

# Responsive Web Design



Programación Web I / Móvil I

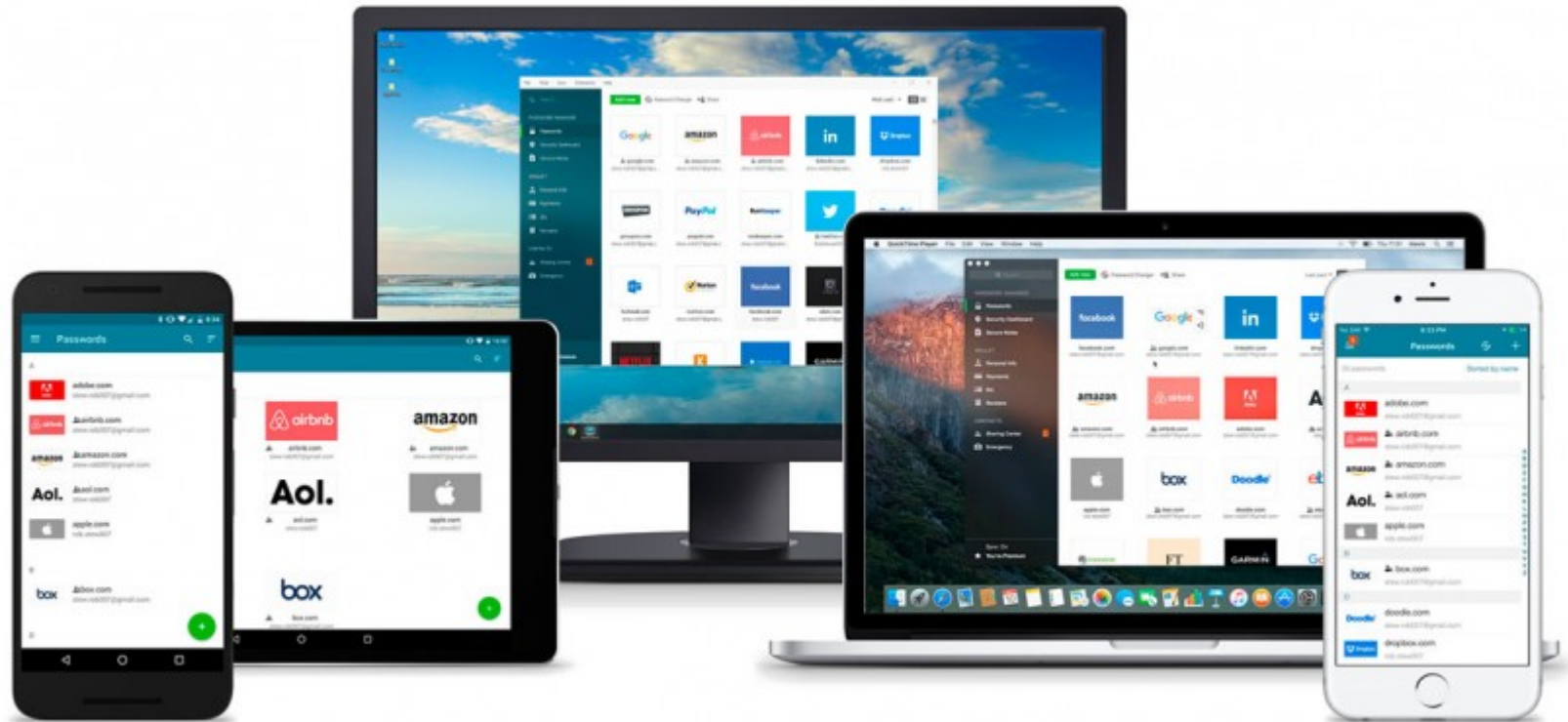
**Comisión Miércoles Noche**

DIS. Alicia Rosenthal

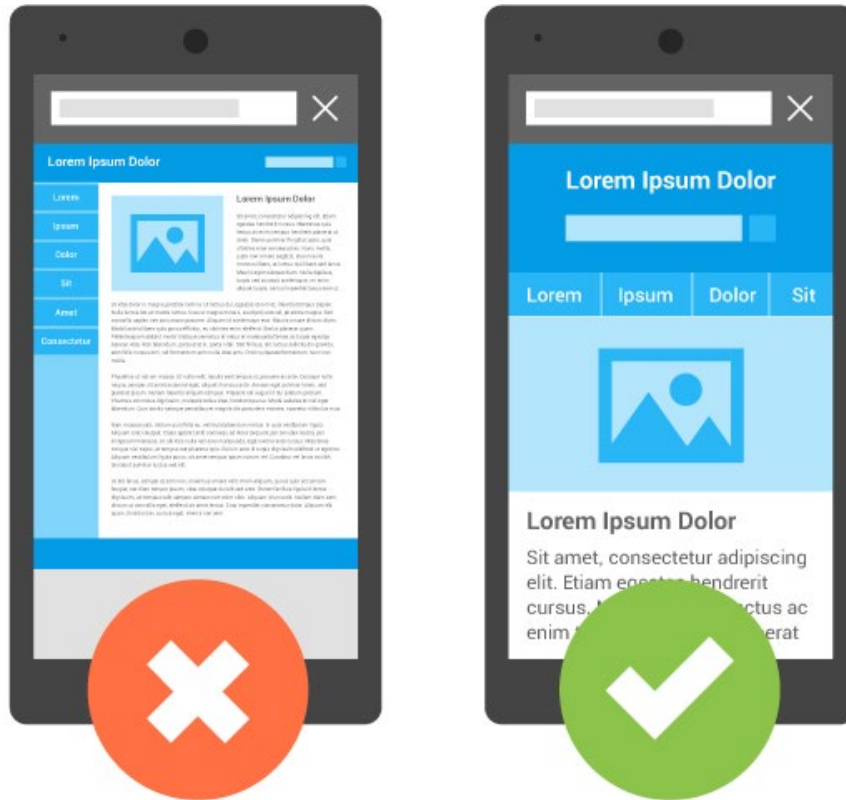
**Comisión Viernes Mañana**

Ing. Gabriel Panik

# Dispositivos



# MetaViewport



[https://www.w3schools.com/css/css\\_rwd\\_viewport.asp](https://www.w3schools.com/css/css_rwd_viewport.asp)

# MetaViewport

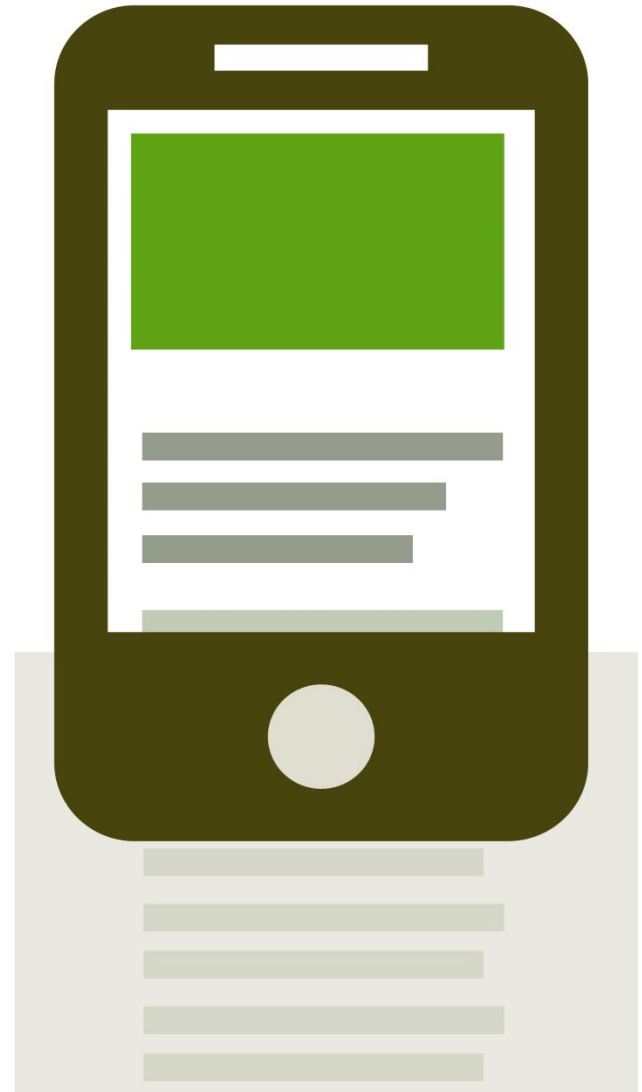
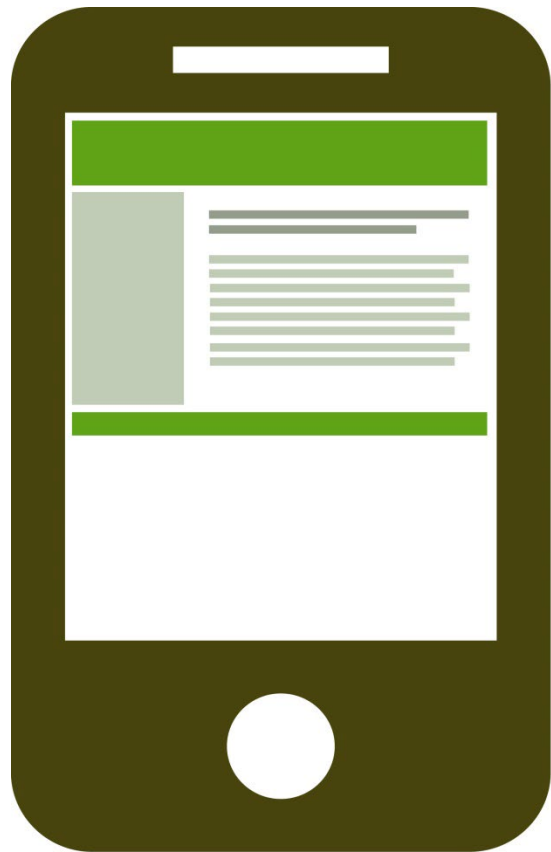
```
<head>
```

```
  <meta name="viewport"  
    content="width=device-width,  
    initial-scale=1, maximum-  
    scale=1">
```

```
</head>
```

Ejemplo: <http://www.edenor.com.ar/>

# MetaViewport



# Medias Queries

**Media Queries** es un módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo.

Se implementa mediante condiciones sobre el CSS que permiten decidir que porciones del mismo deben aplicarse dependiendo el contexto

# Medias Queries

## SINTAXIS:

```
@media type and (condicion) {  
}
```

## Ejemplos de aplicación:

```
@media screen and (min-width: 500px) {  
}
```

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

# Medias Queries

## Dispositivos (media type)

screen

print

all

## Condiciones /expresiones

min-width: valor (expresado en px o en ems)

max-width: valor (expresado en px o en ems)

orientation: landscape | portait



# Medias Queries

**Ejemplo:** Modificar el color de fondo del body si estamos trabajando sobre pantallas con ancho mayor o igual a 768 pixeles:

```
@media screen and (min-width: 768px) {  
    body {  
        background-color: #ccc;  
    }  
}
```

# Medias Queries

**Ejemplo:** Modificar el color del texto de un párrafo a rojo en caso de que el dispositivo posea un ancho en píxeles entre 500 y 700:

```
@media screen (max-width: 700px) and  
(min-width: 500px) {  
  
  p {  
  
    color: red;  
  
  }  
  
}
```

# Mobile First



Responsive Web Design

---

Mobile First Web Design



# Medias Queries

```
body{  
  background: green;  
  font-size: 70%;  
}  
p{  
  font-size: 1em;  
}
```

Estilos Mobile

```
@media screen and (min-width:47em){
```

```
  body{  
    font-size: 90%;  
  }
```

Estilos Tablet

```
}
```

```
@media screen and (min-width:60em){
```

```
  body{  
    font-size: 100%;  
  }
```

Estilos Pcs de  
Escritorio

```
}
```

# Unidades de Medida

Para responsive, nos vamos a manejar con estas unidades de medida:

- **%** porcentaje. Para anchos de columnas, imágenes y video
- **em** para tamaños de fuentes, márgenes, paddings.

**1em = 16 píxeles aprox.**

# Tamaños de fuentes

Establecemos primero los tamaños de fuentes en em.

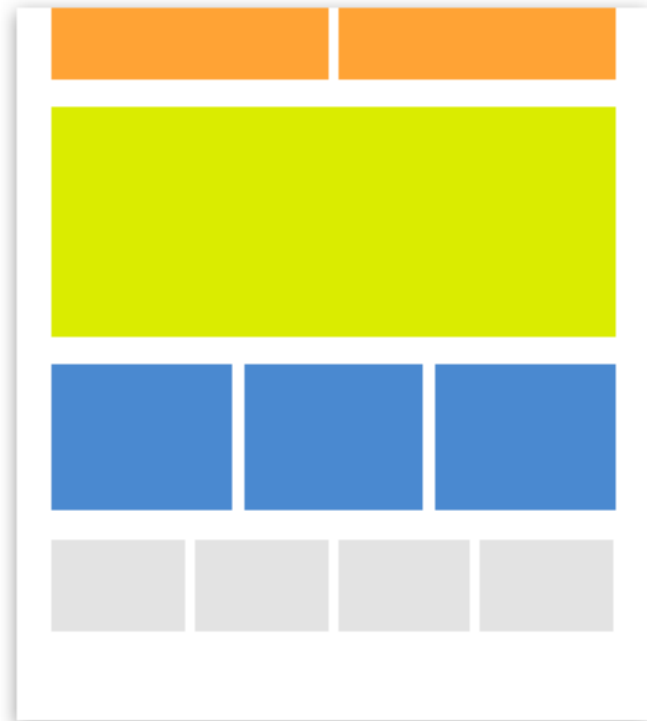
```
h1{  
    font-size: 1.6em;  
}  
h2{  
    font-size:1.5em;  
}  
p{  
    font-size: 1em;  
}
```

# Tamaños de fuentes

Trabajamos en el body con porcentajes para los distintos tamaños.

```
body{
    font-size:70%;
}
@media screen and (min-width:47em){
    body{
        font-size:80%;
    }
}
@media screen and (min-width:60em){
    body{
        font-size:100%;
    }
}
```

# Columnas





# Columnas

```
.col{
  background-color: #7b1873;
  margin-top:1em;
}
@media screen and (min-width:47em){
  .contenedor{
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
  }
  .col{
    width:calc(50% - 1em);
  }
}
@media screen and (min-width:60em){
  .col{
    width:calc(33% - 1em);
  }
}
```

# Imágenes responsivas

Establecemos la propiedad max-width, que hace que nuestra imagen sea adaptable al ancho disponible, hasta llegar a su ancho máximo que es el ancho que tiene el archivo.

El alto, lo ponemos en automático.

```
img{  
    max-width: 100%;  
    height: auto;  
}
```

# Videos Responsives

```
.contenedor {  
    position: relative;  
    padding-bottom: 56.25%;  
    height: 0;  
}
```

```
.contenedor iframe{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
}
```