1. As we saw in the article, there was an article warning about the use of safer languages. Do you think this should be applied to language versions as well? As also mentioned, it is possible that "native" code (Code written in a language like C, for example) can be used and cause problems in these languages, so should there also be an article or perhaps some intervention so that there are more updates in software and its versions for greater security?

2. In a distant future, where formal verification is a "standard", would a language that does not use it and allows logical errors, for example, blocking a thread at the wrong time, make the language unsafe by default?

3. Erlang and Elixir choose not to offer pointers and in this sense to be implicit about how resource allocation is done. The good thing is that it brings security to the programmer, but the way of allocation is not explicit. For example, in Go we have reference semantics (being able to pass a value by copy or reference) and in BEAM we do not have this. In other words, we can have inefficient operations happening, and in a code where performance is a requirement, would Erlang/Elixir still be a good choice?