Desarrollo de Aplicaciones Web Empresariales

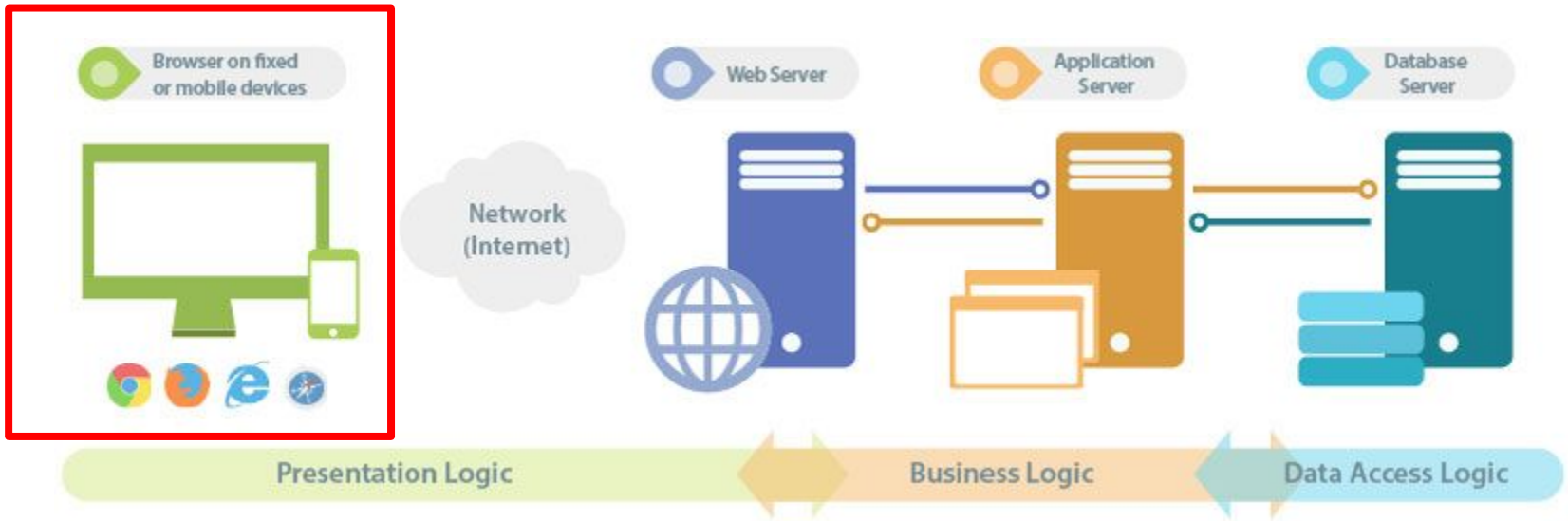
Exequiel Fuentes Lettura
exequiel.fuentes@ucn.cl

Información de contacto

- Exequiel Fuentes Lettura
  - Email: exequiel.fuentes@ucn.cl
  - Horario de Atención: Jueves y Viernes, bloque C

- Departamento de Ingeniería de Sistemas y Computación
  - Oficina: Y1 - 329
  - http://www.disc.ucn.cl

Arquitectura de aplicaciones Web



Una buena app web: Diseño e implementación

- Algunas metas de diseño:
  - Uso intuitivo, esto es, no se necesita tomar un curso o leer el manual de usuario.
  - Realizar la tarea con precisión y rapidez. Proporciona la información necesario y funcional.
  - A los usuarios les gusta la experiencia.
- La parte más difícil de una buena aplicación web es su diseño.
- Aquí hay algunas sugerencias:
  - http://www.goodui.org/#1
  - http://bokardo.com/principles-of-user-interface-design/

Algunos principios de diseño

- Ser consistente
  - Una carga menos para el usuario.
- Proporcionar contexto
  - El usuario no se debería perder en la aplicación.
- Ser rápido
  - No hacer perder el tiempo al usuario.

Consistencia

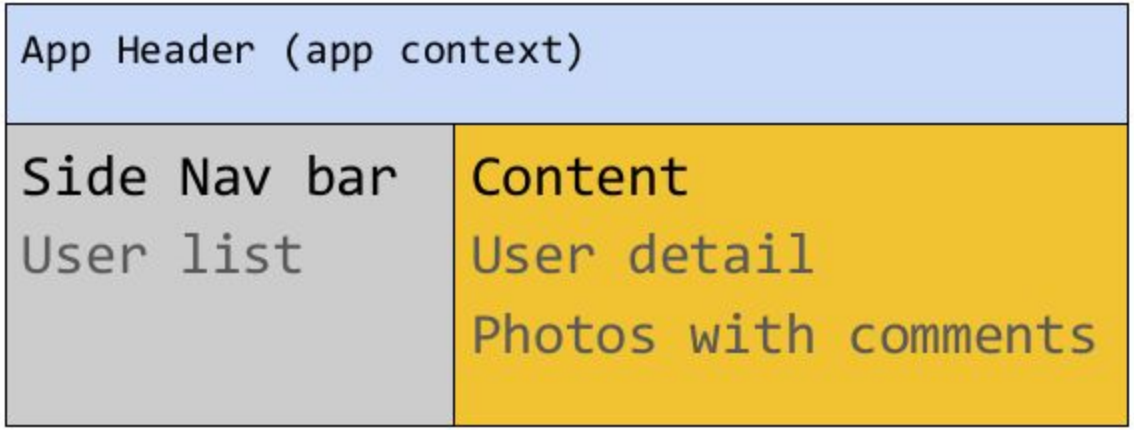
- Una aplicación web debería tener una guía de estilo, la que cubra el aspecto y comportamiento de la aplicación.
  - Estilo: colores, animación, iconos, imágenes, tipografía, etc.
  - Interacciones: menú, botones, diálogos, tablas, listas, etc.
  - Disposición: estructura, herramientas, contenido, etc.
- Patrones, si se hace algo de una forma en diferentes lugares siempre tiene que ser de la misma forma.
  - Reutilizar los componentes
  - Como se manejan los errores, navegación, notificaciones, etc.
- Plantillas de diseño, esto es seguir una estructura familiar
  - Utilizar una plantilla maestra.

Framework en el Front-end

- Ejemplo: Bootstrap (muy popular)
  - CSS: Plantillas, sistema basado en tabla.
  - Componentes HTML: botones, menús, herramientas, listas, etc.
  - Javascript: transiciones, dropdowns, etc.
  - http://getbootstrap.com/
- Angular Material
  - CSS y Angular implementan este diseño.
  - https://material.angularjs.org/latest/

Ejemplo: Angular Material para una app de fotos

- Se pueden encontrar varios diseños en:
  - http://codepen.io/team/AngularMaterial/pens/popular/
- Un diseño clásico puede ser:



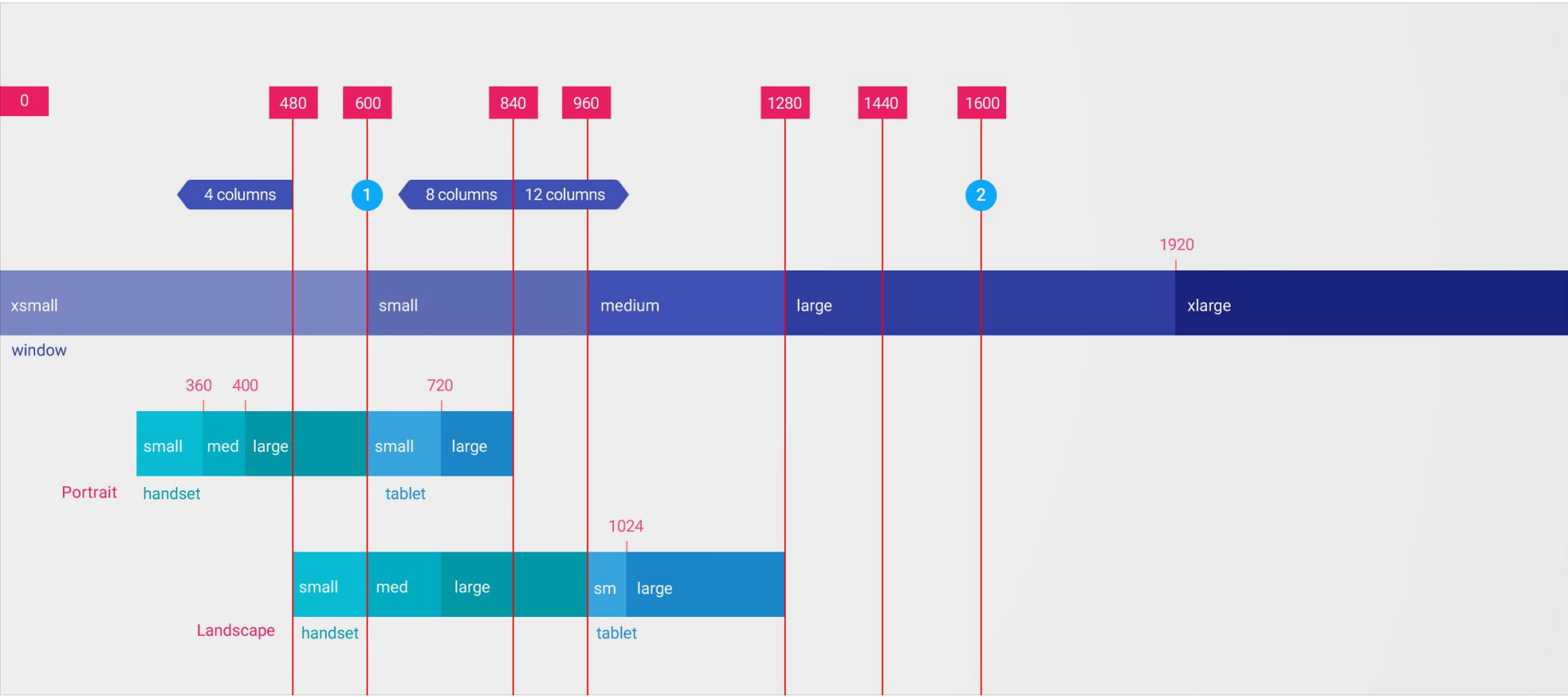
Usar un diseño de tabla para la app

```
<body layout="column">          <!-- Body is a single column with 2 rows
   <md-toolbar layout="row">       <!-- Row #1 is the header
       ...
  </md-toolbar>
  <div flex layout="row">          <!-- Row #2 has two columns
       <md-sidenav>                   <!-- Column #1 is the side nav bar
            ...
       </md-sidenav>
       <md-content flex>          <!-- Column #2 is the content area (flex)
          ...
       </md-content>
   </div>
</body>
```

Usar un diseño de tabla para la app

Soporte para diseño “responsive”

- Existen dos modelos que ayudan a disponer elementos:
  - Grid system (usado por Bootstrap)
  - Flexible Box (usado por Bootstrap y Angular Material)
- Utilizar esto puede hacer una gran diferencia.
- Links:
  - http://getbootstrap.com/css/#grid
  - http://v4-alpha.getbootstrap.com/layout/flexbox-grid/
  - http://flexboxgrid.com/
  - https://material.angularjs.org/latest/layout/introduction


0
480
600
840
960
1280
1440
1600
4 columns
1
8 columns
12 columns
2
1920
xsmall
small
medium
large
xlarge
window
360
400
720
small
med
large
small
large
Portrait
handset
tablet
1024
small
med
large
sm
large
Landscape
handset
tablet

Angular Material: hide/show

hide (display: none)	show (negates hide)	Activates when:
hide-xs	show-xs	width < **600**px
hide-gt-xs	show-gt-xs	width >= **600**px
hide-sm	show-sm	**600**px <= width < **960**px
hide-gt-sm	show-gt-sm	width >= **960**px
hide-md	show-md	**960**px <= width < **1280**px
hide-gt-md	show-gt-md	width >= **1280**px
hide-lg	show-lg	**1280**px <= width < **1920**px
hide-gt-lg	show-gt-lg	width >= **1920**px
hide-xl	show-xl	width >= **1920**px

https://material.angularjs.org/latest/layout/options

Accessible Rich Internet Applications (ARIA)

- Define cómo realizar contenido Web y aplicaciones Web (especialmente las desarrolladas con Ajax y JavaScript) más accesibles a personas con capacidades diferentes.
- Se debe agregar descripciones para los elementos que lo requieran
  - <img aria-label="{{photo.description}}"
  - <a aria-label="Photo of user {{user.name}}" ng-href=
- Links:
  - https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA
  - https://www.w3.org/WAI/PF/aria-practices/Overview.html
  - http://zomigi.com/blog/videos-of-screen-readers-using-aria-updated/

Internacionalización (I18N)

- Los usuarios pueden querer diferentes: textos, números, monedas, gráficos, etc.
- Ejemplo, considera: <h1>Getting Started</h1>
  - <h1>{{i18n.GettingStarted}}</h1>

    <h1 translate>Getting Started</h1>

    <h1>{{"Getting Started" | translate}}</h1>
- No aplicar al contenido generado por el usuario:
  - <h1>Hello {{person.firstName}}</h1>

Pruebas en aplicaciones web

- Pruebas de unidad
  - Cada prueba apunta a un componente en particular y verifica que haga lo que tiene que hacer.
  - Requiere de componentes “mock” para realizar interacciones con otros componentes.
- Pruebas End-to-End (e2e)
  - Ejecuta pruebas contra la aplicación web real.
  - Scripts en el browser se utilizan para probar (Selenium, Protractor, entre otros).
- Métricas: Test Coverage
  - ¿Cuántas líneas de código han sido probadas?
- Se puede usar: Protractor, Karma, Jasmine y PhantomJS.

¿Preguntas?