Nº1 y Nº2 del libro “Aplicaciones Web” de Javier Zofio Jiménez, responda las siguientes preguntas:



#### Preguntas

- 1) ¿Qué es una aplicación web? De ejemplos de aplicaciones web.
- 2) ¿Qué ventajas tienen las aplicaciones web? ¿Cómo se clasifican según el tipo de acceso?
- 3) ¿Cómo es la estructura típica de una aplicación web? ¿Cuántas capas posee y que función cumple cada capa?
- 4) ¿Qué diferencias existen entre las aplicaciones web y las páginas web?
- 5) ¿Qué son las páginas web estáticas y dinámicas? ¿Cómo es el procesamiento de datos en las webs estáticas y en las webs dinámicas?
- 6) ¿Qué es un servidor de aplicaciones?
- 7) ¿Qué es la web 1.0, 2.0, 3.0 y 4.0?
- 8) ¿Qué lenguajes se utilizan generalmente para la programación web?

9) ¿Qué significan las siglas HTML? ¿Qué es HTML5? ¿Qué características tiene? Explique cada una.

10) Complete las siguientes tablas:

Versión HTML	Año	Características
1.0		
2.0		
3.0		
3.2		
4		
5		
6?		

Versión CSS	Año	Características
1		
2		
3		

11) ¿Qué son las etiquetas semánticas? Explica cada una de ellas.

12) ¿Qué es la accesibilidad web? ¿Cuáles son las buenas prácticas a tener en cuenta para la accesibilidad?

13) ¿Qué es una hoja de estilo o CSS?

14) ¿Cómo se pueden agregar estilos en las páginas web? ¿Qué ventajas y desventajas tiene cada método?

15) ¿Qué son los descendientes en HTML?

16) ¿Qué son las pseudo-clases?

17) ¿Qué es el modelo de caja?

18) ¿Qué son y para qué sirven los preprocesadores de CSS? De ejemplos de los más empleados.

19) ¿Qué buenas prácticas se deben seguir a la hora de diseñar una página web?

20) Haga un listado de las etiquetas de HTML que ha empleado en los diversos ejercicios, y explique los atributos y la funcionalidad.



## Ejercicios de HTML



**Recomendación:** Puedes hacer commit en cada etapa de los ejercicios para tener una historia del desarrollo.

- 1) Empleando el template “ej\_noticia”, incluir en la página 3 secciones mediante etiquetas semánticas:
  - a) En el <header> incluir una imagen y un título (la imagen debe ser descargada de internet e incluida en la carpeta ‘Ejercicios-HTML/imagenes’)
  - b) En <section>, separar el texto en párrafos. Debe incluir la utilización de estilos (negrita, itálica y subrayado).
  - c) El <footer> debe contener el autor de la página.
- 2) Empleando el template “ej\_noticia”, incluir diferentes tipos de título:
  - a) **h1** para ‘Diario On-line’
  - b) **h3** para ‘Cordobici’
  - c) **h6** para la fecha de publicación
  - d) **h5** para el sub-título de la noticia
- 3) Empleando el template “ej\_listas” donde hay una receta, emplear una lista no ordenada para los ingredientes y una lista ordenada para los pasos de la preparación. Agregue en la parte inferior una imagen de la receta (la misma se encuentra en la carpeta ‘Ejercicios-HTML/imagenes’)
- 4) Empleando el template “ej\_noticia”, incluir la etiqueta semántica <nav> debajo de la cabecera de la página, y agregar un link a la web de la receta (como si fuera un diario con diferentes secciones).
- 5) Empleando el template “ej\_tablas”, agregue las siguientes etiquetas:
  - a) table, tr, td

Siglas	Congreso o Evento	Carrera	Web	Lugar	Fecha
CPAR	Campus Party	Computación/Sistemas	<a href="http://argentina.campus-party.org/">http://argentina.campus-party.org/</a>	Buenos Aires	25-abr al 28-abr
JAIIO	Jornadas Argentinas de Informática	Computación/Sistemas	<a href="http://47jaiio.sadio.org.ar">http://47jaiio.sadio.org.ar</a>	Buenos Aires	3-sep al 7-sep
WICC	Workshop de Investigadores en Ciencias de la Computación	Computación/Sistemas	<a href="http://wicc2018.unne.edu.ar/">http://wicc2018.unne.edu.ar/</a>	Corrientes	18-mar al 27-abr

- b) thead, tbody y th

Siglas	Congreso o Evento	Carrera	Web	Lugar	Fecha
CPAR	Campus Party	Computación/Sistemas	<a href="http://argentina.campus-party.org/">http://argentina.campus-party.org/</a>	Buenos Aires	25-abr al 28-abr
JAIIO	Jornadas Argentinas de Informática	Computación/Sistemas	<a href="http://47jaiio.sadio.org.ar">http://47jaiio.sadio.org.ar</a>	Buenos Aires	3-sep al 7-sep
WICC	Workshop de Investigadores en Ciencias de la Computación	Computación/Sistemas	<a href="http://wicc2018.unne.edu.ar/">http://wicc2018.unne.edu.ar/</a>	Corrientes	18-mar al 27-abr

- c) Elimine las siglas CPAR, y unifique esa celda con la contigua para que ambas contengan el texto “Campus party”

Siglas	Congreso o Evento	Carrera	Web	Lugar	Fecha
Campus Party		Computación/Sistemas	<a href="http://argentina.campus-party.org/">http://argentina.campus-party.org/</a>	Buenos Aires	25-abr al 28-abr
JAIIO	Jornadas Argentinas de Informática	Computación/Sistemas	<a href="http://47jaiio.sadio.org.ar">http://47jaiio.sadio.org.ar</a>	Buenos Aires	3-sep al 7-sep
WICC	Workshop de Investigadores en Ciencias de la Computación	Computación/Sistemas	<a href="http://wicc2018.unne.edu.ar/">http://wicc2018.unne.edu.ar/</a>	Corrientes	18-mar al 27-abr

- d) Como en la columna “Lugar”, ‘Buenos Aires’ se repite, unificar ambas celdas.

Siglas	Congreso o Evento	Carrera	Web	Lugar	Fecha
Campus Party		Computación/Sistemas	<a href="http://argentina.campus-party.org/">http://argentina.campus-party.org/</a>	Buenos Aires	25-abr al 28-abr
JAIIO	Jornadas Argentinas de Informática	Computación/Sistemas	<a href="http://47jaiio.sadio.org.ar">http://47jaiio.sadio.org.ar</a>	Buenos Aires	3-sep al 7-sep
WICC	Workshop de Investigadores en Ciencias de la Computación	Computación/Sistemas	<a href="http://wicc2018.unne.edu.ar/">http://wicc2018.unne.edu.ar/</a>	Corrientes	18-mar al 27-abr

- e) Al contenido de la columna “Web”, agregar links a las páginas correspondientes.
- 6) Agregue comentarios en la tabla del ejercicio anterior.
- 7) Empleando el template “ej\_formulario”, agregue las siguientes etiquetas:
- form** para agrupar todos los campos
  - input** tipo texto, radio, numero, email, password, checkbox y submit
    - El password debe tener como mínimo 8 caracteres, y como máximo 16
    - Emplee el atributo size=40 en los campos de texto
  - select** y **option** para el selector
  - textarea**
  - Atributo **placeholder** en todos los inputs

### Suscribase a Nuestro Portal de Noticias! Crear una cuenta es rápido!

Nombre:

Apellido:

Sexo: Masculino ☐ Femenino ☐

Edad:

Email:

Usuario:

Password:

Confirmar Password:

Tipo de servicio:

Tema: ☐ Política ☐ Negocios ☐ Tecnología ☐ Salud

Agregue un comentario:

- 8) Empleando el template “ej\_animales”, agregue las etiquetas necesarias para:
- Agregar un audio (el mismo se encuentra en la carpeta “resources”)

- b) Agregar un video (el mismo se encuentra en la carpeta “resources”)
- c) Agregar un video de youtube



### Ejercicios de Accesibilidad

- 9) En el formulario que ha creado, emplee la etiqueta `<label for="mismo_id">`
- 10) Agregue el atributo **alt** con el texto alternativo a todas las imágenes de su sitio web.
- 11) Emplee el tag **caption** para agregar títulos a todas las tablas del sitio web.



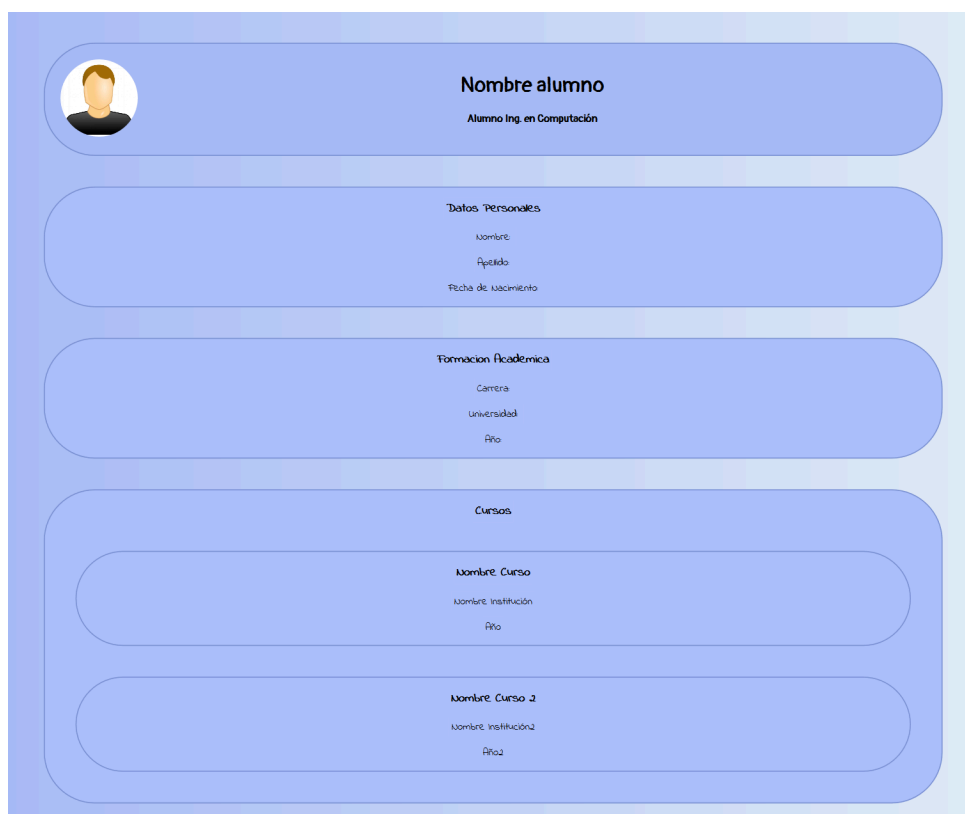
**Recomendación:** Puedes utilizar alguna plataforma como [DevTalles](#), Udemy, Codecademy, Digital House, Code School para aprender y practicar lenguajes como HTML, CSS.



### Ejercicios de CSS

- 12) Empleando el template “ej\_columnas”, agregue estilo en línea para cambiar:
  - a) Color del texto del título de la página.
  - b) Tamaño de la fuente del título de la noticia.
  - c) Tipo de fuente o letra de toda la página.
- 13) Empleando el template “ej\_columnas”, cambie los estilos en línea por estilos incrustados.
  - a) El título de la página debe estar definido por ID
  - b) El título de la noticia debe estar definido por clase
  - c) El tipo de fuente para toda la página debe estar definido por Tag
- 14) Empleando el template “ej\_leyes” y hoja de estilo incrustada, agregue los estilos necesarios para cambiar:
  - a) La primera lista ordenada para visualizarla con números romanos y una tipografía de mayor tamaño (upper-roman).
  - b) La segunda lista ordenada para visualizarla con letras (lower-alpha).
- 15) Modifique el estilo del hipervínculo de la página de “ej\_columnas” empleando pseudo-clases:
  - a) Link visitado o no visitado con el mismo color
  - b) Al posicionar el mouse arriba del link, modificar el tamaño de la fuente
  - c) Al seleccionar el link, que se visualice bold

- 16) Cree una página nueva, emplee de fondo la imagen “fondo\_mario.jpg” ubicada en la carpeta de “imagenes”. Pruebe los diferentes atributos de background: image, attachment, blend-mode, clip, color, image, origin, position, repeat, size.
- 17) Empleando el template “ej\_cv”, agregue su información y cambie los estilos empleando una hoja de estilo enlazada:
- Agregar estilo a los **divs** para hacer que sus bordes sean visibles.
  - Agregar los atributos de **margin** y **padding** para todos los divs.
  - Agregar color de fondo con opacity a los divs
  - Agregue una imagen en el primer <div> y alinee a la izquierda.
  - Centrar todo el contenido del body.
  - Agregar fuentes de google para personalizar la página.
  - Agregue un background al body, que posea gradiente.
  - Redondear las esquinas de la imagen para dejarla circular.
  - Redondear las esquinas de los <div>.
  - Cambiar el **favicon** o icono de la página.
  - Agregue sombras a los textos de los títulos.
  - Agregue sombras a los divs.
  - Asegúrese de por lo menos incluir un estilo por Tag, por ID y por clase



The image shows a screenshot of a CV template with a light blue gradient background. It features several rounded rectangular sections for personal and academic information. The first section includes a circular profile picture placeholder and a title 'Nombre alumno' with a subtitle 'Alumno Ing. en Computación'. Subsequent sections are titled 'Datos Personales', 'Formacion Academica', and 'Cursos', each containing placeholder text for fields like Name, Last Name, Date of Birth, Career, University, Year, Course Name, Institution, and Year.



18) Empleando el template “ej\_columnas”, ordene el texto en 2 columnas.

- a) Empleando cajas flotantes (**float**).
- b) Empleando **column-count**.
- c) Empleando **grid-template-column**

## Diario On-line

### CordoBici

Publicado el 07/03/2018 en Agropecuarias y Ambiente

Un intercambio académico inspiró a Joaquín, Nicolás, Guido y Facundo para elegir el tema de su trabajo final. Dos de ellos estuvieron el año pasado en Sevilla y se quedaron “enamorados” –palabras textuales– de su sistema de bicicletas públicas; de su comodidad y rapidez, y de lo económico y saludable que resultaba.

Pensaron que si funcionaba en una cultura similar a la nuestra, sería posible replicarlo en Córdoba. Esta iniciativa no solo se convirtió en su tesis, que les permitió hacer una semana convertirse en ingenieros industriales, sino que se trata de una valiosa propuesta que la Municipalidad de nuestra ciudad está interesada en implementar.

#### Investigación innovadora

El sistema de bicicletas compartidas no es nuevo, ya funciona en muchos lugares del mundo. En nuestro país lo encontramos en Buenos Aires y Rosario. De hecho en esas ciudades los ingenieros realizaron gran parte de su investigación, que es lo innovador.

Estudiaron su procedimiento, relevaron funcionamientos, contactaron proveedores y mantuvieron contacto con la Municipalidad de Córdoba para conocer el escenario en donde se implementaría el proyecto. A partir de esta información estadística diseñaron un sistema para optimizar lo que ya existe. “Lo que han hecho es mejorar la propuesta, agregando un know how”, explica Henry Martínez, coordinador de la carrera y director de la tesis.

Gracias a una profunda investigación comercial, sumada a una simulación realizada por un programa llamado anylogic, consiguieron localizar de manera estratégica 19 estaciones con 280 bicicletas, en Nueva Córdoba y Ciudad Universitaria.

Además, plantearon un sistema de mantenimiento in situ y preventivo que permitiría reducir de manera notable las roturas de las bicicletas, evitando así accidentes y descontento por parte de los usuarios.

Este sistema se diferencia de los otros de nuestro país al incorporar un sistema de información geográfica, el cual, de acuerdo a la variación de la demanda, la pendiente y el flujo de gente, simula el funcionamiento del mismo. De esta forma, se determina la ubicación ideal de las estaciones y la cantidad óptima de bicicletas en cada una de ellas, previo a la implementación.

Por su parte, otros alumnos de la Facultad de Ingeniería estarían desarrollando el soft para complementar el sistema con el programa que permite hacer el pedido. Se trata de una aplicación para que la bici se destrabe cuando se manda un mensaje de texto.

Esta propuesta mejoraría notablemente no solo la movilidad de los habitantes de Córdoba, sino también su calidad de vida, porque impactará directamente en su salud física y mental, le ahorrará tiempo de traslado y gastos de transporte. De esta manera, la iniciativa de nuestros ingenieros es un valioso aporte a la sustentabilidad de los ciudadanos de nuestra ciudad.

## B

### Ejercicios de Bootstrap

- 19) Empleando el template “ej\_instagram”, visualizar la página en un grid 3x3 que sea fluido según el tamaño de pantalla.
- 20) Empleando el template “ej\_form”, agregue las clases necesarias para visualizar el formulario correctamente.



**Recomendación:** Puedes consultar todos los tags, atributos y estilos disponibles en el estándar en “W3schools” o “MDN”.



### Para aprender más...

- 1) Nombre al menos 3 nuevas etiquetas que surgieron a partir de HTML5
- 2) Nombre 6 atributos nuevos que surgieron a partir de HTML5 para campos en formularios.
- 3) ¿Cómo se incorporan videos y sonido a una página web empleando etiquetas HTML?
- 4) ¿Qué etiquetas se emplean para ordenar la cabecera, pie de página, artículos y sumarios?
- 5) ¿Además de BootStrap, que otros Frameworks Front-End existen?
- 6) ¿Qué son los gestores de contenido? Nombre al menos 2 gestores de contenido.



**¡Trabajo de Evaluación!**: Utiliza las etiquetas recientemente aprendidas para comenzar a generar la página web prototipada.

#### REQUISITOS DEL PRIMER PARCIAL

##### Sobre el Sketch:

- ☐ Deben estar realizados tanto para desktop como para mobile.
- ☐ Debe tener en cuenta los mensajes de error para el usuario.
- ☐ Debe estar guardado en formato PNG, JPG ó PDF.
- ☐ Deben estar guardados en el repositorio dentro de una carpeta llamada "Sketch"

##### Sobre el Wireframe/Mockup:

- ☐ Deben estar realizados con algún programa como: [Figma](#), [AdobeXD](#), Draw.io, Pencil project, Mockups, NinjaMoc, o similares.
- ☐ Deben estar realizados tanto para desktop como para mobile.
- ☐ Debe tener en cuenta los mensajes de error para el usuario.
- ☐ Debe estar guardado en formato PNG, JPG ó PDF.
- ☐ Deben estar guardados en el repositorio dentro de una carpeta llamada "Wireframe" ó "Mockup"

##### Sobre el repositorio:

- ☐ El repositorio del proyecto debe estar en "Github Classroom", con el nombre Proyecto2024-ApellidoAlumno1-ApellidoAlumno2.
- ☐ Modificar el Readme.MD y colocar información del proyecto/página (mínimamente: título del proyecto, autores, link de gh-pages, contenido de la página, listado de tecnologías usadas, etc) y cualquier otra información de interés para el proyecto.
- ☐ La estructura del proyecto debe ser adecuada (crear una carpeta para las imágenes, otra para los sketch/mockups). Es importante que contenga SKETCH y MOCKUP/Wireframe (debe estar en formato: PDF, PNG o JPG)
- ☐ El código debe estar pusheado en el repositorio (emplear gh-pages o configurar github para que se toma a la main como la página a visualizar), y no debe haber diferencias entre main y gh-pages (verificar de realizar el Merge).
- ☐ Publicar la Web empleando GitHubPages
- ☐ El repositorio no debe contener archivos innecesarios (no debe contener .idea o .vsc o .DS\_Store, en todo caso emplear .gitignore)
- ☐ "Lorem ipsum" es sólo válido para los prototipos, NO para la página.

**Sobre el proyecto general:**

- ☐ NO está permitido descargar un template (diseño 100% desde cero)
- ☐ La página principal debe llamarse index
- ☐ Identificar correctamente el código (en Webstorm Ctrl+Alt+L)
- ☐ No debe haber errores de código presentes (realizar Code > Inspect Code para verificar que no haya errores)
- ☐ No debe haber errores en la consola
- ☐ Se debe emplear favicon
- ☐ Emplear alguna fuente de google fonts o subir al proyecto alguna fuente externa (aunque sea para un título)
- ☐ Debe haber navegación entre todas las páginas

**Sobre el HTML:**

- ☐ Todas las etiquetas deben estar en minúscula
- ☐ Poner comillas a todos los atributos
- ☐ Title debe contener el título de la página
- ☐ En el <head></head> incluir las etiquetas <meta> detallando: autor, descripción y palabras clave
- ☐ Emplear al menos 3 etiquetas semánticas (header, nav, aside, main, section, article, footer)
- ☐ Emplear <header></header>. En el contenido de la cabecera debe haber un título <h1></h1>, puede tener color de fondo, algún logotipo, etc.
- ☐ Debe haber por lo menos una etiqueta <img> en la página.
- ☐ La estructura de la página debe estar definida con <div></div>
- ☐ Debe contener al menos 3 elementos de tipo <input> o <select> o <button> que le permitan al usuario ingresar valores para poder realizar un cálculo de un ejercicio o seleccionar opciones.
- ☐ Debe contener un <canvas>, para que posteriormente (en la 2da etapa) se grafique alguna imagen representativa del ejercicio. (el desarrollo del gráfico se realizará cuando se vea la unidad de "Orientación a Eventos")
- ☐ No espaciar con excesivos <br>. Utilizar márgenes, paddings, etc.
- ☐ No utilizar etiquetas deprecadas.
- ☐ Todas las etiquetas que correspondan deben estar correctamente cerradas
- ☐ Los ids de los elementos deben ser unívocos

**Sobre el CSS:**

- ☐ El estilo de los elementos debe establecerse en un archivo CSS (prohibido poner el atributo style a los elementos o emplear estilos incrustados).
- ☐ El CSS debe contar mínimo con un tipo de cada forma (por Tag, por ID y por clase).
- ☐ Se debe emplear pseudoclase
- ☐ No emplear !important

**Sobre accesibilidad:**

- ☐ Toda imagen debe tener su etiqueta alt
- ☐ Todo `<input>` o `<select>` debe tener su `<label>`
- ☐ Los labels deben contener el atributo for (el for debe contener el id del input al cual se referencia)
- ☐ Si hay una tabla en la página, debe contener `<caption></caption>`

## UNIDAD V: Orientación a eventos

La programación dirigida por eventos es la base de lo que llamamos interfaz de usuario. El usuario puede interactuar con el entorno visual haciendo click, escribiendo, arrastrando, etc. El programa debe responder ante estos eventos y realizar acciones, a esto se le llama orientado a eventos.

### TRABAJO PRÁCTICO Nº 6

#### Orientación a Eventos

#### Objetivos de Aprendizaje

- Adquirir conceptos básicos sobre orientación a eventos.
- Adquirir habilidad en el desarrollo de Js para para página web.

#### Unidad temática que incluye este trabajo práctico

- Este trabajo práctico corresponde a la unidad Nº 5 de la programación de la asignatura.

#### Consignas a desarrollar en el trabajo práctico



#### Preguntas

- 1) ¿Qué es la programación dirigida por eventos? De ejemplos de lenguajes orientados a eventos.
- 2) Explique qué es un evento. ¿Qué tipos de eventos hay?
- 3) ¿Qué es el event handler? ¿Qué es el event listener?
- 4) ¿Qué es JavaScript? ¿Qué características tiene? ¿Qué son los identificadores? ¿Qué tipos de variables existen en JS?
- 5) ¿Qué herramientas permiten debuggear Javascript?
- 6) ¿Qué es la jerarquía de objetos en Js?
- 7) ¿Qué es la refactorización de código?
- 8) ¿Qué es EcmaScript?
- 9) ¿Qué diferencia hay entre Canvas y SVG?
- 10) ¿Qué es localStorage? ¿Qué es el Session Storage?



### Ejercicios JavaScript

- 1) Mostrar un “alert” con el texto “Hola Mundo!” que se ejecute en el onload del body.
- 2) Agregue la etiqueta de **script** en la sección **head** y cree una función llamada ‘saludar’ que muestre un alert con “Hola Mundo!”.
- 3) Empleando la página web diseñada en la unidad anterior, escribir una función/funciones JavaScript en un archivo externo para que:
  - a) Al escribir un número en cualquiera de los inputs, la unidad sea convertida.
  - b) Si el valor ingresado no es numérico, muestre un mensaje de error al usuario y blanquee los campos.
  - c) Si el usuario ingresa una ‘,’ la reemplace por un ‘.’

## Convertidor de Unidades

metro=

pulgada=

pie=

yarda=

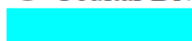
- 4) Documentar las funciones del “Conversor de Unidades” adecuadamente, indicando que hacen las funciones, el nombre del método, que parámetros se le envía y que valor retorna.
- 5) Cree una web con 2 inputs para convertir de grados a radianes. Recuerde que  $180^{\circ} = \pi \text{ rad}$

Grados =  Radianes

- 6) Empleando la página web diseñada en la unidad anterior, escribir una función JavaScript para mostrar u ocultar el div celeste.

## Mostrar/Ocultar Div

☒ Mostrar Div  
☐ Ocultar Div



- 7) Empleando el template de página web existente en el repositorio sobre “operadores matemáticos”, escribir funciones JavaScript para resolver las operaciones matemáticas una vez que el usuario ingresa un valor numérico en ambos inputs. Tener en cuenta:
  - a) El valor del resultado no debe poder ser modificado por el usuario.

- b) Asegúrese de castear los valores de ser necesario.

## Operaciones Matemáticas

$$\boxed{\phantom{00}} + \boxed{\phantom{00}} = \boxed{\phantom{00}}$$

$$\boxed{\phantom{00}} - \boxed{\phantom{00}} = \boxed{\phantom{00}}$$


$$\boxed{\phantom{00}} \times \boxed{\phantom{00}} = \boxed{\phantom{00}}$$

$$\boxed{\phantom{00}} / \boxed{\phantom{00}} = \boxed{\phantom{00}}$$

- 8) Empleando el ejercicio de “Convertidor de Unidades”, modificar las funciones de manera tal que el resultado de la conversión sólo posea 2 decimales. (Emplear Math.Round)
- 9) Empleando el ejercicio de “Operaciones Matemáticas”, modifique el HTML para quitar los inputs de los resultados, y emplear div, y en JavaScript, emplee innerHTML.
- 10) Crear 2 páginas webs. En la primera, incluir un input para ingresar distancia, un selector para seleccionar la unidad y un botón para ir a la segunda página.

distancia=  metros

En la segunda página ingresar un texto con un input para completar con los valores anteriormente ingresados.

 segundaWeb.html#12#metros

La distancia ingresada es

Crear una función JavaScript que sea ejecutada cuando el botón sea presionado. La función debe concatenar el nombre de la segunda web con los valores ingresados por el usuario, separados por '#' y dirigirse a la segunda web.

Crear una función JavaScript que se ejecute cuando la segunda web sea cargada. Debe tomar el valor de la URL y dividirla cada vez que '#' sea encontrado. Debe emplear estos valores y cargarlos en el input.

- 11) Implementar la misma funcionalidad del punto anterior empleando localStorage para “pasar los datos”.



### Ejercicios Canvas

- 1) Implemente una función javascript que dibuje un círculo y un cuadrado en un canvas. Llamarla al cargar la página.
- 2) Implemente una función javascript que dibuje un cuadrículado en un canvas. Llamarla desde un botón en la página.



- 3) Implemente una función que tenga como parámetro dos valores, X e Y, que dibuje una imagen en un canvas desde la posición pasada por parámetro. La página debe contener dos inputs donde el usuario ingresa los valores X e Y, y un botón “Dibujar” desde donde se llama a la función.
- 4) Utilizando la función anterior que dibujaba imágenes y la función setTimeout() de javascript, realice una animación que mueva la imagen desde el borde izquierdo al derecho del canvas.



**Recomendación:** Es importante ir probando la página progresivamente, cada vez que se le realizan cambios. Deben emplearse diferentes navegadores para asegurarse que la funcionalidad es la correcta.



### Para aprender más...

- 1) Nombre al menos 3 librerías JS que se pueden agregar a una página web, y explique brevemente para qué sirve cada una.
- 2) ¿Qué es y para qué sirve: Angular, React, JQuery, KinecticJS, MelonJS, KiwiJS?
- 3) ¿Qué es node.js?



**¡Trabajo de Evaluación!** Utiliza lo recientemente aprendido para agregarle funcionalidad a la página web entregada en la anterior evaluación.

### REQUISITOS DEL SEGUNDO PARCIAL

#### Sobre el proyecto general:

- ☐ Todas las correcciones y mejoras solicitadas durante el primer parcial deben estar corregidas.
- ☐ No debe haber errores presentes en el código (realizar Code > Inspect Code para verificar que no haya errores)
- ☐ No debe haber errores JavaScript presentes (F12 > Consola)

#### Sobre la funcionalidad JavaScript:

Se debe agregar funcionalidad Js a la página HTML+CSS desarrollada

- ☐ Una función que compruebe si los valores ingresados son correctos, y si no lo son, que le indique al usuario por un alert, y que borre/blanquee el contenido del campo.
- ☐ Una función que calcule algo en base a los valores ingresados por el usuario en los inputs.
- ☐ Una función que realice un dibujo sobre un canvas (debe ser representativo y acorde a los valores ingresados).
- ☐ El código JS debe estar en un archivo externo



- ☐ Se debe emplear var, let o const según corresponda para mayor eficiencia.
- ☐ No deben existir funciones innecesarias que no se llamen en ninguna sección del código
- ☐ Las funciones deben estar escritas como función flecha

**Sobre la documentación:**

- ☐ TODAS las funciones JavaScript deben estar comentadas adecuadamente. JsDoc

```
/**
 * Descripción de que hace la función
 * @method Nombre de la función
 * @param {String} ParámetroA - Explicación de qué valor almacena
 * @param {number} ParámetroB - Explicación de qué valor almacena
 * @return {bool} Valor que retorna
 */
```

## UNIDAD VI: Aplicaciones Móviles

Los dispositivos móviles son muy utilizados en la actualidad y el desarrollo de software para ellas debe tenerse en cuenta ciertas particularidades debido a sus características reducidas, métodos de entrada diferentes a un PC y escasos recursos.

### TRABAJO PRÁCTICO Nº 7

#### Aplicaciones Móviles

#### Objetivos de Aprendizaje

- Adquirir conceptos básicos sobre aplicaciones móviles.
- Adquirir habilidad en el desarrollo de aplicaciones móviles.

#### Unidad temática que incluye este trabajo práctico

Este trabajo práctico corresponde a la unidad Nº 6 de la programación de la asignatura.

#### Consignas a desarrollar en el trabajo práctico



Puedes consultar el libro [\*“Diseñando apps para móviles”\*](#) de Javier Cuello y José Vittone.



Primera unidad teórica del libro “Programación multimedia y dispositivos móviles” de Dominguez, Paredes y Santacruz.



#### Preguntas

- 1) ¿Qué es una aplicación móvil? De ejemplos de aplicaciones móviles.
- 2) ¿Cómo se pueden categorizar las aplicaciones móviles?
- 3) ¿Qué lenguajes se utilizan para programar las principales plataformas móviles? (Android, iOS y Windows Mobile)
- 4) ¿Qué diferencias hay entre aplicaciones nativas, web e híbridas? ¿Qué ventajas y desventajas presenta cada una?
- 5) ¿Qué es Cordova? ¿Qué otras alternativas existen como esta?
- 6) ¿Qué significa que un sitio sea responsivo? ¿Qué características tiene? ¿Qué frameworks existen para facilitar el desarrollo de sitios responsivos?
- 7) ¿Qué son las media query? ¿Qué son los media types? ¿Qué es el viewport?



## Ejercicios

- 1) Emplee alguna herramienta para probar si su sitio web es responsive.

Se sugiere usar 'Chrome Developer Tools' Ctrl+Mayus+i

- 2) Cree una nueva página web e incluya la librería 'Bootstrap'. Cree 3 divs que ocupen 4 columnas cada uno. Aplique estilos propios de bootstrap para que las columnas se adapten al tamaño de la pantalla.
- 3) Utilizando Cordova, haga una aplicación híbrida de CV.



**Recomendación:** Puedes utilizar Frameworks para Front-End como Bootstrap o Foundation para facilitar que su página sea responsive.



## Para aprender más...

- 1) ¿Para qué sirven las tiendas de aplicaciones móviles?
- 2) ¿Qué hardware del dispositivo es accesible desde HTML5? ¿GPS, acelerómetros, etc?
- 3) Cree una nueva página web con una tabla e incluya la librería 'Bootstrap'. Pruebe diferentes estilos de tablas.