## 2.6 Struttura della Directory degli esempi

Ogni modulo dell’applicazione ha la seguente struttura (che permette di mantenere separati i file sorgente dai file di compilazione):

* **pom.XML**: File di costruzione del Maven.
* **src/main/java**: i file sorgente Java del modulo.
* **src/main/resources**: I file di configurazione del modulo, ad eccezione delle web applications.
* **src/main/webapp**: pagine web, stylesheets, tag files e immagini (solo per le web applications)
* **src/main/webapp/WEB-INF**: i file di configurazione per le web applications.

Quando un esempio avrà molti moduli contenuti dentro un file EAR, le directories dei sottomoduli useranno le seguenti convenzioni:

* **example-name-app-client:** clients dell’applicazione.
* **example-name-ejb:** Files JAR enterpreise bean
* **example-name-war:** applicazioni web
* **example-name-ear:** applicazioni enterpreise.
* **example-name-common:** Libreria JAR contenenti componenti, classi e files usati da altri moduli.

## 2.9 Debuggare le applicazioni Java EE

Questo capitolo spiega come determinare cosa sta causando un errore nello sviluppo o nell’esecuzione di un’applicazione.

## 2.9.1 Usando Server Log

Un modo per debuggare le applicazioni è di guardare al server log nella directory *domain-dir/logs/server.log*.

Il log contiene gli output del GlassFish Server e delle applicazioni.

Posso inserire nel log messaggi da ogni classe Java dell’applicazione con *system.out.println*e, con la Java Logging API e tramite componenti web con il metodo *ServletContext.logI.*

Usando **NetBeans IDE** gli output appaiono nella finestra Output esattamente come i log del server.

Usando il flag *verbose* della **linea di comando** tutti gli output di logging e debugging appiaiono nella finestra del terminale o del prompt e nel server log.

Se si fa partire il GlassFish Server **in background** le informazioni di debug sono disponibili nel log e possono essere visualizzate con un text editor o la Administration Console Log Viewer.

## 2.9.2 Usando un debugger

GlassFish server supporta il JPDA (Java Platform Debugger Architecture) con cui si può configurare il server in modo che comunichi le informazioni di debug usando un socket.