Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Análisis de los resultados de los sistemas de entrenamiento del Pensamiento Computacional

Analysis of the results of Computational Thinking training systems

Autor: Samuel Valcárcel Arce
Tutora: Coromoto León Hernández

Universidad de La Laguna

4 de julio de 2018

Índice

Índice

Índice

Índice

Índice

Motivación

Definición

Podemos entender como **Pensamiento Computacional** *la capacidad del ser humano para resolver problemas, crear sistemas y entender de qué manera se comporta el ser humano.*

Existen diversas plataformas dedicadas a divulgar de alguna forma este tipo de pensamiento:

- Code.org
- Programamos
- Codecademy

Objetivos

Hitos

- El objetivo principal del proyecto fue integrar en la plataforma una herramienta que facilitara al docente observar de manera gráfica el progreso de sus alumnos en los cursos impartidos.

Problemas

- No se supo como acceder a la **base de datos** de la plataforma de Code.org, junto con la constante actualización de su estructura de su página, hizo imposible su integración

Objetivos

- Para solventar el problema del acceso a la base de datos de la página de Code.org, se optó por diseñar una aplicación que simulara la visualización de los resultados de los alumnos.

Plataforma

Algunas de las características que conforman la aplicación son:

- Posibilidad de crear cursos y secciones de los mismo para la realización de las actividades.
- Registrar a los diferentes alumnos en los talleres.
- Visualización de diferentes gráficas representativas de los resultados.

Tecnologías utilizadas

Ruby on Rails, un entorno de código abierto, para el diseño de toda la plataforma. Se podría definir como:

- Usa el estilo de arquitectura Modelo-Vista-Controlador (MVC), que separa los datos, la lógica y la interfaz de la aplicación.
- Evita la repetición de código.
- Permite el manejo de sesiones y de formularios.

Tecnologías utilizadas

Para gestionar la base de datos de la plataforma, se ha optado por **Active Record**, incluido en Ruby on Rails. Se podría definir como la M en el patrón "Modelo, Vista, Controlador"

- Está preparado para su uso en el entorno de desarrollo (development) y de pruebas (test).
- Para el ámbito de producción, se recomienda un software más potente.

Tecnologías utilizadas

El diseño *front-end* de la aplicación se realizó con **Bootstrap**, un framework de código abierto, que hace uso de HTML, CSS y Javascript, de manera que el usuario pueda usarlo como base para el diseño *responsive* de la plataforma.

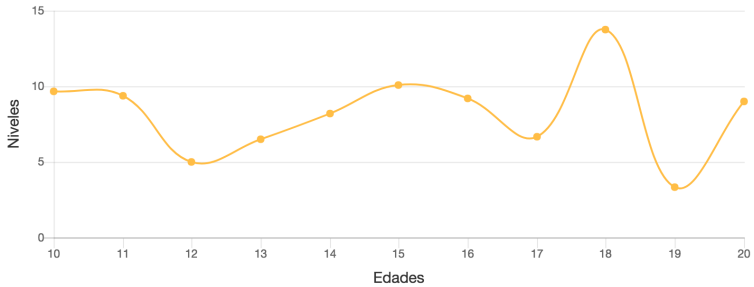
Tecnologías utilizadas

Github es una plataforma de desarrollo en la que cualquier desarrollador puede alojar su proyecto de manera cómoda y sencilla.
Se caracteriza por:

- Colaboración entre los desarrolladores de un proyecto dentro de un mismo repositorio.
- Control de versiones, manejo de ramas, etc.
- Permite la posibilidad de usarlo tanto por línea de comandos, como por su interfaz gráfica de usuario.

Tecnologías utilizadas

Para la visualización de los resultados de los cursos de manera gráfica, se usó la herramienta **Chartkick**, por su compatibilidad con Ruby (y otros lenguajes como Python, Javascript, etc) y su variedad de representaciones.



Introducción

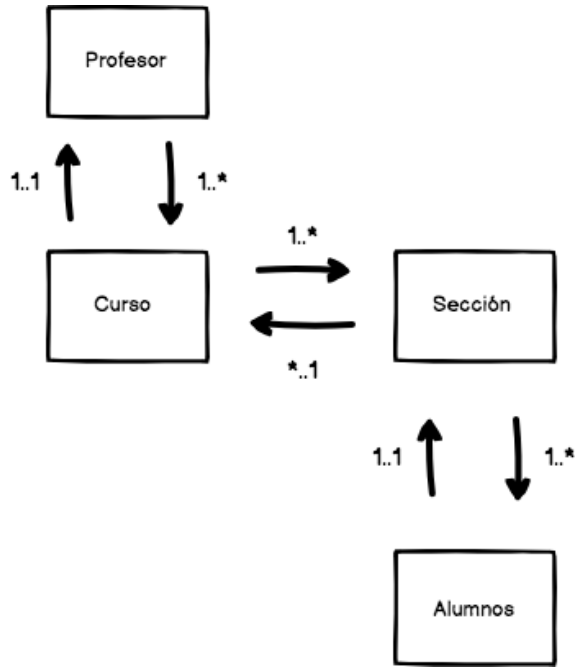
La plataforma CodeCharts se ha diseñado con la intención de simular el funcionamiento de Code.org, de manera que se aporte una representación gráfica a los resultados de los cursos.

Características

- Creación de cursos y sus consiguientes secciones (lugares donde se celebra dicho curso).
- Gestión de los alumnos dentro de cada sección.
- Representación gráfica de los resultados de los talleres/cursos.
- Descarga de informe con las representaciones correspondientes a ese curso o sección.

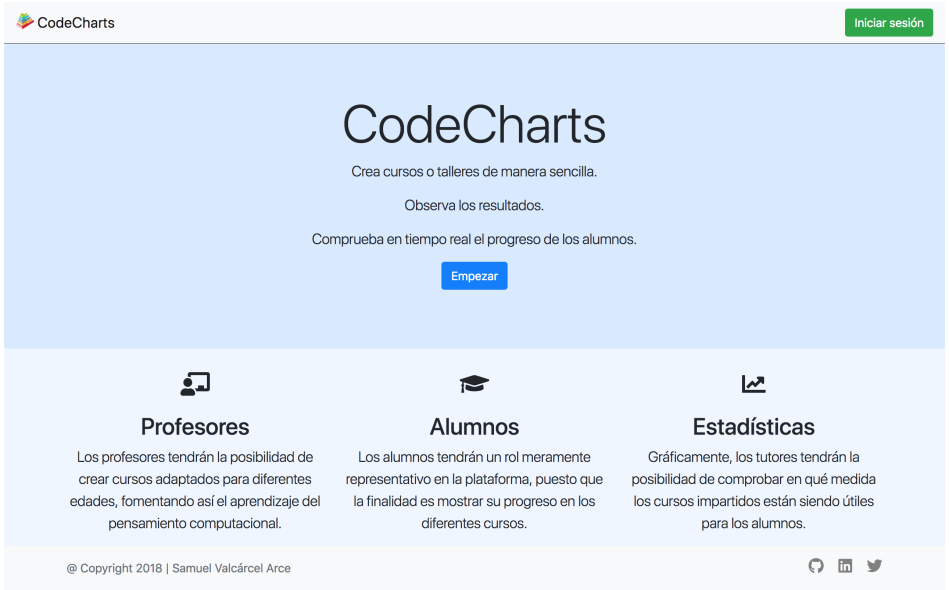
Arquitectura de la aplicación

CodeCharts fue desarrollado en Ruby on Rails, como se menciona en diapositivas anteriores, siguiendo la estructura MVC (Modelo-Vista-Controlador).
Desde un principio se usó una base de datos relacional, que nos proporcionaba Rails con ActiveRecord y SQLite.



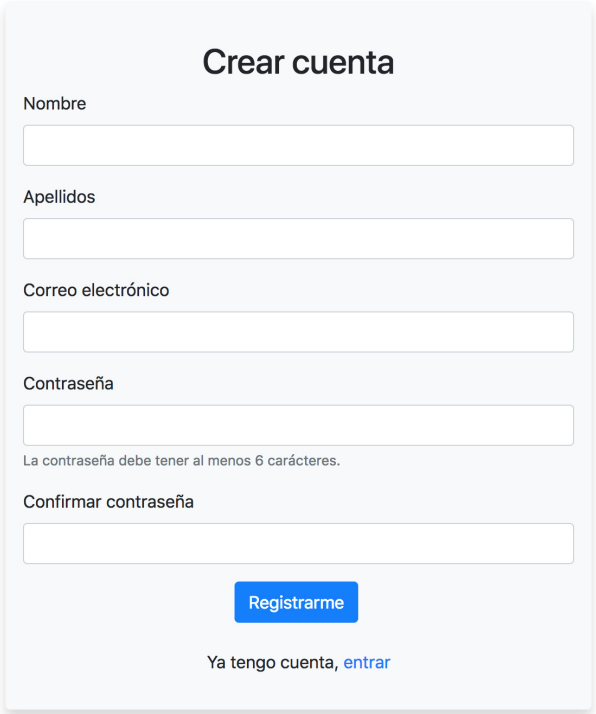
Diseño de la aplicación

La primera vez que se accede a la aplicación, el profesor, que será el que se registre en la aplicación, verá una descripción de en qué consiste la plataforma y lo que ofrece al profesorado en los cursos impartidos.



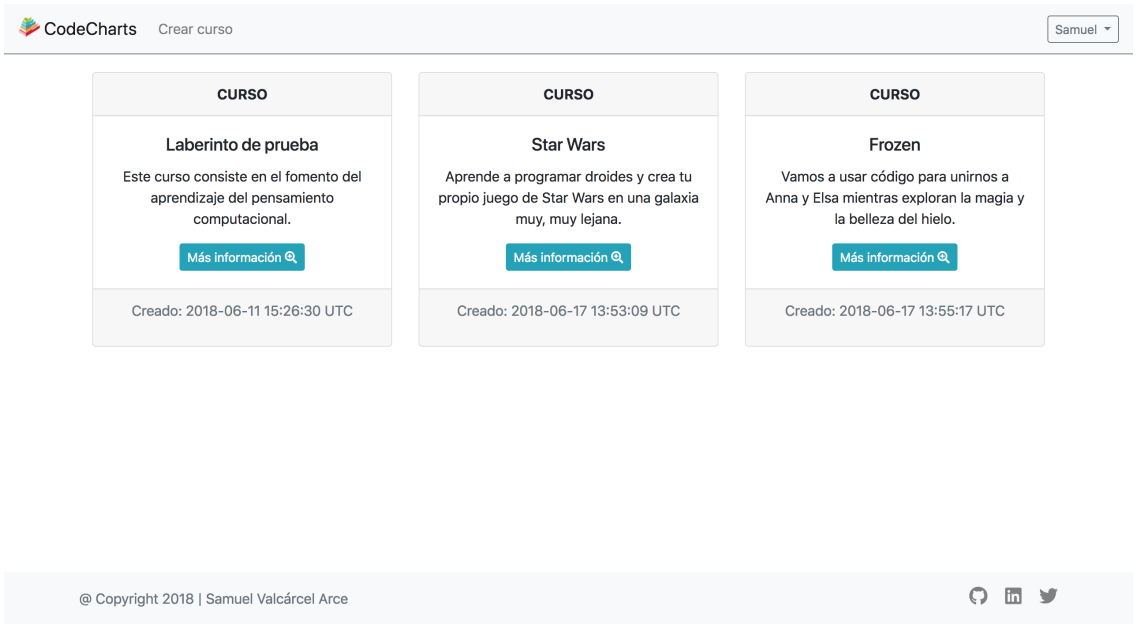
Diseño de la aplicación

El usuario tendrá tanto la posibilidad de crearse una cuenta y, en caso de estar registrado previamente, podrá iniciar sesión con la misma.



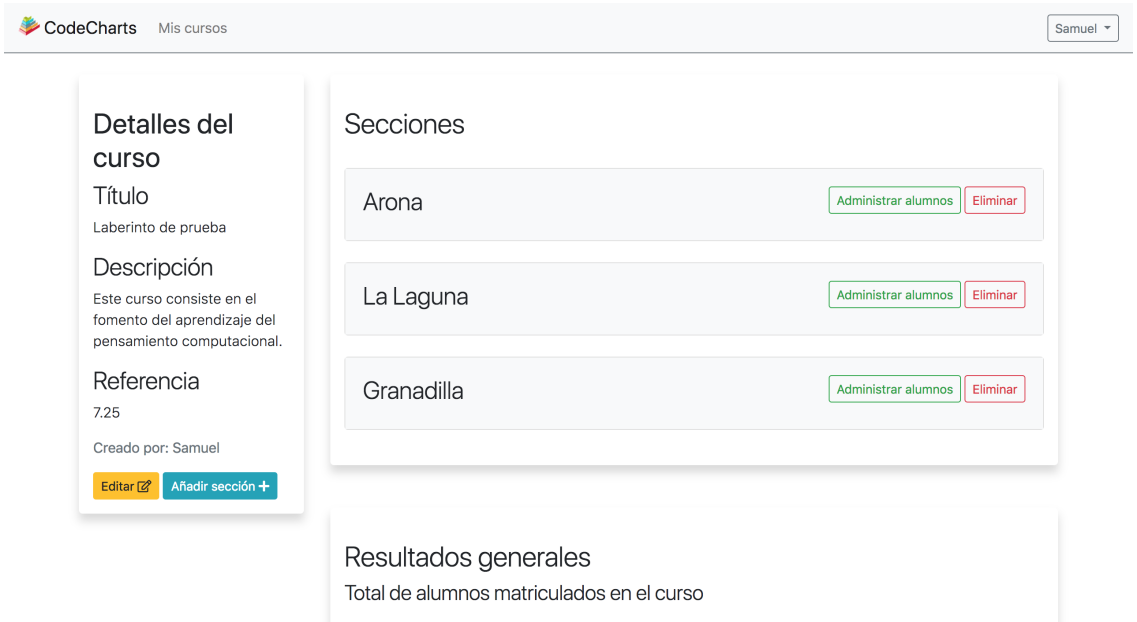
Diseño de la aplicación

Una vez que el profesor haya accedido a la plataforma, tendrá a su disposición los cursos que ha creado previamente en forma de cuadrícula, con su información asociada.



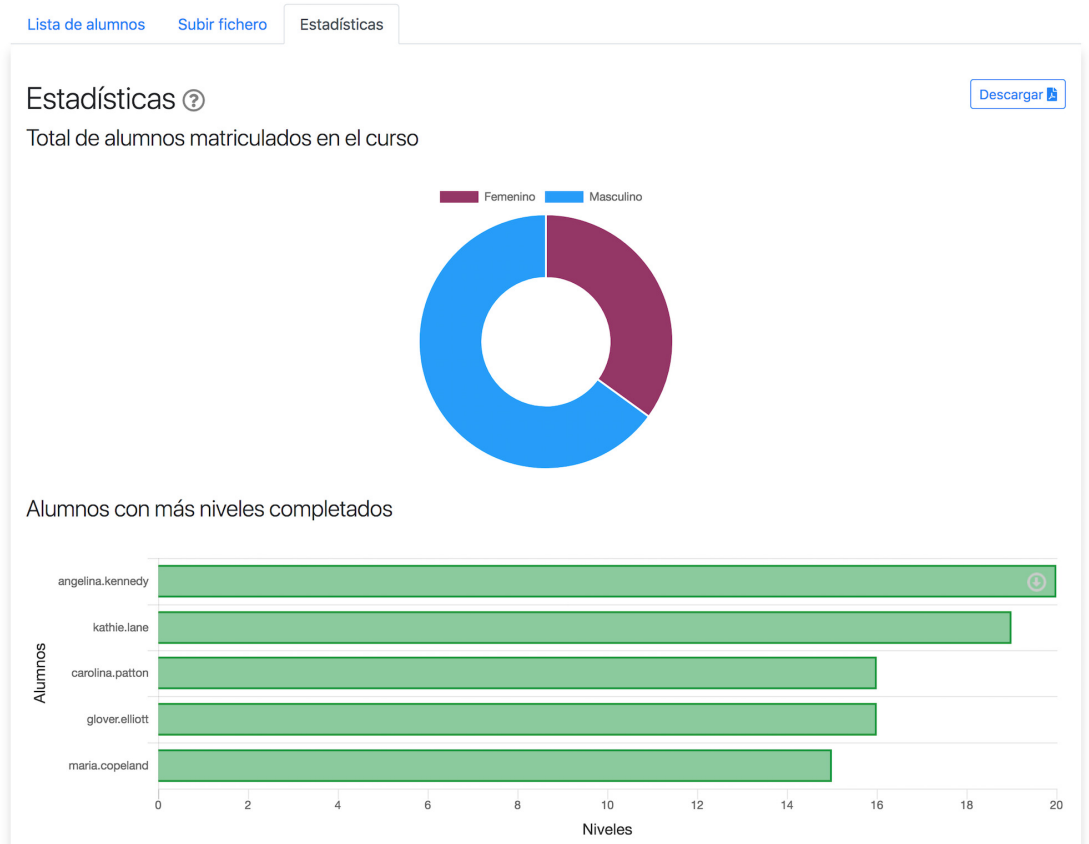
Diseño de la aplicación

Los detalles del curso, junto con las secciones impartidas del mismo, se pueden observar en los detalles del curso.



Diseño de la aplicación

La representación gráfica de los resultados de los cursos/talleres estarán disponibles tanto para verlos en la plataforma, como descargar un informe con los mismos en formato PDF.



Pruebas y verificación

Para llevar a cabo el *testeo* de la plataforma, se usó RSpec, una herramienta utilizada con más frecuencia en el entorno de producción. Se ha verificado algunos de los modelos en la plataforma, como son las secciones, cursos y usuarios.

```ruby
RSpec.describe Course, :type => :model do
    subject { described_class.new }

    it "No se valida sin valor de referencia" do
        subject.value_reference = "Prueba de valor de referencia"
        expect(subject).to_not be_valid
    end

    it "Tiene uno o más usuarios en las secciones de ese curso" do
        should have_many(:users).through(:sections)
    end

    it "Se pueden añadir cursos" do
        @new_course = Course.new(title: "Titulo", body: "Descripcion", value_reference: "10")
        @new_course.save
        expect(Course.find_by(title: "Titulo")).to eq(@new_course)
    end
end
```

```ruby
RSpec.describe Section, :type => :model do
    subject { described_class.new }

    it "Se pueden añadir nuevos usuarios desde fichero" do
        @new_user = User.new(first_name: "Samuel", last_name: "Valcarcel", gender: "Chico", age: "23",
                             username: "cosaca", total_lines: "40", completed_levels: "20", section_id: "1",
                             user_value_reference: "7.5")
        @new_user.save
        expect(User.find_by(first_name: "Samuel")).to eq(@new_user)
    end

    it "Se pueden eliminar usuarios" do
        @new_user = User.new(first_name: "Samuel", last_name: "Valcarcel", gender: "Chico", age: "23",
                             username: "cosaca", total_lines: "40", completed_levels: "20", section_id: "1",
                             user_value_reference: "7.5")
        @new_user.destroy
    end

    it "Se pueden añadir secciones" do
        @new_section = Section.new(name: "Nombre")
        @new_section.save
        expect(Section.find_by(name: "Nombre")).to eq(@new_section)
    end
end
```

Summary and conclusions

Ejemplo

1. *Conclusión 1*

Summary and conclusions

Ejemplo

1. *Conclusión 1*
2. *Conclusión 2*

Bibliografía I

- Code.org. https://code.org/
- Hour of Code. https://hourofcode.com/
- Codecademy. https://www.codecademy.com/es
- Programamos. https://programamos.es/
- Ruby on Rails. https://rubyonrails.org/
- Active Record Basics. http://guides.rubyonrails.org/active_record_basics.html
- Bootstrap. https://getbootstrap.com/
- Chartkick. https://www.chartkick.com/
- Github. https://github.com/
- RSpec. http://rspec.info/

Bibliografía II

- PuntoQ. http://www.bbtk.ull.es/view/institucional/bbtk/Biblioteca_Digital/es
- ACM. https://www.acm.org/publications/magazines
- IEEE. https://www.ieee.org/publications/periodicals.html
- SQLite. https://rubygems.org/gems/sqlite3/versions/1.3.11?locale=es
- Devise. https://rubygems.org/gems/devise
- Gem Bootstrap. https://rubygems.org/gems/bootstrap
- Will-paginate. https://rubygems.org/gems/will_paginate
- Wicked-PDF. https://rubygems.org/gems/wicked_pdf
- Jquery-Rails. https://rubygems.org/gems/jquery-rails

Bibliografía III

- Wilson, Cameron. *Hour of Code: Bringing Research to Scale* http://doi.acm.org.accedys2.bbtk.ull.es/10.1145/2746406
- M. Wing. *COMMUNICATIONS OF THE ACM March* https://www.cs.cmu.edu/{~}CompThink/papers/Wing06.pdf
- Tumlin, Nath. *Teacher Configurable Coding Challenges for Block Languages* http://doi.acm.org.accedys2.bbtk.ull.es/10.1145/3017680.3022467
- Brown, Neil C.C. and Monig, Jens and Bau, Anthony and Weintrop, David. *Panel: Future Directions of Block-based Programming* http://doi.acm.org.accedys2.bbtk.ull.es/10.1145/2839509.2844661