Trabajaremos con un motor de plantillas, utilizaremos en nuestro caso handle bars, nos permite trabajar nuestros archivos html agregándole potentes herramientas, no tendrán que aprender ustedes una sintaxis desde cero todo lo que saben de html se aplica perfectamente y tendremos algunos agregados muy interesantes que nos van a permitir diseñar una plantilla base para no tener que escribir, por ejemplo, no sé el menú en cada una de las páginas, podremos hacerlo en la plantilla principal y luego se replicará en todas las vistas o podemos aplicar parciales, componentes especiales que podemos mostrar en algunas vistas si y en otras no, bien, de algunos conceptos muy interesantes que vamos a comenzar a desarrollar a partir de esta clase con nuestro motor de plantillas para el proyecto handle bars. Muy bien ahora vamos a hacer algo un poco más interesante aunque esto es importante ustedes tienen que entender el funcionamiento de las rutas, cómo vamos entregando al usuario distintas respuestas de acuerdo a la ruta que visita, ese es un concepto fundamental, por eso nos detuvimos en el. Ahora cómo hacer para mostrar el contenido que teníamos en nuestro front-end, por ejemplo, cuando hicimos la página de gatitos en este mismo contexto, para eso vamos a utilizar las vistas, las vistas son archivos html aunque aquí tendrán otra extensión porque vamos a usar un motor de plantillas, pero en definitiva siguen siendo archivos html, son más flexibles las vistas porque pueden recibir, pueden recibir datos, pueden ejecutar allí dentro cierto código javascript a través de funciones que se llaman helpers o ayudantes, bueno, ya lo veremos cuando llegue el momento, por ahora vamos a simplemente copiar todo nuestro proyecto y traerlo aquí, muy bien, vamos a comenzar utilizando un motor de plantillas, hay unos cuantos en el mercado, vamos a utilizar handle bars y handle bars tiene un par de implementaciones, podríamos usar hbs, handle bars a secas, en este caso vamos a usar express handle bars, muy bien, si quieren revisar documentación sobre handle bars y cómo funciona un motor de plantillas simplemente pueden buscar aquí en Google handle bars y aquí tienen la página oficial con algunas notas sobre su uso y una descripción bastante detallada de cada uno de los componentes, ténganlo presente, siempre es bueno repasar la documentación oficial de cualquier paquete que usemos, pero tranquilos tranquila no lo voy a hacer aprender por vuestra cuenta, aquí les vamos a enseñar como, recuerden de todas maneras que los paquetes tienen muchas más cosas que las que podemos ver en un curso, por lo tanto bueno es importante que revisen la documentación, ahora, como instalamos, como ponemos en uso un motor de plantillas, bueno, lo primero es cerrar aquí abajo, estoy en la terminal, controlC, voy a cerrar el proceso y vamos a instalar con lpm y express handle bars, ok, esto nos instala el paquete y ahora tenemos que hacer unas mínimas configuraciones para que este paquete, bueno funcione aquí dentro, podemos comenzar por ejemplo por la estructura de carpetas express handle bars va a necesitar una carpeta llamada vistas en inglés views, views, dentro de la carpeta views va a necesitar una carpeta llamada layouts y podría requerir, está no es mandatoria pero podría requerirla por lo tanto no nos cuesta nada hacerla, una carpeta llamada parciales, bien, aquí tenemos la carpeta vistas, con la carpeta layouts y la carpeta parciales, aquí dentro del layout es requerido un archivo con el nombre main y la extensión hbs, muy bien, por ahora no tendrá nada, no importa, esta es la estructura de carpetas que nos va a pedir para funcionar handel bars, vamos a hacer la configuración para poder utilizar, para montar en nuestro sistema express handel bars, aquí vamos a comentar entonces configuración de express handel bars y vamos a copiar cuatro o cinco comandos que

son los que nos indica la documentación, pero primero que hay que hacer cuando queremos utilizar una dependencia, un paquete en un proyecto de node, lo mismo que hicimos con express ergo, es decir, importarlo, requerir ese paquete, vamos a crear una constante, const, le pondremos hbs que es más corto podríamos ponerle aquí handel bars pero me resulta más corto así y esto va a ser igual a requiere, es decir, va a requerir, va a importar el paquete y el paquete se llama express handel bars, aquí express handel bars, correcto muy bien, ahora sí ya podemos usar ese paquete express handel bars que hemos consolidado en esta variable, en esta constante que se llama hbs, bien, por un tema de bueno que es más corto, entonces diremos que nuestra aplicación va a usar un motor de plantillas engine y aquí diremos que la extensión .hbs y ahora indicaremos que el motor de hbs utilice la extensión text name, esto es una configuración interna verdad, hbs, porque hacemos esto, porque si no hago esto voy a estar obligado a ponerle a los archivos handel bars la extensión handel bars porque la extensión por defecto y no quiero escribir tanto, es un poco es muy largo escribir handel bars como extensión, por lo tanto si quiero escribir hbs o aquí podría poner pepe verdad, pero no tiene mucho sentido, pero podría hacerlo, no importa, vamos a decir hbs pero esta configuración como es una configuración distinta a la original tenemos que aclararla aquí, por lo tanto engine vamos a pasarle un objeto con un parámetro, el parámetro ex name o extensión name o nombre de extensión, hbs, muy bien, luego vamos a hacer el siguiente seteo, vamos a decirle a nuestra app que setee el motor de vistas view, engine, recuerden las vistas son para la parte del front end,, para lo que ve el usuario el resto lo haremos en el back-end para eso necesitamos las vistas y aquí le vamos a decir que se trata de hbs, aquí cuando decimos hbs nos estamos refiriendo a express handle bars que lo hemos importado y metido en esta constante, muy bien, y nos faltan dos configuraciones más, vamos a decir dónde están las vistas, para eso escribiremos add.send, views y vamos a indicar dónde están las vistas aquí, en qué carpeta, views y aquí tendríamos que poner punto y barra y views, views y por último vamos a indicar la estructura de directorios para que pueda llegar a esta carpeta, ok, aquí le indicamos dónde van a estar las vistas pero tenemos que indicarle además la ruta, el camino específico para poder llegar allí, entonces vamos a indicar views, ahora sí y esto será igual al __dir name, ok, y aquí vamos a concatenar con el nombre de la carpeta, muy bien, este es un procedimiento rutinario, esta información ustedes pueden la información de configuración, pues la pueden encontrar fácilmente aquí, nosotros la tomamos precisamente de la documentación aquí en npm express handel bars, bien aquí podrán ver todos estos comandos de dónde de dónde han salido no crean ustedes que tienen que aprenderse de memoria estas cuestiones que parecen muy difíciles, para nada, son instrucciones que nos da el propio fabricante del paquete sobre cómo hacer las configuraciones, ok, cómo podemos hacer las configuraciones de este de esta herramienta, muy bien, perfecto, ok, vamos a proseguir entonces, ahora que tenemos configurado esto vamos a indicar dónde estará nuestra carpeta de archivos estáticos, los archivos estáticos pueden ser archivos de imágenes, vídeos, archivos CSS, que estarán bueno, ubicados siempre en un mismo sitio, para esto tenemos que indicarle a nuestra aplicación dónde estarán esos archivos estáticos que habitualmente se ponen en una carpeta llamada public, no la hemos creado aún, no importa podemos hacerlo ahora o después, el orden en este caso no importa demasiado, ya tenemos una carpeta public por supuesto está vacía, aquí pondremos por ejemplo aquí adentro podríamos crear una carpeta images, podríamos crear una carpeta CSS,

bueno, y otras si las necesitásemos, bien, entonces ahora vamos a indicar dónde está esa carpeta de archivos estáticos, por lo tanto, utilizaremos una funcionalidad del propio express, diremos que nuestra aplicación va a usar, va a hacer uso de express y aquí el método static, que recibe como parámetro el nombre de la carpeta donde están los archivos estáticos y cuál es el nombre, bueno pues acabamos de crearlo aquí public esa nuestra carpeta donde irá a buscar toda esa información, por ahora no haremos más configuraciones, vamos a comenzar a copiarnos nuestro proyecto html para tener control de vistas, es decir, quiero que cuando visiten aquí 3000, bueno, ahora el servidor no está corriendo así que va a dar un error, vamos a volver a correrlo recuerden npm, run, div, quiero que cuando visiten está aquí, no diga hola mundo que nos muestra en nuestra página de gatitos, por ejemplo, eso sería deseable, muy bien, entonces ahora vamos a hacer lo siguiente, vamos a ver dónde tenemos nuestro proyecto, lo tenemos aquí, front-end, aquí tenemos por ejemplo el archivo index, podríamos comenzar por él, vamos simplemente a copiarlo, vamos a copiar nuestro archivo index y vamos a ponerlo aquí dentro en layout, ok, aquí está nuestro archivo index, vamos a examinarlo, queremos que todo esto todo este archivo, vamos a copiar todo, o simplemente podríamos borrar main, es lo mismo y renombrar este archivo como main.hbs, ok, cada vez que se cargue en nuestra ruta principal nuestro sistema de plantilla, nuestro motor va a buscar el archivo main y va a renderizarlo, va a mostrar todo lo que esté en el archivo main, bien, nuestro archivo main va a ser la base para todos los demás, las demás vistas que tengan nuestro proyecto, pero por lo general no ponemos toda esta información en el archivo main, que solemos poner allí, bueno, aquella información que se va a repetir en cada una de las vistas, por ejemplo, cada uno de nuestros archivos de nuestros archivos html tiene que tener el doc type, no necesitamos con handel bars escribirlo en cada uno de los archivos, lo escribimos en el main y cada una de las vistas luego tendrá esa información, también toda esta información y por ejemplo si deseamos que el menú esté en todas las vistas podemos dejarlo aquí, bien, hasta ahí y todo lo demás podríamos quitarlo, es decir, nuestra sección de noticias todo esto podríamos quitarlo, vamos a quitar todo, vamos a dejar hasta este div que es el cierre de nuestro grid container, esto lo borro de aquí, simplemente lo he cortado no lo he eliminado para poder luego pegarnos, muy bien y ahora en nuestro proyecto general anterior en el front end teníamos un archivo index html aquí no tendremos html tendremos hbs como ya vieron en el main del layout, pero es exactamente lo mismo, solo cambia la extensión, por lo tanto en vistas vamos a crear un archivo que se llame index, el nombre es indistinto verdad, yo lo voy a poner index hbs y aquí voy a copiar el resto, lo que cortamos del archivo anterior, ok, aquí tenemos main, etcétera, etcétera, footer, estas secciones estaban sin terminar, quedará igual sin terminar por ahora, bueno, ya lo tenemos, ahora vamos a ver cómo se comporta nuestra aplicación sigue corriendo como ustedes pueden ver acá, por lo tanto si vamos aquí y le damos aquí actualizar, bueno, veremos que por ahora no hay ningún cambio, no debería haberlo ya que aquí nosotros seguimos indicando que cuando visiten la ruta raíz nos diga “hola mundo”, vamos a cambiar eso, no haremos más un send que envía información a la pantalla, haremos un render, el render lo que hace es renderizar una vista, ok, y hasta ahora tenemos solamente esta la vista index hbs, vamos a renderizarla entre comillas ponemos el nombre “index” para que esa vista efectivamente se vea de manera correcta aquí podemos actualizar, ok, lo que tenemos que hacer es insertarla dentro de main, noten que aquí lo que se está renderizando aunque sin los estilos es el navbar solamente se ha

lanzado lo que teníamos en nuestro layout principal, bien, y esto va a funcionar así, aquí dentro del contenedor grid, para que mantenga los estilos de grid que le dimos, vamos a insertar con triple llaves la palabra body que va a hacer esto, magia, cada vez que corramos nuestro proyecto se mostrará la vista main siempre y luego de acuerdo a la ruta que estemos visitando, en este caso será index se va a renderizar el archivo que indiquemos, en este caso se va a realizar index que tiene todo el resto del contenido y donde se va a insertar, donde se va a mostrar, aquí en el body, por lo tanto siempre tendremos cargada esta información sin necesidad de repetirla en cada archivo, fíjense que este index no tiene doc type, no tiene nada, no tiene nada, simplemente comienza directamente en la etiqueta semántica main, no necesitamos repetir toda esa información porque siempre se va a cargar primero main que tiene nuestras fuentes, nuestros estilos e inclusive nuestro menú y aquí dentro luego se inyectará cada página que vayamos visitando, vamos a ver como esto se refleja en nuestra página. Vamos a ver aquí refrescar y fíjense que ahora se insertó el resto del contenido de manera dinámica, bien, ustedes me dirán pero esto es muy distinto a esto, bueno, porque ahora tenemos que hacer unas leves modificaciones en las rutas que hemos indicado, recuerden que aquí se estaba refiriendo a unas rutas, que aquí dentro de nuestro proyecto de node son levemente distintas, son han cambiado un poco, ok, aquí por ejemplo podríamos indicar en nuestro main que ya no tiene que ir a buscar el CSS a esta ruta con punto y barra porque como está en la carpeta public todo lo que está en la carpeta public tiene como características de absoluto, simplemente pondremos CSS styles y nuestro sistema sabrá que tiene que ir a la carpeta CSS que está dentro de public a buscar eso, veamos si efectivamente hay algún cambio, aquí en nuestras vistas, este no es, este es, el que se carga desde local host, vamos a dar enter, bueno, por ahora no tenemos ningún cambio pero vamos a ver si hemos inclusive guardado nuestro archivo que eso también es muy importante, aquí, vamos a ver si tenemos guardados los cambios, claro lo que no tenemos es el archivo CSS dentro de dentro de la carpeta tengo que copiarmelo de nuestro proyecto anterior, del proyecto que hicimos para el front end, recuerdan, vamos a copiar el archivo styles dentro de la carpeta correspondiente CSS, ahora, aquí vamos a volver a lanzar nuestro servicio con rs y aquí refrescamos y ya tenemos algunos estilos, muy bien, nos faltan imágenes y algunas cosas pero fíjense que va respondiendo, luego nos faltaría también por ejemplo copiar todas nuestras imágenes que aquí las tenemos en access , images, vamos a copiar directamente la carpeta images o mejor dicho su contenido porque ya creamos aquí una carpeta de images en public, vamos a llevarla allí, bien, pero también tenemos que actualizar la carga de esas imágenes, por ejemplo fíjense que aquí estamos cargando una imagen, dónde está nuestro archivo, está aquí el mail, ampliemos para que sea más cómodo, fíjense que estamos cargando cosas de access y esto ya no es correcto, simplemente vamos a decir de images sin la barra ahí tendría que cambiarnos el icono, si esto funcionará tendría que habernos puesto aquí el icono de los gatitos y aquí arriba está, fíjense cómo va reaccionando nuestro proyecto a los cambios que hacemos, son simplemente cambios de ruta, decir ok ahora esto ha cambiado levemente, los archivos están en otro sitio, si venimos a index noten que todas las imágenes se cargan desde ./acces-images, tenemos que cambiar todas estas, vamos a ir eliminándolas, ahí guardamos, actualizamos y comienzan a aparecer aquí está el autor del artículo, puedo cambiarlas una por una o controlF y voy a decir que quiero buscar .access/ y aquí decirle que esto lo quiero reemplazar por nada, o sea, voy a borrar esta línea y solo me

quedara images y el nombre de los archivos, se entiende, y ahora guardo, actualizo y boom ya tenemos todas nuestras imágenes y nuestro sitio, bueno, está respondiendo, solamente faltan los gatitos acá para eso tengo que ir y modificar nuestro archivo CSS porque eso lo estamos cargando si no me equivoco en un contenedor aquí, en nuestros estilos, claro, estamos indicando que tiene que ir a ../access y ya no existe más la carpeta access, no en este contexto del proyecto de node, así que actualizó y boom, listo, fíjense que rápidamente hemos adaptado nuestro proyecto con archivos html tradicionales a nuestro entorno que maneja el front end pero también el backend seguimos con más en la próxima clase.