

# Enunciado.

---

## Objetivos

- Añadir adecuadamente recursos multimedia en nuestro sitio web responsive.
- Utilizar el inspector DOM, el editor de estilos y las herramientas para diseño responsive del navegador como ayuda para el desarrollo.
- Implementar adecuadamente las, dirección de arte y hacerlo de forma accesible.

## Descripción

- El ejercicio consta de dos partes, que básicamente se configuran en **desarrollo** y **documentación**. Es necesario realizar y entregar las dos partes de la práctica para poder ser evaluado.

## Instrucciones para la realización del portal:

- La página debe ser **RESPONSIVE**.
- Las opciones del menú están programadas como una lista. Cada opción es un hipervínculo cuyo color de fondo cambia al pasar el ratón por encima (LIBRE EL COLOR).
- La página está compuesta por:
  - Header -NAV-MAIN-Footer
  - Menú del footer estar programada como una lista. Cada opción es un hipervínculo cuyo color de fondo cambia al pasar el ratón por encima (LIBRE EL COLOR).
  - **Las redes sociales del footer deben ser programadas como una lista.** Cada opción es un hipervínculo cuyo color de fondo cambia al pasar el ratón por encima (LIBRE EL COLOR).

Para el uso de los iconos de las redes sociales utilizaremos los iconos de [fontawesome](#)

- El MAIN de la página de module-bunders debe tener un formato de artículo.
- Se valorará que la página esté validada.

El sitio web deberá seguir las características que se especifican a continuación:

- Hay que partir del ejemplo. Sin embargo, puedes hacer las modificaciones que creas oportunas mientras mantengas la misma idea de estructura.

- Uno de los cambios es buscar-crear un nuevo logotipo que se integre en diseño.
- Nos centraremos en la página Module Bundlers y Lenguajes Front-End.

## Página Module Bundlers.

Con su diseño tipo artículo.



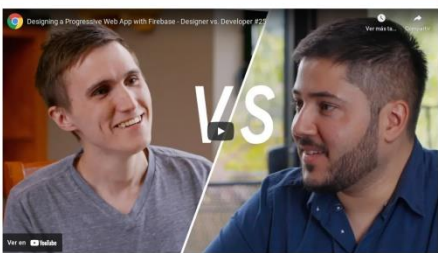
### Module Bundlers

Por nombre del alumno, 21 de noviembre de 2019

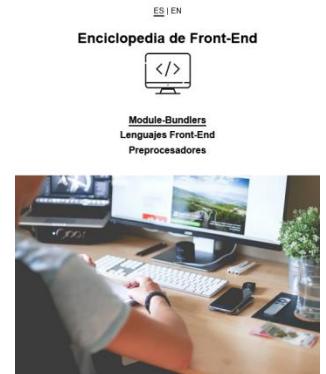


Profesora de ciclos formativos C.I. Politécnico de Estella de la rama de Informática. Profesora Grado Diseño Escuela de Artes de Corella. Diseñadora freelance. Cambiar por un pequeño texto que os describa.

Un module bundler nos permite automatizar una serie de tareas así como gestionar los assets de las dependencias que hemos instalado con npm si estamos utilizando ese gestor de paquetes. Parcel es un nuevo module bundler que promete unos tiempos de building ultrarápidos y zero config, por lo que se postula como un claro rival para la herramienta más usada actualmente: Webpack.



Parcel es un nuevo empaquetador de código que esta ganando popularidad y que puede ser muy útil en debido a su rapidez y características. La ventaja de la que más presumen en su web, y que lo convierten en un serio competidor es su velocidad. Otra de las ventajas más importantes que tiene es que no requiere de configuración, para los que llevamos años usando webpack esto es una agradable novedad, pues la configuración de los empaquetadores suele ser una tarea muy compleja y bastante delicada. El empaquetado de JS, CSS, HTML, archivos, etc es automático, sin que tengamos que añadirle plugins o loaders para definir que hacer con cada tipo de fichero.



### Module Bundlers

Por nombre del alumno, 21 de noviembre de 2019

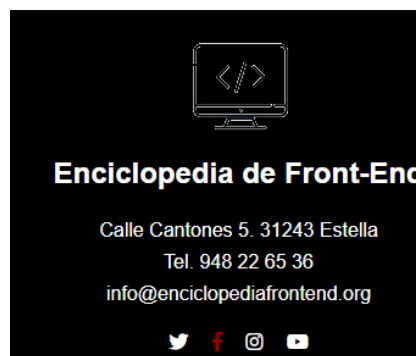


Profesora de ciclos formativos C.I. Politécnico de Estella de la rama de Informática. Profesora Grado Diseño Escuela de Artes de Corella. Diseñadora freelance. Cambiar por un pequeño texto que os describa.

Un module bundler nos permite automatizar una serie de tareas así como gestionar los assets de las dependencias que hemos instalado con npm si estamos utilizando ese gestor de paquetes. Parcel es un nuevo module bundler que promete unos tiempos de building ultrarápidos y zero config, por lo que se postula como un claro rival para la herramienta más usada actualmente: Webpack.



Parcel es un nuevo empaquetador de código que esta ganando popularidad y que puede ser muy útil en debido a su rapidez y características. La ventaja de la que más presumen en su web, y que lo convierten en un serio competidor es su velocidad. Otra de las ventajas más importantes que tiene es que no requiere de configuración, para los que llevamos años usando webpack esto es una agradable novedad, pues la configuración de los empaquetadores suele ser una tarea muy compleja y bastante delicada. El empaquetado de JS, CSS, HTML, archivos, etc es automático, sin que tengamos que añadirle plugins o loaders para definir que hacer con cada tipo de fichero.



# Página Lenguajes Front-End

- Mantendremos una imagen de portada, y colocaremos tres imágenes colocadas en fila que en pantallas de resoluciones más pequeñas pasarán a visualizarse en una sola columna.



Pantallas grandes 1



## Las bases del frontend

A pesar de que hay varios lenguajes que se usan en el frontend, nosotros nos fijaremos en tres elementos : HTML, CSS y JavaScript (aunque HTML y CSS no son lenguajes de programación propiamente dichos ).



## Las bases del frontend

A pesar de que hay varios lenguajes que se usan en el frontend, nosotros nos fijaremos en tres elementos : HTML, CSS y JavaScript (aunque HTML y CSS no son lenguajes de programación propiamente dichos ).



Recorte para pantallas grandes 1

Recorte para pantallas pequeñas 1

- Hay que añadir una imagen destacada representativa del contenido que explica la página. La imagen debe quedar correctamente integrada en el diseño, tanto en versiones para dispositivos móviles como en versiones para dispositivos con pantallas más grandes.
- Además, hay que añadir otras imágenes (como mínimo 2 más), a poder ser en diversos formatos (justificados adecuadamente a la documentación de la práctica). En el ejemplo os he puesto tres.
- alguna de las imágenes, ya sea la destacada como la del resto del contenido, debe incluir **gestión de la dirección de arte**. Hay que indicar en la documentación qué imágenes se han elegido, y justificar las decisiones sobre cómo modificar la dirección de arte cada tamaño de pantalla. Pensad que no solamente hacéis un sitio web, sino que estáis construyendo una parte de vuestro portfolio para enseñar a posibles empleadores. ¡Sed creativos y demostrad qué es lo que sabéis hacer!
- La web debe ser **responsive** y se debe visualizar correctamente desde cualquier dispositivo moderno (teléfono, tableta, ordenador ...). Como vimos en los foros de esta UT, esto implica que los contenidos multimedia (en especial imágenes) deben adecuarse al dispositivo que consulta la web: es importante que se adapten, como mínimo, al tamaño de la pantalla. De nuevo, y sabiendo que habitualmente queremos un diseño tan limpio como sea posible, intentad desarrollar una página que le permita enseñar lo que ha aprendido. Os vendrá muy bien los apuntes del video de Cristina.
- La web debe cumplir las reglas básicas [WCAG 2.1](#) de accesibilidad. En internet existen muchas herramientas que validan que estas reglas se hayan tenido en cuenta, como [AChecker](#). Encontrará más enlaces a la página del [W3C](#).

- **Muy importante (en especial de cara a esta entrega):** Ten en cuenta el uso de material protegido por derechos de autor, como se especifica en el último apartado de este documento. Recuerde cuáles son los [límites y excepciones](#) a la hora de utilizar materiales protegidos por derechos de autor.
- Además de estos requisitos mínimos, puedes ampliar tu proyecto como lo creas oportuno. Por ejemplo: ¿sabes qué es un Favicon y cómo se añade a un proyecto web?

## Recursos necesarios para realizar la Tarea.

- Ordenador.
- Ordenador con acceso a Internet y con herramientas adecuadas para manejar documentos HTML y CSS.
- Editor de imágenes Gimp es una opción más que razonable para este fin.

## Criterios de puntuación. Total 10 puntos.

- **Desarrollo (70%)**
  - Utilización de los formatos adecuados de los diversos archivos multimedia (15%).
  - Adecuación de estos recursos a versiones responsive (30%).
  - Diseñar y ejecutar un pequeño sitio web *responsive* (5%).
  - Adecuación a estándares y calidad del código en general (10%).
  - Publicar el repositorio en GitHub y realizar un deployment en Netlify. 10%
- **Documentación (30%)**
  - Documentación del proceso de desarrollo (15%).
  - Justificación de las decisiones tomadas en el desarrollo (10%).
  - Listado de las imágenes optimizadas y la mejora obtenida de peso (5%).

En la parte de **documentación**, será necesario que documentes todo el proceso de desarrollo de esta práctica, explicando los pasos que has seguido y los resultados obtenidos.

Tienes que documentar todo el proceso de construcción de cada página y las **decisiones que hayas tomado** y las problemáticas que hayas encontrado. También será necesario que cuentes cualquier decisión de

diseño y desarrollo que tomes (por ejemplo: ¿por qué has añadido un formato u otro, por que has decidido hacer la dirección de arte de ese modo?, qué elementos has tenido en cuenta para hacer la versión móvil/desktop de la web ...).

Ten en cuenta que, en un entorno profesional, la documentación es imprescindible para poder trabajar en equipo y en proyectos complejos. Un proyecto debe incluir toda aquella información indispensable para poder trabajar y para poder subirlo a producción. Y esta documentación debe ser exhaustiva pero, al mismo tiempo, concisa.

## **Formato y fecha de entrega**

Se entregará un pdf sin carpeta ni comprimido y allí los link a los repositorios de la práctica. **No se corregirán ejercicios que no respeten este formato.**

Se valorarán también los comentarios explicativos en el código.