

SESIÓN 2:

MÓDULO *SENYALS_UC* –

EL FLIP-FLOP D EN QUARTUS

Esta práctica tiene dos partes diferenciadas. En la primera parte (que es la principal) se aplicarán los conocimientos adquiridos hasta el momento para entrar y simular el módulo **Senyals_UC** que, en función de los pisos en los que hay una solicitud/llamada al ascensor (*Pis_demanat[7..0]*) y del piso en el que se encuentra el ascensor (*Pis_actual[7..0]*), indica a la Unidad de Control si el piso actual está solicitado (salida *P*), y si quedan pisos solicitados por arriba (salida *PP_amunt*) o por debajo (salida *PP_aval*) del piso en el que el ascensor está en estos momentos.

En la segunda parte, muy sencilla, se hará una celda consistente simplemente en un flip-flop D de la biblioteca de Quartus y se simulará. El objetivo de esta segunda parte es conocer detalladamente las características de los flip-flops D de Quartus (algo distintos a los de Logisim-VerilUOC_Desktop), los cuales serán usados en los circuitos secuenciales del controlador del ascensor que se desarrollarán a partir de la tercera sesión de prácticas.

Aquellos equipos que vayan a trabajar en un ordenador del laboratorio, antes de iniciar la sesión tienen que:

- descargar de la nube la carpeta del proyecto comprimida y moverla a **U:\FCPract**
- una vez está la carpeta comprimida en **U:\FCPract**, descomprimir la carpeta
- borrar la carpeta comprimida

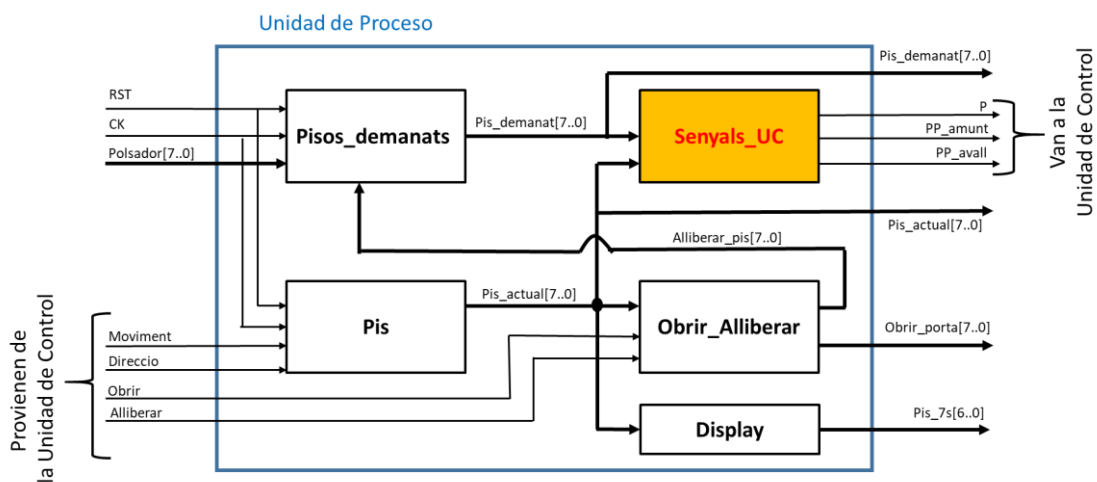
1 MÓDULO *SENYALS_UC*

1.1 DESCRIPCIÓN

Tal como se comentaba en la introducción, este módulo tiene como objetivo informar a la Unidad de Control si hay solicitudes/llamadas al ascensor en:

- a) el piso en que se encuentra el ascensor,
- b) los pisos superiores al que se encuentra en el ascensor
- c) los pisos inferiores al que se encuentra el ascensor

El módulo **Senyals_UC** genera las señales *P*, *PP_amunt* y *PP_avall* a partir del bus *Pis_actual[7..0]*, procedente del módulo **Pis**, y del bus *Pis_demanat[7..0]*, procedente del módulo **Pisos_demanats**.



Veamos con detenimiento cuáles son las entradas y salidas del módulo **Senyals_UC** y su finalidad.



Estas son sus entradas:

- El bus *Pis_demanat[7..0]* indica en qué pisos hay solicitudes al ascensor. Cada bit del bus está asociado a un piso y si vale 1 significa que hay una solicitud al ascensor en ese piso. Por ejemplo, si *Pis_demanat[7..0]* es 11010010, eso significa que hay solicitudes en los pisos 1, 4, 6 y 7 todavía no atendidas.
- El bus *Pis_actual[7..0]* indica el piso en el que se encuentra el ascensor. Cada bit del bus está asociado a un piso y si vale 1 significa que el ascensor se encuentra en ese piso. Por ejemplo, si *Pis_actual[7..0]* es 00000001 eso significa que el ascensor está en el piso cero (la planta baja). Evidentemente, solo uno de los bits de *Pis_actual[7..0]* vale 1.

Las salidas que se han de generar son las siguientes:

- P : Indica si el piso en el que está actualmente el ascensor es uno de los pisos solicitados. Es decir, $P=1$ si $Pis_actual[j] = 1$ y $Pis_demanat[j] = 1$.
- PP_amunt : Indica si hay algún piso solicitado por encima del piso en el que está actualmente el ascensor. Es decir, $PP_amunt=1$ si $Pis_actual[j] = 1$ y para algún valor de $k>j$, $Pis_demanat[k] = 1$.
- PP_avall : Indica si hay algún piso solicitado por debajo del piso en el que está actualmente el ascensor. Es decir, $PP_avall=1$ si $Pis_actual[j] = 1$ y para algún valor de $k<j$, $Pis_demanat[k] = 1$.

Más formalmente,

$$P = \sum_{j=0}^7 (Pis_actual(j) \cdot Pis_demanat(j))$$
$$PP_amunt = \sum_{j=0}^6 Pis_actual(j) \cdot \left[\sum_{k=j+1}^7 Pis_demanat(k) \right]$$
$$PP_avall = \sum_{j=1}^7 Pis_actual(j) \cdot \left[\sum_{k=0}^{j-1} Pis_demanat(k) \right]$$

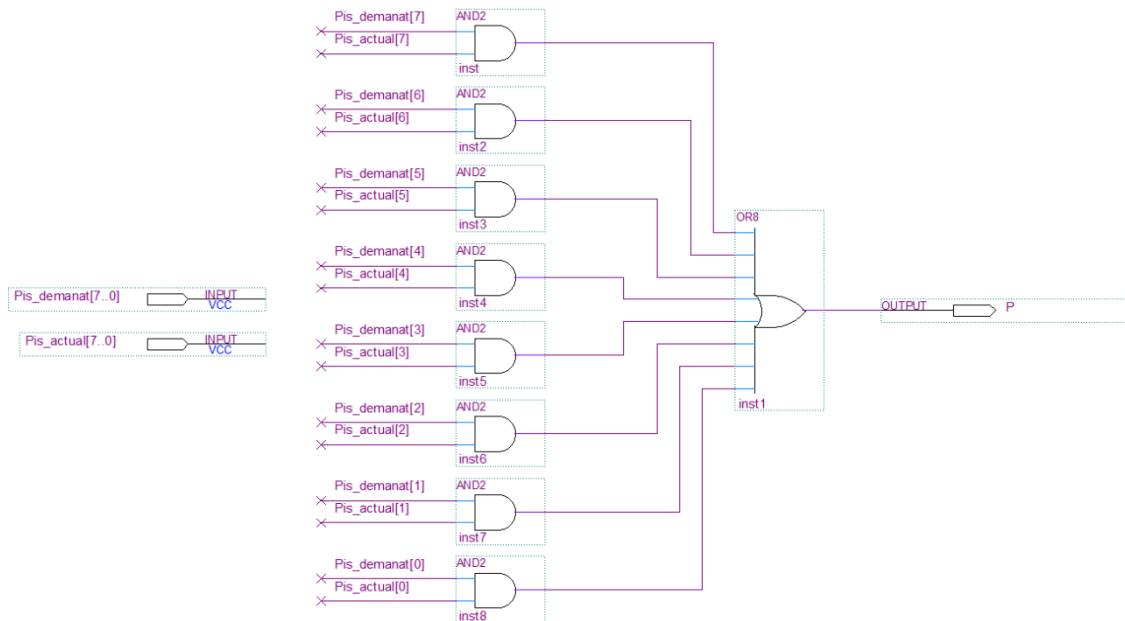
Nota: Los sumatorios indican sumas lógicas, es decir, operaciones OR.

1.2 ENTRADA DEL ESQUEMÁTICO DE SENYALS_UC

En primer lugar, entra en el proyecto que definiste en la anterior sesión: abre Quartus v13.0, clicas en **File** → **Open Project...** y busca la carpeta vinculada al proyecto (**U:\FCPract\4XX_YY** [o **C:\Usuarios\nombre_de_tu_cuenta\FCPract\4XX_YY** en caso de que uses tu propio ordenador]). En ella hay un fichero con el nombre del proyecto (**ControlAscensor.qpf**, de tipo “Quartus II Project File”). Selecciona ese fichero y pulsa el botón **Abrir** para abrir el proyecto.

Para construir el módulo **Senyals_UC** crearemos por separado tres circuitos que nos generen respectivamente las señales P , PP_amunt y PP_avall , y luego los agruparemos.

La construcción del circuito que genera P es muy sencilla; basta con dibujar el esquemático de la página siguiente.



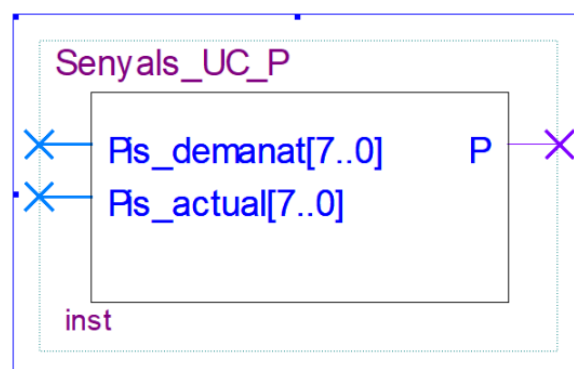
Esquemático del submódulo **Senyals_UC_P** (genera la señal *P*)

Una vez entrado el circuito, guárdalo con el nombre **Senyals_UC_P** en la carpeta del proyecto (U:\FCPract\4XX_YY) y compílalo para detectar y eliminar errores. **Antes de compilar**, recuerda definir el módulo con el que estás trabajando como *Top-level entity* para que Quartus sepa que es éste el esquemático que quieres compilar y no otro.

Una vez hayas entrado y compilado el esquemático **Senyals_UC_P**, construye un símbolo para que, más adelante, puedas utilizar este módulo como si fuese una celda de biblioteca. Para ello basta con clicar en **File → Create/Update → Create Symbol Files for Current File**. Dale el nombre **Senyals_UC_P**, y clicla en **Guardar**. El símbolo ya se ha creado. Si quieres verlo, abre el fichero **Senyals_UC_P.bsf**, ejecutando el comando

File → Open → Indica como Tipo "Graphic Files" → selecciona el fichero Senyals_UC_P.bsf

En la ventana que se abre se muestra el símbolo creado.



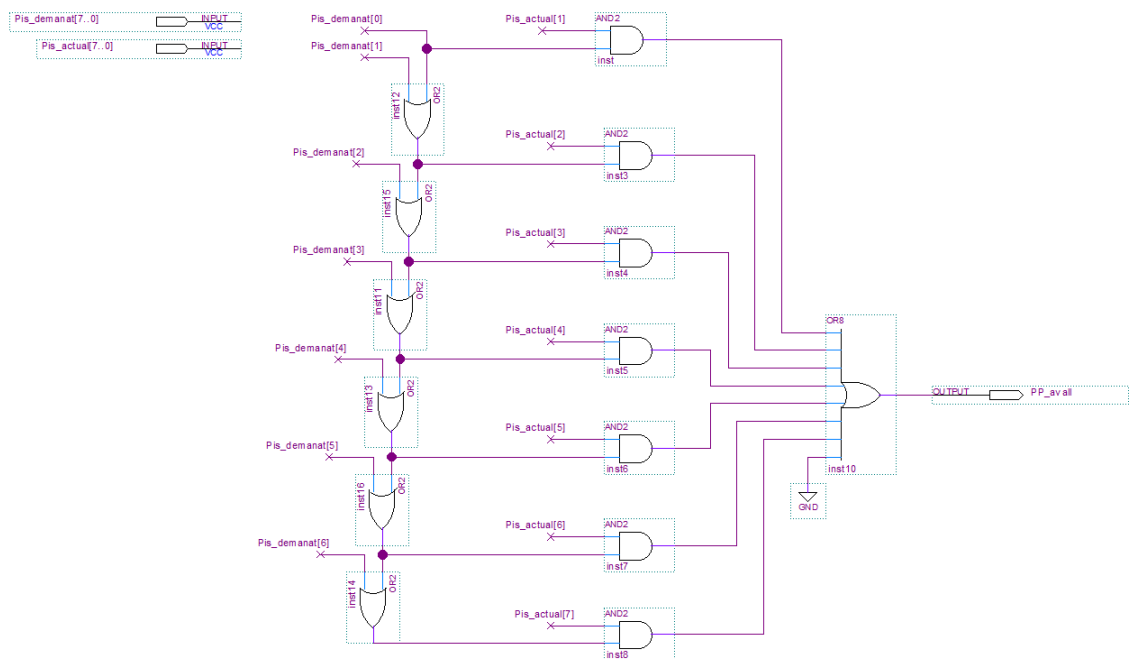
Crear el esquemático de los módulos que generan las señales PP_amunt y de PP_avall es más laborioso. Te explicamos más abajo cómo sería el esquema del módulo que genera PP_avall . El módulo que genera PP_amunt es análogo.

Para agilizar la sesión de prácticas, los esquemáticos de los módulos que generan PP_amunt y de PP_avall (**Senyals_UC_PP_amunt** y **Senyals_UC_PP_avall**) te los damos hechos, pero lee y entiende cómo funcionan porque te será indispensable para realizar la simulación del circuito **Senyals_UC**. Los esquemáticos están en el campus virtual, en la sección **Pràctiques → Enunciats → carpeta “Material sesión 2”**.

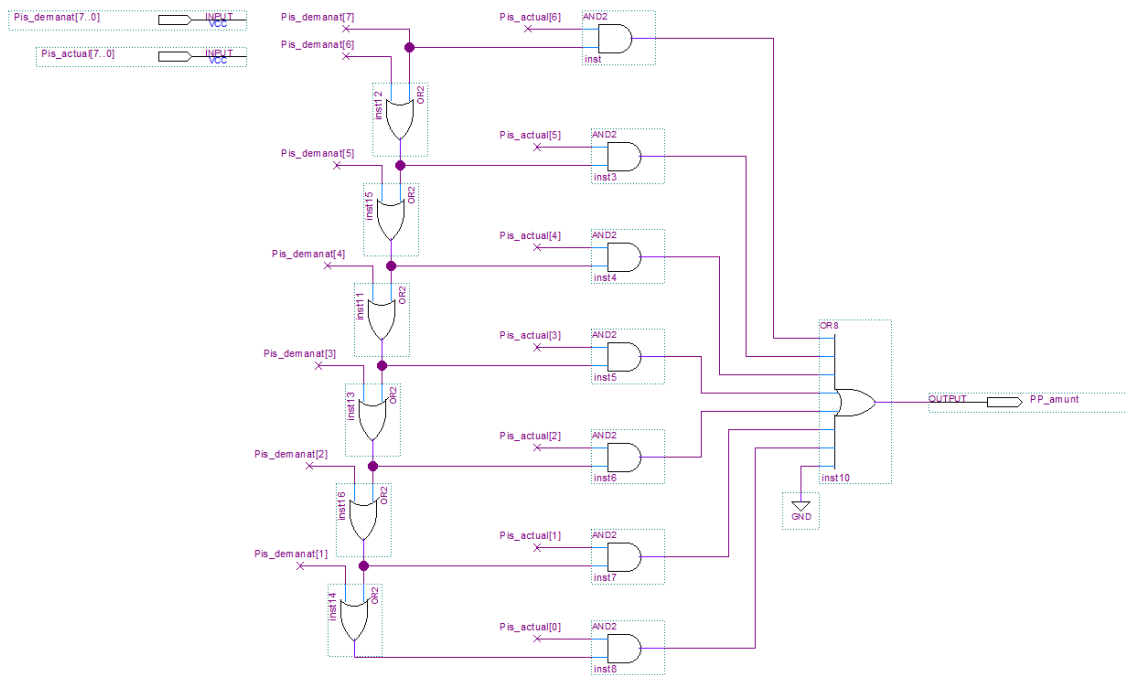
PP_avall debe tomar el valor 1 cuando está solicitado algún piso por debajo del actual; es decir, ...

- Si el ascensor está en el piso 0 $\Rightarrow PP_avall=0$ porque no puede haber ningún piso por debajo seleccionado.
- Si el ascensor está en el piso 1 $\Rightarrow PP_avall=1$ si $Pis_demanat[0]=1$, y 0 en caso contrario.
- Si el ascensor está en el piso 2 $\Rightarrow PP_avall=1$ si $Pis_demanat[0] + Pis_demanat[1]=1$, y 0 en caso contrario.
- Si el ascensor está en el piso 3 \Rightarrow
 $PP_avall = 1$ si $Pis_demanat[0] + Pis_demanat[1] + Pis_demanat[2] = 1$, y 0 en caso contrario.
- etc.

En la siguiente imagen se muestra el esquemático del submódulo **Senyals_UC_PP_avall**, el cual genera la señal PP_avall .



Esquemático del submódulo **Senyals_UC_PP_avall**



Esquemático del submódulo **Senyals_UC_PP_amunt**

De una manera análoga se generaría la señal **PP_amunt**. En la imagen previa se muestra el submódulo **Senyals_UC_PP_amunt** el cual genera la señal **PP_amunt**.

A partir de este momento:



- Añade los ficheros **Senyals_UC_PP_avall.bdf** y **Senyals_UC_PP_amunt.bdf** a tu proyecto. Para ello basta con que arrastres cada fichero .bdf que te has bajado del CV al área de trabajo del editor de esquemas y, a continuación, lo guardes en la carpeta del proyecto Quartus, ejecutando:

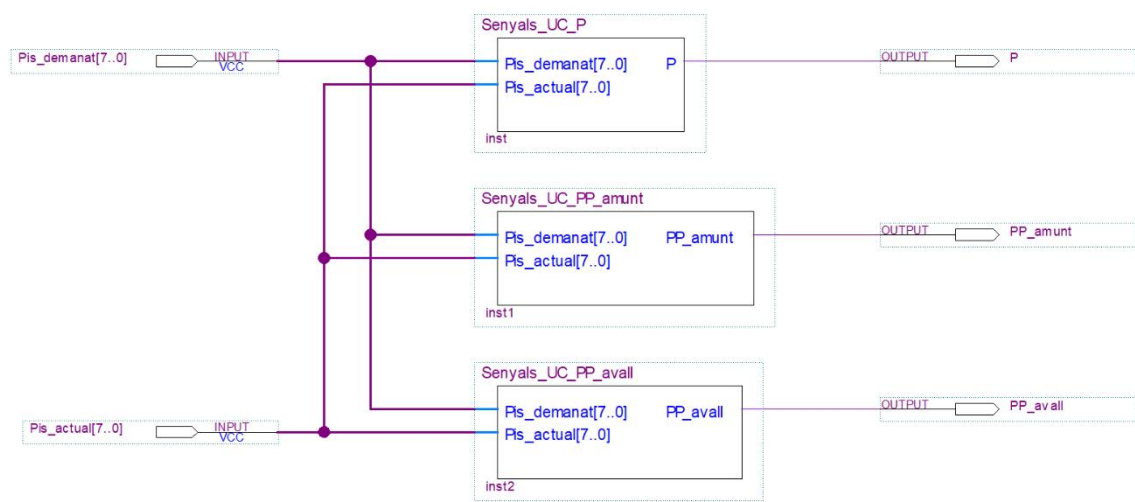
File → Save As...

En el formulario que se abre debes seleccionar **la carpeta del proyecto (U:\FCPract\4XX_YY)**.

- Crea su símbolo para cada uno de los dos módulos como has hecho con el circuito **Senyals_UC_P**.
- Y construye el circuito completo, **Senyals_UC**, como te explicamos a continuación ...

En el editor de esquemas abre un esquemático nuevo, y con los símbolos de los tres módulos creados realiza el esquema que se muestra en la siguiente figura.

Hay un aspecto que has de tener en cuenta. Como ya sabes, para dibujar las líneas de conexión de un bit has de usar el botón con el símbolo  (*Orthogonal Node Tool*). Sin embargo, para dibujar las líneas de conexión consistentes en un bus has de usar el botón  (*Orthogonal Bus Tool*). Como se puede observar en la imagen, esas líneas presentan un grosor mayor.



Esquemático del módulo **Senyals_UC**

A continuación, guarda el circuito resultante en un fichero con el nombre **Senyals_UC.bdf** en la carpeta del proyecto (U:\FCPract\4XX_YY) y genera un símbolo para él (**Senyals_UC.bsf**).

Finalmente, define el módulo **Senyals_UC** como *top-level entity* y realiza su compilación. Asegúrate de que no se producen errores en la compilación y que solo genera 9 *warnings* (estos son *warnings* técnicos e informativos).

1.3 SIMULAR EL FUNCIONAMIENTO DEL MÓDULO *SENYALS_UC*

Los pasos que se siguen para realizar la simulación son análogos a los que seguiste para simular el módulo *Obrir_Alliberar* en la Sesión 1. Por ello no son explicados de forma detallada. Si tienes alguna duda consulta la parte del guion de la sesión 1 donde se explica detalladamente la simulación del módulo *Obrir_Alliberar*.

1.3.1 CREACIÓN Y EDICIÓN DE LOS VECTORES DE SIMULACIÓN

Como ya se vio en la sesión 1 de prácticas, lo primero que hay que hacer cuando queremos simular un esquemático es generar el llamado “fichero de vectores de simulación” (también llamado “fichero de estímulos” o “fichero de formas de onda”).

Se empezará creando el fichero de vectores de simulación. Para ello, ejecuta la orden:

File → New

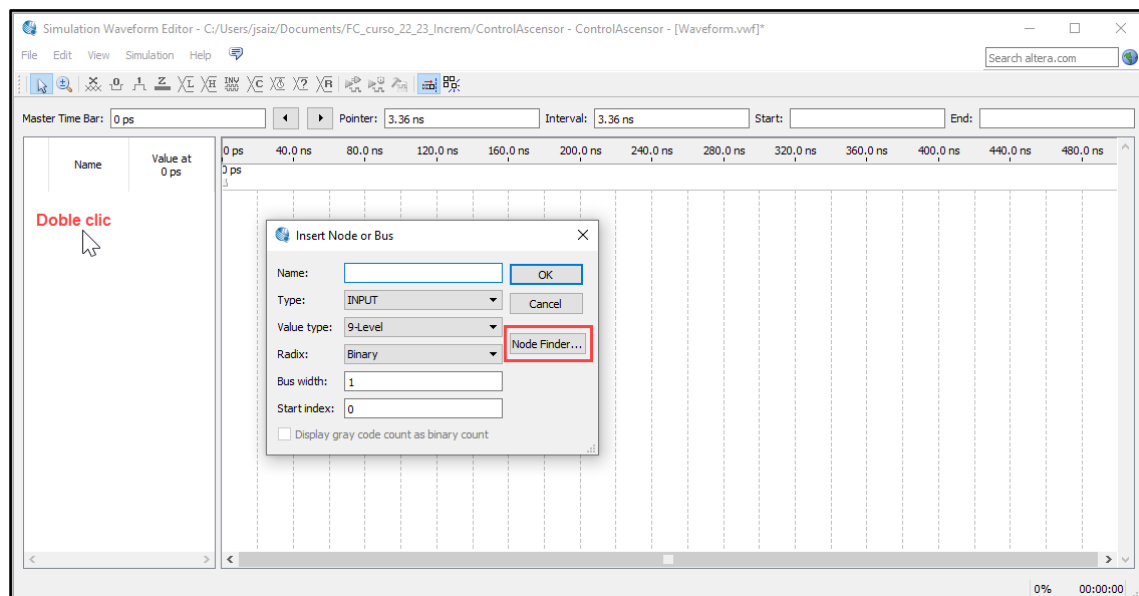
En el menú desplegable que aparece, selecciona **University Program VWF** y clics en **OK**.

Establece la anchura de las franjas temporales del editor de formas de onda (20 ns) y la duración de la simulación (500 ns):

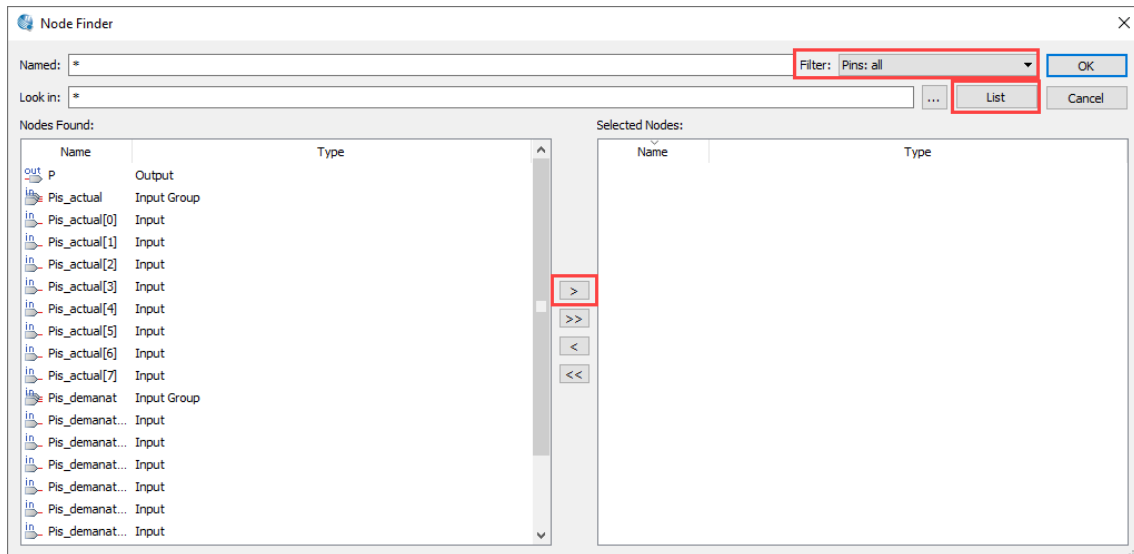
Edit → Grid size... → 20 ns

Edit → Set End Time... → 500 ns

Para introducir las entradas y sus valores, haz doble clic debajo de la columna *Name*. En el formulario que aparece, clics en el botón **Node Finder**.



Se abrirá una nueva ventana desde donde podrás seleccionar los nodos de entrada (*Pis_actual* y *Pis_demanat*) y los nodos de salida (*P*, *PP_amunt* y *PP_avall*):



Selecciona la opción “**Pins: all**” en **Filter**, y clicas en “**List**”. En la sub-ventana “**Nodes Found**” aparecerán todos los nodos de entrada y salida de **Senyals_UC**. Selecciona el nodo *Pis_actual* y clicas en **>**. Repite el mismo proceso para los nodos *Pis_demanat*, *P*, *PP_amunt* y *PP_avall*. Finalmente clicas en **OK**.

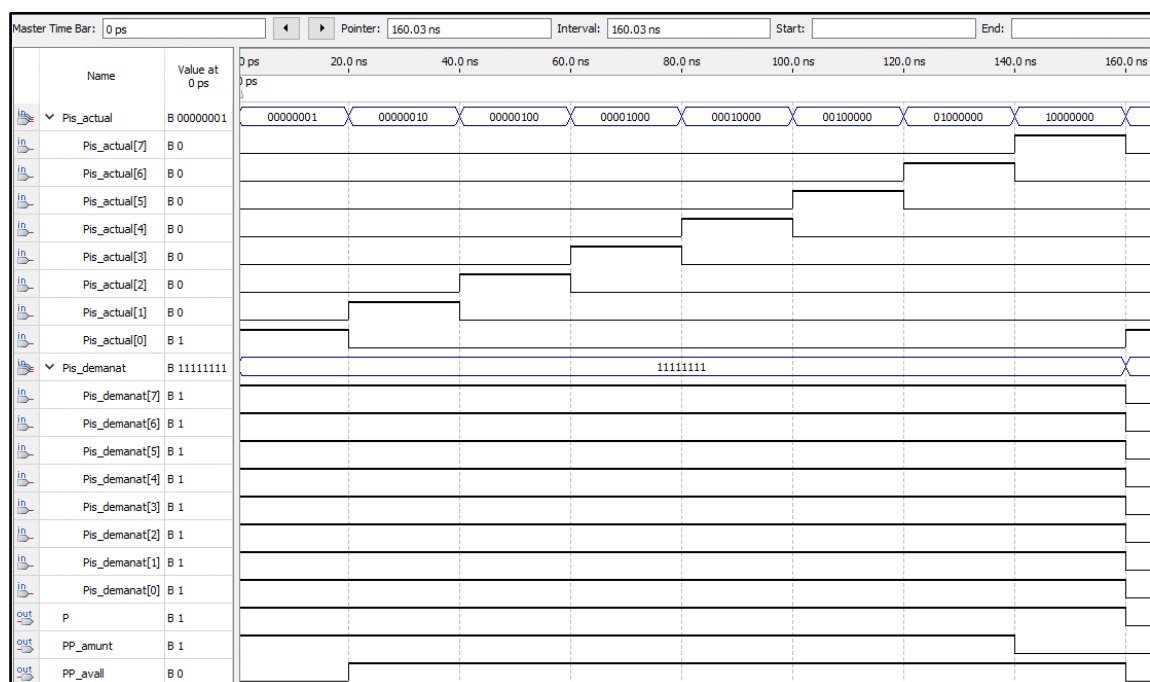
Te volverá a aparecer la ventana **Insert Node or Bus**. En el campo **Radix** selecciona **Binary** (de esa forma el valor de los buses se mostrará en binario en la ventana de simulación) y vuelve a clicar en **OK**. Los nodos se incorporarán a la ventana de simulación. En ella verás todos los nodos de entrada y de salida, en este orden. En general, es recomendable poner siempre en primer lugar los nodos de entrada y después los nodos de salida ya que ello facilitará la legibilidad del resultado de la simulación.

Lo siguiente que hemos de hacer es definir el valor de las entradas (*Pis_actual* y *Pis_demanat*) en cada instante de tiempo. Pero antes hay que establecer una estrategia de test.

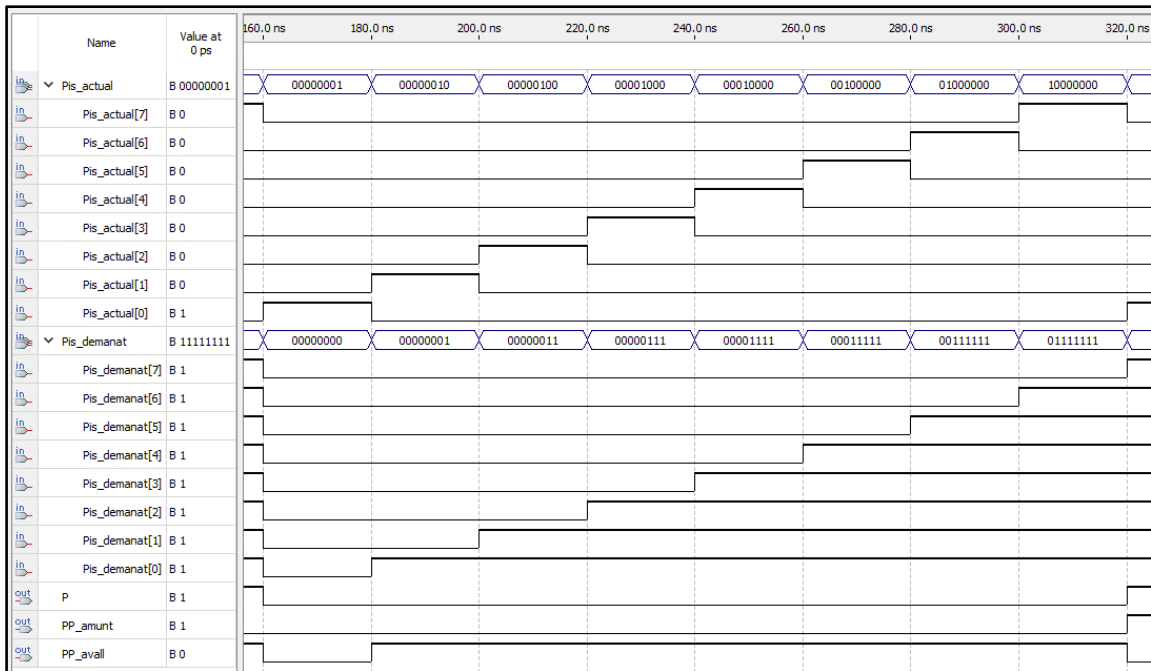
Estrategia de test

Para comprobar que el circuito funciona correctamente haremos lo siguiente:

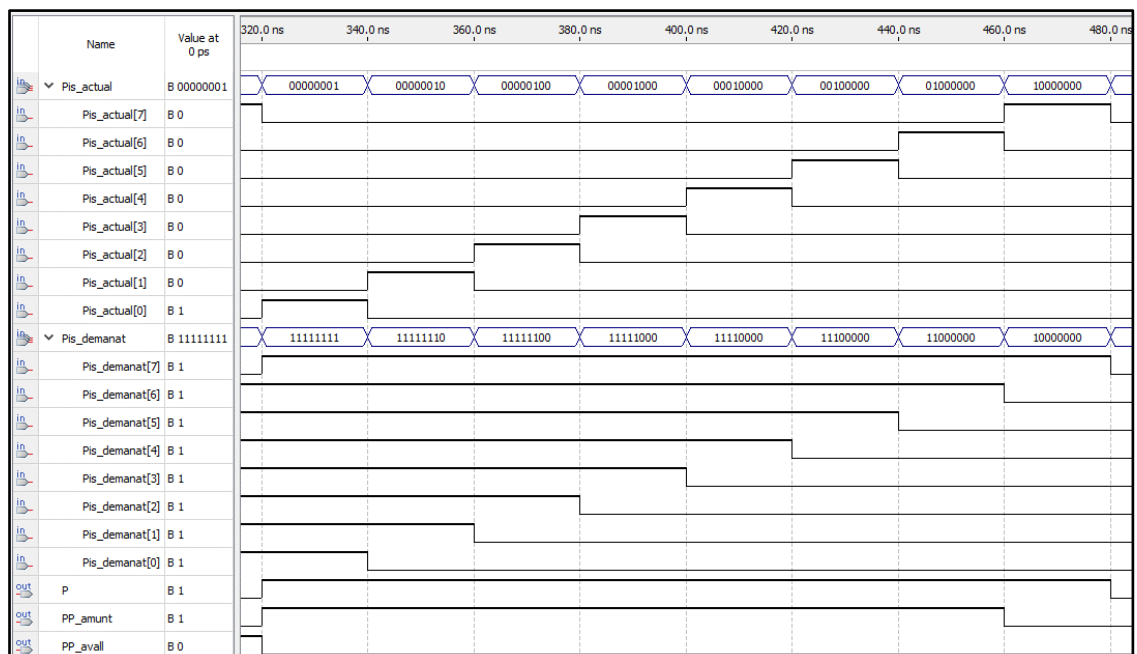
- En primer lugar supondremos que todos los pisos están solicitados ($Pis_demanat[7..0]=11111111$) e iremos suponiendo que el ascensor está primero en el piso 0, luego en el 1, luego en el 2, y así sucesivamente. Lo que hemos de esperar en este caso es que la señal P siempre esté a 1, puesto que el piso actual siempre está solicitado, la señal PP_amunt siempre esté a 1 salvo cuando el piso actual sea el 7 ya que es el último piso y, por tanto, no puede haber pisos solicitados por encima de él, y PP_avall siempre está a 1 salvo cuando el piso actual es el 0, puesto que no hay pisos solicitados por debajo del 0. La simulación **parcial** sería algo como lo siguiente:



- A continuación, supondremos de nuevo que estamos pasando sucesivamente por todos los pisos y que, cuando estamos en el piso j , todos los pisos por debajo del mismo están solicitados. Lo esperable en este caso es que las señales P y PP_amunt estén siempre a 0, y la señal PP_avall esté siempre a 1 salvo cuando el piso actual es el 0:



- Finalmente comprobaremos el caso opuesto en el que, pasando de nuevo por todos los pisos, cuando estemos en el piso j , tanto el piso j como todos los que están por encima del mismo estén solicitados. P deberá ser siempre 1 puesto que el piso actual siempre está solicitado, PP_avall deberá mantenerse a 0 y PP_amunt deberá estar a 1 salvo cuando el piso actual sea el 7:



Con estas comprobaciones nos conformaremos. Es decir, creemos razonablemente que, si con estos valores de entrada las salidas que genera el circuito son las que esperábamos, el circuito funciona correctamente. **Fíjate en la barra de tiempos; no es que hagamos 3 simulaciones, sino que los tres casos los simulamos uno detrás del otro.**

Ahora que ya hemos establecido la estrategia de test que nos permitirá determinar si el circuito se comporta correctamente, hemos de introducir los valores en los nodos de entrada, es decir, dibujar las formas de onda de las entradas. Para ello, tienes que aplicar lo que ya aprendiste en la sesión 1 (recuerda cómo se dibujaban los unos y ceros).


Atención: Solo tienes que establecer los valores en los nodos de entrada, es decir, en *Pis_actual[7..0]* y *Pis_demanat[7..0]*. En ningún caso tienes que poner valores en los nodos de salida (*P*, *PP_amunt* y *PP_avall*). Los valores de los nodos de salida serán calculados por el simulador cuando realice la simulación. Observando los valores de las salidas generados por el simulador, podrás determinar si el circuito se comporta correctamente o, por el contrario, presenta un comportamiento anómalo lo cual sería indicador de la existencia de algún/os error/es en el circuito. En este último caso deberás identificar los errores en el circuito y corregirlos.

Una vez que se hayan entrado todos los valores que se han definido para las entradas (es decir, *Pis_actual* y *Pis_demanat*) en el apartado de **Estrategia de test**, guarda el fichero de simulación (**File** → **Save**) con el nombre **Senyals_UC.vwf** en la carpeta del proyecto (U:\FCPract\4XX_YY).

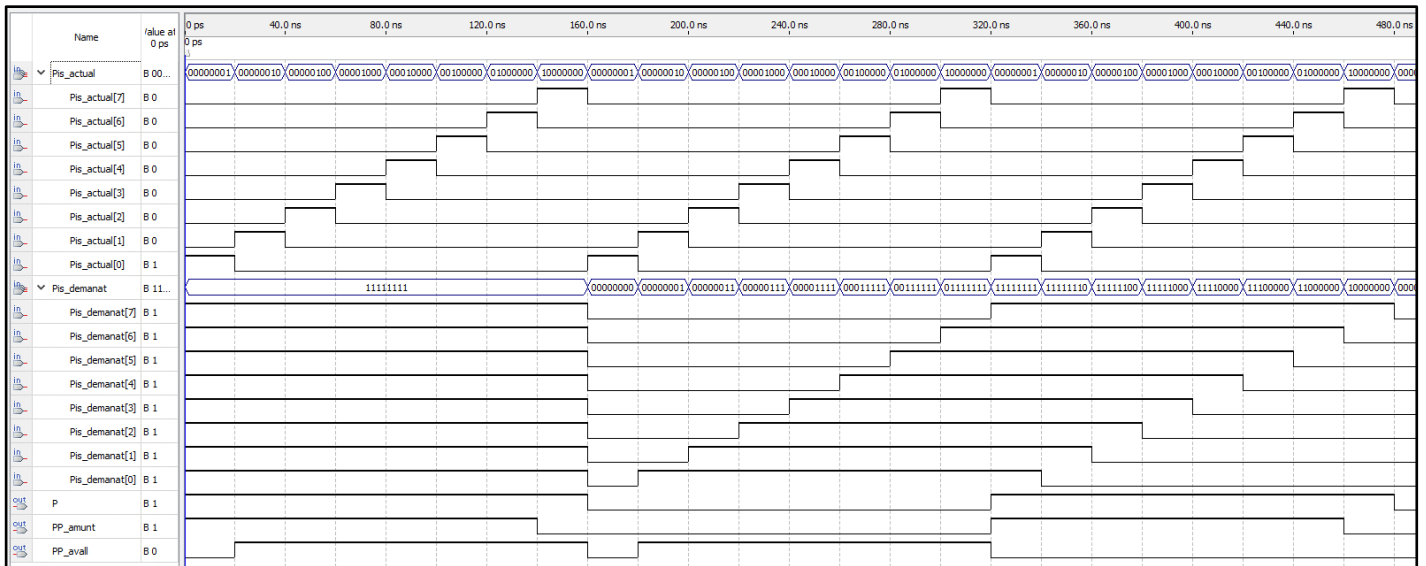
Simulación del circuito

Selecciona el simulador propio de Quartus II (*qsim*) como el simulador que vamos a utilizar:

Simulation → **Options** → **Seleccionar Quartus II Simulator** → **Pulsar OK**

Realiza una simulación funcional. Para ello clica en el botón .

El resultado de la simulación aparece en una ventana como la siguiente, en la que puedes agrandar o reducir la escala de tiempos con el icono de zoom, y en la que puedes moverte a lo largo de la escala de tiempo con la barra de la parte inferior.



Lo único que falta por hacer es comprobar que los valores de las salidas que da el simulador coinciden con los valores esperados en todos los casos. Si no es así, es que el circuito es incorrecto. Identificar las condiciones en las que las salidas no coinciden con las esperadas te dará pistas de donde está el error.

Cuando acabes la simulación, guarda el resultado de la misma ejecutando el comando:

File → Save As ...

En el formulario que aparece **selecciona la subcarpeta `simulation\qsim` del proyecto `(U:\FCPract\4XX_YY\simulation\qsim)`** e indica como nombre del fichero **Senyals_UC_sim.vwf**. Finalmente pulsa el botón **Abrir**.

El fichero con el resultado de la simulación (**Senyals_UC_sim.vwf**) no se considera que forme parte del proyecto y por ese motivo no aparece en la sección *Files* del *Project Navigator*. Para añadirlo al proyecto se ejecutará el comando:

Project → Add/Remove Files in Project...

y se seguirá el mismo procedimiento que se siguió para añadir el fichero **Obrir_Alliberar_sim.vwf** al proyecto en la sesión 1.

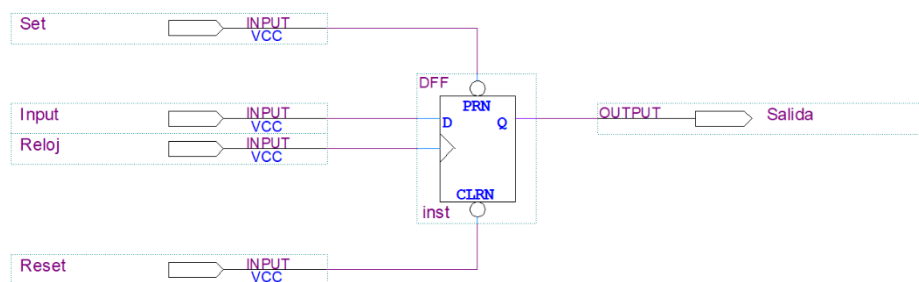
Si has hecho correctamente el proceso ahora se debería ver el fichero `simulation/gsim/Senys UC sim.vwf` en la sección *Files* del *Project Navigator*.

2 EL FLIP FLOP D EN QUARTUS

Si no tienes claro qué es un flip flop D, repásalo, por ejemplo, en los vídeos L5.3 (1/2) y (2/2).

En este ejercicio crearás el esquemático de una celda consistente en un **flip flop D** con entradas asíncronas de **Set** y **Reset**, y lo simularás para comprender perfectamente su funcionamiento. Para ello:


1. **Entra un esquemático** como el de la figura siguiente, consistente en un flip flop D. La celda **dff** la hallarás en la carpeta *primitives/storage* de las bibliotecas de Quartus. Los *ports* de entrada de la celda **dff** son la entrada de datos **D**, la señal de reloj **CK**, y las entradas asíncronas **CLR_N** (Clear o reset negado) y **PR_N** (Preset o set negado). El *port* de salida **Q** nos permite ver el estado del flip flop:

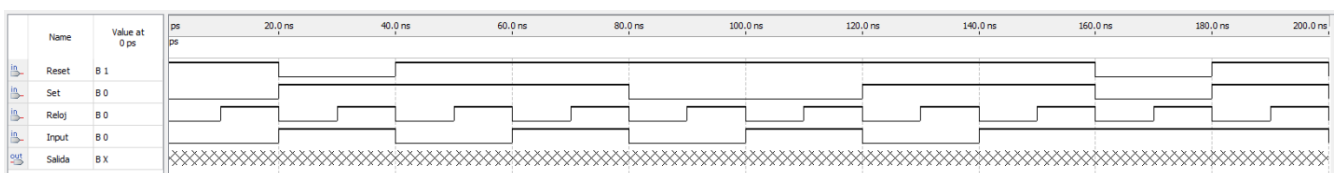


En este flip flop particular de Altera, las entradas asíncronas funcionan de la siguiente manera:

- Cuando **CLR_N** está activa, es decir, cuando **CLR_N=0**, el flip flop se pone en el estado 0, independientemente de las señales **D** y **CK**.
- Cuando **PR_N** está activa, es decir, cuando **PR_N=0**, el flip flop se pone en el estado 1, independientemente de las señales **D** y **CK**.
- En el caso de que **CLR_N** y **PR_N** estén ambas activas, es decir, que ambas valgan 0, el efecto de **CLR_N** tiene prioridad y, por tanto, el flip-flop se pone en el estado 0.

Guarda el esquemático en un fichero con el nombre **FlipFlopD.bdf** en la carpeta del proyecto (U:\FCPract\4XX_YY).

2. **Declara la celda como top-level entity y compila el esquemático.**
3. Construye un fichero de vectores de test como el de la siguiente imagen. No dibujes manualmente la señal de reloj. Para crearla, selecciona la señal Reloj y, a continuación, usa el botón  que genera señales periódicas. En el formulario que se abre indica el periodo de la señal Reloj en el campo **Period** (20 ns).



Guarda el fichero con el nombre **FlipFlopD.vwf** en la carpeta del proyecto (U:\FCPract\4XX_YY).

4. **Simula funcionalmente** el circuito e interpreta, a partir de las formas de onda, el funcionamiento del flip flop *D*. Guarda el resultado de la simulación en un fichero llamado **FlipFlopD_functional_sim.vwf** en la subcarpeta **simulation\qsim** del proyecto (**U:\FCPract\4XX_YY\simulation\qsim**). Añade el citado fichero al proyecto.
5. Haz una **simulación temporal (timing simulation)** y observa las diferencias respecto a la simulación funcional. Guarda el resultado de la simulación en un fichero llamado **FlipFlopD_timing_sim.vwf** en la subcarpeta **simulation\qsim** del proyecto (**U:\FCPract\4XX_YY\simulation\qsim**). Añade el citado fichero al proyecto.
6. Para comprobar que has entendido el funcionamiento del flip-flop D de la biblioteca de Quartus, contesta a las siguientes preguntas (estas preguntas deben ser respondidas en relación al flip-flop D de Quartus de manera general y no haciendo referencia a los casos concretos mostrados en la simulación):
 - a) ¿Qué valor toma el estado del flip flop cuando *Set*=0 y *Reset*=1?
 - b) ¿Qué valor toma el estado del flip flop cuando *Set*=1 y *Reset*=0?
 - c) ¿Qué valor toma el estado del flip-flop cuando *Set*=0 y *Reset*=0?
 - d) ¿Qué valor toma el estado del flip flop cuando *Set*=*Reset*=1 y se produce un flanco de subida del reloj?
 - e) ¿Qué valor toma el estado del flip flop cuando *Set*=*Reset*=1 y se produce un flanco de bajada del reloj?
 - f) Supongamos que el flip flop se halla en el estado 1. ¿Qué valores tendrías que aplicar a las entradas para conseguir que el flip flop se ponga a 0? Si conoces más de una forma, indícalas todas.
 - g) Supongamos que el flip flop se halla en el estado 0. ¿Qué valores tendrías que aplicar a las entradas para conseguir que el flip flop se ponga a 1? Si conoces más de una forma, indícalas todas.

Entrega correspondiente a esta segunda sesión

La entrega consistirá en un fichero .zip cuyo nombre será **FC-S2-4XX_YY.zip**, donde **4XX_YY** es el identificador del equipo de trabajo (por ejemplo, **FC-S2-411_01.zip**), que ha de contener cuatro ficheros:

- el fichero **Senyals_UC.vwf**.

- el fichero **Senyals_UC_sim.vwf**.

- un fichero en **formato PDF** con:

- a) una captura del resultado de la **simulación temporal** de la celda FlipFlopD
- b) una breve explicación indicando qué diferencia conceptualmente una simulación temporal de una simulación funcional
- c) las respuestas a las siete preguntas que aparecen en el punto 6 de la parte 2 de la práctica (la parte relativa al flip-flop D en Quartus)

- un fichero denominado **autoría.txt** con el nombre y el NIU del autor/autores de la entrega.

La entrega del fichero **FC-S2-4XX_YY.zip** se ha de realizar a través del Campus Virtual.