



El futuro digital
es de todos

MinTIC



Ciclo de Vida del Software



Universidad de Caldas

Hola:

Cuando se va a desarrollar *software* se suelen seguir algunas fases como: analizar bien el problema, pensar alternativas de solución y, por supuesto, codificar. Con base en estos enunciados se realizaron pruebas, las cuales se han hecho de una manera informal.

A continuación, se podrá ver que existen unos modelos que definen las fases que se pueden seguir e indican, si se pueden hacer en secuencia o si se pueden hacer al tiempo. Esto no solo ayudará a organizar mejor, sino a tener resultados de mayor calidad, que es lo que siempre se busca cuando se desarrolla *software*.

El desarrollo de *software* por lo general comienza con una idea, ya sea porque se quiere resolver un problema del mundo real, de una persona, de una comunidad o de una organización, pero también porque permite acceder a oportunidades de mercado. En general, a partir de esa idea se hace todo un proceso hasta llegar a entregar un *software* que resuelve esa problemática o atacar una oportunidad de mercado.



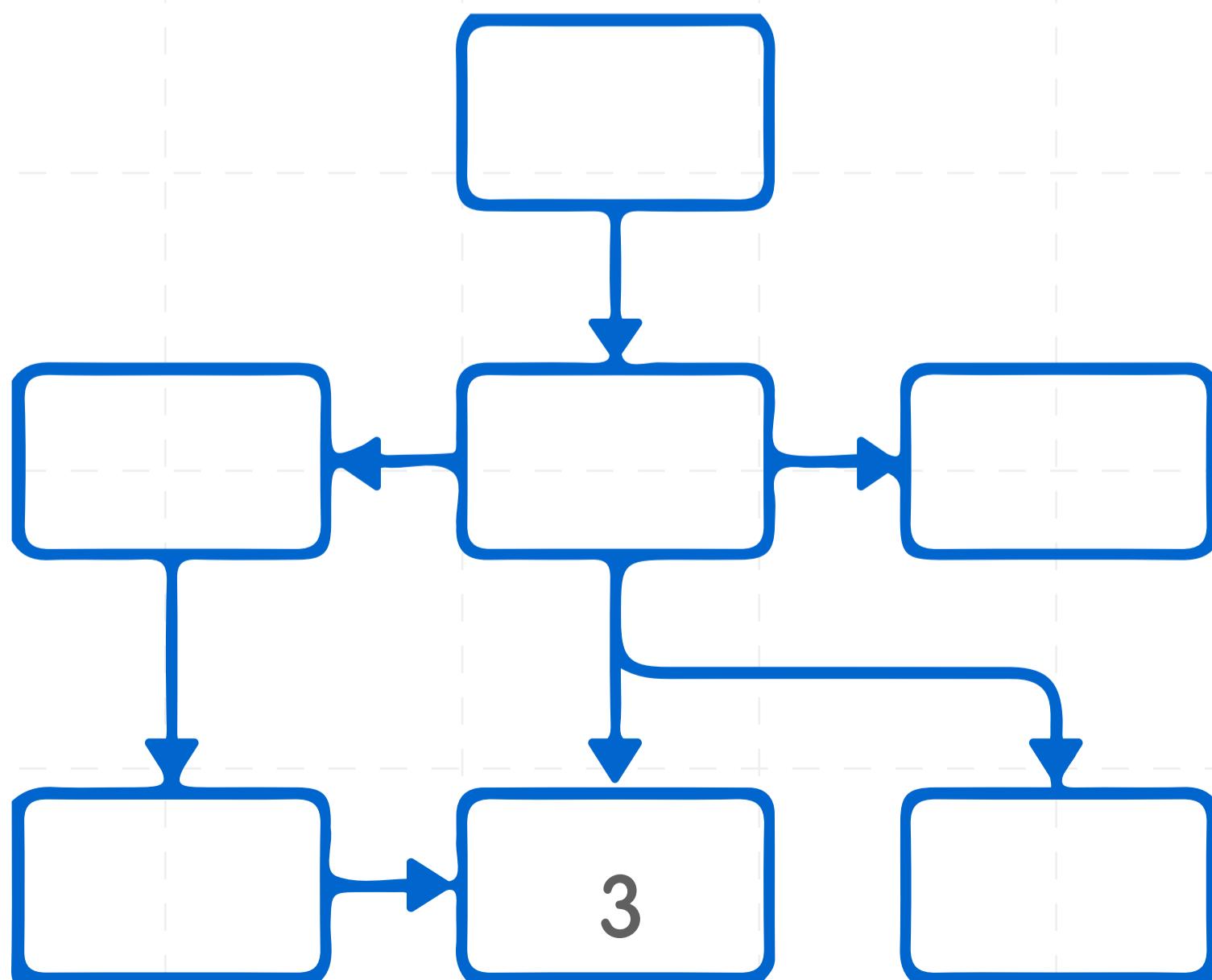
Todo ese conjunto de actividades o fases que se realizan, desde que nace la idea hasta que se entrega el software funcional al usuario se conoce como el **ciclo de vida del desarrollo de software**.

Cuando se empezó con el desarrollo de software, este era un proceso artesanal, un poco de ensayo y error que partía de una idea y se trataba de llevar directamente a la programación. Esto a veces resultaba en una construcción que funcionaba temporalmente, o tal vez algo un poco más estructurado pero que no solía ser muy grande. Esto tenía diferentes problemáticas porque esta forma de construir no son escalables y demandan muchos recursos y experiencia de las personas.

```
$document.on('click', '.check_district_li', function() {
    if($('input', this).is(":checked")) {
        var old_text = $(".choose_city_text").text();
        var remove_text = $(".check_district_lable", this).text();
        var new_text = old_text.replace(" " + remove_text, "");
        $(".choose_city_text").text(new_text);
        var deselected = $('input', this).val();
        var selected_var = $("#division_id").val().toString();
        var new_string = selected_var.replace(deselected + ',', '');
        $('#division_id').val(new_string);
        $(".check_district_lable", this).css("color", "#9da9be");
        $(".customCheckbox", this).removeClass('customCheckboxChecked');
        $('input', this).prop('checked', false);
        $(".loaded_sectors").load("/ajax/load_sectors_bubble.php?district=" + new_string, function() {
            var selected_sectors = $("#sector_id").val();
            var result = selected_sectors.split(',');
            for (i = 0; i < result.length; ++i) {
                $(".loaded_sectors_span_" + result[i]).addClass('sector_checked');
                $('img', ".loaded_sectors_span_" + result[i]).attr("src", "/images/sector_bubble_close.png");
            }
            var sectors = [];
            $.each($(".sector_checked"), function() {
                sectors.push($(this).attr('id'));
            });
            console.log(sectors);
            var sectro_vars = $("#sector_id").val();
            var new_sectors = sectro_vars.replace(sectors + ',', '');
            $("#sector_id").val(new_sectors);
        });
    } else {
        var id = $('input', this).val();
        var old_text = $(".choose_city_text").text();
        var new_text = old_text + ", " + $(".check_district_lable", this).text();
        $(".choose_city_text").text(new_text);
        var selected_var = $("#division_id").val();
        var my_val = selected_var + id + ",";
        $('#division_id').val(my_val);
        $(".check_district_lable", this).css("color", "#363e4a");
        $(".customCheckbox", this).addClass('customCheckboxChecked');
        $('input', this).prop('checked', true);
        $(".loaded_sectors").load("/ajax/load_sectors_bubble.php?district=" + my_val, function() {
            var selected_sectors = $("#sector_id").val();
            var result = selected_sectors.split(',');
            for (i = 0; i < result.length; ++i) {
                $(".loaded_sectors_span_" + result[i]).addClass('sector_checked');
                $('img', ".loaded_sectors_span_" + result[i]).attr("src", "/images/sector_bubble_close.png");
            }
        });
    }
});
```

Lo anterior describe un código que contiene un conjunto de instrucciones muy revueltas que son difíciles de entender, difíciles de mantener y muy riesgosas (están propensas a errores). Por lo tanto, cuando llegaban nuevas demandas era difícil que este software pudiera resolverlas. Es por eso que cuando se tiene que escalar el software, es decir, atender problemas más complejos, o cuando demandan cosas adicionales como seguridad, desempeño, facilidad de uso, se tiene que pensar en formas diferentes de construir. Además, de las exigencias en tiempos de respuesta y, a pesar de que se está haciendo software más complejo, los tiempos se han reducido a través de trabajo en equipo multidisciplinario. Esto quiere decir que hay diferentes áreas que aportan para la construcción de la solución.

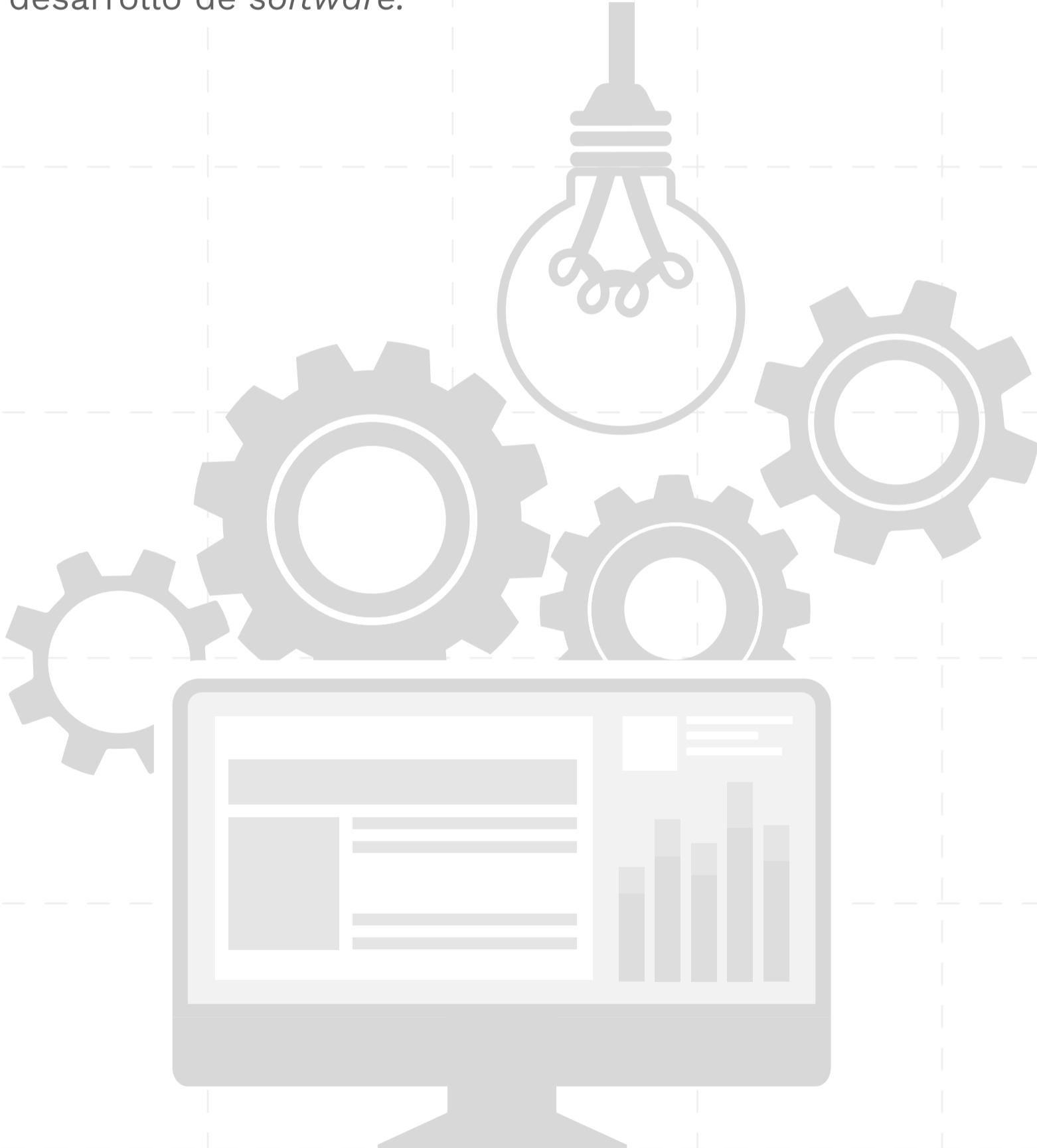
Todo esto ha llevado a pensar que se requiere cierta organización en estos procesos en los que no pueden existir el ensayo y el error, así que con esto nacieron los **modelos de ciclos de vida**, que son los que definen las diferentes fases o actividades que se pueden tener y cómo se relacionan entre sí, es decir, explican cuál fase debe de realizarse primero, cuál debe ser la segunda, cuál es la siguiente, o si se puede hacer algunas en paralelo, si pueden repetirlas o se deben hacer una única vez, etc. Estos tipos de modelos empezaron a ser bastante útiles para el desarrollo de software.



Para poder seleccionar un modelo de ciclo de vida es importante conocer sus características, sus ventajas y sus desventajas.

Algunos de los modelos de ciclos de vida que se usan actualmente son: **Cascada, Modelo en V, Incremental, Prototipos, Espiral y Concurrente.**

Estos no son los únicos, existen otros, pero estos son fundamentales. La mayoría de modelos se basan en estos. Cuatro (4) de ellos serán trabajados en explicaciones posteriores para así tener un panorama de cuáles son características, ventajas y desventajas y de acuerdo a esto se pueda tener una base muy importante para saber qué fases llevar a cabo en un proyecto de desarrollo de software.





Universidad de Caldas